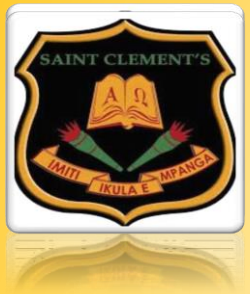
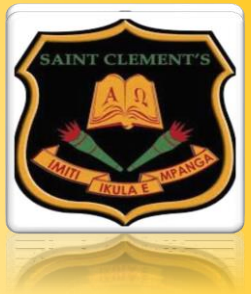


ST CLEMENT'S SECONDARY SCHOOL



PROGRAMMING IN STRUCTURED QUERY LANGUAGE (SQL)

SENIOR SECONDARY GRADE 12

OUTLINE

- *Introduction*
- *Database and SQL fundamentals*
- *SQL Language*
- *Installing and configuring MySQL and XAMPP*
- *Types of SQL commands*
- *Creating Database using DDL*
- *Data Manipulation using DML*
- *Displaying record set from Database*
- *SQL join Operations*
- *Creating views*

Introduction

- SQL stands for Structured Query Language.
- It is a standard, comprehensive language for database creation and manipulation.
- It is supported by DBMS. (Database Management System).
- Examples of DBMS
  - MySQL
  - MS Access
  - Oracle
  - Sybase
  - Informix

Database and SQL Fundamentals

- Data base is defined as a logically organised collection of related data.
- A **database** is a collection of related data.

Relational Database

- Is a database which is organised into tables.
- A table is made up of rows and columns known as fields.

PUPILS

First Name	Surname	Age	Gender	Town
Agrey	Sikaonga	16	Male	Mansa
Bukata	Ching'andu	15	Male	Kitwe
Gideon	Kapela	17	Male	Lusaka
Christine	Mundala	14	Female	Mazabuka

Structured Query Language (SQL)

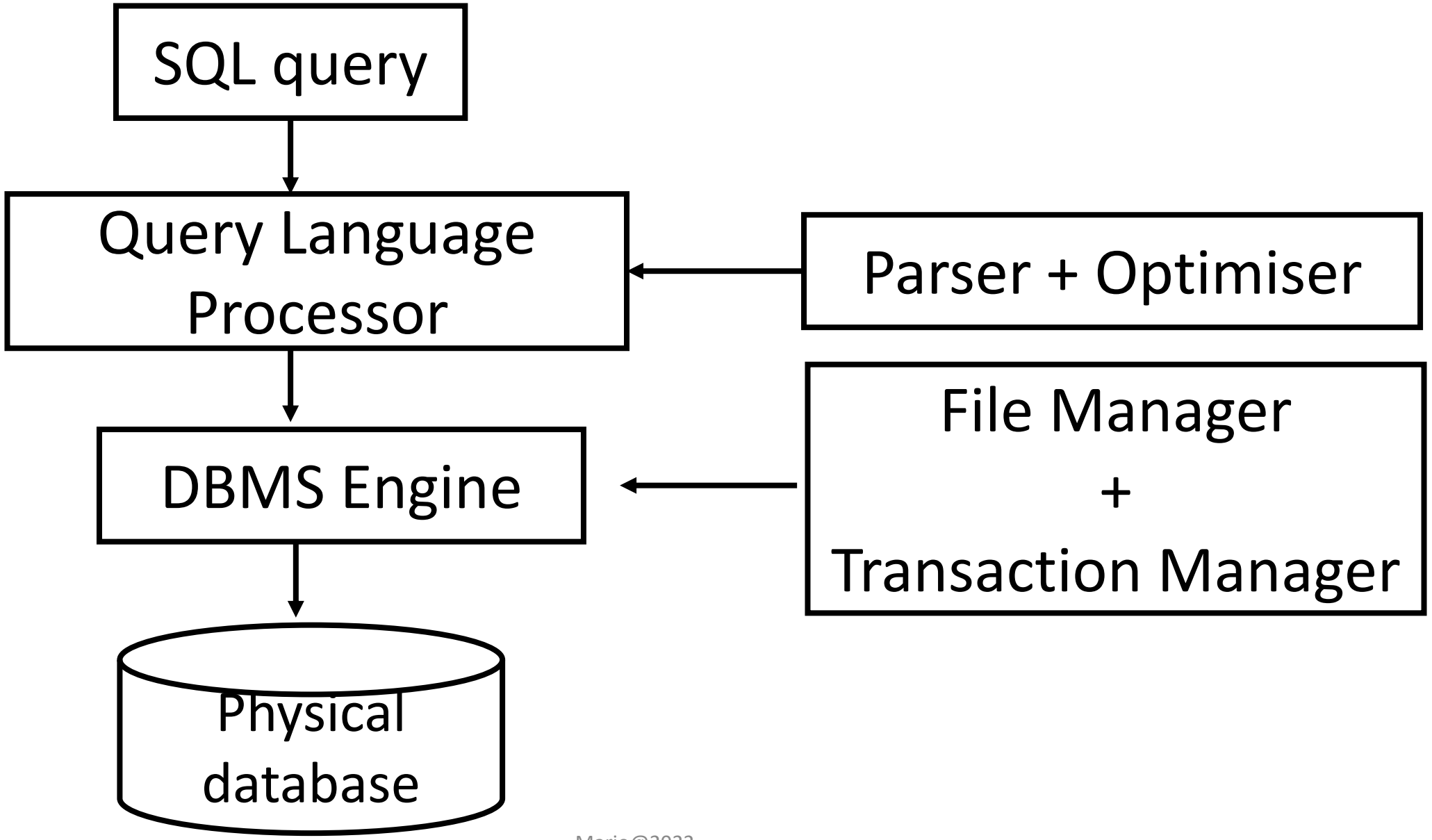
- Is a database sublanguage for querying and modifying relational database.
- It is nonprocedural language used for creating and manipulating data in a relational database.
- It is keyword based language.
- Each statement begins with a keyword.
- The syntax is not case sensitive.

