

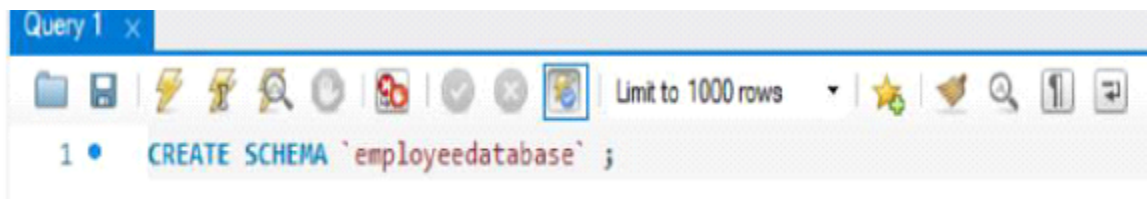
MySQL

MySQL is an open-source relational database management system which is supported by Oracle Company. It is fast, scalable, and an easy to use database management system in comparison with Microsoft SQL Server and Oracle Database.

MySQL Create Schema

A schema is a collection of database objects like tables, triggers, stored procedures, etc. A schema is connected with a user which is known as the schema owner

Syntax mysql> CREATE SCHEMA schemaname;



MySQL CREATE TABLE

MySQL allows us to create a table into the database by using the CREATE TABLE command. Following is a generic syntax for creating a MySQL table in the database.

```
CREATE TABLE [IF NOT EXISTS] table_name(  
  
    column_definition1,  
  
    column_definition2,  
  
    ..... ,  
  
    table_constraints );
```

```

Query 1 x
Limit to 1000 rows
1 CREATE TABLE `employee` (
2   `Emp_ID` INT NOT NULL,
3   `Emp_Name` VARCHAR(45) NOT NULL,
4   `Emp_Age` INT NOT NULL,
5   `Emp_Dep` VARCHAR(45) NOT NULL,
6   `Dep_ID` INT NOT NULL,
7   `Emp_Location` VARCHAR(45) NOT NULL,
8   PRIMARY KEY (`Emp_ID`))

```

MySQL INSERT Statement

The below is generic syntax of SQL INSERT INTO command to insert a single record in MySQL table:

```
INSERT INTO table_name ( field1, field2,...fieldN ) VALUES ( value1,
value2,...valueN );
```

The screenshot shows a MySQL IDE window with a query editor and a result grid. The query editor contains the following SQL statements:

```

1 INSERT INTO `employee` (`Emp_ID`, `Emp_Name`, `Emp_Age`, `Emp_Dep`, `Emp_Location`, `Dep_ID`,
2   `Emp_Salary`, `Emp_Address`)VALUES ('107', 'Ken', '24', 'IT', 'kerala', '112', '40000', 'xyz');
3 select * from `employee`;

```

The result grid displays the data from the `employee` table. The columns are: Emp_ID, Emp_Name, Emp_Age, Emp_Dep, Emp_Location, Dep_ID, Emp_Salary, and Emp_Address. The data rows are as follows:

Emp_ID	Emp_Name	Emp_Age	Emp_Dep	Emp_Location	Dep_ID	Emp_Salary	Emp_Address
101	Rani	32	HR	Banglore	111	25000	abcd
102	Aswathy	45	Sales	Chennai	113	30000	efg
103	Revathy	25	IT	Hydrebad	112	10000	hij
104	Kiran	27	Sales	Raipur	113	15000	klmn
105	Swetha	30	HR	Vizag	111	12000	pqrs
106	Reema	23	IT	Pune	112	40000	lmno
107	Ken	24	IT	kerala	112	40000	xyz

MySQL DELETE Statement

The following are the syntax that illustrates how to use the DELETE statement:

DELETE FROM table_name WHERE condition;

[illegible]

MySQL SELECT Statement

SELECT * FROM tables [WHERE conditions] [GROUP BY fieldName(s)]

[HAVING condition] [ORDER BY fieldName(s)] [OFFSET M][LIMIT N];

[illegible]

