Practical no. 1

Design a student database system that demonstrates the use of SQL DDL (Data Definition Language). Perform the following tasks:

A) DDL Operations:

1. Create a Database named practical.

2. Create a Table named stud with the following fields:

- **rno – Roll number (Primary Key, Auto Increment)**
- **name – Student name (VARCHAR)**
- **age – Age (Integer, NOT NULL)**
- **marks – Marks (Integer)**
- **city – City (VARCHAR)**

3. Apply appropriate constraints (e.g., PRIMARY KEY, NOT NULL, AUTO_INCREMENT).

4. Alter the table structure to add or modify columns as required.

5. Create an Index on the name column to improve search efficiency.

6. Create a View (stud_view) to display all student records.

7. Create another view (stud_view2) to show records of students who belong to a particular city (e.g., Nashik).

8. Rename and Drop views to demonstrate their management.

```
1) CREATE DATABASE Practical;
Query OK, 1 row affected (0.04 sec)

2) CREATE TABLE stud
    -> (
    -> rno INT PRIMARY KEY AUTO_INCREMENT,
    -> name VARCHAR(20),
    -> AGE INT NOT NULL,
    -> marks INT,
    -> city VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.15 sec)

4) ALTER TABLE stud
    -> ADD email VARCHAR(100);
Query OK, 0 rows affected (0.10 sec)
Records: 0  Duplicates: 0  Warnings: 0

OR

     ALTER TABLE stud
    -> MODIFY name VARCHAR(50);
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

5) CREATE INDEX indexname ON stud(name);
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
INSERT INTO stud (name, age, marks, city, email) VALUES
    -> ('Alice', 20, 85, 'Nashik', 'alice@example.com'),
    -> ('Bob', 21, 78, 'Pune', 'bob@example.com'),
    -> ('Charlie', 20, 92, 'Nashik', 'charlie@example.com'),
    -> ('David', 22, 65, 'Mumbai', 'david@example.com');
Query OK, 4 rows affected (0.05 sec)
Records: 4  Duplicates: 0  Warnings: 0

SELECT * FROM stud;
+-----+---------+-----+-------+--------+---------------------+
| rno | name    | Age | marks | city   | email               |
+-----+---------+-----+-------+--------+---------------------+
|   1 | Alice   |  20 |    85 | Nashik | alice@example.com   |
|   2 | Bob     |  21 |    78 | Pune   | bob@example.com     |
|   3 | Charlie |  20 |    92 | Nashik | charlie@example.com |
|   4 | David   |  22 |    65 | Mumbai | david@example.com   |
+-----+---------+-----+-------+--------+---------------------+
4 rows in set (0.00 sec)

6) CREATE VIEW stud_view AS
    -> SELECT rno, name, age, marks, city
    -> FROM stud;
Query OK, 0 rows affected (0.04 sec)

SELECT * FROM stud_view;
+-----+---------+-----+-------+--------+
| rno | name    | age | marks | city   |
+-----+---------+-----+-------+--------+
|   1 | Alice   |  20 |    85 | Nashik |
|   2 | Bob     |  21 |    78 | Pune   |
|   3 | Charlie |  20 |    92 | Nashik |
|   4 | David   |  22 |    65 | Mumbai |
+-----+---------+-----+-------+--------+
4 rows in set (0.00 sec)

7) CREATE VIEW stud_view2 AS
    -> SELECT rno, name, age, marks, city
    -> FROM stud
    -> WHERE city = "Nashik";
Query OK, 0 rows affected (0.03 sec)

SELECT * FROM stud_view2;
+-----+---------+-----+-------+--------+
| rno | name    | age | marks | city   |
+-----+---------+-----+-------+--------+
|   1 | Alice   |  20 |    85 | Nashik |
|   3 | Charlie |  20 |    92 | Nashik |
+-----+---------+-----+-------+--------+
2 rows in set (0.00 sec)
```

8) RENAME TABLE stud_view TO all_students_view;
Query OK, 0 rows affected (0.05 sec)

DESC all_students_view;

Field	Type	Null	Key	Default	Extra
rno	int	NO		0	
name	varchar(50)	YES		NULL	
age	int	NO		NULL	
marks	int	YES		NULL	
city	varchar(50)	YES		NULL	

