Met Office

Observations

+

The MIDAS Handbook

Who is this document for?	*MIDAS Database Administrators*
Purpose of the document	*Guide to the structure of MIDAS*

Template ver 6.3

Observations

Contents

1 Introduction

The MIDAS Handbook is divided into 2 parts:

- Part A describes the fundamental principles of the database
- Part B illustrates how to use it

The MIDAS database is designed to be extensible and usage is varied and dynamic. This documentation is therefore subject to periodic review.Metnet will provide the primary method of access.
Documentation concerning the source of climatological data, and the processing that is carried out before and after the observations are stored in the database, does not form part of the MIDAS Handbook.

Conventions used in the MIDAS Handbook:

- Code and screen displays are in Courier New font.
- SQL keywords are in UPPER CASE
- Names of MIDAS entities and attributes are in lower case
- Program variables are in *italics*

Comments and suggestions for improving the documentation are always welcome.

1.1 Overview of MIDAS

1.2 Purpose

The main aim of MIDAS was to replace the Climate Data Bank, Marine Data Bank and CLIMAT Data Bank with a single database, using industry standard software wherever possible to:

- eliminate data redundancy
- minimise maintenance
- simplify data security and management
- allow a wider range of retrieval methods

As such it is now the principal archive of meteorological observations, available for answering a vast range of enquiries relating to past weather.

1.3 Scope

Data are routinely added to MIDAS in near real-time, so there is a substantial overlap with the MetDB. However, the MetDB encompasses a much wider range of data types for a much more extensive geographical coverage over a much shorter timescale, its main function being to act as the primary data source for numerical modelling.

The potential uses of MIDAS are much more wide-ranging, often covering periods of months, or even years, rather than days, and for "all times for one site" or "all sites for one time". MIDAS data are also subject to much more rigorous quality control since they are destined for permanent retention as the nation's climatological archive.

Validation and other processes act upon these data to ensure that they are:

- syntactically correct

- internally consistent (e.g. a report of snow with a temperature of 20°C is suspect)
- sequentially consistent (e.g. a vast change of temperature between reports is unlikely)
- really consistent (i.e. conform to "buddy" checks with near neighbours) and are within reasonable ranges

1.4 Test Database

The implementation of MIDAS is such that changes of design and usage can be accommodated much more easily than before. Therefore, to protect users when further development is under way, there is a 'test' database. This has the same structure as the production system, but has a limited amount of space for observational data.

1.5 User Group

As well as being used for a wide range of enquiries, the database serves a range of users with quite different backgrounds and requirements. The MIDAS User Group exists to ensure that the needs of **all** these users are met.

2 Relational Database Concepts

2.1 Fundamentals

At its simplest, a database is a collection of data, usually integrated to eliminate redundancy (where possible). It can take several forms:

- hierarchical - where each record can be represented in a tree structure
- network (or CODASYL) - where one record type is defined as the owner of another
- relational - where data are stored in one or more tables whose contents can be related

MIDAS uses the Oracle relational database management system.

2.2 Relational Databases

It is important to realise that the heart of a relational database is a set of tables, where a table is made up of a set of rows, each of which consists of a list of data items, in columns. Each table in a database must have a unique name to identify it, preferably in some meaningful way. For example the table geographic_area in MIDAS includes the following columns:

geog_area_id	geog_area_name	geog_area_type
NYKS	NORTH YORKSHIRE	COUNTY
STRA	STRATHCLYDE	SCOTTISH REGION
UK	UNITED KINGDOM	POLITICAL REGION
...		

A table has the following properties:

- There are no duplicate rows

- Each row represents all the information stored in the database for (in the case above) a single area and for that area alone (e.g. North Yorkshire)
- There are no duplicate column names
- Each column represents just one item of data that is stored in the database for each area (e.g. name)
- For each column, all data values are of the same type (e.g. geog_area_ids are character strings)
- The order of rows or columns is not significant

Physical considerations (such as storage and retrieval via indexes, and alignment on word boundaries) will require a particular order for rows and columns in the database, but a user does not need to be constrained by the actual implementation. Efficiency may sometimes dictate a method that takes advantage of the storage strategy, but data can be retrieved in any order.

A relational system also has a set of operators so that the user can retrieve and manipulate data. These are not described here because they are well covered by standard texts on relational databases. In particular, Part B of this Handbook will make reference to SQL (Structured Query Language) for access to data, but only for illustrative purposes.

2.3 Entity Keys and Relationships

Attributes, or combinations of attributes can be defined as a key of an entity in order to:

- uniquely identify an instance of the entity (each row in a database table must be unique)
- show relationships between entities
- provide a useful way for users to find data.

Primary Key

2.3.1 Primary Key

A Primary Key is an attribute (column), or group of attributes, that exist on a table allowing us to uniquely identify a single row (tuple/record). This value is a determinant, in that from knowing a single value, or subset of values, it allows us to determine all the other attribute values associated with that, and only that row. (e.g. geog_area_id for the table geographic_area).

2.3.2 Foreign Key

Foreign Keys are a functional form of controlled redundancy. The same attribute may appear on two tables. Generally this is the primary key of one table appearing as an attribute on another table. This serves to define a relationship between these two tables with the primary key of one table acting as a foreign key to another table. Section 5 gives more details (e.g. values of loc_geog_area_id for the source table form a link with the geographic_area table using its geog_area_id column).

2.3.3 Alternate Key (or Super Key)

An Alternate Key includes the column/columns used to uniquely identify a row other than by the primary key. Both primary keys and alternate keys are unique.

2.3.4 Non-Unique Key (or Secondary Key)

A Non-Unique is a column, or combination of columns, than can locate several rows in the database. They are mostly used for data retieval and usually contain values that are easier to remeber than the unique keys.

2.3.5 Surrogate Key

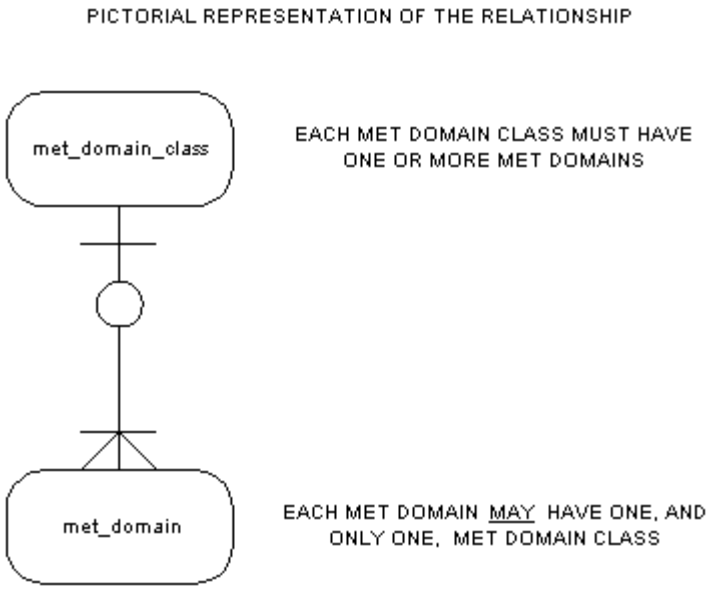
A Surrogate Key is where a simple value (e.g. a unique integer) is substituted in place of a more complex structure (e.g. several attributes concatenated together to create a unique combination).

Keys can be used to define indexes. Just as in a book, an index is an ordered list which indicates where to find items, but with a database it is often useful to provide more than one index for the contents of a table.

MIDAS stores most observational data using an index on the primary key, but additional indexes have been included where analysis shows they will be helpful entry points for users. Users cannot specify these indexes in their queries, but it can also be beneficial to keep the concepts in mind for SQL applications because the database management system uses an optimiser to select the most efficient retrieval strategy, and careful specification of a query can result in an index being used, with obvious performance benefits.

2.3.6 Relationships

The logical data structure diagram in Section 8 shows the relationships between entities; in most cases a foreign key is used. Consider the examples of met_domain and met_domain_class.



met_domain_class	**met_domain**
met_dom_class (PK)	met_domain_name (PK)
….	met_dom_class (FK)
	….

Met domains are grouped together in classes such that each class has one or more members (met domains) and each met domain belongs to only one class. Each class has an identifier, met_dom_class, which forms the primary key (PK) of the entity. Including this identifier as a foreign key (FK) attribute (non-unique) of the met

domain makes the relationship between the two entities. So, if you need to know more about the class, you can use the met_dom_class attribute to retrieve the appropriate met_dom_class record. Similarly, if you start with a met domain class, you can find all the met_domain records which have that value of met_dom_class. Section 5 elaborates on the design of keys for tables containing observational data, and also on making relationships between tables.

3 The Basic Design of MIDAS

3.1 The Data

The MIDAS database contains three main groups of tables:

- Data sources
- Data definitions
- Meteorological data

3.1.1 Data sources

These tables give information about the places where the observations were made, the way these places are identified, the facilities at the sources, and the regions in which they reside.

3.1.2 Data definitions

These include definitions of the meteorological elements in the database, the groups of elements stored in the database, and the types of report made by the sources. Each group of elements is called a met domain. A domain may be a group of elements in a message as well as a group of elements stored in a table.

3.1.3 Meteorological data

The meteorological data also breaks down into three groupings. They are:

CLIMAT	Normals (surface and upper air) Records (surface and upper air)
Marine based observations	Currents Ship SYNOPs, Light Vessel Reports, Marids, OWS etc. Upper Air
Land based observations	SYNOPs, METARs, F3208, NCMs Radiation / Sunshine Rainfall Soil Temperatures Upper Air Winds

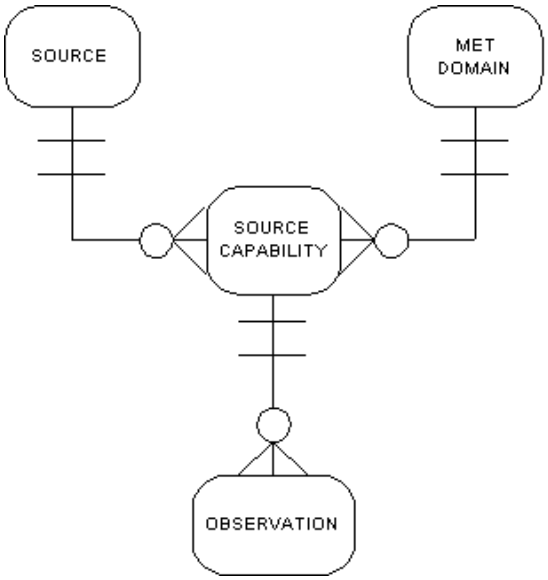
These three groups adequately describe the range of meteorological information to be found in MIDAS, but not the method that MIDAS uses to store this information. MIDAS stores meteorological values, which are grouped into tables. Each table consists of parameters that have similar characteristics, or attributes. MIDAS does not store meteorological messages; although it does hold information about the type of message that supplied the

data. Refer to Sections A6 and A9 for details of what is actually stored for each meteorological data-type.

The database design includes one particularly important table (src_capability) that links all this information together. It can be used to determine the ability of a station to produce meteorological reports of a specified type, when that ability started, if it is still current and where those reports are stored in the database.

3.2 The Structure

MIDAS has a Logical Data Structure diagram and a Physical Data Structure diagram (see Section 8), but for the moment we will use a simplified, conceptual diagram, which shows the nature of the database in the simplest form.



Key:

Rectangles represent entities (tables in the database).

Lines are relationships between tables.

A "crow's foot" indicates a one-to-many relationship

e.g. a source may have many capabilities, but a capability can be for only one source.

A 'O' indicates that the relationship is optional whilst a solid bar '--' indicates a mandatory relationship.

e.g. a source may have one or more capabilities, but a capability must be for one and only one source.

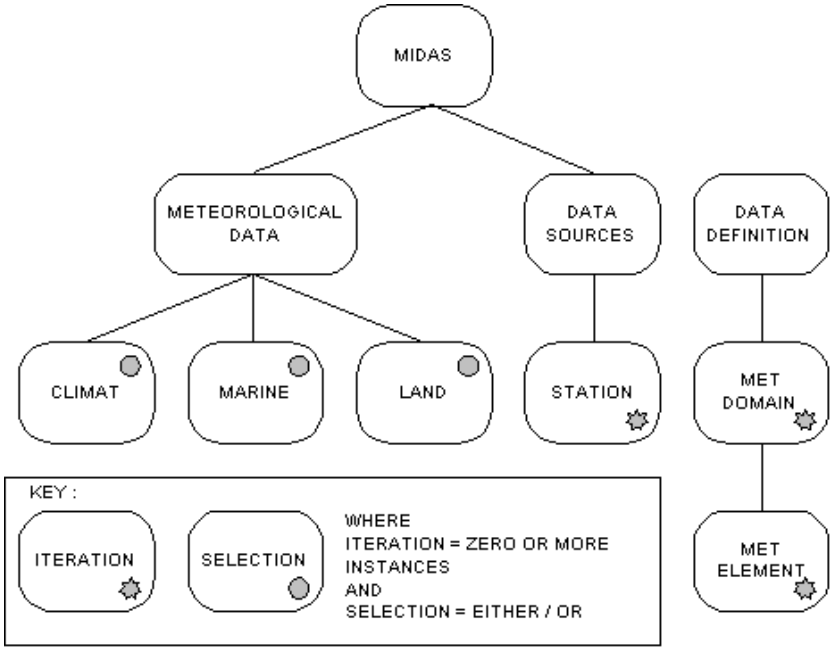
The brief description of these entities is:

Source
Location (position on the earth's surface, name, period of validity) e.g. Heathrow

Met Domain
A collection of meteorological elements e.g. SYNOP

Source Capability
Capability of a particular source to report data according to a specified met domain associated with a particular identifier

Observation
Observed values
To summarise, we can use a structure diagram to represent the nature of MIDAS.



4 Observations and Observation Sub-Types

4.1 Observations, Messages, Reports and Tables

It is convenient to distinguish between:

- an **observation**, a group of one or more meteorological elements, plus some location and time information; other data items are added to an observation before it is encoded into a message.
- a **message**, meteorological information encoded into a pre-defined format for exchange between meteorological centres
- a **report**, some form of summary of (meteorological) data.

Data contained in messages and reports are stored in **tables**. They are structured to facilitate the extraction of selected data from the archive, the selection being based on location, time and meteorological element (where an element is a single meteorological variable such as wind speed, cloud cover or relative humidity). The tables do not store the incoming reports and data files, just the information held by them. (The tables do hold information on how the data were received.)

4.2 Specification of a Data Value

The value of a single meteorological element is only of use if we can fix it in both space and time. Often, several elements may share the same 'fix' and may be stored by MIDAS in the same record. The horizontal spatial co-ordinate system is either referenced to the earth's geometry (latitude and longitude) or to a Cartesian (regular square) grid such as the Ordnance Survey Grid of 10 km squares.

Station identifiers
It is often convenient to refer to a given location on the earth's surface by means of a short-hand notation such as a station name or some other identifier. However, a single location may have several such short-hand references, and a single identifier can refer to more than one location! Therefore, to refer to a location in this way

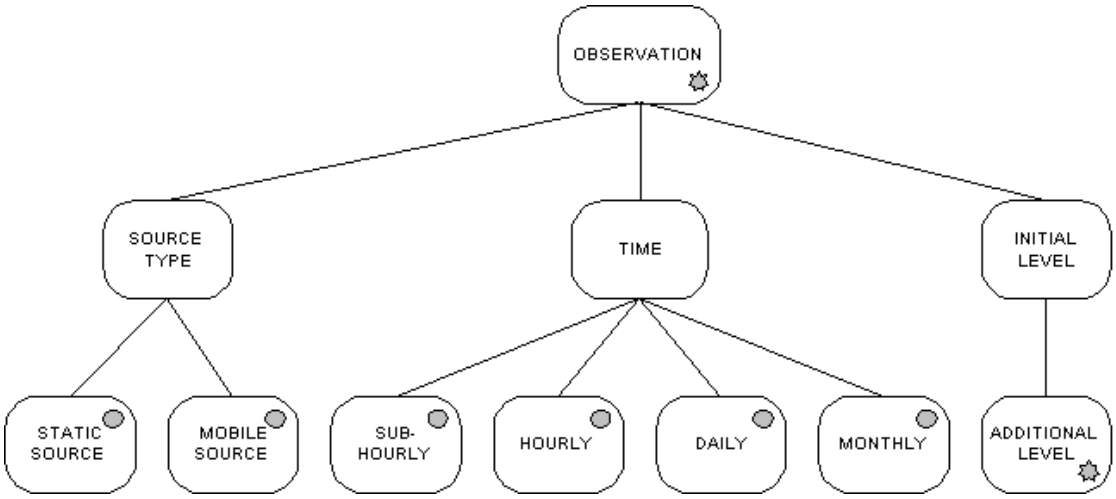
it is necessary to specify the combination of identifier-type and identifier. MIDAS uses its own unique short-hand notation - the src_id, which it associates with the different naming conventions.

Mobile sources of meteorological data have an identifier that is clearly not associated with a fixed location! The spatial co-ordinate information for these sources has to be included with the individual records of meteorological elements. The vertical space co-ordinate may be geometric height, geopotential or pressure level in the atmosphere; geometric depth below the lower boundary of the atmosphere.

MIDAS uses conventional time specifications with a resolution of one minute, using the Oracle DATE data-type.

Each of these space-time ordinates may be a single value, such as 51013□ N or 1135 on 04/11/1978. They can each also be extended: an extent in horizontal co-ordinates is used to define an area (as in radar rainfall elementary areas) while an extent in time may be used to indicate an average or integrated value, (such as the daily rainfall amount) or the period within which a meteorological element occurred, (such as the overnight minimum air temperature). When a time co-ordinate is extended, MIDAS uses the convention that the period is specified by the time of the later end of the period and the duration of the period. MIDAS does not yet contain data that have extended spatial co-ordinates; it is probable that the convention that will be adopted is for the extent to be specified by the lower value and the increment in that co-ordinate value - e.g. {52010□ N, 20□} for 52010 □ N to 52030□ N. (Note that these definitions may not be consistent with previous practices!)

This structure diagram shows the nature of the meteorological values in MIDAS.



Summary:

- In MIDAS, meteorological values are classified according to the type of reporting source. They may be for static sources (e.g. land stations) or from mobile sources (e.g. ships, buoys etc).
- MIDAS contains meteorological values on several time scales (e.g. monthly, daily, hourly or sub-hourly)
- The vertical co-ordinate is significant in the way that MIDAS stores meteorological values. All observations will have one or more levels. The default level will be the surface. Additional levels may occur; if they do, the vertical co-ordinate (e.g. height or depth) is part of the key of the values

4.3 Observation Sub-Types

MIDAS stores observations in tables that either correspond directly to the observation, or to a sub-set of the observation, which means that some reports are separated into several tables. Equally, some of the MIDAS tables correspond to more than one type of observation (e.g. the MIDAS table weather_hrly_ob can contain data

extracted from both SYNOP and METAR messages). The general name for these tables is "Observation sub-types". Observation sub-types provide some independence and isolation from the report format.

- They divide the logical "observation" entity into a number of sub-groups
- Each sub-type is a table of related attributes, influenced by report type and volume
- The logical entity "observation" may map to more than one table
- They give a manageable number of tables and columns, reduce space wastage, and control data redundancy

Section 6 shows how meteorological reports map to MIDAS tables. Details of the tables are in Section 9.

4.4 General Ideas about Observation Sub-Types

4.4.1 Keys

- Observation sub-types are stored via a unique index on the primary key
- A non-unique index permits retrieval of all observations for a specified place
- The primary key of marine observations have both position and identifier

src_id
Each land observation location (source) is allocated a unique number (src_id) that has no significance other than to identify that location in a very simple manner. Each observation table includes the src_id attribute so that an observation (report) distributed over more than one entity (e.g. F3208) can be reconstituted, and static details of the location can be retrieved.

version_num
Most of the observation sub-type tables contain the attribute version_num. This is to allow the data maintainers to store more than one occurrence of a row, as a result of Quality Control (QC).

version_num = 0	Original observation, as reported. A version 1 probably exists as a result of QC (QC fields showing level of QC attained before amendment became necessary)
version_num = 1	Either an original observation, as reported (or correction received before QC), or a QC improvement of it. A version 0 may exist.

Occasionally a version 0 is created but no version 1 is provided, if the original observation is suspect but no equivalent replacement value is available. If the observation sub-type does not have any QC, then the table will not have a version_num column.

4.4.2 Quality Control

As well as meteorological values, observation sub-types can contain additional attributes which store information about the quality control that has been applied. These occur as met_element_name_q and met_element_name_j (i.e. the attribute name depends on the observation sub-type).

met_element_name_q is a five digit number (of the form MESQL), where each digit describes one aspect of the quality of a meteorological element.

M	qc_marker	A combination of two flags to indicate accumulations, and/or a trace of rainfall
E	qc_estimate	Indicates 1 of up to 8 remarks about an estimate
S	qc_status	Indicates 1 of up to 8 descriptions of the value (e.g. reason for suspecting a value)
Q	qc_query	Indicates 1 of up to 8 statements about the original value (e.g. reason for correction)
L	qc_level	Indicates which of 10 possible stages of QC has been reached

Details of the various components of met_element_name QC are given in Section 14.5.1. The values are also documented in the code_detail table of MIDAS (see Section 14.5.3).

met_element_name_j is a single character code which either describes the method of measurement, or further qualifies the meteorological values. The meaning of any value depends on the element being qualified and details are given in Section 14.5.1). The values allocated to this item are also documented in the code_detail table of MIDAS (see Section 14.5.3).

4.4.3 State Indicators

Each of the observation sub-types contains an attribute that is used to describe the current stage in the life of a particular record, from creation to deletion. Values are listed in Section 14.5.2).

4.5 Source Capability and Observations

The conceptual diagram in Section 3.2 shows that the src_capability entity provides the link between the three main components of MIDAS.

- meteorological information
- sources of meteorological information
- definition of the meteorological information

A 'source capability' is the ability of a specified station to report observations of a specified type, using a specified identifier. It shows when the station first gained the ability, and if the station still has the capability. When the enquirer only knows the station and the type of report, the src_capability entity can be used to find in which table(s) the reports are stored. It can also be used to find out if there are likely to be any observations of the required type for the required spot. It is clear that src_capability is a very important entry point into the database.

5 Entity Keys and Relationships in MIDAS

Section 2 provided some insight into why we define entity keys. We will now examine some MIDAS keys in a little more detail, but we will still use general examples rather than specific ones, e.g. "observation" refers to a land observation or a marine observation, and of course a land observation could be one of many types. For specific examples, refer to Section 10, which lists the keys of each of the MIDAS entities.

5.1 Keys of Land Observations

The primary key of a land observation contains a date (possibly including a time), and an identifier (e.g. a station number). The relative significance of attributes within the primary key (how near they are to the "start" of the record) can affect the retrieval strategy, and how quickly a specified observation will be found.

Most of the meteorological data in MIDAS will be partitioned, i.e. data are located in the database using a partition key. For most 'observation' tables, the data is partitioned according to the date and time of the record so

rows will be retrieved most efficiently by specifying the date and time (or a range of dates and times). Additionally, glbl_wx_ob is also partitioned by wmo_region_code, and marine_ob is partitioned by longitude_band_code. Refer to Section 7 for further details of the partitioning.

Always specify the id_type (e.g. DCNN, WMO, etc.) with an id. It uniquely specifies the identifier and also helps the SQL optimiser to choose a suitable index for efficient retrieval. A non-unique index allows retrieval by src_id.

5.2 Keys of Marine Observations

The primary key of a marine observation contains a position, date and id. The id is needed because we can have two ships at the same position. You should specify the id_type, which in this case will probably be either SHIP or BUOY.

- Date and time is most significant (i.e. occurs first) in the primary key
- Latitude and longitude occur second and third in the primary index
- longitude_band_code is the partition key. It is a single character 'A' - 'J' that allows the database to cluster the observations effectively. Always specify this if you can
- marine_ob has an index on id_type, id and ob_time, to allow retrieval of a series of observations for a specified ship

5.3 Relationships between Entities

The logical data structure diagram in Section 8 shows the relationships between entities. In Section 2.3 we saw an example of a foreign key relationship and, in most cases, relationships in MIDAS are formed in this way. Consider another example :

5.3.1 source and geographic_area

Each source record contains a loc_geog_area_id to indicate the geographic area in which it is situated. For any specified loc_geog_area_id, you can find all of the sources in that area. There is a potential problem that a source could appear to be in more than one geographic area, e.g. Beaufort Park is in Berkshire, England and UK. This problem is circumvented by ensuring that the source is connected at the lowest level, i.e. Beaufort Park contains the loc_geog_area_id of BRK (for Berkshire). The relationship to England and the UK is made using the relationship between the geographic_area and geog_area_hier records.

6 Organisation of Data in MIDAS

6.1 Tables

Section 3.1 explained that MIDAS has a distinct grouping of tables. Some hold current information about the data sources and the types of data that are available, known as 'standing' data, while the remainder contain the permanent archive of data from the observation sub-types.

Standing Data

`acquisitions_log`	logs recent land data acquisitions, where QC is not yet complete
`code`	describes the code types used in `code_detail`
`code_detail`	describes the decoded values for each entry in a code table
`cross_reference`	defines a relationship between one source and another
`domain_element`	defines a meteorological element within a `met_domain`
`geog_area_hier`	defines the relationship between geographic areas
`geographic_area`	defines geographic areas in which sources are located
`hydrometric_area`	details of UK Met Office hydrometric areas

identifier	alphanumeric character strings used to identify a source
identifier_type	defines the various types of identifier that may be used
longitude_band_code	defines the range of longitude in each band
mar_acq_log	logs recent marine data acquistions, where QC is not yet complete.
met_domain	defines groups of meteorological elements
met_domain_class	details about the general types of met domain
met_domain_rank	details about the ranking of met domains for an observation
met_elem_class	classifies met elements, to simplify access to their details
met_element	defines met elements independently of domains
source	details of the location where observations are made
report_table_mapping	lists the types of report and the tables where they are stored
src_capability	defines which types of observation a source is capable of making
src_remark	contains textual remarks concerning the source
src_runway	details of the runways at a specified source

See Section 9 for a full description of each table.

acquisitions_log and mar_acq_log are written to whenever a new observation is added, but all other standing data tables are updated as and when required.

Observation Sub-Types

clm_sfc_rec	monthly calculated values CLIMAT from globally located sources
clm_ua_norm	long period surface averages for a CLIMAT upper air station
clm_ua_norm_lvl	CLIMAT upper air standard isobaric surface check values
clm_ua_rec	surface values for upper air CLIMAT reports
clm_ua_rec_lvl	elements from a CLIMAT upper air report at a standard isobaric surface
glbl_wx_ob	3-hrly SYNOPs and METARs for selected overseas stations
marine_current	direction and speeds of marine currents
marine_ob	marine meteorological values
pollen_drnl_ob	pollen grain counts
radt_ob	radiation amounts currently being reported
rain_drnl_ob	rainfall amounts for one or more 24 hour period(s) ending at the specified time
rain_hrly_ob	rainfall amount and duration during the hour(s) ending at the specified time
rain_subhrly_ob	rainfall tip amounts with a time resolution of one minute, from SSER
runway_ob	runway visual ranges measured at the hour specified
soil_min_temp_ob	bare soil minimum temperatures recorded at 09Z each day
soil_tempo_ob	soil temperatures reported daily and hourly
temp_drnl_ob	maximum and minimum temperatures measured over a period of up to 24 hrs
ua_sounding	the non-repeating (i.e. surface) items in an upper-air ascent
ua_sounding_point	the points of an upper-air ascent
weather_drnl_ob	meteorological values measured on a 24 hour time-scale
weather_hrly_ob	SYNOPs and METARs measured during the hour ending at the specified time
wind_mean_ob	wind direction/speed for one or more hours ending at the specified time

See Section 9 for a full description of each table.

These tables are updated whenever appropriate new reports become available (see Section 6.3 below).

6.2 Database Views

SQL can be used to form 'virtual' tables, known as views. To reduce the impact of database changes on users, MIDAS does not allow direct access to the database tables. All access is via database views. These are defined as combinations of attributes from the 'base' tables mentioned above. They can be temporary (created purely for a specific query and then deleted), but a number have been developed for general use.

The database views are grouped into schemata, where each schema is for a defined purpose:

MIDASUPD One-for-one projection of the columns in the base tables. These views are updateable (by authorised users) and do not project any values other than those in the base tables.

MIDASVU The general-purpose schema. This schema contains all the views in the MIDASUPD schema, but they are not updateable. Additionally, the schema contains a series of views designed to simplify end-user extraction (e.g. src_hrly_rain).

A full definition of each view can be found in Section 9 of this Handbook.

6.3 How Reports Map To Tables

This section lists the reports stored in MIDAS, and the name of the table(s) used to store them.

Met Report	Contains	Stored In
CAWS (from CDLs)	Hourly wind	wind_mean_ob
	Hourly temperature & humidity	weather_hrly_ob
	Hourly rainfall	rain_hrly_ob
	Hourly radiation	radt_ob
	Hourly sunshine	weather_hrly_ob
	Daily temperature (09-09Z or 09-21Z & 21-09Z)	temp_drnl_ob
CLM surface normal	Monthly averages for values at CLIMAT surface stations	clm_sfc_norm
CLM upper air normal	Monthly averages for surface values at CLIMAT upper air stations	clm_ua_norm
	Monthly averages for isobaric surface values at CLIMAT upper air stations	clm_ua_norm_lvl
DALE	Hourly mean wind derived from minute mean of wind direction & speed	wind_mean_ob
ESAWS	Hourly radiation	radt_ob
	Hourly soil temperature at 10cm	soil_temp_ob
	Hourly wind	wind_mean_ob
FM 12-VII SYNOP	Hourly weather values	weather_hrly_ob
FM 13-IX SHIP	Hourly marine weather values via GTS	marine_ob
FM 15-V METAR	Hourly weather values	weather_hrly_ob
	Runway visual range	runway_ob
FM 18-X BUOY	Report of a drifting-buoy ob	marine_ob
FM 71-VI CLIMAT	Monthly calculated surface values from global sources	clm_sfc_rec
FM 71-X CLIMAT	Monthly calculated surface values from global sources	clm_sfc_rec
FM 75-VI CLIMAT TEMP	CLIMAT upper air report	clm_ua_rec
		clm_ua_rec_lvl
Light Vessel	Reports from Light Vessels	marine_ob
Marid	Abbreviated marine reports	marine_ob
Marine logbooks	Directions and speeds of marine currents	mar_current
	Hourly marine weather values	marine_ob
Metform 3208	Daily rainfall (09-09Z)	rain_drnl_ob
	Daily temperature	temp_drnl_ob
	Daily soil temperature at 10, 20, 30, 50 & 100 cm	soil_temp_ob

	Daily run of wind	wind_mean_ob
	Other observations	weather_drnl_ob
		weather_hrly_ob
Metform 3445	Hourly values of sunshine duration	weather_hrly_ob
Metform 6910	Analysis of Anemograms	wind_mean_ob
Metform 7113	Hourly rainfall observations	rain_hrly_ob
Form R35	Daily totals of radiation & sunshine	radt_ob
MODLE	Hourly radiation values	radt_ob
NAVY	Marine obs from RN ships	marine_ob
NCM	Daily soil temperatures at 30 & 100 cm	soil_temp_ob
	Daily temperature extremes	temp_drnl_ob
	Daily weather values	weather_drnl_ob
	Rainfall 09-09	rain_drnl_ob
	Rainfall 09-21 & 21-09	rain_hrly_ob
	State of ground	weather_hrly_ob
OWS	Marine obs from OWS, SDB type 19	marine_ob
PLAT/RIG	Marine reports from platforms and rigs	marine_ob
Pollen observations	Pollen grain counts	pollen_drnl_ob
Rainfall postcards (Metform 7133)	Daily/monthly rainfall	rain_drnl_ob
SREW	Hourly rainfall values	rain_hrly_ob
SSER cachettes	Hourly rainfall amounts calculated from tip times	rain_hrly_ob
SSER cachettes	Tip times and amount of rainfall in tip	rain_subhrly_ob
VOF	Marine reports from VOF	marine_ob
Water Authority diskettes	Daily/monthly rainfall	rain_drnl_ob

N.B. This list is correct at the time of writing, but is subject to change. For the most up to date information, users should refer to the report_table_mapping table in MIDAS.

6.4 Ingestion And Backup Schedule

Some reports are automatically and routinely ingested into the database. Others only become available on an intermittent basis and are processed accordingly.

Met Report	**Ingestion Schedule**
CAWS	Monthly, as received
CLM surface normal	Intermittent, very irregular
CLM upper air normal	Intermittent, very irregular
DALE	Monthly, as received
ESAWS	Daily, in a network submitted automatically at 0100Z (see below)

FM 12-VII SYNOP	Daily, in a network submitted automatically at 0100Z (see below)
FM 13-IX SHIP	Daily, in a network submitted automatically at 0100Z (see below)
FM 15-V METAR	Daily, in a network submitted automatically at 0100Z (see below)
FM 18-X BUOY	Daily, in a network submitted automatically at 0100Z (see below)
FM 18-X Ext DRIFTER	Daily, in a network submitted automatically at 0100Z (see below)
FM 62 TRACKOB	Daily, in a network submitted automatically at 0100Z (see below)
FM 63-IX BATHY	Daily, in a network submitted automatically at 0100Z (see below)
FM 64-IX TESAC	Daily, in a network submitted automatically at 0100Z (see below)
FM 71-VI CLIMAT	Reports no longer received
FM 71-X CLIMAT	Not being stored yet
FM 75-VI CLIMAT TEMP	Not being stored yet
Marine logbooks	Intermittent, substantially in arrears
Metform 3208	Monthly, but day of month varies
Metform 3259	Reports no longer received
Metform 3445	Monthly, as received
Metform 6910	Monthly, as received
Metform 7113	Monthly, as received
Met O 1 Form R35	Monthly, as received
MODLE	Monthly, within first 5 days of each month for previous month
NCM	Daily, in a network submitted automatically at 0100Z (see below)
OWS/NAVY	Intermittent
Pollen observations	Daily between March and September
Rainfall postcards - Metform 7133	Monthly, as received
SREW	Daily, in a network submitted automatically at 0100Z (see below)
SSER cachettes	Monthly, but day of month varies
VOF/PLAT	Daily, in a network submitted automatically at 0100Z (see below)
Water Authority diskettes	Monthly, but day of month varies

Daily routine

A network of inter-dependent jobs is submitted at 01:00Z each day, which handles:

- routine ingestion of a number of reports: marine & land SYNOP, SREW, NCM, HCM and METAR
- submission of other jobs that require the ingested data. One initiates a suite of quality control processes, and another submits various Commercial Suite jobs that require MIDAS data.

Exact timings are influenced by a number of factors and can vary appreciably from day to day, but usually these tasks are complete by 01:30Z. The timings of the later jobs are more likely to be unreliable since they depend on progress of the earlier ones. At 09:50Z another network of jobs is submitted to ingest the latest SYNOP, NCM and SREW data, which is usually complete by 10:00Z. Again a 'trigger' job is released for various Commercial Suite jobs.

Other routine tasks

Daily another suite of jobs extracts and stores a day's worth of data from numerous global SYNOP stations.

7 Database Partitioning

Several MIDAS tables are partitioned; i.e. the data is stored in a number of database files instead of a single file. One or more attributes of the table are defined as the partition key, and the value of the key determines in which file a particular row will be stored. Since the separate files are presented to the user as a single table, users can chose to disregard the partition key. However, the speed of response is directly related to the algorithm that the

database must employ to locate your data, and specifying the partition key will allow the database management system to eliminate those partitions that do not contain data in your chosen range. So, always specify the partition key if you can.

_ob tables
For most '_ob' tables, the partition key is ob_time, so specifying the date and time (or range of dates and times) will also give the partition key.

glbl_wx_ob
For glbl_wx_ob the partition key is wmo_region_code and ob_time. wmo_region_code is a single character '1' - '7' (see below).

'1'	Africa
'2'	Asia
'3'	South America
'4'	North and Central America
'5'	South-west Pacific
'6'	Europe
'7'	Antarctic

The user must give both wmo_region_code and ob_time to get the best performance from the database.

marine_ob
For marine_ob the partition key is longitude_band_code and ob_time. longitude_band_code is a single character 'A' - 'J' that defines a range of longitude values. The longitude band codes are derived by the midas_utility_pkg.lon_ban_fnc packaged function (see Section 11 for details). Users should specify longitude_band_code and ob_time in order to get the best performance.

8 Data Structure Diagrams

This section contains diagrams showing the structure of the MIDAS database.

Logical Data Structure

The first is a **logical** perception, showing the entities and the relationships between them. Basically it is an expansion of the conceptual diagram shown in Section 3 showing all the ancillary entities that we are interested in e.g. met elements, hydrometric areas, etc.

- entities are named using their **logical** (primary) names, as defined during the database design process. They are not necessarily the same as the SQL names, which may be more familiar.
- boxes represent entities (the tables) in the database
- lines show relationships between tables
- a "crow's foot" on a line indicates a one-to-many relationship
- a solid bar is a mandatory relationship e.g. a source remark **must** refer to **one and only one** source
- a circle on the line shows an optional relationship e.g. a source **may** have **one or more** source remarks

Physical Data Structure

The second diagram shows how the logical structure has been implemented. This is the **physical** structure. Annotations on the lines show the minimum and maximum cardinality of the relationships.

- Each hydrometric_area **may** be a rainfall catchement for 0 or more sources (minimum 0, maximum = many).
- Each source **may** be situated in one and only one hydrometric_area (minimum = 0, maximum = 1).
- Each code_detail **must** be a value for one and only one code (minimum = 1, maximum = 1).

See Section 9 for details of the attributes in each table. The foreign keys (the keys that enforce the relationships between entities) can be found from the primary key of the master entity. Again, see Section 9 for individual details.

********** Diagrams Under Development **********

9 Entities

9.1 Introduction

An Oracle database has both **physical** and **logical** structures. The **physical** structure is determined by the host operating system, and consists of data files, log files and control files to provide the actual physical storage for database information. The **logical** structure is determined by its **schema** objects (tables, views, indexes etc.) grouped together in some convenient way.

MIDAS is a schema. It has tables but no views.

MIDASVU is another schema. It contains many views but no tables. Users are provided with read-only access to the tables in MIDAS through views in MIDASVU. This schema also contains more specialised views (judicious combinations of tables).

MIDASUPD is yet another schema with views but no tables, its purpose being to control how the base tables in MIDAS are updated.

There are several other schemas in ORAP and ORAT but these are the ones of relevence to most MIDAS users.

Section 9.2 of this handbook describes the MIDAS tables and MIDASUPD views, because these are co-incident. Attributes in the primary key are marked with an asterisk. Foreign-key attributes can be deduced from the primary key of the master entity, by referring to the Physical Structure Diagram.

Section 9.3 describes the MIDASVU views. Where the MIDASVU view is identical to the corresponding MIDASUPD one, then the reader is referred back to MIDASUPD.

N.B. Some of the MIDASVU views are complex, involving joins on two or more tables (listed below). Users of SQL should **not** join these views with other views. In particular, the ones with names starting src combine attributes from the source, src_capability and 'ob' tables. For best performance, do not use these views if you require only the 'ob' attributes.

capability_at_src
met_domain_element
rain_ob
src_drnl_rain
src_drnl_temp
src_drnl_weather
src_glbl_wx
src_hrly_rain
src_hrly_weather
src_mean_wind
src_radiation
src_sfc_clmo
src_sfc_prcp_clmo
src_soil_temp

9.2 MIDAS Tables and MIDASUPD Views - All (except Marine and Upper Air)

Observation Time Constraint
A time constraint exists on all MIDASUPD views, except those for standing data tables, such that no observation data with a date/time greater than one hour in the future can be stored in MIDAS (e.g. 1000 ob can be stored at

0952). Any rejections are notified. This will prevent instances of data 'labelled' with a date later than 'today' being loaded into MIDAS.

TABLE	acquisitions_log

Description: This entity records recent data acquisitions (ingestion) into the database, except those for marine data (see marine_ob.batch_stamp_time). It contains the key of database meteorological records, the stamp times and the batch number of those records. To assist OPR in identifying recent data which is not yet subject to QC, the batch number is indexed. All users may query this table, to ascertain if MIDAS contains the required data; this should prove more efficient than searching the database met records. This table will contain only RECENT records where QC is not yet complete. Typically this will be data less than a month old. OPR will submit a weekly batch job to clear old records from the log. Where the met record does not include ob_time or ob_hour_count as part of its key, they will be set to NULL in this record.

