

Syllabus Content:

1.8.3 Data Definition Language (DDL) and Data Manipulation Language (DML)

-  show understanding that DBMS software carries out:
 - all creation/modification of the database structure using its DDL
 - query and maintenance of data using its DML
-  show understanding that the industry standard for both DDL and DML is Structured Query Language (SQL)
 - show understanding of a given SQL script
 - write simple SQL (DDL) commands using a sub-set of commands for:
 - creating a database (CREATE DATABASE)
 - creating a table definition (CREATE TABLE)
 - changing a table definition (ALTER TABLE)
 - adding a primary key or foreign key to a table (ADD PRIMARY KEY)
 - write a SQL script for querying or modifying data (DML) which are stored in (at most two) database tables
 - Queries:
 - SELECT, FROM, WHERE, ORDER BY, GROUP BY, INNER JOIN
 - Data maintenance:
 - INSERT INTO, DELETE FROM, UPDATE

Queries

Databases allow us to store and filter data to find specific information. A database can be queried using a variety of methods, although this depends on the software you are using.

Databases can use query languages or graphical methods to interrogate the data.

Query language

Query language is a **written language used only to write specific queries**. This is a powerful tool as the user can define precisely what is required in a database. SQL is a popular query language used with many databases.

Query by example (QBE)

QBE allows the user to **create queries based on a template**, usually a set of filters presented in a graphical form. If you are using database software it might have an option to connect blocks and set the filters you want. The system presents a blank record and lets you specify the fields and values that define the query.

Database management software like MySQL, **Microsoft Access** and **Oracle** have front-end graphical interfaces which make it easier to run QBE queries.

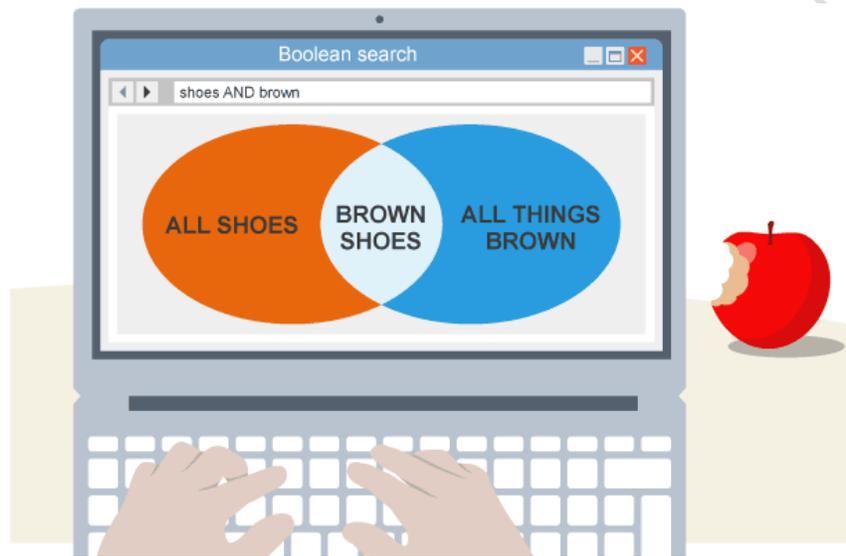
Boolean operators

In a database we often need to filter the data to group certain results. Boolean **operators** are used to filter databases using **AND**, **OR** or **NOT**. They can search multiple fields at the same time to help us retrieve the data that we need. They are used because they provide results which are 'true' or 'false'.

Search engines also make use of Boolean operators to filter results.

AND

AND is used to search records that contain one piece of information AND another.



For example, a database of clothing could be full of different types of items. Each item of clothing has lots of attributes including price, brand, colour, quantity and material.

You might want to search for brown shoes. A query for the words *brown AND shoes* would return results that contain the words brown and shoes.

In general, search engines treat the query *brown shoes* as *brown AND shoes*, which means that all results will contain both words, eg *brown trousers and red shoes for sale*.

