

# SQL

## \*Query to Create Database.

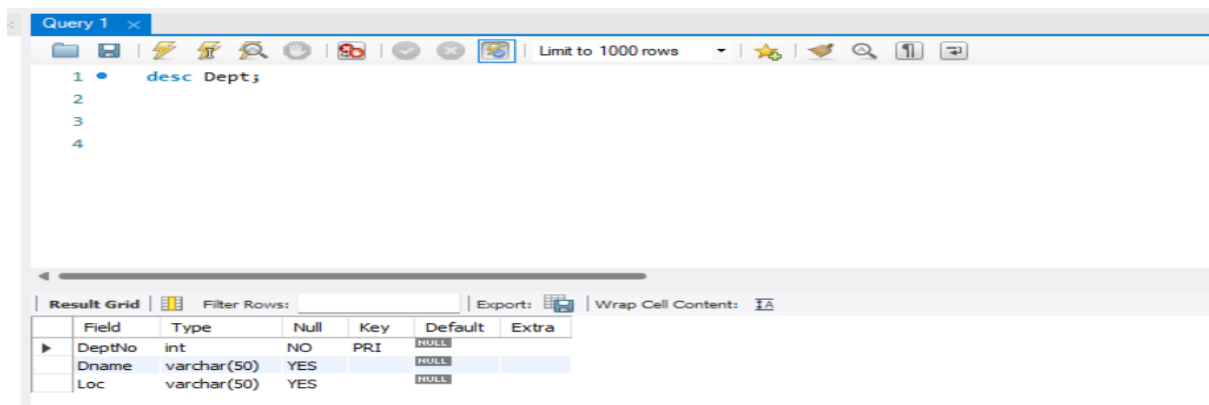
Create database test1;

Show databases;

use test1;

## \*Query to Create Dept table with DeptNo as the primary key.

Create table Dept (DeptNo int PRIMARY KEY,Dname varchar(50),Loc varchar(50));



The screenshot shows a SQL query editor window titled 'Query 1'. The query text is: `desc Dept;`. Below the query, the 'Result Grid' displays the structure of the 'Dept' table. The table has three columns: 'DeptNo' (int, NO, PRI, NULL), 'Dname' (varchar(50), YES, NULL), and 'Loc' (varchar(50), YES, NULL).

Field	Type	Null	Key	Default	Extra
DeptNo	int	NO	PRI	NULL	
Dname	varchar(50)	YES		NULL	
Loc	varchar(50)	YES		NULL	

\*Here are the SQL queries to insert the provided data into these tables:

Inserting data into Dept table...

Insert into Dept (DeptNo, Dname, Loc) values

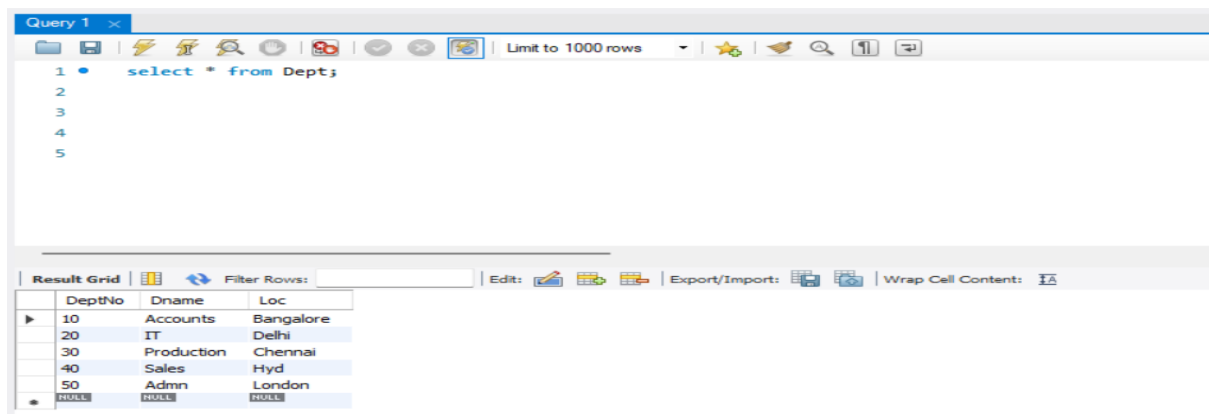
(10, 'Accounts', 'Bangalore'),

(20, 'IT', 'Delhi'),

(30, 'Production', 'Chennai'),

(40, 'Sales', 'Hyd'),

(50, 'Admn', 'London');



The screenshot shows a SQL query editor window titled 'Query 1'. The query text is: `select * from Dept;`. Below the query, the 'Result Grid' displays the data inserted into the 'Dept' table. The table has three columns: 'DeptNo', 'Dname', and 'Loc'. The data rows are: (10, Accounts, Bangalore), (20, IT, Delhi), (30, Production, Chennai), (40, Sales, Hyd), and (50, Admn, London). There is also a row for NULL values.