PK	Attribute	Datatype	Description / Units / Precision
*	ob_end_time	DATE	
*	met_domain_name	VARCHAR2(8)	
*	table_name	VARCHAR2(18)	
*	id_type	VARCHAR2(4)	
*	id	VARCHAR2(8)	
*	ob_hour_count	NUMBER(3)	
*	version_num	NUMBER(1)	
	midas_acq_btch_num	DATE	MIDAS acquisition batch number
	meto_stmp_time	DATE	Met Office receipt stamp time
	midas_stmp_etime	NUMBER(6)	Minutes elapsed time to storage in MIDAS
	src_id	NUMBER(6)	Unique source identifier

TABLE	background_value

Description: midas.background_value contains NWP forecast model hourly background values, for UK land and marine based automatic weather stations. Observations Supply branch uses these values to compile regular reports of AWS quality. midasupd.background_value is the corresponding updatable view. midasvu.land_aws_quality is a read-only view, and joins midas.background_value with midas.weather_hrly_ob. midasvu.marine_aws_quality is a read-only view, and joins midas.background_value with midas.marine_ob and src_capability.

PK	Attribute	Datatype	Description / Units / Precision
*	id	VARCHAR2(8)	
*	id_type	VARCHAR2(4)	Identifier type
*	ob_time	DATE	Date and time of observation
	background_msl_pressure	NUMBER(5,1)	
	background_air_temperature	NUMBER(3,1)	
	background_relative_humidity	NUMBER(3)	
	background_wind_direction	NUMBER(3)	

	background_wind_speed	NUMBER(3)	
	clbw_id	VARCHAR2(8)	The pseudo WMO number for CDL data. Values are > '90000'.
	clbw_id_type	VARCHAR2(4)	Always set to 'WMO'. It allows data extraction processes to be consistent within the application.

TABLE	british_summer_time

Description: The table records the offset of local time (e.g. British Summer Time) for each year since 1916.

PK	Attribute	Datatype	Description / Units / Precision
*	begin_time	DATE	
	end_time	DATE	
	utc_offset	NUMBER(1)	

VIEW	clm_src_capability

Description: midasupd.clm_src_capability is a simple updateable view and is created in the MIDASUPD schema only. It provides one-for-one projection of columns from the base table, but is restricted to rows where met_domain_name is like 'CLM%'. This view allows specified users to update the CLIMAT src_capability records only.

The structure of this view is identical to the src_capability table.

TABLE	clm_ua_norm

Description: Long period surface averages for a CLIMAT upper air station for the specified month. Attributes norm_first_year and norm_last_year specify the period over which the normals are calculated. CLIMAT normals are not subject to quality control, so there are no QC or version number attributes.

PK	Attribute	Datatype	Description / Units / Precision
*	id_type	VARCHAR2(4)	Identifier type
*	id	VARCHAR2(8)	
*	norm_first_year	NUMBER(4)	First year of normal
*	norm_last_year	NUMBER(4)	Last year of normal
*	norm_month	NUMBER(2)	Month of normal
*	met_domain_name	VARCHAR2(8)	
	src_id	NUMBER(6)	Unique source identifier
	rec_st_ind	NUMBER(4)	State indicator for the record
	ua_norm_ob_id	NUMBER(15)	Observation identifier
	mean_stn_lvl_pres	NUMBER(5,1)	Normal - monthly mean pressure 0.1 hpa

	sd_mean_stn_lvl_pres	NUMBER(5,1)	Std dev - monthly mean pressure 0.1 hpa
	mean_sfc_air_temp	NUMBER(3,1)	Normal - monthly mean air temp 0.1 deg C
	sd_mean_sfc_air_temp	NUMBER(2,1)	Std dev - monthly mean air temp 0.1 deg C
	mean_surface_dewpoint	NUMBER(2)	Normal - mean monthly dew point 0.1 deg C
	sd_mean_sfc_dwpt	NUMBER(2,1)	Std dev - mean monthly dew point 0.1 deg C

TABLE	clm_ua_norm_lvl

Description: CLIMAT upper air standard isobaric surface check values for a month and station at a specified upper air pressure level. CLIMAT normals are not subject to quality control, so there are no QC or version number attributes.

PK	Attribute	Datatype	Description / Units / Precision
*	ua_norm_ob_id	NUMBER(15)	Observation identifier
*	pres_coord	NUMBER(5,1)	Air pressure
	mo_mn_geop_ht	NUMBER(6)	Normal - monthly av geopot ht decametres
	sd_ua_mo_mn_geop_ht	NUMBER(5,1)	Std dev - monthly average height decametres
	ua_mo_mn_temp	NUMBER(3,1)	Normal - monthly mean air temp 0.1 deg C
	sd_ua_mo_mn_temp	NUMBER(4,2)	Std dev - monthly mean air temp 0.1 deg C
	ua_mo_mn_dwpt	NUMBER(2)	Normal - mean monthly dew point 0.1 deg C
	sd_ua_mo_mn_dwpt	NUMBER(4,2)	Std dev - mean monthly dew point 0.1 deg C
	ua_mo_mn_ucmp_wind	NUMBER(2)	Mean u-component of vector wind 0.1 knot
	sd_ua_mo_mn_ucmp_wind	NUMBER(3)	Std dev - monthly mean u-component 0.1 knot
	ua_mo_mn_vcmp_wind	NUMBER(2)	Mean v-component of vector wind 0.1 knot
	sd_ua_mo_mn_vcmp_wind	NUMBER(3)	Std dev - monthly mean v-component 0.1 knot

---

TABLE	clm_ua_rec

Description: This entity contains monthly values at the surface received in WMO code FM-75-CLIMAT TEMP which is a monthly report of upper air CLIMAT values. The record is for the specified year, month and station.

PK	Attribute	Datatype	Description / Units / Precision
*	src_id	NUMBER(6)	Unique source identifier
*	ob_date	DATE	Last day of month of observation
*	version_flag	CHAR(1)	Observation version number
	id_type	VARCHAR2(4)	Identifier type
	id	VARCHAR2(8)	
	met_domain_name	VARCHAR2(8)	
	rec_st_ind	NUMBER(4)	State indicator for the record
	mean_stn_lvl_pres	NUMBER(5,1)	Monthly mean surface pressure 0.1 hpa
	mean_sfc_air_temp	NUMBER(3,1)	Mean monthly air temperature 0.1 deg C
	mean_surface_dewpoint	NUMBER(4,1)	Mean monthly dew point 0.1 deg C
	nom_rsnd_hr_id	CHAR(1)	Nominal hour of radio-sonde ascent / code
	mean_stn_lvl_pres_qc_code	CHAR(1)	Quality of mean station-level air pressure / code
	mean_sfc_air_temp_qc_code	CHAR(1)	Quality of mean surface air temperature / code
	mean_surface_dewpoint_qc_code	CHAR(1)	Quality of mean surface dewpoint / code
	meto_stmp_time	DATE	Met Office receipt stamp time
	qual_analyst_id	VARCHAR2(3)	Identifier of QC staff member

TABLE	clm_ua_rec_lvl

Description: This entity contains monthly values at specified pressure levels from CLIMAT upper air reports.

PK	Attribute	Datatype	Description / Units / Precision
*	src_id	NUMBER(6)	Unique source identifier
*	ob_date	DATE	Last day of month of observation
*	version_flag	CHAR(1)	Observation version number
*	pres_coord	NUMBER(5,1)	Upper air atmospheric pressure 0.1 hpa
	ua_month_mean_height	NUMBER(6)	Monthly mean geopotential height metres
	ua_month_mean_air_temp	NUMBER(3,1)	Mean monthly air temperature 0.1 deg C
	ua_month_mean_dewpoint	NUMBER(4,1)	Mean monthly dewpoint 0.1 deg C
	ua_month_steadiness_wind	NUMBER(3)	Monthly wind steadiness factor %

	ua_month_mean_wind_dir	NUMBER(3)	Monthly mean vector wind dir'n deg True
	ua_month_mean_wind_speed	NUMBER(3)	Monthly mean vector wind speed knots
	temp_miss_mo_day_cnt	CHAR(1)	Count of days when air temperature not reported at this pressure level / code
	ua_wind_miss_mo_day_cnt_code	CHAR(1)	Count of days when wind not reported at this pressure level / code
	ua_month_mean_height_qc_code	CHAR(1)	Quality of monthly mean height at this pressure level / code
	ua_month_mean_air_temp_qc_code	CHAR(1)	Quality of monthly mean air temperature at this pressure level / code
	ua_month_mean_dewpoint_qc_code	CHAR(1)	Quality of monthly mean dewpoint at this pressure level / code
	ua_month_mean_wind_qc_code	CHAR(1)	Quality of monthly mean wind at this pressure level / code

---

TABLE	code

Description: The codes described by this entity may be one of the following types: WMO, BUFR, NCM tables, local MIDAS codes. These codes are used in either met reports (i.e. input domains), or to describe the values in the columns of MIDAS observation tables. To facilitate substring searches, each description will start with the code type and number. Each domain element may be explained by one and only one code. Each code must be described by one or more code details. A met element (cf domain element) may be reported or stored using one or more codes.

PK	Attribute	Datatype	Description / Units / Precision
*	code_id	VARCHAR2(8)	Code identifier
	code_name	VARCHAR2(15)	Unique code name
	code_description	VARCHAR2(72)	
	code_bgn_date	DATE	
	code_end_date	DATE	
	rec_st_ind	NUMBER(4)	State indicator

TABLE	code_detail

Description: This entity describes the coded and decoded values for each entry in a code table (see Section 14.5.3).

PK	Attribute	Datatype	Description / Units / Precision
*	code_id	VARCHAR2(8)	Code identifier
*	code_value	VARCHAR2(4)	
*	desc_line_number	NUMBER(4)	Description line number

	rec_st_ind	NUMBER(4)	
	code_dtl_bgn_date	DATE	Code value begin date
	code_dtl_end_date	DATE	
	code_val_dsc	VARCHAR2(340)	Code value description

TABLE	cross_reference

Description: This entity defines a cross reference relationship between one source and another. It could be one of a set of cross references of a specified type (e.g. a set of *n* daily rainfall buddy stations for a specified station). Cross references may be needed for the following reasons: QC purposes, administering, calibration or collecting stations, previous stations for climatological sequences, etc. A met_domain_name may be linked for indexing purposes but its use should also be indicated within the name of the association_type as it is not a primary key field.

PK	Attribute	Datatype	Description / Units / Precision
*	ref_from_src_id	NUMBER(6)	from src_id
*	ref_to_src_id	NUMBER(6)	to src_id
*	association_type	VARCHAR2(28)	
*	cross_ref_bgn_date	DATE	
	cross_ref_end_date	DATE	
	met_domain_name	VARCHAR2(8)	
	rec_st_ind	NUMBER(4)	State indicator

TABLE	domain_element

Description: This entity defines a met_element within a met_domain. The domain in which the element belongs may be:

- an input grouping (i.e. a meteorological report such as a SYNOP or METAR)
- a storage grouping (i.e. the name of a database table such as weather_hrly_ob
- an output grouping (i.e. a desired view of the data such as DMWR or DSS report)

Each domain element has the following attributes:

- the name of the domain of which it is part
- the id of the met element
- the position of the element within the domain
- the id of the code in which the element is reported or stored
- the current state of the domain element

Optionally, when the domain is an input domain (i.e. met report), the name of the domain (i.e. table) in which the element will be stored is also available. When the domain is a storage domain (i.e. a database table), attribute str_met_dom_name will be null (if it was set it would merely point back to itself). If a change to a domain_element is required, then a new met_domain must be created and the old one (and its elements) closed.

ob_table_name and ob_column_name are used to record where the met_element is stored in MIDAS. This information can also be deduced from report_table_mapping and all_tab_columns. This duplication will be resolved at the next opportunity.

PK	Attribute	Datatype	Description / Units / Precision
*	met_domain_name	VARCHAR2(8)	
*	met_element_id	NUMBER(5)	Met element identifier
	dom_elem_pos_num	NUMBER(4)	
	str_met_dom_name	VARCHAR2(8)	
	code_id	VARCHAR2(8)	Code identifier
	rec_st_ind	NUMBER(4)	State indicator
	ob_table_name	VARCHAR2(30)	
	ob_column_name	VARCHAR2(30)	

TABLE	domain_element_type

Description: This entity resolves the many to many relationship between the midas.domain_element table and the midas.identifier_type table.

PK	Attribute	Datatype	Description / Units / Precision
*	met_domain_name	VARCHAR2(8)	Met domain name
*	met_element_id	NUMBER(5)	Met element identifier
*	id_type	VARCHAR2(4)	Identifier type

TABLE	geographic_area

Description: These records define the geographic areas in which sources are located or who they are administered by, such as England, Great Britain, Europe, Scotland, Wales, Canada, etc. MIDAS will not store details of geographic areas for which there are no observations.

PK	Attribute	Datatype	Description / Units / Precision
*	geog_area_id	VARCHAR2(4)	Geographic area identifier
	geog_area_type	VARCHAR2(16)	Geographic area type
	geog_area_name	VARCHAR2(72)	Geographic area name
	rec_st_ind	NUMBER(4)	State indicator
	geog_area_dsc	VARCHAR2(24)	
	geog_area_upd_date	DATE	Record updated date

TABLE	geog_area_hier

Description: Each record defines a relationship between two geographic areas e.g. Berkshire within England, England, within Great Britain, England within Europe. This table will be used to implement a strictly hierarchical

structure between geographic areas, until such time as there is a proven need for the more elaborate network relationships between them.
e.g. Northern Ireland is part of UK only; counties of Southern Ireland are part of Eire only; there is NO geographic area "Island of all Ireland" containing both N. Ireland and Eire.

PK	Attribute	Datatype	Description / Units / Precision
*	wthn_geog_area_id	VARCHAR2(4)	Member geographic area
*	cntn_geog_area_id	VARCHAR2(4)	Owner geographic area
	geog_area_hier_upd_date	DATE	Record updated date
	rec_st_ind	NUMBER(4)	State indicator

VIEW	glbl_wx_ob

Description: This entity contains meteorological values observed at 3-hrly intervals by non-UK stations, as reported in SYNOP and METAR codes. The primary use of the entity requires the data in climatological sequence, i.e. all times for a station. New observations are added daily.

The attributes of this entity are similar to those of the weather_hrly_ob table, but with significant differences. Some of the attributes refer to 24hr values, but the primary user has requested that the data are not normalised. The period of maximum and minimum temperatures are defined by WMO Regional Reporting Practices. Where there is a 'WMO commitment', data will be retained permanently, but oldest data for other stations will be deleted annually. No QC is performed on the attribute values listed below, but QC flags may have been set. The flags themselves are held in attributes qc_flag_list_1 and qc_flag_list_2, each 1-bit flag in it corresponding to an attribute with a QC-flag. For software to interpret the flag lists please contact the MIDAS Team.

The table has two-dimensional partitioning, using wmo_region_code and ob_time.

PK	Attribute	Datatype	Description / Units / Precision
*	id	VARCHAR2(8)	
*	id_type	VARCHAR2(4)	Identifier type
*	ob_time	DATE	Date and time of observation
*	met_domain_name	VARCHAR2(8)	
	src_id	NUMBER(6)	Unique source identifier
	wmo_region_code	CHAR(1)	'1' - '7'
	rec_st_ind	NUMBER(4)	
	wind_direction	NUMBER(3)	Degs true
	wind_speed	NUMBER(3)	Knots
	prst_wx_id	CHAR(2)	Present weather code
	past_wx_id_1	CHAR(1)	Past weather code #1
	past_wx_id_2	CHAR(1)	Past weather code #2
	cld_ttl_amt_id	CHAR(1)	Total cloud amount code oktas
	cld_base_amt_id	CHAR(1)	Cloud base amount code
	cld_base_ht	NUMBER(4)	Cloud base height code decametres
	low_cld_type_id	CHAR(1)	Low cloud type code
	med_cld_type_id	CHAR(1)	Medium cloud type code
	hi_cld_type_id	CHAR(1)	High cloud type code

	wind_speed_unit_id	CHAR(1)	Wind speed unit code
	visibility	NUMBER(4)	Decametres
	air_temperature	NUMBER(3,1)	Surface air temperature 0.1 deg C
	wetb_temp	NUMBER(3,1)	Wet bulb temperature 0.1 deg C
	dewpoint	NUMBER(3,1)	Dewpoint temperature 0.1 deg C
	stn_pres	NUMBER(5,1)	Station air pressure 0.1 hpa
	pres_sfc	NUMBER(5,1)	Pressure surface
	pres_sfc_ht	NUMBER(5)	Height of pressure surface
	msl_pressure	NUMBER(5,1)	Mean sea level air pressure 0.1 hpa
	pres_tdcy_amt	NUMBER(4,1)	Air pressure tendency
	prcp_ob_hr_cnt	NUMBER(3)	Precipitation hour count
	prcp_amt	NUMBER(5,1)	Precipitation amount
	cld_amt_id_1	CHAR(1)	Layer cloud amount code #1
	cloud_type_id_1	CHAR(1)	Cloud type code #1
	cld_base_ht_id_1	NUMBER(4)	Cloud base height code #1 decametres
	cld_amt_id_2	CHAR(1)	Layer cloud amount code #2
	cloud_type_id_2	CHAR(1)	Cloud type code #2
	cld_base_ht_it_2	NUMBER(4)	Cloud base height code #2 decametres
	cld_amt_id_3	CHAR(1)	Layer cloud amount code #3
	cloud_type_id_3	CHAR(1)	Cloud type code #3
	cld_base_ht_id_3	NUMBER(4)	Cloud base height code #3 decametres
	max_air_temp	NUMBER(3,1)	Maximum air temperature
	min_air_temp	NUMBER(3,1)	Minimum air temperature
	ground_state_id	CHAR(2)	Ground state id
	min_grss_temp	NUMBER(3,1)	Grass temperature
	snow_depth	NUMBER(4)	Snow depth
	sun_ob_hr_cnt	NUMBER(3)	Sunshine observation hour count
	sun_dur	NUMBER(3,1)	Sunshine duration
	q24hr_prcp_amt	NUMBER(5,1)	24hr precipitation amount
	q24hr_pres_tdcy_amt	NUMBER(4,1)	24hr air pressure tendency
	vert_vsby	NUMBER(3)	Vertical visibility decametres
	src_opr_type	CHAR(1)	Source operation type code
	gust_spd_type_code	CHAR(1)	Gust speed type code 0 = 10 minute gust 1 = As past weather, etc. See 910 - 914 of code 3778
	max_gust_speed	NUMBER(3)	Maximum gust speed
	runway_name	VARCHAR2(4)	Runway name
	rnwy_vis_rnge	NUMBER(4)	Runway visual range

	alt_pres	NUMBER(4)	Altimeter pressure 0.1 hpa
	qc_flag_list_1	NUMBER(9)	QC bit flags for 9 specified items
	qc_flag_list_2	NUMBER(9)	

TABLE	hydrometric_area

Description: This entity contains details for a UK Met Office hydrometric area. Each area contains a number of rainfall stations. For each area, the first and last possible rainfall station identifiers are defined. Each area is wholly contained within one Water Authority area.

PK	Attribute	Datatype	Description / Units / Precision
*	hydr_area_id	NUMBER(4)	Hydrometric area identifier
	water_auth_name	VARCHAR2(22)	Water authority name
	frst_rfall_stn_num	CHAR(6)	First rainfall station number
	last_rfall_stn_num	CHAR(6)	Last rainfall station number
	hydr_area_upd_date	DATE	Record updated date
	rec_st_ind	NUMBER(4)	State indicator

TABLE	identifier_type

Description: This table defines the various types of source identifiers that may be used (e.g. ICAO, WMO, DCNN)

PK	Attribute	Datatype	Description / Units / Precision
*	id_type	VARCHAR2(4)	Identifier type
	id_type_dsc	VARCHAR2(240)	Identifier type description
	id_type_upd_date	DATE	Record updated date
	rec_st_ind	NUMBER(4)	State indicator

TABLE	met_domain

Description: A met domain is a group of meteorological elements that are of interest to MIDAS users, uniquely identified by its name. Each met domain is assigned to a class. It may be an input grouping, or a storage group, or an output grouping. An input grouping is a meteorological report, e.g. synop. A storage group is the name of the MIDAS table in which the meteorological elements are stored. An output grouping corresponds to some desired view of the data, e.g. DMWR, or DSS report.

Every change to the elements used within a met domain will require a new met domain.

PK	Attribute	Datatype	Description / Units / Precision
*	met_domain_name	VARCHAR2(8)	

	met_dom_class	VARCHAR2(28)	
	met_dom_bgn_date	DATE	Effective from date
	met_dom_end_date	DATE	Effective to date
	rec_st_ind	NUMBER(4)	State indicator
	dom_usg_id	CHAR(1)	Met domain usage identifier
	met_domain_dsc	VARCHAR2(80)	Met domain description

TABLE	met_domain_class

Description: This entity contains details about the types of met domain used to report and store meteorological data.

PK	Attribute	Datatype	Description / Units / Precision
*	met_dom_class	VARCHAR2(28)	
	dom_clas_dsc	VARCHAR2(72)	Met domain class description
	met_dom_clas_upd_date	DATE	Record updated date
	rec_st_ind	NUMBER(4)	State indicator

TABLE	met_domain_rank

Description: This entity contains details about the ranking of report types for each met domain.

PK	Attribute	Datatype	Description / Units / Precision
*	table_type	VARCHAR2(25)	The type of observational data required e.g. 'RADIATION', 'HRLY_RAIN'
	met_domain	VARCHAR2(8)	The met domain name. Foreign key to met_domain.met_domain_name
	rank	NUMBER(2,0)	The domain rank for the observation

TABLE	met_element

Description: This entity defines individual meteorological items (elements) independently of their organisation into meteorological observations or output views (met domains). It does not contain the values for the meteorological item, merely the definition of it. BUFR specifications are used in the definition of the appropriate attributes. The definition of each met element implies a Datatype and format for the attribute, e.g. temperatures are stored as integers, to a precision of 0.1 deg. C., and are defined as such in the data dictionary. MIDAS stores data in consistent units, but these may not be the units in which the element was reported. The met_element table contains minimum and maximum values of "open ended" elements (e.g. air temperature). This allows the storage process to validate data upon input. The current maximum and minimum values can be inspected by users.

PK	Attribute	Datatype	Description / Units / Precision

*	met_element_id	NUMBER(5)	Met element identifier
	met_element_name	VARCHAR2(32)	
	met_element_class	VARCHAR2(28)	
	met_elem_bgn_date	DATE	Effective from date
	met_elem_end_date	DATE	Effective to date
	rec_st_ind	NUMBER(4)	State indicator
	met_element_description	VARCHAR2(200)	Definitive description of met element
	minimum_value	NUMBER(5)	
	maximum_value	NUMBER(5)	
	scale_factor	NUMBER(2)	

TABLE	met_elem_class

Description: This entity classifies met_element records, to make access to their details easier. The classification system is based on BUFR classes (XX), but this is subject to further refinement.

PK	Attribute	Datatype	Description / Units / Precision
*	met_element_class	VARCHAR2(28)	Met element class description
	elem_clas_dsc	VARCHAR2(72)	
	met_elem_clas_upd_date	DATE	Record updated date
	rec_st_ind	NUMBER(4)	State indicator

VIEW	mo_sfc_elem

Description: Contains monthly surface values of specified met elements. The met_element_id column defines and constraints the met element, via a foreign-key relationship. The parent mo_sfc_rec constrains ob_date and met_domain_name, and implicitly constrains id_type and id. The midasupd.mo_sfc_elem view restricts rows to those with the appropriate storage met domain name, i.e. 'MOSFCELM'.

PK	Attribute	Datatype	Description / Units / Precision
*	mo_sfc_elem_prtn_id	CHAR(2)	Partition key (found via domain_element and met_domain_element)
*	met_element_id	NUMBER(5)	
*	ob_date	DATE	
*	src_id	NUMBER(6)	
*	met_domain_name	VARCHAR2(8)	e.g. 'CLM71-11', 'CARLOS' etc
*	version_num	NUMBER(1)	
	rec_st_ind	NUMBER(4)	
	met_elem_day_cnt	NUMBER(2)	Count of days when the met element occurred
	met_elem_hour_cnt	NUMBER(3)	Count of hours when the met element occurred
	met_elem_min_val	NUMBER(6,2)	Minimum value of the met element

	met_elem_min_val_day_num	NUMBER(2)	Day on which the min value occurred
	met_elem_mean_val	NUMBER(6,2)	Mean value of the met element
	met_elem_sd_val	NUMBER(6,2)	Standard deviation
	met_elem_max_val	NUMBER(6,2)	Maximum value of the met element
	met_elem_max_val_day_num	NUMBER(2)	Day on which the max value occurred
	met_elem_miss_hour_cnt	NUMBER(3)	Count of hours when the element was missing
	met_elem_amt	NUMBER(6,2)	e.g. rainfall amount
	met_elem_occr_prc	NUMBER(3)	Percent of occurrences, e.g. NE wind
	met_elem_qual_id	CHAR(1)	Quality of the met element
	met_elem_sbst_hour_cnt	NUMBER(3)	Count of hours when substitute values used.
	met_elem_sbst_code	CHAR(2)	Defines substitute values
	last_update_time	DATE	Stores date/time when table was last updated using a trigger.

VIEW	mo_sfc_rec

Description: This table stores parent values for one or more mo_sfc_elem entries, for the given source, month and met domain name. It defines the id_type and id used by the source, and constrains ob_date to the last day of the month. The mo_sfc_elem entries are calculated monthly values from globally located sources. These values are received in WMO codes FM-71-VI-CLIMAT and FM-71-X1-CLIMAT, or are 'CARLOS' values derived from Midas data.

PK	Attribute	Datatype	Description / Units / Precision
*	src_id	NUMBER(6)	
*	ob_date	DATE	
*	met_domain_name	VARCHAR2(8)	e.g. 'CLM71-11', 'CARLOS' etc.
	rec_st_ind	NUMBER(4)	
	id_type	VARCHAR2(4)	
	id	VARCHAR2(8)	
	wind_speed_unit_id	CHAR(1)	
	temp_read_type_id	CHAR(1)	
	max_temp_read_hour_num	NUMBER(2)	Time (Hour) when maximum air temperature is read.
	min_temp_read_hour_num	NUMBER(2)	
	qual_analyst_id	VARCHAR2(3)	Identifier of QC staff member

TABLE	pollen_drnl_ob

Description: This table stores pollen observations from a variety of tree and plant speicies from observing sites

in the UK. The observations are normally received by e-mail on a daily basis between March and September and then loaded to this table.

PK	Attribute	Datatype	Description / Units / Precision
*	ob_date	DATE	
*	version_num	NUMBER(1)	
*	met_domain_name	VARCHAR2(6)	Always POLLEN
*	id_type	VARCHAR2(4)	Always POLL
*	id	VARCHAR2(6)	
	src_id	NUMBER(6)	
	rec_st_ind	NUMBER(4)	
	corylus	NUMBER(4)	Precipitation duration (<24 hr) minutes
	alnus	NUMBER(4)	Number of alnus pollen grains per metre cubed over a 24 hour period
	salix	NUMBER(4)	Number of salix pollen grains per metre cubed over a 24 hour period
	betula	NUMBER(4)	Number of betula pollen grains per metre cubed over a 24 hour period
	fraxinus	NUMBER(4)	Number of fraxinus pollen grains per metre cubed over a 24 hour period
	ulmus	NUMBER(4)	Number of ulmus pollen grains per metre cubed over a 24 hour period
	quercus	NUMBER(4)	Number of quercus pollen grains per metre cubed over a 24 hour period
	platanus	NUMBER(4)	Number of platanus pollen grains per metre cubed over a 24 hour period
	poaceae	NUMBER(4)	Number of poaceae pollen grains per metre cubed over a 24 hour period
	urtica	NUMBER(4)	Number of urtica pollen grains per metre cubed over a 24 hour period
	artemisia	NUMBER(4)	Number of artemisia pollen grains per metre cubed over a 24 hour period
	ambrosia	NUMBER(4)	Number of ambrosia pollen grains per metre cubed over a 24 hour period
	meto_stmp_time	DATE	Met Office receipt stamp time

TABLE	prime_drnl_rain_ob

Description: This entity contains daily, best estimate rainfall data by src_id. This table holds daily rainfall data back to 1958. New data is generally loaded monthly following the completion of quality control; therefore data should be available for the period up to approximately six months ago. Ad hoc updates may occur outside this cycle. This table also serves to simplify rainfall data query and extraction.

PK	Attribute	Datatype	Description / Units / Precision
*	src_id	NUMBER(6)	Source identifier

*	ob_end_time	DATE	Date and time at end of observation
	prcp_amt	NUMBER(5,1)	Precipitation amount
	last_update_time	DATE	Time of last update (maintained by trigger)

VIEW	radt_ob

Description: This entity contains hourly and daily radiation amounts, including those no longer being reported. In all cases, ob_end_time and ob_hour_count define the observation period. Values are either for an hour or for 24 hours.

- Hourly - approximately 50 UK stations and 100 overseas stations report hourly radiation, but only 3 have reported direct irradiation.
- Daily - 24 hour values are usually for 0000-2359Z on the stated day. Occasionally the period may be 09-09Z, ending at 0900Z on the stated day. There should only be one 24 hour report per station per day. MIDAS will not store daily values derived from the sum of hourly values.

Non-key attributes are from mp.dradiatn, mp.hradiatn.ylast2yr and mc.hradname.

PK	Attribute	Datatype	Description / Units / Precision
*	ob_end_time	DATE	Date and time at end of observation
*	id_type	VARCHAR2(4)	Identifier type
*	id	VARCHAR2(8)	
*	ob_hour_count	NUMBER(3)	Observation hour count
*	version_num	NUMBER(1)	Observation version number
*	met_domain_name	VARCHAR2(8)	
	src_id	NUMBER(6)	Unique source identifier
	rec_st_ind	NUMBER(4)	State indicator for the record
	glbl_irad_amt	NUMBER(4)	Global solar irradiation amount Kjoules/ sq metre
	glbl_irad_amt_q	NUMBER(5)	QC code - global irradiation amt
	difu_irad_amt	NUMBER(4)	Diffuse solar irradiation amount Kjoules/ sq metre
	difu_irad_amt_q	NUMBER(5)	QC code - diffuse irradiation amt
	direct_irad	NUMBER(4)	Direct irradiation amount Kjoules/ sq metre
	direct_irad_q	NUMBER(5)	QC code - direct irradiation
	irad_bal_amt	NUMBER(4)	Irradiation balance amount Kjoules/ sq metre
	irad_bal_amt_q	NUMBER(5)	QC code - irradiation balance amt
	glbl_s_lat_irad_amt	NUMBER(4)	Mean global S latitude radiation Kjoules/ sq metre
	glbl_s_lat_irad_amt_q	NUMBER(5)	QC code - global S lat irad amt
	glbl_horz_ilmn	NUMBER(4)	Global horizontal illumination Kjoules/ sq metre
	glbl_horz_ilmn_q	NUMBER(5)	QC code - global horizontal illumination

	meto_stmp_time	DATE	Met Office receipt stamp time
	midas_stmp_etime	NUMBER(6)	Elapsed time to storage in MIDAS minutes

---

TABLE	radt_ob_v2

Description: This table is the underlying entity for the midasupd.radt_ob view and has the identical structure.

VIEW	rain_drnl_ob

Description: This entity contains rainfall amounts measured over one or more 24 hour periods, ending on the stated day. Attributes ob_end_day and ob_day_count define the observation period. It does NOT contain 09-21Z and 21-09Z rainfall; they are attributes of the rain_hrly_ob entity. The entity can contain rainfall of two types:

- Daily rainfall over a 24 hour period - usually 09-09Z.
  In this case ob_day_count = 1, the default value
- Daily rainfall from irregular sources - those which report monthly or weekly.
  In this case ob_day_count = the number of days over which the rainfall is measured.

Some stations report rainfall at 10Z, so time is included as an attribute, although there is no time field on the rainfall postcard. To prevent multiple observations for a station for a day, attribute ob_day_count is not in the primary key. The identifier is a rainfall station number. Some stations report daily rainfall via F3208 using their DCNN. MIDAS will store them as rows in this table using their rainfall number, i.e. id_type = 'rain' and id = rainfall_number.

Non-key attributes are copied from mp.drain and mc.drain.

PK	Attribute	Datatype	Description / Units / Precision
*	ob_date	DATE	Date of observation
*	id	VARCHAR2(8)	
*	id_type	VARCHAR2(4)	Identifier type
*	version_num	NUMBER(1)	
*	met_domain_name	VARCHAR2(8)	
	ob_end_ctime	NUMBER(4)	Clock-time at end of observation
	ob_day_cnt	NUMBER(3)	Observation day count
	src_id	NUMBER(6)	Unique source identifier
	rec_st_ind	NUMBER(4)	State indicator for the record
	prcp_amt	NUMBER(5,1)	Precipitation amount 0.1 mm
	ob_day_cnt_q	NUMBER(5)	QC code - day count
	prcp_amt_q	NUMBER(5)	QC code - precipitation amount
	prcp_amt_j	CHAR(1)	Descriptor - precipitation amount
	meto_stmp_time	DATE	Met Office receipt stamp time
	midas_stmp_etime	NUMBER(6)	Elapsed time to storage in MIDAS minutes

VIEW	rain_hrly_ob

Description: This entity contains rainfall amount (and duration from tilting syphon gauges) during the hour (or hours) ending at the specified time. Normally it will NOT be used to store precipitation measured over 24 hours or more - that will use the rain_drnl_ob table. However, where a station normally reports 09-21Z and 21-09Z values, an original value which is deemed to be erroneous can be replaced by a 24-hour value. ob_hour_count will be set to the number of hours (always less than 24). The data in this table are of the following types:

- Hourly rainfall (ob_hour_count = 1)
- 09-21Z or 21-09Z rainfall from NCM and F7113 (ob_hour_count = 12)
- Rainfall amounts measured over a number of hours (e.g. 06-18Z from SYNOPS - not stored at present)

The identifier is a rainfall station number. Non-key attributes are taken from mp.hrain and mp.dmwr.

PK	Attribute	Datatype	Description / Units / Precision
*	ob_end_time	DATE	Date and time at end of observation
*	id_type	VARCHAR2(4)	Identifier type
*	id	VARCHAR2(8)	
*	ob_hour_count	NUMBER(3)	Observation hour count
*	version_num	NUMBER(1)	Observation version number
*	met_domain_name	VARCHAR2(8)	
	src_id	NUMBER(6)	Unique source identifier
	rec_st_ind	NUMBER(4)	State indicator for the record
	prcp_amt	NUMBER(5,1)	Precipitation amount 0.1 mm
	prcp_dur	NUMBER(4)	Precipitation duration (<24 hr) minutes
	prcp_amt_q	NUMBER(5)	QC code - precipitation amount
	prcp_dur_q	NUMBER(5)	QC code - precipitation duration
	prcp_amt_j	CHAR(1)	Descriptor - precipitation amount
	meto_stmp_time	DATE	Met Office receipt stamp time
	midas_stmp_etime	NUMBER(6)	Elapsed time to storage in MIDAS minutes

VIEW	rain_subhrly_ob

Description: This entity contains rainfall tip amounts with a time resolution of one minute, from SSER. The identifier is DCNN or rainfall station number. Non-key attributes are taken from mp.srain. There are no QC attributes because, in May 1995, OPR confirmed that there would not be any quality control performed on sub-hourly rainfall.

PK	Attribute	Datatype	Description / Units / Precision
*	ob_time	DATE	Date and time of observation
*	id_type	VARCHAR2(4)	Identifier type

*	id	VARCHAR2(8)	
*	met_domain_name	VARCHAR2(8)	
	src_id	NUMBER(6)	Unique source identifier
	prcp_tip_amt	NUMBER(4,3)	Precipitation tip amount 0.001 mm
	rec_st_ind	NUMBER(4)	State indicator for the record
	meto_stmp_time	DATE	Met Office receipt stamp time
	midas_stmp_etime	NUMBER(6)	Elapsed time to storage in MIDAS minutes

---

VIEW	report_table_mapping

Description: This entity records the valid combinations of id_type and met_domain for each ob table.

PK	Attribute	Datatype	Description / Units / Precision
*	report_met_domain_name	VARCHAR2(8)	
*	report_id_type	VARCHAR2(4)	
*	table_name	VARCHAR2(18)	
	stored_id_type	VARCHAR2(4)	
	domain_table_begin_date	DATE	
	domain_table_end_date	DATE	
	archive_area_code	VARCHAR2(4)	Met domain rank. Precedence of the met domain within this table.

VIEW	runway_ob

Description: This entity contains runway visual ranges measured during the hour ending at the stated date and time. Identifier is ICAO-id or WMO number. Non-key attributes are taken from mop.hdata.egxx. At present only runway visual ranges are stored, but state of runway can be accommodated here if needed.

PK	Attribute	Datatype	Description / Units / Precision
*	ob_time	DATE	Date and time of observation
*	id	VARCHAR2(8)	
*	id_type	VARCHAR2(4)	Identifier type
*	runway_name	VARCHAR2(4)	
	met_domain_name	VARCHAR2(8)	
	src_id	NUMBER(6)	Unique source identifier
	rec_st_ind	NUMBER(4)	State indicator for the record
	rnwy_vis_rnge	NUMBER(4)	Runway visual range metres
	rnwy_vis_rnge_q	NUMBER(5)	Quality code - runway visual range Only present on base table
	rnwy_vis_rnge_j	CHAR(1)	Descriptor - runway visual range Only present on base table

TABLE	sfc_clmo_elem

Description: The midas.sfc_clmo_elem table contains monthly and annual surface climatological averages and met element check values for surface CLIMAT reports. These values are calculated by the CARLOS and CLIMAT teams. The owner midas.sfc_clmo_rec constrains clmo_end_date and met_domain_name, and implicitly constrains id_type and id. A sfc_clmo_elem row is for the specified src_id, clmo_end_date, year_count, met_domain_name and met_element_id. midasupd.sfc_clmo_elem is a view of midas.sfc_clmo_elem, with one-for-one projection of the columns. midasvu.sfc_clmo_elem is a read-only synonym of midasupd.sfc_clmo_elem.

PK	Attribute	Datatype	Description / Units / Precision
*	src_id	NUMBER(6)	Unique source identifier
*	clmo_end_date	DATE	
*	clmo_year_cnt	NUMBER(2)	
*	met_domain_name	VARCHAR2(8)	
	month_number	NUMBER(2)	
	met_element_id	NUMBER(5)	
	rec_st_ind	NUMBER(4)	State indicator for the record
	met_elem_ob_per	NUMBER(8)	
	met_elem_ob_mo_cnt	NUMBER(3)	
	met_elem_mean_val	NUMBER(6,2)	
	met_elem_sd_val	NUMBER(6,2)	
	met_elem_medn_val	NUMBER(6,2)	
	met_elem_75prc_val	NUMBER(6,2)	
	met_elem_25prc_val	NUMBER(6,2)	
	met_elem_high_cnf_val	NUMBER(6,2)	
	met_elem_low_cnf_val	NUMBER(6,2)	
	coef_skew_prcp_amt	NUMBER(4,3)	

TABLE	sfc_clmo_rec

Description: The midas.sfc_clmo_rec table is the owner of midas.sfc_clmo_elem, which contains monthly and annual surface climatological averages and met element check values for surface CLIMAT reports. These values are calculated by the CARLOS and CLIMAT teams. A sfc_clmo_rec row is for the specified src_id, clmo_end_date, year_count and met_domain_name. This table is the owner of a one or more sfc_clmo_elem entries. It defines the id_type and id used by the source, constrains clmo_end_date to the last day of the year, and enforces relationships to midas.source and midas.src_met_domain. midasupd.sfc_clmo_rec is the corresponding updatable view, with one-for-one projection of the columns. It projects all of the rows. midasvu.sfc_clmo_rec is a read-only synonym of midasupd.sfc_clmo_rec. carlos.mo_sfc_rec is a corresponding updatable view, with one-for-one projection of the columns, but is restricted to CARLOS rows (using met_domain_name).

PK	Attribute	Datatype	Description / Units / Precision
*	src_id	NUMBER(6)	Unique source identifier
*	clmo_end_date	DATE	

*	clmo_year_cnt	NUMBER(2)	
*	met_domain_name	VARCHAR2(8)	
	rec_st_ind	NUMBER(4)	State indicator for the record
	id	VARCHAR2(8)	
	id_type	VARCHAR2(4)	Identifier type

VIEW	soil_temp_ob

Description: This entity contains soil temperatures reported daily and hourly. Time (hour) is part of the key. Daily soil temperatures are usually measured at 0900Z. NCMs send 30 and 100 cm temperatures. F3208 can send 5, 10, 20, 30, 50 and 100 cm temperatures, but all values are optional. Hourly soil temperatures are usually at 10 cm. Identifier is climatological number (DCNN) or WMO station number. Non-key attributes are copied from mp.soil and mp.hrsoil.yyy.

