**Definitions**

A **database** is an organized collection of data that can be accessed, managed, and updated efficiently. It stores information in a structured format, making it easy to retrieve and analyze data. Within a database, a **table** is a set of rows and columns where data is stored. Each row represents a unique record, and each column holds specific types of information.

**SQL** (Structured Query Language) is the standard language used to communicate with and manipulate data in relational databases. It allows users to perform tasks such as querying data, updating records, and managing database structures. A **primary key** is a unique identifier for each record in a table. It ensures that every row is uniquely identified and prevents duplication or null values in that column.

A **SELECT query** is used to retrieve data from a table in a database. It allows users to specify which columns and records they want to view. To narrow down the results of a SELECT query, a **WHERE clause** is used to filter records based on specific conditions. If a user wants to avoid duplicate values in a query result, the **DISTINCT** keyword can be used to return only unique entries from a column.

An **alias** in SQL provides a temporary name for a column or table in a query, often making the output more readable or convenient. The **ORDER BY** clause is used to sort the result set of a query based on one or more columns in ascending or descending order. Lastly, **SQL functions** are built-in operations that perform calculations on data, such as COUNT() to count the number of rows, or AVG() to find the average of numeric values.

PART 2 SQL practice script

-- 1. Create the Students table

CREATE TABLE Students (

StudentID INT,

Name VARCHAR(100),

Age INT,

Gender VARCHAR(10),

GradeLevel VARCHAR(20),

School VARCHAR(100),

City VARCHAR(50),

AverageScore DECIMAL(5,1)

);

-- 2. Insert the sample records

INSERT INTO Students VALUES

(1, 'Alice Njeri', 16, 'Female', 'Form 2', 'Green Hill Academy', 'Nairobi', 78.5),

(2, 'Brian Otieno', 17, 'Male', 'Form 3', 'Kisumu Boys High', 'Kisumu', 84.3),

(3, 'Cynthia Wambui', 15, 'Female', 'Form 2', 'Ridgeways Girls', 'Nairobi', 91.0),

(4, 'David Mwangi', 16, 'Male', 'Form 2', 'Alliance High', 'Kikuyu', 74.0),

(5, 'Eva Akinyi', 18, 'Female', 'Form 4', 'Moi Girls Eldoret', 'Eldoret', 88.2),

(6, 'Felix Kiptoo', 17, 'Male', 'Form 3', 'Kapsabet Boys', 'Kapsabet', 82.7),

(7, 'Grace Muthoni', 14, 'Female', 'Form 1', 'Starehe Girls', 'Thika', 77.1),

(8, 'Hassan Abdalla', 15, 'Male', 'Form 2', 'Mombasa Secondary', 'Mombasa', 69.8),

(9, 'Irene Cherono', 16, 'Female', 'Form 3', 'Kabarak High', 'Nakuru', 86.4),

(10, 'John Kamau', 17, 'Male', 'Form 4', 'Lenana School', 'Nairobi', 80.0),

(11, 'Kevin Omondi', 15, 'Male', 'Form 2', 'Maseno School', 'Kisumu', 72.5);

-- 3. Display all records

SELECT \* FROM Students;

-- 4. Display only the distinct grade levels

SELECT DISTINCT GradeLevel FROM Students;

-- 5. Show student names and their grade levels using column aliases

SELECT Name AS StudentName, GradeLevel AS Class FROM Students;

-- 6. Display names and ages of students older than 15

SELECT Name, Age FROM Students WHERE Age > 15;

-- 7. Display all students sorted by Age in ascending order

SELECT \* FROM Students ORDER BY Age ASC;

-- 8. Display all female students sorted by Name in descending order

SELECT \* FROM Students WHERE Gender = 'Female' ORDER BY Name DESC;

-- 9. Count the number of students in the table

SELECT COUNT(\*) AS TotalStudents FROM Students;

-- 10. Show the average age of students using an aggregate function

SELECT AVG(Age) AS AverageAge FROM Students;