DeptNo	Dname	Loc
10	Accounts	Bangalore
20	IT	Delhi
30	Production	Chennai
40	Sales	Hyd
50	Admn	London
NULL	NULL	NULL

**\*Create Emp table with EmpNo as the primary key and DeptNo as a foreign key referencing Dept(DeptNo):**

```
Create table Emp (EmpNo int PRIMARY KEY, Ename varchar(50), Sal int, Hire_Date
date, Commission int, DeptNo int, Mgr int, FOREIGN KEY (DeptNo) REFERENCES
Dept(DeptNo));
```

The screenshot shows the Oracle SQL Developer interface. The top toolbar includes icons for file operations, execution, and search. The 'Query 1' window displays the SQL statement `desc Emp;`. The 'Result Grid' window shows the table structure for the EMP table, including fields like EmpNo, Ename, Sal, Hire\_Date, Commission, DeptNo, and Mgr, along with their data types, nullability, and primary/foreign key status.

	Field	Type	Null	Key	Default	Extra
▶	EmpNo	int	NO	PRI	NULL	
	Ename	varchar(50)	YES		NULL	
	Sal	int	YES		NULL	
	Hire_Date	date	YES		NULL	
	Commission	int	YES		NULL	
	DeptNo	int	YES	MUL	NULL	
	Mgr	int	YES		NULL	

\*Inserting data into Emp table

Insert into Emp (EmpNo, Ename, Sal, Hire Date, Commission, DeptNo, Mgr) values

(1001, 'Sachin', 19000, '1980-01-01', 2100, 20, 1003),

(1002, 'Kapil', 15000, '1970-01-01', 2300, 10, 1003),

(1003, 'Stefen', 12000, '1990-01-01', 500, 20, 1007),

(1004, 'Williams', 9000, '2001-01-01', NULL, 30, 1007),

(1005, 'John', 5000, '2005-01-01', NULL, 30, 1006),

(1006, 'Dravid', 19000, '1985-01-01', 2400, 10, 1007),

```
(1007, 'Martin', 21000, '2000-01-01', 1040, NULL, NULL);
```

[illegible]

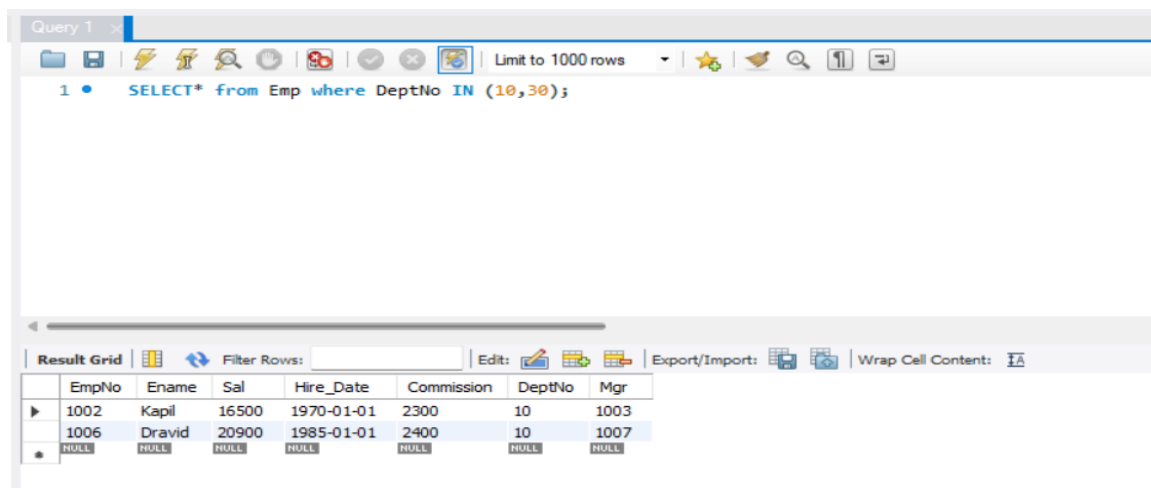
# Queries:

1. Select employee details of dept number 10 or 30.

*Query:*

```
SELECT* from Emp where DeptNo IN (10,30);
```

*Output:*



The screenshot shows a database query editor window titled 'Query 1'. The query entered is 'SELECT\* from Emp where DeptNo IN (10,30);'. Below the query editor, the 'Result Grid' is displayed, showing the following data:

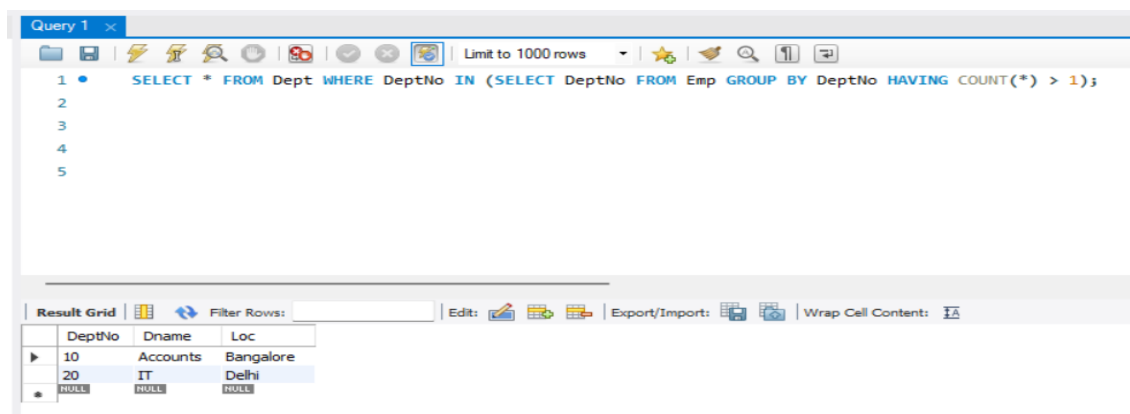
EmpNo	Ename	Sal	Hire_Date	Commission	DeptNo	Mgr
1002	Kapil	16500	1970-01-01	2300	10	1003
1006	Dravid	20900	1985-01-01	2400	10	1007
HULL	HULL	HULL	HULL	HULL	HULL	HULL

2. Write a query to fetch all the dept details with more than 1 Employee.

*Query:*

```
SELECT * FROM Dept WHERE DeptNo IN (SELECT DeptNo FROM Emp GROUP BY DeptNo HAVING COUNT(*) > 1);
```

*Output:*



The screenshot shows a database query editor window titled 'Query 1'. The query entered is 'SELECT \* FROM Dept WHERE DeptNo IN (SELECT DeptNo FROM Emp GROUP BY DeptNo HAVING COUNT(\*) > 1);'. Below the query editor, the 'Result Grid' is displayed, showing the following data:

DeptNo	Dname	Loc
10	Accounts	Bangalore
20	IT	Delhi
HULL	HULL	HULL

3. Write a query to fetch employee details whose name starts with the letter “S”.

*Query:*

```
SELECT * FROM Emp WHERE Ename LIKE 'S%';
```

*Output:*

Query 1

1 SELECT \* FROM Emp WHERE Ename LIKE '%S%';

EmpNo	Ename	Sal	Hire_Date	Commission	DeptNo	Mgr
1001	Sachin	20900	1980-01-01	2100	20	1003
1003	Stefen	13200	1990-01-01	500	20	1007

4. Select Emp Details Whose experience is more than 2 years.

Query:

SELECT \* FROM Emp WHERE DATEDIFF(CURDATE(), Hire\_Date) > 730;

Output:

Query 1

1 SELECT \* FROM Emp WHERE DATEDIFF(CURDATE(), Hire\_Date) > 730;

2

3

4

5

EmpNo	Ename	Sal	Hire_Date	Commission	DeptNo	Mgr
1001	Sachin	20900	1980-01-01	2100	20	1003
1002	Kapil	16500	1970-01-01	2300	10	1003
1003	Stefen	13200	1990-01-01	500	20	1007
1006	Dravid	20900	1985-01-01	2400	10	1007
1007	Martin	23100	2000-01-01	1040	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

5. Write a SELECT statement to replace the char “a” with “#” in Employee Name.

Query:

SELECT REPLACE(Ename, 'a', '#') AS Modified\_Ename FROM Emp;

Output:

Query 1

1 SELECT REPLACE(Ename, 'a', '#') AS Modified\_Ename FROM Emp;

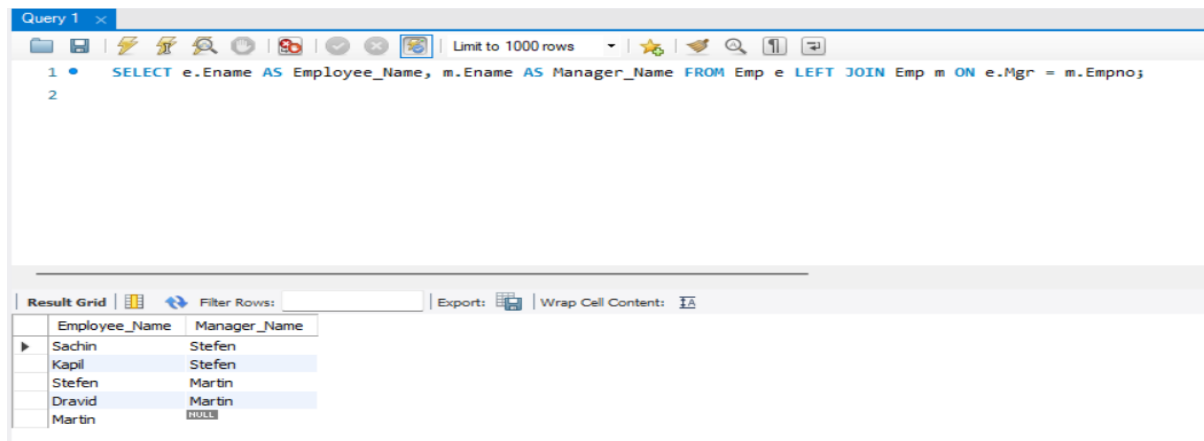
Modified_Ename
S#chin
K#pil
Stefen
Dr#vid
M#rtin

