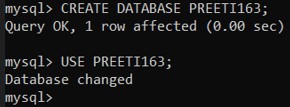
**AIM:-Creation of database of student and faculty table.**

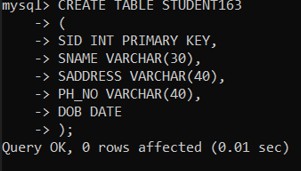
**Student (sid, sname, address, ph\_no,DOB)**

**Faculty (fid, sid, fname, dept, DOJ)**

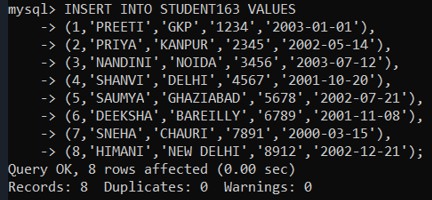
**DATABASE CREATION**



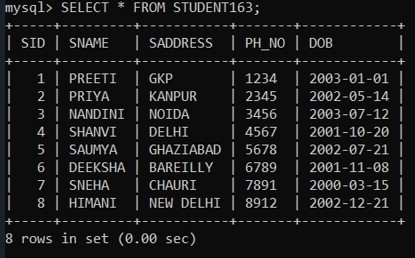
**STUDENT TABLE CREATION**



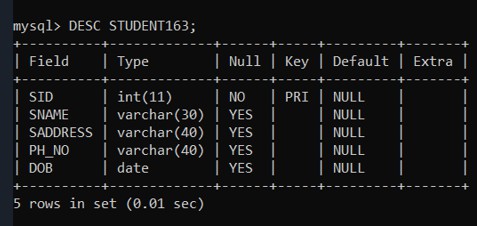
**VALUES INSERTION IN STUDENT TABLE**



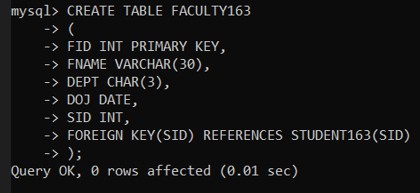
**STUDENT TABLE DISPLAY**



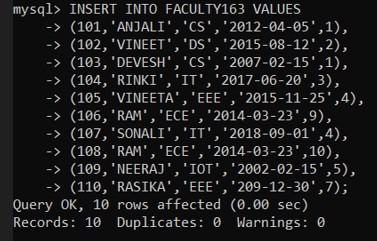
**STUDENT TABLE DESCRIPTION**



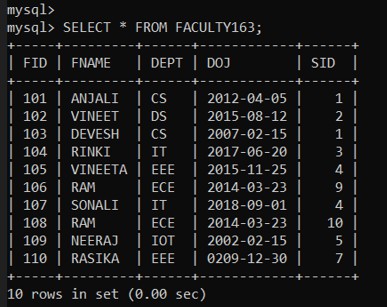
**FACULTY TABLE CREATION**



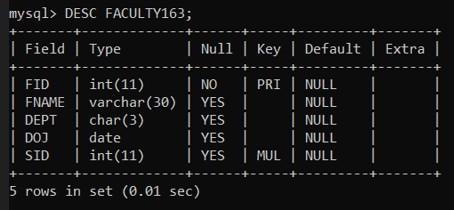
**VALUES INSERTION IN FACULTY TABLE**



**FACULTY TABLE DISPLAY**



**FACULTY TABLE DESCRIPTION**

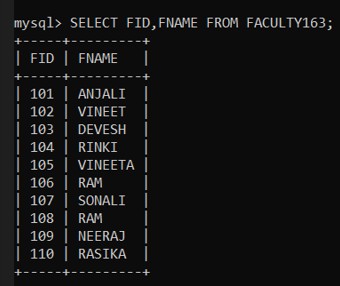


**To perform all the basic SQL Operations on Student and Faculty Table.**

**SELECT Statement:** The **SELECT** statement is used to select data from a database.

**SYNTAX:** SELECT column1, column2,....

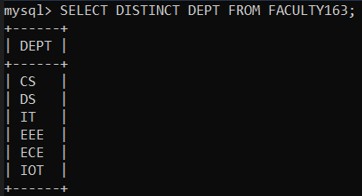
FROM table\_name;



