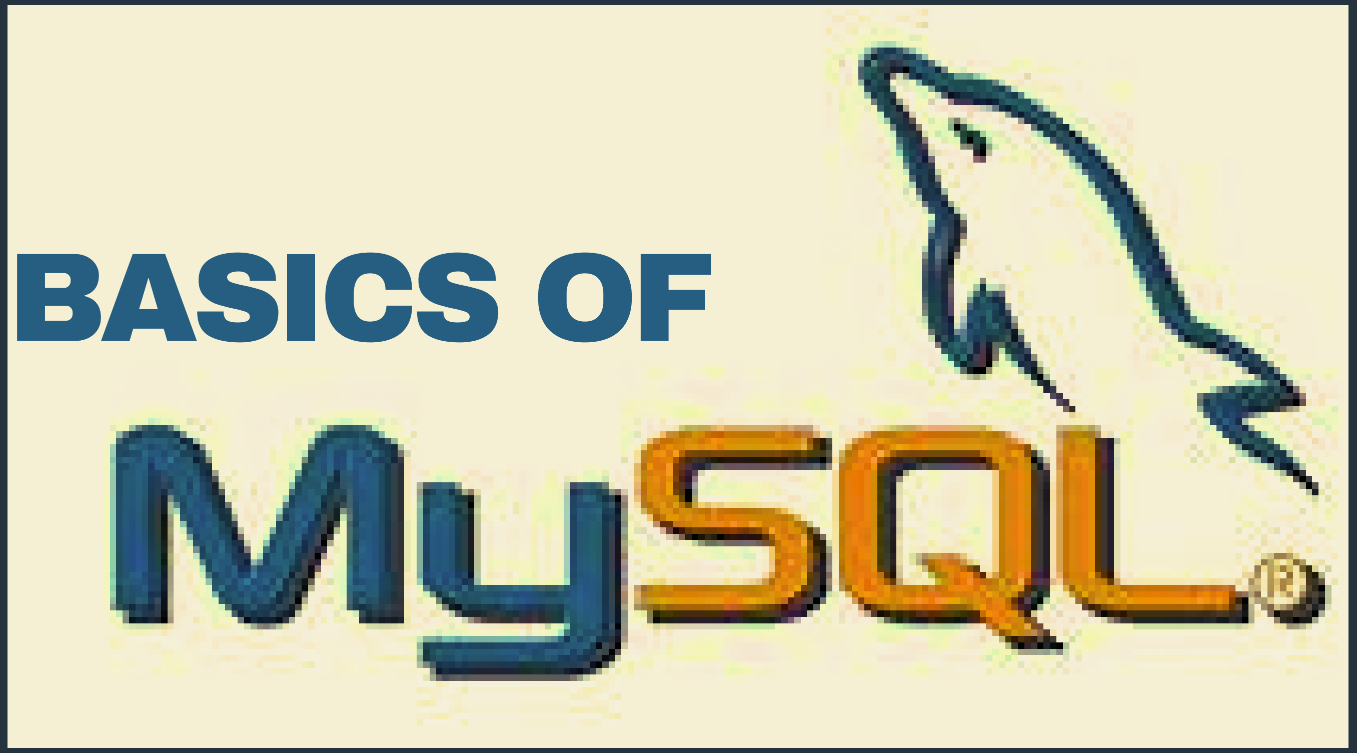
TOPICS

1. Introduction to data & database
2. Introduction SQL & MySQL
3. Database Creation in MySQL
4. CRUD Operation in MySQL
5. Data Types in MySQL
6. Constrains in MySQL
7. Filtering Clauses in MySQL
8. Aggerate Function in MySQL

NAME : MANAS RANJAN PATRA



Data

Data is a collection of raw fact and figures that have no meaning their own.

Database

A database is an organized collection of data stored in a structured way that it can be easily acced, managed and updated.

RDBMS

RDBMS (Relational Database Management System) is a type of database system that store data in structured table and uses SQL for managing and querying data.

SQL

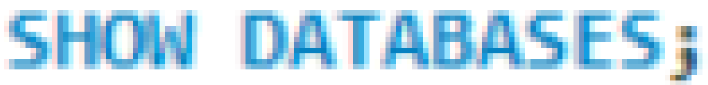
SQL (Structured Query Language) is a programming language used to store, manage and retrieve data from databases.

MySQL

MySQL is a relational database management system (RDBMS) that uses SQL to store and manage data. It is open-source, fast, and widely used in applications and websites.