6. Write a query to fetch employee name and his/her manager name.

*Query:*

```
SELECT e.ename AS Employee_Name, m.ename AS Manager_Name FROM Emp e LEFT JOIN Emp m ON e.Mgr = m.EMPNO;
```

*Output:*



The screenshot shows a SQL query editor window titled 'Query 1'. The query is: `SELECT e.ename AS Employee_Name, m.ename AS Manager_Name FROM Emp e LEFT JOIN Emp m ON e.Mgr = m.EMPNO;`. Below the query, the 'Result Grid' displays the output. The grid has two columns: 'Employee\_Name' and 'Manager\_Name'. The data rows are: Sachin (Manager: Stefen), Kapil (Manager: Stefen), Stefen (Manager: Martin), Dravid (Manager: Martin), and Martin (Manager: NULL).

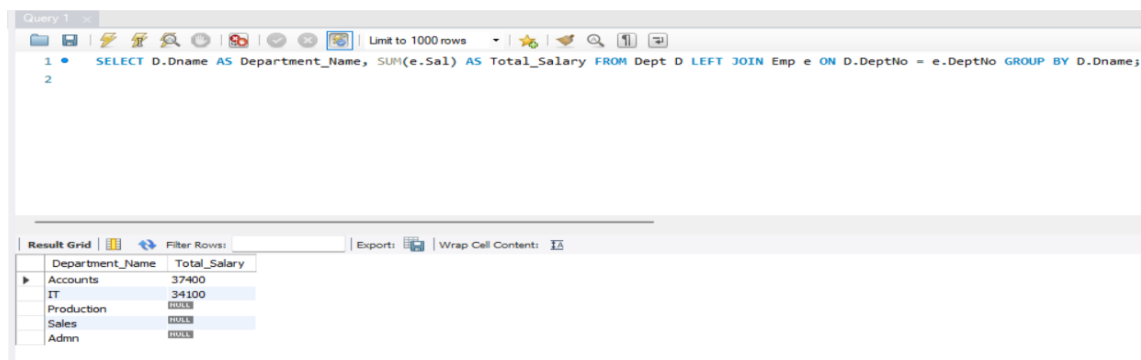
Employee_Name	Manager_Name
Sachin	Stefen
Kapil	Stefen
Stefen	Martin
Dravid	Martin
Martin	NULL

7. Fetch Dept Name , Total Salary of the Dept.

*Query:*

```
SELECT D.Dname AS Department_Name, SUM(e.Sal) AS Total_Salary FROM Dept D LEFT JOIN Emp e ON D.DeptNo = e.DeptNo GROUP BY D.Dname;
```

*Output:*



The screenshot shows a SQL query editor window titled 'Query 1'. The query is: `SELECT D.Dname AS Department_Name, SUM(e.Sal) AS Total_Salary FROM Dept D LEFT JOIN Emp e ON D.DeptNo = e.DeptNo GROUP BY D.Dname;`. Below the query, the 'Result Grid' displays the output. The grid has two columns: 'Department\_Name' and 'Total\_Salary'. The data rows are: Accounts (Total Salary: 37400), IT (Total Salary: 34100), Production (Total Salary: 0), Sales (Total Salary: 0), and Admin (Total Salary: 0).

Department_Name	Total_Salary
Accounts	37400
IT	34100
Production	0
Sales	0
Admin	0

8. Write a query to fetch **ALL** the employee details along with department name, department location, irrespective of employee existence in the department.

*Query:*

```
SELECT e.*, D.Dname AS Department_Name, D.Loc AS Department_Location FROM Dept D LEFT JOIN Emp e ON D.DeptNo = e.DeptNo;
```

*Output:*

Query 1 x

Limit to 1000 rows

```

1 • SELECT e.*, D.Dname AS Department_Name, D.Loc AS Department_Location
2 FROM Dept D
3 LEFT JOIN Emp e ON D.DeptNo = e.DeptNo;
4

```

Result Grid

EmpNo	Ename	Sal	Hire_Date	Commission	DeptNo	Mgr	Department_Name	Department_Location
1006	David	20900	1985-01-01	2400	10	1007	Accounts	Bangalore
1002	Kapil	16500	1970-01-01	2300	10	1003	Accounts	Bangalore
1003	Stefen	13200	1990-01-01	500	20	1007	IT	Delhi
1001	Sachin	20900	1980-01-01	2100	20	1003	IT	Delhi
NULL	NULL	NULL	NULL	NULL	NULL	NULL	Production	Chennai
NULL	NULL	NULL	NULL	NULL	NULL	NULL	Sales	Hyd
NULL	NULL	NULL	NULL	NULL	NULL	NULL	Admin	London

9. Write an update statement to increase the employee salary by 10 %.

*Query:*

UPDATE Emp SET Sal = Sal \* 1.1;

*Output:*

Query 1 x

Limit to 1000 rows

```

1 • UPDATE Emp SET Sal = Sal * 1.1;