**DISTINCT Statement:** The **SELECT DISTINCT** statement is used to return only distinct (different) values.

**SYNTAX:** SELECT DISTINCT column1, column2,..

FROM table\_name;

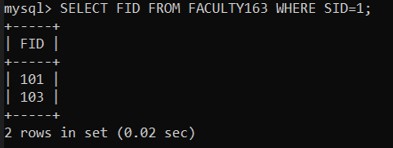


**WHERE Clause:**The **WHERE** clause is used to filter records.It is used to extract

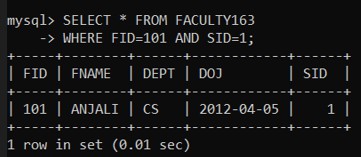
that fulfill a specified condition.

**SYNTAX:** SELECT column1, column2, …

FROM table\_name WHERE condition;

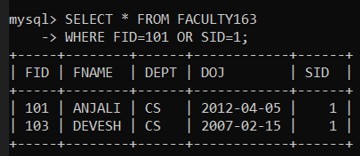


**AND Operator:**The **AND** operator displays a record if all the conditions separated by **AND** are TRUE.

**SYNTAX:** SELECT column1, column2, …FROM table\_name WHERE condition1 AND condition2 AND condition3 …;

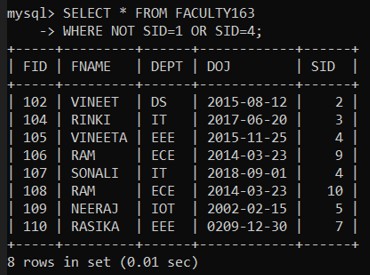
**OR Operator:** The **OR** operator displays a record if any of the conditions separated by **OR** is TRUE.

**SYNTAX:** SELECT column1, column2,.. FROM table\_name WHERE condition1 OR condition2 OR condition3 ...;



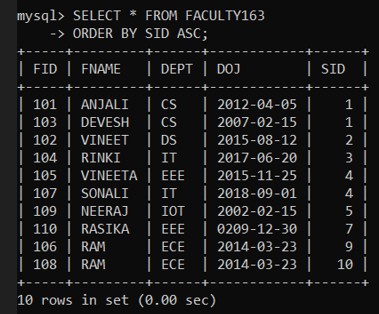
**NOT Operator:**The **NOT** operator displays a record if the condition(s) is NOT TRUE.

**SYNTAX:** SELECT column1, column2, …

FROM table\_name WHERE NOT condition;

**ORDER BY KEYWORD:**The **ORDER BY** keyword is used to sort the result-set in ascending or descending order.

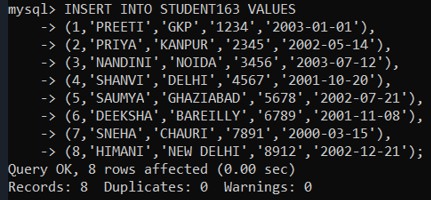
**SYNTAX:** SELECT column1, column2, … FROM table\_name ORDER BY column1, column2, ... ASC|DESC;



**INSERT INTO:**The **INSERT INTO** statement is used to insert new records in a table.

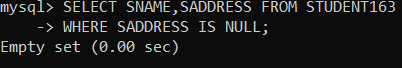
**SYNTAX:** INSERT INTO table\_name

VALUES (value1, value2, value3, ...);



**NULL VALUES:**A field with a NULL value is a field with no value.

If a field in a table is optional, it is possible to insert a new record or update a record without adding a value to this field. Then, the field will be saved with a NULL value.

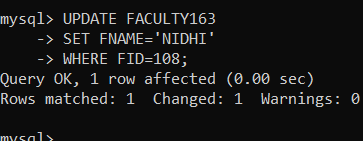
**SYNTAX:** SELECT column\_names FROM table\_name WHERE column\_name IS NULL;

**UPDATE Statement:**The **UPDATE** statement is used to modify the existing records in a table.

**SYNTAX:** UPDATE table\_name

SET column1 = value1, column2 = value2, …

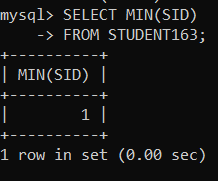
WHERE condition;



**MIN Function:**The **MIN()** function returns the smallest value of the selected column.

**SYNTAX:** SELECT MIN(column\_name)

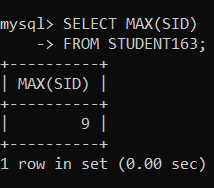
FROM table\_name WHERE condition;



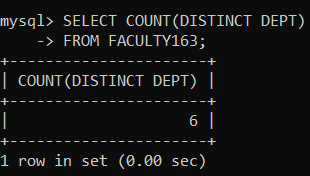
**MAX Function:**The **MAX()** function returns the largest value of the selected column.

