

## Module : 7

### Topic : RDBMS & Database Programming with JDBC

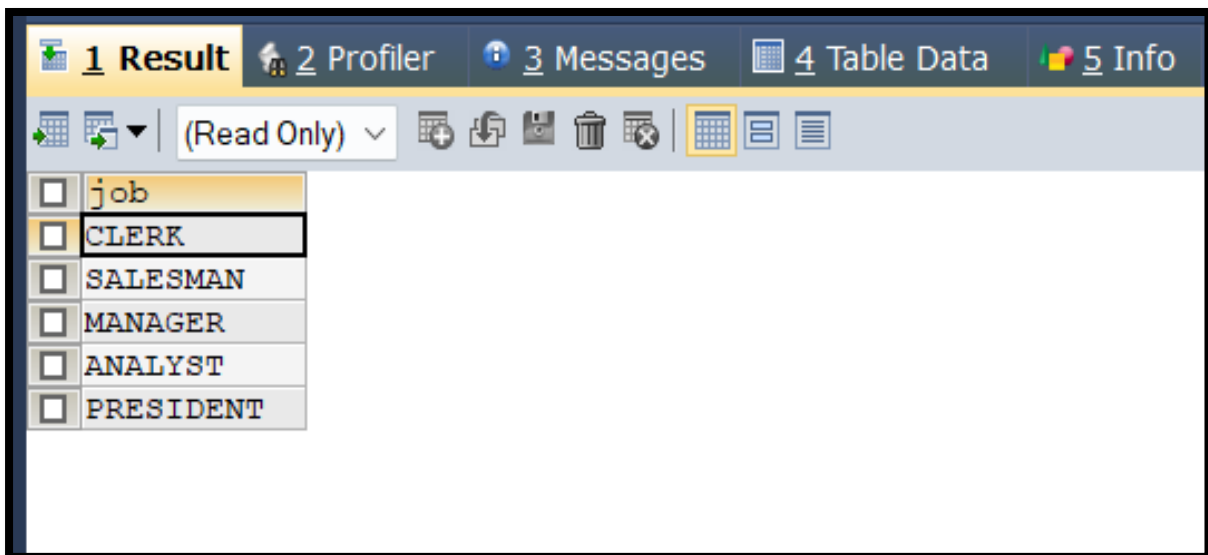
Name : Abhay Gajjar

---

### Queries :

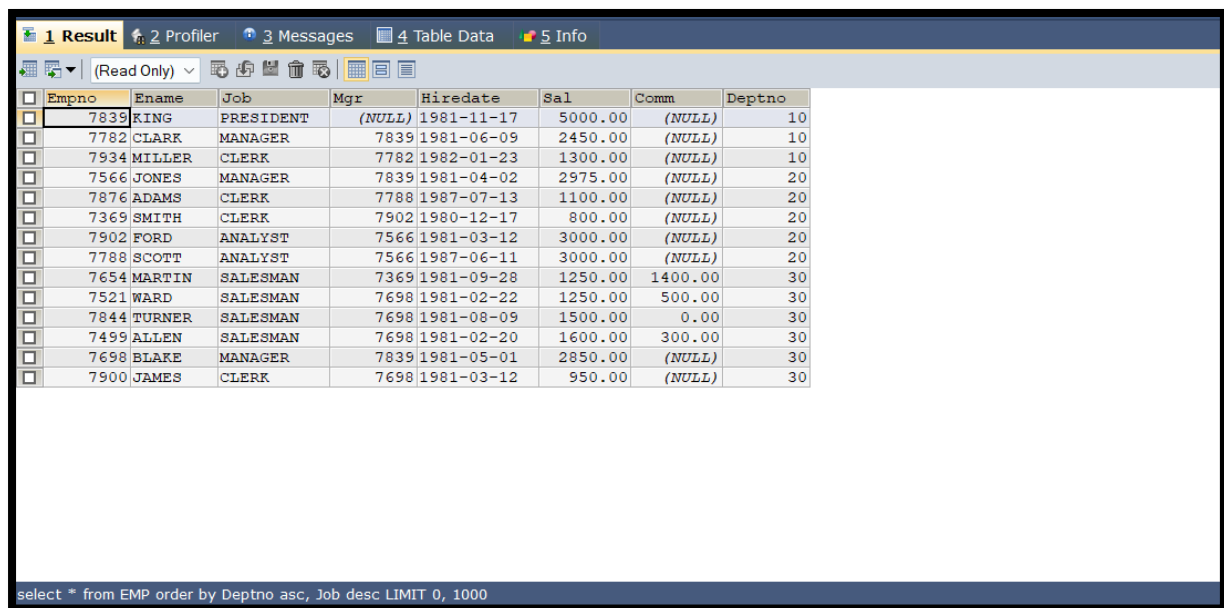
1) Select unique job from EMP table ?

=> SELECT DISTINCT job FROM emp;



**2) List the details of the emps in asc order of the Dptnos and desc of Jobs ?**

=> `SELECT * FROM EMP ORDER BY Deptno ASC, Job DESC;`



The screenshot shows a database query result in a 'Table Data' tab. The table contains 20 rows of employee data, sorted by department number (Deptno) in ascending order and then by job title in descending order. The columns are Empno, Ename, Job, Mgr, Hiredate, Sal, Comm, and Deptno. The data is as follows:

Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
7839	KING	PRESIDENT	(NULL)	1981-11-17	5000.00	(NULL)	10
7782	CLARK	MANAGER	7839	1981-06-09	2450.00	(NULL)	10
7934	MILLER	CLERK	7782	1982-01-23	1300.00	(NULL)	10
7566	JONES	MANAGER	7839	1981-04-02	2975.00	(NULL)	20
7876	ADAMS	CLERK	7788	1987-07-13	1100.00	(NULL)	20
7369	SMITH	CLERK	7902	1980-12-17	800.00	(NULL)	20
7902	FORD	ANALYST	7566	1981-03-12	3000.00	(NULL)	20
7788	SCOTT	ANALYST	7566	1987-06-11	3000.00	(NULL)	20
7654	MARTIN	SALESMAN	7369	1981-09-28	1250.00	1400.00	30
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30
7844	TURNER	SALESMAN	7698	1981-08-09	1500.00	0.00	30
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00	(NULL)	30
7900	JAMES	CLERK	7698	1981-03-12	950.00	(NULL)	30

select \* from EMP order by Deptno asc, Job desc LIMIT 0, 1000

**3) Display all the unique job groups in the descending order?**

=> `SELECT DISTINCT(Job) FROM EMP ORDER BY Job DESC;`

1 Result 2 Profiler 3 Messages 4 Table Data 5 Info

(Read Only)

Job
SALESMAN
PRESIDENT
MANAGER
CLERK
ANALYST

select DISTINCT(Job) from EMP order by Job Desc LIMIT 0, 1000

#### 4) List the emps who joined before 1981.

=> SELECT \* FROM EMP WHERE Hiredate < '1981-01-01';

1 Result 2 Profiler 3 Messages 4 Table Data 5 Info

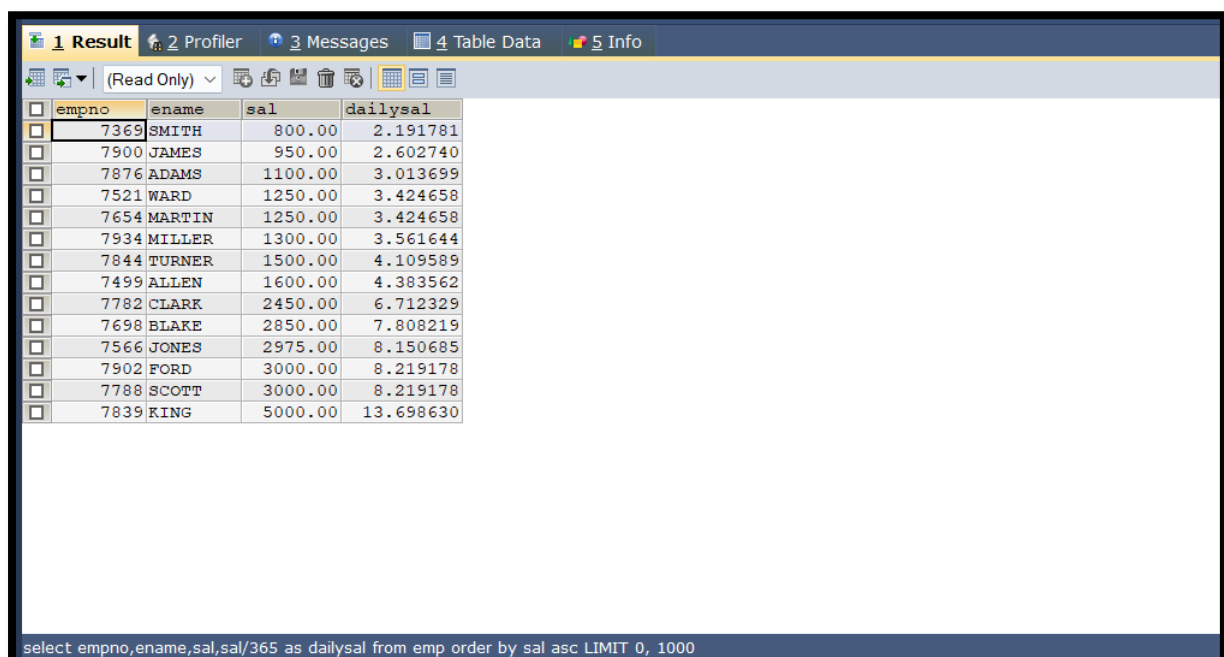
(Read Only)

Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
7369	SMITH	CLERK	7902	1980-12-17	800.00	(NULL)	20

select \* from EMP where Hiredate < '1981-01-01' LIMIT 0, 1000

**5) List the Empno, Ename, Sal, Daily sal of all emps in the asc order of Annsal.**

=> `SELECT empno,ename,sal,sal/365 AS dailysal  
FROM emp ORDER BY sal ASC;`



The screenshot shows a database query result window with a table of employee data. The table has four columns: empno, ename, sal, and dailysal. The data is sorted by salary in ascending order. The SQL query at the bottom of the window is: `select empno,ename,sal,sal/365 as dailysal from emp order by sal asc LIMIT 0, 1000`

empno	ename	sal	dailysal
7369	SMITH	800.00	2.191781
7900	JAMES	950.00	2.602740
7876	ADAMS	1100.00	3.013699
7521	WARD	1250.00	3.424658
7654	MARTIN	1250.00	3.424658
7934	MILLER	1300.00	3.561644
7844	TURNER	1500.00	4.109589
7499	ALLEN	1600.00	4.383562
7782	CLARK	2450.00	6.712329
7698	BLAKE	2850.00	7.808219
7566	JONES	2975.00	8.150685
7902	FORD	3000.00	8.219178
7788	SCOTT	3000.00	8.219178
7839	KING	5000.00	13.698630

**6) List the Empno, Ename, Sal, Exp of all emps working for Mgr 7369.**

=> `SELECT Empno,Ename,Sal,(YEAR(CURDATE())-  
YEAR(Hiredate)) AS EXP FROM EMP WHERE  
Mgr=7369;`

1 Result				
2 Profiler				
3 Messages				
4 Table Data				
5 Info				
(Read Only)				
Empno	Ename	Sal	exp	
7654	MARTIN	1250.00	44	

select Empno,Ename,Sal,(year(curdate())-year(Hiredate)) as exp from EMP where Mgr=7369 LIMIT 0, 1000