```

10. Write a statement to delete employees belong to Chennai location.

*Query:*

DELETE FROM Emp WHERE DeptNo IN (SELECT DeptNo FROM Dept WHERE Loc = 'Chennai');

*Output:*

Query 1 x

Limit to 1000 rows

```

1 DELETE FROM Emp WHERE DeptNo IN (SELECT DeptNo FROM Dept WHERE Loc = 'Chennai');

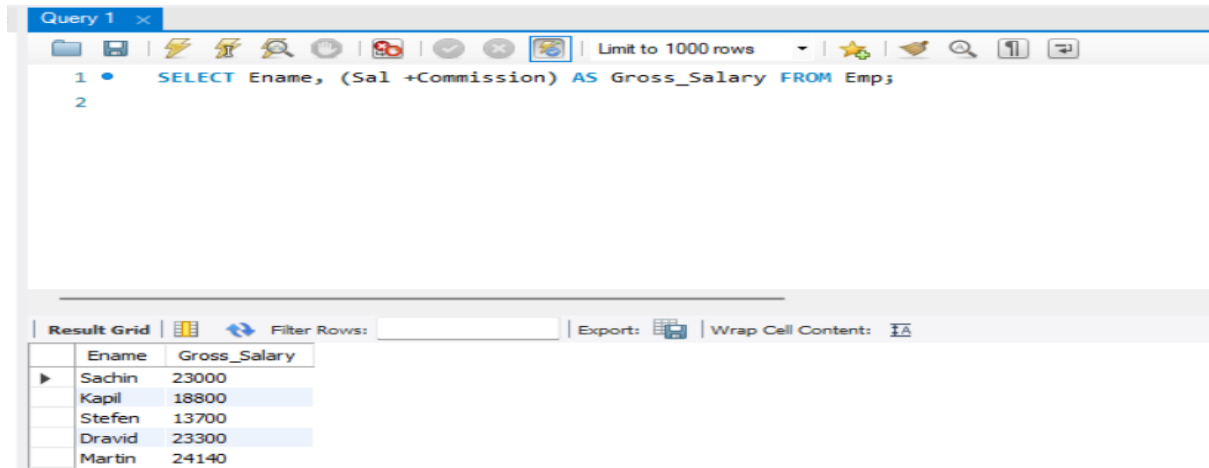
```

11. Get Employee Name and gross salary (Sal + commission) .

Query:

```
SELECT Ename, (Sal +Commission) AS Gross_Salary FROM Emp;
```

Output:



The screenshot shows a SQL Developer window titled 'Query 1'. The query editor contains the following SQL statement:

```
1 • SELECT Ename, (Sal +Commission) AS Gross_Salary FROM Emp;
2
```

Below the query editor, the 'Result Grid' tab is active, displaying the results of the query. The results are as follows:

Ename	Gross_Salary
Sachin	23000
Kapil	18800
Stefen	13700
Dravid	23300
Martin	24140

12. Increase the data length of the column Ename of Emp table from 100 to 250 using ALTER statement.

Query:

```
ALTER TABLE Emp MODIFY COLUMN Ename varchar(250);
```

Output:



The screenshot shows a SQL Developer window titled 'Query 1'. The query editor contains the following SQL statement:

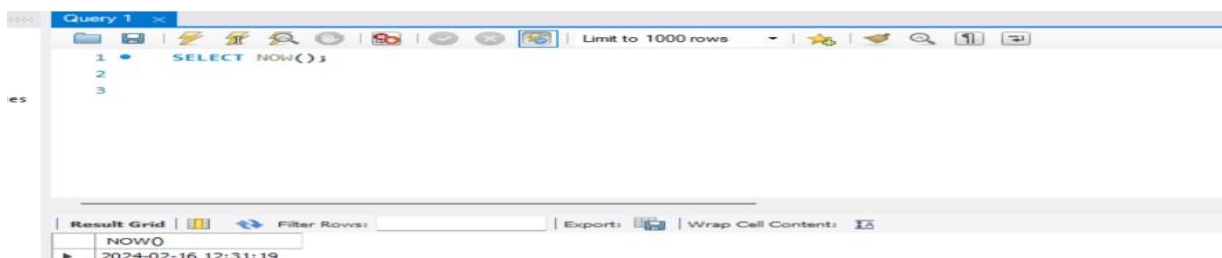
```
1 • ALTER TABLE Emp MODIFY COLUMN Ename varchar(250);
2
```

13. Write query to get current datetime.

Query:

```
SELECT NOW();
```

Output:



The screenshot shows a SQL Developer window titled 'Query 1'. The query editor contains the following SQL statement:

```
1 • SELECT NOW();
2
3
```

Below the query editor, the 'Result Grid' tab is active, displaying the results of the query. The results are as follows:

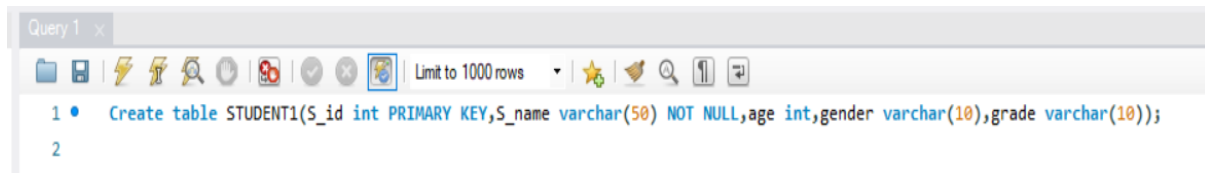
NOW()
2024-02-16 12:31:19

14. Write a statement to create STUDENT table, with related 5 columns.

*Query:*

Create table STUDENT1(S\_id int PRIMARY KEY,S\_name varchar(50) NOT NULL,age int ,gender varchar(10),grade varchar(10));

*Output:*

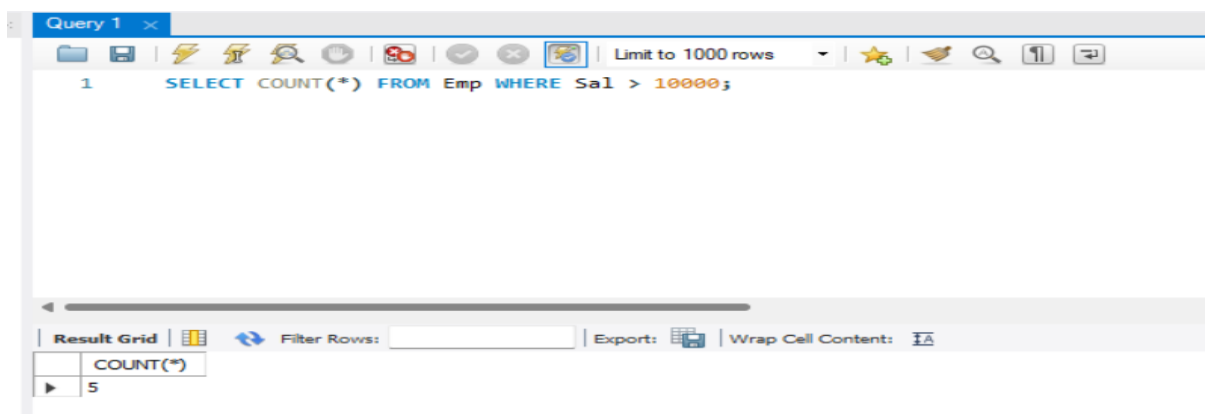


15. Write a query to fetch number of employees in who is getting salary more than 10000.

*Query:*

SELECT COUNT(\*) FROM Emp WHERE Sal>10000;

*Output:*

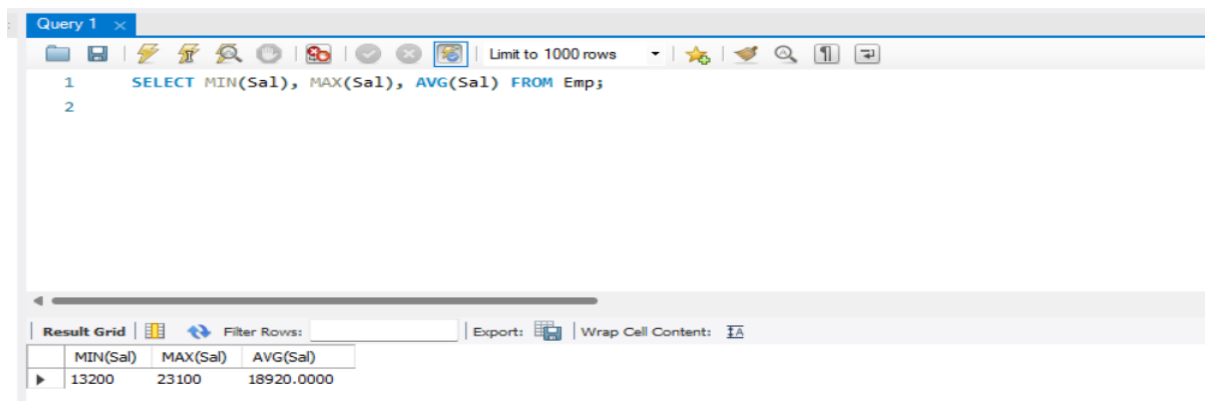


16. Write a query to fetch minimum salary, maximum salary and average salary from emp table.

*Query:*

SELECT MIN(Sal), MAX(Sal), AVG(Sal) FROM Emp;