PK	Attribute	Datatype	Description / Units / Precision
*	ob_time	DATE	Date and time of observation
*	id_type	VARCHAR2(4)	Identifier type
*	id	VARCHAR2(8)	
*	met_domain_name	VARCHAR2(8)	
*	version_num	NUMBER(1)	Observation version number
	src_id	NUMBER(6)	Unique source identifier
	rec_st_ind	NUMBER(4)	State indicator for the record
	q5cm_soil_temp	NUMBER(3,1)	5cm soil temperature 0.1 deg C
	q5cm_soil_temp_q	NUMBER(5)	QC code - 5cm soil temperature
	q5cm_soil_temp_j	CHAR(1)	Descriptor - 5cm soil temp
	q10cm_soil_temp	NUMBER(3,1)	10cm soil temperature 0.1 deg C
	q10cm_soil_temp_q	NUMBER(5)	QC code - 10cm soil temperature
	q10cm_soil_temp_j	CHAR(1)	Descriptor - 10cm soil temp
	q20cm_soil_temp	NUMBER(3,1)	20cm soil temperature 0.1 deg C
	q20cm_soil_temp_q	NUMBER(5)	QC code - 20cm soil temperature
	q20cm_soil_temp_j	CHAR(1)	Descriptor - 20cm soil temp
	q30cm_soil_temp	NUMBER(3,1)	30 cm soil temperature 0.1 deg C
	q30cm_soil_temp_q	NUMBER(5)	QC code - 30cm soil temperature
	q30cm_soil_temp_j	CHAR(1)	Descriptor - 30cm soil temp
	q50cm_soil_temp	NUMBER(3,1)	50cm soil temperature 0.1 deg C
	q50cm_soil_temp_q	NUMBER(5)	QC code - 50cm soil temperature
	q50cm_soil_temp_j	CHAR(1)	Descriptor - 50cm soil temp
	q100cm_soil_temp	NUMBER(3,1)	100cm soil temperature 0.1 deg C
	q100cm_soil_temp_q	NUMBER(5)	QC code - 100cm soil temperature
	q100cm_soil_temp_j	CHAR(1)	Descriptor - 100cm soil temp

	meto_stmp_time	DATE	Met Office receipt stamp time
	midas_stmp_etime	NUMBER(6)	Elapsed time to storage in MIDAS minutes

---

TABLE	source

Description: A source is a station where meteorological readings are made. Its location is defined as the location of the barometer or the rain gauge, or other principal instrument. A source changes its identity (i.e. it becomes a new source) when the location of the principal instrument changes by more than a specified amount, e.g. by 400 metres or more for a rainfall station. A source may change its identity under other circumstances (e.g. a change of exposure or if it closes and re-opens). A source must have at least one capability, and that must use an identifier of a specified id_type. Start and end dates refer to the opening and final closing of the source - it may have been closed for one or more periods within these dates. A source may be re-opened, and re-use a src_id, provided the details defined in this entity are the same. A source will not exist if it has no observations, but one may be created in advance, where it is known that a station is due to open. A source is in a fixed position. OPR cannot supply or maintain source information for ships. On-station Ocean Weather Ships are treated as a fixed source with a notional latitude and longitude. They will have a source record, with a src_name of 'OWS A', 'OWS C', etc. and appropriate call-sign identifiers. Latitude and longitude at time of report are attributes of the report. This entity does not describe the reporting practice of individual elements or report types.

The src_id, src_name, high_prcn_lat, high_prcn_lon, loc_geog_area_id, rec_st_ind and grid_ref_type are all mandatory columns. The grid_ref_type will default to a value of 'XX' for overseas stations. The high_prcn_lat and high_prcn_lon will be automatically calculated on insert for those stations identified by grid reference values. The src_name value is validated on insert and update to ensure that it is unique for all stations in the UK and Ireland. Duplicate source names may exist within the database historically but all new source names will be checked for uniqueness.

NB: The entity has a self-referencing relationship, using parent_src_id, as required by the Metadata project. It also supports cross-referencing to other sources for a specified purpose, using relationships with the cross_reference entity. This duplication will be resolved at the next opportunity.

PK	Attribute	Datatype	Description / Units / Precision
*	src_id	NUMBER(6)	Unique source identifier
	src_name	VARCHAR2(40)	
	high_prcn_lat	NUMBER(5,3)	0.001 deg
	high_prcn_lon	NUMBER(6,3)	0.001 deg
	loc_geog_area_id	VARCHAR2(4)	Identifier for geographic area where source is located
	src_bgn_date	DATE	Effective from date
	rec_st_ind	NUMBER(4)	State indicator
	src_type	VARCHAR2(8)	Source type
	grid_ref_type	VARCHAR2(4)	Grid reference type
	east_grid_ref	NUMBER(6)	Easting grid reference
	north_grid_ref	NUMBER(7)	Northing grid reference
	hydr_area_id	NUMBER(4)	Hydrometric area identifier
	post_code	VARCHAR2(9)	
	src_end_date	DATE	Effective to date
	elevation	NUMBER(4)	Metres

	wmo_region_code	CHAR(1)	
	parent_src_id	NUMBER(6)	
	zone_time	NUMBER(2)	Hours
	drainage_stream_id	VARCHAR2(4)	Drainage stream identifier
	src_upd_date	DATE	Record updated date
	mtce_ctre_code	VARCHAR2(4)	
	place_id	NUMBER(6)	

TABLE	src_capability

Description: This entity defines which types of observation (met_domain) a source is capable of producing (and MIDAS will store), e.g. London/Gatwick is capable of producing SYNOPs and NCMs, while Southend is only capable of producing METARs. It does not record whether those reports have actually been produced. Some stations (e.g. Beaufort Park) use more than one identifier of the same type (e.g. WMO number 03693 for manned observations and 03694 for SAMOS) and so there will be two capabilities for this source. Changes over time are recorded. A src_capability is closed when the src_cap_end_date attribute is set to a value before the current date. If the capability is subsequently required again by the source, then the record may be re-opened by either resetting the end date or by creating a new record. A capability is not automatically created upon receipt of a new source or new meteorological domain. A source capability can be deleted when it is open.

The prime_capability_flag is a single character to indicate if the capability is the prime one of its type for the specified station, i.e. the prime daily rainfall capability for the station. Valid values are T and F. For each met domain, one and only one capability can be set to prime at one time. When SAMOS or similar equipment is trialed at a site, the site may continue to report using its current identifier and prime_capability_flag = T while the SAMOS uses a new capability and identifier with prime_capability_flag = F. When the trial concludes and the SAMOS becomes operational, the old capability is changed to prime_capability_flag = F (and is usually closed), while the new capability (and identifier) is updated to prime_capability_flag = T.

NB: Currently, rcpt_method_name and comm_mthd_id are both attributes of this entity. The rcpt_method_name attribute is carried forward from CA-IDMS, while comm_mthd_id is required by the Metadata project. At the first available opportunity,rcpt_method_name will be removed. All capability records with rec_st_ind = 2000 are located in the midas.src_capability_nodata table.

PK	Attribute	Datatype	Description / Units / Precision
*	id_type	VARCHAR2(4)	Identifier type
*	id	VARCHAR2(8)	
*	met_domain_name	VARCHAR2(8)	
*	src_cap_bgn_date	DATE	Effective from date
	src_id	NUMBER(6)	Unique source identifier
	rec_st_ind	NUMBER(4)	State indicator
	prime_capability_flag	CHAR(1)	See above
	src_cap_end_date	DATE	Effective to date
	first_online_ob_yr	NUMBER(4)	Year of latest observation
	db_segment_name	VARCHAR2(12)	
	rcpt_method_name	VARCHAR2(20)	Cross-reference between identifiers at a source
	data_retention_period	NUMBER(3)	
	comm_mthd_id	NUMBER(6)	

TABLE	src_capability_nodata

Description: This structure of this entity is identical to the midas.src_capability table, but it only contains records with rec_st_ind = 2000. These records have no data recorded in MIDAS for this capability. They have been placed in this table to simplify views and prevent cross-products in queries.

TABLE	src_met_domain

Description: The midas.src_met_domain table contains the valid combinations of src_id and met_domain_name. This allows the database to enforce a relationship between observation tables and src_capability. To expedite the Metadata Project, we will not enforce relationships between src_met_domain and either the source or met_domain tables at present. These relationships will be added when convenient to the Metadata project, and the direct relationships between source and src_capability and between met_domain and src_capability will be removed at the same time. A src_capability trigger inserts rows into src_met_domain when a new combination of src_id and met_domain_name is stored in src_capability. midasupd.src_met_domain is the corresponding updatable view, with one-for-one projection of the columns. midasvu.src_met_domain is a read-only synonym of midasupd.src_met_domain.

PK	Attribute	Datatype	Description / Units / Precision
*	src_id	NUMBER(6)	Unique source identifier
*	met_domain_name	VARCHAR2(8)	

TABLE	src_remark

Description: This table contains textual remarks concerning the location, topography, and exposure of a source, as well as general remarks and subjective comments on quality. Remarks can be added for a source even if it is closed (e.g. to indicate why it closed, or when it is likely to re-open, or to comment upon the observations from it). The src_rmrk_num attribute is stored DEScending so that when searching the associated index, the latest remarks will be found first.

PK	Attribute	Datatype	Description / Units / Precision
*	src_id	NUMBER(6)	Unique source identifier
*	src_rmrk_num	NUMBER(4)	Remark number
	rmrk_type	VARCHAR2(30)	Remark type
	src_rmrk_bgn_date	DATE	Effective from date
	src_rmrk_txt	VARCHAR2(120)	Remark text
	rec_st_ind	NUMBER(4)	State indicator
	src_rmrk_end_date	DATE	Effective to date

TABLE	src_runway

Description: A runway is a facility at a source.

PK	Attribute	Datatype	Description / Units / Precision
*	src_id	NUMBER(6)	Unique source identifier
*	runway_name	VARCHAR2(4)	Name of the runway
	rec_st_ind	NUMBER(4)	State indicator
	src_fac_bgn_date	DATE	Effective from date
	src_fac_end_date	DATE	Effective to date
	runway_height	NUMBER(5)	Runway height in metres
	rnwy_sig_ind	VARCHAR2(1)	METDB STTE indicator
	rnwy_bearing	NUMBER(5)	Runway bearing

TABLE	synthetic_glbl_wx

Description: The midas.synthetic_glbl_wx table contains meteorological values synthesised at 3-hrly intervals for non-UK stations. These values are created by Technical / Forecasting Developments Branch / Local Forecasting R&D group (LF R&D) and are used to provide adequate data coverage for data-sparse areas. Daily ingestion is not required - data will be added as and when LF R&D generate the pseudo-observations. An initial set of pseudo-station data will be added for areas of the World requested by Met Office defence customers. Once these data are added to MIDAS there will not be routine updating of pseudo-observations for these stations. Extra pseudo-observations might be added for these stations at the request of the customer. New pseudo-station data will be added as and when the customer requires. These additions will usually be in response to World events of interest to our defence customers. The attributes of this entity are similar to those of midas.weather_hrly_ob, but with significant differences. Some of the attributes refer to 24hr values. The data are not normalised. The period of maximum and minimum temperatures are defined by - Max temp at 18Z and Min temp at 06Z, both cover the previous 24 hour period to the time.
Minimum volume = 5 years of data (per station)
Maximum volume = 25 years of data (per station)
Data will be retained permanently, although a stations data might be completely replaced if the science used to create the pseudo-observations is improved sufficiently to warrant a complete replacement.

Neither id_type nor id are attributes of this table, because they relate to reporting practices and therefore are not appropriate to a table of synthetic values. The met_domain_name atttribute is also absent, because all the rows would have the same value. This table does not have a relationship to src_capability. It does have a foreign-key relationhip to source, using src_id. Records in source will be created to identify the locations for which synthetic_glbl_wx values are created. These synthetic sources will have a src_type of 'SYNTHETIC', and will be created within a block, with adequate allowance for future growth.

The following measures will ensure that casual users cannot mistake these synthetic values for observed values:

- The synthetic values reside in a separate table. The table name and attributes do not contain the phrase Ob.
- Only specified users will have access to synthetic_glbl_wx.
- There will not be any views linking synthetic_glbl_wx to any other table.

Retrievals will usually ask for all the data at a specified src_id, i.e. are more likely to query by place rather than time. Data will be added to the table a station at a time. The table has two-dimensional partitioning, using wmo_region_code and time. midasupd.synthetic_glbl_wx is the corresponding updatable view, with one-for-one projection of the columns.

PK	Attribute	Datatype	Description / Units / Precision
*	time	DATE	Date and time of observation
*	src_id	NUMBER(6)	Unique source identifier
*	wmo_region_code	CHAR(1)	1 - 7
	wind_direction	NUMBER(3)	Degs true
	wind_speed	NUMBER(3)	Knots
	prst_wx_id	CHAR(2)	Present weather code
	past_wx_id_1	CHAR(1)	Past weather code #1
	past_wx_id_2	CHAR(1)	Past weather code #2
	cld_ttl_amt_id	CHAR(1)	Total cloud amount code oktas
	cld_base_amt_id	CHAR(1)	Cloud base amount code
	cld_base_ht	NUMBER(4)	Cloud base height code decametres
	low_cld_type_id	CHAR(1)	Low cloud type code
	med_cld_type_id	CHAR(1)	Medium cloud type code
	hi_cld_type_id	CHAR(1)	High cloud type code
	visibility	NUMBER(4)	Decametres
	air_temperature	NUMBER(3,1)	Surface air temperature 0.1 deg C
	wetb_temp	NUMBER(3,1)	Wet bulb temperature 0.1 deg C
	dewpoint	NUMBER(3,1)	Dewpoint temperature 0.1 deg C
	stn_pres	NUMBER(5,1)	Station air pressure 0.1 hpa
	pres_sfc	NUMBER(5,1)	Pressure surface
	pres_sfc_ht	NUMBER(5)	Height of pressure surface
	msl_pressure	NUMBER(5,1)	Mean sea level air pressure 0.1 hpa
	pres_tdcy_amt	NUMBER(4,1)	Air pressure tendency
	prcp_ob_hr_cnt	NUMBER(3)	Precipitation hour count
	prcp_amt	NUMBER(5,1)	Precipitation amount
	cld_amt_id_1	CHAR(1)	Layer cloud amount code #1
	cloud_type_id_1	CHAR(1)	Cloud type code #1
	cld_base_ht_id_1	NUMBER(4)	Cloud base height code #1 decametres
	cld_amt_id_2	CHAR(1)	Layer cloud amount code #2
	cloud_type_id_2	CHAR(1)	Cloud type code #2
	cld_base_ht_id_2	NUMBER(4)	Cloud base height code #2 decametres
	cld_amt_id_3	CHAR(1)	Layer cloud amount code #3
	cloud_type_id_3	CHAR(1)	Cloud type code #3
	cld_base_ht_id_3	NUMBER(4)	Cloud base height code #3 decametres
	max_air_temp	NUMBER(3,1)	Maximum air temperature
	min_air_temp	NUMBER(3,1)	Minimum air temperature
	ground_state_id	CHAR(2)	Ground state id

	min_grss_temp	NUMBER(3,1)	Grass temperature
	snow_depth	NUMBER(4)	Snow depth
	sun_ob_hr_cnt	NUMBER(3)	Sunshine observation hour count
	sun_dur	NUMBER(3,1)	Sunshine duration
	q24hr_prcp_amt	NUMBER(5,1)	24hr precipitation amount
	q24hr_pres_tdcy_amt	NUMBER(4,1)	24hr air pressure tendency
	vert_vsby	NUMBER(3)	Vertical visibility decametres
	max_gust_speed	NUMBER(3)	Maximum gust speed

VIEW	temp_drnl_ob

Description: This entity contains maximum and minimum temperatures measured over a period of up to 24 hours. ob_end_time and ob_hour_count define the observation period. Attributes with different time-scales will not be compressed into a single row. For air temperatures, the observation period will usually be one of the following: 09-09Z, 09-21Z, 21-09Z. For minimum grass temperature and minimum concrete temperature ob_end_time is most usually 0900. Where the period of exposure is known to be 21-09Z, then ob_hour_count = 12. For some stations, the period of exposure is not known, but could be as much as 09-09Z, so a nominal ob_hour_count = 12 is used. MIDAS will store rows in this table using climatological station number, i.e. id_type = 'DCNN'. If the station does not have a DCNN, it will use WMO number, i.e. id_type = 'WMO'. Non-key attributes are from mp.dmwr and mc.daily.

PK	Attribute	Datatype	Description / Units / Precision
*	ob_end_time	DATE	Date and time at end of observation
*	id_type	VARCHAR2(4)	Identifier type
*	id	VARCHAR2(8)	
*	ob_hour_count	NUMBER(3)	Observation hour count
*	version_num	NUMBER(1)	Observation version number
*	met_domain_name	VARCHAR2(8)	
	src_id	NUMBER(6)	Unique source identifier
	rec_st_ind	NUMBER(4)	State indicator for the record
	max_air_temp	NUMBER(3,1)	Maximum air temperature 0.1 deg C
	min_air_temp	NUMBER(3,1)	Minimum air temperature 0.1 deg C
	min_grss_temp	NUMBER(3,1)	Minimum grass temperature 0.1 deg C
	min_conc_temp	NUMBER(3,1)	Minimum concrete temperature 0.1 deg C
	max_air_temp_q	NUMBER(5)	QC code - max air temperature
	min_air_temp_q	NUMBER(5)	QC code - min air temperature
	min_grss_temp_q	NUMBER(5)	QC code - min grass temp
	min_conc_temp_q	NUMBER(5)	QC code - min concrete temp
	max_air_temp_j	CHAR(1)	Descriptor - max air temp
	min_air_temp_j	CHAR(1)	Descriptor - min air temp
	min_grss_temp_j	CHAR(1)	Descriptor - min grass temp
	min_conc_temp_j	CHAR(1)	Descriptor - min concrete temp

	meto_stmp_time	DATE	Met Office receipt stamp time
	midas_stmp_etime	NUMBER(6)	Elapsed time to storage in MIDAS minutes

---

VIEW	temp_min_soil_ob

Description: This entity contains bare soil minimum temperatures recorded at 09Z each day. These data were stored in mp.dmwr and mc.daily. MIDAS stores rows in this table using climatological station number, i.e. id_type = 'DCNN'. ob_end_time = '0900, and ob_hour_count = 12, are constants, and are not included as attributes. These temperatures have not been reported since 1970. The corresponding views for this table are called midasupd.soil_min_temp_ob and midasvu.soil_min_temp_ob

PK	Attribute	Datatype	Description / Units / Precision
*	id_type	VARCHAR2(4)	Identifier type
*	id	VARCHAR2(8)	
*	ob_date	DATE	Date of observation
*	version_num	NUMBER(1)	Version number Not present in corresponding views
*	met_domain_name	VARCHAR2(8)	
	src_id	NUMBER(6)	Unique source identifier
	rec_st_ind	NUMBER(4)	State indicator for the record
	min_soil_temp	NUMBER(3,1)	Minimum soil temperature 0.1 deg C
	min_soil_temp_q	NUMBER(5)	Quality code - minimum soil temperature Not present in corresponding views

---

VIEW	weather_drnl_ob

Description: This entity contains meteorological values measured on a 24 hour time scale. ob_end_time and ob_hour_count define the observation period. Sunshine duration, snow day, hail day, thunder day and gale day are usually for the period 0000-2359Z, i.e. ob_end_time = '2359' and ob_hour_count = 24. Concrete state, lying snow, snow depth, fresh snow depth and fresh mountain snowfall are "spot" values, usually at 0900Z, i.e. ob_end_time = 0900 and ob_hour_count = 0. Daily rainfall, daily radiation, daily temperature ranges and bare soil minimum temperatures are stored in separate entities. Day of fog at 0900Z is not included. MIDAS will store rows in this table using climatological station number, e.g. id_type = 'DCNN'. If the station does not have a DCNN, it will use id_type = 'WMO'. Non-key attributes are taken from mp.dmwr and mc.daily. Attributes with different time scales are not compressed into a single row.

PK	Attribute	Datatype	Description / Units / Precision
*	ob_end_time	DATE	Date and time at end of observation
*	id	VARCHAR2(8)	
*	id_type	VARCHAR2(4)	Identifier type
*	ob_hour_count	NUMBER(3)	Observation hour count
*	version_num	NUMBER(1)	Observation version number
*	met_domain_name	VARCHAR2(8)	

	src_id	NUMBER(6)	Unique source identifier
	rec_st_ind	NUMBER(4)	State indicator for the record
	cs_24hr_sun_dur	NUMBER(3,1)	Campbell-Stokes sunshine duration 0.1 hr
	wmo_24hr_sun_dur	NUMBER(3,1)	
	conc_state_id	CHAR(1)	Concrete state code
	lying_snow_flag	CHAR(1)	Lying snow flag
	snow_depth	NUMBER(4)	Snow depth cm
	frsh_snw_amt	NUMBER(4)	Fresh snow amount cm
	snow_day_id	CHAR(1)	Snow day code
	hail_day_id	CHAR(1)	Hail day code
	thunder_day_flag	CHAR(1)	Thunder day flag
	gale_day_flag	CHAR(1)	Gale day flag
	frsh_mnt_snwfall_flag	CHAR(1)	Fresh mountain snowfall flag
	lying snow height	NUMBER(3)	Lying snow height decametres
	cs_24hr_sun_dur_q	NUMBER(5)	QC code - Campbell-Stokes sunshine duration
	wmo_24hr_sun_dur_q	NUMBER(5)	
	conc_state_id_q	NUMBER(5)	QC code - concrete state code
	snow_depth_q	NUMBER(5)	QC code - snow depth
	frsh_snw_amt_q	NUMBER(5)	QC code - fresh snow amount
	snow_day_id_q	NUMBER(5)	QC code - snow day code
	hail_day_id_q	NUMBER(5)	QC code - hail day code
	thunder_day_flag_q	NUMBER(5)	QC code - thunder day flag
	gale_day_flag_q	NUMBER(5)	QC code - gale day flag
	lying_snow_ht_q	NUMBER(5)	QC code – lying snow height
	meto_stmp_time	DATE	Met Office receipt stamp time
	midas_stmp_etime	NUMBER(6)	Elapsed time to storage in MIDAS minutes

---

VIEW	weather_hrly_ob

Description: This entity contains SYNOPs and METARs measured during the hour ending at the stated date and time. The identifier is climatological station number, DCNN or WMO station number, or ICAO-id. Non-key attributes are taken from mp.hmwr, mc.hdata and mop.hdata.egxx. It also contains sunshine duration measured during the hour ending at the specified time. Hourly sunshine is reported using Metform 3445, and the value is reported as "hour beginning". The MIDAS ingestion software makes the necessary transformation so that the observations are stored at "hour ending", consistent with other hourly data.

PK	Attribute	Datatype	Description / Units / Precision
*	ob_time	DATE	Date and time of observation
*	id	VARCHAR2(8)	
*	id_type	VARCHAR2(4)	Identifier type

*	met_domain_name	VARCHAR2(8)	
*	version_num	NUMBER(1)	Observation version number
	src_id	NUMBER(6)	Unique source identifier
	rec_st_ind	NUMBER(4)	State indicator for the record
	wind_speed_unit_id	CHAR(1)	Wind speed unit code
	src_opr_type	CHAR(1)	Source operation type code
	wind_direction	NUMBER(3)	Wind direction degs true
	wind_speed	NUMBER(3)	Wind speed knots
	prst_wx_id	CHAR(2)	Present weather code
	past_wx_id_1	CHAR(1)	Past weather code #1
	past_wx_id_2	CHAR(1)	Past weather code #2
	cld_ttl_amt_id	CHAR(1)	Total cloud amount code oktas
	low_cld_type_id	CHAR(1)	Low cloud type code
	med_cld_type_id	CHAR(1)	Medium cloud type code
	hi_cld_type_id	CHAR(1)	High cloud type code
	cld_base_amt_id	CHAR(1)	Cloud base amount code
	cld_base_ht	NUMBER(4)	Cloud base height code decametres
	visibility	NUMBER(4)	Visibility decametres
	msl_pressure	NUMBER(5,1)	Mean sea level air pressure 0.1 hpa
	cld_amt_id_1	CHAR(1)	Layer cloud amount code #1
	cloud_type_id_1	CHAR(1)	Cloud type code #1
	cld_base_ht_id_1	NUMBER(4)	Cloud base height code #1 decametres
	cld_amt_id_2	CHAR(1)	Layer cloud amount code #2
	cloud_type_id_2	CHAR(1)	Cloud type code #2
	cld_base_ht_id_2	NUMBER(4)	Cloud base height code #2 decametres
	cld_amt_id_3	CHAR(1)	Layer cloud amount code #3
	cloud_type_id_3	CHAR(1)	Cloud type code #3
	cld_base_ht_id_3	NUMBER(4)	Cloud base height code #3 decametres
	cld_amt_id_4	CHAR(1)	Layer cloud amount code #4
	cloud_type_id_4	CHAR(1)	Cloud type code #4
	cld_base_ht_id_4	NUMBER(4)	Cloud base height code #4 decametres
	vert_vsby	NUMBER(3)	Vertical visibility decametres
	air_temperature	NUMBER(3,1)	Air temperature 0.1 deg C
	dewpoint	NUMBER(3,1)	Dewpoint temperature 0.1 deg C
	wetb_temp	NUMBER(3,1)	Wet bulb temperature 0.1 deg C

	rltv_hum	NUMBER(4,1)	Calculated relative humidity
	stn_pres	NUMBER(5,1)	Station air pressure 0.1 hpa
	alt_pres	NUMBER(4)	Altimeter pressure 0.1 hpa
	ground_state_id	CHAR(2)	Ground state code
	q10mnt_mxgst_spd	NUMBER(3)	10 minute maximum gust speed knots
	cavok_flag	CHAR(1)	cavok flag
	cs_hr_sun_dur	NUMBER(3,1)	Campbell-Stokes hour sunshine duration
	wmo_hr_sun_dur	NUMBER(3,1)	
	wind_direction_q	NUMBER(5)	QC code - wind direction
	wind_speed_q	NUMBER(5)	QC code - wind speed
	prst_wx_id_q	NUMBER(5)	QC code - present weather code
	past_wx_id_1_q	NUMBER(5)	QC code - past weather code #1
	past_wx_id_2_q	NUMBER(5)	QC code - past weather code #2
	cld_ttl_amt_id_q	NUMBER(5)	QC code - cloud total amount
	low_cld_type_id_q	NUMBER(5)	QC code - low cloud type code
	med_cld_type_id_q	NUMBER(5)	QC code - medium cloud type code
	hi_cld_type_id_q	NUMBER(5)	QC code - high cloud type code
	cld_base_amt_id_q	NUMBER(5)	QC code - cloud base amount
	cld_base_ht_q	NUMBER(5)	QC code - cloud base height
	visibility_q	NUMBER(5)	QC code - visibility
	msl_pressure_q	NUMBER(5)	QC code - msl pressure
	air_temperature_q	NUMBER(5)	QC code - air temperature
	dewpoint_q	NUMBER(5)	QC code - dewpoint
	wetb_temp_q	NUMBER(5)	QC code - wet bulb
	ground_state_id_q	NUMBER(5)	QC code - ground state code
	cld_amt_id_1_q	NUMBER(5)	QC code - layer cloud amount #1
	cloud_type_id_1_q	NUMBER(5)	QC code - cloud type code #1
	cld_base_ht_id_1_q	NUMBER(5)	QC code - cloud base ht code #1
	cld_amt_id_2_q	NUMBER(5)	QC code - layer cloud amount #2
	cloud_type_id_2_q	NUMBER(5)	QC code - cloud type code #2
	cld_base_ht_id_2_q	NUMBER(5)	QC code - cloud base ht code #2
	cld_amt_id_3_q	NUMBER(5)	QC code - layer cloud amount #3
	cloud_type_id_3_q	NUMBER(5)	QC code - cloud type code #3
	cld_base_ht_id_3_q	NUMBER(5)	QC code - cloud base ht code #3
	cld_amt_id_4_q	NUMBER(5)	QC code - layer cloud amount #4
	cloud_type_id_4_q	NUMBER(5)	QC code - cloud type code #4
	cld_base_ht_id_4_q	NUMBER(5)	QC code - cloud base ht code #4
	vert_vsby_q	NUMBER(5)	QC code - vertical visibility
	stn_pres_q	NUMBER(5)	QC code - station pressure
	alt_pres_q	NUMBER(5)	QC code - altimeter pressure
	q10mnt_mxgst_spd_q	NUMBER(5)	QC code - 10 min max gust speed
	cs_hr_sun_dur_q	NUMBER(5)	

	wmo_hr_sun_dur_q	NUMBER(5)	
	wind_direction_j	CHAR(1)	Descriptor - wind direction
	wind_speed_j	CHAR(1)	Descriptor - wind speed
	prst_wx_id_j	CHAR(1)	Descriptor - present weather
	past_wx_id_1_j	CHAR(1)	Descriptor - past weather #1
	past_wx_id_2_j	CHAR(1)	Descriptor - past weather #2
	cld_amt_id_j	CHAR(1)	Descriptor - cloud total amt
	cld_ht_j	CHAR(1)	Descriptor - cloud base ht
	visibility_j	CHAR(1)	Descriptor - visibility
	msl_pressure_j	CHAR(1)	Descriptor - msl pressure
	air_temperature_j	CHAR(1)	Descriptor - air temperature
	dewpoint_j	CHAR(1)	Descriptor - dewpoint
	wetb_temp_j	CHAR(1)	Descriptor - wet bulb
	rltv_hum_j	VARCHAR2(1)	Descriptor - relative humidity
	vert_vsby_j	CHAR(1)	Descriptor - vertical vis
	stn_pres_j	CHAR(1)	Descriptor - station pressure
	alt_pres_j	CHAR(1)	Descriptor - altimeter press
	q10mnt_mxgst_spd_j	CHAR(1)	Descriptor - 10 min max gust
	meto_stmp_time	DATE	Met Office receipt stamp time
	midas_stmp_etime	NUMBER(6)	Elapsed time to storage in MIDAS minutes
	snow_depth	NUMBER(4)	Snow depth cm
	snow_depth_q	NUMBER(5)	QC code - snow depth

VIEW	wind_mean_ob

Description: This entity contains values of mean wind and gust direction and speed measured during one or more hours ending at the stated date and time. "Hour ending" is contrary to the CDB practice, which uses hour beginning at the stated time, but provides a uniform practice in MIDAS, and simplifies the creation of SQL views. The rows in this table are of two types:

- Wind and gust direction and speed measured over a WHOLE hour
  There may be hourly gust data from SYNOPs (with an id_type of WMO) where the gust speed is 25 knots or more. However these will be overwritten by later input - eg. hourly climate message data (from ESAWS or SAMOS), DALE tapes or Metform 6910.
- Mean value of wind speed for a 24 hour period, usually 09-09Z, obtained from run-of-wind.

This table is NOT intended for:

- Ten minute winds recorded in SYNOPs and METARs - they are stored in weather_hourly_ob
- Mean speeds for 24 hours, obtained by meaning the hourly means - they are derived.

Identifier is DCNN + anemograph site number.

PK	Attribute	Datatype	Description / Units / Precision

*	ob_end_time	DATE	Data and time at end of observation
*	id_type	VARCHAR2(4)	Identifier type
*	id	VARCHAR2(8)	
*	ob_hour_count	NUMBER(3)	Observation hour count
	met_domain_name	VARCHAR2(8)	
*	version_num	NUMBER(1)	Observation version number
	src_id	NUMBER(6)	Unique source identifier
	rec_st_ind	NUMBER(4)	State indicator for the record
	mean_wind_dir	NUMBER(3)	Mean wind direction degs true
	mean_wind_speed	NUMBER(4,1)	Mean wind speed knots
	max_gust_dir	NUMBER(3)	Direction of maximum gust degs true
	max_gust_speed	NUMBER(3)	Speed of maximum gust knots
	max_gust_ctime	NUMBER(4)	Clock-time of maximum gust hhmm
	mean_wind_dir_q	NUMBER(5)	QC code - mean wind direction
	mean_wind_speed_q	NUMBER(5)	QC code - mean wind speed
	max_gust_dir_q	NUMBER(5)	QC code - maximum gust direction
	max_gust_speed_q	NUMBER(5)	QC code - maximum gust speed
	max_gust_ctime_q	NUMBER(5)	QC code - maximum gust time
	mean_wind_dir_j	CHAR(1)	Descriptor - mean wind dirn
	mean_wind_speed_j	CHAR(1)	Descriptor - mean wind speed
	max_gust_dir_j	CHAR(1)	Descriptor - max gust dirn
	max_gust_speed_j	CHAR(1)	Descriptor - max gust speed
	meto_stmp_time	DATE	Met Office receipt stamp time
	midas_stmp_etime	NUMBER(6)	Elapsed time to storage in MIDAS minutes

VIEW	year_sfc_elem

Description: Midas.Year_Sfc_Elem table contains annual surface values of specified met elements. The Met_Element_Id column defines and constrains the met element via a foreign-key relationship. The owner Year_Sfc_Rec constrains Ob_Date and Met_Domain_Name, and implicitly defines Id_Type and Id. Carlos.Year_Sfc_Elem is the corresponding updatable view, with one-for-one projection of the columns. This view restricts rows to those with the appropriate met domain name, i.e. CARLOS and storage table name, i.e. Year_Sfc_Elem.

PK	Attribute	Datatype	Description / Units / Precision
*	Src_id	NUMBER(6)	
*	ob_date	DATE	
*	met_domain_name	VARCHAR2(8)	e.g. 'CARLOS'
*	met_element_id	NUMBER(5)	
*	version_num	NUMBER(1)	
*	rec_st_ind	NUMBER(4)	

	met_elem_year_day_cnt	NUMBER(2)	Count of days in year when the met element occurred
	met_elem_year_hour_cnt	NUMBER(3)	Count of hours in year when the met element occurred
	met_elem_min_val	NUMBER(6,2)	Minimum value of the met element
	met_elem_min_val_year_day_num	NUMBER(2)	Day of the year on which the min value occurred
	met_elem_mean_val	NUMBER(6,2)	Mean value of the met element
	met_elem_sd_val	NUMBER(6,2)	Standard deviation
	met_elem_max_val	NUMBER(6,2)	Maximum value of the met element
	met_elem_max_val_year_day_num	NUMBER(2)	Day of the year on which the max value occurred
	met_elem_miss_year_hour_cnt	NUMBER(3)	Count of hours in year when the element was missing
	met_elem_amt	NUMBER(6,2)	e.g. rainfall amount
	met_elem_occr_prc	NUMBER(3)	Percent of occurrences, e.g. NE wind
	met_elem_qual_id	CHAR(1)	Quality of the met element
	met_elem_sbst_year_hour_cnt	NUMBER(3)	Count of hours in year when substitute values used.
	met_elem_sbst_code	CHAR(2)	Defines substitute values
	last_update_time	DATE	Stores date/time when table was last updated using a trigger.

VIEW	year_sfc_rec

Description: This table is the owner of a one or more Year_Sfc_Elem entries for the given Source, Year and Met_Domain_Name. It defines the Id_Type and Id used by the Source, and constrains Ob_Date to the last day of the year. The Year_Sfc_Elem entries are annual calculated values from globally located sources. These values are primarily Carlos rainfall values derived from Midas data.

PK	Attribute	Datatype	Description / Units / Precision
*	src_id	NUMBER(6)	
*	ob_date	DATE	
*	met_domain_name	VARCHAR2(8)	e.g. 'CARLOS' etc.
	rec_st_ind	NUMBER(4)	
	id_type	VARCHAR2(4)	
	id	VARCHAR2(8)	
	wind_speed_unit_id	CHAR(1)	

9.3 MIDAS Tables and MIDASUPD Views - Marine

Observation Time Constraint
A time constraint exists on all MIDASUPD views, except those for standing data tables, such that no observation data with a date/time greater than one hour in the future can be stored in MIDAS (e.g. 1000 ob can be stored at

0952). Any rejections are notified. This will prevent instances of data 'labelled' with a date later than 'today' being loaded into MIDAS.

Cross references between Rigs and light vessels that have data stored in MIDAS under different IDs.
ID cross references

TABLE	mar_acq_log

Description: This entity records recent marine data acquisitions into the database. It contains the key of the data records (including lat/long), the stamp times and the batch number of those records. To assist in identifying recent data which is not yet subject to QC, the batch number is indexed.

All users may query this table, to ascertain if MIDAS contains their required data; this should prove more efficient than searching the meteorological data records. However, this table will contain only RECENT records where QC is not yet complete - typically this will be data less than 45 days old. A routine job will be used to remove data older than this from the data.

Attributes latitude and longitude are part of the primary key of this record, and have a precision of 0.1 degree. The attribute ob_hour_count is not included, because all marine observations are for the stated hour. Only precipitation has a period, and this is not part of its primary key. Attribute met_domain_name is not part of the primary key, because ships only store one type of report at a time in one table.

Procedure MAR_ACQ_DEL each day deletes data greater than 45 days old.

PK	Attribute	Datatype	Description / Units / Precision
*	ob_time	DATE	Observation date and time
*	latitude	NUMBER(3,1)	0.1 deg
*	longitude	NUMBER(4,1)	0.1 deg
*	id	VARCHAR2(8)	Identifier
*	id_type	VARCHAR2(4)	Identifier type
*	version_flag	CHAR(1)	
	met_domain_name	VARCHAR2(8)	
	longitude_band_code	CHAR(1)	
	batch_stamp_time	DATE	Met Office receipt stamp time
	ob_rcpt_code	CHAR(1)	
	midas_stmp_etime	NUMBER(6)	Elapsed time to storage in MIDAS minutes

---

TABLE	marine_climatology

Description: This table contains mean and standard deviation values of air temperature and sea-surface temperature for a 5-degree square and month. Latitude and longitude are for the SW corner of the square and have a precision of 0.1 degrees for consistency with other marine tables, but will be specified in whole degrees. The table will not contain values for squares which are wholly on land. These values are used for the daily quality control of marine observations. The period for which the values apply and the method of calculation are not known.

PK	Attribute	Datatype	Description / Units / Precision
*	month_number	NUMBER(2)	
*	latitude	NUMBER(3,1)	
*	longitude	NUMBER(4,1)	
	mean_air_temp	NUMBER(3,1)	
	sd_air_temperature	NUMBER(4,2)	
	mean_sea_surface_temperature	NUMBER(3,1)	
	sd_sea_surface_temperature	NUMBER(4,2)	

TABLE	marine_current

Description: This entity contains direction and speeds of marine currents. Time and position are the mid-points, calculated from a first and final position and times. Non-key attributes are copied from ma.main.ocn.

PK	Attribute	Datatype	Description / Units / Precision
*	ob_month	NUMBER(2)	Observation month
*	latitude	NUMBER(3,1)	0.1 deg
*	longitude	NUMBER(4,1)	0.1 deg
*	id	VARCHAR2(8)	
*	id_type	VARCHAR2(4)	Identifier type
*	met_domain_name	VARCHAR2(8)	
	rec_st_ind	NUMBER(4)	State indicator for the record
	ob_year	NUMBER(4)	Observation year
	ob_day	NUMBER(2)	Observation day
	ob_end_ctime	NUMBER(4)	Clock-time at observation end
	current_ob_period	NUMBER(2)	Duration of observation hours
	curr_dir	NUMBER(3)	Current direction
	current_speed	NUMBER(4,2)	0.1 knot
	ship_mean_draught	NUMBER(3)	Metres
	country_format_num	NUMBER(5)	

TABLE	marine_ice_ob

Description: The midas.marine_ice_ob table contains ice observations from ships. The observations are made during the hour ending at the stated date and time and are defined by position (latitude and longitude) and by time. Duplicates can exist at a specified position and time, e.g. when ships are alongside for bunkering, so the identifier of the ship is part of the primary key of the entity. Identifier is either call-sign or buoy-id, depending upon report type. Only about 2% of marine observations include ice reports, so the icing attributes of a marine report are stored in a separate table because of their very low occurrence, and to correct a problem with pre-1979 reports. Each marine_ice_ob row may contain 0, 1 or 2 corresponding entries in midas.marine_ob. A marine_ice_ob without a corresponding marine_ob will be very rare. Each marine_ob row may contain 0, 1 or 2 corresponding rows in marine_ice_ob. Most marine_ob rows will not have a corresponding entry in marine_ice_ob. Although the icing attributes do not have any QC, QEv may decide to include this table when the date, time or position of the corresponding marine_ob entry is QCd.

midasupd.marine_ice_ob is the corresponding updateable view, with one-for-one projection of the columns.
midasvu.marine_ice_ob is a read-only view, with one-for-one projection of the columns.