1 • `SELECT * FROM employeeDatabase.project;`

Result Grid	Filter Rows:	Edit:	Export/Import:	Wrap Cell
ID_Project	Project_Name	Project_Location	Project_Strength	
11	Health_Care	Banglore	5	
12	EPay	Hyderabad	7	
13	Stats	Chennai	3	
14	CSecure	Kerala	6	
* NULL	NULL	NULL	NULL	

TABLE DEPARTMENT

Result Grid	Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
111	HR	Banglore	11	
112	It	Chennai	12	
113	Sales	Pune	15	
114	Research	Hyderabad	13	

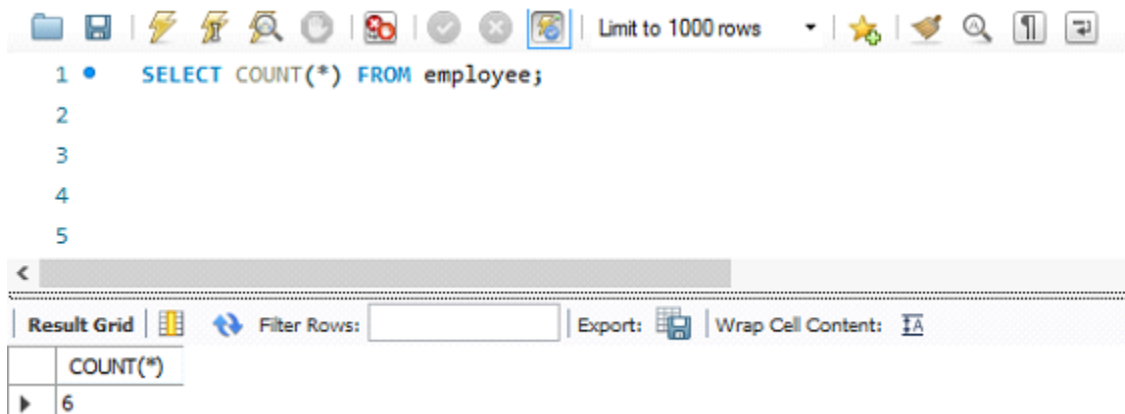
Result Grid	Filter Rows:	Edit:	Export/Import:	W
E_ID	Dependent_Name	Dependent_Relation		
101	Surabhi	wife		
102	Mahesh	Brother		
104	Rama Rao	father		
105	Sujatha	Mother		
* NULL	NULL	NULL		

SQL Aggregate Functions

SQL aggregation function is used to perform the calculations on multiple rows of a single column of a table. It returns a single value.

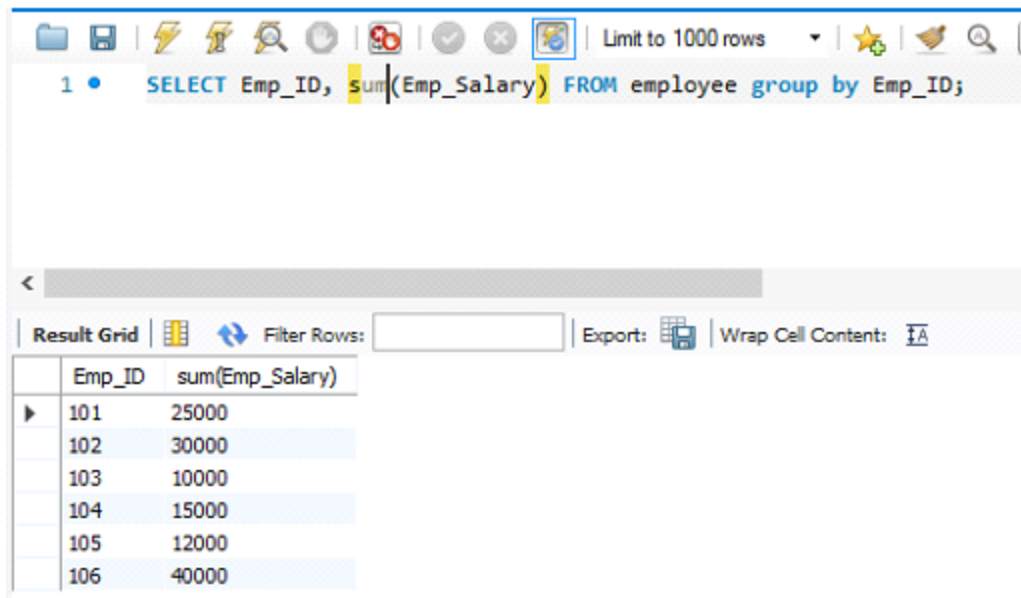
COUNT FUNCTION

COUNT function is used to Count the number of rows in a database table. It can work on both numeric and non-numeric data types.



