

INTRODUCTION TO THE SQL LANGUAGE USED IN MIST

SQL is a basic database language used to get data from a database. The language and syntax is very simple to learn and it is a very powerful way to mine your data. While SQL as a language is simple, like with all languages, either human or computer, the combination of the words in the language can create very complex patterns and requests. So it is with SQL. This document is not intended to teach all the ways the SQL language can be combined, nor is it a comprehensive list of all language features. The intent here is to provide a basic understanding of SQL so that the database procedures we will create in MIST are somewhat demystified. SQL is simple English, and it can be read as such. Any SQL Query you can think of as a sentence. In fact, the words “Query” and “Sentence” are interchangeable in this document. As you go through this tutorial you should start to see more about how SQL can be read and perhaps even write some of your own.

1. Most Basic Language Words: **SELECT** and **FROM**

There are two key words in SQL that are required for any SQL query. These are **SELECT** and **FROM**.

SELECT “selects” items from a table, and **FROM** is from where those items are to be selected. **FROM** usually refers to a table in the database, but it can refer to other items as well (more on this later, but for **now** assume anything that follows **FROM** is a table).

Example A1

Between the key words **SELECT** and **FROM** in an SQL Query is always found a list of what you want to select from the table of choice. That is, lets say you want to select from the table `GROUND_PATROL_OBSERVATIONS`. This table has many field of data, but for this example, we will just select one field, the `OBSERVATION` column.

- We want can write this query sentence in a standard English sentence, where the relevant SQL text is in **bold**:

“**select** the field **observation** from the table **ground_patrol_observations**”

- In the above example the words in **bold** are all we need to retain to create an SQL sentence. We do not need the word “table” since MIST already knows that **ground_patrol_observations** is a table. The sentence above then simply becomes our SQL query of:

```
SELECT OBSERVATION FROM GROUND_PATROL_OBSERVATIONS
```

Note: for SQL in MIST, all the SQL language words should be **CAPITALIZED**

- The results of the query then are:

```
OBSERVATION
```

```
Gaur
```

```
Gaur
```

```
Position
```

```
Position
```

```
Gaur... and so on
```

Example A2

- If we want to return more fields, we just add them as a comma delimited text list. For example, our sentence above can become:

“select the fields `observation`, `observation_code` from the table `ground_patrol_observations`”

- Our SQL sentence then simply becomes:

```
SELECT OBSERVATION, OBSERVATION_CODE FROM GROUND_PATROL_OBSERVATIONS
```

- The results of the query then are:

OBSERVATION	OBSERVATION_CODE
Gaur	Tracks
Gaur	Tracks
Position	Position
Position	Position
Gaur	Tracks
Gaur	Tracks

Example A3

There is a special character in SQL that means select “all the fields”. This character is the star character “*”.

- We want to write out the following sentence in a standard English sentence:

“select all the fields from the table `ground_patrol_observations`”

- The sentence above then simply becomes our SQL query of:

```
SELECT * FROM GROUND_PATROL_OBSERVATIONS
```

- The results of the query are then:

PA_ID	GP_ID	PATROL_DAY	WAYPOINT	WAYPOINT_EASTING	WAYPOINT_NORTHING
HKK	1	1	1	1183257600	187416288
HKK	1	1	2	1183322240	187519776
HKK	1	1	3	1183244800	187583264
HKK	1	1	4	1183304320	187732592

2. How to Limit Data Records using the WHERE Clause

The SQL key word **WHERE** is used to limit the results of your query to a specific subset of data.

Example B1:

We want to limit our query in example A2 to “Gaur Tracks” only. So we want to limit the records we get back from our query by 2 items (1) Gaur and (2) Tracks. We do not want to retrieve any other data like Position. We do this by using a **WHERE** clause in our SQL query.

- We can write this new **WHERE** query sentence again in a standard English sentence:

“**select** the fields **observation**, **observation_code** from the table **ground_patrol_observations** where the field **observation** is equal to 'Gaur' and the field **observation_code** is equal to 'Tracks'”

- In the above example the words in **bold** are again what we need to keep. Notice also in bold is the word “**and**”. We also need to replace the phrase “*is equal to*” with the simple “**=**” sign. The sentence above then simply becomes our SQL query of:

```
SELECT OBSERVATION, OBSERVATION_CODE
FROM GROUND_PATROL_OBSERVATIONS
WHERE OBSERVATION='Gaur' AND OBSERVATION_CODE='Tracks'
```

Note: We have started to break up the lines of the SQL into rows, this makes the SQL easier to read. Also, note that 'Gaur' and 'Tracks' are in single quotes and are not fully capitalized, only the first character. This is because only the first characters in the database as data are capitalized. Any text in a **WHERE** clause you wish to search for must match exactly to the values in the database. For example, using 'tracks' will not work as the “T” is not capitalized. As an exercise, try using a small “t” as “tracks” and observe the results.