- To retrieve records from a relational database, a user creates a request (query) that instructs a DBMS to retrieve specific records. e.g. SELECT *First name = ‘Agrey’ FROM Pupils.*

Functions of SQL

- Create and modify database structure
- Add new records into database.
- Display information from a database.
- Modify (delete/update) database content
- Enforce database security.

How SQL performs query operations



Exercise

SQL Syntax (Rules)

- An SQL statement starts with a keyword like CREATE and the entire statement ends with a semicolon (;).
- SQL commands such as CREATE cannot be split across lines.
- SQL commands are not case sensitive. This means *SELECT* and *select* are the same.
- For readability, keyword clauses should be placed on separate lines.

SQL Data Types

- Data types are used to specify the type of data that can be stored in a database.

Data Type	Description
CHAR (size)	Fixed-length character string the size is specified in parenthesis. Max 255 bytes.
VARCHAR	Variable –length character string. Max is specified in parenthesis
BIT	Takes a binary value which is either a 0 or 1
DECIMAL	Takes a value with a decimal places.
INT	Numeric values
DATE	Stores the date
TIME	Stores time of the day
TEXT	Variable-length alphanumeric data with maximum length of 2, 147, 483, 647 characters.

Types of SQL commands

- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- Data Control Language (DCL)

Creating a Database using DDL

- The general form of SQL statement for creating a database.

CREATE DATABASE DatabaseName;

e.g. to create a database known as StclementsBoys.

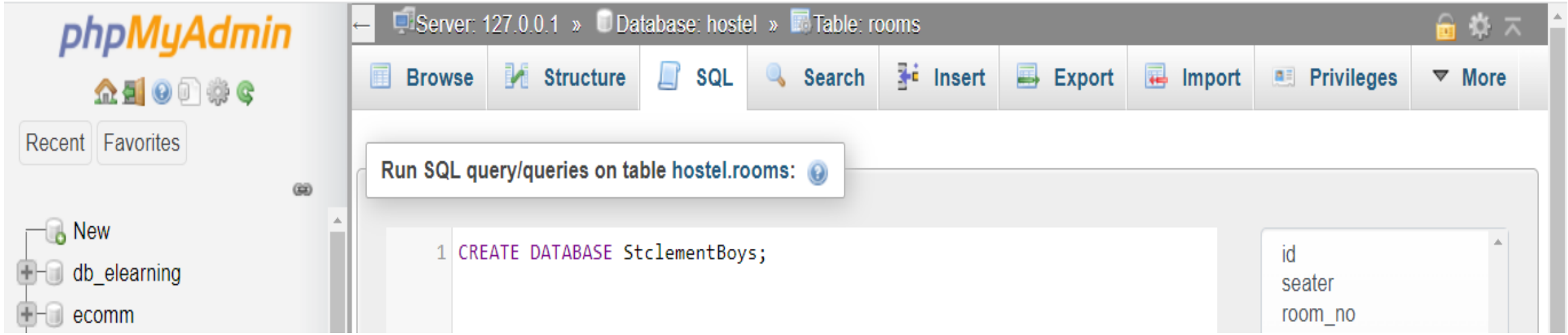
CREATE DATABASE StclementsBoys;

Data Definition Language (DDL)

- Defining a database involves specifying the structure, data type and rules (constraints) of the data to be stored in a database.
- It provides a set of commands to create, altering and removing database content.