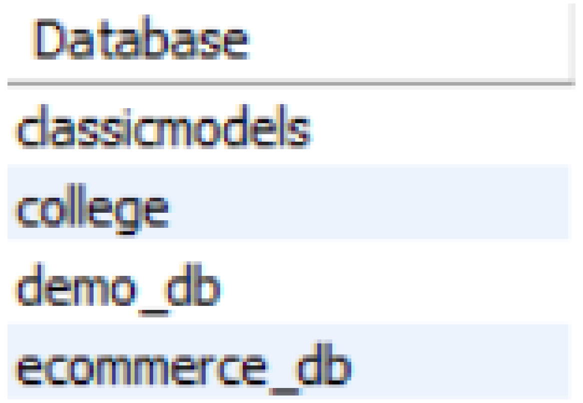
Database

1. List Down All Databases



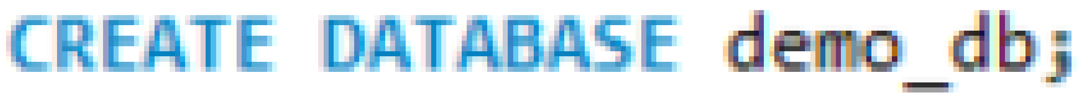
```
SHOW DATABASES;
```

O/P



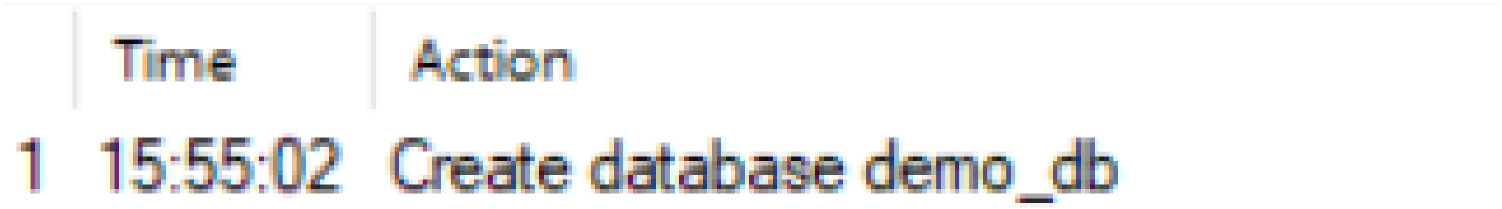
```
Database
classicmodels
college
demo_db
ecommerce_db
```

2. Create Database



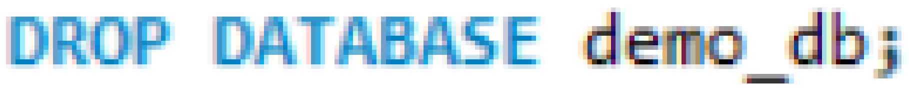
```
CREATE DATABASE demo_db;
```

O/P



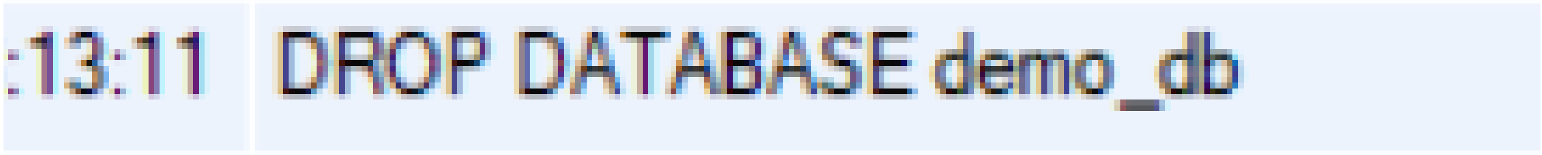
```
   Time      Action
1  15:55:02  Create database demo_db
```