**SYNTAX:** SELECT MAX(column\_name)

FROM table\_name WHERE condition;

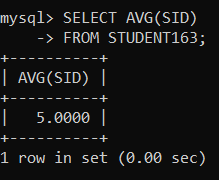


**COUNT() Function:**The **COUNT()** function returns the number of rows that matches a specified criterion.

**SYNTAX:** SELECT COUNT(column\_name) FROM table\_name WHERE condition;

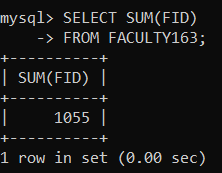
**AVG() Function:**The **AVG()** function returns the average value of a numeric column.

**SYNTAX:** SELECT AVG(column\_name)

FROM table\_name WHERE condition;

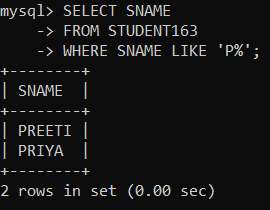
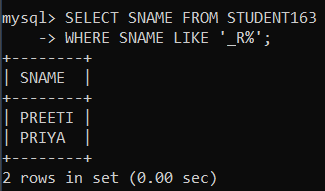
**SUM() Function:**The **SUM()** function returns the total sum of a numeric column.

**SYNTAX:** SELECT SUM(column\_name)

FROM table\_name WHERE condition;

**LIKE Operator:**The **LIKE** operator is used in a **WHERE** clause to search for a specified pattern in a column.The percent sign (%) represents zero, one, or multiple characters.

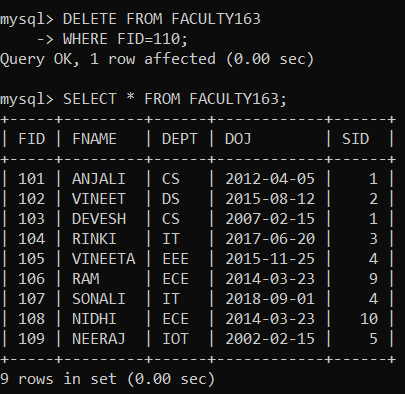
The underscore sign (\_) represents one, single character.

**SYNTAX:** SELECT column1, column2,.. FROM table\_name WHERE columnN LIKE pattern;

**DELETE Statement:**The **DELETE** statement is used to delete existing records in a table.

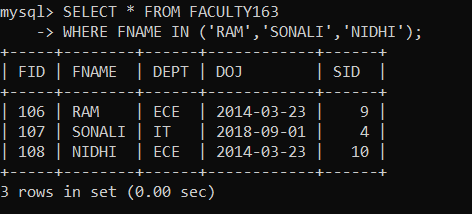
**SYNTAX:** DELETE FROM table\_name

WHERE condition;



**IN Operator:**The IN operator allows you to specify multiple values in a WHERE clause.The IN operator is a shorthand for multiple OR conditions.

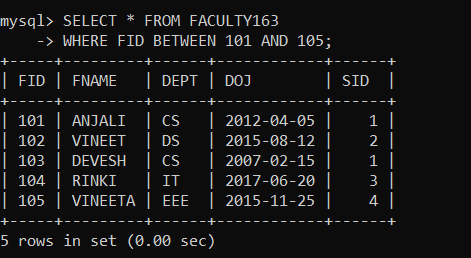
**SYNTAX:** SELECT column\_name(s) FROM table\_name WHERE column\_name IN (value1, value2, ...);



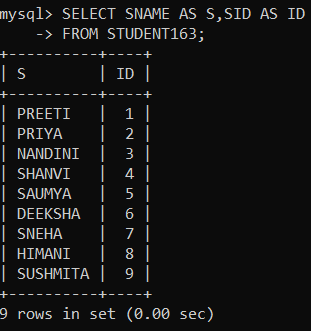
**BETWEEN Operator:**The BETWEEN operator selects values within a given range. The values can be numbers, text, or dates.