OR

OR is used to search for records that contain EITHER one piece of information OR another.

It is used to request an alternative, for example *black shoes OR white shoes*. This would present results for **any shoes that were black or white**.

Most search engines use the OR function best if the search statements are defined by speech marks, eg "*brown shoes*" OR "*black jeans*" would show pages which either contain brown shoes or black jeans.

NOT

NOT is used to exclude results.

The query *shoes NOT brown* will return results that **contain the word shoes but NOT the word brown**.

Some search engines use a minus sign in front of the word, instead of NOT, eg *-brown*. In these cases, if you run a search for *Doctor Who Capaldi* this will include all results with all of the words *Doctor*, *Who* and *Capaldi*, but the search phrase *Doctor Who -Capaldi* return the same results, excluding all results for *Capaldi*.

Arithmetic operators

A query can also be performed using **arithmetic operators**. These help to make specific searches related to numerical data.

Functions of arithmetic operators

This table shows some arithmetic operators and their functions:

Operator	Meaning
=	Equals
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
<>	Not equal to

Using arithmetic operators

The table below shows some BBC TV programme listings:

ID	Title	Genre	Duration (mins)	Channel
----	-------	-------	-----------------	---------

ID	Title	Genre	Duration (mins)	Channel
01	EastEnders	Drama	30	BBC1
02	Dragons' Den	Entertainment	60	BBC2
03	The Voice	Entertainment	75	BBC3
04	Blue Peter	Children's	25	CBBC
05	Wild Brazil	Nature	60	BBC4
06	Match of the Day	Sport	80	BBC1
07	Dick and Dom	Comedy	10	CBBC

Queries are useful for searching for specific conditions. You might want to find entertainment programmes on BBC3. A query for these conditions would look like this:

```
SELECT * FROM Programmes  
WHERE Genre='Entertainment'  
AND Channel='BBC3';
```

This would return the programme 'The Voice'.

You may want a programme that is less than 20 minutes long or is a nature programme. A query for these conditions would look like this:

```
SELECT * FROM Programme  
WHERE Duration < 20  
OR Genre = 'Nature';
```

This would return the programmes "Dick and Dom" and "Wild Brazil".

SQL

SQL is a programming language used to search and query databases. SQL gives you the ability to customise your queries. It is often used within database programs.

Uses of SQL

SQLite and MySQL are popular open source database applications. If you use a blog site, it is very likely that MySQL has been running behind the scenes to store entries as records and allow you to add, edit and delete blog posts.

SQL databases can be used to create lots of applications for use on the internet. They are often used by large companies because they allow the data tables to be stored on secure servers. A SQL server simply stores the data for a database - it does not provide front-end features. Therefore, the user does not need to know that a database is working behind a website.

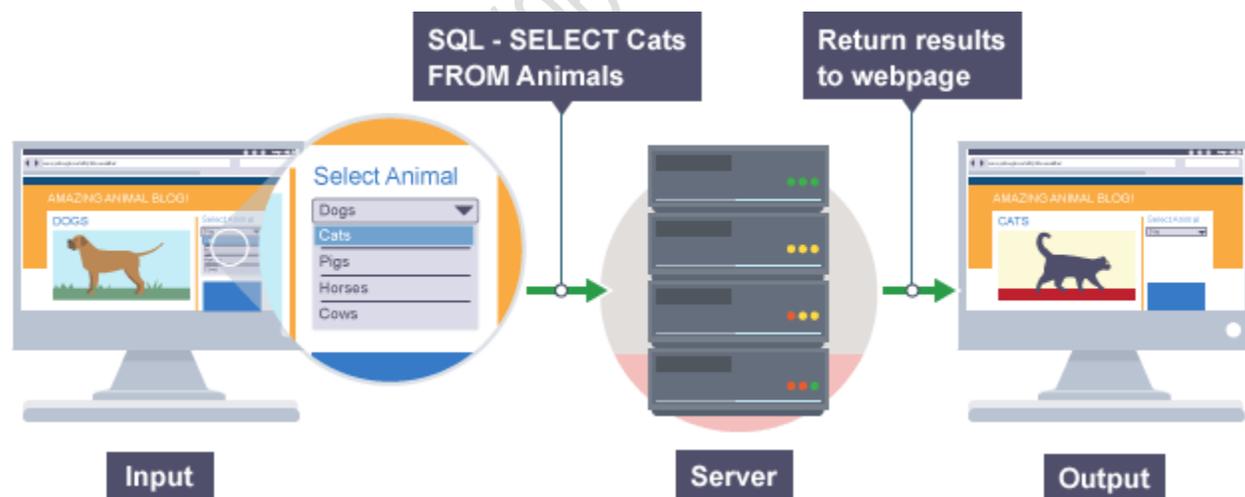
A SQL database is manipulated using SQL code. SQL has also been designed so that lots of users can access it at the same time - it has a high capacity for storage space.

