# ORMEDIAN RESEARCH INSTITUTE

## TOPIC:

## DATABASE

**SUBTOPIC:** INTRODUCTION TO SQL (PART 1)

## By:

**ADEKOLA OLUWASEUN O.**

***DECEMBER 20, 2022***

# CONTENTS COVERED

- *What SQL is*
- *What SQL is used for*
- *Is SQL Relational or Not relational?*
- *SQL Syntax*
- *SELECT and DISTINCT SELECT Statement*
- *Result Set*
- *Database Table look and what a table is*
- *Application of LIMIT and WHERE clause*
- *Application of AND, OR, NOT and other operators.*
- *Appendix*

# SQL

- *Short for Structured Query Language*

- *SQL is a standard language for accessing and manipulating databases.*

*It is considered as a domain-specific language used in programming an designed for managing data held in a relational database management system.*

# *WHAT SQL IS USED FOR*

- *SQL can execute queries against a database*
- *SQL can retrieve data from a database*
- *SQL can insert records in a database*
- *SQL can update records in a database*
- *SQL can delete records from a database*
- *SQL can create new databases*
- *SQL can create new tables in a database*
- *SQL can create stored procedures in a database*
- *SQL can create views in a database*
- *SQL can set permissions on tables, procedures, and views*

# RELATIONAL OR NON-RELATIONAL

- *A relational database is **a collection of information that organizes data points with defined relationships for easy access**.*
- *Another point is that it organizes Data into rows and columns in form of tables.*
- *Hence SQL is the basis of every Relational Database Management System*

# *SQL SYNTAX*

- *Generally SQL is not case sensitive. Meaning that select is the same as SELECT.*
- *It uses semicolon (;) as the separator for statements. Multiple statement to the same call to server can be separated using this semi-colon.*

- *SELECT UPDATE*
- *DELETE*
- *INSERT INTO*
- *CREATE DATABASE*
- *ALTER DATABASE*
- *CREATE TABLE*
- *ALTER TABLE*
- *DROP TABLE*
- *CREATE INDEX*
- *DROP INDEX*

# USING SELECT and DISTINC SELECT STATEMENTS

***SELECT**: Select statement is used to select data from the database. It can be used to perform the following: Selecting all fields, Selecting single column, Selecting multiple columns, Selecting a particular row*

***DISTINCT SELECT**: This is used to return unique values in a column. i.e it doesn't return repeated value twice.*

***COUNT DISTINCT:** This returns the number of the unique values in a column.*

# RESULT SET

*TABLE THAT IS RETURNED AS THE RESULT OF EXTRACTED DATA FROM THE ORIGINAL TABLE. For example, if we try to select a particular field from a table, say customers address field from table named customer, the returned result is called result set.*

# TABLE 1:

***This is a table stored in a database, and it contains some records of students from three different department in the faculty of Engineering.***
***The name given to this table in my database is Students. This table name and its cells will referenced throughout every implementation part of this module.***

| S/N | FirstName | LastName | MatricNo | Department | Level | EntryMode | Age | Sex |
|---|---|---|---|---|---|---|---|---|
| 1 | Oluwaseun | Adekola | 160211014 | ECE | 500 | UTME | 26 | Male |
| 2 | Abosede | Adedoyin | 180231032 | CPE | 300 | UTME | 23 | Female |
| 3 | Samuel | Adebayo | 180211044 | ECE | 300 | UTME | 24 | Male |
| 4 | Damilola | Osekita | 170221022 | MECH | 400 | DIRECT ENTRY | 25 | Female |
| 7 | Ayoola | Abiodun | 170221032 | MECH | 400 | DIRECT ENTRY | 27 | Male |
| 8 | Michael | Ogedengbe | 160231056 | CPE | 500 | UTME | 29 | Male |
| 9 | Taiwo | Adeoti | 180211054 | ECE | 300 | DIRECT ENTRY | 26 | Male |

# UNDERSTANDING WHAT A TABLE IS IN RDBMS

- *Every table is broken up into smaller entities called fields. The fields in the Student table consist of FirstName, MatricNo, Department, Level etc. A field is a column in a table that is designed to maintain specific information about every record in the table. A record is a horizontal entity in a table.*
- *A record, also called a row, is each individual entry that exists in a table. For example the student Table shown in **Table 1** of this presentation shows 9 records.*
- *A column is a vertical entity in a table that contains all information associated with a specific field in a table.*

# THE USE OF LIMIT CLAUSE IN SQL

*LIMIT : This is basically used in SQL to restrict how many rows are returned from a query Or simply say that it is used to specify the number of records to return. A table may have about 20 rows and we may need to output only the 6 rows out of all.*