5 rows in set (0.00 sec)

DROP VIEW all_students_view;
Query OK, 0 rows affected (0.01 sec)

DESC all_students_view;
ERROR 1146 (42S02): Table 'practical.all_students_view' doesn't exist

B) DML Operations:

1. Create a Database named practical

2. Create a Table named stud with the following fields:

- **rno – Roll number (Primary Key, Auto Increment)**
- **name – Student name (VARCHAR)**
- **age – Age (Integer, NOT NULL)**
- **marks – Marks (Integer)**
- **city – City (VARCHAR)**

3. Insert at least 10 student records into the table with different values for name, age, marks, and city.

4. Update specific records (e.g., modify age or city of a student).

5. Delete a record based on roll number or name.

6. Display all student records.

7. Retrieve student names starting with the letter 'S'.

8. Retrieve student names ending with 'i'.

9. Display students within a specific roll number range.

10. Sort and display student records by name and by marks.

11. Perform aggregate functions such as MAX(), MIN(), SUM(), AVG(), and COUNT().

12. Use GROUP BY to find the number of students and highest marks per age group.

```
1) CREATE DATABASE Practical;
Query OK, 1 row affected (0.04 sec)

2) CREATE TABLE stud
    -> (
    -> rno INT PRIMARY KEY AUTO_INCREMENT,
    -> name VARCHAR(20),
    -> AGE INT NOT NULL,
    -> marks INT,
    -> city VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.15 sec)

3) INSERT INTO stud (name, age, marks, city) VALUES
    -> ('Alice', 20, 85, 'Nashik'),
    -> ('Bob', 21, 78, 'Pune'),
    -> ('Charlie', 20, 92, 'Nashik'),
    -> ('David', 22, 65, 'Mumbai'),
    -> ('Eve', 21, 95, 'Nashik'),
    -> ('Frank', 20, 70, 'Pune'),
    -> ('Grace', 22, 88, 'Mumbai'),
    -> ('Samir', 23, 76, 'Delhi'),
    -> ('Shivani', 20, 90, 'Nashik'),
    -> ('Amit', 21, 82, 'Pune');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0



select * from stud;
```

```
+-----+---------+-----+-------+--------+
| rno | name    | age | marks | city   |
+-----+---------+-----+-------+--------+
|   1 | Alice   |  20 |    85 | Nashik |
|   2 | Bob     |  21 |    78 | Pune   |
|   3 | Charlie |  20 |    92 | Nashik |
|   4 | David   |  22 |    65 | Mumbai |
|   5 | Eve     |  21 |    95 | Nashik |
|   6 | Frank   |  20 |    70 | Pune   |
|   7 | Grace   |  22 |    88 | Mumbai |
|   8 | Samir   |  23 |    76 | Delhi  |
|   9 | Shivani |  20 |    90 | Nashik |
|  10 | Amit    |  21 |    82 | Pune   |
+-----+---------+-----+-------+--------+
10 rows in set (0.00 sec)
```

4) UPDATE stud
**   -> SET age = 23**
**   -> WHERE rno = 4;**

```
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

5) DELETE FROM stud
**   -> WHERE rno = 6;**

```
Query OK, 1 row affected (0.04 sec)
```

6) SELECT * FROM stud;

```
+-----+---------+-----+-------+--------+
| rno | name    | age | marks | city   |
+-----+---------+-----+-------+--------+
|   1 | Alice   |  20 |    85 | Nashik |
|   2 | Bob     |  21 |    78 | Pune   |
|   3 | Charlie |  20 |    92 | Nashik |
|   4 | David   |  23 |    65 | Mumbai |
|   5 | Eve     |  21 |    95 | Nashik |
|   7 | Grace   |  22 |    88 | Mumbai |
|   8 | Samir   |  23 |    76 | Delhi  |
|   9 | Shivani |  20 |    90 | Nashik |
|  10 | Amit    |  21 |    82 | Pune   |
+-----+---------+-----+-------+--------+
9 rows in set (0.00 sec)
```

7) SELECT name FROM stud
**   -> WHERE name LIKE "S%";**