Command	Description
CREATE	Creates new database, table views or other objects in a database.
ALTER	Modifies structure of an existing database objects such inserting a new field in a table
DROP	Deletes a named database or table and its content. If it is desired that only content is removed without deleting the table, DELETE command is used.

CREATE DATABASE StclementsBoys;



- Once you finish writing SQL statements, click Go button at the bottom to run the query

SHOW DATABASES;

Your SQL query has been executed successfully

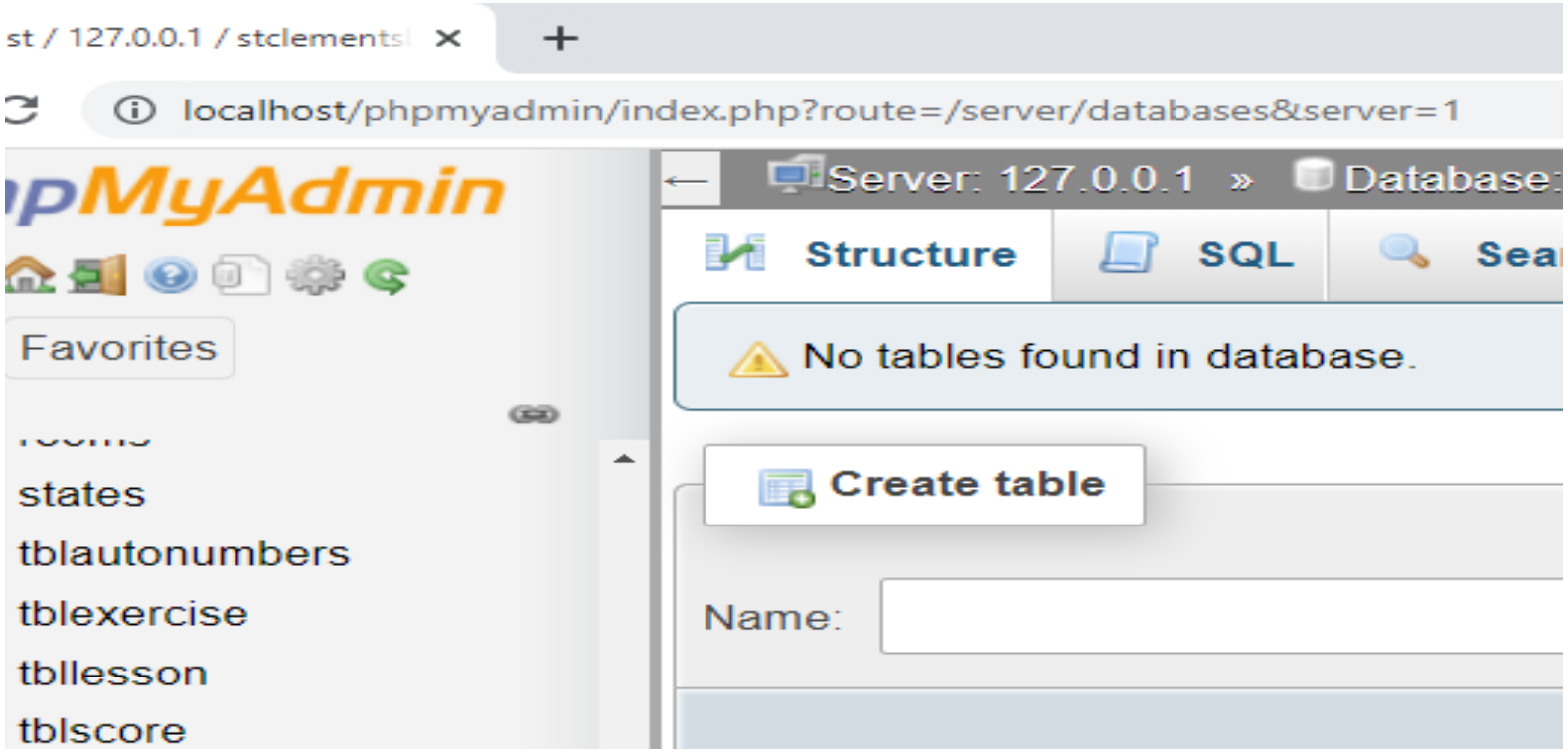
```
SHOW DATABASES
```

+ Options

Database
db_elearning
ecomm
hostel
information_schema
mysql
performance_schema

To open a database to perform other operation like creating a table to use the database of the USE command.

USE DatabaseName;



CREATE TABLE Command

Syntax

```
CREATE TABLE table_name (
column1 DATATYPE,
column2 DATATYPE,
column3 DATATYPE,
);
```

In brackets comes the list defining the name and data types of each column.

```
CREATE TABLE grade_twelve (
firstname VARCHAR (15),
surname VARCHAR (20),
age INT (3),
gender VARCHAR (6),
phone VARCHAR (30),
town VARCHAR (30),
);
```

localhost / 127.0.0.1 / stclements

localhost/phpmyadmin/index.php?route=/database/sql&db=stclementsboys

phpMyAdmin

cent Favorites

New
db_elearning
ecomm
New
cart
category
details
products

Server: 127.0.0.1 » Database: stclementsboys

Structure SQL Search Query Export Import

Run SQL query/queries on database stclementsboys:

```
1 CREATE TABLE grade_twelve(
2 firstname VARCHAR(15),
3 surname VARCHAR(20),
4 age INT(3),
5 gender VARCHAR(6),
6 phone VARCHAR(30) ,
7 town VARCHAR(30),
8 );
```

Mario@2022

NOTE

- Use an open parenthesis before the beginning of the columns and the closing parenthesis followed by a semicolon.
- Separate the column definitions statements with commas.
- The table and column names must start with a letter and should not exceed 30 characters in length.