SUM Function

Sum function is used to calculate the sum of all selected columns. It works on numeric fields only.

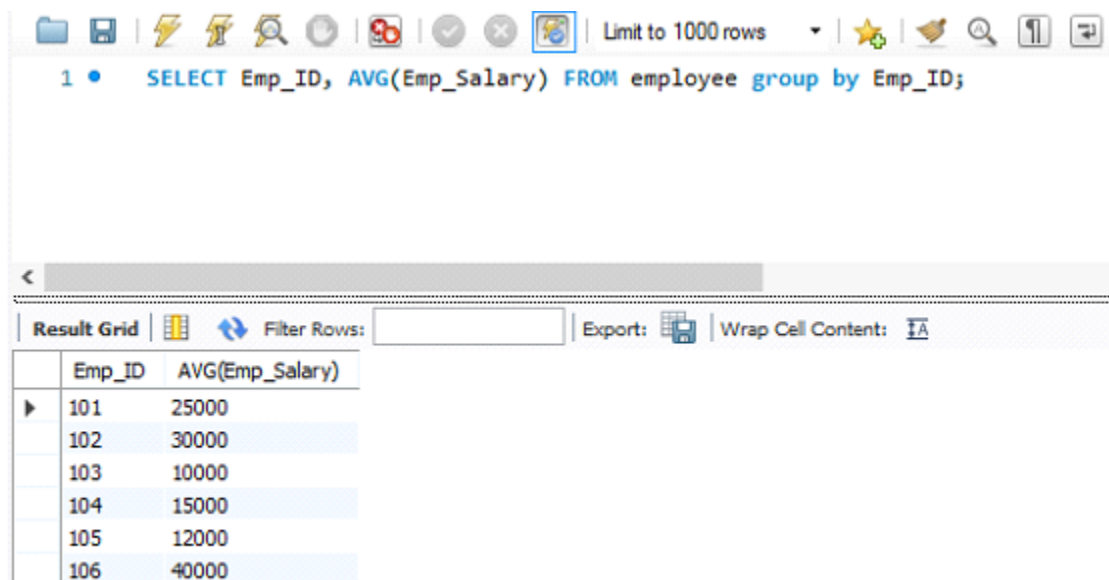


The screenshot shows a SQL IDE interface. At the top, there is a toolbar with various icons and a dropdown menu set to "Limit to 1000 rows". Below the toolbar, a SQL query is entered in a text area: `1 • SELECT Emp_ID, sum(Emp_Salary) FROM employee group by Emp_ID;`. The word "sum" is highlighted in yellow. Below the query editor, there is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Export:" button, and a "Wrap Cell Content:" checkbox. The result grid itself is a table with two columns: "Emp_ID" and "sum(Emp_Salary)". It contains six rows of data, with the first row highlighted in blue.

Emp_ID	sum(Emp_Salary)
101	25000
102	30000
103	10000
104	15000
105	12000
106	40000

AVG function

The AVG function is used to calculate the average value of the numeric type. AVG function returns the average of all non-Null values.