**7) Display all the details of the emps who's Comm. Is more than their Sal?**

=> `SELECT * FROM emp WHERE comm > sal;`

1 Result 2 Profiler 3 Messages 4 Table Data 5 Info							
(Read Only)							
Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
7654	MARTIN	SALESMAN	7369	1981-09-28	1250.00	1400.00	30

select \* from emp where comm > sal LIMIT 0, 1000

**8) List the emps who are either 'CLERK' or 'ANALYST' in the Desc order.**

=> `SELECT * FROM emp WHERE job='clerk' OR job='analyst' ORDER BY job DESC;`

1 Result

2 Profiler

3 Messages

4 Table Data

5 Info

(Read Only)

<input type="checkbox"/>	Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
<input type="checkbox"/>	7369	SMITH	CLERK	7902	1980-12-17	800.00	(NULL)	20
<input type="checkbox"/>	7876	ADAMS	CLERK	7788	1987-07-13	1100.00	(NULL)	20
<input type="checkbox"/>	7900	JAMES	CLERK	7698	1981-03-12	950.00	(NULL)	30
<input type="checkbox"/>	7934	MILLER	CLERK	7782	1982-01-23	1300.00	(NULL)	10
<input type="checkbox"/>	7788	SCOTT	ANALYST	7566	1987-06-11	3000.00	(NULL)	20
<input type="checkbox"/>	7902	FORD	ANALYST	7566	1981-03-12	3000.00	(NULL)	20

select \*

from emp

where job='clerk' or job='analyst'

order by job desc

LIMIT 0, 1000

**9) List the emps Who Annual sal ranging from 22000 and 45000.**

=> `SELECT * FROM emp WHERE (sal*12) BETWEEN 22000 AND 45000;`

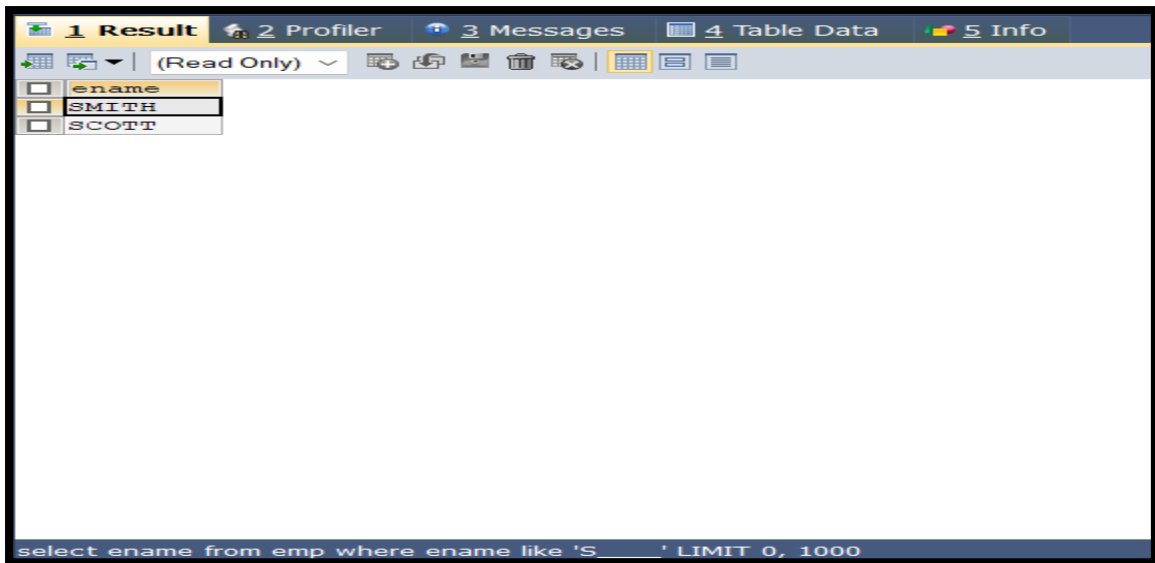
1 Result 2 Profiler 3 Messages 4 Table Data 5 Info							
(Read Only)							
Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
7566	JONES	MANAGER	7839	1981-04-02	2975.00	(NULL)	20
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00	(NULL)	30
7782	CLARK	MANAGER	7839	1981-06-09	2450.00	(NULL)	10
7788	SCOTT	ANALYST	7566	1987-06-11	3000.00	(NULL)	20
7902	FORD	ANALYST	7566	1981-03-12	3000.00	(NULL)	20

select \* from emp where (sal\*12) between 22000 and 45000 LIMIT 0, 1000

10) List the Enames those are starting with 'S' and with five characters.