- You can verify if the table has been created successfully by looking at the message displayed by the SQL server, otherwise you can use **DESC** Command to view the structure.

SQL CONSTRAINTS

- A constraints is a rule associated with a field or table that ensures accuracy and reliability of the data entered into the database.
- **NOT NULL:** ensures that the column must contain a value
- **DEFAULT:** provides a default value for a column when none is specified.
- **UNIQUE:** ensures that all values in a column are unique.
- **PRIMARY KEY:** Uniquely identifies a record in a database table. It also enforces NOT NULL constraints. The field cannot duplicated.
- **FOREGN KEY:** a column or a combination of columns, whose values match a primary key in a related table.
- **CHECK:** it ensures that all values in a column satisfy certain conditions.

NOT NULL

```
CREATE TABLE employee (
StaffID INT NOT NULL,
firstname VARCHAR (15),
surname VARCHAR (20),
age INT (3),
gender VARCHAR (6),
phone VARCHAR (30),
Salary DECIMAL (18, 2),
);
```

UNIQUE

```
CREATE TABLE employee (
StaffID INT NOT NULL UNIQUE,
firstname VARCHAR (15),
surname VARCHAR (20),
age INT (3),
gender VARCHAR (6),
phone VARCHAR (30),
Salary DECIMAL (18, 2),
);
```

PRIMARY KEY

```
CREATE TABLE customer (
customerID INT PRIMARY KEY,
firstname VARCHAR (15),
surname VARCHAR (20),
age INT (3),
gender VARCHAR (6),
phone VARCHAR (30),
town VARCHAR (30),
);
```

FOREGN KEY

```
CREATE TABLE orders   (
orderID INT NOT NULL,
Orderdate DATETIME,
customerID INT FOREGN KEY REFERENCES
customerID (customerID),
Amount DOUBLE,
PRIMARY KEY (orderID)
);
```

Check

```
CREATE TABLE computers (
serialID INT PRIMARY KEY,
brand VARCHAR (15),
model VARCHAR (20),
age INT (3),
price DECIMAL (10, 2) CHECK (price >0),
StaffID INT FORGN KEY REFERENCES
Employee (staffId)
);
```

Dropping constraints

- Any constraint defined can be dropped using ALTER TABLE with DROP CONSTRAINTS clause.