The BETWEEN operator is inclusive: begin and end values are included.

**SYNTAX:** SELECT column\_name(s)

FROM table\_name WHERE column\_name BETWEEN value1 AND value2;

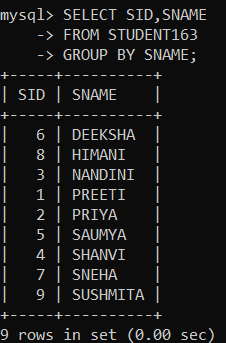
**ALIASES:**Aliases are used to give a table, or a column in a table, a temporary name. An alias only exists for the duration of that query. An alias is created with the AS keyword.

**SYNTAX:** SELECT column\_name AS alias\_name FROM table\_name;

**GROUP BY:**The GROUP BY statement groups rows that have the same values into summary rows. The GROUP BY statement is often used with aggregate functions (COUNT(), MAX(), MIN(), SUM(), AVG()) to group the result-set by one or more columns.

**SYNTAX:** SELECT column\_name(s)

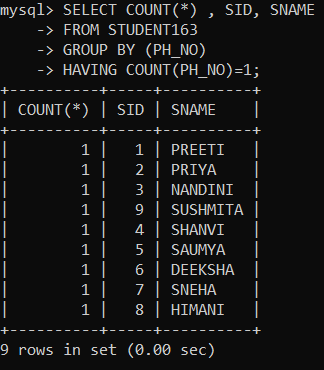
FROM table\_name WHERE condition GROUP BY column\_name(s)

ORDER BY column\_name(s);

**HAVING Clause:**The HAVING clause was added to SQL because the WHERE keyword cannot be used with aggregate function.

**SYNTAX:** SELECT column\_name(s) FROM table\_name WHERE condition

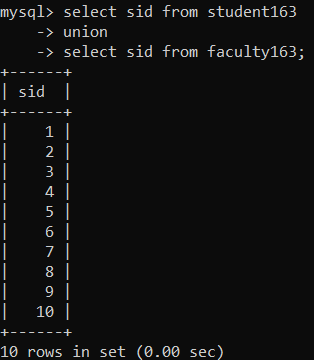
GROUP BY column\_name(s) HAVING condition

ORDER BY column\_name(s);

**UNION Operator:**The UNION operator is used to combine the result-set of two or more SELECT statements. Every SELECT statement within UNION must have the same number of columns The columns must also have similar data types.

**SYNTAX:**SELECT column\_name(s) FROM table1 UNION

SELECT column\_name(s) FROM table2;

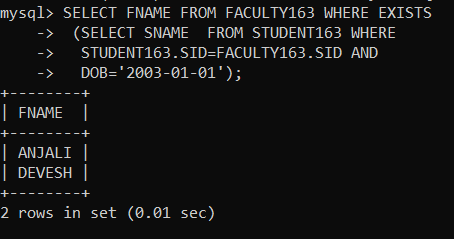


**EXISTS Operator:**

The EXISTS operator is used to test for the existence of any record in a subquery. The EXISTS operator returns TRUE if the subquery returns one or more records. **SYNTAX:**SELECT column\_name(s)

FROM table\_name WHERE EXISTS

(SELECT column\_name FROM table\_name WHERE condition);



**ANY Operator:**The ANY operator:

**:-**returns a boolean value as a result

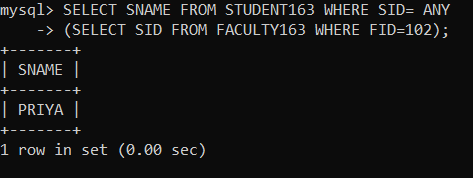
:-returns TRUE if ANY of the subquery values meet the condition

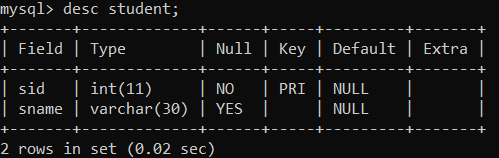
ANY means that the condition will be true if the operation is true for any of the values in the range.

