

29	Alias Name	SELECT column1 AS Alias1, column2 AS Alias2...FROM table_name;	SELECT FID AS Faculty_ID, FirstName AS Faculty_Name FROM faculty;
	<pre>mysql> SELECT FID AS Faculty_ID, FirstName AS Faculty_Name -> FROM faculty; +-----+-----+ Faculty_ID Faculty_Name +-----+-----+ 201 Anisha 202 Ridhi 203 Raktim 204 Aradhya 205 Animesh +-----+-----+ 5 rows in set (0.00 sec)</pre>		
30	Between Operator	SELECT <i>column_name(s)</i> FROM <i>table_name</i> WHERE <i>column_name</i> BETWEEN <i>value1</i> AND <i>value2</i> ;	SELECT * FROM student WHERE student_id BETWEEN 1 and 4;
	<pre>mysql> SELECT * FROM student -> WHERE student_id BETWEEN 1 and 4; +-----+-----+-----+ student_id name course +-----+-----+-----+ 1 Akash CSE 2 Seema ECE 4 Ridhi ML +-----+-----+-----+ 3 rows in set (0.20 sec)</pre>		

31	Addition of New Column and its Value	First initialize the column value as NULL and then Update the values.	<pre>mysql> ALTER TABLE student -> ADD gpa decimal(3,2) NULL; Query OK, 0 rows affected (1.22 sec) Records: 0 Duplicates: 0 Warnings: 0 mysql> UPDATE student SET gpa = 7.65 WHERE student_id =1; Query OK, 1 row affected (0.10 sec) Rows matched: 1 Changed: 1 Warnings: 0 mysql> UPDATE student SET gpa = 6.57 WHERE student_id =2; Query OK, 1 row affected (0.09 sec) Rows matched: 1 Changed: 1 Warnings: 0 mysql> UPDATE student SET gpa = 5.57 WHERE student_id =4; Query OK, 1 row affected (0.09 sec) Rows matched: 1 Changed: 1 Warnings: 0 mysql> UPDATE student SET gpa = 8.78 WHERE student_id =5; Query OK, 1 row affected (0.09 sec) Rows matched: 1 Changed: 1 Warnings: 0</pre>
32	In Operator	SELECT column_name(s) FROM table_name WHERE column_name IN (value1, value2, ...);	<pre>mysql> select * from student -> where course IN ('CSE','ML'); +-----+-----+-----+-----+ student_id name course gpa +-----+-----+-----+-----+ 1 Akash CSE 7.65 4 Ridhi ML 5.57 5 Riju ML 8.78 6 Rakesh CSE NULL +-----+-----+-----+-----+ 4 rows in set (0.03 sec)</pre>

33	Like Operator	<p>SELECT * FROM Customers</p> <ol style="list-style-type: none"> 1. WHERE CustomerName LIKE 'a%'; - Finds any values that start with "a" 2. WHERE CustomerName LIKE '%a' - Finds any values that end with "a" 3. WHERE CustomerName LIKE '%or%' - Finds any values that have "or" in any position 4. WHERE CustomerName LIKE '_r%' - Finds any values that have "r" in the second position 5. WHERE CustomerName LIKE 'a_%' - Finds any values that start with "a" and are at least 2 characters in length 6. WHERE CustomerName LIKE 'a__%' Finds any values that start with "a" and are at least 3 characters in length 7. WHERE ContactName LIKE 'a%o' Finds any values that start with "a" and ends with "o" 	<pre>mysql> SELECT * from student -> WHERE name LIKE 'R%' -> ;</pre> <table border="1"> <thead> <tr> <th>student_id</th><th>name</th><th>course</th><th>gpa</th></tr> </thead> <tbody> <tr> <td>4</td><td>Ridhi</td><td>ML</td><td>5.57</td></tr> <tr> <td>5</td><td>Riju</td><td>ML</td><td>8.78</td></tr> <tr> <td>6</td><td>Rakesh</td><td>CSE</td><td>NULL</td></tr> </tbody> </table> <p>3 rows in set (0.03 sec)</p> <pre>mysql> SELECT * from student -> WHERE name LIKE '%h';</pre> <table border="1"> <thead> <tr> <th>student_id</th><th>name</th><th>course</th><th>gpa</th></tr> </thead> <tbody> <tr> <td>1</td><td>Akash</td><td>CSE</td><td>7.65</td></tr> <tr> <td>6</td><td>Rakesh</td><td>CSE</td><td>NULL</td></tr> </tbody> </table> <p>2 rows in set (0.00 sec)</p>	student_id	name	course	gpa	4	Ridhi	ML	5.57	5	Riju	ML	8.78	6	Rakesh	CSE	NULL	student_id	name	course	gpa	1	Akash	CSE	7.65	6	Rakesh	CSE	NULL
student_id	name	course	gpa																												
4	Ridhi	ML	5.57																												
5	Riju	ML	8.78																												
6	Rakesh	CSE	NULL																												
student_id	name	course	gpa																												
1	Akash	CSE	7.65																												
6	Rakesh	CSE	NULL																												
34	Arithmetic Operators	+, /, *, -	<p>SELECT FirstName, LastName, (Bonus+Salary) as Total_Sal FROM faculty WHERE (Bonus+ Salary)<50000;</p>																												

```
mysql> SELECT FirstName, LastName, (Bonus+Salary) as Total_Sal
-> FROM faculty
-> WHERE (Bonus+ Salary)<50000;

+-----+-----+-----+
| FirstName | LastName | Total_Sal |
+-----+-----+-----+
| Aradhya   | Choudhury | 43800 |
| Animesh   | Yadav    | 49400 |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

OPERATORS IN SQL*PLUS

Type	Symbol / Keyword	Where to use
Arithmetic	+, -, *, /	To manipulate numerical column values, WHERE clause
Comparison	=, !=, <, <=, >, >=, between, not between, in, not in, like, not like	WHERE clause
Logical	and, or, not	WHERE clause, Combining two queries

Functions:

- Single Row Functions
- Group functions

Single Row Functions

- SQL supplies a rich library of in-built functions which can be employed for various tasks. The essential capabilities of a functions can be the **case conversion of strings**, **in-string** or **substring operations**, **mathematical computations on numeric data**, and **date operations on date type values**.
-
- SQL Functions optionally take arguments from the user and mandatorily return a value.

On a broader category, there are two types of functions :-

Single Row functions - Single row functions are the one who work on single row and return one output per row. For example, length and case conversion functions are single row functions.

Multiple Row functions - Multiple row functions work upon group of rows and return one result for the complete set of rows. They are also known as Group Functions.

Single row functions are used in SELECT command and included in WHERE clause, Order by clause

35	UPPER / UCASE	SELECT UPPER(columnname) FROM tablename WHERE condition;	mysql> SELECT UPPER(name), LOWER(course) -> FROM student -> WHERE student_id = 6; +-----+ UPPER(name) LOWER(course) +-----+ RAKESH cse +-----+ 1 row in set (0.23 sec)
	LOWER	SELECT LOWER(columnname) FROM tablename WHERE condition;	
	TRIM()- Removes leading and trailing spaces from a string.	SELECT TRIM("string to trim") AS TrimmedString;	mysql> SELECT TRIM(" WELCOME ") AS TrimmedString; +-----+ TrimmedString +-----+ WELCOME +-----+ 1 row in set (0.08 sec)