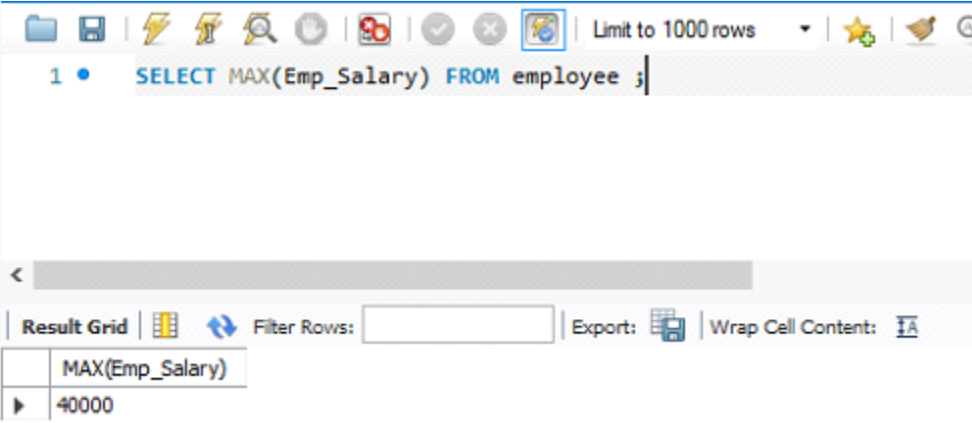


The screenshot shows a SQL IDE interface, similar to the one above. The SQL query in the text area is: `1 • SELECT Emp_ID, AVG(Emp_Salary) FROM employee group by Emp_ID;`. The word "AVG" is highlighted in blue. Below the query editor, the "Result Grid" section is visible, showing a table with two columns: "Emp_ID" and "AVG(Emp_Salary)". The table contains the same six rows of data as the first screenshot, with the first row highlighted in blue.

Emp_ID	AVG(Emp_Salary)
101	25000
102	30000
103	10000
104	15000
105	12000
106	40000

MAX Function

MAX function is used to find the maximum value of a certain column. This function determines the largest value of all selected values of a column.

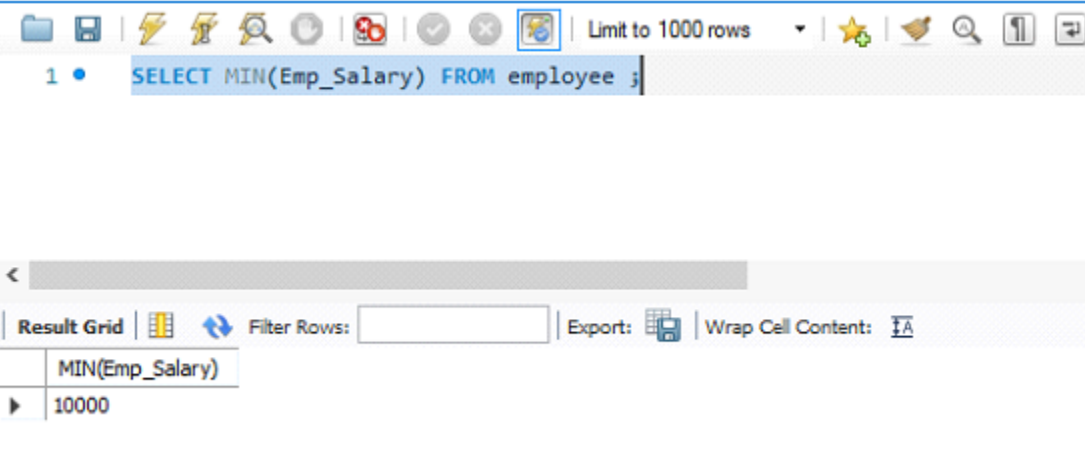


The screenshot shows a SQL query editor with the following SQL query: `SELECT MAX(Emp_Salary) FROM employee ;`. The query is executed, and the result is displayed in a table with one row and one column. The column is labeled `MAX(Emp_Salary)` and the value is `40000`.

MAX(Emp_Salary)
40000

MIN Function

MIN function is used to find the minimum value of a certain column. This function determines the smallest value of all selected values of a column.



The screenshot shows a SQL query editor with the following SQL query: `SELECT MIN(Emp_Salary) FROM employee ;`. The query is executed, and the result is displayed in a table with one row and one column. The column is labeled `MIN(Emp_Salary)` and the value is `10000`.

MIN(Emp_Salary)
10000

MySQL JOINS

MySQL JOINS are used with SELECT statements. It is used to retrieve data from multiple tables. It is performed whenever you need to fetch records from two or more tables.