Syntax

SQL is an important programming language and understanding how to use it is a very important skill.

The statements used in SQL code are quite similar to the words that you would normally use to find information. Because the statements are quite similar, it makes it easier to remember SQL syntax.

For example: "**SELECT** these fields **FROM** this table **WHERE** this is happening".



Working with SQL: Step 1

The following examples show how to use SQL for basic database functions. We will work through a series of steps to create this table, showing a record for a collection of BBC programmes:

ID	Title	Genre	Duration
01	EastEnders	Drama	30
02	Newsnight	Current affairs	50
03	The Voice	Entertainment	75
04	Blue Peter	Children's	25
05	Wild Brazil	Nature	60
06	Sherlock	Drama	90
07	Top Gear	Entertainment	60

Data Definition Language (DDL)

Data Definition Language (DDL) is the part of SQL provided for creating or altering tables. These commands only create the structure. They do not put any data into the database.

Creating a table

A table can be created in SQL code using the following template:

```
CREATE TABLE tablename  
(column1 datatype,  
column2 datatype,  
column3 datatype);
```

The creator of the table has to decide:

- the name of the table - "tablename"
- the title of each field - "column"

- the data type that is required for each field - "datatype"(eg character strings - 'varchar' or numbers 'int')

In the TV programmes example the table can be created using the following SQL code:

```
CREATE TABLE Programmes
(ID int(2),
Title varchar(20),
Genre varchar(20),
Duration int(3));
```

The data type **varchar** indicates that only character strings are allowed, and the number in brackets indicates the maximum number of digits for each data type.

These examples show that once the database has been created the tables can be created and the attributes defined. It is possible to define a primary key and a foreign key within the **CREATE TABLE** command but the **ALTER TABLE** command can be used as shown (it can also be used to add extra attributes).

```
CREATE DATABASE BandBooking;
CREATE TABLE Band (
BandName varchar2(25),
NumberOfMembers number(1));
ALTER TABLE Band ADD PRIMARY KEY (BandName) ;
ALTER TABLE Band-Booking ADD FOREIGN KEY (BandName
REFERENCES Band (BandName);
```

Data Manipulation Language (DML):

Data Manipulation Language (DML) is used when a database is first created, to populate the tables with data. It can then be used for ongoing maintenance. The following code shows a selection of the use of the commands:

```
INSERT INTO Band ('ComputerKidz',5);
INSERT INTO Band-Booking (BandName, BookingID)
VALUES ('ComputerKidz', '2016/023');
UPDATE Band
SET NumberOfMembers = 6;
DELETE FROM BandName
WHERE BandName = 'ITWizz';
```

Working with SQL: Step 2



You can query this table using SQL code.