```
+---------+
| name    |
+---------+
| Samir   |
| Shivani |
+---------+
2 rows in set (0.00 sec)
```

8) SELECT name FROM stud
**   -> WHERE name LIKE "%i";**

```
+---------+
| name    |
+---------+
| Shivani |
+---------+
1 row in set (0.00 sec)
```

9) SELECT * FROM stud
**   -> WHERE rno BETWEEN 2 AND 5;**

```
+-----+---------+-----+-------+--------+
| rno | name    | age | marks | city   |
+-----+---------+-----+-------+--------+
|   2 | Bob     |  21 |    78 | Pune   |
|   3 | Charlie |  20 |    92 | Nashik |
|   4 | David   |  23 |    65 | Mumbai |
|   5 | Eve     |  21 |    95 | Nashik |
+-----+---------+-----+-------+--------+
4 rows in set (0.01 sec)
```

10) SELECT * FROM stud
**   -> ORDER BY name;**

```
+-----+---------+-----+-------+--------+
| rno | name    | age | marks | city   |
+-----+---------+-----+-------+--------+
|   1 | Alice   |  20 |    85 | Nashik |
|  10 | Amit    |  21 |    82 | Pune   |
|   2 | Bob     |  21 |    78 | Pune   |
|   3 | Charlie |  20 |    92 | Nashik |
|   4 | David   |  23 |    65 | Mumbai |
|   5 | Eve     |  21 |    95 | Nashik |
|   7 | Grace   |  22 |    88 | Mumbai |
|   8 | Samir   |  23 |    76 | Delhi  |
|   9 | Shivani |  20 |    90 | Nashik |
+-----+---------+-----+-------+--------+
9 rows in set (0.03 sec)
```

SELECT * FROM stud
**   -> ORDER BY marks DESC;**

```
+-----+---------+-----+-------+--------+
| rno | name    | age | marks | city   |
+-----+---------+-----+-------+--------+
|   5 | Eve     |  21 |    95 | Nashik |
|   3 | Charlie |  20 |    92 | Nashik |
|   9 | Shivani |  20 |    90 | Nashik |
|   7 | Grace   |  22 |    88 | Mumbai |
|   1 | Alice   |  20 |    85 | Nashik |
|  10 | Amit    |  21 |    82 | Pune   |
|   2 | Bob     |  21 |    78 | Pune   |
|   8 | Samir   |  23 |    76 | Delhi  |
|   4 | David   |  23 |    65 | Mumbai |
+-----+---------+-----+-------+--------+
9 rows in set (0.00 sec)
```

11) SELECT MAX(marks) as Highest_marks FROM stud;

```
+---------------+
| Highest_marks |
+---------------+
|            95 |
+---------------+
1 row in set (0.01 sec)
```

SELECT MIN(marks) as Lowest_marks FROM stud;

```
+--------------+
| Lowest_marks |
+--------------+
|           65 |
+--------------+
1 row in set (0.03 sec)
```

SELECT SUM(marks) as Total_marks FROM stud;

```
+-------------+
| Total_marks |
+-------------+
|         751 |
+-------------+
1 row in set (0.04 sec)
```

SELECT AVG(marks) as Average_marks FROM stud;

```
+---------------+
| Average_marks |
+---------------+
|       83.4444 |
+---------------+
1 row in set (0.00 sec)
```

SELECT COUNT(rno) as Total_students FROM stud;

```
+----------------+
| Total_students |
+----------------+
|              9 |
+----------------+
1 row in set (0.04 sec)
```

12) SELECT
-> age,
-> COUNT(rno) AS Number_of_Students,
-> MAX(marks) AS Highest_Marks_In_Group
-> FROM stud
-> GROUP BY age;

```
+-----+--------------------+-----------------------+
| age | Number_of_Students | Highest_Marks_In_Group |
+-----+--------------------+-----------------------+
|  20 |                  3 |                    92 |
|  21 |                  3 |                    95 |
|  23 |                  2 |                    76 |
|  22 |                  1 |                    88 |
+-----+--------------------+-----------------------+
4 rows in set (0.00 sec)
```