PK	Attribute	Datatype	Description / Units / Precision
*	ob_time	DATE	Date and time of observation
*	latitude	NUMBER(3,1)	0.1 deg
*	longitude	NUMBER(4,1)	0.1 deg
*	id	VARCHAR2(8)	
*	id_type	VARCHAR2(4)	Identifier type
	met_domain_name	VARCHAR2(8)	
	ice_version_flag	CHAR(1)	Observation version number
	ob_rcpt_code	CHAR(1)	Code for reception method
	rec_st_ind	NUMBER(3)	State indicator for the record
	country_format_num	NUMBER(5)	Country format number
	ice_format_id	VARCHAR2(2)	Pre1979: kind of ice 1979+: concentration/arrangement of sea ice code 0639
	ice_age_id	VARCHAR2(2)	Pre1979: effect of ice on navigation 1979+: stage of development code 3739
	land_ice_type_id	VARCHAR2(2)	Pre1979: bearing of principal ice edge 1979+: ice of land origin code 0439
	ice_edge_id	VARCHAR2(2)	Pre1979: distance of ice edge from reporting ship 1979+: bearing of principal ice edge code 0739
	ice_cond_trnd	VARCHAR2(2)	Pre1979: Orientation of ice edge 1979+: Ice situation and trend over proceeding 3 hours code 5239
	ice_accr_type	CHAR(1)	Ice accretion type
	ice_accr_thkn	NUMBER(3)	Ice accretion thickness cm
	ice_accr_rate	CHAR(1)	Ice accretion rate code
	ice_type_code	CHAR(1)	e2 kind of ice Pre1979 reports only
	ice_navigation_effect_code	CHAR(1)	K effect of ice on navigation Pre1979 reports only
	ice_edge_distance_code	CHAR(1)	r distance of ice edge from reporting ship Pre1979 reports only
	ice_edge_orientation_code	CHAR(1)	e orientation of ice edge Pre1979 reports only
	ice_format_id_q	NUMBER(5)	QC code – Concentration
	ice_age_id_q	NUMBER(5)	QC code – Stage of development

	land_ice_type_id_q	NUMBER(5)	QC code – Ice of land origin
	ice_edge_id_q	NUMBER(5)	QC code – Bearing of principal edge
	ice_cond_trnd_q	NUMBER(5)	QC code – Ice situation and trend
	ice_accr_type_q	NUMBER(5)	QC code – Accretion type
	ice_accr_thkn_q	NUMBER(5)	QC code – Accretion thickness
	ice_accr_rate_q	NUMBER(5)	QC code – Accretion rate

Triggers

Trigger Name	marine_ice_upd
When Fired	Row level update trigger fired after each row is updated
Description	If changes are made to any of the fields which can be QC'd then the old values from before the update are stored for use by the trigger MARINE_ICE_UPD_V0. Details are only stored if the field(s) that has been updated was not previously set to NULL

Trigger Name	marine_ice_upd_v0
When Fired	Statement level update trigger fired after updates
Description	The trigger will retain the orginal version of the MARINE_ICE_OB record as a version ZERO row. If a version ZERO doesn't exist then insert one. If version ZERO exists but contains NULL values for met elements then update the version ZERO record with the non-NULL values. If a version ZERO exists and non-NULL values are present, then these will not be updated. This protects the original non-NULL values that were entered.

Trigger Name	marine_ice_del_adr
When Fired	Statement level update trigger fired after deletes
Description	The trigger will remove the associated v0 rows if a v1 row is being deleted.

Trigger Name	marine_ice_del_ad
When Fired	Table level update trigger fired after deletes
Description	The trigger will remove the associated v0 rows if a v1 row is being deleted.

TABLE	marine_ob

Description: The midas.marine_ob table contains marine meteorological values measured during the hour ending at the stated date and time. Marine reports are defined by position (latitude and longitude) and by time. Duplicates can exist at a specified position and time, e.g. when ships are alongside for bunkering, so the identifier of the ship or buoy is part of the primary key of the entity. Identifier is either call-sign or buoy-id, depending upon report type. Most marine observations do not include cloud layers, ice reports or precipitation. Only about 2% of marine observations include these values so ice attributes are stored in midas.marine_ice_ob. The version_flag is a CHAR(1) variation of version_num - this was introduced to improve space utilisation and shorten the primary key.

Most data enquiries are of two types:

- a small area for a few days or,
- a small area for many years or a whole time-series

Consequently, there are Fortran data extraction routines for one 10-deg-sq for one day, and one 10-deg-sq for one month. To retrieve a few days, the first routine is called a number of times. To get many years the second routine is called repeatedly. Two additional Fortran data extraction routines are provided for QC. These extract one ship (ID) for one day and one ship (ID) for one month. They allow checks on the continuity of observations. All new data gets QCd. Sometimes, large batches of old data (e.g. 1,000,000 obs) are received, and this also gets QCd, but this does not happen very often. When UK land data is more than three years old, the "all times for one place" type of enquiry tends to take over from the "all places for one time" queries. Marine enquiries do not have this change in usage.

This table has two-dimensional partitioning using longitude_band_code (A -J) and by ob_time, with an equi-partioned primary index. midasupd.marine_ob is the corresponding updateable view, with one-for-one projection of the columns.

There are five triggers associated with this table. They are listed after the description of the marine_ob table.

PK	Attribute	Datatype	Description / Units / Precision
*	ob_time	DATE	Date and time of observation
*	latitude	NUMBER(3,1)	0.1 deg
*	longitude	NUMBER(4,1)	0.1 deg
*	id	VARCHAR2(8)	
*	id_type	VARCHAR2(4)	Identifier type
*	version_flag	CHAR(1)	Observation version number
*	longitude_band_code	CHAR(1)	
	met_domain_name	VARCHAR2(8)	
	ob_rcpt_code	CHAR(1)	Code for reception method
	rec_st_ind	NUMBER(4)	State indicator for the record
	country_format_num	NUMBER(5)	Country format number
	src_opr_type	CHAR(1)	Source operation type code
	wind_direction	NUMBER(3)	Degs true
	wind_speed	NUMBER(3)	Knots
	prst_wx_id	CHAR(2)	Present weather code
	past_wx_id_1	CHAR(1)	Past weather code #1
	past_wx_id_2	CHAR(1)	Past weather code #2
	cld_ttl_amt_id	CHAR(1)	Total cloud amount code

	low_cld_type_id	CHAR(1)	Low cloud type code
	med_cld_type_id	CHAR(1)	Medium cloud type code
	hi_cld_type_id	CHAR(1)	High cloud type code
	cld_base_amt_id	CHAR(1)	Cloud base amount code
	cld_base_ht	NUMBER(4)	Cloud base height decametres
	msl_pressure	NUMBER(5,1)	Mean sea level air pressure 0.1 hpa
	visibility	NUMBER(4)	Decametres
	air_temperature	NUMBER(3,1)	Air temperature 0.1 deg C
	dewpoint	NUMBER(3,1)	Dewpoint temperature 0.1 deg C
	wetb_temp	NUMBER(3,1)	Wetbulb temperature 0.1 deg C
	rltv_hum	NUMBER(4,1)	Calculated relative humidity
	sea_temperature	NUMBER(3,1)	0.1 deg C
	pres_tdcy_id	CHAR(1)	Pressure tendency characteristic code
	pres_tdcy_amt	NUMBER(4,1)	Amount of pressure change 0.1 hpa
	wind_wave_per	NUMBER(3,1)	Wind wave period 0.1 sec
	wind_wave_ht	NUMBER(3,1)	Wind wave height 0.1 metre
	swell_dir_1	NUMBER(3)	Swell direction code #1
	swell_period_1	NUMBER(3,1)	Swell wave period #1 0.1 sec
	swell_height_1	NUMBER(3,1)	Swell height #1 0.1 metre
	swell_dir_2	NUMBER(3)	Swell direction code #2
	swell_period_2	NUMBER(3,1)	Swell wave period #2 0.1 sec
	swell_height_2	NUMBER(3,1)	Swell height #2 0.1 metre
	msr_wave_dir	NUMBER(3)	Wave direction deg true
	msr_wave_per	NUMBER(3,1)	Measured wave period 0.1 sec
	msr_wave_height	NUMBER(3,1)	Measured wave height 0.1 metre
	ship_direction	CHAR(1)	Ship direction code
	ship_distance	NUMBER(3)	Ship distance nautical miles
	location_q	NUMBER(5)	QC code - location
	wind_direction_q	NUMBER(5)	QC code - wind direction
	wind_speed_q	NUMBER(5)	QC code - wind speed
	prst_wx_id_q	NUMBER(5)	QC code - present weather code

	past_wx_id_1_q	NUMBER(5)	QC code - past weather code #1
	past_wx_id_2_q	NUMBER(5)	QC code - past weather code #2
	cld_ttl_amt_id_q	NUMBER(5)	QC code - cloud total amount
	low_cld_type_id_q	NUMBER(5)	QC code - low cloud type code
	med_cld_type_id_q	NUMBER(5)	QC code - medium cloud type code
	hi_cld_type_id_q	NUMBER(5)	QC code - high cloud type code
	cld_base_amt_id_q	NUMBER(5)	QC code - cloud base amount
	cld_base_ht_q	NUMBER(5)	QC code - cloud base height
	visibility_q	NUMBER(5)	QC code - visibility
	msl_pressure_q	NUMBER(5)	QC code - msl pressure
	air_temperature_q	NUMBER(5)	QC code - air temperature
	dewpoint_q	NUMBER(5)	QC code - dewpoint
	wetb_temp_q	NUMBER(5)	QC code - wetbulb temperature
	sea_temperature_q	NUMBER(5)	QC code - sea temperature
	wind_wave_per_q	NUMBER(5)	QC code - wind wave period
	wind_wave_ht_q	NUMBER(5)	QC code - wind wave height
	swell_dir_1_q	NUMBER(5)	QC code - swell direction #1
	swell_period_1_q	NUMBER(5)	QC code - swell period #1
	swell_height_1_q	NUMBER(5)	QC code - swell height #1
	swell_dir_2_q	NUMBER(5)	QC code - swell direction #2
	swell_period_2_q	NUMBER(5)	QC code - swell period #2
	swell_height_2_q	NUMBER(5)	QC code - swell height #2
	msr_wave_per_q	NUMBER(5)	QC code - wave period
	msr_wave_height_q	NUMBER(5)	QC code - msr half metre wave ht
	location_j	CHAR(1)	Descriptor - location
	wind_direction_j	CHAR(1)	Descriptor - wind direction
	wind_speed_j	CHAR(1)	Descriptor - wind speed
	prst_wx_id_j	CHAR(1)	Descriptor - present weather
	past_wx_id_j	CHAR(1)	Descriptor - past weather
	cld_ht_j	CHAR(1)	Descriptor - cloud height
	visibility_j	CHAR(1)	Descriptor - visibility
	msl_pressure_j	CHAR(1)	Descriptor - msl pressure
	air_temperature_j	CHAR(1)	Descriptor - air temperature
	dewpoint_j	CHAR(1)	Descriptor - dewpoint
	wetb_temp_j	CHAR(1)	Descriptor - wetbulb temp
	rltv_hum_j	VARCHAR2(1)	Descriptor - relative humidity
	sea_temperature_j	CHAR(1)	Descriptor - sea temperature
	wind_wave_j	CHAR(1)	Descriptor - wind wave
	swell_j	CHAR(1)	Descriptor - swell wave
	msr_wave_j	CHAR(1)	Descriptor - measured wave
	batch_stamp_time	DATE	Time at start of the ingestion batch containing this row

	metdb_delay_etime	NUMBER(6)	Difference between time of observation and time of storage in MetDB
	cld_amt_id_1	CHAR(1)	Layer cloud amount code #1
	cloud_type_id_1	CHAR(1)	Cloud type code #1
	cld_base_ht_id_1	NUMBER(4)	Cloud base height code #1 decametres
	cld_amt_id_2	CHAR(1)	Layer cloud amount code #2
	cloud_type_id_2	CHAR(1)	Cloud type code #2
	cld_base_ht_id_2	NUMBER(4)	Cloud base height code #2 decametres
	cld_amt_id_3	CHAR(1)	Layer cloud amount code #3
	cloud_type_id_3	CHAR(1)	Cloud type code #3
	cld_base_ht_id_3	NUMBER(4)	Cloud base height code #3 decametres
	cld_amt_id_4	CHAR(1)	Layer cloud amount code #4
	cloud_type_id_4	CHAR(1)	Cloud type code #4
	cld_base_ht_id_4	NUMBER(4)	Cloud base height code #4 decametres
	vert_vsby	NUMBER(3)	Vertical visibility
	prcp_ob_hr_cnt	NUMBER(3)	Precipitation hour count
	prcp_amt	NUMBER(5,1)	Precipitation amount mm
	cld_lyr_1_q	NUMBER(5)	QC code - layer cloud amount #1, cloud type code #1 and cloud base ht code #1
	cld_lyr_2_q	NUMBER(5)	QC code - layer cloud amount #2, cloud type code #2 and cloud base ht code #2
	cld_lyr_3_q	NUMBER(5)	QC code - layer cloud amount #3, cloud type code #3 and cloud base ht code #3
	cld_lyr_4_q	NUMBER(5)	QC code - layer cloud amount #4, cloud type code #4 and cloud base ht code #4
	vert_vsby_q	NUMBER(5)	QC code - vertical visibility
	max_gust_per	NUMBER(5)	Period of maximum gust relative to current time, in minutes e.g. -60 is last hour
	max_gust_spd	NUMBER(3)	Max gust is knots
	max_gust_spd_q	NUMBER(5)	QC code - max gust speed

Triggers

Trigger Name	marine_ob_ins
When Fired	Row level insert and update trigger fired before each row is inserted/updated

Description	Calls the function to calculate the longitude band code from the longitude and then insert it into column LONGITUDE_BAND_CODE

Trigger Name	marine_ob_upd
When Fired	Row level update trigger fired after updates
Description	The trigger serves two purposes. Firstly it maintains the primary keys on all of the child tables i.e. when a change is made to any of the primary key fields on MARINE_OB (ob_time, latitude, longitude, id, id_type) updates are performed on MAR_ACQ_LOG, MARINE_ICE_OB and MARINE_VOSCLIM_OB to maintain the referential integrity. Secondly if changes are made to any of the fields which can be QC'd then the old values from before the update are stored for use by the trigger MARINE_OB_UPD_V0. Details are only stored if the field(s) that has been updated was not previously set to NULL

Trigger Name	marine_ob_upd_v0
When Fired	Statement level update trigger fired after updates
Description	The trigger will retain the orginal version of the MARINE_OB record as a version ZERO row. If a version ZERO doesn't exist then insert one. If version ZERO exists but contains NULL values for met elements then update the version ZERO record with the non-NULL values. If a version ZERO exists and non-NULL values are present, then these will not be updated. This protects the original non-NULL values that were entered.

Trigger Name	marine_ob_rh_trgr
When Fired	Row level insert and update trigger fired before each row is inserted/updated
Description	Calls the function to calculate the relative humidity given the air temperature, pressure and wet bulb temperature. The result is then stored in the RLTV_HUM column

Trigger Name	marine_ob_acq
When Fired	Row level insert trigger fired after row is inserted
Description	Calls the function to insert a record into the MAR_ACQ_LOG

Trigger Name	marine_ob_del_adr
When Fired	Row level delete trigger fired after row is deleted
Description	Deletes v0 ob if the associated v1 ob is deleted. Will also delete associated v1 and v0 ob in the following tables: marine_ice_ob, marine_vosclim_ob and mar_acq_log. Also deletes associated v1

	and v0 in marine_ice_ob, marine_voscoim_ob and mar_acq_log if a v0 row is deleted, which does not have an associated v1 row.

Trigger Name	marine_ob_del_ad
When Fired	Table level delete trigger fired after row is deleted
Description	Deletes v0 ob if the associated v1 ob is deleted. Will also delete associated v1 and v0 ob in the following tables: marine_ice_ob, marine_vosclim_ob and mar_acq_log. Also deletes associated v1 and v0 in marine_ice_ob, marine_voscoim_ob and mar_acq_log if a v0 row is deleted, which does not have an associated v1 row.

TABLE	marine_vosclim_ob

Description: The midas.marine_vosclim_ob table contains VOSCLIM data to support IMMT2 and 3 collected from observations from ships. The observations are made during the hour ending at the stated date and time and are defined by position (latitude and longitude) and by time. Duplicates can exist at a specified position and time, e.g. when ships are alongside for bunkering, so the identifier of the ship is part of the primary key of the entity. Identifier is either call-sign or buoy-id, depending upon report type. Only about 1.5% of marine observations include VOSCLIM data, so the VOSCLIM attributes of a marine report are stored in a separate table because of their very low occurrence. Each marine_vosclim_ob row may contain 0, 1 or 2 corresponding entries in midas.marine_ob. A marine_vosclim_ob without a corresponding marine_ob will be very rare. Each marine_ob row may contain 0, 1 or 2 corresponding rows in marine_vosclim_ob. Most marine_ob rows will not have a corresponding entry in marine_vosclim_ob. The QC attributes have been included although this data cannot currently be QC'd via MIDS.

midasupd.marine_vosclim_ob is the corresponding updateable view, with one-for-one projection of the columns. midasvu.marine_vosclim_ob is a read-only view, with one-for-one projection of the columns.

*	ob_time	DATE	Date and time of observation
*	latitude	NUMBER(3,1)	0.1 deg
*	longitude	NUMBER(4,1)	0.1 deg
*	id	VARCHAR2(8)	
*	id_type	VARCHAR2(4)	Identifier type
*	vosclim_version_flag	CHAR(1)	Observation version number
	ship_heading	NUMBER(3)	Ships heading; the direction to which the bow is pointing
	ship_ground_course	NUMBER(3)	Ship ground course; the direction the vessel actually moves over the fixed earth
	ship_ground_speed	NUMBER(2)	The speed the vessel moves over the fixed earth
	cargo_deck_height	NUMBER(2)	Maximum height in meters of deck cargo above summer maximum
	ref_level_from_sea	NUMBER(2)	Departure of reference level from actual sea level

	rltv_wind_dir	NUMBER(3)	Relative wind direction in degrees off the bow
	rltv_wind_speed	VARCHAR2(2)	Relative wind speed
	ship_heading_q	NUMBER(5)	QC code – Ship heading
	ship_ground_course_q	NUMBER(5)	QC code – Ship ground course
	ship_ground_speed_q	NUMBER(5)	QC code – Ship ground speed
	cargo_deck_height_q	NUMBER(5)	QC code – Cargo deck height
	ref_level_from_sea_sign_q	NUMBER(5)	QC code – Ref level from sea, sign
	ref_level_from_sea_height_q	NUMBER(5)	QC code – Ref level from sea, height
	rltv_wind_dir_q	NUMBER(5)	QC code – Relative wind direction
	rltv_wind_speed_q	NUMBER(5)	QC code – Relative wind speed

Triggers

Trigger Name	marine_vosclim_upd
When Fired	Row level update trigger fired after each row is updated
Description	If changes are made to any of the fields which can be QC'd then the old values from before the update are stored for use by the trigger MARINE_VOSCLIM_UPD_V0. Details are only stored if the field(s) that has been updated was not previously set to NULL

Trigger Name	marine_vosclim_upd_v0
When Fired	Statement level update trigger fired after updates
Description	The trigger will retain the orginal version of the MARINE_VOSCLIM_OB record as a version ZERO row. If a version ZERO doesn't exist then insert one. If version ZERO exists but contains NULL values for met elements then update the version ZERO record with the non-NULL values. If a version ZERO exists and non-NULL values are present, then these will not be updated. This protects the original non-NULL values that were entered.

Trigger Name	marine_vosclim_del_adr
When Fired	Statement level delete trigger fired after deletes
Description	The trigger fires when a v1 row is deleted to delete the associated v0 row.

Trigger Name	marine_vosclim_del_ad
When Fired	Table level delete trigger fired after deletes

Description	The trigger fires when a v1 row is deleted to delete the associated v0 row.

9.4 MIDAS Tables and MIDASUPD Views- Upper Air

VIEW	ua_sounding

Description: The midas.ua_sounding table contains the non-repeating parameters of an upper air ascent, i.e. attributes common to the entire ascent. The ascent may originate from land stations or from ships, and may be a full TEMP or a PILOT. Every upper air ascent will have one and only one occurrence in the ua_sounding table and one or more rows in the ua_sounding_point table. The entity contains the latitude and longitude of an upper air ascent from ships. Retrieval by ship latitude or longitude is not required, but retrieval by call sign is, so identifier is indexed. Some queries require a long period of obs for a limited area, e.g. for radar duct investigation.

Upper air data is not given quality control by QEv Team, but the ua_sounding_point table contains four QC_code elements. The MIDAS data storage process sets these four attributes by converting the QC attributes present in the MetDB. These QC flags are originally set by the unified model, and are known as merge elements. The historic CDB data contained various flags, and these have been incorporated into the four QC_code elements of ua_sounding_point. The three valid combinations of id_type and met_domain_name are WMO/UAPLT, WMO/UATMP and SHIP/UATMPSHP.

The first year of data is 1948, for UK and a few overseas stations, with substantial increase in volumes from 1975. For WMO blocks 03, 04, some of 06 and a few others stations such as Gibraltar, we have a WMO commitment to store all of the data. For other WMO blocks, the requirement is to store 10 years worth of obs for selected stations. Data ingestion by monthly batch is deemed adequate. The upper air archive is maintained as a series of sequential tape datasets in MASS, using the SQL*Loader utility to populate the MIDAS tables with the desired data. The flat files are organised by period (e.g. 1960-1969) and by groups of stations.

The name of the MASS dataset is stored in abbreviated form in the db_seg_name attribute of the src_capability table. The contents of src_capability.db_seg_name must be prefixed with MSD5.MIDUPAIR.and yyyy must be interpreted to the last year of a 10-year period, using src_cap_end_date. Thus src_capability.db_seg_name = A03998.Yyyyy shows that the data are in MSD5.MIDUPAIR.A03998.Y1959, MSD5.MIDUPAIR.A03998.Y1969, etc., depending on the range of dates between src_cap_bgn_date and src_cap_end_date.

A procedure allows customers to restore data from the MASS archive, and optionally to load the data into MIDAS using the SQL*Loader utility. The SQL*Loader utility operates in Direct Path mode. midasupd.ua_sounding is the corresponding updateable with one-for-one projection of the columns. midasvu.ua_sounding is a read-only view.

See Section 12 for details of how to use this table.

PK	Attribute	Datatype	Description / Units / Precision
*	id	VARCHAR2(8)	
*	id_type	VARCHAR2(4)	UAPLT, UATMP, UATMPSHP
*	ob_time	DATE	Date and time of observation

	met_domain_name	VARCHAR2(8)	
	src_id	NUMBER(6)	
	rec_st_ind	NUMBER(4)	
	ua_asc_lnch_ctm	NUMBER(4)	Time of launch
	wind_shr_abv_max_wind	NUMBER(4)	Wind shear above max wind
	wind_shr_blw_max_wind	NUMBER(4)	Wind shear below max wind
	cld_base_amt_id	CHAR(1)	
	low_cld_type_id	CHAR(1)	
	cld_base_ht	NUMBER(4)	Cloud base height decametres
	med_cld_type_id	CHAR(1)	
	hi_cld_type_id	CHAR(1)	
	radar_type_id	CHAR(2)	
	sonde_type_id	CHAR(2)	
	latitude	NUMBER(3,1)	UATMPSHP only 0.1 deg
	longitude	NUMBER(4,1)	UATMPSHP only 0.1 deg

VIEW	ua_sounding_point

Description: The midas.ua_sounding_point table contains the repeating values of an upper air ascent, i.e. all the measurements at one level. All upper air ascents have one or more points. Date and time are part of the key, so too is the upper air level (i.e. elevation).

Upper air data is not given quality control by QEv Team, but the table contains four QC code elements. The MIDAS data storage process sets these four attributes by converting the QC attributes present in the MetDB. These QC flags are originally set by the unified model, and are known as merge elements. The historic CDB data contained various flags, and these have been incorporated into the four QC code elements.

Relative humidity and humidity mixing ratio are not stored. MIDAS provides packaged functions to calculate these values during retrieval. Refer to the description of ua_sounding for details of how the upper air archive is maintained in MASS.

midasupd.ua_sounding_point is the corresponding updateable view with one-for-one projection of the columns, and midasvu.ua_sounding_point is a read-only view.

See Section 12 for details of how to use this table.

PK	Attribute	Datatype	Description / Units / Precision
*	id	VARCHAR2(8)	
*	id_type	VARCHAR2(4)	UAPLT, UATMP, UATMPSHP
*	ob_time	DATE	Date and time of observation
*	ua_sndg_pt_num	NUMBER(3)	
	vert_sig_code	NUMBER(3)	2 = Significant level winds 4 = Significant level temperature 8 = Maximum wind 16 = Tropopause

			32 = Standard level 64 = Surface level
	ua_height	NUMBER(5)	Height metres
	pres_coord	NUMBER(5,1)	Air pressure co-ordinate 0.1 Hpa
	ua_air_temperature	NUMBER(3,1)	0.1 deg.C.
	ua_dewpoint	NUMBER(3,1)	0.1 deg.C.
	ua_wind_dir	NUMBER(3)	Degrees true
	ua_wind_speed	NUMBER(3)	Knots
	ua_height_qc_code	CHAR(1)	Quality of height element. The code is set based on BUFR table 055019 as follows: A = Accumulated value E = Estimated value, manually verified H = Homologous value, rigorously tested and manually approved I = Interpolated value, automatically generated, not verified, i.e. any of these 05519 flags set: 11, 15, 16. M = Missing Value N = Not tested, but within observed climatological boundaries, physical constraints, logical limits and current coding practices. Any of the following 05519 flags set: 10, 12, 17, 18, 19, meaning 17 set and 20 not set etc. R = Record-breaking value, verified and manually approved. 05519 flag 13 set. S = Suspect value outside climatological boundaries, no means to verify, (any of these 05519 flags set: 14, 20, 21, 22, 23, 24) T = Tested value, manually checked but not perfectly homologous. BUFR table 055019 is a 24-bit flag table that defines QC Flags for an upper air level. These flags are set by the NWP model, and are known as merge elements. The values of 05519 flags are: 3 Partial layer 4 Tropopause 5 Max wind 6 Significant wind 7 Significant temperature 8 Standard 9 Surface 10 Superadiabatic 11 Interpolation

			12 Hydrostatic 13 Extreme value 14 Inconsistent 15 Data correct (e.g. sign correction) 16 Perm correct (fixed correction) 7 Clim performed 18 Back performed (background check done) 19 Buddy performed 20 Clim reject 21 Perm reject (blacklisted) 22 Back reject (background check failed) 23 Buddy reject 24 Final reject (Do not use in analysis) MIDAS supplies supply values 4 - 9 via the vert_sig_code attribute.
	ua_air_temperature_qc_code	CHAR(1)	See ua_height_qc_code
	ua_relative_humidity_qc_code	CHAR(1)	See ua_height_qc_code MIDAS stores dewpoint not relative humidity. This code therefore implies the quality of the dewpoint (and of the derived relative humidity).
	ua_wind_qc_code	CHAR(1)	See ua_height_qc_code

9.5 MIDASVU Views

VIEW	capability_at_src

Description: This database view joins the source table with src_capability, to allow easy selection of src_capability by source attributes, e.g. to find all capabilities in a selected post code.

Attribute	Datatype	Description / Units / Precision
src_id	NUMBER(6)	
src_name	VARCHAR2(40)	
id_type	VARCHAR2(4)	
id	VARCHAR2(8)	
met_domain_name	VARCHAR2(8)	
src_cap_bgn_date	DATE	
src_cap_end_date	DATE	
prime_capability_flag	CHAR(1)	See midas.src_capability
rcpt_method_name	VARCHAR2(20)	
db_segment_name	VARCHAR2(12)	
data_retention_period	NUMBER(3)	
loc_geog_area_id	VARCHAR2(4)	
post_code	VARCHAR2(9)	
wmo_region_code	CHAR(1)	

high_prcn_lat	NUMBER(5,3)	
high_prcn_lon	NUMBER(6,3)	
grid_ref_type	VARCHAR2(4)	
east_grid_ref	NUMBER(6)	
north_grid_ref	NUMBER(7)	
elevation	NUMBER(4)	
hydr_area_id	NUMBER(4)	
drainage_stream_id	VARCHAR2(4)	
zone_time	NUMBER(2)	
src_bgn_date	DATE	
src_end_date	DATE	

VIEW	met_domain_element

Description: This database view joins the midas.met_domain table with the midas.met_element and midas.domain_element tables. It allows the user to easily identify all of the elements in a domain, or all of the domains where an element is used.

Attribute	Datatype	Description / Units / Precision
met_domain_name	VARCHAR2(8)	
met_dom_class	VARCHAR2(28)	
dom_usg_id	CHAR(1)	
met_domain_dsc	VARCHAR2(80)	
met_element_id	NUMBER(5)	
met_element_name	VARCHAR2(32)	
dom_elem_pos_num	NUMBER(4)	
str_met_dom_name	VARCHAR2(8)	
code_id	VARCHAR2(8)	
minimum_value	NUMBER(5)	
maximum_value	NUMBER(5)	
scale_factor	NUMBER(2)	
met_element_class	VARCHAR2(28)	

VIEW	mo_sfc_rec_elem

Description: This view projects all of the columns and all of the rows from mo_sfc_elem, joins with mo_sfc_rec to project id and id_type, and joins with met_element to project met_element_name. This allows selection by id, id_type and met_element_name. For more information please see the mo_sfc_elem and mo_sfc_rec table definitions in Section 9.2.

Attribute	Datatype	Description / Units / Precision
mo_sfc_elem_prtn_id	CHAR(2)	
src_id	NUMBER(6)	

id_type	VARCHAR2(4)	
id	VARCHAR2(8)	
ob_date	DATE	
met_domain_name	VARCHAR2(8)	e.g. 'CLM71-11', 'CARLOS'
met_element_id	NUMBER(5)	
met_element_name	VARCHAR2(32)	
version_num	NUMBER(1)	
met_elem_day_cnt	NUMBER(2)	Count of days when the element occurred
met_elem_hour_cnt	NUMBER(3)	Count of hours when the element occurred
met_elem_min_val	NUMBER(6,2)	Minimum value of the element
met_elem_min_val_day_num	NUMBER(2)	Day on which the min value occurred
met_elem_mean_val	NUMBER(6,2)	Mean value of the element
met_elem_sd_val	NUMBER(6,2)	Standard deviation
met_elem_max_val	NUMBER(6,2)	Maximum value of the element
met_elem_max_val_day_num	NUMBER(2)	Day on which the max value occurred
met_elem_miss_day_cnt	NUMBER(2)	Count of days when the element was missing
met_elem_miss_hour_cnt	NUMBER(3)	Count of hours when the element was missing
met_elem_amt	NUMBER(6,2)	e.g. Rainfall amount
met_elem_occr_prc	NUMBER(3)	Percent of occurrences, e.g. NE wind
met_elem_qual_id	CHAR(1)	Quality of the element
met_elem_sbst_day_cnt	NUMBER(2)	Count of days when substitute values used.
met_elem_sbst_code	CHAR(2)	Defines substitute values
wind_speed_unit_id	CHAR(1)	
temp_read_type_id	CHAR(1)	
max_temp_read_hour_num	NUMBER(2)	
min_temp_read_hour_num	NUMBER(2)	
qual_analyst_id	VARCHAR2(3)	Identifier of QC staff member

VIEW	rain_ob

Description: This database view is a UNION of rain_drnl_ob with rain_hrly_ob. It allows the user to select daily and/or hourly rainfall, regardless of which table the data are stored in.

Attribute	**Datatype**	**Description / Units / Precision**
ob_date	VARCHAR2(10)	
ob_end_ctime	NUMBER(4)	
ob_hour_count	NUMBER(3)	
id_type	VARCHAR2(4)	
id	VARCHAR2(8)	
met_domain_name	VARCHAR2(8)	
version_num	NUMBER(1)	

src_id	NUMBER(6)	
prcp_amt	NUMBER(5,1)	
prcp_amt_q	NUMBER(5)	
prcp_amt_j	CHAR(1)	
prcp_dur	NUMBER(4)	
prcp_dur_q	NUMBER(5)	
ob_day_cnt_q	NUMBER(5)	
rain_date	VARCHAR2(10)	
table_name	CHAR(12)	
meto_stmp_time	DATE	
midas_stmp_etime	NUMBER(6)	
rec_st_ind	NUMBER(4)	

VIEW	src_drnl_rain

Description: This view joins rain_drnl_ob with selected attributes from source and src_capability. It checks that the rain_drnl_ob record is consistent with its src_capability - if it is not, the view will not return the row. The view allows users to select from rain_drnl_ob using attributes of the source, e.g. src_name or post_code. It also projects rain_date, which is the observation time 'thrown' to the appropriate rainfall day - however, we recommend that users should use the midas.midas_utility_pkg.throw_ob_date_fnc function in preference to this attribute, because the function provides a more flexible method of throwing the date.

Attribute	Datatype	Description / Units / Precision
src_name	VARCHAR2(40)	
loc_geog_area_id	VARCHAR2(4)	
post_code	VARCHAR2(9)	
grid_ref_type	VARCHAR2(4)	
east_grid_ref	NUMBER(6)	
north_grid_ref	NUMBER(7)	
elevation	NUMBER(4)	
high_prcn_lat	NUMBER(5,3)	
high_prcn_lon	NUMBER(6,3)	
hydr_area_id	NUMBER(4)	
drainage_stream_id	VARCHAR2(4)	
src_id	NUMBER(6)	
id_type	VARCHAR2(4)	
id	VARCHAR2(8)	
met_domain_name	VARCHAR2(8)	
prime_capability_flag	CHAR(1)	See midas.src_capability
ob_date	DATE	
ob_end_ctime	NUMBER(4)	
ob_day_cnt	NUMBER(3)	
ob_day_cnt_q	NUMBER(5)	
version_num	NUMBER(1)	
prcp_amt	NUMBER(5,1)	

prcp_amt_q	NUMBER(5)	
prcp_amt_j	CHAR(1)	
meto_stmp_time	DATE	
midas_stmp_etime	NUMBER(6)	
rain_date	VARCHAR2(10)	ob_date thrown to the appropriate rainfall day.

VIEW	src_drnl_temp

Description: This view joins temp_drnl_ob with selected attributes from source and src_capability. It checks that the temp_drnl_ob record is consistent with its src_capability - if it is not, the view will not return the row. The view allows users to select from temp_drnl_ob using attributes of the source, e.g. src_name or post_code. It also projects ob_group_date_1 and ob_group_date_2, which are the observation time 'thrown' to the day of maximum temperature and minimum temperature - however, we recommend that users should use the midas.midas_utility_pkg.throw_ob_date_fnc function in preference to these attributes, because the function provides a more flexible method of throwing these dates.

Attribute	Datatype	Description / Units / Precision
src_name	VARCHAR2(40)	
loc_geog_area_id	VARCHAR2(4)	
post_code	VARCHAR2(9)	
grid_ref_type	VARCHAR2(4)	
east_grid_ref	NUMBER(6)	
north_grid_ref	NUMBER(7)	
elevation	NUMBER(4)	
high_prcn_lat	NUMBER(5,3)	
high_prcn_lon	NUMBER(6,3)	
src_id	NUMBER(6)	
id_type	VARCHAR2(4)	
id	VARCHAR2(8)	
met_domain_name	VARCHAR2(8)	
prime_capability_flag	CHAR(1)	See midas.src_capability
ob_end_time	DATE	
ob_hour_count	NUMBER(3)	
version_num	NUMBER(1)	
max_air_temp	NUMBER(3,1)	
min_air_temp	NUMBER(3,1)	
min_grss_temp	NUMBER(3,1)	
min_conc_temp	NUMBER(3,1)	
max_air_temp_q	NUMBER(5)	
min_air_temp_q	NUMBER(5)	
min_grss_temp_q	NUMBER(5)	
min_conc_temp_q	NUMBER(5)	
max_air_temp_j	CHAR(1)	
min_air_temp_j	CHAR(1)	

min_grss_temp_j	CHAR(1)	
min_conc_temp_j	CHAR(1)	
meto_stmp_time	DATE	
midas_stmp_etime	NUMBER(6)	
ob_group_date_1	VARCHAR2(10)	ob_end_time thrown to day of maximum temperature.
ob_group_date_2	VARCHAR2(10)	ob_end_time thrown to day of minimum temperature.
rec_st_ind	NUMBER(4)	State indicator of the temp_drnl_ob record

---

VIEW	src_drnl_weather

Description: This view joins weather_drnl_ob with selected attributes from source and src_capability. It checks that the weather_drnl_ob record is consistent with its src_capability - if it is not, the view will not return the row. The view allows users to select from weather_drnl_ob using attributes of the source, e.g. src_name or post_code.

Attribute	Datatype	Description / Units / Precision
src_name	VARCHAR2(40)	
loc_geog_area_id	VARCHAR2(4)	
post_code	VARCHAR2(9)	
high_prcn_lat	NUMBER(5,3)	
high_prcn_lon	NUMBER(6,3)	
elevation	NUMBER(4)	
grid_ref_type	VARCHAR2(4)	
east_grid_ref	NUMBER(6)	
north_grid_ref	NUMBER(7)	
src_id	NUMBER(6)	
id_type	VARCHAR2(4)	
id	VARCHAR2(8)	
met_domain_name	VARCHAR2(8)	
prime_capability_flag	CHAR(1)	See midas.src_capability
ob_end_time	DATE	
ob_hour_count	NUMBER(3)	
version_num	NUMBER(1)	
cs_24hr_sun_dur	NUMBER(3,1)	
cs_24hr_sun_dur_q	NUMBER(5)	
wmo_24hr_sun_dur	NUMBER(3,1)	
wmo_24hr_sun_dur_q	NUMBER(5)	
conc_state_id	CHAR(1)	
conc_state_id_q	NUMBER(5)	
lying_snow_flag	CHAR(1)	
snow_depth	NUMBER(4)	
snow_depth_q	NUMBER(5)	

frsh_snw_amt	NUMBER(4)	
frsh_snw_amt_q	NUMBER(5)	
snow_day_id	CHAR(1)	
snow_day_id_q	NUMBER(5)	
hail_day_id	CHAR(1)	
hail_day_id_q	NUMBER(5)	
thunder_day_flag	CHAR(1)	
thunder_day_flag_q	NUMBER(5)	
gale_day_flag	CHAR(1)	
gale_day_flag_q	NUMBER(5)	
frsh_mnt_snwfall_flag	CHAR(1)	
meto_stmp_time	DATE	
midas_stmp_etime	NUMBER(6)	

VIEW	src_glbl_wx

Description: This view joins glbl_wx_ob with selected attributes from source and src_capability. It checks that the glbl_wx_ob record is consistent with its src_capability - if it is not, the view will not return the row. The view allows users to select from glbl_wx_ob using attributes of the source, e.g. src_name. Always specify wmo_region_code ('1' - '7') if possible, because it is the partition key of glbl_wx_ob, and will significantly improve the response time of the query.