3. Delete Database



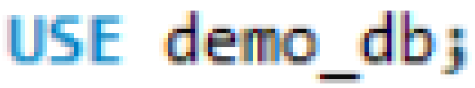
```
DROP DATABASE demo_db;
```

O/P



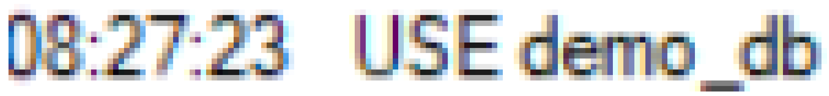
```
:13:11  DROP DATABASE demo_db
```

4. Use Database



```
USE demo_db;
```

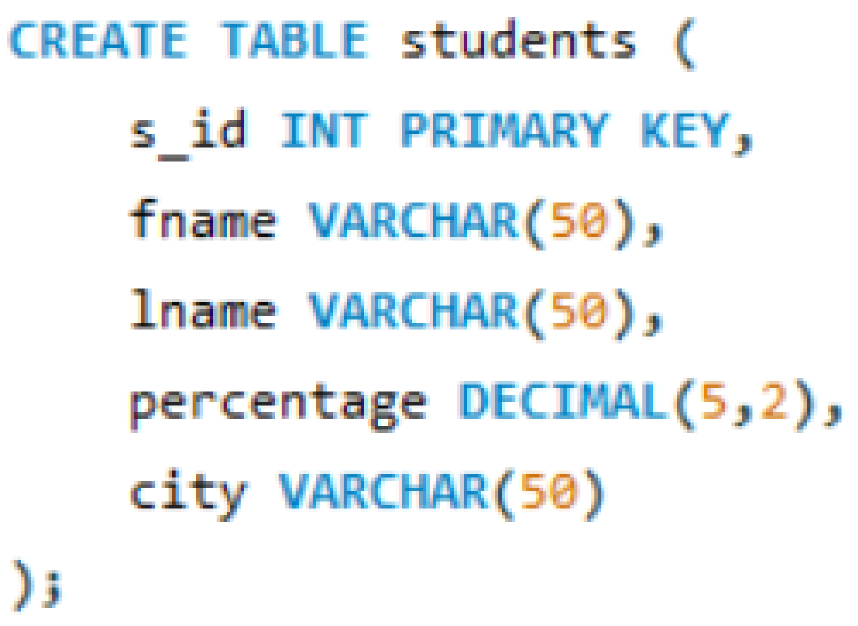
O/P



```
08:27:23  USE demo_db
```

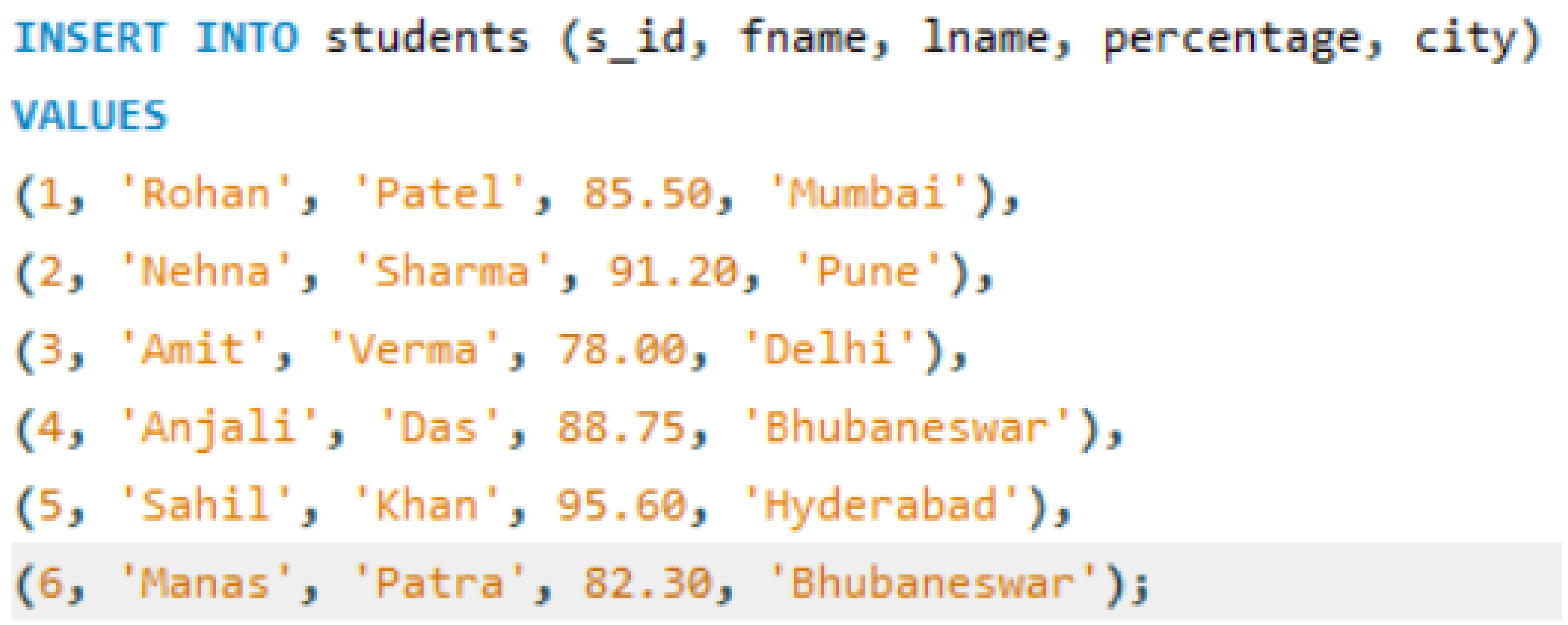
CRUD Operation For Table

1. Create Query



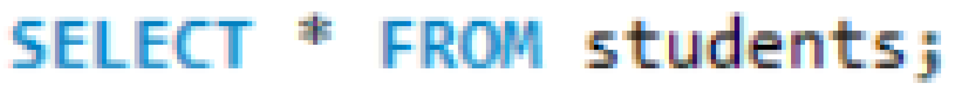
```sql
CREATE TABLE students (
    s_id INT PRIMARY KEY,
    fname VARCHAR(50),
    lname VARCHAR(50),
    percentage DECIMAL(5,2),
    city VARCHAR(50)
);
```