There are three types of MySQL joins:

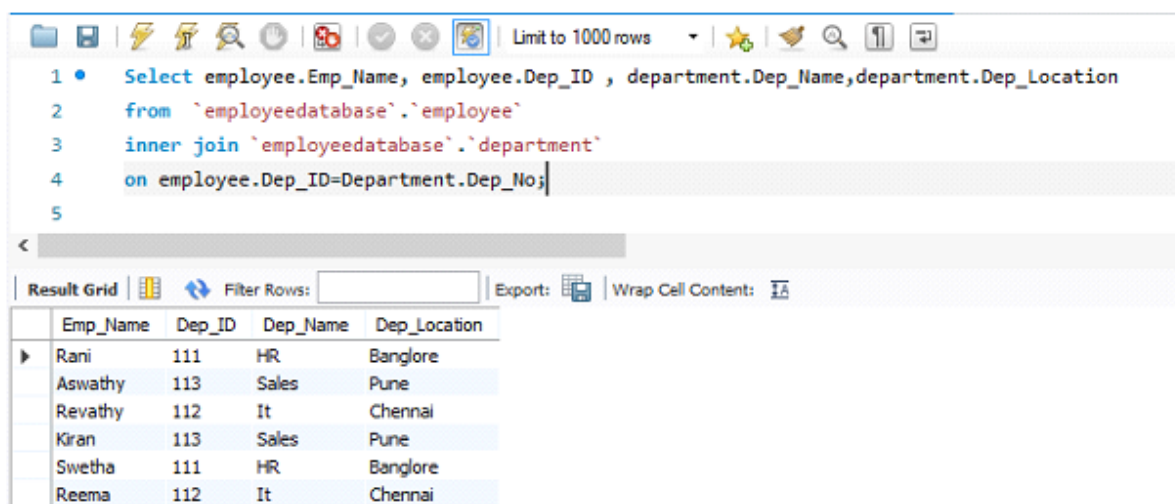
- MySQL INNER JOIN (or sometimes called simple join)
- MySQL LEFT OUTER JOIN (or sometimes called LEFT JOIN)
- MySQL RIGHT OUTER JOIN (or sometimes called RIGHT JOIN)

MySQL Inner Join

The MySQL Inner Join is used to return only those results from the tables that **match** the specified condition and hides other rows and columns. MySQL assumes it as a default Join, so it is optional to use the Inner Join keyword with the query.

SELECT columns FROM table1 INNER JOIN table2 ON condition1

INNER JOIN table3 ON condition2...;



The screenshot shows a MySQL query editor with a toolbar at the top. The query is as follows:

```
1 • Select employee.Emp_Name, employee.Dep_ID , department.Dep_Name,department.Dep_Location
2 from `employee`
3 inner join `department`
4 on employee.Dep_ID=Department.Dep_No;
5
```

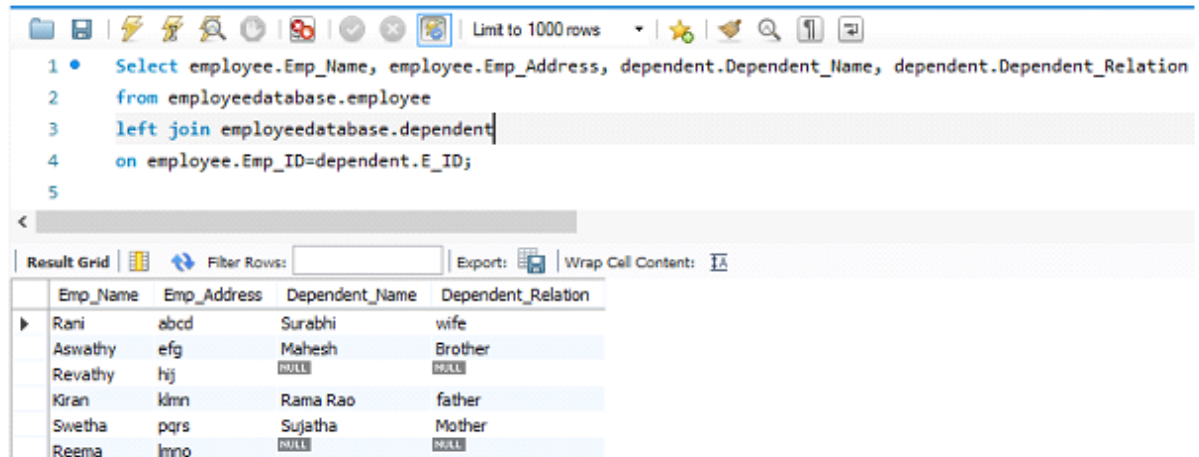
Below the query editor, the 'Result Grid' tab is active, displaying the following data:

	Emp_Name	Dep_ID	Dep_Name	Dep_Location
▶	Rani	111	HR	Banglore
	Aswathy	113	Sales	Pune
	Revathy	112	It	Chennai
	Kiran	113	Sales	Pune
	Swetha	111	HR	Banglore
	Reema	112	It	Chennai

MySQL LEFT JOIN

Left Join clause returns all the rows from the left table and matched records from the right table or returns Null if no matching record is found. This Join can also be called a **Left Outer Join** clause.

```
SELECT columns FROM table1 LEFT [OUTER] JOIN table2 ON Join_Condition;
```



The screenshot shows a SQL query editor with the following query:

```
1 • Select employee.Emp_Name, employee.Emp_Address, dependent.Dependent_Name, dependent.Dependent_Relation
2 from employeedatabase.employee
3 left join employeedatabase.dependent
4 on employee.Emp_ID=dependent.E_ID;
5
```

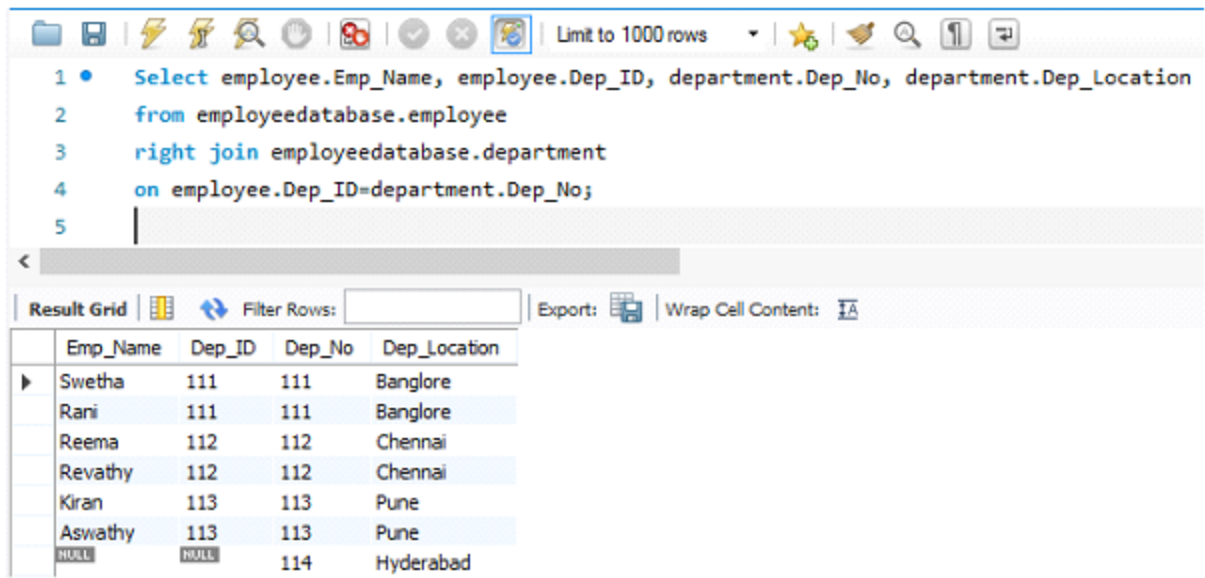
Below the query, the results are displayed in a grid. The grid has columns: Emp_Name, Emp_Address, Dependent_Name, and Dependent_Relation. The results are as follows:

Emp_Name	Emp_Address	Dependent_Name	Dependent_Relation
Rani	abcd	Surabhi	wife
Aswathy	efg	Mahesh	Brother
Revathy	hij	NULL	NULL
Kiran	klmn	Rama Rao	father
Swetha	pqrs	Sujatha	Mother
Reema	lmno	NULL	NULL

MySQL RIGHT JOIN

The Right Join is used to joins two or more tables and returns all rows from the right-hand table, and only those results from the other table that fulfilled the join condition. If it finds unmatched records from the left side table, it returns Null value.

```
SELECT column_list FROM Table1 RIGHT [OUTER] JOIN Table2 ON
join_condition;
```



The screenshot shows a SQL IDE with a query editor and a result grid. The query is a right join between employee and department tables. The result grid shows 7 rows of data.

```

1 • Select employee.Emp_Name, employee.Dep_ID, department.Dep_No, department.Dep_Location
2   from employeedatabase.employee
3   right join employeedatabase.department
4   on employee.Dep_ID=department.Dep_No;
5

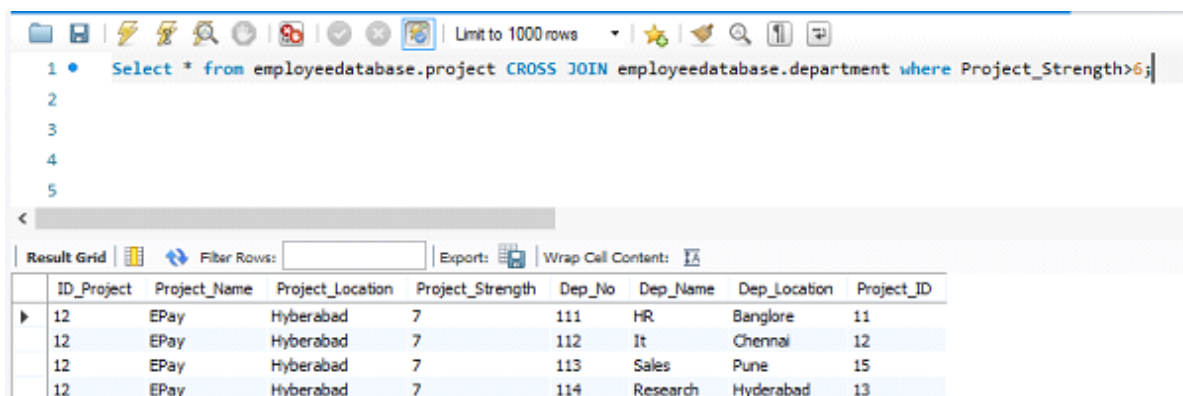
```

	Emp_Name	Dep_ID	Dep_No	Dep_Location
▶	Swetha	111	111	Banglore
	Rani	111	111	Banglore
	Reema	112	112	Chennai
	Revathy	112	112	Chennai
	Kiran	113	113	Pune
	Aswathy	113	113	Pune
	NULL	NULL	114	Hyderabad

MySQL CROSS JOIN

MySQL CROSS JOIN is used to combine all possibilities of the two or more tables and returns the result that contains every row from all contributing tables. The CROSS JOIN is also known as CARTESIAN JOIN, which provides the Cartesian product of all associated tables.

SELECT column-lists FROM table1 CROSS JOIN table2;



The screenshot shows a SQL IDE with a query editor and a result grid. The query is a cross join between project and department tables, filtered by Project_Strength > 6. The result grid shows 4 rows of data.

```

1 • Select * from employeedatabase.project CROSS JOIN employeedatabase.department where Project_Strength>6;
2
3
4
5

```

	ID_Project	Project_Name	Project_Location	Project_Strength	Dep_No	Dep_Name	Dep_Location	Project_ID
▶	12	EPay	Hyderabad	7	111	HR	Banglore	11
	12	EPay	Hyderabad	7	112	It	Chennai	12
	12	EPay	Hyderabad	7	113	Sales	Pune	15
	12	EPay	Hyderabad	7	114	Research	Hyderabad	13