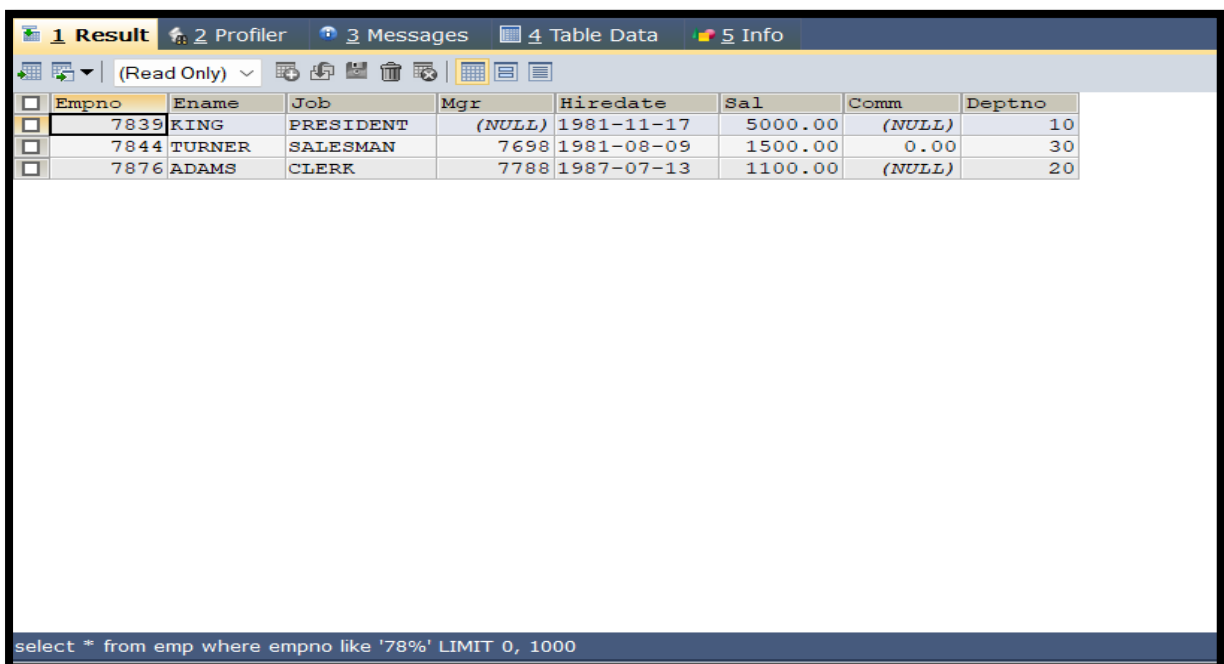
=> SELECT ename FROM emp WHERE ename LIKE 'S\_\_\_\_\_';





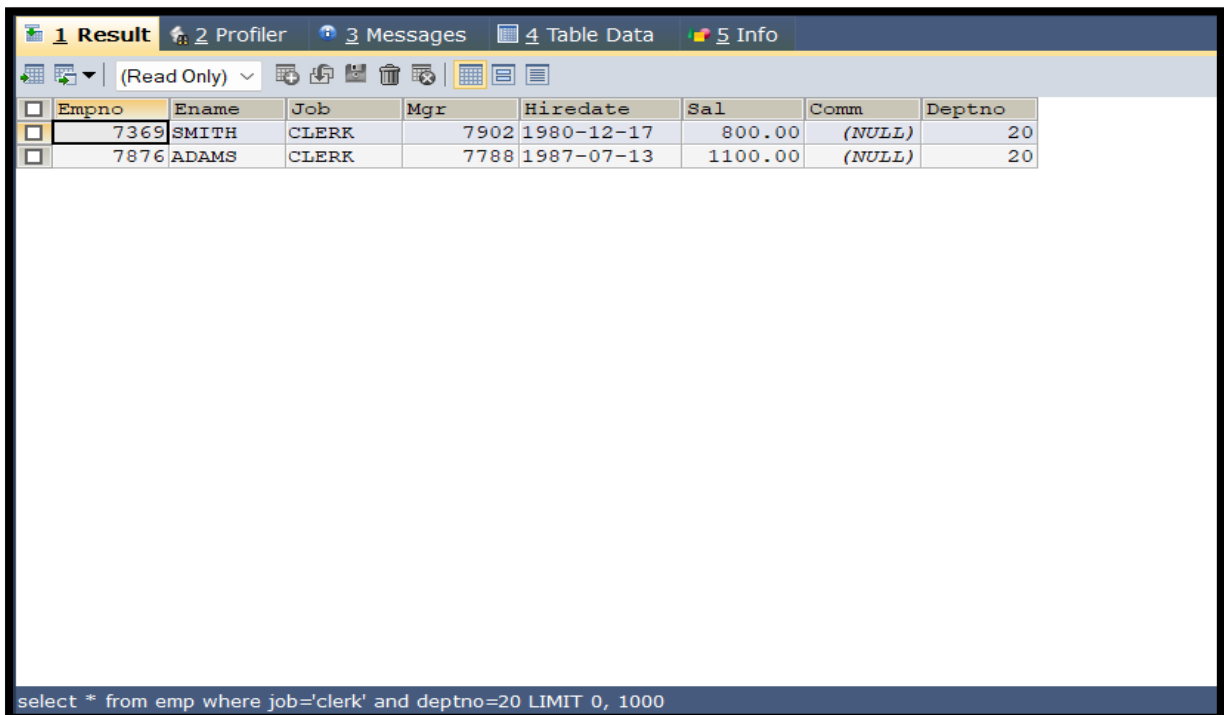
11) List the emps whose Empno not starting with digit 78.

=> `SELECT * FROM emp WHERE empno LIKE '78%';`



## 12) List all the Clerks of Deptno 20.

=> `SELECT * FROM emp WHERE job='clerk' AND deptno=20;`



The screenshot shows a database query result in a web interface. The interface has tabs for '1 Result', '2 Profiler', '3 Messages', '4 Table Data', and '5 Info'. The '1 Result' tab is active, showing a table with 8 columns: Empno, Ename, Job, Mgr, Hiredate, Sal, Comm, and Deptno. There are two rows of data. The first row is for employee 7369, SMITH, a CLERK, reporting to manager 7902, hired on 1980-12-17, with a salary of 800.00 and no commission. The second row is for employee 7876, ADAMS, a CLERK, reporting to manager 7788, hired on 1987-07-13, with a salary of 1100.00 and no commission. Both employees are in department 20. At the bottom of the window, the SQL query is displayed: `select * from emp where job='clerk' and deptno=20 LIMIT 0, 1000`.

Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
7369	SMITH	CLERK	7902	1980-12-17	800.00	(NULL)	20
7876	ADAMS	CLERK	7788	1987-07-13	1100.00	(NULL)	20

## 13) List the Emps who are senior to their own MGRS.

=> `SELECT e.empno, e2.ename FROM emp e , emp e2 WHERE e.mgr = e2.empno AND e.Hiredate < e2.Hiredate;`

1 Result		2 Profiler	3 Messages	4 Table Data	5 Info
(Read Only)					
empno	ename				
7369	FORD				
7499	BLAKE				
7521	BLAKE				
7566	KING				
7698	KING				
7782	KING				
7900	BLAKE				
7902	JONES				

select e.empno, e2.ename from emp e , emp e2 where e.mgr = e2.empno and e.Hiredate < e2.Hiredate LIMIT 0, 1000

**14) List the Emps of Deptno 20 who's Jobs are same as Deptno10.**

=> `SELECT * FROM emp WHERE deptno=20 AND job IN (SELECT job FROM emp WHERE deptno=10);`

1 Result 2 Profiler 3 Messages 4 Table Data 5 Info								
(Read Only)								
Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno	
7369	SMITH	CLERK	7902	1980-12-17	800.00	(NULL)	20	
7566	JONES	MANAGER	7839	1981-04-02	2975.00	(NULL)	20	
7876	ADAMS	CLERK	7788	1987-07-13	1100.00	(NULL)	20	

select \* from emp where deptno=20 and job in (select job from emp where deptno=10) LIMIT 0, 1000

**15) List the Emps who's Sal is same as FORD or SMITH in desc order of Sal.**

=> `SELECT * FROM emp WHERE sal = (SELECT sal FROM emp WHERE ename='ford') OR sal = (SELECT sal FROM emp WHERE ename='smith') ORDER BY sal DESC;`

1 Result 2 Profiler 3 Messages 4 Table Data 5 Info							
(Read Only)							
Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
7788	SCOTT	ANALYST	7566	1987-06-11	3000.00	(NULL)	20
7902	FORD	ANALYST	7566	1981-03-12	3000.00	(NULL)	20
7369	SMITH	CLERK	7902	1980-12-17	800.00	(NULL)	20

select \* from emp where sal = (select sal from emp where ename='ford') or sal = (select sal from emp where ename='smith') order by sal desc LIMIT 0, 1000

16) List the emps whose jobs same as SMITH or ALLEN.

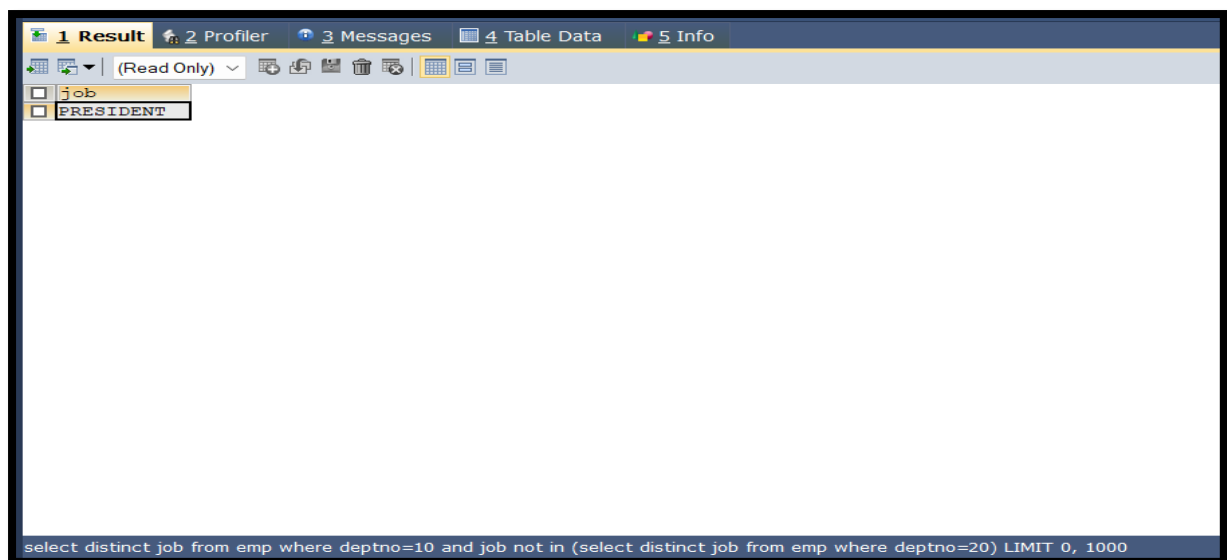
=> SELECT \* FROM emp WHERE job IN (SELECT job FROM emp WHERE ename = 'smith') OR job IN (SELECT job FROM emp WHERE ename = 'allen');

1 Result 2 Profiler 3 Messages 4 Table Data 5 Info							
(Read Only)							
Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
7369	SMITH	CLERK	7902	1980-12-17	800.00	(NULL)	20
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30
7654	MARTIN	SALESMAN	7369	1981-09-28	1250.00	1400.00	30
7844	TURNER	SALESMAN	7698	1981-08-09	1500.00	0.00	30
7876	ADAMS	CLERK	7788	1987-07-13	1100.00	(NULL)	20
7900	JAMES	CLERK	7698	1981-03-12	950.00	(NULL)	30
7934	MILLER	CLERK	7782	1982-01-23	1300.00	(NULL)	10

select \* from emp where job in (select job from emp where ename = 'smith') or job in (select job from emp where ename = 'allen') LIMIT 0, 1000

**17) Any jobs of deptno 10 those that are not found in deptno 20.**

=> `SELECT DISTINCT job FROM emp WHERE deptno=10 AND job NOT IN (SELECT DISTINCT job FROM emp WHERE deptno=20);`



The screenshot shows a database query result window with a toolbar at the top containing icons for grid, text, and other functions. Below the toolbar, there is a dropdown menu set to '(Read Only)'. The main area of the window displays a table with two columns: 'job' and 'PRESIDENT'. The 'job' column is highlighted in orange. At the bottom of the window, the SQL query is displayed: `select distinct job from emp where deptno=10 and job not in (select distinct job from emp where deptno=20) LIMIT 0, 1000`.

job
PRESIDENT

**18) Find the highest sal of EMP table.**

=> `SELECT MAX(sal) AS HighestSalary FROM emp;`

The screenshot shows the SQL Developer interface with the '1 Result' tab active. The query 'select max(sal) as HighestSalary from emp LIMIT 0, 1000' is executed. The result is displayed in a table with one column 'HighestSalary' and one row with the value '5000.00'.

