Basic SQL Concepts

1. Creating Databases and Tables

A **database** is a structured collection of data. In SQL, you can create a new database using the following command:

```
CREATE DATABASE SchoolDB;
```

After creating a database, you can create **tables** inside it. A table stores data in rows and columns, similar to a spreadsheet.

Example of creating a table:

```
CREATE TABLE Students ( StudentID INT PRIMARY KEY, FirstName VARCHAR(50), LastName VARCHAR(50), Age INT, Grade DECIMAL(3,1) );
```

Here, we use **data types** such as:

- `INT` → whole numbers.
- `VARCHAR(n)` → text of up to *n* characters.
- `DECIMAL(p,s)` → numbers with decimals (for example, grades like 9.5).

---

2. Inserting Data into Tables

After creating a table, we can add (or **insert**) data into it using the `INSERT INTO` statement:

```
INSERT INTO Students (StudentID, FirstName, LastName, Age, Grade) VALUES (1, 'Maria', 'Lopez', 20, 9.5);
```

You can also insert multiple records at once:

```
INSERT INTO Students VALUES (2, 'John', 'Smith', 19, 8.7), (3, 'Ana', 'Torres', 21, 9.0);
```

---

3. Using SELECT, WHERE, ORDER BY, and GROUP BY

- **SELECT** → retrieves data from a table.

- **WHERE** → filters data according to a condition.
- **ORDER BY** → sorts results in ascending (`ASC`) or descending (`DESC`) order.
- **GROUP BY** → groups rows that have the same values in specified columns.

Examples:

```
-- Show all students SELECT * FROM Students; -- Show students older than 19 SELECT FirstName, Age FROM Students WHERE Age > 19; -- Show students ordered by grade SELECT FirstName, Grade FROM Students ORDER BY Grade DESC; -- Group students by age and count how many have the same age SELECT Age, COUNT(*) AS NumberOfStudents FROM Students GROUP BY Age;
```

---

4. Applying Basic Aggregate Functions

Aggregate functions perform calculations on a group of rows and return a single value.
Common aggregate functions are:

- `COUNT()` → counts the number of rows.
- `AVG()` → calculates the average.
- `MAX()` and `MIN()` → find the highest or lowest value.
- `SUM()` → adds all values together.

Example:

```
SELECT COUNT(*) AS TotalStudents, AVG(Grade) AS AverageGrade, MAX(Grade) AS HighestGrade, MIN(Grade) AS LowestGrade FROM Students;
```

---

5. Writing Simple Joins Between Two Tables

A **JOIN** combines rows from two or more tables based on a related column.
Example with two tables:

```
CREATE TABLE Courses ( CourseID INT PRIMARY KEY, CourseName VARCHAR(50) ); CREATE TABLE Enrollments ( StudentID INT, CourseID INT );
```

To see which student is enrolled in which course:

```
SELECT Students.FirstName, Courses.CourseName FROM Students JOIN Enrollments ON Students.StudentID = Enrollments.StudentID JOIN Courses ON Enrollments.CourseID = Courses.CourseID;
```

This query connects the `Students`, `Enrollments`, and `Courses` tables to show complete information.