ID	Title	Genre	Duration
01	EastEnders	Drama	30
02	Newsnight	Current affairs	50
03	The Voice	Entertainment	75
04	Blue Peter	Children's	25
05	Wild Brazil	Nature	60
06	Sherlock	Drama	90

You may decide that you wish to sort the programmes by duration. The SQL code needed would look like the example below:

```
SELECT Programmes.Duration, Programmes.Title  
FROM Programmes  
ORDER BY Programmes.Duration;
```

- the **SELECT** statement states which fields to look at - the Title and Duration fields
- the **FROM** statement states which table to look at - Programmes
- the **ORDER BY** statement sorts the Duration field in ascending order by default

This table shows the results from this query:

Duration	Title
25	Blue Peter
30	EastEnders
50	Newsnight
60	Wild Brazil

Duration	Title
75	The Voice
90	Sherlock

Working with SQL: Step 3

The SQL **WHERE** statement is used to isolate one record or several records with similar attributes.

ID	Title	Genre	Duration
01	EastEnders	Drama	30
02	Newsnight	Current affairs	50
03	The Voice	Entertainment	75
04	Blue Peter	Children's	25
05	Wild Brazil	Nature	60
06	Sherlock	Drama	90

The following code searches the Title field of the table to find the words 'The Voice'.

```
SELECT Programmes.ID, Programmes.Title, Programmes.Genre,
Programmes.Duration
FROM Programmes
WHERE ((Programmes.Title)="The Voice");
```

- the **WHERE** statement specifies which text to look for

This table shows the results from this query:

ID	Title	Genre	Duration
----	-------	-------	----------

ID	Title	Genre	Duration
03	The Voice	Entertainment	75

Alternatively, you could find all of the programmes which are less than 30 minutes long using this code:

...

```
WHERE ((Programmes.Duration)<30);
```

This table shows the results from this query:

ID	Title	Genre	Duration
04	Blue Peter	Children's	25

Wildcards

The wildcard uses the * symbol, and is used in place of any number of unknown characters. For example, the following code searches for all programmes with the letter i in the title:

...

```
WHERE ((Programmes.Title) LIKE "*i*");
```

This table shows the results from this query:

ID	Title	Genre	Duration
02	Newsnight	Current Affairs	50
03	The Voice	Entertainment	75
05	Wild Brazil	Nature	60

Adding and editing data with SQL

SQL can also be used to add and edit the data stored on an SQL server.

Adding records

To add new data you use the function **INSERT INTO**.

If a new BBC programme is created and you want to add it to the database, then you would need to use the **INSERT INTO** function followed by the **VALUES** separated by a comma:

```
INSERT INTO Programmes  
VALUES (07, "Top Gas", "Entertainment", 60);
```

Note that quotes surround string entries. Numbers do not have quotes.

After inputting this code the table would look like this:

ID	Title	Genre	Duration
01	EastEnders	Drama	30
02	Newsnight	Current affairs	50
03	The Voice	Entertainment	75
04	Blue Peter	Children's	25
05	Wild Brazil	Nature	60
06	Sherlock	Drama	90
07	Top Gas	Entertainment	60

Editing records

SQL also has the **UPDATE** function for editing data.

In this example, there was an error with the previous entry and you need to change the name from "Top Gas" to "Top Gear". You need to:

1. Identify the table to be updated using **UPDATE** - `UPDATE Programmes`
2. Identify what the field needs to be changed to using **SET** - `SET Programmes.Title = "Top Gear"`
3. Identify which record needs to be updated using **WHERE** - `WHERE Programmes.ID = 07`

In full, this is:

```
UPDATE Programmes  
SET Programmes.Title = "Top Gear"  
WHERE Programmes.ID = 07;
```

This will go to the **Programmes** table, find the programme with an **ID** of 07 and change the **Title** field to **Top Gear**. The amended table will look like this:

ID	Title	Genre	Duration
01	EastEnders	Drama	30
02	Newsnight	Current affairs	50
03	The Voice	Entertainment	75
04	Blue Peter	Children's	25
05	Wild Brazil	Nature	60
06	Sherlock	Drama	90
07	Top Gear	Entertainment	60

References:

<https://www.bbc.com/education/guides/z37tb9q/revision/8>