Attribute	Datatype	Description / Units / Precision
src_name	VARCHAR2(40)	
loc_geog_area_id	VARCHAR2(4)	
wmo_region_code	CHAR(1)	
high_prcn_lat	NUMBER(5,3)	
high_prcn_lon	NUMBER(6,3)	
elevation	NUMBER(4)	
zone_time	NUMBER(2)	
src_id	NUMBER(6)	
id_type	VARCHAR2(4)	
id	VARCHAR2(8)	
met_domain_name	VARCHAR2(8)	
ob_time	DATE	
wind_direction	NUMBER(3)	
wind_speed	NUMBER(3)	
prst_wx_id	CHAR(2)	
past_wx_id_1	CHAR(1)	
past_wx_id_2	CHAR(1)	
cld_ttl_amt_id	CHAR(1)	
cld_base_amt_id	CHAR(1)	
cld_base_ht	NUMBER(4)	
low_cld_type_id	CHAR(1)	
med_cld_type_id	CHAR(1)	

hi_cld_type_id	CHAR(1)	
wind_speed_unit_id	CHAR(1)	
visibility	NUMBER(4)	
air_temperature	NUMBER(3,1)	
wetb_temp	NUMBER(3,1)	
dewpoint	NUMBER(3,1)	
stn_pres	NUMBER(5,1)	
pres_sfc	NUMBER(5,1)	
pres_sfc_ht	NUMBER(5)	
pres_tdcy_amt	NUMBER(4,1)	
prcp_ob_hr_cnt	NUMBER(3)	
prcp_amt	NUMBER(5,1)	
cld_amt_id_1	CHAR(1)	
cloud_type_id_1	CHAR(1)	
cld_base_ht_id_1	NUMBER(4)	
cld_amt_id_2	CHAR(1)	
cloud_type_id_2	CHAR(1)	
cld_base_ht_id_2	NUMBER(4)	
cld_amt_id_3	CHAR(1)	
cloud_type_id_3	CHAR(1)	
cld_base_ht_id_3	NUMBER(4)	
max_air_temp	NUMBER(3,1)	
min_air_temp	NUMBER(3,1)	
min_grss_temp	NUMBER(3,1)	
ground_state_id	CHAR(2)	
snow_depth	NUMBER(4)	
sun_ob_hr_cnt	NUMBER(3)	
sun_dur	NUMBER(3,1)	
q24hr_prcp_amt	NUMBER(5,1)	
q24hr_pres_tdcy_amt	NUMBER(4,1)	
vert_vsby	NUMBER(3)	
src_opr_type	CHAR(1)	
gust_spd_type_code	CHAR(1)	
max_gust_speed	NUMBER(3)	
runway_name	VARCHAR2(4)	
rnwy_vis_rnge	NUMBER(4)	
alt_pres	NUMBER(4)	
qc_flag_list_1	NUMBER(5)	
qc_flag_list_2	NUMBER(5)	

VIEW	src_hrly_rain

Description: This view joins rain_hrly_ob with selected attributes from source and src_capability. It checks that the rain_hrly_ob record is consistent with its src_capability - if it is not, the view will not return the row. The view

allows users to select from rain_hrly_ob using attributes of the source, e.g. src_name or post_code. It also projects rain_date, which is the observation time 'thrown' to the appropriate rainfall day.

Attribute	Datatype	Description / Units / Precision
src_name	VARCHAR2(40)	
loc_geog_area_id	VARCHAR2(4)	
post_code	VARCHAR2(9)	
grid_ref_type	VARCHAR2(4)	
east_grid_ref	NUMBER(6)	
north_grid_ref	NUMBER(7)	
elevation	NUMBER(4)	
high_prcn_lat	NUMBER(5,3)	
high_prcn_lon	NUMBER(6,3)	
hydr_area_id	NUMBER(4)	
drainage_stream_id	VARCHAR2(4)	
src_id	NUMBER(6)	
id	VARCHAR2(8)	
id_type	VARCHAR2(4)	
met_domain_name	VARCHAR2(8)	
prime_capability_flag	CHAR(1)	See midas.src_capability
ob_end_time	DATE	
ob_hour_count	NUMBER(3)	
version_num	NUMBER(1)	
prcp_amt	NUMBER(5,1)	
prcp_amt_q	NUMBER(5)	
prcp_amt_j	CHAR(1)	
prcp_dur	NUMBER(4)	
prcp_dur_q	NUMBER(5)	
meto_stmp_time	DATE	
midas_stmp_etime	NUMBER(6)	
rain_date	VARCHAR2(10)	ob_end_time thrown to the appropriate rainfall day

---

VIEW	src_hrly_weather

Description: This view joins weather_hrly_ob with selected attributes from source and src_capability. It checks that the weather_hrly_ob record is consistent with its src_capability - if it is not, the view will not return the row. The view allows users to select from weather_hrly_ob using attributes of the source, e.g. src_name or post_code.

Attribute	Datatype	Description / Units / Precision
src_name	VARCHAR2(40)	
loc_geog_area_id	VARCHAR2(4)	
post_code	VARCHAR2(9)	
grid_ref_type	VARCHAR2(4)	

east_grid_ref	NUMBER(6)	
north_grid_ref	NUMBER(7)	
elevatioN	NUMBER(4)	
high_prcn_lat	NUMBER(5,3)	
high_prcn_lon	NUMBER(6,3)	
src_id	NUMBER(6)	
id_type	VARCHAR2(4)	
id	VARCHAR2(8)	
met_domain_name	VARCHAR2(8)	
prime_capability_flag	CHAR(1)	See midas.src_capability
ob_time	DATE	
version_num	NUMBER(1)	
wind_speed_unit_id	CHAR(1)	
src_opr_type	CHAR(1)	
wind_direction	NUMBER(3)	
wind_speed	NUMBER(3)	
prst_wx_id	CHAR(2)	
past_wx_id_1	CHAR(1)	
past_wx_id_2	CHAR(1)	
cld_ttl_amt_id	CHAR(1)	
low_cld_type_id	CHAR(1)	
med_cld_type_id	CHAR(1)	
hi_cld_type_id	CHAR(1)	
cld_base_amt_id	CHAR(1)	
cld_base_ht	NUMBER(4)	
visibility	NUMBER(4)	
vert_vsby	NUMBER(3)	
cld_amt_id_1	CHAR(1)	
cloud_type_id_1	CHAR(1)	
cld_base_ht_id_1	NUMBER(4)	
cld_amt_id_2	CHAR(1)	
cloud_type_id_2	CHAR(1)	
cld_base_ht_id_2	NUMBER(4)	
cld_amt_id_3	CHAR(1)	
cloud_type_id_3	CHAR(1)	
cld_base_ht_id_3	NUMBER(4)	
cld_amt_id_4	CHAR(1)	
cloud_type_id_4	CHAR(1)	
cld_base_ht_id_4	NUMBER(4)	
air_temperature	NUMBER(3,1)	
dewpoint	NUMBER(3,1)	
wetb_temp	NUMBER(3,1)	
msl_pressure	NUMBER(5,1)	
stn_pres	NUMBER(5,1)	
alt_pres	NUMBER(4)	
ground_state_id	CHAR(2)	

q10mnt_mxgst_spd	NUMBER(3)	
cavok_flag	CHAR(1)	
cs_hr_sun_dur	NUMBER(3,1)	
wmo_hr_sun_dur	NUMBER(3,1)	
wind_direction_q	NUMBER(5)	
wind_speed_q	NUMBER(5)	
prst_wx_id_q	NUMBER(5)	
past_wx_id_1_q	NUMBER(5)	
past_wx_id_2_q	NUMBER(5)	
cld_ttl_amt_id_q	NUMBER(5)	
low_cld_type_id_q	NUMBER(5)	
med_cld_type_id_q	NUMBER(5)	
hi_cld_type_id_q	NUMBER(5)	
cld_base_amt_id_q	NUMBER(5)	
cld_base_ht_q	NUMBER(5)	
visibility_q	NUMBER(5)	
vert_vsby_q	NUMBER(5)	
air_temperature_q	NUMBER(5)	
dewpoint_q	NUMBER(5)	
wetb_temp_q	NUMBER(5)	
ground_state_id_q	NUMBER(5)	
cld_amt_id_1_q	NUMBER(5)	
cloud_type_id_1_q	NUMBER(5)	
cld_base_ht_id_1_q	NUMBER(5)	
cld_amt_id_2_q	NUMBER(5)	
cloud_type_id_2_q	NUMBER(5)	
cld_base_ht_id_2_q	NUMBER(5)	
cld_amt_id_3_q	NUMBER(5)	
cloud_type_id_3_q	NUMBER(5)	
cld_base_ht_id_3_q	NUMBER(5)	
cld_amt_id_4_q	NUMBER(5)	
cloud_type_id_4_q	NUMBER(5)	
cld_base_ht_id_4_q	NUMBER(5)	
msl_pressure_q	NUMBER(5)	
stn_pres_q	NUMBER(5)	
alt_pres_q	NUMBER(5)	
q10mnt_mxgst_spd_q	NUMBER(5)	
cs_hr_sun_dur_q	NUMBER(5)	
wmo_hr_sun_dur_q	NUMBER(5)	
wind_direction_j	CHAR(1)	
wind_speed_j	CHAR(1)	
prst_wx_id_j	CHAR(1)	
past_wx_id_1_j	CHAR(1)	
past_wx_id_2_j	CHAR(1)	
cld_amt_id_j	CHAR(1)	
cld_ht_j	CHAR(1)	

visibility_j	CHAR(1)	
vert_vsby_j	CHAR(1)	
air_temperature_j	CHAR(1)	
dewpoint_j	CHAR(1)	
wetb_temp_j	CHAR(1)	
msl_pressure_j	CHAR(1)	
stn_pres_j	CHAR(1)	
alt_pres_j	CHAR(1)	
q10mnt_mxgst_spd_j	CHAR(1)	
meto_stmp_time	DATE	
midas_stmp_etime	NUMBER(6)	
rec_st_ind	NUMBER(4)	State indicator

VIEW	src_mean_wind

Description: This view joins wind_mean_ob with selected attributes from source and src_capability. It checks that the wind_mean_ob record is consistent with its src_capability - if it is not, the view will not return the row. The view allows users to select from wind_mean_ob using attributes of the source, e.g. src_name or post_code. It also projects wind_group_date, which throws the midnight observation back to the previous day.

Attribute	Datatype	Description / Units / Precision
src_name	VARCHAR2(40)	
loc_geog_area_id	VARCHAR2(4)	
post_code	VARCHAR2(9)	
grid_ref_type	VARCHAR2(4)	
east_grid_ref	NUMBER(6)	
north_grid_ref	NUMBER(7)	
elevation	NUMBER(4)	
high_prcn_lat	NUMBER(5,3)	
high_prcn_lon	NUMBER(6,3)	
src_id	NUMBER(6)	
id_type	VARCHAR2(4)	
id	VARCHAR2(8)	
met_domain_name	VARCHAR2(8)	
prime_capability_flag	CHAR(1)	See midas.src_capability
ob_end_time	DATE	
ob_hour_count	NUMBER(3)	
version_num	NUMBER(1)	
mean_wind_dir	NUMBER(3)	
mean_wind_dir_q	NUMBER(5)	
mean_wind_dir_j	CHAR(1)	
mean_wind_speed	NUMBER(4,1)	
mean_wind_speed_q	NUMBER(5)	
mean_wind_speed_j	CHAR(1)	
max_gust_dir	NUMBER(3)	

max_gust_dir_q	NUMBER(5)	
max_gust_dir_j	CHAR(1)	
max_gust_speed	NUMBER(3)	
max_gust_speed_q	NUMBER(5)	
max_gust_speed_j	CHAR(1)	
max_gust_ctime	NUMBER(4)	
max_gust_ctime_q	NUMBER(5)	
meto_stmp_time	DATE	
midas_stmp_etime	NUMBER(6)	
wind_group_date	VARCHAR2(10)	Throws the midnight ob back to the previous day

---

VIEW	src_radiation

Description: This view joins radt_ob with selected attributes from source and src_capability. It checks that the radt_ob record is consistent with its src_capability - if it is not, the view will not return the row. The view allows users to select from radt_ob using attributes of the source, e.g. src_name or post_code.

Attribute	**Datatype**	**Description / Units / Precision**
src_name	VARCHAR2(40)	
loc_geog_area_id	VARCHAR2(4)	
post_code	VARCHAR2(9)	
grid_ref_type	VARCHAR2(4)	
east_grid_ref	NUMBER(6)	
north_grid_ref	NUMBER(7)	
elevation	NUMBER(4)	
high_prcn_lat	NUMBER(5,3)	
high_prcn_lon	NUMBER(6,3)	
src_id	NUMBER(6)	
id_type	VARCHAR2(4)	
id	VARCHAR2(8)	
met_domain_name	VARCHAR2(8)	
prime_capability_flag	CHAR(1)	See midas.src_capability
ob_end_time	DATE	
ob_hour_count	NUMBER(3)	
version_num	NUMBER(1)	
glbl_irad_amt	NUMBER(4)	
glbl_irad_amt_q	NUMBER(5)	
difu_irad_amt	NUMBER(4)	
difu_irad_amt_q	NUMBER(5)	
direct_irad	NUMBER(4)	
direct_irad_q	NUMBER(5)	
irad_bal_amt	NUMBER(4)	
irad_bal_amt_q	NUMBER(5)	
glbl_s_lat_irad_amt	NUMBER(4)	

glbl_s_lat_irad_amt_q	NUMBER(5)	
glbl_horz_ilmn	NUMBER(4)	
glbl_horz_ilmn_q	NUMBER(5)	
meto_stmp_time	DATE	
midas_stmp_etime	NUMBER(6)	

VIEW	src_soil_temp

Description: This view joins soil_temp_ob with selected attributes from source and src_capability. It checks that the soil_temp_ob record is consistent with its src_capability - if it is not, the view will not return the row. The view allows users to select from soil_temp_ob using attributes of the source, e.g. src_name or post_code.

Attribute	Datatype	Description / Units / Precision
src_name	VARCHAR2(40)	
loc_geog_area_id	VARCHAR2(4)	
post_code	VARCHAR2(9)	
grid_ref_type	VARCHAR2(4)	
east_grid_ref	NUMBER(6)	
north_grid_ref	NUMBER(7)	
elevation	NUMBER(4)	
high_prcn_lat	NUMBER(5,3)	
high_prcn_lon	NUMBER(6,3)	
src_id	NUMBER(6)	
id_type	VARCHAR2(4)	
id	VARCHAR2(8)	
met_domain_name	VARCHAR2(8)	
prime_capability_flag	CHAR(1)	See midas.src_capability
ob_time	DATE	
version_num	NUMBER(1)	
q5cm_soil_temp	NUMBER(3,1)	
q5cm_soil_temp_q	NUMBER(5)	
q5cm_soil_temp_j	CHAR(1)	
q10cm_soil_temp	NUMBER(3,1)	
q10cm_soil_temp_q	NUMBER(5)	
q10cm_soil_temp_j	CHAR(1)	
q20cm_soil_temp	NUMBER(3,1)	
q20cm_soil_temp_q	NUMBER(5)	
q20cm_soil_temp_j	CHAR(1)	
q30cm_soil_temp	NUMBER(3,1)	
q30cm_soil_temp_q	NUMBER(5)	
q30cm_soil_temp_j	CHAR(1)	
q50cm_soil_temp	NUMBER(3,1)	
q50cm_soil_temp_q	NUMBER(5)	
q50cm_soil_temp_j	CHAR(1)	

q100cm_soil_temp	NUMBER(3,1)	
q100cm_soil_temp_q	NUMBER(5)	
q100cm_soil_temp_j	CHAR(1)	
meto_stmp_time	DATE	
midas_stmp_etime	NUMBER(6)	

10 Entity Keys and Table Indexes

Entity Keys
Users do not need to have a detailed knowledge of the indexes in the MIDAS database. However, an overview of the indexes available to the database Query Optimiser may help users to construct their queries in such a way that they are most likely to achieve the desired results with maximum efficiency.

Key:

table_name	index_name	column_name

---

acquisitions_log	acquisition_log_pk	ob_end_time met_domain_name table_name id_type id ob_hour_count version_num midas_acq_btch_num
	ix_midas_batch	table_name met_domain_name midas_acq_btch_num
	ix_midas_batch2	table_name met_domain_name midas_acq_btch_num

---

british_summer_time	c_bst_pk	begin_time

---

clm_src_capability	See src_capability	

---

clm_ua_norm	pk2638clm_ua_norm	src_id norm_first_year norm_last_year norm_month
	ix_clmurn_si	src_id

---

clm_ua_norm_lvl	pk2643clm_ua_norm_lvl	src_id norm_first_year norm_last_year norm_month pres_coord

clm_ua_rec	c_clmumr_ob_id_u	ua_mo_ob_id
	pk2639clm_ua_rec	src_id ob_date version_flag
	ix_clmumr_si	src_id

clm_ua_rec_lvl	pk2642clm_ua_rec_l	src_id ob_date pres_coord version_flag

code	c_code_pk	code_id

code_detail	c_coddet_pk	code_id code_value desc_line_number

cross_reference	c_xref_pk	ref_from_src_id ref_to_src_id association_type cross_ref_end_date
	ix_association	association_type
	ix_ref_to	ref_to_src_id
	ix_xref_domain	met_domain_name

domain_element	c_domelm_pk	met_domain_name met_element_id
	ix_str_dom_elem	str_met_dom_name met_element_id dom_elem_pos_num

geog_area_hier	c_geoghi_pk	wthn_geog_area_id cntn_geog_area_id
	ix_geog_contains	cntn_geog_area_id

geographic_area	c_geogar_pk	geog_area_id
	ix_geog_area	geog_area_name
	ix_geog_area_typ	geog_area_type

glbl_wx_ob	c_gwob_pk	id id_type ob_time wmo_region_code

hydrometric_area	c_hydrar_pk	hydr_area_id

identifier_type	c_identy_pk	id_type

mar_acq_log	mar_acq_pk	ob_time latitude longitude id id_type version_flag
	mar_acq_log_btch	mar_acq_log_btch
	mar_acq_log_idx	met_domain_name midas_acq_btch_num

marine_climatology	pk2635marine_climatology	month_number latitude longitude

marine_current	marcur_pk	ob_month latitude longitude id id_type

	marcur_srccap_fk	id_type id met_domain_name

marine_ob	c_marob_pk	ob_time latitude longitude id id_type version_flag longitude_band_code
	ix_marob_id	id id_type ob_time version_flag
	ix_batch_time	batch_stamp_time (meto_stmp_time)

met_domain	c_metdom_pk	met_domain_name

met_domain_class	c_domcls_pk	met_dom_class

met_elem_class	c_elmcls_pk	met_element_class

met_element	c_metelm_name	met_element_name
	c_metelm_pk	met_element_id
	met_element_class	met_element_class

mo_sfc_elem	mo_sfc_elem_pk	mo_sfc_elem_prtn_i met_element_id ob_date src_id met_domain_name version_num
	mo_sfc_elem_mo_sfc_rec_fk	src_id ob_date met_domain_name
	mo_sfc_elem_dom_elem_fk	met_domain_name met_element_id

mo_sfc_rec	pkmo_sfc_rec	src_id ob_date met_domain_name

radt_ob	c_radtob_pk	id id_type ob_end_time ob_hour_count version_num met_domain_name
	c_radtob_si	src_id

radt_ob_v2	radt_ob_v2_pk	id id_type ob_end_time ob_hour_count version_num met_domain_name
	radt_ob_v2_src_id_ix	src_id
	radt_ob_v2_src_met_dom_fk	src_id met_domain_name
	radt_ob_v2_upd_trgr_key	glbl_irad_amt difu_irad_amt difu_irad irad_bal_amt glbl_horz_ilmn

rain_drnl_ob	c_rndyob_pk	id id_type ob_date version_num met_domain_name
	ix_rndyob_src_id	src_id

rain_hrly_ob	c_rnhrob_pk	id id_type ob_end_time version_num met_domain_name ob_hour_count
	ix_rnhrob_si	src_id

rain_subhrly_ob	c_rnshob_pk	ob_time id_type id met_domain_name
	ix_rnshob_si	src_id

report_table_mapping	c_domtbl_pk	stored_id_type report_met_domain_name report_id_type table_name

runway_ob	c_rnwyob_pk	ob_time id id_type runway_name met_domain_name
	ix_rnwyob_si	src_id

sfc_clmo_elem	sfc_clmo_elem_pk	src_id clmo_end_date clmo_year_cnt met_domain_name month_number met_element_id
	sfc_clmo_elem_dom_elem_fk	met_domain_name met_element_id
	sfc_clmo_elem_sfc_clmo_rec_fk	src_id clmo_end_date clmo_year_cnt met_domain_name

sfc_clmo_rec	sfc_clmo_rec_pk	src_id clmo_end_date clmo_year_cnt met_domain_name
	sfc_clmo_rec_src_met_domain_fk	src_id met_domain_name

soil_min_temp_ob	txbgob_pk	id id_type ob_date met_domain_name

	txbgob_srccap_fk	id_type id met_domain_name
	txgob_src_id_ix	src_id

soil_temp_ob	c_soilob_pk	id id_type ob_time met_domain_name version_num
	ix_soilob_si	src_id

source	c_src_pk	src_id
	ix_post_code	post_code
	ix_src_hydro	hydr_area_id
	ix_src_loc_geo	loc_geog_area_id
	ix_src_name	src_name

src_capability	c_srccap_end_date	id id_type met_domain_name src_cap_end_date
	c_srccap_pk	id id_type met_domain_name src_cap_bgn_date
	ix_src_cap_dom	met_domain_name
	ix_src_cap_si	src_id

src_met_domain	src_met_domain_pk	src_id met_domain_name

src_remark	c_srcrem_pk	src_id src_rmrk_num

src_runway	c_runway_pk	src_id runway_name

synthetic_glbl_wx	synthetic_glbl_wx_pk	src_id time wmo_region_code
	synthetic_glbl_wx_prtn_key	wmo_region_code time
	synthetic_glbl_wx_src_id_fk	src_id

temp_drnl_ob	c_tdyob_pk	id id_type ob_end_time ob_hour_count version_num met_domain_name
	ix_tdyob_src_id	src_id

ua_sounding	c_ua_sndg_pk	id id_type ob_time

ua_sounding_point	c_ua_sndg_pt_pk	id id_type ob_time ua_sndg_pt_num

weather_drnl_ob	c_wxdyob_pk	id id_type ob_end_time ob_hour_count version_num met_domain_name
	ix_wxdyob_si	src_id

weather_hrly_ob	c_wxhrob_pk	id id_type ob_time version_num met_domain_name
	ix_wxhrob_src_id	src_id

wind_mean_ob	c_wndmob_pk	id id_type ob_end_time version_num ob_hour_count
	ix_wndmob_si	src_id
	ix_wndmob_time	ob_end_time

11 MIDAS Packaged Functions

MIDAS includes a number of database packages. These packages contain a number of functions to decode database values or calculate derived values. These functions simplify retrieval for end-users and also centralise and control the functional logic. The packaged functions are available in the midas schema and are called using the *schema.package.function* notation. The package details are given below with examples of use. If you require more information about a packaged function then please contact a member of the midas team.

11.1.1 midas_utility_pkg

decode_j_fnc
decode_q_fnc
decode_to_decamatres_fnc
decode_wx_code_fnc
feet_to_decametres_fnc
get_code_val_dsc_id_fnc
get_code_val_dsc_name_fnc
high_date_fnc
kjoule_to_watt_hour_fnc
knots_to_beaufort_fnc
lon_band_fnc
low_date_fnc
rank_met_domain_fnc
throw_ob_date_fnc

11.1.2 midas_humidity_pkg

calc_dwpt_fnc
calc_hmr_fnc
calc_rh_fnc
calc_svp_fnc
calc_ua_hmr_fnc
calc_ua_rh_fnc
calc_vp_fnc
calc_wbt_fnc

* MIDAS also contains a number of database triggers and packaged procedures. These are considered beyond the scope of this document, but details are available on request from the MIDAS Team.

FUNCTION	decode_j_fnc

Purpose: Decodes a J descriptor e.g. sea_temperature_j. If the _j value is not matched the function will return NULL.

Parameters: p_j_code_val VARCHAR2 - The value to be decoded.
p_j_code_name VARCHAR2 - The code name for the type of column to be decoded. (See section 14.5.3)

Returns: The code value description of the descriptor's value from the code_detail table.

Description: The function accepts the code value and code name for the descriptor and returns the description as VARCHAR2

Example:

```
SELECT wob.id
,      TO_CHAR(wob.ob_time,'YYYY-MM-DD HH24:MI') ob_time
,      wob.wetb_temp wetb
,      wob.wetb_temp_j wetb_j
,      midas.midas_utility_pkg.decode_j_fnc
       (
       wob.wetb_temp_j,
       'TEMPERATURE'
       ) j_dsc
FROM   midas.weather_hrly_ob wob
WHERE  wob.id_type = 'WMO'
AND    wob.id LIKE '0300%'
AND    wob.met_domain_name = 'SYNOP'
AND    wob.ob_time BETWEEN TO_DATE('2000-01-20 09:00',
                           'YYYY-MM-DD HH24:MI')
                   AND     TO_DATE('2000-01-22 10:00',
                           'YYYY-MM-DD HH24:MI');
```

FUNCTION	decode_q_fnc

Purpose : Gives the description of each value in a QC attribute e.g. dewpoint_q.

Parameters: p_q_code_val VARCHAR2 - The value to be decoded.
p_q_code_name VARCHAR2 - The code name for the type of column to be decoded. (See section 14.5.3)

Returns: The QC item description for the QC value taken from the code_detail table.

Description: _q columns contain a 5-digit number, and each digit represents a different QC item. From the left, they are: Marker, Estimate, Suspect, Query, and Level. The paramters for the function are the column value to be decoded e.g. wetb_temp_q and the item to be decoded; this can be specified using any of the following:
'1', 'M', 'Marker'
'2', 'E', 'Estimate'
'3', 'S', 'Suspect'
'4', 'Q', 'Query'
'5', 'L', 'Level'
The function returns the description of the specified QC item as VARCHAR2. If the _q value has been inadvertently stored as -32768, then the function will return NULL. Invalid values will return the character equivalent of the value.

Example:

```
SELECT wob.id
,      TO_CHAR(wob.ob_time,'YYYY-MM-DD HH24:MI') ob_time
,      wob.msl_pressure msl_pr
,      wob.msl_pressure_q msl_pr_q
,      midas.midas_utility_pkg.decode_q_fnc(wob.msl_pressure_q,
                                            'S') msl_pr_susp
FROM   midas.weather_hrly_ob wob
WHERE  wob.id_type = 'WMO'
AND    wob.id LIKE '03005%'
AND    wob.met_domain_name = 'SYNOP'
AND    wob.ob_time BETWEEN TO_DATE('2000-01-20 09:00',
                           'YYYY-MM-DD HH24:MI')
                   AND     TO_DATE('2000-01-22 10:00',
                           'YYYY-MM-DD HH24:MI');
```


FUNCTION	decode_to_decametres_fnc

Purpose: Converts coded values into decametres.

Parameters: p_code_val VARCHAR2 - The value to be decoded.
p_met_elem VARCHAR2 - The code name for the met element to be decoded.

Returns: Cloud height or visibility in decametres as NUMBER(5)

Description: MIDAS stores 8-group cloud height and visibility as values, not codes, and the stored values are in decametres. This function can be used to convert coded data into decametres during storage. The function will convert a code into the corresponding value in decametres for either cloud height (h_sh_s and h_th_t) or visibility (VV and V_sV_s). Inputs to the function are coded value and met element (i.e. cloud height or visibility). Since the input is a code, it is assumed to be of character datatype, and must be enclosed within quotes. Values less than 10 must include a leading zero. Where a range of values applies (e.g. code figure 91 in h_th_t) the function will return the lower value. The values for codes 1677 (CLOUD_HEIGHT) and 4377 (HORIZONTAL_VIS) are stored in the code_detail table (See section 14.5.3).

Example:

```
SELECT midas.midas_utility_pkg.decode_to_decametres_fnc
      ('56',
       'CLOUD_HEIGHT') cld_ht
,     midas.midas_utility_pkg.decode_to_decametres_fnc
      ('56',
       'HORIZONTAL_VIS') vis
FROM   dual;
```

FUNCTION	decode_wx_code_fnc

Purpose: Returns present or past weather code descriptions for a particular weather code.

Parameters: p_code_value VARCHAR2 - a present or past weather code e.g. prst_wx_id from weather_hrly_ob
p_src_opr_type VARCHAR2 - the source operation type code for the site (manned or automatic)
p_cld_ttl_amt_id VARCHAR2 - the cloud total amount (CLD_TTL_AMT_ID)

Returns: The code value description for the weather code value

Description: The function first determines which WMO code table holds the description of the weather code supplied, using the weather code and the source operation type code. It then calls function get_code_val_dsc_id_fnc to obtain the weather code description, which becomes the return value of decode_wx_code_fnc. The function currently returns only the first line of each weather description, but can return an up to 340 character string

Examples : **To obtain weather description for present weather code in** midas.weather_hrly_ob

```
SELECT wob.id
,     TO_CHAR(wob.ob_time,'YYYY-MM-DD HH24:MI')  ob_time
,     wob.prst_wx_id                          prst_wx
,     wob.cld_ttl_amt_id
,     wob.src_opr_type
,     midas.midas_utility_pkg.decode_wx_code_fnc(wob.prst_wx_id
                                      ,wob.src_opr_type
                                      ,wob.cld_ttl_amt_id)
FROM   midas.weather_hrly_ob   wob
WHERE  wob.src_opr_type >= '4'
AND    prst_wx_id IS NOT NULL
AND    wob.met_domain_name = 'SYNOP'
AND    wob.ob_time = TO_DATE('2001-06-01 00:00','YYYY-MM-DD HH24:MI');
```


To obtain weather description for first past weather code in midas.weather_hrly_ob

```
SELECT wob.id
,      TO_CHAR(wob.ob_time,'YYYY-MM-DD HH24:MI')  ob_time
,      wob.past_wx_id_1                           past_wx_1
,      wob.cld_ttl_amt_id
,      wob.src_opr_type
,      midas.midas_utility_pkg.decode_wx_code_fnc(wob.past_wx_id_1
                                                 ,wob.src_opr_type
                                                 ,wob.cld_ttl_amt_id)
FROM   midas.weather_hrly_ob   wob
WHERE  wob.id_type = 'WMO'
AND    wob.id IN ('03495','03772','03266', '03069','03763')
AND    wob.met_domain_name = 'SYNOP'
AND    wob.ob_time = TO_DATE('2001-06-01 00:00','YYYY-MM-DD HH24:MI');
```

To obtain weather description for present weather code in midas.glbl_wx_ob

```
SELECT gwo.id
,      TO_CHAR(gwo.ob_time,'YYYY-MM-DD HH24:MI')  ob_time
,      gwo.prst_wx_id                             prst_wx
,      gwo.cld_ttl_amt_id
,      gwo.src_opr_type
,      midas.midas_utility_pkg.decode_wx_code_fnc(gwo.prst_wx_id
                                                 ,gwo.src_opr_type
                                                 ,gwo.cld_ttl_amt_id)
FROM   midas.glbl_wx_ob    gwo
WHERE  gwo.id_type = 'WMO'
AND    gwo.id LIKE '076%'
AND    gwo.met_domain_name = 'SYNOP';
```

FUNCTION	feet_to_decametres_fnc

Purpose: Converts feet to decametres or decametres to feet

Parameters: p_feet NUMBER - The value, in feet, to be converted

Returns: The converted value in decametres (NUMBER(9,5))

Description: Most MIDAS values are in decametres (rather than metres). This function takes an input value in feet, and converts it to decametres. The range of input values is constrained to match the maximum value of visibility that MIDAS can store. The output can readily be converted into metres, and the inverse of the function can be applied to convert decametres to feet.

Examples : **Feet To Decametres**

```
SELECT '20'                                          ft
,      midas.midas_utility_pkg.feet_to_decametres_fnc(20)       dm
,      10 * midas.midas_utility_pkg.feet_to_decametres_fnc(20)  m
FROM   dual;
```

Decametres To Feet

```
SELECT '30'                                          dm
,      30 / midas.midas_utility_pkg.feet_to_decametres_fnc(1)  ft
FROM   dual;
```

FUNCTION	get_code_val_dsc_id_fnc

Purpose: Returns code description for a particular weather code given the relevant WMO code table number.

Parameters: p_code_id VARCHAR2 - The WMO code table number (code_id in the code table).
p_code_value VARCHAR2 - The value to be decoded.
p_desc_line_num NUMBER - Optional (default = 1)

Returns: The code value description for the weather code value

Description: The function selects the weather code description required , using the given WMO weather code table number, the weather code and line number of the description needed. The function returns NULL for all invalid values.

Example: **To obtain the first line of the weather description for present weather code in** midas.weather_hrly_ob

```
SELECT wob.id
,     TO_CHAR(wob.ob_time,'YYYY-MM-DD HH24:MI') ob_time
,     wob.prst_wx_id                    prst_wx
,     wob.past_wx_id_2
,     wob.cld_ttl_amt_id
,     wob.src_opr_type
,     midas.midas_utility_pkg.get_code_val_dsc_id_fnc(
                                        '4677',
                                        wob.prst_wx_id
                                        )
FROM   midas.weather_hrly_ob    wob
WHERE  wob.id_type = 'WMO'
AND    wob.id = '03495'
AND    wob.met_domain_name = 'SYNOP'
AND    wob.ob_time = TO_DATE('2001-06-01 00:00','YYYY-MM-DD HH24:MI');
```

FUNCTION	get_code_val_dsc_name_fnc

Purpose: Returns code description for a particular weather code given the relevant code name.

Parameters: p_code_name VARCHAR2 - The weather code name (code_name in the code table).
p_code_value VARCHAR2 - The value to be decoded.
p_desc_line_num NUMBER - Optional (default = 1)

Returns: The code value description for the weather code value

Description: The function selects the weather code description required , using the given code name, the weather code and line number of the description needed. The function returns NULL for all invalid values.

Example: **To obtain the first line of the weather description for present weather code in** midas.weather_hrly_ob

```
SELECT wob.id
,     TO_CHAR(wob.ob_time,'YYYY-MM-DD HH24:MI') ob_time
,     wob.prst_wx_id                    prst_wx
,     wob.past_wx_id_2
,     wob.cld_ttl_amt_id
,     wob.src_opr_type
,     midas.midas_utility_pkg.get_code_val_dsc_name_fnc(
                                        'PRES_WX_MAN',
                                        wob.prst_wx_id
                                        )
FROM   midas.weather_hrly_ob    wob
WHERE  wob.id_type = 'WMO'
AND    wob.id = '03495'
AND    wob.met_domain_name = 'SYNOP'
AND    wob.ob_time = TO_DATE('2001-06-01 00:00','YYYY-MM-DD HH24:MI');
```

FUNCTION	high_date_fnc

Purpose: Gives the highest date and time permitted in MIDAS.

Parameters: None.
Returns: Highest date and time
Description: This function has no parameters. It returns the highest date (and time) permitted in MIDAS, i.e. TO_DATE('3999-12-31 00:00','YYYY-MM-DD HH24:MI').

Example:
```
SELECT TO_CHAR(midas.midas_utility_pkg.high_date_fnc,
               'YYYY-MM-DD HH24:MI') high
FROM   dual;
```

FUNCTION	kjoule_to_watt_hour_fnc

Purpose: Convert radiation from KJoules/sq metre to Watt-Hours/sq metre.
Parameters: p_irad_amt NUMBER - The irradiation amount.
Returns: Irradiation amount in Watt Hours. Returns NULL for invalid values.
Description: Conversion supplied by C D Hall, 1J = 1 Watt/Second. 3600J = 1Watt/hr.

Example:
```
SELECT midas.midas_utility_pkg.kjoule_to_watt_hour_fnc(glbl_irad_amt)
FROM   midas.radt_ob_v2
WHERE  ROWNUM < 10;
```

FUNCTION	knots_to_beaufort_fnc

Purpose: Converts wind speed in knots to the equivalent Beaufort force number or description.
Parameters: p_knots NUMBER - Wind speed in knots
Returns: Beaufort force number or description. Returns NULL for invalid values.
Description: The function returns both Beaufort number and Description. Users may apply left substring to get number only. Beaufort descriptions are taken from WMO 306 Manual on Codes 1.1 A, 1995.

Example:
```
SELECT midas.midas_utility_pkg.knots_to_beaufort_fnc(30) Beaufort
FROM   dual;
```

FUNCTION	lon_band_fnc

Purpose: Gives code for a band of longitude.
Parameters: p_longitude NUMBER - Longitude.
Returns: Longitude band code
Description: This function returns the longitude_band_code for a specified longitude. It derives the longitude band code for insertion into the marine_ob table. This attribute forms part of the primary key of marine_ob and is the primary partition key for this table. This function should be used in most marine_ob queries to enable use of partitions.

Example:
```
SELECT 15.6                                    longitude
,      midas.midas_utility_pkg.lon_band_fnc(15.6) longitude_band_code
FROM   dual;
```

FUNCTION	low_date_fnc

Purpose: Gives the lowest date and time permitted in MIDAS.
Parameters: None.
Returns: Lowest date and time

Description: This function has no parameters. It returns the lowest date (and time) permitted in MIDAS, i.e. TO_DATE('0001-01-01 00:00','YYYY-MM-DD HH24:MI').

Example:
```
SELECT TO_CHAR(midas.midas_utility_pkg.low_date_fnc,
               'YYYY-MM-DD HH24:MI') low
FROM   dual;
```

FUNCTION	rank_met_domain_fnc

Purpose : Provides the precedence of a met domain

Parameters: p_tab_type VARCHAR2 - Table type.
p_met_dom VARCHAR2 - Met domain name.
Values as found in the met_domain_rank table.

Returns: Met domain rank as NUMBER.

Description: The function gives the relative worth (precedence) of a met domain within specified tables, so it can be used to get the 'best' rainfall figures for a station. Table type defines the table or view that the function is being used with e.g. 'RADIATION'. Met domain precedence is defined in the met_domain_rank table. Invalid values return NULL.

Example:
```
SELECT TO_CHAR(rdo.ob_date,'YYYY-MM-DD HH24:MI') ob_date
,      rdo.id
,      rdo.met_domain_name met_dom
,      midas.midas_utility_pkg.rank_met_domain_fnc
         ('drnl_rain',
          ,rdo.met_domain_name) rank
,      rdo.ob_end_ctime ob_ctm
,      rdo.ob_day_cnt
,      rdo.src_id
,      rdo.prcp_amt
FROM   midas.rain_drnl_ob  rdo
WHERE  rdo.id_type = 'RAIN'
AND    rdo.id LIKE '69%'
AND    rdo.ob_date BETWEEN TO_DATE('2000-01-01 00:00',
                           'YYYY-MM-DD HH24:MI')
                   AND     TO_DATE('2000-01-10 23:59',
                           'YYYY-MM-DD HH24:MI');
```

FUNCTION	throw_ob_date_fnc

Purpose: Throws observation date and time forwards or back by one day, depending upon clock-time.

Parameters: p_ob_time DATE - Observation date and time (in Oracle date format).
p_clock_time NUMBER - Clock-time limit.
p_direction NUMBER - Direction of throw (-1 = Throw back; +1 = Throw forward).

Returns: Date (VARCHAR2) to which the observation belongs.

Description: Some met elements, e.g. rainfall and temperature, are measured over 12-hour periods. These values may belong to the prior or next day, depending upon the observing period. This function compares ob date and time with the clock time limit supplied and returns the date +/- 1 day.
If the direction is -1 and the clock time of the ob time is less than or equal to the clock time limit then the thrown ob date is the ob date minus one day.
If the direction is +1 and the clock time of the ob_time is greater than the clock time limit then the thrown ob date is the ob date plus one day.
The function returns a date as VARCHAR2(10) to ensure unambiguous grouping by the return value. Observations for the hour 0000 are usually considered as being for the hour

ending at midnight, i.e. last hour of the day, but 2400 is not a valid time in the Oracle date datatype. The function returns NULL for all invalid values.

Examples :

```
SELECT midas.midas_utility_pkg.throw_ob_date_fnc
       (TO_DATE('2000-02-29 23:00','YYYY-MM-DD HH24:MI'),
       ,0600
       ,-1
       ) thrown_date
FROM   dual;


SELECT tob.id
,      tob.met_domain_name
,      tob.ob_hour_count
,      tob.max_air_temp
,      tob.min_air_temp
,      SUBSTR(midas.midas_utility_pkg.throw_ob_date_fnc
         (tob.ob_end_time-1
         ,1200
         ,+1)
       ,1,10) maxt_day
,      SUBSTR(midas.midas_utility_pkg.throw_ob_date_fnc
         (tob.ob_end_time
         ,1200
         ,+1)
       ,1,10) mint_day
FROM   midas.temp_drnl_ob   tob
WHERE  tob.id_type = 'DCNN'
AND    tob.id LIKE '02%'
AND    tob.ob_end_time BETWEEN TO_DATE('1999-01-01 00:01',
                                 'YYYY-MM-DD HH24:MI')
                       AND     TO_DATE('1999-01-04 23:59',
                                 'YYYY-MM-DD HH24:MI');
```

FUNCTION	calc_dwpt_fnc

Purpose: Calculates dew-point.

Parameters: p_air_temp NUMBER - Air temperature.
p_wetb_temp NUMBER - Wet-bulb temperature.
p_air_pres NUMBER - Air pressure.
p_st_flag NUMBER - Screen-temperature flag, i.e. exposure-type.

Returns: Dew-point temperature as NUMBER

Description: If air temperature or wet bulb are missing, then the function will return NULL. Set the screen temperatures flag to 1.0 for screen temperatures, otherwise 0.0. If air pressure is missing, the function will default to 1000.0.

