SQL queries [Basic]

📌 *Step 1: Database & Table Basics*

```sql
1   -- Show existing databases
2   SHOW databases;
3
4   -- Create a new database
5   CREATE DATABASE schooldb;
6
7   -- Use the database
8   USE schooldb;
9
10  -- Create a student table
11  CREATE TABLE student (
12      id INT AUTO_INCREMENT PRIMARY KEY,
13      name VARCHAR(50),
14      email VARCHAR(100),
15      age INT
16  );
17
```

📌 *Step 2: Insert Data*

```sql
1   -- Insert a single record
2   INSERT INTO student (name, email, age)
3   VALUES ('Alice', 'alice@example.com', 22);
4
5   -- Insert multiple records
6   INSERT INTO student (name, email, age)
7   VALUES
8   ('Bob', 'bob@example.com', 20),
9   ('Carol', 'carol@example.com', 23),
10  ('David', 'david@example.com', 21);
11
```

📌 *Step 3: Select / Retrieve Data*

```sql
1   -- Select all columns
2   SELECT * FROM student;
3
4   -- Select specific columns
5   SELECT name, age FROM student;
6
7   -- Use WHERE clause
8   SELECT * FROM student WHERE age > 21;
9
```

```sql
10  -- Use ORDER BY
11  SELECT * FROM student ORDER BY age ASC;
12
13  -- Use LIMIT
14  SELECT * FROM student LIMIT 2;
15
16  -- Use DISTINCT
17  SELECT DISTINCT age FROM student;
18
```

📌 Step 4: Update & Delete

```sql
1  -- Update a record
2  UPDATE student SET age = 24 WHERE name = 'Alice';
3
4  -- Delete a record
5  DELETE FROM student WHERE name = 'Bob';
6
```

📌 Step 5: Filtering with WHERE

```sql
1   -- Use comparison operators
2   SELECT * FROM student WHERE age <= 22;
3
4   -- Use BETWEEN
5   SELECT * FROM student WHERE age BETWEEN 20 AND 22;
6
7   -- Use LIKE for pattern matching
8   SELECT * FROM student WHERE name LIKE 'C%';  -- names starting with 'C'
9
10  -- Use IN
11  SELECT * FROM student WHERE age IN (21, 23);
12
```

📌 Step 6: Aggregate Functions

```sql
1   -- Get total number of students
2   SELECT COUNT(*) FROM student;
3
4   -- Get average age
5   SELECT AVG(age) FROM student;
6
7   -- Get oldest student
8   SELECT MAX(age) FROM student;
9
10  -- Get youngest student
11  SELECT MIN(age) FROM student;
12
13  -- Sum (not always meaningful for age, but for demo)
14  SELECT SUM(age) FROM student;
```

📌 Step 7: GROUP BY and HAVING

```sql
-- Count how many students of each age
SELECT age, COUNT(*) FROM student GROUP BY age;

-- Use HAVING (filter after GROUP BY)
SELECT age, COUNT(*) FROM student GROUP BY age HAVING COUNT(*) > 1;

```

📌 Step 8: Table Alterations

```sql
-- Add a new column
ALTER TABLE student ADD gender VARCHAR(10);

-- Rename a column (MySQL 8+)
ALTER TABLE student RENAME COLUMN gender TO sex;

-- Drop a column
ALTER TABLE student DROP COLUMN sex;

```

📌 Step 9: Dropping Tables and Database

```sql
-- Drop a table
DROP TABLE enrollment;

-- Drop a database
DROP DATABASE schooldb;

```

Document by **Suyash**😇