HighestSalary
5000.00

select max(sal) as HighestSalary from emp LIMIT 0, 1000

## 19) Find details of highest paid employee.

=> SELECT \* FROM emp WHERE Sal = (SELECT MAX(Sal) FROM emp);

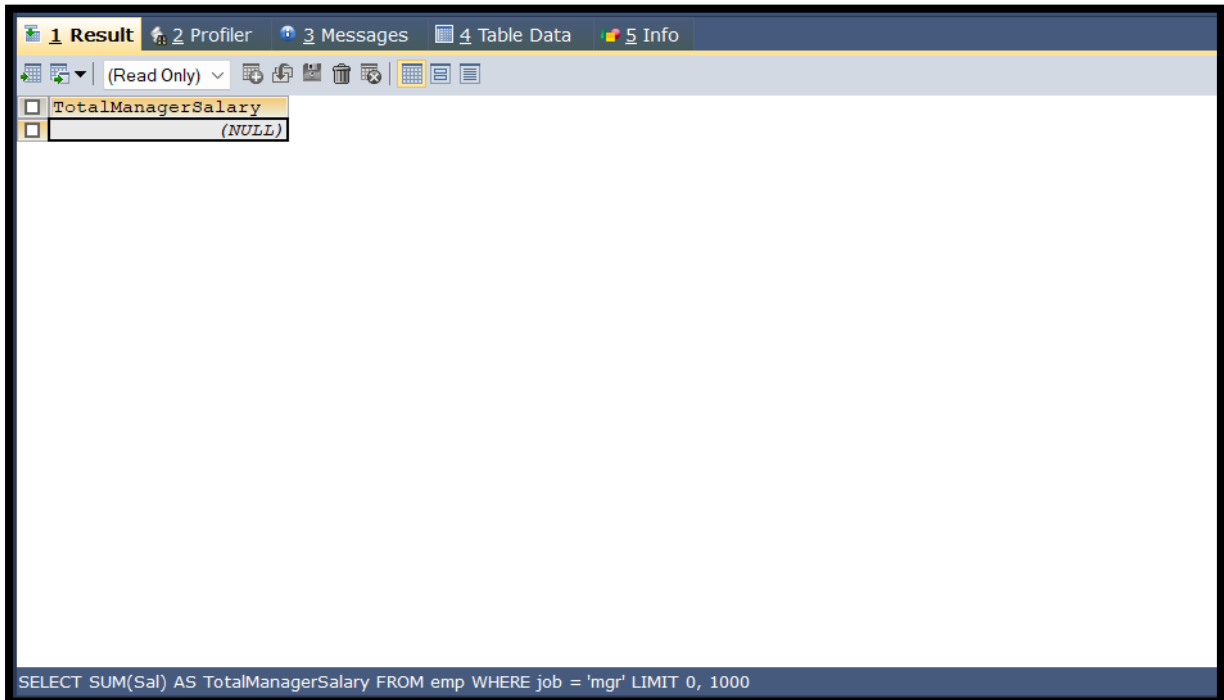
The screenshot shows the SQL Developer interface with the '1 Result' tab active. The query 'SELECT \* FROM emp WHERE Sal = (SELECT MAX(Sal) FROM emp) LIMIT 0, 1000' is executed. The result is displayed in a table with columns: Empno, Ename, Job, Mgr, Hiredate, Sal, Comm, and Deptno. The result shows one row for employee 7839, KING, PRESIDENT, with a salary of 5000.00.

Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
7839	KING	PRESIDENT	(NULL)	1981-11-17	5000.00	(NULL)	10

SELECT \* FROM emp WHERE Sal = (SELECT MAX(Sal) FROM emp) LIMIT 0, 1000

**20) Find the total sal given to the MGR.**

=> `SELECT SUM(Sal) AS TotalManagerSalary  
FROM emp WHERE job = 'mgr';`



TotalManagerSalary
(NULL)

SELECT SUM(Sal) AS TotalManagerSalary FROM emp WHERE job = 'mgr' LIMIT 0, 1000

**21) List the emps whose names contains 'A'.**

=> `SELECT * FROM emp WHERE Ename LIKE '%A%';`



1 Result 2 Profiler 3 Messages 4 Table Data 5 Info								
(Read Only)								
<input type="checkbox"/>	Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
<input type="checkbox"/>	7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30
<input type="checkbox"/>	7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30
<input type="checkbox"/>	7654	MARTIN	SALESMAN	7369	1981-09-28	1250.00	1400.00	30
<input type="checkbox"/>	7698	BLAKE	MANAGER	7839	1981-05-01	2850.00	(NULL)	30
<input type="checkbox"/>	7782	CLARK	MANAGER	7839	1981-06-09	2450.00	(NULL)	10
<input type="checkbox"/>	7876	ADAMS	CLERK	7788	1987-07-13	1100.00	(NULL)	20
<input type="checkbox"/>	7900	JAMES	CLERK	7698	1981-03-12	950.00	(NULL)	30

SELECT \* FROM emp WHERE Ename LIKE '%A%' LIMIT 0, 1000

**22) Find all the emps who earn the minimum Salary for each job wise in ascending order.**

=> SELECT E.\*

FROM emp E

JOIN (

SELECT Job, Sal

FROM emp

WHERE (Job, Sal) IN ( SELECT Job, MIN(Sal) FROM emp GROUP BY Job ) ) M ON E.Job = M.Job AND E.Sal = M.Sal ORDER BY E.Job ASC;

Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
7902	FORD	ANALYST	7566	1981-03-12	3000.00	(NULL)	20
7788	SCOTT	ANALYST	7566	1987-06-11	3000.00	(NULL)	20
7788	SCOTT	ANALYST	7566	1987-06-11	3000.00	(NULL)	20
7902	FORD	ANALYST	7566	1981-03-12	3000.00	(NULL)	20
7369	SMITH	CLERK	7902	1980-12-17	800.00	(NULL)	20
7782	CLARK	MANAGER	7839	1981-06-09	2450.00	(NULL)	10
7839	KING	PRESIDENT	(NULL)	1981-11-17	5000.00	(NULL)	10
7654	MARTIN	SALESMAN	7369	1981-09-28	1250.00	1400.00	30
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30
7654	MARTIN	SALESMAN	7369	1981-09-28	1250.00	1400.00	30

SELECT E.\* FROM emp E JOIN ( SELECT Job, Sal FROM emp WHERE (Job, Sal) IN ( SELECT Job, MIN(Sal) FROM emp GROUP BY Job ) ) M ON E.Job = M.Job AND E.Sal = M.Sal ORDER BY E.Job ASC LIMIT 0, 1000

**23) List the emps whose sal greater than Blake's sal.**

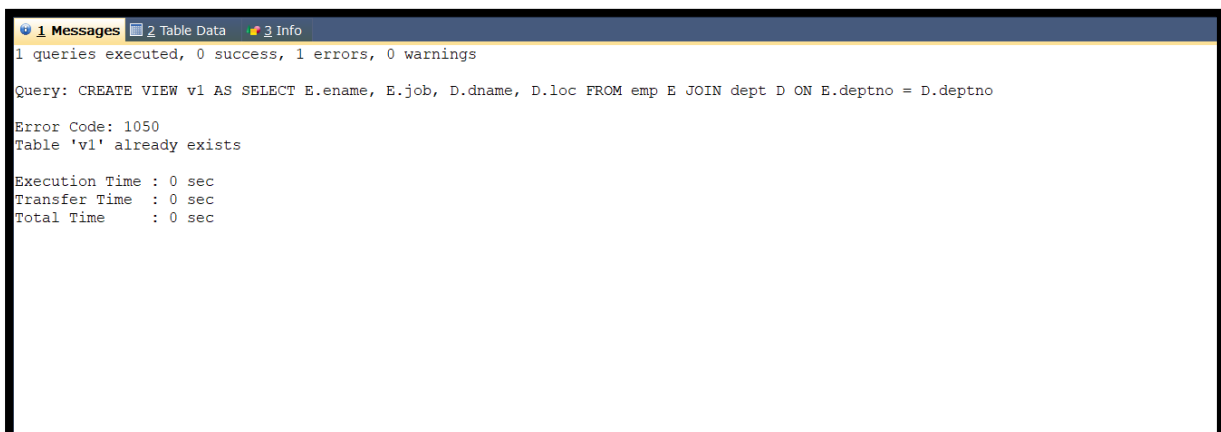
=> SELECT \* FROM emp WHERE Sal > (SELECT Sal FROM emp WHERE Ename = 'BLAKE');

Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
7566	JONES	MANAGER	7839	1981-04-02	2975.00	(NULL)	20
7788	SCOTT	ANALYST	7566	1987-06-11	3000.00	(NULL)	20
7839	KING	PRESIDENT	(NULL)	1981-11-17	5000.00	(NULL)	10
7902	FORD	ANALYST	7566	1981-03-12	3000.00	(NULL)	20

SELECT \* FROM emp WHERE Sal > (SELECT Sal FROM emp WHERE Ename = 'BLAKE') LIMIT 0, 1000

**24) Create view v1 to select ename, job, dname, loc whose deptno are same.**

```
=> CREATE VIEW v1 AS  
SELECT E.ename, E.job, D.dname, D.loc  
FROM emp E  
JOIN dept D  
ON E.deptno = D.deptno;
```

A screenshot of a SQL query execution window. The window has a title bar with '1 Messages', '2 Table Data', and '3 Info'. The main area displays the following text:  
1 queries executed, 0 success, 1 errors, 0 warnings  
Query: CREATE VIEW v1 AS SELECT E.ename, E.job, D.dname, D.loc FROM emp E JOIN dept D ON E.deptno = D.deptno  
Error Code: 1050  
Table 'v1' already exists  
Execution Time : 0 sec  
Transfer Time : 0 sec  
Total Time : 0 sec

**25) Create a procedure with dno as input parameter to fetch ename and dname.**

```
=> DELIMITER //  
CREATE PROCEDURE GetEmployeeDetails(IN dno INT)  
BEGIN
```

```
SELECT E.ename, D.dname  
FROM emp E  
JOIN dept D ON E.deptno = D.deptno  
WHERE E.deptno = dno;  
END //  
DELIMITER ;
```