EXAMPLE

ALTER TABLE employee DROP PRIMARY KEY;

Removing Database and Tables

- The two SQL commands used to modify an existing database and table structure are ALTER and DROP.
- To completely delete a database, use drop<*database_name*> *clause.*
- To completely delete a table, use drop<*table_name*> *clause.*

DROP DATABASE databasename;

DROP TABLE tablename;

Example

DROP DATABASE stclements;

DROP TABLE teachers;

Exercise

Data manipulation Language. (DML)

- To manipulate a database, DML commands are used for selecting, inserting updating and deleting data.
- Manipulating Database involves operations such as retrieving specific data and updating the database.
- It use INSERT, UPDATE and DELETE COMMANDS.

Command	Description
INSERT	Inserts data into a database table
UPDATE	Modifies records in a database table
DELETE	Deletes records from a database
SELECT	Used for retrieving data from a database

INSERT COMMAND

- It used to add new record into a table.
- Use INSERT INTO clause followed by the table name and then the list of column names then followed by the keyword VALUES within a list of values enclosed in parenthesis as shown below:
- **INSERT INTO tablename (column 1,**
  column 2,……
  column n,)

VALUES (fistname, surname, …….last_value);

Example

```
INSERT INTO teachers (firstname, surname, age,
phone)
VALUES (`Mario`, `Makhanga`, `30`, `0972222222`);
```

UPDATE COMMAND

- The UPDATE command is used to modify field values of one or more records that match the specified criteria in the WHERE clause.
- It is made of three clauses UPDATE, SET and WHERE.

UPDATE “tablename”

SET “columnname” = “newvalue”

WHERE <condition>

Example

UPDATE pupils

SET firstname = ‘Boyd’

WHERE pupilExamNo = 170;

- The first statement starting with UPDATE command specifies that one or more tables to be modified.
- The condition WHERE clause is a Boolean expression that specifies the criteria to be met for the field to be updated.

DELETE COMMAND

- It is a statement which is used to remove records from the table.
- It is made of two clauses DELETE FROM and WHERE.

DELETE FROM <tablename>

WHERE <condition>

Example

DELETE FROM customers

WHERE firstname = 'Kabengele';

- To delete all the records from a table enter DELETE FROM followed by table name but leave WHERE

DELETE FROM employee;

DISPLAYING RECORDS FROM DATABASE

- To display data from a database, SQL has keywords SELECT and FROM which can be used to create simple to complex queries.

Querying single table

- The basic form of the SELECT statement used to retrieve data from a single table is made of three clauses SELECT, FROM and WHERE.

SELECT <column1, column2, column.....>

FROM <table>

WHERE <condition>

- The column names that follow **SELECT** determines the columns to be returned in the results (dynaset).
- The dynaset with selected columns is referred to as the **projection**.
- To create the projection of all table columns use the **asterisk** “*” in place of columns
- The table name follows the keyword **FROM** specifies the table to be queried to retrieve desired results.
- **WHERE** clause specifies the condition to be satisfied for records to be displayed.

Example

SELECT * FROM customers

customerID	firstname	surname	age	gender	phone	town
1	Nathan	Kapapi	15	male	+26097	Lsk
2	Lumuno	Bweendo	14	male	+26092	Kitwe
3	Gideon	Mbulo	16	male	+26072	Mansa
4	Danstan	Mutaba	12	Male	+26078	Samfy
5	Lungu	Madalitso	17	Female	+26096	Mongu

SQL OPERATORS AND WILDCARDS

SQL OPERATORS

- SQL operators are used to define a Boolean condition that must be satisfied for a record to be displayed

1. Relational operators
2. Logical operators
3. Distinct operators

Relational operators

- These are Boolean operators used to compare two values.

1. < less than
2. > greater than
3. = equal to
4. <> not equal to

Example

```
SELECT firstname, town, age
FROM customers
WHERE age >= 15;
```

Logical Operators

- These are comparisons operators are used to combine multiple conditions in SQL

1. LIKE
2. AND
3. OR
4. NOT
5. BETWEEN

Example

SELECT firstname, town, age

FROM customers

WHERE (gender = ‘male’) **AND** (age >=16);

Distinct

- It is used in conjunction with SELECT statement to eliminate duplicate records
- **Note:** SQL can also use arithmetic operators such as +, -, /, *, %
- **e.g SELECT 20 + 15;**