*Let's assume the student table as an example, say we only need the first 5 row records and nothing more than that should be returned in our result-set.*

**CODE EXAMPLE**

***SELECT *FROM Students LIMIT 6;***

*The output from this will simply be the there entries from row 1 to row 6.*

# THE USE OF WHERE CLAUSE IN SQL

***WHERE :*** *This is a clause that is used when fetching data from a single a single table or joining with more tables together. It is basically used to specify conditions or extra criteria of data or rows that will be affected by a given specified SQL Statement.*

*It is used with other SQL statements such as SELECT, UPDATE and DELETE.*

# Using WHERE Clause with IN, LIKE, BETWEEN and other operators.

***Using IN clause with WHERE Clause**: this is used when there is need to return outputs from a cell or multiple cells which have different values. For example, department column has ECE, CPE and MECH.*

***CODE EXAMPLE***

```
SELECT *FROM Students WHERE Department IN ("ECE", "CPE");
```

***Using LIKE clause with WHERE clause:** To return specific pattern of value from a table. e.g if we need to output records of students whose lastname begins with letter "A", we will put "A%" after the LIKE clause.*

***CODE EXAMPLE***

```
SELECT *FROM Students WHERE LastName LIKE "A%";
```

***Using BETWEEN with WHERE clause**: This is used to generate range of results from a column. Let's say we want to generate a certain age range from students Age;*

***CODE EXAMPLE***

```
SELECT * FROM Students WHERE Age BETWEEN 20 AND 25;
```

***Using WHERE clause with Operators:** this works as assignment or as comparison means but it depends on the operator used. Operators such as <, >, =, <=, >=, !=, etc.*

***CODE EXAMPLE***

```
SELECT * FROM Students WHERE Age <= 27;
```

# Using AND operator in SQL

***AND:*** *this is used when there is need to fetch data from the table given that two different field (Columns) values have been specified in the condition.*

***Let's use the "Students" Table described in Table 1 of this presentation.***

*Let's say we need to fetch records  of students in ECE department whose EntryMode is DirectEntry*

**CODE EXAMPLE**

```
SELECT *FROM Students WHERE Department= "ECE" AND EntryMode="Direct Entry";
```

*We can also fetch just one or more field(column) record e.g MatricNo and LEVEL of Direct entry students in ECE department*

```
SELECT MatricNo , Level  FROM Students WHERE EntryMode = "Direct Entry" AND Department= "ECE";
```

# Using OR in SQL

***OR** : this is used when there is need to fetch data usually of different values from either two or more columns(field). For example: ASSUME we are to fetching records of students in either department of ECE, CPE or MECH as the output.*

## CODE EXAMPLE

```
SELECT *FROM Students WHERE Department= "ECE" OR Department="CPE" OR Department="MECH";
```

*Also,*

*Assume we need to fetch data from just one or more field(column) records say MatricNo and Level of students who are either in department of ECE or CPE or MECH*

```
SELECT MatricNo, Level  FROM Students WHERE EntryMode = "Direct Entry" AND (Department= "ECE" OR Department="CPE" OR Department = "MECH");
```

# Combining AND with OR Operators

*AND operator can be used with OR operator in cases where there is need to filter data in a specific way. It can be used to obtain records of only some categories of data.*

*Assume we need to obtain the records of only 500 Level students but who are either in ECE or MECH.*

CODE EXAMPLE:

***SELECT *FROM Students WHERE Level = 500 AND (Department="ECE" OR Department="MECH");***

# Using NOT operator in SQL

***NOT:*** *this operator is used with WHERE clause to filter records, basically for selecting rows for which statement is FALSE.*

*Lets assume we only need records of others students' department except from mechanical department by using the "Students table" as the original table. This will return a result-set that contains the records of other students that are not in mech department.*

CODE EXAMPLE:

***SELECT *FROM Students WHERE NOT Department = "MECH";***

# Using NOT with AND operator

***NOT*** *operator can be used with* ***AND*** *operator when there is need to return table records after we have exempted some records.*

*For example, we may wish to produce an output of the Students table such that records of only students of two different departments e.g. (CPE and ECE) are exempted from the records that will be in the output.*

CODE EXAMPLE:

```
SELECT *FROM Students WHERE NOT Department="CPE" AND NOT Department="ECE";
```

*We may also output particular fields(columns) of choice instead of the whole records :*

CODE EXAMPLE:

```
SELECT FirstName, LastName, MatricNo  FROM Students WHERE NOT Department = "CPE" AND NOT Department="ECE";
```

# *THANKS FOR VIEWING*

**NOTE:** ***Subsequent topics under SQL will be discussed in Part 2***

***Also, all Implementations as far as this presentation is concerned are supported by MySQL. All have been tested on XAMPP Server.***