Inserting values into Student Table



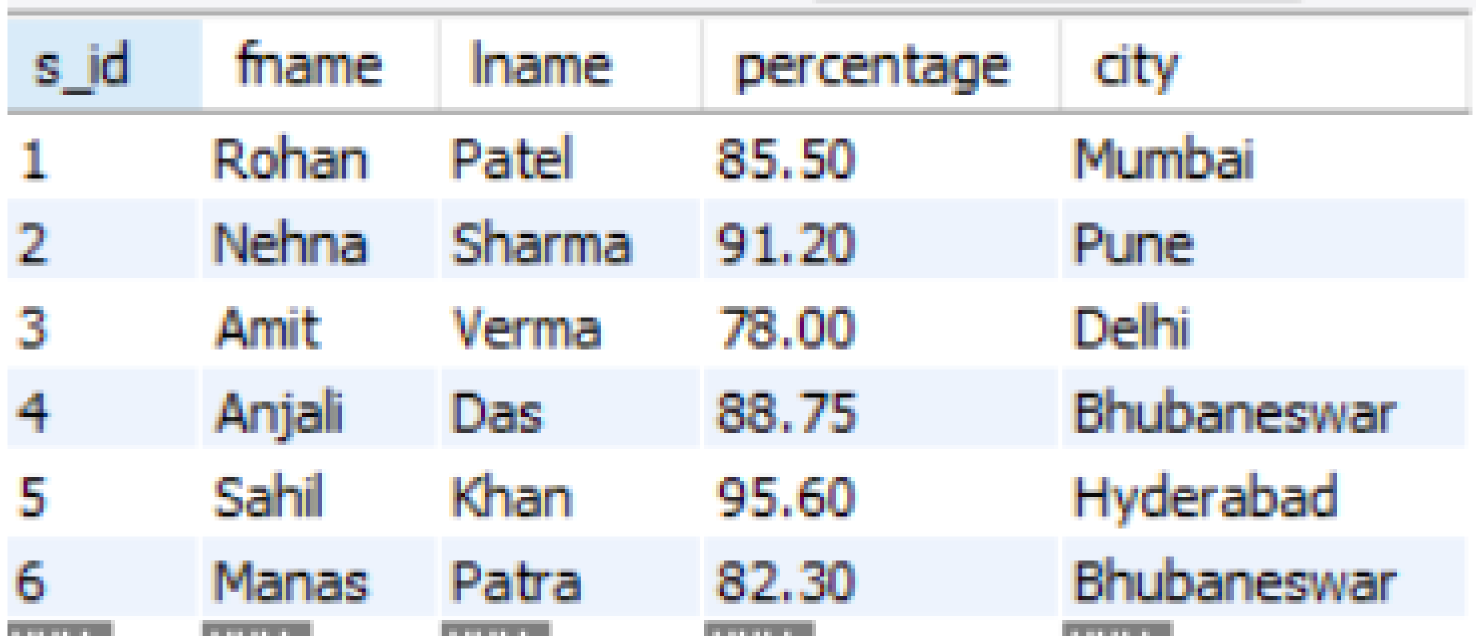
```sql
INSERT INTO students (s_id, fname, lname, percentage, city)
VALUES
(1, 'Rohan', 'Patel', 85.50, 'Mumbai'),
(2, 'Nehna', 'Sharma', 91.20, 'Pune'),
(3, 'Amit', 'Verma', 78.00, 'Delhi'),
(4, 'Anjali', 'Das', 88.75, 'Bhubaneswar'),
(5, 'Sahil', 'Khan', 95.60, 'Hyderabad'),
(6, 'Manas', 'Patra', 82.30, 'Bhubaneswar');
```

2. Read



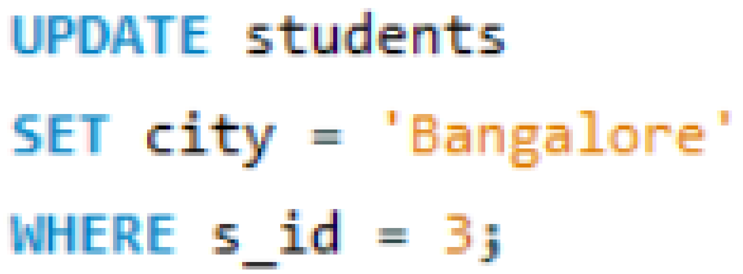
```sql
SELECT * FROM students;
```

O/P



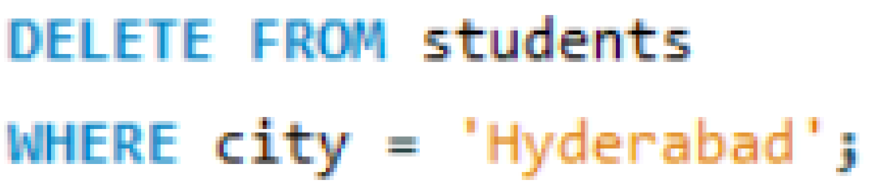
s_id	fname	lname	percentage	city
1	Rohan	Patel	85.50	Mumbai
2	Nehna	Sharma	91.20	Pune
3	Amit	Verma	78.00	Delhi
4	Anjali	Das	88.75	Bhubaneswar
5	Sahil	Khan	95.60	Hyderabad
6	Manas	Patra	82.30	Bhubaneswar