Examples :

```
SELECT wob.id
,      TO_CHAR(wob.ob_time,'YYYY-MM-DD HH24:MI') wob_time
,      wob.air_temperature
,      wob.dewpoint
,      wob.wetb_temp
,      wob.msl_pressure
,      midas.midas_humidity_pkg.calc_dwpt_fnc
         (wob.air_temperature
         ,wob.wetb_temp
         ,wob.msl_pressure
         ,1.0) calc_dwpt
FROM   midas.weather_hrly_ob  wob
```



```
WHERE  wob.id_type = 'WMO'
AND    wob.id = '03772'
AND    wob.met_domain_name = 'SYNOP'
AND    wob.ob_time BETWEEN TO_DATE('1999-02-01 00:00',
                           'YYYY-MM-DD HH24:MI')
                   AND     TO_DATE('1999-02-03 23:59',
                           'YYYY-MM-DD HH24:MI');
```

FUNCTION	calc_hmr_fnc

Purpose: Calculates humidity mixing ratio.

Parameters: p_air_temp NUMBER - Air temperature.
p_wetb_temp NUMBER - Wet-bulb temperature.
p_air_pres NUMBER - Air pressure.
p_st_flag NUMBER - Screen-temperature flag, i.e. exposure-type.

Returns: Humidity Mixing Ratio as NUMBER

Description: If air temperature, wet bulb temperature or air pressure are NULL, then the function will return NULL. Set the screen-temperatures flag to 1.0 for screen temperatures or to 0.0 for aspirated temperatures. Other values default to 1.0.
NB: This function calculates HMR using the formula: HMR = 63197.0 * VP / (MSL_PRES - VP)
As described in Met Glossary and used in HORACE etc.
However, MOP3.LAND.FORT(HUMIDS) uses: HMR = 63197.0 * VP / (MSL_PRES + VP)

Example:

```
SELECT who.id
,      TO_CHAR(who.ob_time,'YYYY-MM-DD HH24:MI') ob_time
,      who.air_temperature
,      who.wetb_temp
,      who.msl_pressure
,      midas.midas_humidity_pkg.calc_hmr_fnc(who.air_temperature
                                   ,who.wetb_temp
                                   ,who.msl_pressure
                                   ,1.0) hmr
FROM   midas.weather_hrly_ob  who
WHERE  who.id_type = 'WMO'
AND    who.id = '03031'
AND    who.met_domain_name = 'SYNOP'
AND    who.ob_time BETWEEN TO_DATE('1999-02-01 00:00',
                           'YYYY-MM-DD HH24:MI')
                   AND     TO_DATE('1999-02-15 23:59',
                           'YYYY-MM-DD HH24:MI');
```

FUNCTION	calc_rh_fnc

Purpose: Calculates relative humidity.

Parameters: p_air_temp NUMBER - Air temperature.
p_wetb_temp NUMBER - Wet-bulb temperature.
p_air_pres NUMBER - Air pressure.
p_st_flag NUMBER - Screen-temperature flag, i.e. exposure-type.
p_trap_invalid VARCHAR2 - To suppress invalid values.

Returns: Relative Humidity as NUMBER

Description: If either air temperature or wet bulb temperature are NULL, then the function will return NULL. Set the screen-temperatures flag to 1.0 for screen temperatures or to 0.0 for aspirated temperatures. Other values default to 1.0. If air pressure is NULL,then a default of 1000.0 is used. If the boolean trap invalid flag is set to 'T' or is not set, then: if the calculated relative humidity is > 100% in certain frost conditions it is reset to 100%; or if

the calculated relative humidity is > 100% in other conditions, or is < 0%, it is reset to NULL. If the boolean trap invalid flag is set to 'F' then the calculated relative humidity is not reset even if outside range 0-100%.

NB: Relative humidity is available as a derived column on both the marine_ob and weather_hrly_ob tables.

Example:

```
SELECT who.id
,     TO_CHAR(who.ob_time,'YYYY-MM-DD HH24:MI') ob_time
,     who.air_temperature
,     who.dewpoint
,     who.wetb_temp
,     who.msl_pressure
,     midas.midas_humidity_pkg.calc_rh_fnc(who.air_temperature
                                          ,who.wetb_temp
                                          ,who.msl_pressure
                                          ,1.0
                                          ,'F') rltv_hum_fnc
,     who.rltv_hum rltv_hum_col
FROM   midas.weather_hrly_ob  who
WHERE  who.id_type = 'WMO'
AND    who.id LIKE '0300%'
AND    who.met_domain_name = 'SYNOP'
AND    who.ob_time BETWEEN TO_DATE('2000-01-20 09:00',
                           'YYYY-MM-DD HH24:MI')
                AND    TO_DATE('2000-01-20 12:00',
                           'YYYY-MM-DD HH24:MI');
```

FUNCTION	calc_svp_fnc

Purpose: Gives saturation vapour pressure for the specified temperature.

Parameters: p_air_temp NUMBER - Air temperature.
p_rh_flag VARCHAR2 - Relative humidity flag. Optional (default = 'N').

Returns: Saturation vapour pressure as NUMBER

Description: Calculates saturation vapour pressure from temperature and ice-bulb flag. Ice-bulb flag is set when dry-bulb air temperature < 0.0 deg.C. If relative humidity flag is set then saturation vapour pressure is calculated over water and never over ice despite temperature. By default it is not set.

Example:

```
SELECT who.id
,     TO_CHAR(who.ob_time,'YYYY-MM-DD HH24:MI') ob_time
,     who.air_temperature
,     who.wetb_temp
,     midas.midas_humidity_pkg.calc_svp_fnc(who.air_temperature) dry
,     midas.midas_humidity_pkg.calc_svp_fnc(who.wetb_temp) wet
FROM   midas.weather_hrly_ob  who
WHERE  who.id_type = 'WMO'
AND    who.id LIKE '0300%'
AND    who.met_domain_name = 'SYNOP'
AND    who.ob_time BETWEEN TO_DATE('2000-01-20 09:00',
                           'YYYY-MM-DD HH24:MI')
                AND    TO_DATE('2000-01-20 12:00',
                           'YYYY-MM-DD HH24:MI');
```

FUNCTION	calc_ua_hmr_fnc

Purpose: Calculate Humidity Mixing Ratio for upper air ascents.
Parameters: p_air_temp NUMBER - Air temperature.
p_dwpt NUMBER - Dewpoint.
p_air_pres NUMBER - Air pressure.
Returns: Humidity Mixing Ratio as NUMBER.
Description: This function calculates humidity mixing ratio from air temperature, dewpoint and air pressure, and gives more accurate results for upper air ascents than alternative methods, which use wet bulb temperature. The function returns NULL for all invalid values.

Example:

```
SELECT usp.id
,    TO_CHAR(usp.ob_time,'YYYY-MM-DD HH24:MI') ob_time
,    usp.ua_sndg_pt_num
,    usp.vert_sig_code
,    usp.ua_height
,    usp.pres_coord
,    usp.ua_air_temperature
,    usp.ua_dewpoint
,    midas.midas_humidity_pkg.calc_ua_hmr_fnc(usp.ua_air_temperature
                                    ,usp.ua_dewpoint
                                    ,usp.pres_coord) ua_hmr
FROM   midas.ua_sounding_point  usp
WHERE  usp.id_type = 'WMO'
AND    usp.id = '03496'
AND    usp.ob_time > TO_DATE('1989-05-01 00:00',
                             'YYYY-MM-DD HH24:MI')
AND    usp.ua_sndg_pt_num BETWEEN 40 AND 52;
```

FUNCTION	calc_ua_rh_fnc

Purpose: Calculate relative humidity for upper-air ascents.
Parameters p_air_temp NUMBER - Air temperature.
p_dwpt NUMBER - Dewpoint.
p_air_pres NUMBER - Air pressure.
Returns: Relative Humidity as NUMBER.
Description: Adapted from Met.O.22 subroutine, as developed by D Akeroyd (May 1986) and L Gibson (May 1987). The Met.O.22 subroutine was used in the SMR UA datasets (from which CDB datasets were derived). Returns NULL for invalid values.
Reference: Met.Mag., 114, No.1351, Feb 1985, pp49-56.
NB: An existing MIDAS function (calc_rh_fnc) calculates RH from air temperature, wet bulb and air pressure.

Example:

```
SELECT usp.id
,    TO_CHAR(usp.ob_time,'YYYY-MM-DD HH24:MI') ob_time
,    usp.ua_sndg_pt_num
,    usp.vert_sig_code
,    usp.ua_height
,    usp.pres_coord
,    usp.ua_air_temperature
,    usp.ua_dewpoint
,    midas.midas_humidity_pkg.calc_ua_rh_fnc(usp.ua_air_temperature
                                   ,ua_dewpoint
                                   ,pres_coord) ua_rh
FROM   midas.ua_sounding_point  usp
WHERE  usp.id_type = 'WMO'
AND    usp.id = '03496'
AND    usp.ob_time > TO_DATE('1989-05-01 00:00',
```



```
                    'YYYY-MM-DD HH24:MI')
AND    usp.ua_sndg_pt_num BETWEEN 40 AND 52;
```

FUNCTION	calc_vp_fnc

Purpose: Calculates vapour pressure using Regnault's formula.

Parameters: p_air_temp NUMBER - Air temperature.
p_wetb_temp NUMBER - Wet bulb temperature.
p_air_pres NUMBER - Air pressure.
p_st_flag NUMBER - Screen temperature flag i.e. exposure type.

Returns: Vapour pressure as NUMBER.

Description: If either air temperature or wet bulb temperature are NULL, then the function will return NULL. Set the screen temperatures flag to 1.0 for screen temperatures or to 0.0 for aspirated temperatures. Any other value will default to 1.0. If air pressure is NULL a default of 1000.0 is used.

Example:

```
SELECT who.id
,      TO_CHAR(who.ob_time,'YYYY-MM-DD HH24:MI') ob_time
,      who.air_temperature
,      who.wetb_temp
,      who.msl_pressure
,      midas.midas_humidity_pkg.calc_vp_fnc(who.air_temperature
                                  ,who.wetb_temp
                                  ,who.msl_pressure
                                  ,1.0) vpres
FROM   midas.weather_hrly_ob  who
WHERE  who.id_type = 'WMO'
AND    who.id LIKE '0300%'
AND    who.met_domain_name = 'SYNOP'
AND    who.ob_time BETWEEN TO_DATE('2000-01-20 09:00',
                           'YYYY-MM-DD HH24:MI')
                AND    TO_DATE('2000-01-20 12:00',
                           'YYYY-MM-DD HH24:MI');
```

FUNCTION	calc_wbt_fnc

Purpose: Calculates wet bulb temperature.

Parameters: p_air_temp NUMBER - Air temperature.
p_dwpt NUMBER - Dew point temperature.
p_wetb_temp NUMBER - Wet bulb temperature.

Returns: Wet bulb temperature as NUMBER.

Description: Wet bulb temperature is both an argument to the function and the return value. If the input contains a value for wet bulb then the function will return that same value. If the input wet bulb temperature is NULL then the function will attempt to calculate a value for it. (In a set of inputs some of the values may be NULL while other are not). The function will return NULL for any of the following conditions:

- Air temp is NULL
- Dewpoint is NULL
- Air temp < -40.0 deg.C. or > 40.0 deg.C.
- Dewpoint > air temp
- Function fails to converge on a satisfactory value after 4 iterations

The function calculates wet bulb to three decimal places, and does not throw, round or truncate the return value.

Description: **1. Compare calculated wet-bulb with existing stored values.**

```
SELECT who.id
,     TO_CHAR(who.ob_time,'YYYY-MM-DD HH24:MI') ob_time
,     who.air_temperature
,     who.dewpoint
,     who.wetb_temp
,     midas.midas_humidity_pkg.calc_wbt_fnc(who.air_temperature
                                    ,who.dewpoint
                                    ,NULL) wbt_calc
FROM   midas.weather_hrly_ob   who
WHERE  who.id_type = 'WMO'
AND    who.id = '03005'
AND    who.met_domain_name = 'SYNOP'
AND    who.ob_time BETWEEN TO_DATE('1999-02-03 00:00',
                             'YYYY-MM-DD HH24:MI')
               AND     TO_DATE('1999-02-11 23:59',
                             'YYYY-MM-DD HH24:MI');
```

2. Return existing wet-bulb where stored, otherwise calculate one.

```
SELECT who.id
,     TO_CHAR(who.ob_time,'YYYY-MM-DD HH24:MI') ob_time
,     who.air_temperature
,     who.dewpoint
,     who.wetb_temp
,     midas.midas_humidity_pkg.calc_wbt_fnc(who.air_temperature
                                    ,who.dewpoint
                                    ,who.wetb_temp) wbt_calc
FROM   midas.weather_hrly_ob who
WHERE  who.id_type = 'WMO'
AND    who.id = '03005'
AND    who.met_domain_name = 'SYNOP'
AND    who.ob_time BETWEEN TO_DATE('1999-02-03 00:00',
                             'YYYY-MM-DD HH24:MI')
               AND     TO_DATE('1999-02-11 23:59',
                             'YYYY-MM-DD HH24:MI');
```

12 Upper Air Observations

12.1 Introduction

This document refers specifically to upper air data within MIDAS, and replaces all earlier MIDAS upper air documents.

Midas stores selected upper air data in database tables, and considerably more data in the upper air archive. The upper air archive consists of a series of datasets, maintained in MASS.

The MIDAS database tables provide sufficient space for 500 station/years of observations, but the content of the database is not fixed. The tables are treated as a 'scratch' area, into which customers may load (subject to space availability) data of their choice from the upper air archive.

The MIDAS Upper Air sub-system consists of the following components:

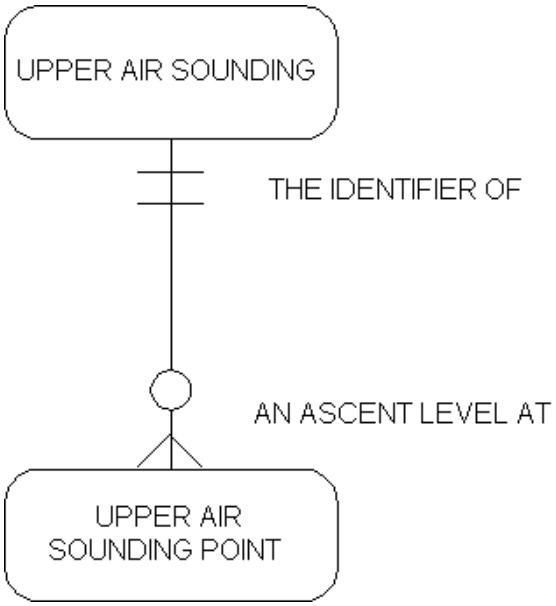
- Database tables and views
- Data (Upper Air observations stored in MIDAS and MASS)
- Standing data
- Software modules for routine data storage
- Software modules to load data into the database tables
- Software modules to query the data

The components listed above provide for the storage, manipulation and querying of upper air observations from land stations and ships. Each of these components is described in detail in the following sections.Upper air monthly averages and 30-year normals are also available, but are not covered in this document.

SQL code and sample program outputs are shown in `Courier font.`

12.1.1 Database Tables and Views.

MIDAS provides two tables for upper air observations, `UA_SOUNDING` and `UA_SOUNDING_POINT`



- `UA_SOUNDING` contains values that are common to all of the points in an upper air sounding (ascent), e.g. launch time, radar type, sonde type, etc.
- `UA_SOUNDING_POINT` describes the atmosphere at the specified level of an ascent, e.g. height, air pressure, temperature.

Each upper air ascent will have one `UA_SOUNDING` and may have one or more `UA_SOUNDING_POINT`. Each `UA_SOUNDING_POINT` belongs to one and only one `UA_SOUNDING`.

As previously mentioned, the contents of the database are not fixed. Customers may load data of their choice into the database tables. Customers cannot gain direct access to the database tables. Database views are

provided in both the MIDASUPD and MIDASVU schemas. All customers have both update and retrieval access via the MIDASUPD schema, and all users have retrieval access using the MIDASVU schema.

See Section 9.4 for a full description of the tables and their columns.

Relative humidity and humidity mixing ratio are not stored. MIDAS provides functions to calculate these values during retrieval.

12.1.2 Standing Data

The Midas SOURCE table describes the upper air stations. See *Section 9.2* for a description of the table.

The Midas SRC_CAPABILITY table shows:

- The types of report the station can make (e.g. TEMP or PILOT)
- The period for which data is available
- Where data is located

See Section 9.2 for a general description of the SRC_CAPABILITY table and its columns.

Some columns of SRC_CAPABILITY have specific meanings for upper air obs. They are:

Column Name	Data-Type	Description, Units
Met_Domain_Name	VARCHAR2(8)	Type of report. UAPLT = FM32 PILOT UATMP = FM35 TEMP UATMPSHP = FM36 TEMP SHIP
First_Online_Ob_Yr	NUMBER(4)	Year of the latest report
DB_Segment_Name	VARCHAR2(12)	e.g. B02999.Yyyyy. The contents of src_capability.db_seg_name must be prefixed with MSD5.MIDUPAIR., and yyyy must be interpreted to the last year of a 10-year period, using src_cap_end_date. Thus src_capability.db_seg_name = A03998.Yyyyy shows that the data are in MSD5.MIDUPAIR.A03998.Y1959, MSD5.MIDUPAIR.A03998.Y1969, etc., depending on the range of dates between src_cap_bgn_date and src_cap_end_date. When yyyy is replaced by a figure (e.g. U03998.Y1959) this indicates that the data are only in one dataset, MSD5.MIDUPAIR.A03998.Y1959 in this case. Recent data will also be available in the database tables.
Data_Retention_Period	NUMBER(3)	Number of years for which data is retained

See 3.1 for a list of the archive dataset names.

A sample of SRC_CAPABILITY data is shown below.

```
SRC_ID ID    MET_DOM CAP_BGN CAP_END LATEST_YR DB_SEG_NAME  RETENTION
------ ----- ------- ------- ------- --------- ------------ ---------
433    --    UATMP   ---     ---     2000      U03998.Yyyyy 999
1198   03496 UATMP   1999-   3999-   2000      U03998.Yyyyy 999
440    03502 UATMP   12-01   12-31   2000      U03998.Yyyyy 999
605    03590 UATMP   1999-   3999-   2000      U03998.Yyyyy 999
888    03649 UAPLT   12-01   12-31   2000      U03998.Yyyyy 999
888    03743 UATMP   1999-   3999-   2000      U03998.Yyyyy 999
1395   03743 UATMP   12-01   12-31   2000      U03998.Yyyyy 999
811    03808 UATMP   1999-   3999-   2000      U03998.Yyyyy 999
1490   03882 UATMP   12-01   12-31   2000      U03998.Yyyyy 999
17090  03920 UATMP   1999-   3999-   2000      U03998.Yyyyy 999
       03953         12-01   12-31
```



```
1999-  3999-
12-01  12-31
1999-  3999-
12-01  12-31
1999-  3999-
12-01  12-31
1999-  3999-
12-01  12-31
1999-  3999-
12-01  12-31
```

For convenience, `SOURCE` and `SRC_CAPABILITY` tables are joined in a view, `MIDASVU.CAPABILITY_AT_SRC`. See Section 9.3 for a description of this view.

12.1.3 3. Upper Air Data

The MIDAS upper air data is very voluminous, so it is primarily stored as a series of offline datasets within MASS. A subset of the data is duplicated within the MIDAS tables, and customers may add to the online data in the MIDAS tables by restoring the MASS data and loading it into MIDAS.

The MIDAS tables also contain standing data that provides details of the stations that make upper air observations and of the names of the MASS files where their upper air obs are stored.

Loading upper air data into MIDAS from the upper air MASS archive is a multi-step process, i.e.

- Query standing data to determine data availability and the generic name of the dataset where the required data is situated.
- Restore the files from MASS
- Load the obs using SQL*Loader utility
- Interrogate the upper air obs using either batch or online methods
- Notify the MIDAS Team when the customer has completed their data enquiry.

A Batch Suite Diagram (BSGP0002) is available on request from the Midas Team [redacted].

A set of software modules is available to facilitate the identification, restore and loading of the upper air obs, and a procedure to facilitate this process is provided.

When the upper air data has been restored from MASS, the customer may FTP them to another location or read from them directly, without loading them into the MIDAS tables. This approach is recommended when dealing with large volumes of data and the queries are not particularly sophisticated.

For more complex queries, the customer may load the data into Midas, and will thus be able to easily select data by several parameters, join it with standing data (or other Midas data), employ database functions, group the data, etc.

3.1 Archive Data

The upper air reports are maintained as a series of datasets in MASS. Each dataset contains the reports for a region and period, e.g. `MSD5.MIDUPAIR.A03998.Y1979` contains reports for UK and Ireland for the period 1970 - 1979. The datasets currently available are:

ID Range	DB_Seg_Name	Files
All SHIP reports	SHIPS.Y*yyyy*	MSD5.MIDUPAIR.SHIPS.Y1999 MSD5.MIDUPAIR.SHIPS.Y2002
WMO 01001 - 01415	B01999.Y*yyyy*	MSD5.MIDUPAIR.B01999.Y1999 MSD5.MIDUPAIR.B01999.Y2002
WMO 02185 - 02963	B02999.Y*yyyy*	MSD5.MIDUPAIR.B02999.Y1999 MSD5.MIDUPAIR.B02999.Y2002

WMO 03005 - 03953	A03998.Y*yyyy*	MSD5.MIDUPAIR.A03998.Y1959 MSD5.MIDUPAIR.A03998.Y1969 MSD5.MIDUPAIR.A03998.Y1979 MSD5.MIDUPAIR.A03998.Y1989 MSD5.MIDUPAIR.A03998.Y1999 MSD5.MIDUPAIR.A03998.Y2002
WMO 04018 - 04390	A0449999.Y*yyyy*	MSD5.MIDUPAIR.A04999.Y1979 MSD5.MIDUPAIR.A04999.Y1989 MSD5.MIDUPAIR.A04999.Y1999 MSD5.MIDUPAIR.A04999.Y2002
WMO 06011 - 06290	A06399.Y*yyyy*	MSD5.MIDUPAIR.A06399.Y1979 MSD5.MIDUPAIR.A06399.Y1989 MSD5.MIDUPAIR.A06399.Y1999 MSD5.MIDUPAIR.A06399.Y2002
WMO 06447 - 06610	B06999.Y*yyyy*	MSD5.MIDUPAIR.B06999.Y1999 MSD5.MIDUPAIR.B06999.Y2002
WMO 07110 - 07761	B07999.Y*yyyy*	MSD5.MIDUPAIR.B07999.Y1999 MSD5.MIDUPAIR.B07999.Y2002
WMO 08001 - 08589 (Except 08495)	B09999.Y*yyyy*	MSD5.MIDUPAIR.B09999.Y1999 MSD5.MIDUPAIR.B09999.Y2002
WMO 10035 - 11952	B11999.Y*yyyy*	MSD5.MIDUPAIR.B11999.Y1999 MSD5.MIDUPAIR.B11999.Y2002
WMO 12120 - 12982	B12999.Y*yyyy*	MSD5.MIDUPAIR.B12999.Y1999 MSD5.MIDUPAIR.B12999.Y2002
WMO 13130 - 16754	B16999.Y*yyyy*	MSD5.MIDUPAIR.B16999.Y1999 MSD5.MIDUPAIR.B16999.Y2002
WMO 17030 - 16607	B17999.Y*yyyy*	MSD5.MIDUPAIR.B17999.Y1999 MSD5.MIDUPAIR.B17999.Y2002
WMO 20107 - 38507	B39999.Y*yyyy*	MSD5.MIDUPAIR.B39999.Y1999 MSD5.MIDUPAIR.B39999.Y2002
WMO 40179 - 40875(Except 40648)	B40999.Y*yyyy*	MSD5.MIDUPAIR.B40999.Y1999 MSD5.MIDUPAIR.B40999.Y2002
WMO 4114 - 60020(Except 45004 pre 1979)	B60999.Y*yyyy*	MSD5.MIDUPAIR.B60999.Y1999 MSD5.MIDUPAIR.B60999.Y2002
WMO 61902 - 89009(Except 88889)	B99999.Y*yyyy*	MSD5.MIDUPAIR.B99999.Y1999 MSD5.MIDUPAIR.B99999.Y2002
WMO UK Overseas stations, i.e. 08495, 40648, 45004 (pre July 1979), 61901, 888889, 89022	UKOVRS.Y*yyyy*	MSD5.MIDUPAIR.UKOVRS.Y1999 MSD5.MIDUPAIR.UKOVRS.Y2002

Additionally, for each ID range there is a dataset with M*mm* appended to the name (e.g. MSD5.MIDUPAIR.B17999.Y2002.M*mm*) these contain the very latest data, but can be ignored because they duplicate the data in the online database tables.

12.1.4 Online Data

Customers may load the online tables with data of their choice, selected from the upper air archive that is maintained in MASS.

Any online data duplicates part of the corresponding archive data (i.e. the online tables contain copies of the selected archive data). Customers should be aware that they can update the online data, but these updates will not be copied back to the archive, and will be visible to all customers.

The database tables have space for 500 station/years of data, which is about 10% of the total available in the archive tapes.

Customers can load the tables with data of their choice, e.g.:

- All times for a station (or a group of stations)
- All stations for a period, e.g. All reports for October 1956
- A mixture of the above

The process of loading the online tables from the archive datasets is described in sections 6 and 7.

With effect from January 2000, the routine ingestion process stores the new reports in the online database as well as the archive, so MIDAS contains all reports within the Met Office area of responsibility since then.

There are no housekeeping routines to remove records from the database tables (the anticipated low usage does not warrant it) so customers must raise a Remedy incident when they have completed their enquiries, so that extraneous data may be deleted.

12.1.5 Routine Data Storage

A batch suite runs after the 15th of each month, to extract selected upper air reports for the previous month from MetDB. The suite processes these reports and stores them in both the online database and the archive.

Full details of the data storage process are recorded in MIDAS work instruction DDGWI015 and in Batch Suite BSGP0001 of the Select SE CASE tool, maintained by the Database Development Tech Centre.

12.1.6 How to: Find out what data is available online

The data that you may require may already be available in the MIDAS upper air tables. The following short SQL program will display each WMO number and report type, along with the dates of the first and last ascents and the count of ascents for this station.

```
-- MSD5.MIDUPAIR.SQL(AUDIT1)
SELECT                                  /*+ FULL(s) */
s.id_type
,s.id
,s.src_id
,s.met_domain_name                      Met_Dom
,MIN(TO_CHAR(s.ob_time,'YYYY-MM-DD'))   Min_Date
,MAX(TO_CHAR(s.ob_time,'YYYY-MM-DD'))   Max_Date
,COUNT(*)                               Rec_Cnt
FROM midasvu.ua_sounding s
```



```
WHERE s.id_type = 'WMO'                    -- To get ships, remove
                                           this line
GROUP BY
s.id_type
,s.id
,s.src_id
,s.met_domain_name ;
```

This program can be run inter-actively or in batch. See Section 10 for details of how to run in batch mode. To display SHIP reports as well as land stations, remove the WHERE s.id_type = 'WMO' clause. Remember that this program displays what is currently loaded into the database tables - much more data may be available in the archive.

Templates for more detailed audit programs are available in the GPCS upper air SQL library, see MSD5.MIDUPAIR.SQL(AUDIT2)and MSD5.MIDUPAIR.SQL(AUDIT3). Copy the required templates to your own library

12.1.7 How to: Find out what data is in the upper air archive (and where it is located)

The SOURCE and SRC_CAPABILITY tables can be searched using an application such as The Extractor or any online query software, or batch SQL, to determine the stations and dates required, An example batch SQL program is shown:

```
SELECT
sc.id                                  wmo
,SUBSTR(s.src_name,1,18)               src_name
,s.loc_geog_area_id                    cty
,sc.met_domain_name                    met_dom
,TO_CHAR(sc.src_cap_bgn_date,'DD-
MM-YYYY')                              cap_bgn
,TO_CHAR(sc.src_cap_end_date,'DD-
MM-YYYY')                              cap_end
,sc.db_segment_name                    file_name
,sc.first_online_ob_yr                 latest_ob
FROM midasvu.source    s
,midasvu.src_capability   sc
WHERE sc.src_id = s.src_id
AND sc.id_type = 'WMO'
AND sc.id LIKE '08%'
AND sc.met_domain_name LIKE 'UA%'
ORDER BY
sc.id
,sc.met_domain_name
,sc.src_cap_bgn_date;
```

```
WMO   SRC_NAME           CTY  MET_DOM CAP_BGN CAP_END FILE_NAME    LATEST_OB
----- ------------------ ---- ------- ------- ------- ------------ ---------
--    -                       -       ---     ---     -
08001 LA CORUNA          E    UATMP   01-01-  31-12-  B12999.Yyyyy 2002
08023 SANTANDER          E    UATMP   1990    3999    B12999.Yyyyy 2002
08160 ZARAGOZA/AEROPUERT E    UATMP   01-01-  31-12-  B12999.Yyyyy 2002
08221 MADRID/BARAJAS     E    UATMP   1990    3999    B12999.Yyyyy 2002
08301 PALMA DE MALLORCA  E    UATMP   01-01-  31-12-  B12999.Yyyyy 2002
08430 MURCIA             E    UATMP   1990    3999    B12999.Yyyyy 2002
08495 GIBRALTAR, NORTH F GIB  UAPLT   01-01-  31-12-  UKOVRS.Yyyyy 1996
```



```
08495 GIBRALTAR, NORTH F GIB  UATMP  1990    3999    UKOVRS.Yyyyy 2002
08508 LAJES/SANTA RITA ( PORT UATMP  01-01-  31-12-  B12999.Yyyyy 2002
08522 FUNCHAL            MDRA UATMP  1990    3999    B12999.Yyyyy 2002
08579 LISBOA/GAGO COUTIN PORT UATMP  01-01-  31-12-  B12999.Yyyyy 2002
08594 SAL                CAPV UATMP  1990    3999    B12999.Yyyyy 2002
                                     01-04-  13-03-
                                     1957    1996
                                     01-01-  31-12-
                                     1948    3999
                                     01-01-  31-12-
                                     1990    3999
                                     01-01-  31-12-
                                     2001    3999
                                     01-01-  31-12-
                                     1990    3999
                                     01-01-  31-12-
                                     1990    3999
12 rows selected.
```

See Section 9.3 for column definitions.

12.1.8 How to: Restore data from the MASS archive

Upper air data is classified as A or B list data according to its data retention period.
A list data is retained forever
B list data is retained for the last 10 years

Since 01 January 2000, selected upper air obs (i.e. obs for A-list stations and all ships in the North Atlantic) are loaded into the `UA_SOUNDING` and `UA_SOUNDING_POINT` tables of MIDASP routinely.

The remainder of the Upper air data is stored in MASS archive. A GPCS procedure (written in REXX) is provided to facilitate access to the archive data. The procedure allows the customer to select upper air data and restore it to a datafile, and optionally to load it into the database tables in MIDAST or MIDASP where it can be accessed by the usual Oracle database facilities.

To use the script, customers must log onto GPCS.
The script asks for the following information and uses this information to construct a network of 4 jobs.

1. Land or Sea Data
2. Restore or Restore and Load
3. Start date (yyyy,mm,dd)
4. End date (yyyy,mm,dd)
5. Required WMO Block. With the exception of UK overseas data, only one WMO block can be selected at a time
6. Whether selected stations are required
7. Tic Code for the jobs
8. Priority for the jobs
9. Customer's email name eg.jhayhurst
10. Whether MidasT or MidasP database loading is required
11. Customers are then invited to select particular stations from those available within the WMO block

The first job allocates the necessary dataset into which the second job runs a MASS restore job and places the whole of the dataset into the pre allocated file. The customer is notified by email when this is complete.
If the customer has specified that the data should be loaded to the database tables, then a third job is run to trim

the complete dataset to the customer's specified requirements, i.e. selected stations, and selected data periods. The data can be loaded to either ORAT or ORAP. The Customer is notified by email when the data has been loaded into the tables.

Requirements of the procedure in order to function are:

Look up table	Msd5.midupair.rexx. table
Job skeletons	Msd5.midupair.cntl (CLLOCX)
	Msd5.midupair.cntl (MREST2)
	Msd5.midupair.cntl (TRIMOUTP)
	Msd5.midupair.cntl(UPNEXT)

The table msd5.midupair.rexx.table is updated monthly as the final job in the upper air ingestion suite.

Instructions to run the procedure

Prerequisites for use are:

- A valid Cosmos User id and password.
- Knowledge of the tic code to which the jobs are charged

All screen information requests are shown enclosed by a box and in bold All responses are in bold and unboxed

Login to GPCS with User id and password.

At the READY command enter
Exec 'MSD5.midupair.cntl (upair)' and press the Right hand control key or the Enter key

All replies are sent by pressing either the Right hand control key or the Enter key after selection.

To escape from the procedure at any time press ESC key and then enter HI to exit.
This may not respond immediately if the procedure is processing a reply and the ESC key may need to be pressed a second time before the screen displays the exit message.

A code will appear to inform the customer of the session id.This is used as part of the datafile identification, e.g. A127.

Enter Land (L) or Sea (S) Upper Air data?

The customer must then select either L for land-based data or S for ships.
It does not matter if the reply is in upper or lower case.

Data restore only (RO) or Restore and load data to Oracle (RL)?

Start date (yyyymmdd)?

Enter the year month and day of the start of the required data.
NB: A-list data begins in 1949
B-list data begins in 1999

End date (yyyymmdd)?

Enter the year month and day of the end of the required data period

Enter nn where nn is the WMO block or OV for UK overseas stations

Enter the two-digit WMO block number, which must be between 01 and 99, or enter 'OV' for selection from the UK-overseas data, which includes such places as Gibraltar, Falkland Islands, Ascension etc.

See Appendix B for the details on WMO Block Numbers.

Do you wish to select stations within a WMO block Y/N ?

Answer N to load all stations within the WMO block.

An answer of 'Y' will present a list of stations to select from, after the job card information has been supplied.

At some point in the dialogue the screen will display a full list of the available stations for the block selected.
NB. For some blocks (e.g. block 03) this may be a lengthy list.

If the customer has chosen to select stations from this list, each station in the list will be presented to the customer and the customer must indicate whether or not the station is required by entering 's' or pressing the control key for non selection.

This could be a tedious process and therefore it may be better to restore all the stations within a block by entering 'N'

block is 03

Enter tic code for jobs

There is no check made for valid tic codes.
It is the customer's responsibility to ensure that a correct tic code is used to allow the job to run.

Enter priority for jobs

Valid priorities are 1,2,5,6,8,10.
A sensible choice is priority 8 if the data is required urgently and the amount of data moderate, Otherwise priority 5 should achieve restoration of the data overnight.

Enter the name part of your email address please- to allow notification of restore /loader completion

Again, it is the Customer's responsibility to enter the correct email name address, to ensure that he/she is notified of job completion.

The screen will display a full list of the available stations for the block selected.
NB. For some blocks this may be a lengthy list (e.g. Block 03)

If the customer has chosen to select stations from this list, each station in the list will be presented to the customer and the customer must indicate whether or not the station is required by entering 's' or pressing the control key for non selection.

This could be a tedious process and therefore it may be better to restore all the stations within a block by entering 'N'

Once selection has been made, the screen displays a list of stations selected

Stations selected are
03005
03023
03026

followed by the dataset name in which the JCL to submit the job (s) has been placed

T22JH.REX.A127

And the network id of the jobs in case they need to be cancelled.

N22JH12
Your data will be restored to T22JH.A127.UPTEMP.DATA
and will be available shortly
READY

The procedure can be used again once the ready command is displayed by repeating the instructions.

When the data has been restored, the customer will receive email notification
When the data has been loaded to MIDAST or MIDASP the customer will receive email notification.

Alternatives to selection displayed above
E.g. Restore and load will ask for database to load into. MIDAST or MIDASP

Enter Land (L) or Sea (S) Upper Air data ?

s (no station selection with s option)

Data restore only (RO) or Restore and load data to oracle (RL)?

rl

Start date (yyyymmdd)?

19990101

End date (yyyymmdd)?

19991231

Enter tic code for jobs

cds01a

Enter priority for jobs

8

Enter the name part of your email address please- to allow notification of restore/loader completion

jhayhurst

If data required is pre 2000 or a relatively small amount it can be loaded to MIDAST more efficiently

Enter 't' for system requirement MIDAST or enter 'p' for system requirement MIDASP

t
T22JH.REX.A778
N22JH77
Your data will be restored to T22JH.A778.UPTEMP.DATA
and will be available shortly
READY

The restore time for the data depends on MASS scheduling.
The dataset name listed above should be noted, as it is not repeated in the email notification.

12.1.9 How to: Work from the restored archive file(s)

Once the archived dataset has been restored from MASS, the customer may:

- Load it into Midas, to allow more sophisticated queries (see corresponding sections). If the customer has restored the archived data using the RL option, the loading will be done automatically.
- Send it to another location;
- Process it exactly as any other dataset.

If only simple processing is required, it is not necessary to load the data into the Midas tables.

A sample of data is shown below. In this case the data is for part of WMO block 06.

```
WMO 6011 20020601105617115 UATMP 1056 33 786  1026    671
WMO 6011 200206011056      11003 54   64 9.4  8.917   14SNNN
WMO 6011 200206011056      21000 79   32 9.4  8.618   20NNNN
WMO 6011 200206011056      003
                           932.0      2       200     29SSSN
```



```
WMO 6011 200206011056       004 925.0 724  32 7    6.7205  30NNNN
WMO 6011 200206011056       005 889.0 900  2       205     29SSSN
WMO 6011 200206011056       006 850.0 1417 32 4.2  3.92    31NNNN
WMO 6011 200206011056       007 844.0       4  4    3.6     SNNS
WMO 6011 200206011056       008 832.0       2       205     32SSSN
WMO 6011 200206011056       009 800.0       2       205     29SSSN
WMO 6011 200206011056       010 725.0       4  -3.3 -4      SNNS
WMO 6011 200206011056       011 701.0       4  -4.7 -7.1    SNNS
WMO 6011 200206011056       012 700.0 2973 32 -4.7 -7.1205 18NNNN
WMO 6011 200206011056       013 692.0       2       205     18SSSN
WMO 6011 200206011056       014 610.0       2       195     35SSSN
WMO 6011 200206011056       015 600.0       2       195     36SSSN
WMO 6011 200206011056       016 555.0       4  -
                                               14.7 -17.7   SNNS
WMO 6011 200206011056       017 500.0 5550 32 -
                                               20.3 -
                                                    23.4205 44NNNN
WMO 6011 200206011056       018 456.0       4  -
                                               25.5 -28.2   SNNS
WMO 6242 20020627120023559 UAPLT 1200
WMO 6242 200206271200       301 850.0 1500 32 275  14      S
WMO 6242 200206271200       302   1500 2  275  14      S
WMO 6242 20020627100023559 UAPLT 1000
WMO 6242 200206271000       301 850.0 1500 32 290  12      S
WMO 6242 200206271000       302   1500 2  290  12      S
```

The data are sorted so that a `UA_SOUNDING` row is followed by the corresponding `UA_SOUNDING_POINT` rows.

The archive files are defined below. Notice that character blanks in columns 30:32 of the data show the row to be a `UA_SOUNDING` one, while a number in columns 30:32 show the row to be a `UA_SOUNDING_POINT`.

Row definition, UA_SOUNDING

Column Name	Position	Len	Datatype
id_type	01:04	4	CHARACTER
id	05:12	8	CHARACTER
ob_time	13:24	12	DATE "YYYYMMDDHH24MI"
src_id	25:29	5	INTEGER EXTERNAL
met_domain_name	33:40	8	CHARACTER
ua_asc_lnch_ctm	41:44	4	CHARACTER INTEGER

wind_shr_abv_max_wind	45:47	3	CHARACTER INTEGER
wind_shr_blw_max_wind	48:5:	3	CHARACTER INTEGER
cld_base_amt_id	51:51	1	CHARACTER
low_cld_type_id	52:52	1	CHARACTER
cld_base_ht	53:55	4	CHARACTER INTEGER
med_cld_type_id	56:56	1	CHARACTER
hi_cld_type_id	57:57	1	CHARACTER
radar_type_id	58:59	2	CHARACTER
sonde_type_id	60:61	2	CHARACTER
latitude	62:66	5	CHARACTER DECIMAL (ships only)
longitude	67:72	6	CHARACTER DECIMAL (ships only)

Row definition, UA_SOUNDING_POINT

Column Name	Position	Len	Datatype
id_type	01:04	4	CHARACTER
id	05:12	8	CHARACTER
ob_time	13:24	12	DATE "YYYYMMDDHH24MI"
ua_sndg_pt_num	30:32	3	CHARACTER INTEGER
pres_coord	33:38	6	CHARACTER INTEGER
ua_height	39:43	5	CHARACTER INTEGER
vert_sig_code	44:46	3	CHARACTER INTEGER
ua_air_temperature	47:51	5	CHARACTER INTEGER
ua_dewpoint	52:56	5	CHARACTER INTEGER
ua_wind_dir	57:59	3	CHARACTER INTEGER
ua_wind_speed	60:62	3	CHARACTER INTEGER
ua_height_qc_code	63:63	1	CHARACTER
ua_air_temperature_qc_code	64:64	1	CHARACTER
ua_relative_humidity_qc_code	65:65	1	CHARACTER
ua_wind_qc_code	66:66	1	CHARACTER

12.1.10 How to: Load archived data into the online tables

Loading the archived data into the online tables allows the customer to employ more sophisticated queries, see Section 10 for details.

