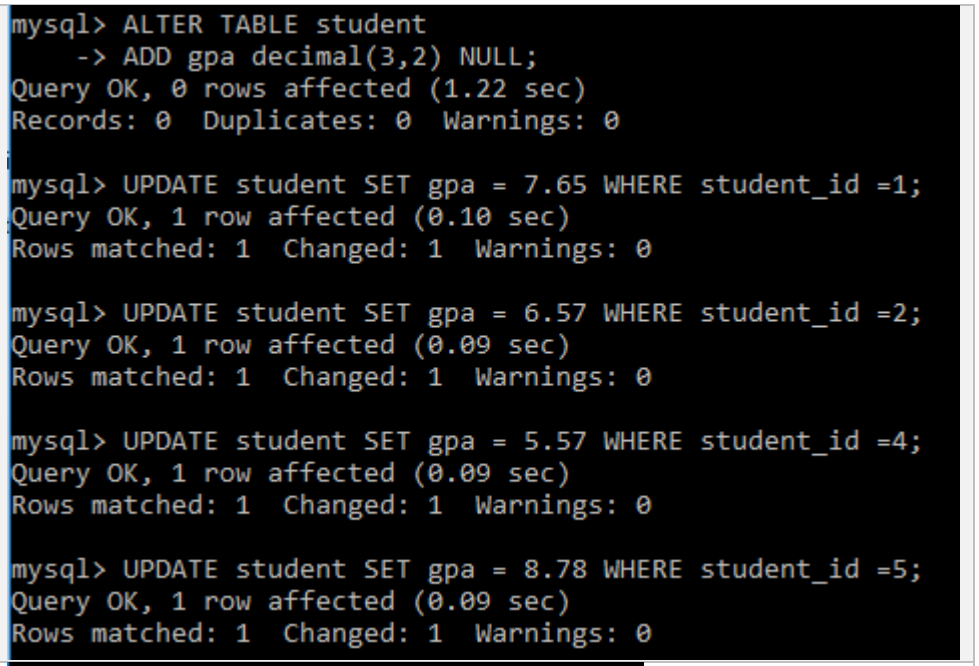
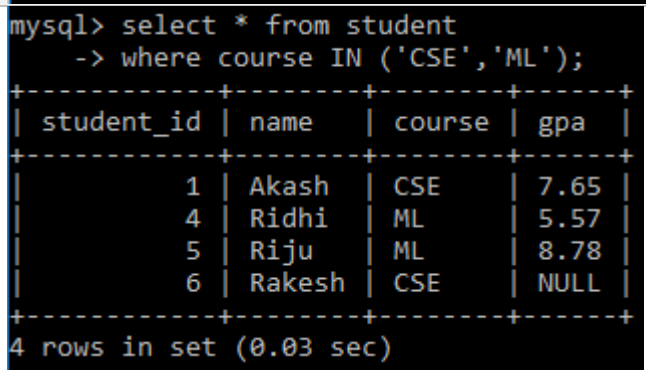
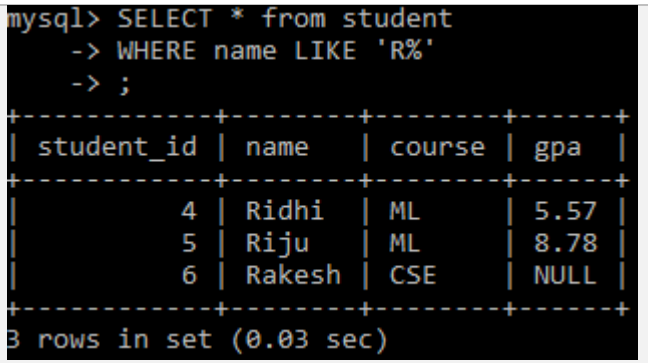
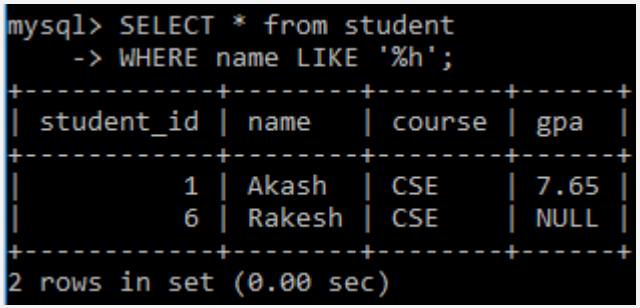
29	**Alias Name**	SELECT column1 AS Alias1, column2 AS Alias2…FROM table_name;	SELECT FID AS Faculty_ID, FirstName AS Faculty_Name FROM faculty;

```
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)
```

30	**Between Operator**	SELECT *column_name(s)* FROM *table_name* WHERE *column_name* BETWEEN *value1* AND *value2;*	SELECT * FROM student WHERE student_id BETWEEN 1 and 4;

```
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)
```

31	**Addition of New Column and its Value**	First initialize the column value as NULL and then Update the values.	
32	**In Operator**	SELECT column_name(s) FROM table_name WHERE column_name IN (value1, value2, ...);	

33	**Like Operator**	SELECT * FROM Customers 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"	 
34	**Arithmetic Operators**	+, /, *. –	SELECT FirstName, LastName, (Bonus+Salary) as Total_Sal FROM faculty WHERE (Bonus+ Salary)<50000;

```
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

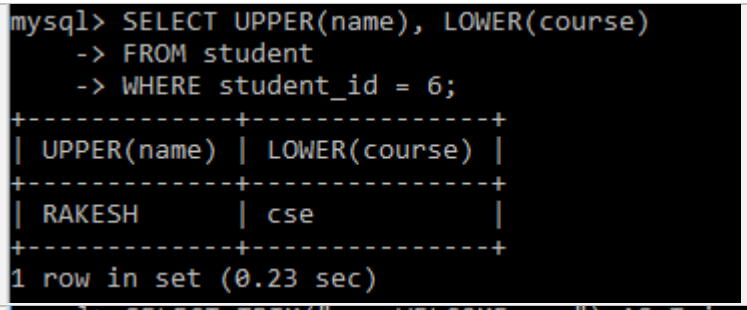
- SQL supplies a rich library of in-built functions which can be employed for various tasks. The essential capabilities of a functions can be the **case conversion of strings**, **in-string** or **substring operations, mathematical computations on numeric data**, and **date operations on date type values**.
-
- 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;	
	LOWER	SELECT LOWER(columnname) FROM tablename WHERE condition;	
	TRIM()- Removes leading and trailing spaces from a string.	SELECT TRIM("string to trim") AS TrimmedString;