

# SQL

Thomas Boxall

1st February 2022

## 1 What is SQL

- SQL stands for Structured Query Language
- Is a language used to setup, search and manipulate databases
- Keywords such as **SELECT** must be in capitals
- Database languages are divided into DDL (*Data Definition Language*) and DML (*Data Manipulation Language*)
- DDL defines the database (eg, creates tables and links between them)
- DML is used for searching and manipulating those tables
- SQL is both.

## 2 Syntax

SQL Syntax is based off of statements. SQL keywords are not case sensitive, however the exam board are case sensitive, they should be written in upper case. SQL statements end with a semicolon.

### 2.1 SELECT

The **SELECT** statement is used to select data from a database. The data returned is stored in a result table, called the result-set. The following SQL code will display the contents of column1 and column2.

```
SELECT column1, column2, ...  
FROM table_name;
```

If you wanted to select all the fields of data in a table you would use a **\*** in the location of the columns which you want to display.

```
SELECT * FROM table_name;
```

### 2.2 WHERE

The **WHERE** statement is used to filter record. It is used to extract only those records that fulfill a specified condition.

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

### 2.3 LIKE

The **LIKE** operator is used in a **WHERE** clause to search for a specified pattern in a column. Often the **\%** wildcard and the **\*** wildcard are used in conjunction with the **LIKE** operator.

```
SELECT column1, column2, ...  
FROM table_name  
WHERE columnN LIKE pattern;
```

## 2.4 AND and OR

The WHERE clause can be combined with AND and OR operators. The AND and OR operators are used to filter records based on more than one condition:

- The AND operator displays a record if all the conditions separated by AND are TRUE.
- The OR operator displays a record if any of the conditions separated by OR is TRUE.

### AND Syntax

```
SELECT column1, column2, ...
FROM table_name
WHERE condition1 AND condition2 AND condition3 ...;
```

### OR Syntax

```
SELECT column1, column2, ...
FROM table_name
WHERE condition1 OR condition2 OR condition3 ...;
```

## 2.5 DELETE

The DELETE statement is used to delete existing records in a table

```
DELETE FROM table_name WHERE condition;
```

Be careful when deleting records, the WHERE clause in the statement is really important. If it is omitted, all records in the table will be deleted.

It is possible to delete all rows in a table without deleting the table. This means that the table structure, attributes and indexes will be intact.

```
DELETE FROM table_name;
```

## 2.6 INSERT INTO

The INSERT INTO statement is used to insert new records in a table. It is possible to write the INSERT INTO statement in two ways

- (a) Specify both the column names and the values to be inserted

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

- (b) If you are adding values for all the columns of the table, you do not need to specify the column names in the SQL query. However, make sure the order of the values is in the same order as the columns in the table

```
INSERT INTO table_name
VALUES (value1, value2, value3, ...);
```

## 2.7 DROP TABLE

The DROP TABLE statement is used to drop an existing table in a database.

```
DROP TABLE table_name;
```

## 2.8 INNER JOIN

The INNER JOIN keyword selects records that have matching values in both tables.

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```

It is also possible to join three tables together in the same way

```
SELECT Orders.OrderID, Customers.CustomerName, Shippers.ShipperName
FROM ((Orders
INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID)
INNER JOIN Shippers ON Orders.ShipperID = Shippers.ShipperID);
```

## 2.9 Wildcards

A wildcard character is used to substitute one or more characters in a string.  
The \* wildcard is used to represent everything.  
The % wildcard is used to represent zero or more characters.

## 2.10 UPDATE

The UPDATE statement is used to modify the existing records in a table.

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

## 2.11 SELECT DISTINCT

The SELECT DISTINCT statement is used to return only distinct (different) values. Inside a table, a column often contains many duplicate values; and sometimes you only want to list the different (distinct) values.

```
SELECT DISTINCT column1, column2, ...
FROM table_name;
```

## 2.12 ORDER BY

The ORDER BY keyword is used to sort the result-set in ascending or descending order. The ORDER BY keyword sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.

```
SELECT column1, column2, ...
FROM table_name
ORDER BY column1, column2, ... ASC|DESC;
```

## 2.13 NULL Values

A field with a NULL value is a field with no value. If a field in a table is optional, it is possible to insert a new record or update a record without adding a value to this field. Then, the field will be saved with a NULL value. To test for NULL values, SQL has special operators: IS NULL and IS NOT NULL.  
For checking if the record is NULL:

```
SELECT column_names
FROM table_name
WHERE column_name IS NULL;
```

For checking if the record is not NULL:

```
SELECT column_names
FROM table_name
WHERE column_name IS NOT NULL;
```

## 2.14 Functions

SQL has a number of functions which can be used to manipulate the data further.

### 2.14.1 MIN()

The MIN() function returns the smallest value of the selected column.

```
SELECT MIN(column_name)
FROM table_name
WHERE condition;
```

### 2.14.2 MAX()

The MAX() function returns the largest value of the selected column.

```
SELECT MAX(column_name)
FROM table_name
WHERE condition;
```

### 2.14.3 COUNT()

The COUNT() function returns the number of rows that matches a specified criterion.

```
SELECT COUNT(column_name)
FROM table_name
WHERE condition;
```

### 2.14.4 AVG()

The AVG() function returns the average value of a numeric column.

```
SELECT AVG(column_name)
FROM table_name
WHERE condition;
```

### 2.14.5 SUM()

The SUM() function returns the total sum of a numeric column.

```
SELECT SUM(column_name)
FROM table_name
WHERE condition;
```

## 2.15 IN

The IN operator allows you to specify multiple values in a WHERE clause. The IN operator is a shorthand for multiple OR conditions.

```
SELECT column_name(s)
FROM table_name
WHERE column_name IN (value1, value2, ...);
```

## 2.16 PRIMARY KEY

The PRIMARY KEY constraint uniquely identifies each record in a table. Primary keys must contain UNIQUE values, and cannot contain NULL values. A table can have only ONE primary key; and in the table, this primary key can consist of single or multiple columns (fields).

The following SQL creates a PRIMARY KEY on the "ID" column when the "Persons" table is created.

```
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int,
    PRIMARY KEY (ID)
);
```

## 2.17 FOREIGN KEY

The FOREIGN KEY constraint is used to prevent actions that would destroy links between tables. A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the PRIMARY KEY in another table. The table with the foreign key is called the child table, and the table with the primary key is called the referenced or parent table.

The following SQL creates a FOREIGN KEY on the "PersonID" column when the "Orders" table is created.

```
CREATE TABLE Orders (
    OrderID int NOT NULL,
    OrderNumber int NOT NULL,
    PersonID int,
    PRIMARY KEY (OrderID),
    FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)
);
```

### 3 References

Information and code samples taken from [www.w3schools.com/sql/](http://www.w3schools.com/sql/)