3. Update



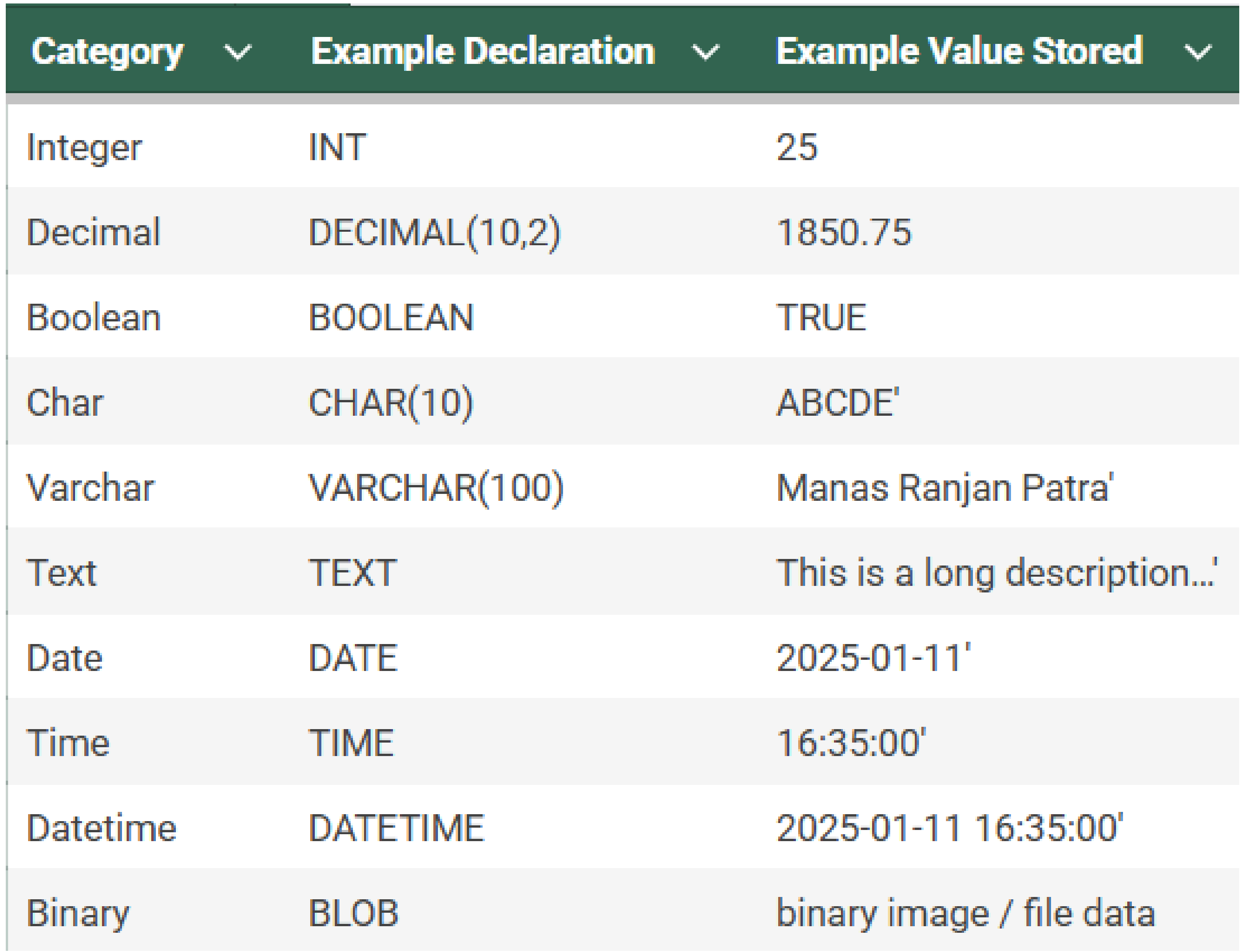
```sql
UPDATE students
SET city = 'Bangalore'
WHERE s_id = 3;
```

4. Delete

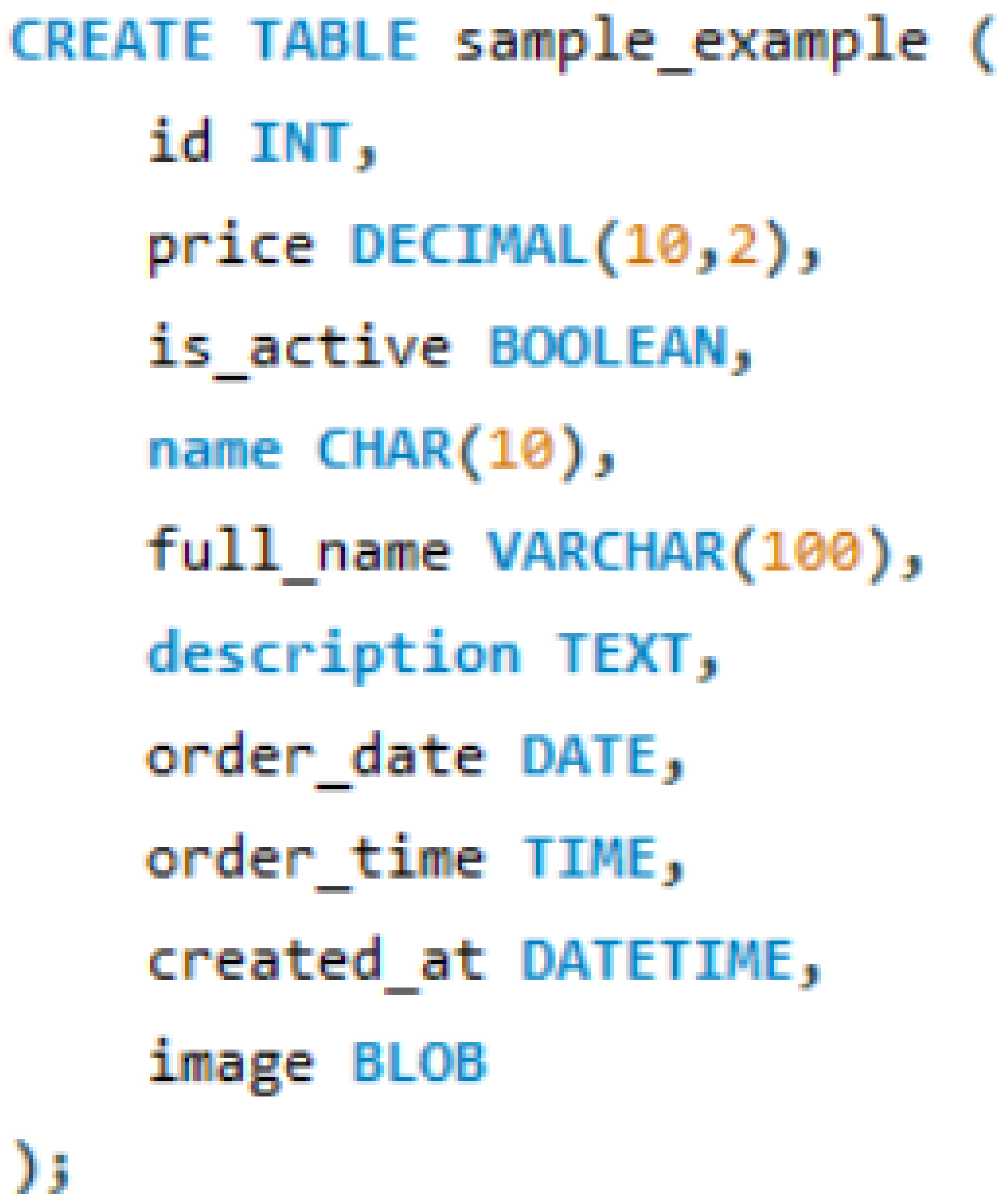


```sql
DELETE FROM students
WHERE city = 'Hyderabad';
```

Data Types



Category	Example Declaration	Example Value Stored
Integer	INT	25
Decimal	DECIMAL(10,2)	1850.75
Boolean	BOOLEAN	TRUE
Char	CHAR(10)	ABCDE'
Varchar	VARCHAR(100)	Manas Ranjan Patra'
Text	TEXT	This is a long description...'
Date	DATE	2025-01-11'
Time	TIME	16:35:00'
Datetime	DATETIME	2025-01-11 16:35:00'
Binary	BLOB	binary image / file data



```sql
CREATE TABLE sample_example (
    id INT,
    price DECIMAL(10,2),
    is_active BOOLEAN,
    name CHAR(10),
    full_name VARCHAR(100),
    description TEXT,
    order_date DATE,
    order_time TIME,
    created_at DATETIME,
    image BLOB
);
```

MySQL Constraints

Constrains are rules applied on columns of a table to ensure validity, accuracy and integrity of data. They act like data guards that restrict invalid entries.

Types of constrain:

1. NOT NULL: Ensures a column cannot have NULL values.

2. UNIQUE: Ensures all values in a column are unique.

3. Primary key: Combination of NOT NULL and UNEQUE Identifiers each record uniquely.

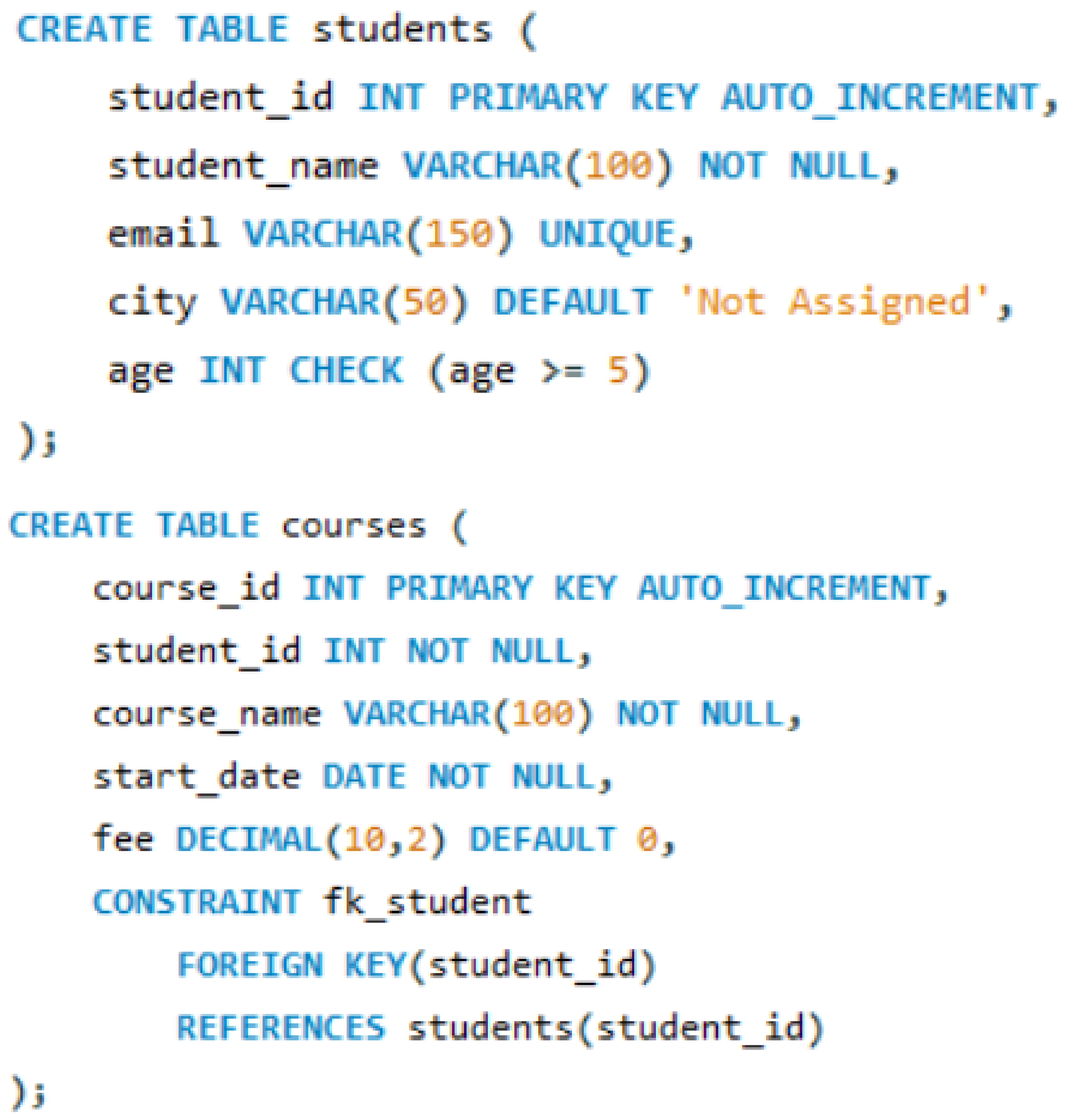
4. FOREIGN KEY: Enforces link between tables.

5. CHECK: Restricts values within a condition.

6. DEFAULT: Assigns a default value if none is provided.

7. INDEX: (not a constrain) Improves search performance.

Example

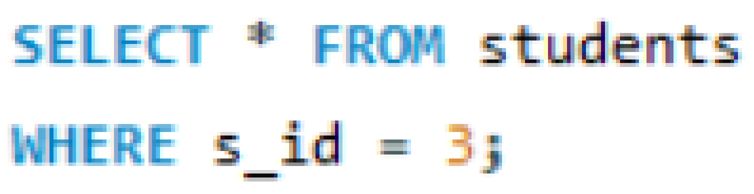


```sql
CREATE TABLE students (
    student_id INT PRIMARY KEY AUTO_INCREMENT,
    student_name VARCHAR(100) NOT NULL,
    email VARCHAR(150) UNIQUE,
    city VARCHAR(50) DEFAULT 'Not Assigned',
    age INT CHECK (age >= 5)
);

CREATE TABLE courses (
    course_id INT PRIMARY KEY AUTO_INCREMENT,
    student_id INT NOT NULL,
    course_name VARCHAR(100) NOT NULL,
    start_date DATE NOT NULL,
    fee DECIMAL(10,2) DEFAULT 0,
    CONSTRAINT fk_student
        FOREIGN KEY(student_id)
        REFERENCES students(student_id)
);
```

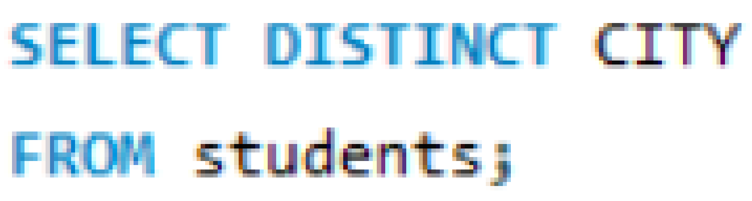
Filtering Clauses

1. WHERE



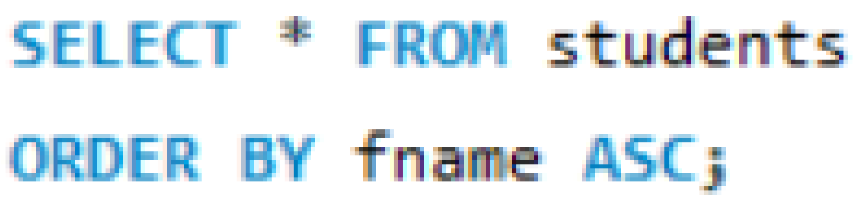
```
SELECT * FROM students
WHERE s_id = 3;
```

2. DISTINCT

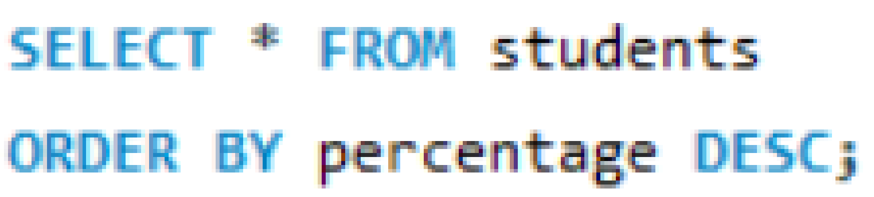


```
SELECT DISTINCT CITY
FROM students;
```

3. ORDER BY [ASC, DESC]

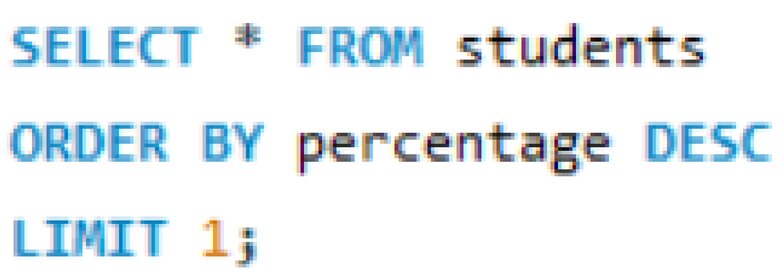


```
SELECT * FROM students
ORDER BY fname ASC;
```



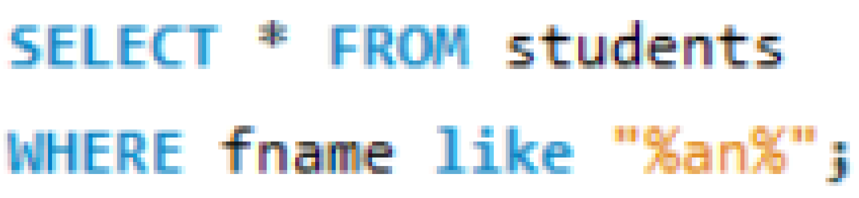
```
SELECT * FROM students
ORDER BY percentage DESC;
```