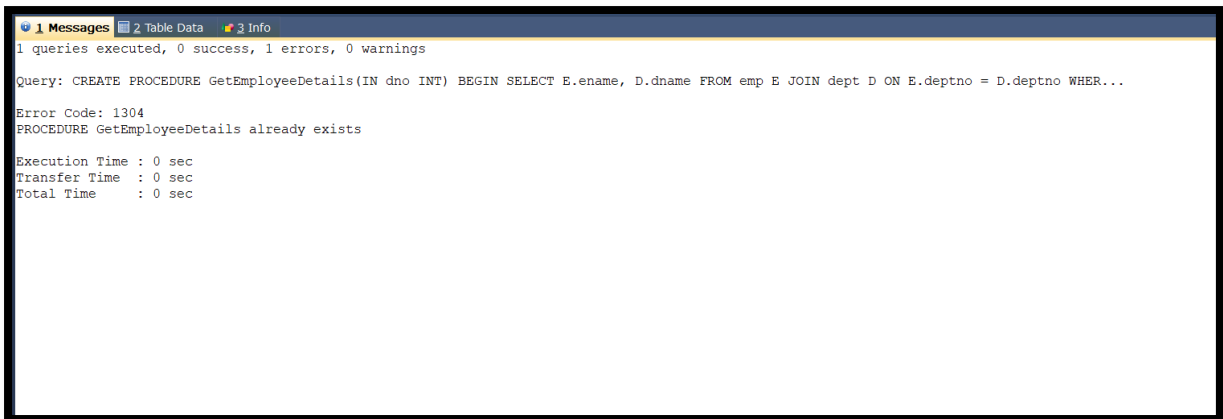


The screenshot shows a SQL execution window with three tabs: 'Messages', 'Table Data', and 'Info'. The 'Messages' tab is active, displaying the following text:

```
1 queries executed, 0 success, 1 errors, 0 warnings  
Query: CREATE PROCEDURE GetEmployeeDetails(IN dno INT) BEGIN SELECT E.ename, D.dname FROM emp E JOIN dept D ON E.deptno = D.deptno WHER...  
Error Code: 1304  
PROCEDURE GetEmployeeDetails already exists  
Execution Time : 0 sec  
Transfer Time : 0 sec  
Total Time : 0.004 sec
```

**26) Add column Pin with bigint data type in table student.**

=> ALTER TABLE student ADD COLUMN Pin BIGINT;



1 Messages 2 Table Data 3 Info

1 queries executed, 0 success, 1 errors, 0 warnings

Query: CREATE PROCEDURE GetEmployeeDetails(IN dno INT) BEGIN SELECT E.ename, D.dname FROM emp E JOIN dept D ON E.deptno = D.deptno WHER...

Error Code: 1304  
PROCEDURE GetEmployeeDetails already exists

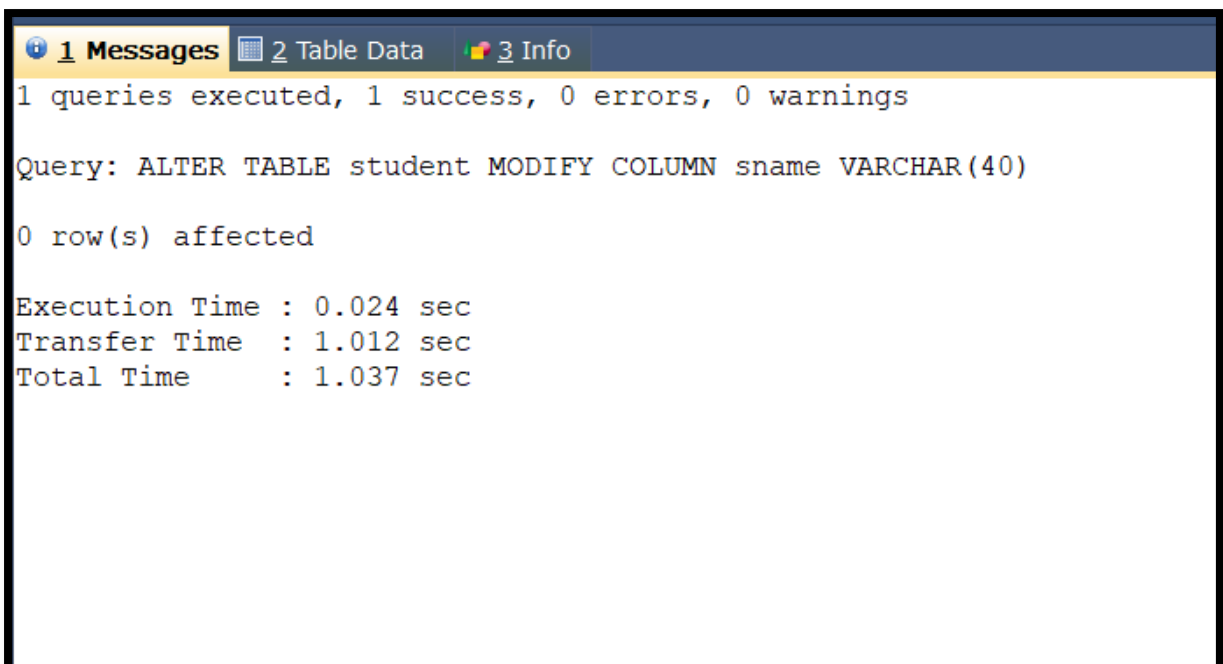
Execution Time : 0 sec  
Transfer Time : 0 sec  
Total Time : 0 sec

**27) Modify the student table to change the sname length from 14 to 40.**

=> ALTER TABLE student

MODIFY COLUMN sname VARCHAR(40);

DESCRIBE student;



1 Messages 2 Table Data 3 Info

1 queries executed, 1 success, 0 errors, 0 warnings

Query: ALTER TABLE student MODIFY COLUMN sname VARCHAR(40)

0 row(s) affected

Execution Time : 0.024 sec  
Transfer Time : 1.012 sec  
Total Time : 1.037 sec

**28) Create trigger to insert data in emp\_log table whenever any update of sal in EMP table. You can set action as 'New Salary'.**

=> DELIMITER //

```
CREATE TRIGGER after_sal_update
AFTER UPDATE ON EMP
FOR EACH ROW
BEGIN
    IF NEW.sal <> OLD.sal THEN
        INSERT INTO emp_log (Empno, ACTION,
ChangeDate)
        VALUES (NEW.Empno, 'New Salary', NOW());
    END IF;
END //
```

DELIMITER ;

```
1 Messages 2 Table Data 3 Info
1 queries executed, 1 success, 0 errors, 0 warnings

Query: CREATE TRIGGER after_sal_update AFTER UPDATE ON EMP FOR EACH ROW BEGIN IF NEW.sal <> OLD.sal THEN INSERT INTO emp_log (Empno, Ac...

0 row(s) affected

Execution Time : 0.019 sec
Transfer Time  : 1.003 sec
Total Time     : 1.023 sec
```

----- \* ----- \* ----- \*