Before attempting to load archive (offline) data into the online tables, please ensure that the following steps have been completed:

- Check that the data are not already online.
- Find if the required data are in the archive, and the name of the archive file.
- Restore the data from the MASS archive
- Can the data be processed from the restored archive file?

Procedure MSD5.MIDUPAIR.CNTL(UPAIR) allows customers to easily restore data from the upper air archive in MASS and load it into Midas; see Section 7 for details. The procedure uses the SQL*Loader Oracle utility, but customers may elect to use the utility independently of the procedure.

MSD5.MIDUPAIR.SQL(MOD0015F) contains a script for SQL*Loader Oracle utility. Customers should copy this script to their own library and amend it as necessary.

```
-- MSD5.MIDUPAIR.SQL(MOD0015G)

LOAD DATA
APPEND
INTO TABLE midasupd.ua_sounding
WHEN src_id != ' '
AND id='03496 '
AND 13:20='19560402'
 (id_type              POSITION(01:04) CHAR,
 id                    POSITION(05:12) CHAR,
 ob_time               POSITION(13:24) DATE "YYYYMMDDHH24MI",
 src_id                POSITION(25:29) INTEGER EXTERNAL,
 met_domain_name       POSITION(33:40) CHAR,
 ua_asc_lnch_tm        POSITION(41:44) INTEGER EXTERNAL,
 wind_shr_abv_max_wind POSITION(45:47) INTEGER EXTERNAL,
 wind_shr_blw_max_wind POSITION(48:50) INTEGER EXTERNAL,
 cld_base_amt_id       POSITION(51:51) CHAR,
 low_cld_type_id       POSITION(52:52) CHAR,
 cld_base_ht           POSITION(53:55) INTEGER EXTERNAL,
 med_cld_type_id       POSITION(56:56) CHAR,
 hi_cld_type_id        POSITION(57:57) CHAR,
 radar_type_id         POSITION(58:59) CHAR,
 sonde_type_id         POSITION(60:61) CHAR,
 --latitude            POSITION(62:66) DECIMAL EXTERNAL,
 --longitude           POSITION(67:72) DECIMAL EXTERNAL
)
INTO TABLE midasupd.ua_sounding_point
WHEN ua_sndg_pt_num != ' '
AND id='03496 '
AND 13:20='19560402'
 (id_type                     POSITION(01:04) CHAR,
 id                           POSITION(05:12) CHAR,
 ob_time                      POSITION(13:24) DATE "YYYYMMDDHH24MI",
 ua_sndg_pt_num               POSITION(30:32) INTEGER EXTERNAL,
 pres_coord                   POSITION(33:38) INTEGER EXTERNAL,
 ua_height                    POSITION(39:43) INTEGER EXTERNAL,
 vert_sig_code                POSITION(44:46) INTEGER EXTERNAL,
 ua_air_temperature           POSITION(47:51) DECIMAL EXTERNAL,
 ua_dewpoint                  POSITION(52:56) DECIMAL EXTERNAL,
 ua_wind_dir                  POSITION(57:59) INTEGER EXTERNAL,
 ua_wind_speed                POSITION(60:62) INTEGER EXTERNAL,
 ua_height_qc_code            POSITION(63:63) CHAR,
 ua_air_temperature_qc_code   POSITION(64:64) CHAR,
 ua_relative_humidity_qc_code POSITION(65:65) CHAR,
 ua_wind_qc_code              POSITION(66:66) CHAR
)
```


The script loads both UA_SOUNDING and UA_SOUNDING_POINT tables. UA_SOUNDING must precede UA_SOUNDING_POINT rows. Customers may amend the conditions in the WHEN clause to suit their requirements, but src_id != ' ' is obligatory for loading UA_SOUNDING, as is ua_sndg_pt_num != ' ' for loading UA_SOUNDING_POINT.

Customers may specify any of the named parameters in the WHEN clause, e.g.

```
AND id_type ='SHIP'
AND id = 'DBBH '
AND ob_time = '200001032300'.
```

They may also specify a substring of the input, e.g. AND 13:20='19560402' would load all reports (in the dataset) for the given day.

Parameters may be rendered inactive by prefixing them with the comment characters '--'. The sample script above illustrates this, because latitude and longitude are not stored for land reports, so they are deactivated. Parameters in the 'WHEN'clause can also be deactivated if required. When loading ship reports, remember to reactivate the latitude and longitude attributes.

MSD5.MIDUPAIR.CNTL(MOD0015F) contains the JCL to run the SQL*Loader utility. Customers should copy this to their own library and amend it as required.

```
-- MSD5.MIDUPAIR.CNTL(MOD0015F)
//* Job Card required
//*
//PROCLIB JCLLIB ORDER=MCD.DBA.PROC
//*
//*-------------------------------------------------------------------*
//* ORACLE SQL*LOADER BATCH PROCESSOR *
//*-------------------------------------------------------------------*
//*
//SQL EXEC ORATLOAD,PARM='++/DD/LDRPARM',TIME=10
//SYSIN DD DUMMY
//LDRPARM DD *
CONTROL=/DD/CTL
DATA=/DD/DAT
BAD=/DD/BAD
LOG=/DD/LOG
USERID=/
/*
//CTL DD DSN=MSD5.MIDUPAIR.SQL(MOD0015F),DISP=SHR
//DAT DD DSN=MSD5.MIDUPAIR.U03998.Y2004,DISP=SHR
//BAD DD SYSOUT=*,DCB=LRECL=72
//LOG DD SYSOUT=*
//* DIS DD DUMMY
//* SQLLOGIN DD DUMMY
//
```

Use EXEC ORATLOAD to load to the MidasT Test database, and EXEC ORAPLOAD to load to the MidasP Production database.

//CTL specifies the file containing the SQL*Loader control statements.
//DAT specifies the file containing the input data (found from Section 7 above).

ERRORS file

The utility lists (counts) the number of records failing the 'WHEN' selection criteria, so if the customer has specified tight requirements (e.g. just one or two stations for just a few days) then the output will include a very large number of rejects. This is entirely normal.

If required, subsequent reports from the same region or from other regions (i.e. other datasets) may be appended.

12.1.11 How to: Query data in the upper air tables

The Midas tables provide great flexibility in querying the upper air data. The following facilities are available:

- Selection criteria in a WHERE clause (e.g. select by id, range of dates, height, pressure, vertical significance)
- Standard single-row number functions (e.g. FLOOR(), SQRT()) ;
- Met Office function for Rltv_Hum and Hum_Mix_Rto;
- Group functions, e.g. AVG(), MAX(), MAX(), STDDEV()

Various online SQL query tools (e.g. e.g. SQL Navigator, GQL, Extractor, etc.) are available to query the obs once they have been loaded into the database.

Alternatively, batch methods may be better suited to the large volumes of upper air data. SQL*Plus is an Oracle-supplied utility which can be used to run batch SQL scripts.

Template SQL scripts are presented below. They can be copied and modified as required. To provide a compact output for this Handbook, some columns have been omitted by changing the row into a comment. They can be re-instated by removing the -- comment marks..

```
-- Select from UA_Sounding only. See MSD5.MIDUAPAIR.SQL(SNDG)
SELECT
s.id
,TO_CHAR(s.Ob_Time,'YYYY-MM-DD HH24:MI')    Ob_Time
,s.met_domain_name                          met_dom
,s.src_id
,s.wind_shr_abv_max_wind                    wsa
,s.wind_shr_blw_max_wind                    wsb
,s.cld_base_amt_id
,s.low_cld_type_id                          l
,s.cld_base_ht                              cld_ht
,s.med_cld_type_id                          m
,s.hi_cld_type_id                           h
,s.radar_type_id                            ra
,s.sonde_type_id
FROM MidasVu.ua_Sounding s
WHERE
s.id_type = 'WMO'
AND s.id = '08495'
AND s.ob_time BETWEEN TO_DATE('20020605 06:00','YYYYMMDD HH24:MI')
AND TO_DATE('20020610 23:59','YYYYMMDD HH24:MI')
ORDER BY
s.id
,uas.ob_time DESC;
```

ID	OB_TIME	MET_DOM	SRC_ID	WSA	WSB	C	L	CLD_H	M	H	RA	SO
8495	05/06/2002 00:11	UATMP	1585		13	5	8	60	0	0	8	62
8495	04/06/2002 11:13	UATMP	1585	7	21	6	8	60	3	8	8	62
8495	03/06/2002 23:23	UATMP	1585	24	11	7	8	30	8	4	8	62
8495	03/06/2002 11:15	UATMP	1585	9	12	1	5	100	0	2	8	62
8495	02/06/2002 23:17	UATMP	1585			2	6	60	0	2	8	62


```
8495     02/06/2002 11:17 UATMP    1585              3 8 60    0 0 8  62
8495     01/06/2002 23:54 UATMP    1585              0 0 250   0 1 8  62
8495     01/06/2002 11:16 UATMP    1585              0 0 250   0 0 8  62
```

8 rows selected.

```
-- Select from UA_Sounding_Point. See MSD5.MIDUPAIR.SQL(SNGPT)
```

An example of using the relative humidity function is included as a comment. Remove the comment (--) marks to activate the function.

```
SELECT
p.id
,TO_CHAR(p.ob_time,'YYYY-MM-DD HH24:MI') Ob_Time
,p.ua_sndg_pt_num                        pt
,p.pres_coord                            pres
,p.ua_height                             ht
,p.vert_sig_code                         sig
,p.ua_air_temperature                    air_temp
,p.ua_dewpoint                           dwpt
,p.ua_wind_dir                           ddd
,p.ua_wind_speed                         fff
,p.ua_height_qc_code                     h
,p.ua_air_temperature_qc_code            t
,p.ua_relative_humidity_qc_code          r
,p.ua_wind_qc_code                       w
--,midas.uppr_air_rltv_hum(p.ua_air_temperature
-- ,p.ua_dewpoint
-- ,p.pres_coord) rh
FROM MidasVu.UA_Sounding_Point p
WHERE
p.id_type ='SHIP'
AND p.id = 'ZCBP6'
AND p.ob_time BETWEEN TO_DATE('20011006 00:00','YYYYMMDD HH24:MI')
AND TO_DATE('20011007 23:59','YYYYMMDD HH24:MI')
--AND p.ua_sndg_pt_num BETWEEN 040 AND 052
--AND p.pres_coord IS NOT NULL
--AND p.ua_air_temperature IS NOT NULL
--AND p.ua_dewpoint IS NOT NULL
ORDER BY
p.id
,p.ob_time
,p.pres_coord DESC
,p.ua_height;
```

ID	OB_TIME	PT	PRES	HT	SIG	AIR_TEMP	DWPT	DDD	FFF	H	T	R	W
ZCBP6	06/10/2001 22:25	1	1000	-139	32					N	S	S	S
ZCBP6	06/10/2001 22:25	2	981		64	12	7.1	310	35	S	N	N	N
ZCBP6	06/10/2001 22:25	3	947		4	8.2	4.6			S	N	N	S
ZCBP6	06/10/2001 22:25	4	925	509	32	6.6	3.8	320	35	N	N	N	N


```
ZCBP6   06/10/2001 22:25   5   882          4   3.2   2.8          S   N N S
ZCBP6   06/10/2001 22:25   6   868   900    2                  315  36  S S S N
etc.,
etc.
ZCBP6   07/10/2001 10:16   1   1001         64  10    6      310  52  S N N N
ZCBP6   07/10/2001 10:16   2   1000  33     32  9.6   5.4    310  52  N N N N
ZCBP6   07/10/2001 10:16   3   925   671    32  2.8   0.2    320  47  N N N N
ZCBP6   07/10/2001 10:16   4   924          4   2.8   0.2          S N N S
etc.,
etc.
ZCBP6   07/10/2001 16:58   1   1013         64  9.4   5.4    285  27  S N N N
ZCBP6   07/10/2001 16:58   2   1008         4   7.8   1.8          S N N S
ZCBP6   07/10/2001 16:58   3   1000  134    32  7.2   1.2    290  25  N N N N
ZCBP6   07/10/2001 16:58   4   952          2                  305  20  S S S N
ZCBP6   07/10/2001 16:58   5   925   768    32  1.2   0.6    315  20  N N N N
etc.,
etc.
168 rows selected.

-- Select from UA_SOUNDING joined with UA_SOUNDING_POINT.
MSD5.MIDUPAIR.SQL(SNDGALL)

SELECT
s.id
 ,TO_CHAR(s.ob_time,'YYYY-MM-DD HH24MI') ob_time
 ,s.met_domain_name                      met_dom
 ,s.src_id
 ,s.wind_shr_abv_max_wind                sa
 ,s.wind_shr_blw_max_wind                sb
 ,s.cld_base_ht                          Cld_Ht
 ,p.ua_sndg_pt_num                       Pt
 ,p.pres_coord                           Pres
 ,p.ua_height                            Ht
 ,p.vert_sig_code                        Sig
 ,p.ua_air_temperature                   air_temp
 ,p.ua_dewpoint                          dwpt
 ,p.ua_wind_dir                          ddd
 ,p.ua_wind_speed                        fff
 ,p.ua_height_qc_code                    h
 ,p.ua_air_temperature_qc_code           t
```



```
,p.ua_relative_humidity_qc_code          r
,p.ua_wind_qc_code                       w
FROM midasvu.ua_sounding s
,midasvu.ua_sounding_point p
WHERE
s.id_type = 'WMO'
AND s.id = '03920'
AND s.ob_time BETWEEN TO_DATE('20020602 00:00','YYYYMMDD HH24:MI')
AND TO_DATE('20020602 23:59','YYYYMMDD HH24:MI')
AND p.id_type = 'WMO'
AND p.id = '03920'
AND p.ob_time BETWEEN TO_DATE('20020602 00:00','YYYYMMDD HH24:MI')
AND TO_DATE('20020602 23:59','YYYYMMDD HH24:MI')
AND p.id_type = s.id_type
AND p.id = s.id
AND p.ob_time = s.ob_time
ORDER BY
s.id
,s.ob_time
,p.pres_coord DESC
,p.ua_height;
```

The output is voluminous, so is not displayed.

12.1.12 How to: Select data in its most compact form, using substitution variables, and including the required JCL

```
//* Job Card
//*
//PROCLIB JCLLIB ORDER=(DSM.STORAGE.JCL,MCD.DBA.PROC)
//*
//*
//EXECUTE EXEC ORAPSQL,PARM='-S /',TIME=(01,59)
//SYSOUT DD SYSOUT=*,DCB=LRECL=105
//SQLLOGIN DD *
SET TRIMSPOOL ON
SET SERVEROUTPUT ON
EXECUTE DBMS_OUTPUT.ENABLE(50000);
SET FEEDBACK OFF
SET PAGESIZE 0
SET HEADING OFF
/*
//SYSIN DD *
-- SELECT FROM MIDASVU.UA_SOUNDING, IN MOST COMPACT FORM --
-- Define Columns
COLUMN ID FORMAT A8 HEADING X TRUNCATED;
COLUMN ID_TYPE FORMAT A4 HEADING X TRUNCATED;
COLUMN OB_TIME FORMAT A14 HEADING X TRUNCATED;
COLUMN MET_DOMAIN_NAME FORMAT A8 HEADING X TRUNCATED;
COLUMN SRC_ID FORMAT 999990 HEADING X TRUNCATED;
COLUMN REC_ST_IND FORMAT 9990 HEADING X TRUNCATED;
COLUMN UA_ASC_LNCH_CTM FORMAT 9990 HEADING X TRUNCATED;
COLUMN WIND_SHR_ABV_MAX_WIND FORMAT 9990 HEADING X TRUNCATED;
COLUMN WIND_SHR_BLW_MAX_WIND FORMAT 9990 HEADING X TRUNCATED;
COLUMN CLD_BASE_AMT_ID FORMAT A1 HEADING X TRUNCATED;
COLUMN LOW_CLD_TYPE_ID FORMAT A1 HEADING X TRUNCATED;
COLUMN CLD_BASE_HT FORMAT 9990 HEADING X TRUNCATED;
COLUMN MED_CLD_TYPE_ID FORMAT A1 HEADING X TRUNCATED;
COLUMN HI_CLD_TYPE_ID FORMAT A1 HEADING X TRUNCATED;
```



```
COLUMN RADAR_TYPE_ID FORMAT A2 HEADING X TRUNCATED;
COLUMN SONDE_TYPE_ID FORMAT A2 HEADING X TRUNCATED;
COLUMN LATITUDE FORMAT 990.9 HEADING X TRUNCATED;
COLUMN LONGITUDE FORMAT 9990.9 HEADING X TRUNCATED;
--
-- Define Variables
DEFINE V_BGN_DATE = "20020605 0000"
DEFINE V_END_DATE = "20020610 2359"
DEFINE V_ID = "08495"
DEFINE V_Id_Type = "WMO"
DEFINE V_MET_DOMAIN = "UATMP"
--
SET LINESIZE 105
-- SET COLSEP ','
SET TRANSACTION READ ONLY;
--
SELECT
S.ID
--,S.ID_TYPE
,TO_CHAR(S.OB_TIME,'YYYYMMDDHH24MISS') OB_TIME
,S.MET_DOMAIN_NAME
,S.SRC_ID
--,S.REC_ST_IND
,S.UA_ASC_LNCH_CTM
,S.WIND_SHR_ABV_MAX_WIND
,S.WIND_SHR_BLW_MAX_WIND
,S.CLD_BASE_AMT_ID
,S.LOW_CLD_TYPE_ID
,S.CLD_BASE_HT
,S.MED_CLD_TYPE_ID
,S.HI_CLD_TYPE_ID
,S.RADAR_TYPE_ID
,S.SONDE_TYPE_ID
--,S.LATITUDE
--,S.LONGITUDE
FROM MIDASVU.UA_SOUNDING S
WHERE
S.ID_TYPE = '&V_ID_TYPE'
AND S.ID = '&V_ID'
AND S.OB_TIME BETWEEN TO_DATE('&V_BGN_DATE','YYYYMMDD HH24MI')
AND TO_DATE('&V_END_DATE','YYYYMMDD HH24MI')
;
/*
//

//* Job Card
//*
//PROCLIB JCLLIB ORDER=(DSM.STORAGE.JCL,MCD.DBA.PROC)
//*
//EXECUTE EXEC ORAPSQL,PARM='-S /',TIME=(01,59)
//SYSOUT DD SYSOUT=*,DCB=LRECL=87
//SQLLOGIN DD *
SET TRIMSPOOL ON
SET SERVEROUTPUT ON
EXECUTE DBMS_OUTPUT.ENABLE(50000);
SET FEEDBACK OFF
SET PAGESIZE 0
SET HEADING OFF
/*
```



```
//SYSIN DD *
-- SELECT FROM MIDASVU.UA_SOUNDING_POINT, IN MOST COMPACT FORM
--
-- Define Columns
COLUMN ID FORMAT A8 HEADING X TRUNCATED;
COLUMN ID_TYPE FORMAT A4 HEADING X TRUNCATED;
COLUMN OB_TIME FORMAT A14 HEADING X TRUNCATED;
COLUMN UA_SNDG_PT_NUM FORMAT 990 HEADING X TRUNCATED;
COLUMN VERT_SIG_CODE FORMAT 990 HEADING X TRUNCATED;
COLUMN UA_HEIGHT FORMAT 99990 HEADING X TRUNCATED;
COLUMN PRES_COORD FORMAT 99990.9 HEADING X TRUNCATED;
COLUMN UA_AIR_TEMPERATURE FORMAT 990.9 HEADING X TRUNCATED;
COLUMN UA_DEWPOINT FORMAT 990.9 HEADING X TRUNCATED;
COLUMN UA_WIND_DIR FORMAT 990 HEADING X TRUNCATED;
COLUMN UA_WIND_SPEED FORMAT 990 HEADING X TRUNCATED;
COLUMN UA_HEIGHT_QC_CODE FORMAT A1 HEADING X TRUNCATED;
COLUMN UA_AIR_TEMPERATURE_QC_CODE FORMAT A1 HEADING X TRUNCATED;
COLUMN UA_RELATIVE_HUMIDITY_QC_CODE FORMAT A1 HEADING X TRUNCATED;
COLUMN UA_WIND_QC_CODE FORMAT A1 HEADING X TRUNCATED;
--
-- Define Variables
DEFINE V_BGN_DATE = "20020605 0000"
DEFINE V_END_DATE = "20020610 2359"
DEFINE V_ID = "08495"
DEFINE V_ID_TYPE = "WMO"
--
SET LINESIZE 87
-- SET COLSEP ','
SET TRANSACTION READ ONLY;
--
SELECT
P.ID
--,P.ID_TYPE
,TO_CHAR(P.OB_TIME,'YYYYMMDDHH24MISS') OB_TIME
,P.UA_SNDG_PT_NUM
,P.VERT_SIG_CODE
,P.UA_HEIGHT
,P.PRES_COORD
,P.UA_AIR_TEMPERATURE
,P.UA_DEWPOINT
,P.UA_WIND_DIR
,P.UA_WIND_SPEED
,P.UA_HEIGHT_QC_CODE
,P.UA_AIR_TEMPERATURE_QC_CODE
,P.UA_RELATIVE_HUMIDITY_QC_CODE
,P.UA_WIND_QC_CODE
FROM MIDASVU.UA_SOUNDING_POINT P
WHERE
P.ID_TYPE = '&V_ID_TYPE'
AND P.ID = '&V_ID'
AND P.OB_TIME BETWEEN TO_DATE('&V_BGN_DATE','YYYYMMDD HH24MI')
AND TO_DATE('&V_END_DATE','YYYYMMDD HH24MI')
;
/*
//
```

`MSD5.MIDUPAIR.SQL` contains template SQL scripts.


```
(AUDIT1)   List the current contents of the database
(AUDIT2)   List the current contents of the database
(AUDIT3)   List the current contents of the database
(SEL01)    Query UA_SOUNDING
(SNDG)     Query UA_SOUNDING_POINT
(SNDGC)    Query UA_SOUNDING_POINT, in most compact
(SNDGALL)  form
(SNDGPT)   Query UA_SOUNDING_POINT joined with
(SNDGPTC)  UA_SOUNDING
(SRCCAP)   Query UA_SOUNDING_POINT, using database
           functions
           Query UA_SOUNDING_POINT, in most compact
           form
           Query SRC_CAPABILITY (e.g. to find tape
           dataset name)
```

`MSD5.MIDUPAIR.CNTL(SQLPLUS)` contains the JCL to run SQL scripts using the SQL*Plus utility. Copy this JCL to your own library and amend as necessary. Again, use ORATSQL for Test Database and ORAPSQL for Production database.

12.1.13 How to: Delete data from the online database

The online upper air tables have space for approximately 500 station/years of data. Customers are requested to notify the Midas Team when they have completed their data investigation so that the space can be made available to other customers.

Customers should raise a Remedy Incident, assigned to 'Midas Team', and listing the data that can be deleted. The support team will ensure that the data are removed.

12.1.14 Appendix A - Associations

Upper air design objects are documented in the `MIDASRS` Design Database of Systems Engineer.

Batch Suite General Picture	BSGP0002	Upper Air Extraction Routines
Batch Suite General Picture	BSGP0001	Upper Air Store/Delete - Oracle
Dataset General Form	D0000036	MSD5 MIDUPAIR Tyyyymm LANDDATA
Dataset General Form	D0000037	MSD5 MIDUPAIR Tyyyymm SHIPDATA
Dataset General Form	D0000038	MSD5 MIDUPAIR Unnnnn Yyyyyy
Function	F113	Store upper air observations
Module Set	MOD0015	Store upper air reports
Network ELH	ELH00445	Upper Air Sounding
Transaction	Xctn0011	Amend upper air observations
Transaction	Xctn0012	Delete time-expired upper air obs
Transaction	Xctn0010	upper air observations

Transaction	Xctn0009	Store upper air observations

13 Control Of MIDAS

13.1 Specification Of Responsibilities

TS(DD) Database & Archiving Group will provide operational support for:

- all technical issues relating to the database management system
- logical and physical design changes of the database
- maintenance of logical views of database tables
- regular security tasks - archive/recovery
- maintenance of ingestion software - whether from the operational real time databases or from COSMOS datasets
- maintenance of the OPR Fortran interface software

'Maintenance' includes the correction of errors in the software, changing the software to meet changes in external specifications, (e.g. the format of data retrieval from the MetDB), and the documentation.

TS(OPR3) will be responsible for:

- submitting and checking of **all meteorological data ingestion tasks** (the routine daily transfer from other databases is an automated process that is regularly re-submitted to COSMOS)
- the routine running of programs to update 'standing data'

Users will be responsible for their own MIDAS applications programs.

The TS(DD) Database & Archiving Group **may** be able to provide some assistance with programming difficulties and database performance issues **but**, as with other central computing issues, day-to-day problems should be reported to the COSMOS Help Desk on extension 6666.

13.2 What To Do If The Database Design Is Inadequate

1. Consult with your MIDAS User Group representative. There may be known reasons for the inadequacy.
2. If not, arrange to submit a MIDAS Change Request form to the MIDAS Support Team.

The effort required, and the impact on other users and procedures will be analysed. If, as a result of this, the change is agreed then it will be added to the Team's work schedule.

Changes are classified as either structural or procedural. Structural changes are those that affect the physical implementation of the MIDAS database, i.e. changes to tables, attributes, indexes. Procedural changes are those that affect the way in which the data in the database are defined, interpreted or used. Program changes are procedural. A structural change will almost always create associated program and procedural changes.

The MIDAS change control and configuration management procedures can be found at MIDAS Work Instruction

A report, showing the association of changes, problems and requirements to the various releases of MIDAS is available on request from the MIDAS Change Controller.

13.3 What To Do If The Data Details Are Wrong

OPR3 have the ultimate responsibility for all data in MIDAS.

Mistakes in observational data may be in the process of being rectified, but if not then OPR quality control staff will have the appropriate authority to change the data.

Errors in standing data can also be handled by OPR staff. However, procedures are available to add new details - e.g. a new met domain, a pertinent remark, a cross reference, a useful code table, etc.

13.4 Emergency Arrangements

In the event of an emergency that results in MIDAS being unavailable for an extended period, the following arrangements will apply:

- The MIDAS Team will act as the main point of contact for all incidents of this type.
- For all major incidents, the Corporate Database team will liaise primarily with the MIDAS team, who will be responsible for onward transmission of relevant information to users of the MIDAS service.
- The MIDAS Team will liaise with an agreed list of contacts for each of the following specified functional areas, and inform them of progress during the incident. The MIDAS Team will liaise with either the primary contact **or** a secondary contact (depending upon availability). Each of the contacts will be responsible for onward transmission of relevant information to **all** interested parties in their functional area. These 'functional contacts' will advise the MIDAS Team of the likely impact of the incident upon their business area.

Functional Area	**Responsible for advising:**
OPR	All OPR users, including Edinburgh
F/C (CP)	Commercial Suite product recipients Commercial enquiry officers

- If necessary, the Corporate Database team or the MIDAS team will inform the COSMOS Help Desk of the incident and its progress.
- The MIDAS Team will issue e-mail messages to an agreed list of users, advising them of progress. However, since some users only read e-mail infrequently this will be considered as a back-up mechanism to the 'quick response' chain outlined above. Although the MIDAS team will commit to issuing e-mail bulletins it remains the responsibility of functional contacts to inform their team members.
- As yet there is no cover outside of normal working hours.

14 Aspects of using MIDAS

14.1 Background

Oracle Server is installed on the GPCS to provide a relational database management system for the Met Office. Currently we have two "instances" running – **ORAP** and **ORAT**, offering "production" and "test" facilities respectively.

14.1.1 Structure

An Oracle database has both *physical* and *logical* structures. The **physical** structure is determined by the host operating system, and consists of data files, log files and control files to provide the actual physical storage for database information. The **logical** structure is determined by its *schema* objects (tables, views, indexes etc.) grouped together in some convenient way.

MIDAS is a schema. It has tables but no views.

MIDASVU is another schema. It contains many views but no tables. Users are provided with read-only access to the tables in MIDAS through views in MIDASVU. This schema also contains more specialised views (judicious combinations of tables).

MIDASUPD is yet another schema with views but no tables, its purpose being to control how the base tables in MIDAS are updated.

There are several other schemas in ORAP and ORAT but these are the ones of relevence to most MIDAS users.

14.1.2 Datatypes

The predominant datatypes used in MIDAS are:

Character - two forms are used to store character strings; `CHAR` for fixed length strings and `VARCHAR2` for variable length strings.
e.g.
`PRST_WX_ID` is defined as `CHAR(2)`, and *always* takes up 2 bytes.
`SRC_RMRK_TXT` is defined as `VARCHAR2(120)` and can take *up to* 120 characters.

Number - this is used to store fixed and floating point numbers using the general form `column_name NUMBER (precision, scale)` where `precision` represents the total number of digits, and `scale` the number of digits to the right of the decimal point.
e.g.
`WIND_DIRECTION NUMBER(3)` allows for a 3 digit integer.
`AIR_TEMPERATURE NUMBER(3,1)` allows values between -99.9 and 99.9.

Date - this stores the year (including the century), month, day, hour, minute and second of a date. Oracle actually uses its own internal format to store dates so some sort of conversion is always required for input and output. The standard format is `DD-MON-YYYY` as in `13-NOV-1992`. However this can be changed for a particular user *session* with the `ALTER SESSION` statement.
e.g.

```
ALTER SESSION SET NLS_DATE_FORMAT = 'YYYY-MM-DD HH24:MI'
```

Otherwise, for a particular *query* the `TO_DATE` function with an appropriate format mask can achieve the required result.
e.g.

```
TO_DATE('November 13, 1992','MONTH DD, YYYY')
```

If no time portion of a date field is entered it defaults to midnight.

Date arithmetic:

- You can add or subtract number constants as well as other dates.
- Oracle interprets number constants in arithmetic date expressions as numbers of days. e.g.`SYSDATE+1` is tomorrow, `SYSDATE-7` is one week ago and `SYSDATE+(10/1440)` is ten minutes from now.
- Subtraction of one date from another returns the number of days between the two dates.

Oracle also provides a number of date functions:

`ADD_MONTHS(d,n)`	date `d` plus `n` months
`LAST_DAY(d)`	date of the last day of the month in `d`
`MONTHS_BETWEEN(d1,d2)`	number of months between dates `d1` and `d2`
`NEXT_DAY(d,char)`	date of the first weekday named by `char` that is later than date `d`
`ROUND(d[,fmt])`	date `d` rounded to the unit specified by the format model `fmt`; omit `fmt` and `d` is rounded to the nearest day
`TRUNC(d[,fmt])`	date `d` with the time portion truncated by the format model `fmt`; omit `fmt` and `d` is truncated to the nearest day

Null - A null is the absence of a value in a column or row. It indicates missing, unknown or inapplicable data and should not be used to imply any other value, such as zero or 'blank'.

- Arithmetic expressions containing a null value evaluate to null
- To identify nulls in SQL use the IS NULL predicate
- To convert nulls to non-nulls use the NVL function

14.1.3 Access To Data

The basic tool for defining and manipulating data in a relational database is SQL (pronounced *sequel*). This is a simple but powerful language whereby single, sentence-like statements can generate significant amounts of database activity. Some will only return an indication of success or failure (e.g. `CREATE TABLE` …), whereas a `SELECT` statement can return many thousands of rows of data.

These statements can be executed in isolation, or embedded in a procedural language (such as Fortran or C) to add functionality to an application.

In addition Oracle has a procedural language of its own called PL/SQL. This allows a developer to exert some control over the flow of a sequence of SQL statements, to use variables and write error-handling procedures.

Blocks of SQL and PL/SQL statements can be grouped together as a unit to solve a specific problem or perform a set of related tasks, and then stored centrally in the database where they can be executed by a user or a database application.

Centralisation is an important aspect of this feature. Not only does it mean that network traffic between the application and the database is minimised, thus improving performance, but also that only one copy of a piece of logic is required. In turn this means that there is less excuse for re-inventing software modules, and maintenance of software is made easier.

14.1.4 Security

Oracle has an extensive range of powerful security features to control the access rights of database users. Since the GPCS provides the operating environment RACF is used for an initial user identification check. Having established that the user is authorised to use GPCS facilities Oracle then maintains a constant check on what that user tries to do against what it has been told is allowable. To do this various ‘roles’ have been established, each with specific rights and privileges, and a user is allocated one or more roles, as deemed appropriate.

Thus to access MIDAS, a potential user must:

1. have a valid userid
2. be registered in the Oracle security system

Administration of userids is controlled by departmental RACF administrators, whereas Oracle security is the responsibility of the Corporate Database team.

14.1.5 Application Environments

Oracle has a *client-server* architecture whereby the whole database system logically divides into 2 parts.

The *client* portion is the front-end application which has no direct data access responsibilities; it concentrates on requesting, processing and presenting the data that it receives. This application can be running on a powerful workstation with a network connection to the host platform, or part of a batch program running directly on the GPCS itself.

The *server* portion runs Oracle software and handles the functions required for concurrent shared data access. It receives and processes SQL and PL/SQL statements originating from client applications.

GPCS based applications

- Raw SQL can be submitted to the server using Oracle's SQL*Plus utility, using JCL of the form

```
//PROCLIB JCLLIB ORDER=MCD.DBA.PROC
//SQL EXEC ORAPSQL,PARM='/',TIME=n
//ORA@ORAP DD DUMMY
//SQLLOGIN DD DUMMY
//SYSIN DD DSN=hlq.lib_name.DATA(member_name),DISP=SHR
```

  / in the `PARM` field requests connection to Oracle with the current userid/password
  the `SYSIN` dataset contains the required SQL

  SQL*Plus offers many refinements, details of which can be found in the appropriate manual (follow the link from the MIDAS home page).
  The following is an example of what is possible:

```
//PROCLIB JCLLIB ORDER=MCD.DBA.PROC
//SQL EXEC ORAPSQL,PARM='-S /'
//ORA@ORAP DD DUMMY
//MYFILE DD DSN=hlq.file_name.DATA,DISP=(MOD,CATLG),
// STORCLAS=SCDATPRK,MGMTCLAS=MCSNC4,
// SPACE=(TRK,(1,1)),DCB=(RECFM=FB,LRECL=132,BLKSIZE=13200)
//SQLLOGIN DD DUMMY
//SYSIN DD *
SPOOL /DD/MYFILE
SET PAGESIZE 0
SET FEEDBACK OFF
--
ALTER SESSION SET NLS_DATE_FORMAT = 'YYYY-MM-DD HH24MI';
--
SELECT
OB_END_TIME, VERSION_NUM, ID, MEAN_WIND_DIR, MEAN_WIND_SPEED
FROM
MIDASVU.WIND_MEAN_OB
WHERE
ID_TYPE='WIND' AND ID='511303'
AND OB_END_TIME BETWEEN TO_DATE('1998010100','YYYYMMDDHH24')
AND TO_DATE('1998013100','YYYYMMDDHH24')
AND VERSION_NUM=1;
/*
```


where:

`-S` in the `PARM` field switches on 'silent' mode, suppressing all SQL*Plus information, command prompts, banners, etc.
`//MYFILE` defines a dataset which `SPOOL /DD/MYFILE` spools output to
`SET PAGESIZE` 0 suppresses all headings, page breaks, titles, etc.
`SET FEEDBACK OFF` suppresses the display of the number of records returned by the query with the result that the output from the query is available in a dataset for subsequent reading by some other process, without any extraneous information.

- SQL commands can be embedded in Fortran code, provided the module is presented to Oracle's Pro*Fortran pre-compiler first to convert all SQL code to standard Fortran. More extensive guidance notes are available from the Database Services group if required.

PC based applications

- There are a number of commercially available database query tools available, running under Microsoft Windows using 'point and click' methods to generate the necessary SQL for *ad hoc* queries. Microsoft Query is a relatively simple example.
- For more specific requirements, sophisticated applications can be developed using the likes of Microsoft Access or Visual Basic. For example the Technical Development group of Forecasting division have a PC-based tool that masks much of the intricacy of MIDAS and allows their commercial enquiry officers to select and retrieve data for external customers. Similarly a comprehensive interactive quality control application has been developed.

  These options require Oracle Client software to be installed on the PC.

14.2 Query Techniques

14.2.1 Using A Table Alias

An **alias** is a name that can be used to uniquely identify a database table without using its full name and can be used to simplify the appearance of SQL code.

```
SELECT so.src_id, so.src_name, so.loc_geog_area_id, ga.geog_area_id,
ga.geog_area_name
FROM midasvu.source so
, midasvu.geographic_area ga
WHERE ga.geog_area_id = so.loc_geog_area_id
AND so.src_id = 9;
```

With more complex queries, involving joins across several tables, the need for readability in the join criteria becomes even more important. Note that if an alias is specified it must be used instead of the table name

throughout the SQL statement, but it is only valid for that statement. Sometimes, however, the use of an alias is essential. Consider the following *invalid* example:

```
SELECT src_name, src_id, id_type, id, ob_end_time, prcp_amt
FROM midasvu.source
, midasvu.rain_hrly_ob
WHERE ob_end_time BETWEEN TO_DATE('199512010000','YYYYMMDDHH24MI') AND
TO_DATE('199512012359','YYYYMMDDHH24MI')
AND midasvu.source.src_id = midasvu.rain_hrly_ob.src_id;
```

Oracle responds with an error message of the form "`column ambiguously defined`". This is because `SRC_ID` occurs in the `SOURCE` table **and** the `RAIN_HRLY_OB` table. Re-writing the code as:

```
SELECT so.src_name, so.src_id, rh.id_type, rh.id, rh.ob_end_time, rh.prcp_amt
FROM midasvu.source so
, midasvu.rain_hrly_ob rh
WHERE rh.ob_end_time BETWEEN TO_DATE('199512010000','YYYYMMDDHH24MI') AND
TO_DATE('199512012359','YYYYMMDDHH24MI')
AND so.src_id = rh.src_id;
```

makes the distinction clear.

14.2.2 Sub-Queries

Sometimes a query will require information that is not immediately known, but can be deduced from another query.

e.g. Inferring `SRC_ID`

The key of the `SOURCE` record is `SRC_ID`, but its value is unlikely to be widely known, as it bears no obvious relation to the source itself. So it needs to be extracted from the `SRC_CAPABILITY` table first. However, it need not be a separate query with the result being manually fed into the next one - they can be nested.

```
SELECT so.src_name, so.east_grid_ref, so.north_grid_ref
FROM midasvu.source so
WHERE so.src_id =
(SELECT sc.src_id
FROM midasvu.sc_capability sc
WHERE sc.met_domain_name = 'SYNOP'
AND sc.id_type = 'WMO'
AND sc.id = '03100');
```

In this case the sub-query will return a single value. If it is likely to return a number of values (for example, if the `MET_DOMAIN_NAME` is not specified) the third line should use the **`IN`** operator:

```
WHERE so.src_id IN
(SELECT sc.src_id
FROM midasvu.sc_capability sc
WHERE sc.id_type = 'WMO'
AND sc.id = '03100');
```

e.g. Using an alternative identifier

Hourly data may be stored with a WMO identifier, but need to be retrieved using an equivalent DCNN.

```
SELECT wh.id, wh.ob_time, wh.msl_pressure, wh.air_temperature
FROM midasvu.weather_hrly_ob wh
WHERE wh.id_type = 'WMO'
AND wh.ob_time BETWEEN TO_DATE('199512010000','YYYYMMDDHH24MI') AND
TO_DATE('199512012359','YYYYMMDDHH24MI')
AND wh.id =
(SELECT sc1.id
FROM midasvu.src_capability sc1
WHERE scl.met_domain_name = 'SYNOP'
```



```
AND sc1.id_type = 'WMO'
AND sc1.src_id =
(SELECT sc2.src_id
FROM midasvu.src_capability sc2
WHERE sc2.met_domain_name = 'NCM'
AND sc2.id_type = 'DCNN'
AND sc2.ID = '5113'));
```

14.2.3 Table Joins

Quite often a query will require data from more than one table. For example, among the details stored in the `source` table is a code which can be used to cross refer to a record stored in the `geographic_area` table, which contains more information about that area. Thus, to display details of sources, including the name of the county in which they are located, requires both tables to be accessed. This can be achieved in a single query by **joining** the two tables temporarily, for the duration of the query. Effectively, this means forming pairs of rows by matching the contents of related columns in each table. In the example above each table has a column with a geographic area code, so the requirement can be met by:

```
SELECT so.src_name, so.east_grid_ref, so.north_grid_ref, ga.geog_area_name
FROM midasvu.source so
, midasvu.geographic_area ga
WHERE so.loc_geog_area_id = ga.geog_area_id
AND so.src_name LIKE 'HEATH%';
```

Equating the contents of the two columns like this limits the number of rows returned by the query. **Without this condition <u>each</u> row in one table would be matched with <u>all</u> those in the other, giving a large (and inaccurate) result table.** When a row in one table does not have a match in the other, then there is no corresponding row in the result table. If this is not what is required, it is possible to return unmatched rows as well.