*Output:*



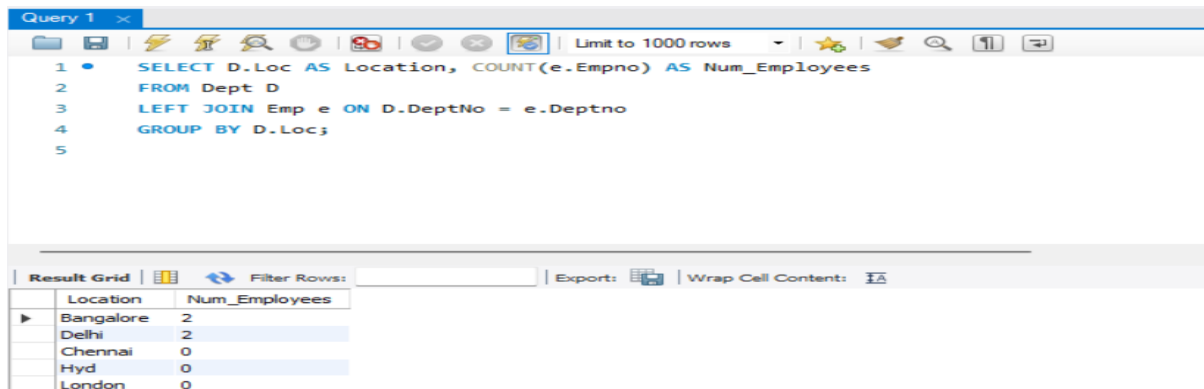


17. Write a query to fetch number of employees in each location

*Query:*

```
SELECT D.Loc AS Location, COUNT(e.Empno) AS Num_Employees
FROM Dept D
LEFT JOIN Emp e ON D.DeptNo = e.Deptno
GROUP BY D.Loc;
```

*Output:*



The screenshot shows a SQL Developer window titled 'Query 1'. The query editor contains the following SQL statement:

```
1 • SELECT D.Loc AS Location, COUNT(e.Empno) AS Num_Employees
2 FROM Dept D
3 LEFT JOIN Emp e ON D.DeptNo = e.Deptno
4 GROUP BY D.Loc;
5
```

Below the query editor is the 'Result Grid' tab, which displays the following data:

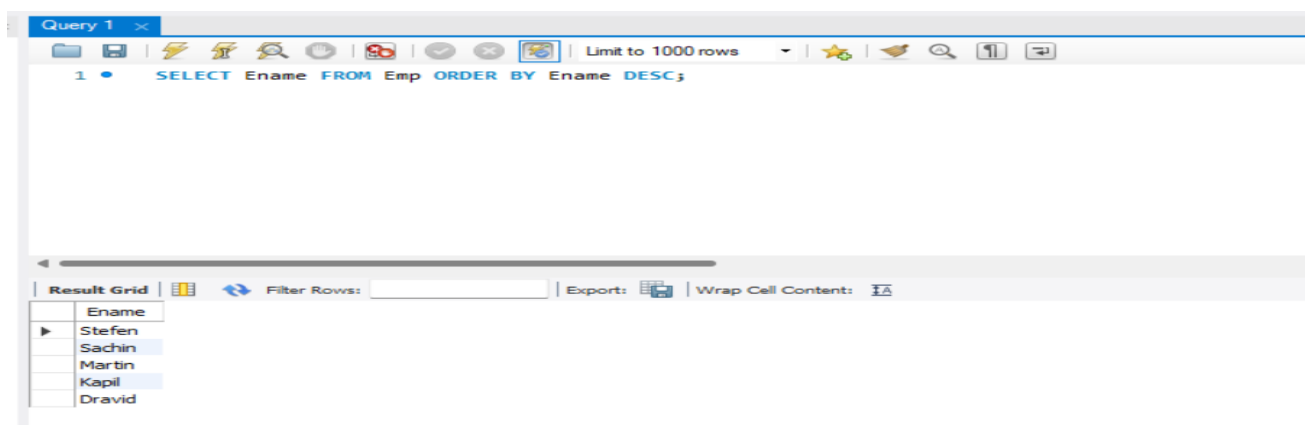
Location	Num_Employees
Bangalore	2
Delhi	2
Chennai	0
Hyd	0
London	0

18. Write a query to display employee names in descending order.

*Query:*

```
SELECT Ename FROM Emp ORDER BY Ename DESC;
```

*Output:*



The screenshot shows a SQL Developer window titled 'Query 1'. The query editor contains the following SQL statement:

```
1 • SELECT Ename FROM Emp ORDER BY Ename DESC;
```

Below the query editor is the 'Result Grid' tab, which displays the following data:

Ename
Stefen
Sachin
Martin
Kapil
David

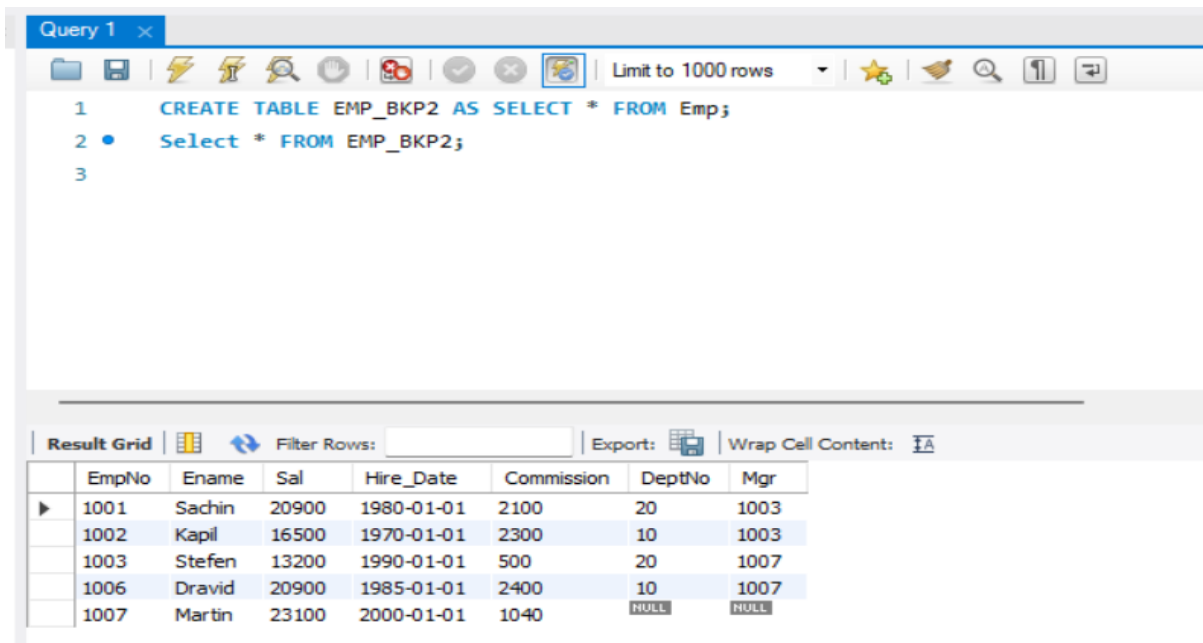
19. Write a statement to create a new table(**EMP\_BKP**) from the existing **EMP** table .

*Query:*

```
CREATE TABLE EMP_BKP2 AS SELECT * FROM Emp;
```

```
Select * FROM EMP_BKP2;
```

*Output:*



Query 1

```
1 CREATE TABLE EMP_BKP2 AS SELECT * FROM Emp;
2 Select * FROM EMP_BKP2;
3
```

Limit to 1000 rows

Result Grid

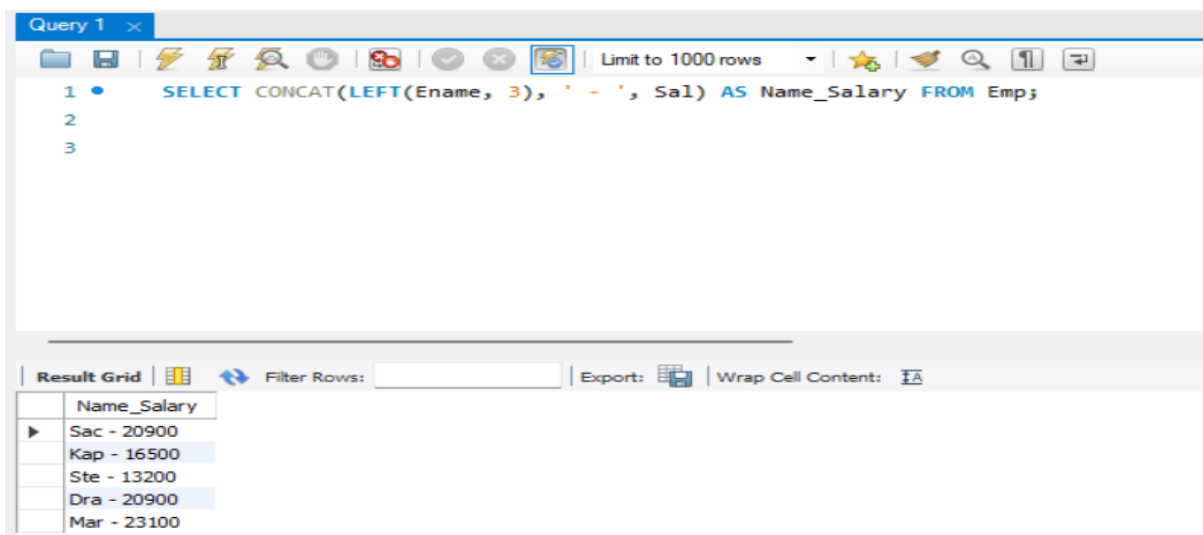
EmpNo	Ename	Sal	Hire_Date	Commission	DeptNo	Mgr
1001	Sachin	20900	1980-01-01	2100	20	1003
1002	Kapil	16500	1970-01-01	2300	10	1003
1003	Stefen	13200	1990-01-01	500	20	1007
1006	Dravid	20900	1985-01-01	2400	10	1007
1007	Martin	23100	2000-01-01	1040	NULL	NULL

20. Write a query to fetch first 3 characters from employee name appended with salary.

*Query:*

```
SELECT CONCAT(LEFT(Ename, 3), ' - ', Sal) AS Name_Salary FROM Emp;
```

*Output:*



Query 1

```
1 SELECT CONCAT(LEFT(Ename, 3), ' - ', Sal) AS Name_Salary FROM Emp;
2
3
```

Limit to 1000 rows

Result Grid

Name_Salary
Sac - 20900
Kap - 16500
Ste - 13200
Dra - 20900
Mar - 23100