**SYNTAX:**SELECT column\_name(s) FROM table\_name

WHERE column\_name operator ANY (SELECT column\_name

FROM table\_name WHERE condition);

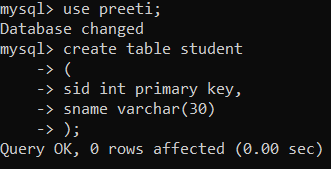


**AIM:To perform join operation on database.**

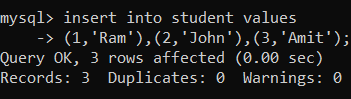
**DATABASE CREATION:**



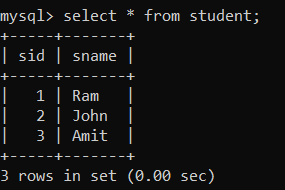
**CREATE TABLE (A):**



**INSERTION:**



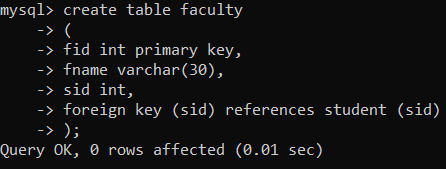
**DISPLAY:**



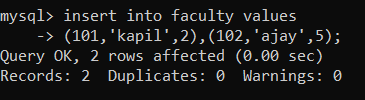
**DESCRIPTION:**

**CREATE TABLE (B):**

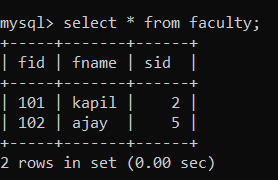
**CREATION**



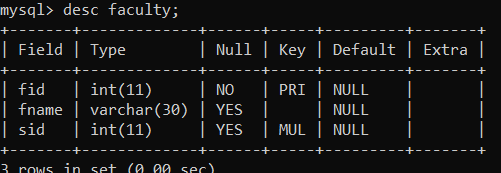
**INSERTION:**



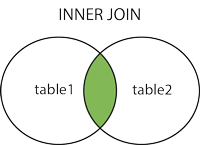
**DISPLAY**



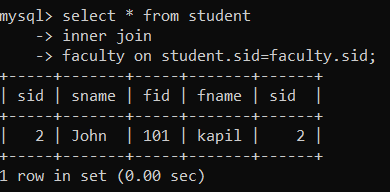
**DESCRIPTION**



**JOINS:**A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

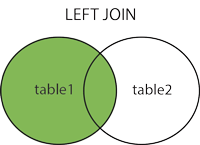
1. **INNER JOIN:**The INNER JOIN keyword selects records that have matching values in both Table.

**SYNTAX:** SELECT column\_name(s) FROM table1 INNER JOIN table2

ON table1.column\_name = table2.column\_name;

1. **LEFT JOIN:**The LEFT JOIN keyword returns all records from the left table (table1), and the matching records from the right table (table2).

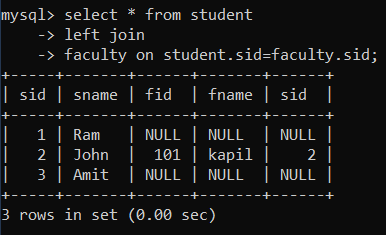
The result is 0 records from the right side, if there is no match.



**SYNTAX:** SELECT column\_name(s)

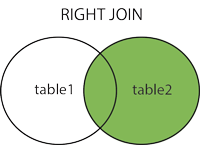
FROM table1 LEFT JOIN table2

ON table1.column\_name = table2.column\_name;

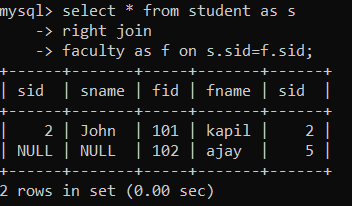


**RIGHT JOIN:**The **RIGHT JOIN** keyword returns all records from the right table (table2), and the matching records from the left table (table1).

The result is 0 records from the left side, if there is no match.



**SYNTAX:** SELECT column\_name(s) FROM table1 RIGHT JOIN table2

ON table1.column\_name = table2.column\_name;

**EXPERIMENT- 05**