- The results of the query then are simply:

OBSERVATION	OBSERVATION_CODE
Gaur	Tracks
Gaur	Tracks
Gaur	Tracks
Gaur	Tracks

3. How to Link Data Between Tables using JOIN

MIST has many tables and many of these tables are what are called “dependent” on other tables. A dependent table has more, related data in another table. This design is one of database efficiency. For example, there is a **GROUND_PATROLS** table in MIST. Each record in this table is for only **one** patrol. The observations seen on any one patrol are stored in the **GROUND_PATROL_OBSERVATIONS** table. There are **many** patrol observations in the **GROUND_PATROL_OBSERVATIONS** table for any one patrol in the **GROUND_PATROLS** table. Thus there is what is called a “one to many” relation between the **GROUND_PATROLS** and the **GROUND_PATROL_OBSERVATIONS** tables. The SQL key word **JOIN** is used to link two or more tables so the data is shown **as if** the data were from a single table. The single data from the

GROUND_PATROLS is repeated in the query output only in virtual memory in the computer, but not actually in the database or on the hard disk; this is what makes the storage of data using related tables so efficient and redundant or repetitive data is not actually stored.

Example C1:

We want to combine three tables in MIST because we want the data for the park ID (**PA_ID**), the ground patrol type (**GPTYPE**) of the patrol, the patrol day (**PATROL_DAY**) and the rest time (**REST_TIME**) of the patrol. These tables are:

- 1) **GROUND_PATROLS** (stores each patrol reference e.g.: GPTYPE)
- 1) **GROUND_PATROL** (stores references to different day of the same patrol e.g.: PATROL_DAY and REST_TIME)
- 3) **GROUND_PATROL_OBSERVATIONS** (stores the actual waypoints and observations at each waypoint).

We use the **JOIN** key word to combine tables. Joins have other key words that are used with them, such as “left”, “right” and “outer” but for now we will just simply use **JOIN** by itself.

- We can write this new **JOIN** query sentence again in a standard English sentence:

select the fields **a.pa_id, a.gptype, b.patrol_day, b.rest_time, c.observation, c.observation_code**
from the table **ground_patrol a**
 with a **join** to the **table ground_patrols b**
 by linking **on** (common fields with ground_patrol)
 with a **join** to the **table ground_patrol_observations c**
 by linking **on** (common fields with ground_patrols)

- In the above example the words in **bold** are again what we need to keep. Also added after each table name is a small character (a,b and c) following each table and prefixing each of the field names. This is so we (and SQL) can know from what table each field originates. There is one trick in converting the simple English to SQL, and that is to know what are the “common fields” referred to in the sentence above. In this case you need to either know the common fields or use an appropriate database program to look them up. The SQL syntax for the “common fields” list is shown below.

```

SELECT A.PA_ID,A.GPTYPE,B.PATROL_DAY,B.REST_TIME,C.OBSERVATION,
        C.OBSERVATION_CODE
FROM GROUND_PATROLS A
JOIN GROUND_PATROL B
      ON (A.PA_ID=B.PA_ID AND A.GP_ID=B.GP_ID)
JOIN GROUND_PATROL_OBSERVATIONS C
      ON (B.PA_ID=C.PA_ID AND B.GP_ID=C.GP_ID)
  
```

- The results of the query then is:

PA_ID	GPTYPE	PATROL_DAY	REST_TIME	OBSERVATION	OBSERVATION_CODE
HKK	F	1	0	Gaur	Tracks
HKK	F	2	0	Gaur	Tracks
HKK	F	3	0	Position	Position
HKK	F	1	0	Position	Position

Finally, To know what tables link to other tables, use the MIST Browse Tables tool. the Key Fields tab will show a visual diagram of the related parent tables in MIST from any table you select in the main table list on the left side of the Browse Tables window:

