# SQL Query Processing

After creating the database school, the table student and fees are created and the data is stored into it.

**School** → **Database**

**student** → **Table 1**

| Rollno | Name | Gender | Marks | DOB |
|---|---|---|---|---|
| 1 | PRATIK | M | 95 | 22-Apr-2003 |
| 2 | GARIMA | F | 85 | 05-Mar-2004 |
| 3 | KHUSHI | F | 92 | 24-Jan-2003 |
| 4 | BABU | NULL | NULL | 15-Aug-2000 |
| 5 | AMIT | M | 60 | 26-Jan-2002 |
| 6 | SHARMILA | F | 90 | 12-Dec-2003 |

**fees** → **Table 2**

| Rollno | Name | Fees | Attendance |
|---|---|---|---|
| 1 | PRATIK | 5250.00 | 98 |
| 2 | GARIMA | 1250.00 | 70 |
| 4 | BABU | 1250.00 | 75 |
| 6 | SHARMILA | 5250.00 | 85 |

Now we will learn query processing on the table student and fees. Retrieving and display the information from one or more database tables is done using **SELECT** command.

1. **Retrieving data from a table by SELECT command:** It is used to retrieve an information asked for by using the command **SELECT**.

   Syntax:SELECT[<col-name1>,<col-name2>…] FROM <table name>;

   Example 1:

   **mysql>SELECT Rollno, Name, Marks FROM student;**

   Result 1:

| Rollno | Name | Marks |
|---|---|---|
| 1 | PRATIK | 95 |
| 2 | GARIMA | 85 |
| 3 | KHUSHI | 92 |
| 4 | BABU | NULL |
| 5 | AMIT | 60 |
| 6 | SHARMILA | 90 |

   Example 2:

**mysql>SELECT * FROM student;**

Result 2:

| Rollno | Name | Gender | Marks | DOB |
|---|---|---|---|---|
| 1 | PRATIK | M | 95 | 22-Apr-2003 |
| 2 | GARIMA | F | 85 | 05-Mar-2004 |
| 3 | KHUSHI | F | 92 | 24-Jan-2003 |
| 4 | BABU | NULL | NULL | 15-Aug-2000 |
| 5 | AMIT | M | 60 | 26-Jan-2002 |
| 6 | SHARMILA | F | 90 | 12-Dec-2003 |

2. **SELECT with DISTINCT keyword:** It is used to retrieve an information by eliminating duplicate rows.

Syntax: SELECT DISTINCT <col-name> FROM <table name>;

Example :

**mysql>SELECT DISTINCT Gender FROM student;**

Result :

| Gender |
|---|
| M |
| F |
| NULL |

3. **SELECT with WHERE clause:** It is used to search/ retrieve specific information by using **WHERE** clause in the **SELECT** command.

Syntax: SELECT [<col-name1>,….] FROM <table name>
WHERE<condition>;

Example 1 :

```
mysql>SELECT Rollno, Name, Marks
      FROM student  WHERE Marks>=90;
```

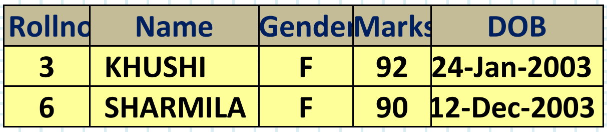
Result 1:

| Rollno | Name | Marks |
|---|---|---|
| 1 | PRATIK | 95 |
| 3 | KHUSHI | 92 |
| 6 | SHARMILA | 90 |

Example 2 :

```
mysql>SELECT * FROM student
      WHERE (Marks>=90 AND Gender='F');
```

Result 2:



| Rollnc | Name | Gender | Marks | DOB |
|---|---|---|---|---|
| 3 | KHUSHI | F | 92 | 24-Jan-2003 |
| 6 | SHARMILA | F | 90 | 12-Dec-2003 |

.

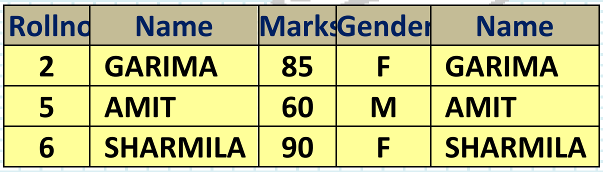
4. **SELECT with WHERE clause, BETWEEN..AND clause:** It retrieves those records/ rows, where the value/data is satisfied between two given values. Given values are also included. For that **BETWEEN..AND** clause with **WHERE** clause is used in the **SELECT** command.

Syntax: SELECT [<col-name1>,….] FROM <table name>
WHERE <col-name>
BETWEEN<value1>AND<value2>;

Example 1 :

```
mysql>SELECT Rollno, Name, Marks,Gender, Name
      FROM student
      WHERE Marks BETWEEN 60 AND 90;
```

Result 1:

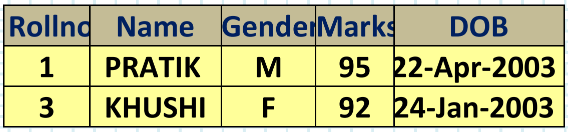


| Rollnc | Name | Marks | Gender | Name |
|---|---|---|---|---|
| 2 | GARIMA | 85 | F | GARIMA |
| 5 | AMIT | 60 | M | AMIT |
| 6 | SHARMILA | 90 | F | SHARMILA |

Example 2 :

```
mysql>SELECT * FROM student
      WHERE Marks NOT BETWEEN 60 AND 90;
```

Result 2:



| Rollnc | Name | Gender | Marks | DOB |
|---|---|---|---|---|
| 1 | PRATIK | M | 95 | 22-Apr-2003 |
| 3 | KHUSHI | F | 92 | 24-Jan-2003 |

.

. **SELECT with WHERE clause, IN clause:** It retrieves those records/ rows where the value/data is present in the mentioned column. For that **IN** clause with **WHERE** clause is used in the **SELECT** command.

Syntax: SELECT [<col-name1>,….] FROM <table name>
WHERE <col-name>
IN[<data1, data2…>];

Example 1 :

```
mysql>SELECT Rollno, Name, Marks
      FROM student
      WHERE Name IN (‘GARIMA’,’AMIT’);
```

Result 1:

| Rollno | Name | Marks |
|---|---|---|
| 2 | GARIMA | 85 |
| 5 | AMIT | 60 |

Example 2 :

```
mysql> SELECT Rollno, Name, Marks FROM student
       WHERE Name NOT IN (‘GARIMA’,’AMIT’);
```

Result 2 :

| Rollno | Name | Marks |
|---|---|---|
| 1 | PRATIK | 95 |
| 3 | KHUSHI | 92 |
| 4 | BABU | NULL |
| 6 | SHARMILA | 90 |

Before going to next clause we must know about **wildcard characters**. There are two important type of wildcard card characters.

%(Percentage symbol) - **Represent multiple characters.**

_ (Under Score symbol) - **Represent single character.**

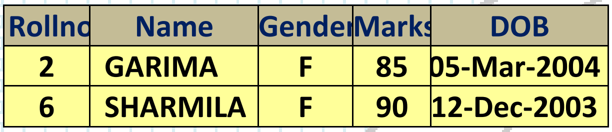
6. **SELECT with WHERE clause, LIKE clause:** It retrieve those records where condition of wildcard character satisfied(matching) with the data in the column. For that **LIKE** clause with **WHERE** clause is used in the **SELECT** command.

Syntax: SELECT [<col-name1>,….] FROM <table name>
WHERE <col-name>
LIKE<wildcard characters>

Example 1 :

**mysql>SELECT * FROM student WHERE Name LIKE "%A" ;**

Result 1:

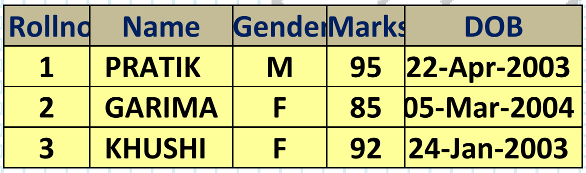


| Rollno | Name | Gender | Marks | DOB |
|---|---|---|---|---|
| 2 | GARIMA | F | 85 | 05-Mar-2004 |
| 6 | SHARMILA | F | 90 | 12-Dec-2003 |

Example 2 :

**mysql>SELECT * FROM student**
**WHERE Name LIKE "_ _ _ _ _ _" ;**

Result 2 :

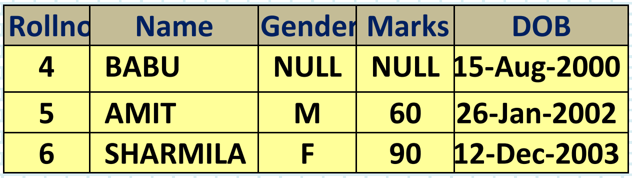


| Rollno | Name | Gender | Marks | DOB |
|---|---|---|---|---|
| 1 | PRATIK | M | 95 | 22-Apr-2003 |
| 2 | GARIMA | F | 85 | 05-Mar-2004 |
| 3 | KHUSHI | F | 92 | 24-Jan-2003 |

Example 3 :

**mysql>SELECT * FROM student**
**WHERE Name NOT LIKE "_ _ _ _ _ _" ;**

Result 3 :

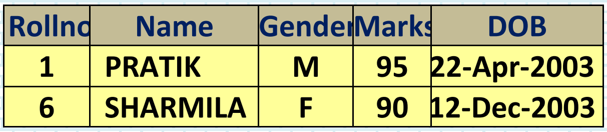


| Rollno | Name | Gender | Marks | DOB |
|---|---|---|---|---|
| 4 | BABU | NULL | NULL | 15-Aug-2000 |
| 5 | AMIT | M | 60 | 26-Jan-2002 |
| 6 | SHARMILA | F | 90 | 12-Dec-2003 |

Example 4 :

**mysql>SELECT * FROM student**
**WHERE Name LIKE “_ _ A%” ;**

Result 4 :

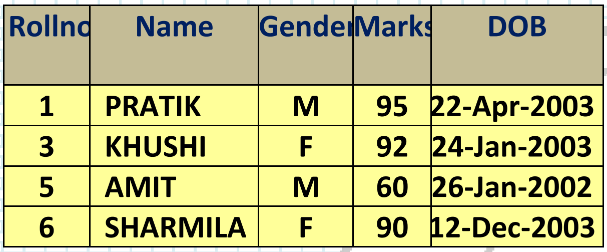


| Rollno | Name | Gender | Marks | DOB |
|---|---|---|---|---|
| 1 | PRATIK | M | 95 | 22-Apr-2003 |
| 6 | SHARMILA | F | 90 | 12-Dec-2003 |

Example 5 :

**mysql>SELECT * FROM student**
**WHERE Name NOT LIKE “_A%” ;**

Result 5 :



| Rollno | Name | Gender | Marks | DOB |
|---|---|---|---|---|
| 1 | PRATIK | M | 95 | 22-Apr-2003 |
| 3 | KHUSHI | F | 92 | 24-Jan-2003 |
| 5 | AMIT | M | 60 | 26-Jan-2002 |
| 6 | SHARMILA | F | 90 | 12-Dec-2003 |

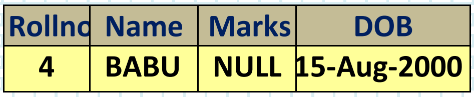
7. **SELECT with WHERE clause, IS NULL clause:** It retrieves those records/ rows where the particular column has the NULL value. For that **IS NULL** clause is used with **WHERE** clause in the **SELECT** command.

Syntax: SELECT [<col-name1>,….] FROM <table name>
WHERE <col-name>IS NULL;

Example 1 :

**mysql>SELECT Rollno, Name, Marks,DOB**
**FROM student WHERE Marks IS NULL;**

Result 1:

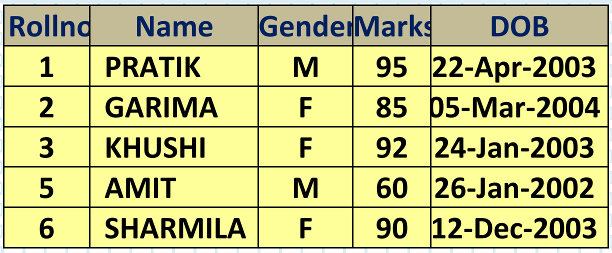


| Rollno | Name | Marks | DOB |
|---|---|---|---|
| 4 | BABU | NULL | 15-Aug-2000 |

Example 2 :

**mysql>SELECT Rollno, Name, Gender,Marks, DOB**
**FROM student WHERE Marks IS NOT NULL;**

Result 2 :



| Rollno | Name | Gender | Marks | DOB |
|---|---|---|---|---|
| 1 | PRATIK | M | 95 | 22-Apr-2003 |
| 2 | GARIMA | F | 85 | 05-Mar-2004 |
| 3 | KHUSHI | F | 92 | 24-Jan-2003 |
| 5 | AMIT | M | 60 | 26-Jan-2002 |
| 6 | SHARMILA | F | 90 | 12-Dec-2003 |

8. **SELECT with ORDER BY clause:** It allows sorting (ascending or descending) of the query results by one or more columns. To get the sorted output, **ORDER BY** clause is used with **WHERE** clause in the **SELECT** command.

**Note:**

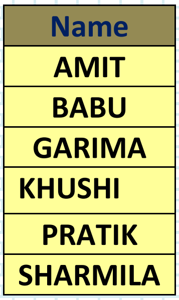
- The default order is ascending order.
- For ascending order syntax used is **ASC**.
- For descending order syntax used is **DESC**.
- The **ORDER BY** clause does not sort the data in the actual table, only the result that appeared is sorted.

Syntax: SELECT <col-list> FROM <table name>
WHERE <condition>ORDER BY<col-name>;

Example 1 :

```
mysql>SELECT Name FROM student
      ORDER BY Name ~~ASC~~;
```

Result 1:



| Name |
|---|
| AMIT |
| BABU |
| GARIMA |
| KHUSHI |
| PRATIK |
| SHARMILA |

Example 2 :

```
mysql>SELECT Rollno, Name, Gender, Marks
      FROM student WHERE Marks>70
      ORDER BY Marks DESC;
```

Result 2 :

| Rollno | Name | Gender | Marks |
|---|---|---|---|
| 2 | GARIMA | F | 85 |
| 6 | SHARMILA | F | 90 |
| 3 | KHUSHI | F | 92 |
| 1 | PRATIK | M | 95 |

9. **SELECT with GROUP BY clause:** The aggregate function (SUM , AVG , MAX , MIN , and COUNT) that appears with the **GROUP BY** clause in SELECT command provides information about each group. It returns one row for each group.

Syntax:SELECT<col-name>, FUNCTION name(<col-name>)
FROM <table name>
GROUP BY <col-name>;

Example :

```
mysql>SELECT Gender, AVG(Marks)
      FROM student
      GROUP BY Gender;
```

Result :

| Gender | AVG(Marks) |
|---|---|
| NULL | NULL |
| F | 89.0 |
| M | 77.5 |

F = 85+92+90 =267 267/3 = 89.0
M = 95 +60 = 155 155/2 = 77.5

10. **SELECT with HAVING clause:** This clause applies to group rather than rows. (Group means all Male students or Female Students or all PGTs etc.).

**Now find the max marks obtained by the Boys and Girls.**

Syntax:
```
SELECT<col-name>, FUNCTION name(<col-name>)
 FROM <table name> GROUP BY <col-name>
 HAVING <col-name> CLAUSE (‘type of condition')
```

Example 1 :

```
mysql>SELECT Gender, Max(Marks)
      FROM student
      GROUP BY Gender
      HAVING Gender IN (‘M’,’F’);
```

Result 1:

| Gender | MAX(Marks) |
|---|---|
| F | 92 |
| M | 95 |

Example 2:

```
mysql>SELECT Gender, Max(Marks)
      FROM student
      GROUP BY Gender
      HAVING Gender LIKE(‘_’);
```

Result 2:

| Gender | MAX(Marks) |
|---|---|
| F | 92 |
| M | 95 |

**Note:** The **HAVING** search condition are almost identical as **WHERE** search conditions. The only difference is that **WHERE** search conditions cannot include aggregate functions while **HAVING** search conditions include these functions.

# Thank you