4. LIMIT



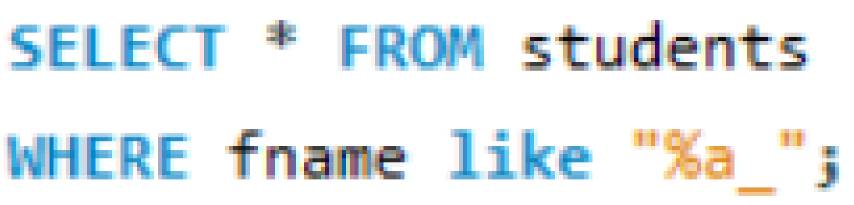
```
SELECT * FROM students
ORDER BY percentage DESC
LIMIT 1;
```

5. LIKE



```
SELECT * FROM students
WHERE fname like "%an%";
```

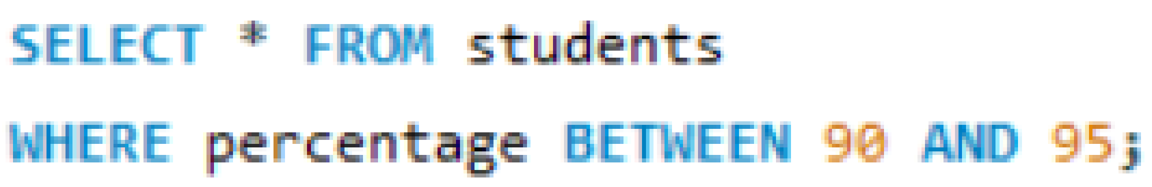
“%A%”



```
SELECT * FROM students
WHERE fname like "%a_";
```

“%A_”

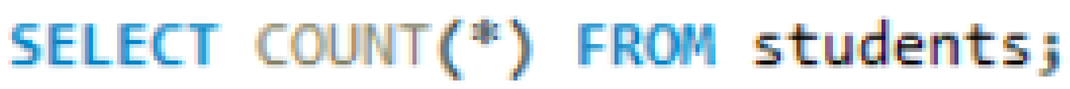
6. BETWEEN



```
SELECT * FROM students
WHERE percentage BETWEEN 90 AND 95;
```

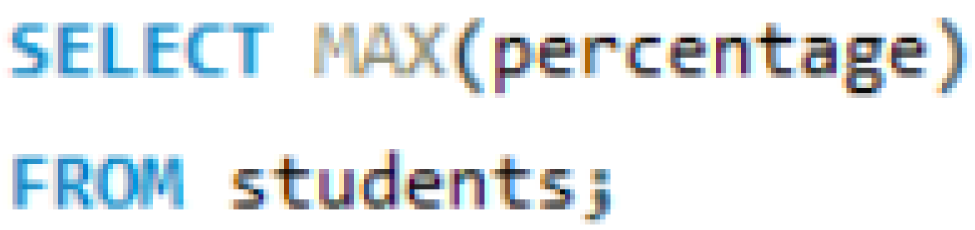
AGGREGATE FUNCTION

1. COUNT( )



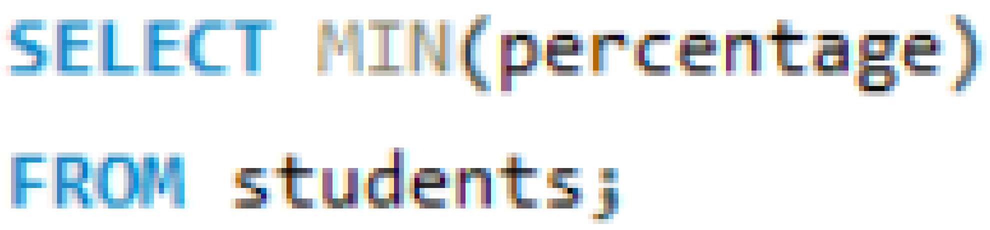
```
SELECT COUNT(*) FROM students;
```

2. MAX( )



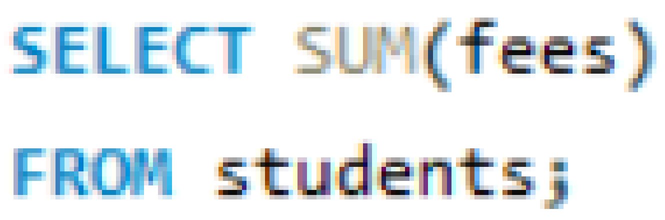
```
SELECT MAX(percentage)
FROM students;
```

3. MIN( )



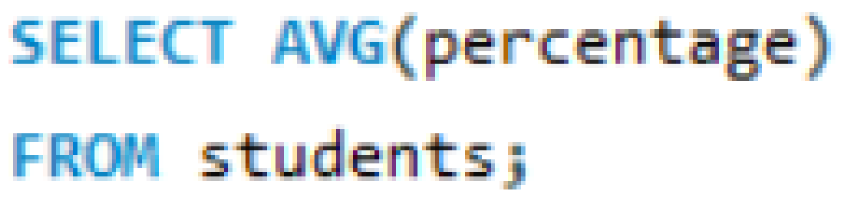
```
SELECT MIN(percentage)
FROM students;
```

4. SUM( )



```
SELECT SUM(fees)
FROM students;
```

5. AVG( )



```
SELECT AVG(percentage)
FROM students;
```

Thank You