For example, some weather observations have corresponding values in the `background_value` table. To select the corresponding values a table join will be required, but it is also a requirement to return observations that have no corresponding background values. In this case an ***outer join*** is used. An outer join is achieved by placing the symbol '(+)' following the column that may not contain a value.

```
SELECT wh.id, wh.src_id, wh.air_temperature, bv.background_air_temperature
FROM midasvu.weather_hrly_ob wh
, midasupd.background_value bv
WHERE wh.met_domain_name = 'SYNOP'
AND wh.id_type = 'WMO'
AND wh.ob_time BETWEEN (sysdate - 10) AND sysdate
AND wh.id = b.id (+)
AND wh.id_type = b.id_type (+)
AND wh.ob_time = b.ob_time (+);
```

The query above would include rows for observations without corresponding background values. The `background_air_temperature` will return values where a corresponding row is found and NULL where there is no corresponding row.

14.2.4 Set Operators

A join results in rows being returned from different tables in a *single query*, according to common values existing in corresponding columns.
Set operators can be used to manipulate the results returned by *multiple queries*, provided that the number and types of columns in those queries are identical.

- `union` returns all rows from two queries, except that any duplicate rows are eliminated
- `union all` returns all rows from two queries, including any duplicate rows

- `intersect` returns all rows that are common to both two queries
- `minus` returns all rows from the first query that are not returned by the second

14.2.5 The Group By And Order By Clauses

Group By

The `group by` clause allows users to summarise the rows of a table into groups.

```
SELECT post_code, COUNT(src_name)
FROM midasvu.source
GROUP BY post_code;
```

This query would list the number of sources for each post code.

Since group functions cannot be in a `where` clause, SQL uses a `having` clause to restrict the groups to be displayed.

```
SELECT post_code, COUNT(src_name)
FROM midasvu.source
GROUP BY post_code
HAVING COUNT(src_name) > 5;
```

Order By

Columns are returned to a query in the order specified in the `select` statement, but rows are returned in an order determined by the access strategy of the SQL optimiser. However, an `order by` clause can be added to sort the rows before they are finally returned.

```
SELECT src_name, east_grid_ref, north_grid_ref
FROM midasvu.source
ORDER BY src_name
```

This query will list all sources in alphabetical order.

The `order by` clause need not specifically reference a column name; it can be an aggregate function or a numerical reference to the required item in the `select` statement.

```
SELECT post_code, COUNT(src_name)
FROM midasvu.source
GROUP BY post_code
HAVING COUNT(src_name) > 5
ORDER BY COUNT(src_name);

SELECT post_code, COUNT(src_name)
FROM midasvu.source
GROUP BY post_code
HAVING COUNT(src_name) > 5
ORDER BY 2;
```

Both the above queries return the same result.

N.B. Under most circumstances the return of rows to a query is buffered, so using an `order by` clause may have a dramatic effect on the performance, since all the data must be retrieved before any ordering can be done.

14.3 Useful MIDAS Tables

14.3.1 Start At The Source

The design of MIDAS allows for a flexible approach to data access, but until you have gained some experience a disciplined approach can pay dividends.

Faced with a requirement for meteorological data for Little Tiddleypush, check in the source table to see if there is a likelihood of finding any data. If there isn't a source (i.e. station), then there will not be a source-capability, and consequently no data.

There are numerous ways of using SQL to search for suitable sources, depending on what information is known about them.

- Find sources by name

  ```
  SELECT src_id, src_name, high_prcn_lat, high_prcn_lon, src_bgn_date, src_end_date
  FROM midasvu.source
  WHERE src_name LIKE 'LITTLE TID%';
  ```

This will retrieve all sources whose name begins with the specified characters.

- Find sources by grid reference

  ```
  SELECT src_id, src_name, grid_ref_type, east_grid_ref, north_grid_ref, src_bgn_date, src_end_date
  FROM midasvu.source
  WHERE grid_ref_type = 'OS'
  AND east_grid_ref BETWEEN value_E1 and value_E2
  AND north_grid_ref BETWEEN value_N1 and value_N2;
  ```

This will retrieve all sources located in the NGR rectangle specified.

- Find sources by post-code

  ```
  SELECT src_id, src_name, grid_ref_type, east_grid_ref, north_grid_ref, src_bgn_date, src_end_date
  FROM midasvu.source
  WHERE post_code LIKE 'RG12%';
  ```

This will retrieve all sources with a similar post code.

14.3.2 Check The Capability

Having found a source, either by one of the methods outlined above or from past knowledge, check to see if the source is capable of producing the data you require. Again, if there is no capability then there will be no data. If you used one of the methods in B3.1, you will already have the unique src_id of the location you are interested in.

```
SELECT id_type, id, src_id, met_domain_name, src_cap_bgn_date, src_cap_end_date
FROM midasvu.src_capability
WHERE src_id = value;
```

This will tell you what reports are available from this location.

14.3.3 Sources, Identifiers And SRC_ID

A source, in MIDAS terms, is a place of origin for meteorological observations. It can have several different labels, or identifiers. Heathrow could be referred to by: 03772 (WMO), 5113 (DCNN), 247436 (RAINFALL) or EGLL (ICAO)

Some identifiers, such as rainfall numbers, have a single purpose but others serve several functions. For example, Lerwick reports synops and upper air ascents using its wmo number.

src_id is a unique number that is applied to a source. Each new source that is added to MIDAS is allocated its own src_id. The number itself is not directly related to the source as such, but it is used to distinguish between different locations. This means that it is an effective shorthand notation for the specification of that location, which would normally require lat/long, or an NGR pair, and altitude. src_id is included as an attribute of the observation entities to allow users to join values in different tables without knowledge of all of the identifiers that a source can use.

For example, suppose you require a certain station's hourly air temperatures (from the weather_hrly_ob table, which uses WMO) and hourly rainfall (from the rain_hrly_ob table, which uses rainfall number). You know the WMO of the station, but you can't remember the rainfall number. The following request should find the data you require:

```
SELECT wh.ob_time, wh.air_temperature, rh.prcp_amt
FROM midasvu.weather_hrly_ob wh
, midasvu.rain_hrly_ob rh
WHERE wh.id_type= 'WMO'
AND wh.ID = '03772'
AND wh.ob_time BETWEEN TO_DATE('2000050100','YYYYMMDDHH24')
AND TO_DATE('2000050123','YYYYMMDDHH24')
AND rh.src_id = wh.src_id
AND rh.ob_end_time = wh.ob_time;
```

Notice that we did not need to specify the rainfall number (though we could have found it using the src_capability record).

src_id is an excellent means of joining tables, but it is not necessarily a good parameter to use for extracting data. For example, assume you require some rainfall data for Little Tiddleypush, and you know that its src_id is 54321.

```
SELECT rd.ob_date,rd.prcp_amt
FROM midasvu.rain_drnl_ob rd
WHERE rd.src_id = 54321
AND rd.ob_date >= '1980';
```

The query shows that no data are received after 1982. Why? The station moved during that year and the new position warranted a new NGR. Each MIDAS source is at a specific position, and has a separate src_id.

```
SELECT so.src_name, so.src_id, so.north_grid_ref, so.east_grid_ref, so.src_bgn_date,
so.src_end_date
FROM midasvu.source so
WHERE so.src_name LIKE 'LITTLE TID%';
```

This query reports the position of all appropriate sources. The mistake would not have occurred if the capability had been checked first (14.3.2).

14.3.4 Marine Reports Are Different

Marine reports differ from land station reports in a number of important respects.

- Ship reports are identified by the ship's call sign, so the id_type is SHIP and id is the call sign.
- A great many ships do not include a valid call sign in their reports; the call sign may be missing or invalid. When this occurs, Midas will substitute the call sign value SHIP. We call these "generic" ships, because we know nothing of them.
- Logically, there is no such thing as a source record for ships, because a source implies a fixed point. However an Ocean Weather Ship is an exception, with a number of ships occupying the station position. In this case, there will be a source record, with a number of capabilities.
- The attributes of the primary key of ship reports are different to reports from land stations. The primary key includes position as well as identifier. Position is more significant in the key than identifier.
- It is possible to have two ships at the same position.

14.3.5 Code Tables

As you might expect, many of the values in MIDAS are stored using meteorological codes. To assist the user there is a `code_detail` table which lists coded values and their meanings for many WMO code tables e.g.

code_id	code_name	Description
0200	0200	WMO code 0200 Characteristic of pressure tendency.
3551	3551	WMO code 3551 Rate of ice accretion on ships.
4677	4677	WMO code 4677 Present weather reported from a manned station.
8007	PRESSURE	MIDAS met element measurement code - Pressure.

The query below will return a `code_val_dsc` of : 'Rain, not freezing, continuous, heavy at time of ob.'

```
SELECT cd.code_val_dsc
FROM midasvu.code_detail cd
WHERE cd.code_id = '4677'
AND cd.code_value = '65';
```

14.3.6 Remarks About A Source

Entries in the `src_remark` table are classified by a range of types.

`ASSOCIATED RAINFALL`	documents the historical association of rainfall stations with a DCNN. (Also associated by the `cross_reference` table)
`DATA ROUTE`	describes how the data reach the Met Office
`HISTORICAL NOTES`	significant events from the past
`IDENTIFIERS`	lists the various identifiers that are/have been applicable to a source
`INSTRUMENTATION`	details of some of the instruments used at a site
`MEASUREMENT`	significant aspects of the measurement of values at a site
`MISSING DATA`	lists periods of known missing data
`OBSERVING PRACTICE`	describes the frequency of observations at various times
`SITE INFORMATION`	describes site details

The entries were merely compiled as part of the process of loading station details and are by no means complete. There is no restriction on the type and extent of remarks that can be stored. For example, comments on exposure and data quality might be considered useful. If the information is available it can be stored.

14.3.7 How To Find Where A Met Element Is Stored

As a total novice, interested in a particular met element, how can you find out where it is stored? If you know what type of report it is received in, e.g. SYNOP or NCM, then use section 9 of this handbook to find the table where it is stored. Alternatively, use the `met_element` table to find the id of the element you are interested in.

```
SELECT me.met_element_id, me.met_element_name
FROM midasvu.met_element me
WHERE me.met_element_name LIKE 'WIND SP%';
```

This should tell you that the `met_element_id` of `WIND SPEED` is `11012`. You can now use this fact to query the `domain_element` table to find all met domains that the element occurs in.


```
SELECT de.met_domain_name, de.str_met_dom_name
FROM midasvu.domain_element de
WHERE de.met_element_id = 11012;
```

`str_met_dom_name` contains a shorthand version of the name of the table in which the element is stored. If required, you can also query the `met_domain` table using the value of `met_domain` in `domain_element`

```
SELECT *
FROM midasvu.met_domain md
WHERE md.met_domain_name = 'SYNOP';
```

In the `met_domain` table, those entries with a met domain usage id (`dom_usg_id`) of "I" indicate 'Input' or 'Reported' domains; others, with a `dom_usg_id` of "S" specify the tables where elements are stored.

14.4 SQL Performance Factors

Many factors affect the performance of any computer system, but when using an **ad-hoc** query language such as SQL, the way in which the query is structured is particularly important. This section of The MIDAS Handbook explains some of the performance implications of certain queries, and presents some of our experiences with MIDAS and SQL.

Any user contributions to this section are especially welcome.

14.4.1 Specify ID_TYPE when extracting by ID

An identifier consists of a value, e.g. 03344 and a type, e.g. WMO. The type is as much an important part of identification as the value, in that they should form a unique combination (e.g. EGLL on its own might be Heathrow's ICAO indicator or a ship call sign). For this reason the combination always forms a significant part of the key. The SQL optimiser attempts to find the best route to the records you want by examining what you have specified. If you provide enough information for it to use an index, it will, if not, it will conclude that it has to sweep the entire area, and that takes time. If you supply an `id_type` as well as an `id` there is a much better chance of getting a quick response.

14.4.2 Selecting By Date/Time

If no time portion of a date field is entered it defaults to midnight which can give unexpected results if the selection criteria are not specified unambiguously. It is always better to use the `TO_DATE` function with an approriate format mask.

```
SELECT *
FROM midasvu.weather_hrly_ob who
WHERE who.id_type = 'WMO'
AND who.id = '03160'
AND who.ob_TIME BETWEEN '01-JAN-1989' AND '02-JAN-1989'
```

returns 24 rows for Jan 1st and 1 for Jan 2nd (for midnight), whereas

```
SELECT *
FROM midasvu.weather_hrly_ob who
WHERE who.id_type = 'WMO'
AND who.id = '03160'
AND who.ob_time BETWEEN TO_DATE('01-JAN-1989 0000','dd-mon-yyyy hh24mi')
AND TO_DATE( '02-JAN-1989 2359','dd-mon-yyyy hh24mi')
```


returns 48 rows.

14.4.3 Use Of Partition Keys

Most of the observational data tables have been "partitioned". Effectively this means that a large table can be thought of as a collection of smaller ones, all with the same name. However, if the SQL optimiser is given sufficient information, it will be able to limit the search for data to the right partition(s). All these tables have date/time in the partition strategy, but the main component for marine and global data is based on a regional division.

glbl_wx_ob

For `glbl_wx_ob`, the partition key is `wmo_region_code` plus `ob_time`, where `wmo_region_code` is a single character in the range 1 - 7 representing the following regions:

1 Africa
2 Asia
3 South America
4 North and Central America
5 South-west Pacific
6 Europe
7 Antarctic

Performance will be enhanced if the appropriate value for a given station is provided, either directly if known, or inferred from its associated `source` record.

```
SELECT *
FROM midasvu.glbl_wx_ob gwo
WHERE gwo.id_type = 'WMO'
AND gwo.id = '62002'
AND gwo.ob_time BETWEEN TO_DATE ('2000010100','yyyymmddhh24')
AND TO_DATE ('2000013123','yyyymmddhh24')
AND gwo.wmo_region_code =
(SELECT src.wmo_region_code
FROM midasvu.source src
WHERE src.src_id IN
(SELECT cap.src_id
FROM midasvu.src_capability cap
WHERE cap.id_type = 'WMO'
AND cap.id = '62002'))
```

marine_ob

The `marine_ob` table is partitioned by `longitude_band_code` and by `ob_time`. `longitude_band_code` is a single character in the range A - J, each covering a set of longitude values. The required value can be determined from the `lon_band` function which has longitude as an argument. Best performance will require specification of both `longitude_band_code` and `ob_time`.

```
SELECT *
FROM midasvu.marine_ob mo
WHERE mo.latitude BETWEEN 50.4 AND 50.6
AND mo.latitude BETWEEN -10.6 AND -10.4
AND mo.ob_time BETWEEN TO_DATE ('2000010100','yyyymmddhh24')
AND TO_DATE ('2000013123','yyyymmddhh24')
AND mo.longitude_band_code = midasvu.lon_band(-10.5)
```


14.4.4 Performance Implications Of Order By

Data retrieval is buffered, which means that initial display of data should be fairly rapid, even if the full query will eventually return a lot more rows. However, using `ORDER BY` implies that **all** rows have to be retrieved, into some temporary workspace area, before they can be shuffled according to the requirements of the clause. Inevitably, this takes longer and can be appreciable for a large query.

14.4.5 Performance Implications Of Sub-Queries, Outer Joins etc

Inevitably, the more complex the query - several tables joined or several nested sub queries - the more likely that performance will suffer. It is also much more likely that, while syntactically correct, the query will not actually reflect requirements.

14.5 Static Values - Quality Control Information

Observation sub-types can contain attributes that store information about the quality control that has been applied. These take the format met_element_name_q and met_element_name_j. The attribute name depends on the observation sub-type.

`met_element_name_q`
This attribute is a five digit number (of the form MESQL), where each digit describes one aspect of the quality of a meteorological element.

M	qc_marker	A combination of two flags to indicate accumulations, and/or a trace of rainfall
E	qc_estimate	Indicates 1 of up to 8 remarks about an estimate
S	qc_status	Indicates 1 of up to 8 possible descriptions of the value of the element (e.g. reason for suspecting a value)
Q	qc_query	Indicates 1 of up to 8 statements about the original value (e.g. a reason for correcting a data value)
L	qc_level	Indicates which of 10 possible stages of QC has been reached

qc_marker (code_id 8021) when set, indicates a precipitation trace or an accumulated value (e.g. rainfall or maximum temperature over more than one day; maximum gust speed over more than one hour).

0	NULL
1	Trace of precipitation
2	Accumulation
-2	Accumulated trace of precipitation

qc_estimate (code_id 8022) provides information about an estimated or corrected value. In this context an "estimate" is a value either derived by the QC teams where the observer or the automatic instrument has not provided a value, or has been provided retrospectively from the observing station after the original report was ingested with the element in question missing.

0	Value is not an estimate or correction, or information on the estimate/correction is not available
1	Estimate/correction derived automatically from a program with no manual intervention
2	Estimate/corrected value has been set manually (with or without assistance from a program)
3	Estimate/correction has been obtained retrospectively from the observer/station
4	Precipitation estimate/correction has been derived from a snow/rain equivalence, or trace estimate/correction has been set for consistency with present weather
5	Correction has been obtained by changing units of measurement
6	Correction has been obtained by applying a systematic adjustment
7	Measurement impossible, because of snow, etc.

qc_status (code_id 8023) indicates the status of the data value.

0 Observed and not suspect
1 Observed and suspect (i.e. has failed the latest QC check), or there are strong grounds for suspecting the accuracy of the observation
2 An estimate where the original value is not available
3 An estimate where the original value is missing and cannot be retrieved
4 An unreliable estimate (used in radiation QC)
5 A correction (a reported value is assumed to be in error)
6 Value reverted to original
7 Original value verified by observer
8 Original value verified by quality control analyst

qc_query (code_id 8024) gives information about the original (as reported) value.

0 Original value is/was not queried, or no information available
1 Failed SDB or MetDB QC check
2 Failed MIDAS validation
3 Failed climate QC marine position or movement check
4 Failed climate QC range check
5 Failed climate QC internal consistency check
6 Failed climate QC sequence check
7 Failed climate QC areal check

qc_level (code_id 8025) indicates the climate QC stage reached. Level 0 - no processing; Level 9 - normal processing complete. The number of values between will vary with the element, and with time as QC procedures change. This item will not indicate the application or otherwise of MIDAS validation checks, nor whether visual checks of data have been carried out, e.g. of data on forms.

0 Initial climate QC program not run
1 Initial climate QC program has run
2 Initial QC queries processed
3 Spare
4 Further range or internal consistency or sequence checks job(s) run and queries processed
5 Spare
6 Final (or only) areal or buddy job run and queries processed
7 Spare
8 Final (or only) monthly job(s) run and queries processed
9 Normal QC complete

`met_element_name_j`
This attribute is a single character code which either describes the method of measurement, or further qualifies the meteorological values. The meaning of any value depends on the element being qualified. The values allocated to this item are also documented in the `code_detail` table of MIDAS.

Cloud (code_id 8004)

NULL
A Not originally reported in 1949 codes
B Nominal cloud height
C Height measured
D Amount originally measured in tenths
E Amount is derived from METAR code $N_sN_sN_s$
F Values from LCBR (ix = 5, 6 or 7)

Precipitation (code_id 8006)

NULL

A	Reading from autographic instrument
B	Amount due to dew, fog or frost
C	Amount due to snowfall
D	Rainfall converted from inches
E	Snow depth originally measured in inches

Pressure (code_id 8007)

	NULL
A	Measured by barometer not calibrated in millibars, or from barograph
B	Measured in whole units

Sea & Swell Waves (code_id 8008)

	NULL
A	Estimated, using 36 point compass
B	Estimated, using 36 point compass, descriptive codes
C	Estimated, using 36 point compass, not units of 0.5 metres/sec
D	Estimated, not using 36 point compass, using descriptive codes
E	Estimated, not using 36 point compass, using units of 0.5 metres/sec
F	Estimated, not using 36 point compass, not using units of 0.5 metres/sec
G	Measured, using units other than 0.5 metres/sec
H	Measured, using units of 0.5 metres/sec
J	Mixed wave measured
K	High precision data supplied in tenths of metre
L	measured using a shipbourne wave recorder
M	measured using buoy
N	measured using another method

Temperature (code_id 8010)

	NULL
A	Reading from autographic instrument
B	Original measured in degrees Fahrenheit
C	Original measured to nearest whole degree Fahrenheit
D	Original measured to nearest 0.5 degree Fahrenheit
E	Original measured to nearest whole degree Celsius
F	Original measured to nearest 0.5 degree Celsius
G	Iced Wetbulb (previously Spare)
H	Wet bulb not frozen, registering below 0.0 degrees Celsius
J	Wet bulb wick is assumed to have dried out
K	Wet bulb is derived from air temperature and dew point
L	Iced Wetbulb derived from air/dewpoint temp (previously Spare)
M	Aspirated
N	Aspirated and original in degrees Fahrenheit
P	Aspirated and original to nearest whole degree Fahrenheit
Q	Aspirated and original to nearest 0.5 degree Fahrenheit
R	Aspirated and original to nearest whole degree Celsius
S	Aspirated and original to nearest 0.5 degree Celsius
T	Max / min obtained from SAWS hourly values
U	Original temperature measured in 0.1 degrees F, and depth in inches
V	Original temperature measured in whole degrees F, and depth in inches
W	Original temperature measured in 0.1 degrees F, and depth at 24 inches
X	Original temperature measured in whole degrees F, and depth at 24 inches
Y	Original temperature measured in 0.1 degrees F, and depth at 48 inches

Z	Original temperature measured in whole degrees F, and depth at 48 inches

Visibility (code_id 8011)

	NULL
A	Measured rather than estimated

Weather (past and present) (code_id 8012)

	NULL
A	Derived from "AB" codes
B	Present weather is derived from code 4680
C	Present weather is derived from code 4678

Speed / Direction (code_id 8013)

	NULL
A	Speed originally estimated in metres/sec
B	Speed originally estimated in knots
C	Speed originally measured in metres/sec
D	Speed originally measured in knots
E	Speed originally measured in miles per hour
F	Converted from kilometres to whole knots
G	Mean wind derived from run of wind - unspecified units
H	Mean wind derived from run of wind, converted from kilometres to whole knots
J	Direction converted from 8 point compass
K	Direction converted from 16 point compass
L	Direction converted from 32 point compass
M	Speed originally Beaufort scale

Location (code_id 8014)

	NULL
A	Stationary
B	Position reported to within 30 minutes
C	Position reported to within 10 minutes
D	Position reported to within 5 minutes
E	Position reported from dead reckoning from previous position
F	OWS not in area
G	Observation time calculated from Local Apparent Time

Sea Temperature (code_id 8015)

	NULL
a	Original measured to nearest whole degree C using condenser inlet
b	Original measured to nearest whole degree C using trailing thermistor
c	Original measured to nearest whole degree C using hull contact sensor
d	Original measured to nearest whole degree C using through hull sensor
e	Original measured to nearest whole degree C using radiation thermometer
f	Original measured to nearest whole degree C using bait tanks thermometer
g	Original measured to nearest whole degree C method unknown
A	Original measured in 0.1 degree F using bucket
B	Original measured in 0.1 degree F not using bucket
C	Original measured to nearest 0.5 degree F using bucket
D	Original measured to nearest 0.5 degree F not using bucket
E	Original measured to nearest whole degree F using bucket
F	Original measured to nearest whole degree F not using bucket

G	Original measured in 0.1 degree C using bucket
H	Original measured in 0.1 degree C not using bucket
I	Original measured in 0.1 degree C using condenser inlet
J	Original measured in 0.1 degree C using trailing thermistor
K	Original measured in 0.1 degree C using hull contact sensor
L	Original measured in 0.1 degree C using through hull sensor
M	Original measured in 0.1 degree C using radiation thermometer
N	Original measured in 0.1 degree C using bait tanks thermometer
O	Original measured in 0.1 degree C method unknown
P	Original measured to nearest 0.5 degree C using bucket
Q	Original measured to nearest 0.5 degree C not using bucket
R	Original measured in 0.5 degree C using condenser inlet
S	Original measured in 0.5 degree C using trailing thermistor
T	Original measured in 0.5 degree C using hull contact sensor
U	Original measured in 0.5 degree C using through hull sensor
V	Original measured in 0.5 degree C using radiation thermometer
W	Original measured in 0.5 degree C using bait tanks thermometer
X	Original measured in 0.5 degree C method unknown
Y	Original measured to nearest whole degree C using bucket
Z	Original measured to nearest whole degree C not using bucket

14.5.1 Static Values - State Indicators

Observation sub-types can contain attributes that store information about the quality control that has been applied. These take the format met_element_name_q and met_element_name_j. The attribute name depends on the observation sub-type.

`met_element_name_q`
This attribute is a five digit number (of the form MESQL), where each digit describes one aspect of the quality of a meteorological element.

M	qc_marker	A combination of two flags to indicate accumulations, and/or a trace of rainfall
E	qc_estimate	Indicates 1 of up to 8 remarks about an estimate
S	qc_status	Indicates 1 of up to 8 possible descriptions of the value of the element (e.g. reason for suspecting a value)
Q	qc_query	Indicates 1 of up to 8 statements about the original value (e.g. a reason for correcting a data value)
L	qc_level	Indicates which of 10 possible stages of QC has been reached

qc_marker (code_id 8021) when set, indicates a precipitation trace or an accumulated value (e.g. rainfall or maximum temperature over more than one day; maximum gust speed over more than one hour).

0	NULL
1	Trace of precipitation
2	Accumulation
-2	Accumulated trace of precipitation

qc_estimate (code_id 8022) provides information about an estimated or corrected value. In this context an "estimate" is a value either derived by the QC teams where the observer or the automatic instrument has not provided a value, or has been provided retrospectively from the observing station after the original report was ingested with the element in question missing.

0	Value is not an estimate or correction, or information on the estimate/correction is not available
1	Estimate/correction derived automatically from a program with no manual intervention

2 Estimate/corrected value has been set manually (with or without assistance from a program)
3 Estimate/correction has been obtained retrospectively from the observer/station
4 Precipitation estimate/correction has been derived from a snow/rain equivalence, or trace estimate/correction has been set for consistency with present weather
5 Correction has been obtained by changing units of measurement
6 Correction has been obtained by applying a systematic adjustment
7 Measurement impossible, because of snow, etc.

qc_status (code_id 8023) indicates the status of the data value.

0 Observed and not suspect
1 Observed and suspect (i.e. has failed the latest QC check), or there are strong grounds for suspecting the accuracy of the observation
2 An estimate where the original value is not available
3 An estimate where the original value is missing and cannot be retrieved
4 An unreliable estimate (used in radiation QC)
5 A correction (a reported value is assumed to be in error)
6 Value reverted to original
7 Original value verified by observer
8 Original value verified by quality control analyst following supplementary investigation

qc_query (code_id 8024) gives information about the original (as reported) value.

0 Original value is/was not queried, or no information available
1 Failed SDB or MetDB QC check
2 Failed MIDAS validation
3 Failed climate QC marine position or movement check
4 Failed climate QC range check
5 Failed climate QC internal consistency check
6 Failed climate QC sequence check
7 Failed climate QC areal check

qc_level (code_id 8025) indicates the climate QC stage reached. Level 0 - no processing; Level 9 - normal processing complete. The number of values between will vary with the element, and with time as QC procedures change. This item will not indicate the application or otherwise of MIDAS validation checks, nor whether visual checks of data have been carried out, e.g. of data on forms.

0 Initial climate QC program not run
1 Initial climate QC program has run
2 Initial QC queries processed
3 Spare
4 Further range or internal consistency or sequence checks job(s) run and queries processed
5 Spare
6 Final (or only) areal or buddy job run and queries processed
7 Spare
8 Final (or only) monthly job(s) run and queries processed
9 Normal QC complete

`met_element_name_j`
This attribute is a single character code which either describes the method of measurement, or further qualifies the meteorological values. The meaning of any value depends on the element being qualified. The values allocated to this item are also documented in the `code_detail` table of MIDAS.

Cloud (code_id 8004)

NULL
A Not originally reported in 1949 codes

B	Nominal cloud height
C	Height measured
D	Amount originally measured in tenths
E	Amount is derived from METAR code $N_sN_sN_s$
F	Values from LCBR (ix = 5, 6 or 7)

Precipitation (code_id 8006)

	NULL
A	Reading from autographic instrument
B	Amount due to dew, fog or frost
C	Amount due to snowfall
D	Rainfall converted from inches
E	Snow depth originally measured in inches

Pressure (code_id 8007)

	NULL
A	Measured by barometer not calibrated in millibars, or from barograph
B	Measured in whole units

Sea & Swell Waves (code_id 8008)

	NULL
A	Estimated, using 36 point compass
B	Estimated, using 36 point compass, descriptive codes
C	Estimated, using 36 point compass, not units of 0.5 metres/sec
D	Estimated, not using 36 point compass, using descriptive codes
E	Estimated, not using 36 point compass, using units of 0.5 metres/sec
F	Estimated, not using 36 point compass, not using units of 0.5 metres/sec
G	Measured, using units other than 0.5 metres/sec
H	Measured, using units of 0.5 metres/sec
J	Mixed wave measured
K	High precision data supplied in tenths of metre
L	measured using a shipbourne wave recorder
M	measured using buoy
N	measured using another method

Temperature (code_id 8010)

	NULL
A	Reading from autographic instrument
B	Original measured in degrees Fahrenheit
C	Original measured to nearest whole degree Fahrenheit
D	Original measured to nearest 0.5 degree Fahrenheit
E	Original measured to nearest whole degree Celsius
F	Original measured to nearest 0.5 degree Celsius
G	Iced Wetbulb (previously Spare)
H	Wet bulb not frozen, registering below 0.0 degrees Celsius
J	Wet bulb wick is assumed to have dried out
K	Wet bulb is derived from air temperature and dew point
L	Iced Wetbulb derived from air/dewpoint temp (previously Spare)
M	Aspirated
N	Aspirated and original in degrees Fahrenheit
P	Aspirated and original to nearest whole degree Fahrenheit
Q	Aspirated and original to nearest 0.5 degree Fahrenheit

R	Aspirated and original to nearest whole degree Celsius
S	Aspirated and original to nearest 0.5 degree Celsius
T	Max / min obtained from SAWS hourly values
U	Original temperature measured in 0.1 degrees F, and depth in inches
V	Original temperature measured in whole degrees F, and depth in inches
W	Original temperature measured in 0.1 degrees F, and depth at 24 inches
X	Original temperature measured in whole degrees F, and depth at 24 inches
Y	Original temperature measured in 0.1 degrees F, and depth at 48 inches
Z	Original temperature measured in whole degrees F, and depth at 48 inches

Visibility (code_id 8011)

	NULL
A	Measured rather than estimated

Weather (past and present) (code_id 8012)

	NULL
A	Derived from "AB" codes
B	Present weather is derived from code 4680
C	Present weather is derived from code 4678

Speed / Direction (code_id 8013)

	NULL
A	Speed originally estimated in metres/sec
B	Speed originally estimated in knots
C	Speed originally measured in metres/sec
D	Speed originally measured in knots
E	Speed originally measured in miles per hour
F	Converted from kilometres to whole knots
G	Mean wind derived from run of wind - unspecified units
H	Mean wind derived from run of wind, converted from kilometres to whole knots
J	Direction converted from 8 point compass
K	Direction converted from 16 point compass
L	Direction converted from 32 point compass
M	Speed originally Beaufort scale

Location (code_id 8014)

	NULL
A	Stationary
B	Position reported to within 30 minutes
C	Position reported to within 10 minutes
D	Position reported to within 5 minutes
E	Position reported from dead reckoning from previous position
F	OWS not in area
G	Observation time calculated from Local Apparent Time

Sea Temperature (code_id 8015)

	NULL
a	Original measured to nearest whole degree C using condenser inlet
b	Original measured to nearest whole degree C using trailing thermistor
c	Original measured to nearest whole degree C using hull contact sensor
d	Original measured to nearest whole degree C using through hull sensor
e	Original measured to nearest whole degree C using radiation thermometer

f	Original measured to nearest whole degree C using bait tanks thermometer
g	Original measured to nearest whole degree C method unknown
A	Original measured in 0.1 degree F using bucket
B	Original measured in 0.1 degree F not using bucket
C	Original measured to nearest 0.5 degree F using bucket
D	Original measured to nearest 0.5 degree F not using bucket
E	Original measured to nearest whole degree F using bucket
F	Original measured to nearest whole degree F not using bucket
G	Original measured in 0.1 degree C using bucket
H	Original measured in 0.1 degree C not using bucket
I	Original measured in 0.1 degree C using condenser inlet
J	Original measured in 0.1 degree C using trailing thermistor
K	Original measured in 0.1 degree C using hull contact sensor
L	Original measured in 0.1 degree C using through hull sensor
M	Original measured in 0.1 degree C using radiation thermometer
N	Original measured in 0.1 degree C using bait tanks thermometer
O	Original measured in 0.1 degree C method unknown
P	Original measured to nearest 0.5 degree C using bucket
Q	Original measured to nearest 0.5 degree C not using bucket
R	Original measured in 0.5 degree C using condenser inlet
S	Original measured in 0.5 degree C using trailing thermistor
T	Original measured in 0.5 degree C using hull contact sensor
U	Original measured in 0.5 degree C using through hull sensor
V	Original measured in 0.5 degree C using radiation thermometer
W	Original measured in 0.5 degree C using bait tanks thermometer
X	Original measured in 0.5 degree C method unknown
Y	Original measured to nearest whole degree C using bucket
Z	Original measured to nearest whole degree C not using bucket

14.5.2 Static Values - State Indicators

A state indicator is an attribute of each table that is used to describe the current stage in the life of a particular record, from creation to deletion.

State Indicator settings and their meanings.

1001	Normal ingestion of observation at creation
1002	Normal ingestion of a multi level observation such as upper air at creation
1003	Addition of observation level
1004	Receive a COR before normal observation received
1005	Receive a COR before normal multi level observation received
1006	Receive a COR to observation level
1007	Addition of a missing value
1008	Receive a COR after the observation received but before QC started
1009	Recceive a COR to an observation level after normal receipt but before QC started
1010	Start of QC ob extracted for QC checks
1011	QC level raised on Version 1
1012	CreateVersion 0 First Qc amend to an attribute other than just change of qc level
1013	Version 0 exists with no version 1
1014	Version 1 exists as apportioned/corrected data with no version 0

Working with Version 1 as corrected data

1022	Version1 Creation Version 0 is frozen as original data state indicator 1012
1023	Version 1 of multi level ob
1024	QC amend to Version 1 observation multi level
1025	Change to qc level in Version 1
1026	Receive subsequent qc amendments
1027	Decision to Archive
1028	Archive observation
1029	COR of Key item- pre QC - mark for deletion

14.5.3 Static Values - Codes In code_detail Table

The `code_detail` table contains decodes of various WMO and other code tables that are used for values stored in MIDAS.

code_id	code_name	Description
0200	0200	WMO code 0200 Characteristic of pressure tendency.
0265	0265	WMO code 0265 Type of measuring equipment used.
0439	0439	WMO code 0439 Ice of land origin.
0500	0500	WMO code 0500 Genus of cloud.
0509	0509	WMO code table 0509 Clouds of genera Cirrus, Cirrocumulus and Cirrostratus.
0513	0513	WMO code table 0513 Clouds of the genera Stratocumulus, Stratus, Cumulus, etc.
0515	0515	WMO code 0515 Clouds of the genera Altocumulus, Altostratus, etc.
0639	0639	WMO code 0639 Concentration or arrangement of sea ice.
0700	0700	WMO code 0700 Direction or bearing in one figure.
0739	0739	WMO code 0739 True bearing of principal ice edge.
0901	0901	WMO code 0901 State of ground without snow or measureable ice cover.
0975	0975	WMO code 0975 State of ground with snow or measureable ice cover.
1400	1400	WMO code 1400 Time of observations used to compute mean values.
1600	1600	WMO code 1600 Height above surface of base of lowest cloud seen.
1751	1751	WMO code 1751 Ice accretion on ships.
2262	2262	WMO code 2262 Indicator for digitization.
2263	2263	WMO code 2263 Method of salinity/depth measurement.
2264	2264	WMO code 2264 Duration and time of current measurement.
2265	2265	WMO code 2265 Period of current measurement (drift method).
2266	2266	WMO code 2266 Indicator for the method of current measurement.
2267	2267	WMO code 2267 Method of removing ship velocity and motion from current.
2700	2700	WMO code 2700 Cloud cover.
3363	3363	WMO code 3363 Quality of measurement.
3551	3551	WMO code 3551 Rate of ice accretion on ships.
3570	3570	WMO code 3570 Amount of precipitation, water equiv., or deposit
3590	3590	WMO code 3590 Amount of precipitation during period preceding the ob.
3739	3739	WMO code 3739 Stage of ice development.
4451	4451	WMO code 4451 Ship's average speed made good during 3 hours before ob.
4561	4561	WMO code 4561 Past weather.

4677	4677	WMO code 4677 Present weather reported from a manned station.
4680	4680	WMO code 4680 Present weather reported from an automatic weather
5239	5239	WMO code 5239 Present ice situation and trend of conditions over 3 hrs.
6000	6000	NCM table 24 Day of thunder.
6001	6001	NCM table 23 Day of hail, ice, etc.
6002	6002	NCM table 27 Day of snow or sleet.
6003	6003	NCM table 25 Day of fog.
6004	6004	NCM table 26 Day of gale.
6005	6005	NCM table 22 State of concrete.
6006	6006	CDB flag Snow lying.
0 08 001	0 08 001	BUFR code 0 08 001 Vertical sounding significance.
0 20 062	0 20 062	BUFR code 0 20 062 State of ground.
8004	CLOUD	MIDAS met element measurement code - Cloud amount and height.
8006	PRECIPITATION	MIDAS met element measurement code - Precipitation.
8007	PRESSURE	MIDAS met element measurement code - Pressure.
8008	SWELL_WAVES	MIDAS met element measurement code - Sea state, Waves, Swell.
8010	TEMPERATURE	MIDAS met element measurement code - Temperature, Wet-bulb.
8011	VISIBILITY	MIDAS met element measurement code - Visibility.
8012	WEATHER	MIDAS met element measurement code - Weather
8013	WIND_SPEED_DIR	MIDAS met element measurement code - Direction and speed.
8014	LOCATION	MIDAS met element measurement code - Location.
8015	SEA_TEMPERATURE	MIDAS met element measurement code - Sea temperature
8016	RLTV_HUM	MIDAS met element measurement code - Relative Humidity.
8021	M	MIDAS met element quality code - QC marker for trace or accumulation.
8022	E	MIDAS met element quality code - QC estimate or correction
8023	S	MIDAS met element quality code - Reason for suspecting a value
8024	Q	MIDAS met element quality code - Check level
8025	L	MIDAS met element quality code - Level of Quality control.
8030	8030	MIDAS OB_RCPT_CODE, i.e. Ob reception method code

Document responsibilities

	Role Title	Name
Owner	Observations Quality Manager	[redacted]
Owner	Senior IT Practitioner	[redacted]
Author	Operational Meteorologist	[redacted]
Reviewer	Senior IT Practitioner	[redacted]

Document history

Document Register:

Observations Quality Management Team Register

Document identity and location:

OBQM_DEV_The_MIDAS_Handbook.pdf

Version	Issue date	Review due	Reviewer	Change description
1.0	03/11/2016	05/12/2017	[redacted]	Created from Metnet Content
1.0	05/12/2017	05/12/2018	[redacted]	Moved to ISO template
1.0	24/05/2018	24/05/2019	[redacted]	Reviewed and no changes
1.1	19/07/2018	24/05/2019	[redacted]	Added flag 8 to qc_status code at section 14.5.
1.2	09/08/2018	24/05/2019	[redacted]	Added pollen data table in section 6 and 9.2 and lying_snow_ht to weather_drnl_ob table in section 9.2.
1.2	09/09/2019	09/09/2020	[redacted]	Reviewed and no changes