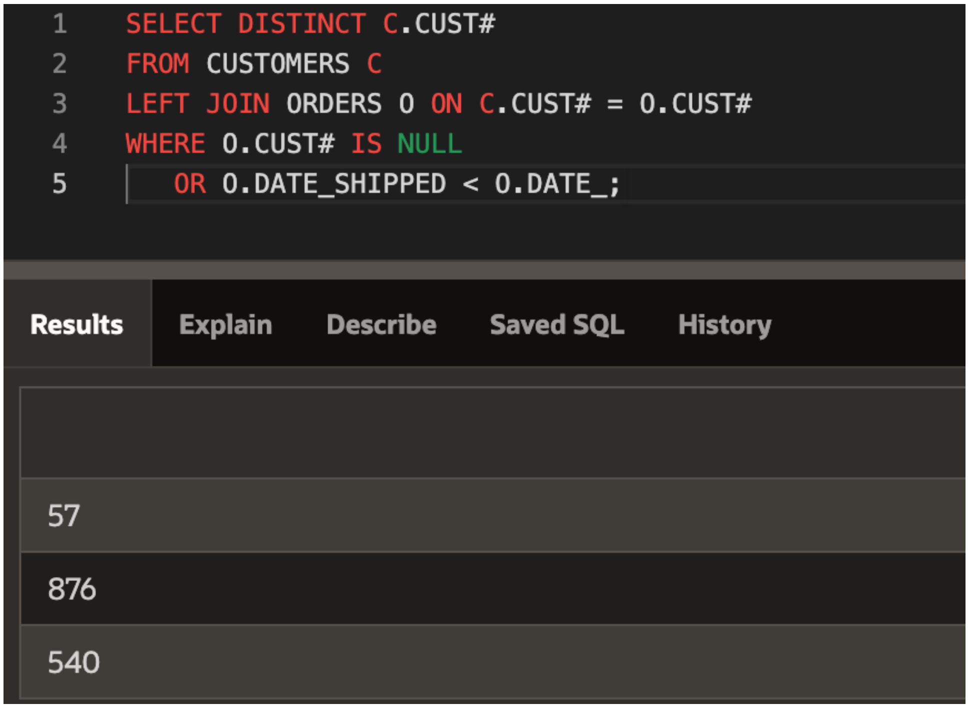
ASSIGNMENT 4

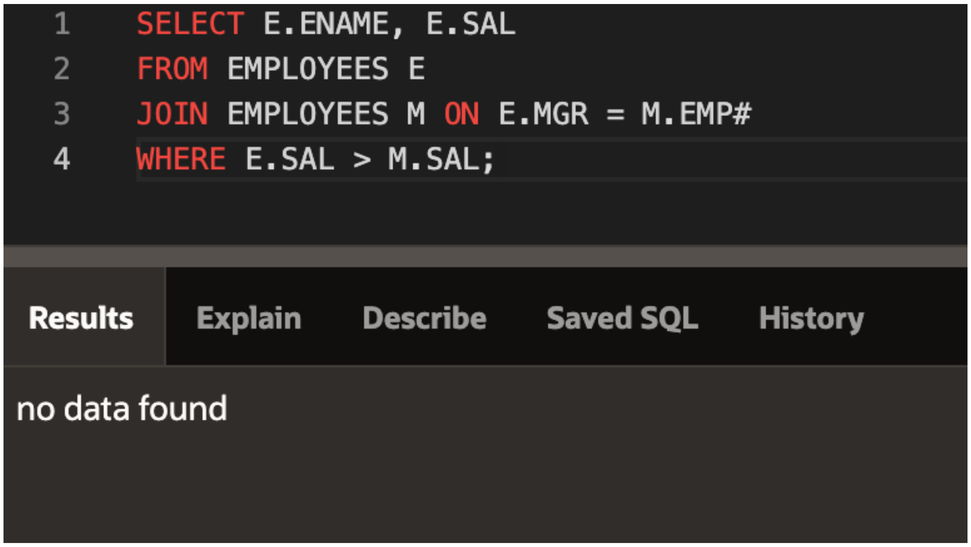
21) Find all the customers who have not places orders or who have received an order that was shipped before it was ordered.



```sql
SELECT DISTINCT C.CUST#
FROM CUSTOMERS C
LEFT JOIN ORDERS O ON C.CUST# = O.CUST#
WHERE O.CUST# IS NULL
   OR O.DATE_SHIPPED < O.DATE_;
```

```
57
876
540
```

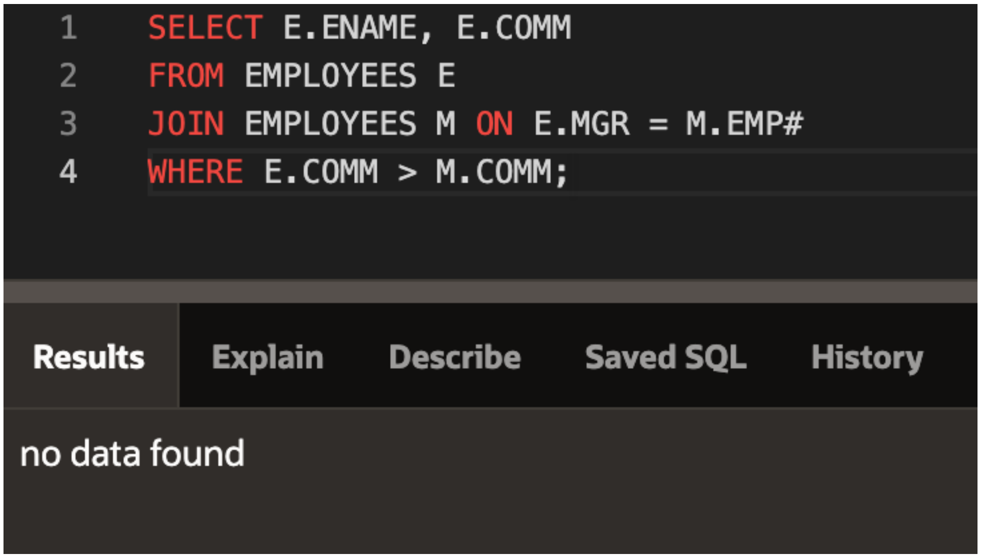
22) List the names and salaries of all the employees who earn more than their managers.



```sql
SELECT E.ENAME, E.SAL
FROM EMPLOYEES E
JOIN EMPLOYEES M ON E.MGR = M.EMP#
WHERE E.SAL > M.SAL;
```

```
no data found
```

23) List all employees who made more commission than their managers.



```
SELECT E.ENAME, E.COMM
FROM EMPLOYEES E
JOIN EMPLOYEES M ON E.MGR = M.EMP#
WHERE E.COMM > M.COMM;
```

```
no data found
```

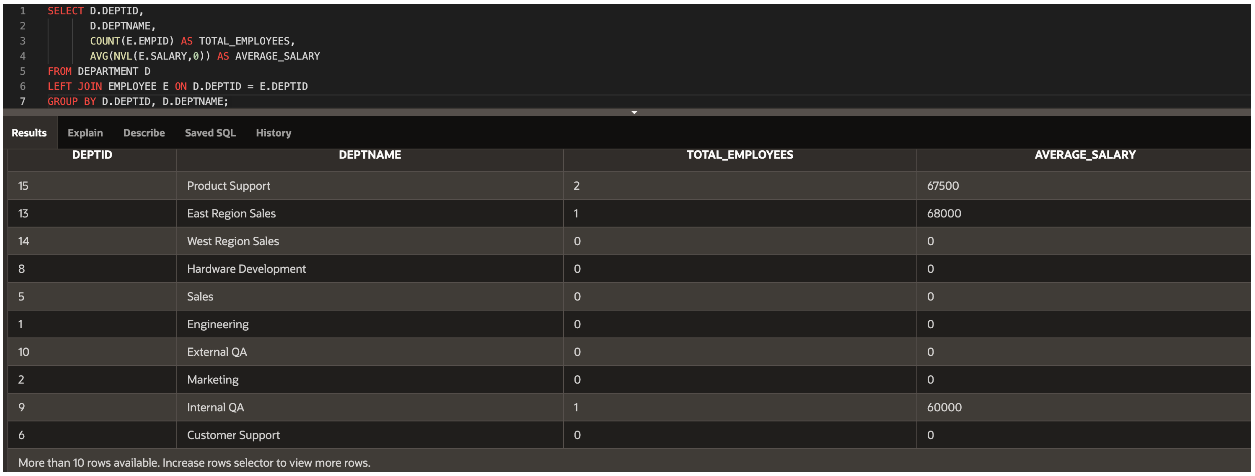
24) How does the database store a view? Demonstrate this with an SQL Statement.

Instead of storing a view as a physical table, a database stores it as a stored query in the database schema. The system tables in the database contain the definition of a view when it is created. The database engine dynamically runs the underlying query specified in the view's definition in response to a user query, and it then presents the results as though they were obtained from a table.

For example, creating a view named "EmployeeNames" that selects names from the "Employees" table:

CREATE VIEW EmployeeNames AS
SELECT Name FROM Employees;

25) Retrieve departments along with the total number of employees and the average salary in each department.

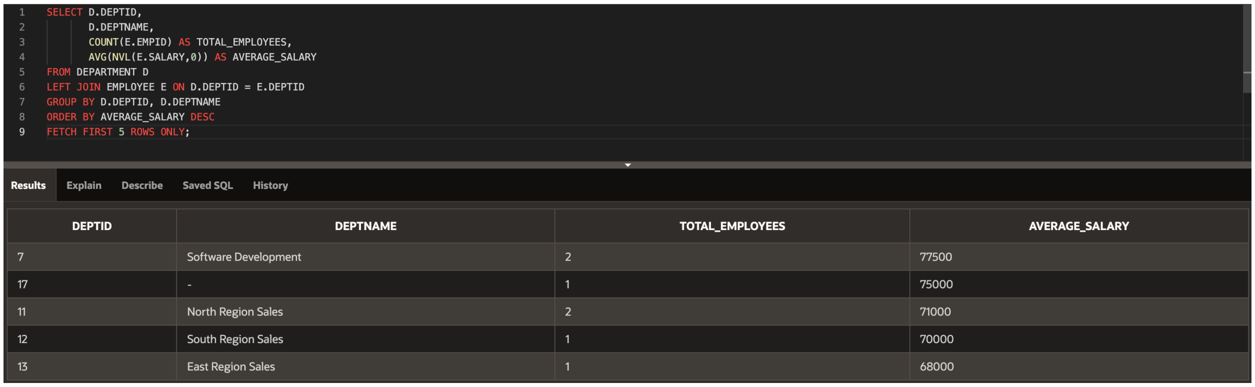


```
SELECT D.DEPTID,
       D.DEPTNAME,
       COUNT(E.EMPID) AS TOTAL_EMPLOYEES,
       AVG(NVL(E.SALARY,0)) AS AVERAGE_SALARY
FROM DEPARTMENT D
LEFT JOIN EMPLOYEE E ON D.DEPTID = E.DEPTID
GROUP BY D.DEPTID, D.DEPTNAME;
```

DEPTID	DEPTNAME	TOTAL_EMPLOYEES	AVERAGE_SALARY
15	Product Support	2	67500
13	East Region Sales	1	68000
14	West Region Sales	0	0
8	Hardware Development	0	0
5	Sales	0	0
1	Engineering	0	0
10	External QA	0	0
2	Marketing	0	0
9	Internal QA	1	60000
6	Customer Support	0	0

More than 10 rows available. Increase rows selector to view more rows.

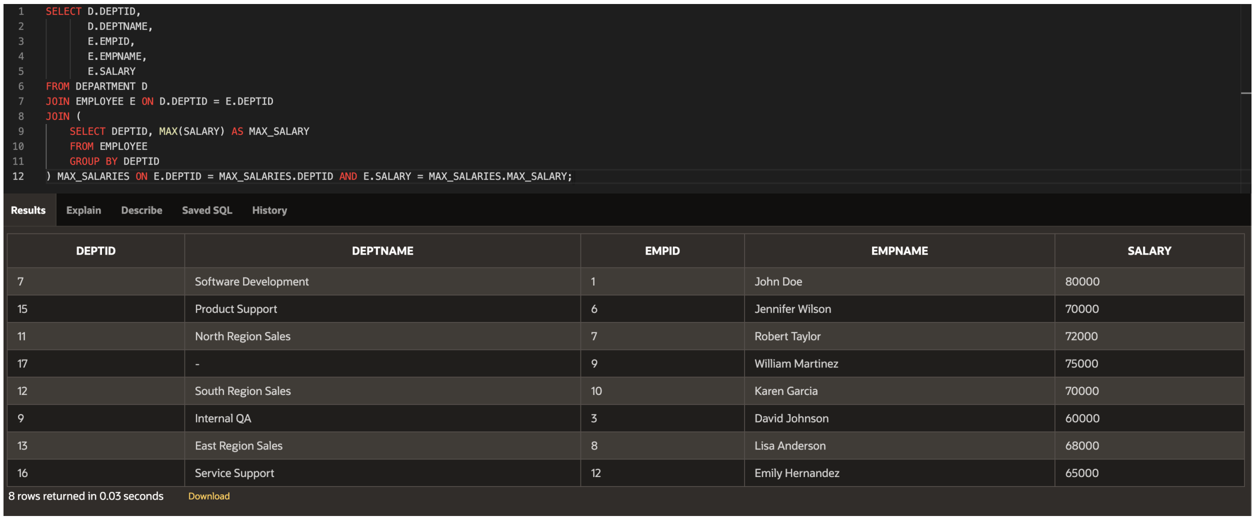
26) Retrieve the top 5 departments with the highest average salary.



```
SELECT D.DEPTID,
       D.DEPTNAME,
       COUNT(E.EMPID) AS TOTAL_EMPLOYEES,
       AVG(NVL(E.SALARY,0)) AS AVERAGE_SALARY
FROM DEPARTMENT D
LEFT JOIN EMPLOYEE E ON D.DEPTID = E.DEPTID
GROUP BY D.DEPTID, D.DEPTNAME
ORDER BY AVERAGE_SALARY DESC
FETCH FIRST 5 ROWS ONLY;
```

DEPTID	DEPTNAME	TOTAL_EMPLOYEES	AVERAGE_SALARY
7	Software Development	2	77500
17	-	1	75000
11	North Region Sales	2	71000
12	South Region Sales	1	70000
13	East Region Sales	1	68000

27) Retrieve departments along with the employee who has the highest salary in each department.

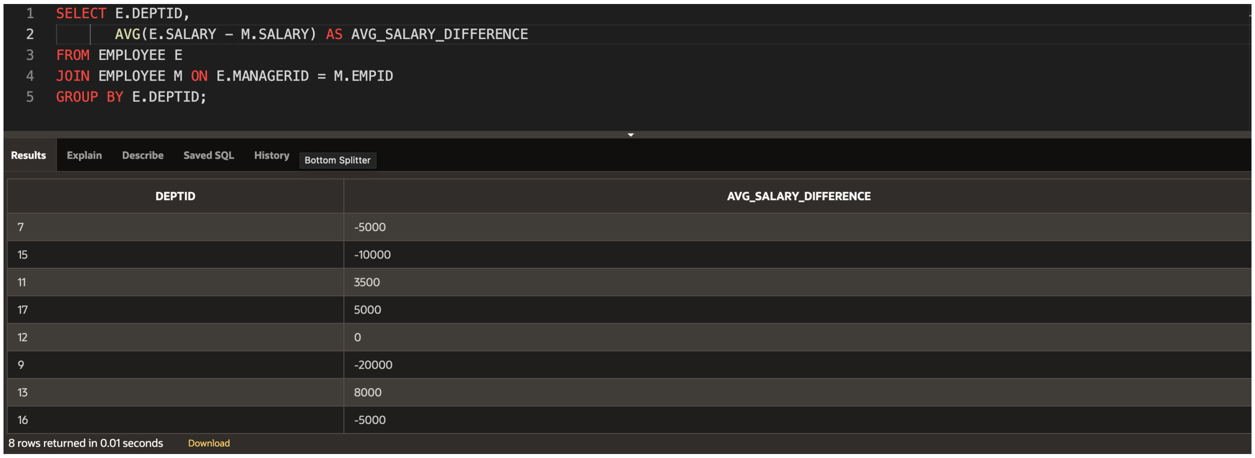


```
SELECT D.DEPTID,
       D.DEPTNAME,
       E.EMPID,
       E.EMPNAME,
       E.SALARY
FROM DEPARTMENT D
JOIN EMPLOYEE E ON D.DEPTID = E.DEPTID
JOIN (
    SELECT DEPTID, MAX(SALARY) AS MAX_SALARY
    FROM EMPLOYEE
    GROUP BY DEPTID
) MAX_SALARIES ON E.DEPTID = MAX_SALARIES.DEPTID AND E.SALARY = MAX_SALARIES.MAX_SALARY;
```

DEPTID	DEPTNAME	EMPID	EMPNAME	SALARY
7	Software Development	1	John Doe	80000
15	Product Support	6	Jennifer Wilson	70000
11	North Region Sales	7	Robert Taylor	72000
17	-	9	William Martinez	75000
12	South Region Sales	10	Karen Garcia	70000
9	Internal QA	3	David Johnson	60000
13	East Region Sales	8	Lisa Anderson	68000
16	Service Support	12	Emily Hernandez	65000

8 rows returned in 0.03 seconds

28) Retrieve the average salary difference between employees and their managers in each department.

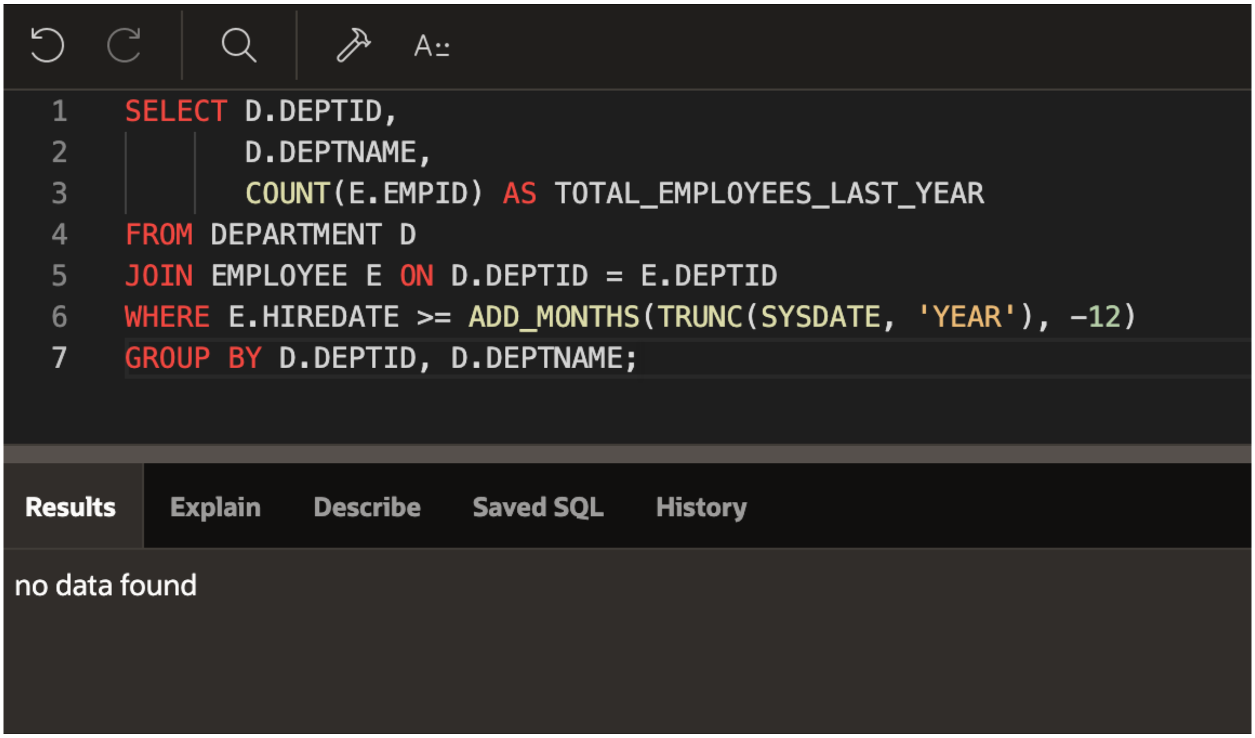


```
SELECT E.DEPTID,
       AVG(E.SALARY - M.SALARY) AS AVG_SALARY_DIFFERENCE
FROM EMPLOYEE E
JOIN EMPLOYEE M ON E.MANAGERID = M.EMPID
GROUP BY E.DEPTID;
```

DEPTID	AVG_SALARY_DIFFERENCE
7	-5000
15	-10000
11	3500
17	5000
12	0
9	-20000
13	8000
16	-5000

8 rows returned in 0.01 seconds

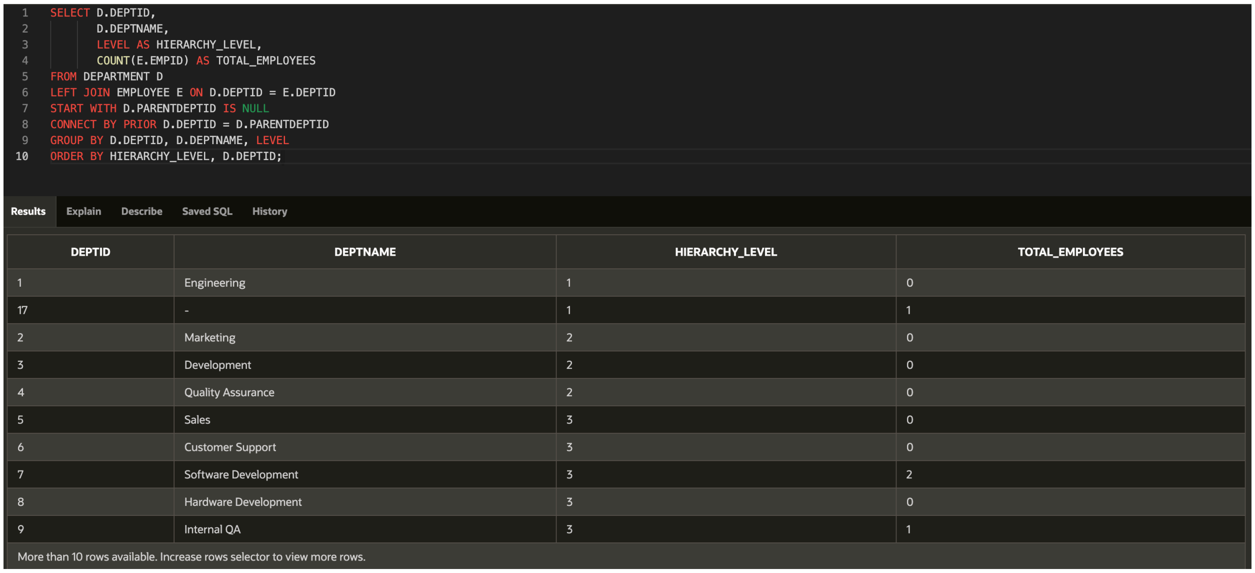
29) Retrieve departments along with the total number of employees who joined in the last year.



```
SELECT D.DEPTID,
       D.DEPTNAME,
       COUNT(E.EMPID) AS TOTAL_EMPLOYEES_LAST_YEAR
FROM DEPARTMENT D
JOIN EMPLOYEE E ON D.DEPTID = E.DEPTID
WHERE E.HIREDATE >= ADD_MONTHS(TRUNC(SYSDATE, 'YEAR'), -12)
GROUP BY D.DEPTID, D.DEPTNAME;
```

no data found

30) Retrieve the department hierarchy with the count of employees at each level of the hierarchy.

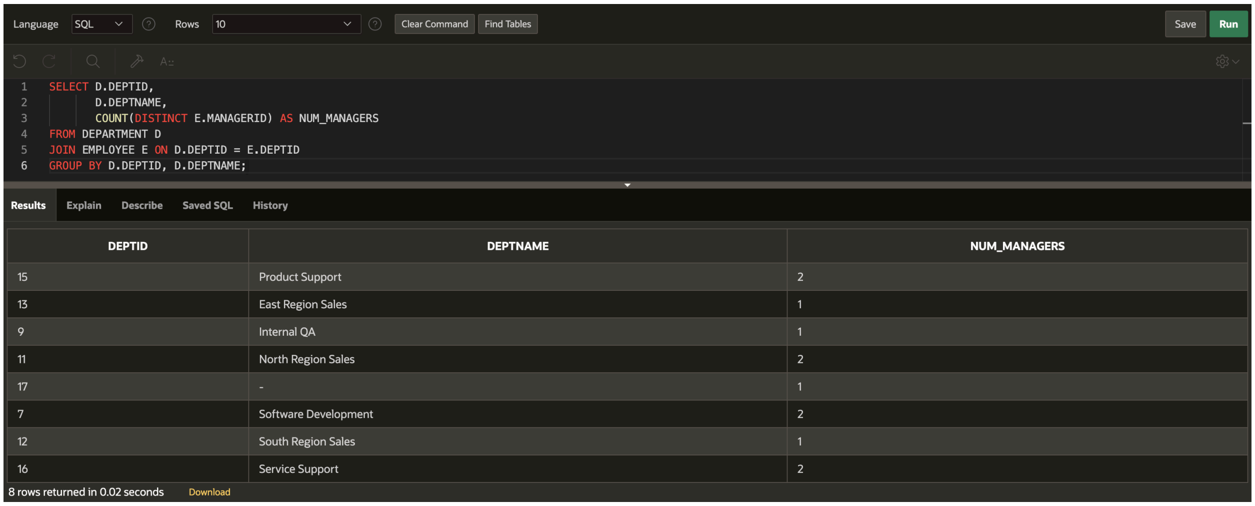


```
SELECT D.DEPTID,
       D.DEPTNAME,
       LEVEL AS HIERARCHY_LEVEL,
       COUNT(E.EMPID) AS TOTAL_EMPLOYEES
FROM DEPARTMENT D
LEFT JOIN EMPLOYEE E ON D.DEPTID = E.DEPTID
START WITH D.PARENTDEPTID IS NULL
CONNECT BY PRIOR D.DEPTID = D.PARENTDEPTID
GROUP BY D.DEPTID, D.DEPTNAME, LEVEL
ORDER BY HIERARCHY_LEVEL, D.DEPTID;
```

DEPTID	DEPTNAME	HIERARCHY_LEVEL	TOTAL_EMPLOYEES
1	Engineering	1	0
17	-	1	1
2	Marketing	2	0
3	Development	2	0
4	Quality Assurance	2	0
5	Sales	3	0
6	Customer Support	3	0
7	Software Development	3	2
8	Hardware Development	3	0
9	Internal QA	3	1

More than 10 rows available. Increase rows selector to view more rows.

31) Retrieve departments along with the number of managers in each department.

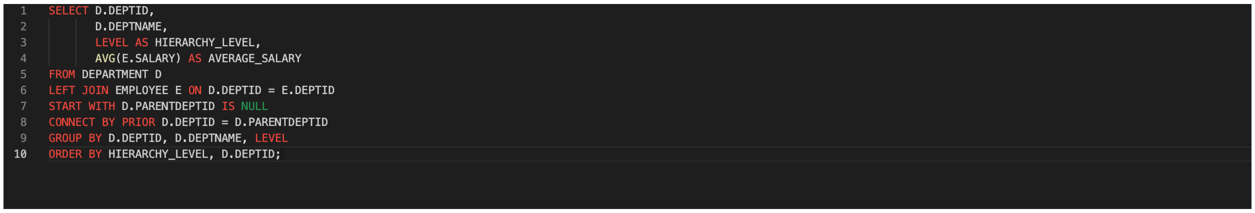


```
SELECT D.DEPTID,
       D.DEPTNAME,
       COUNT(DISTINCT E.MANAGERID) AS NUM_MANAGERS
FROM DEPARTMENT D
JOIN EMPLOYEE E ON D.DEPTID = E.DEPTID
GROUP BY D.DEPTID, D.DEPTNAME;
```

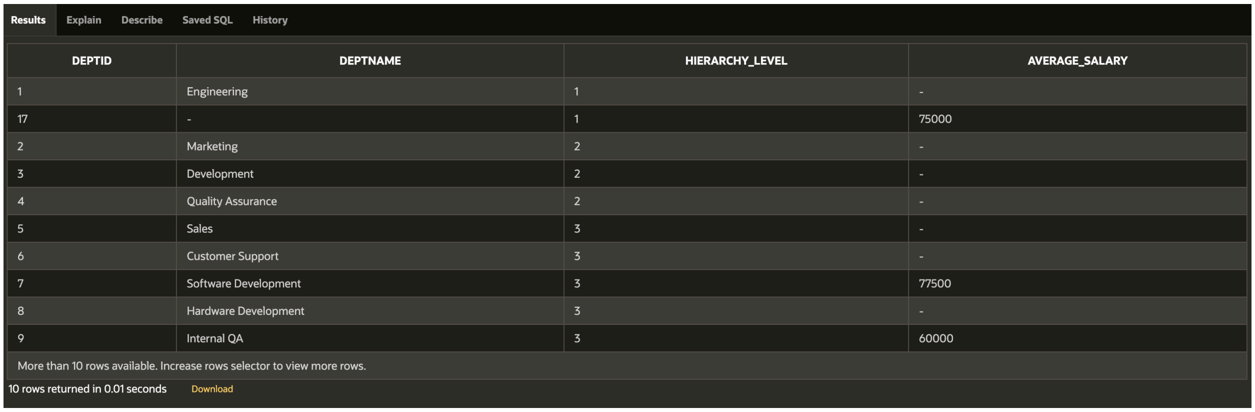
DEPTID	DEPTNAME	NUM_MANAGERS
15	Product Support	2
13	East Region Sales	1
9	Internal QA	1
11	North Region Sales	2
17	-	1
7	Software Development	2
12	South Region Sales	1
16	Service Support	2

8 rows returned in 0.02 seconds

32) Retrieve the hierarchical structure of departments along with the average salary at each level.



```
SELECT D.DEPTID,
       D.DEPTNAME,
       LEVEL AS HIERARCHY_LEVEL,
       AVG(E.SALARY) AS AVERAGE_SALARY
FROM DEPARTMENT D
LEFT JOIN EMPLOYEE E ON D.DEPTID = E.DEPTID
START WITH D.PARENTDEPTID IS NULL
CONNECT BY PRIOR D.DEPTID = D.PARENTDEPTID
GROUP BY D.DEPTID, D.DEPTNAME, LEVEL
ORDER BY HIERARCHY_LEVEL, D.DEPTID;
```

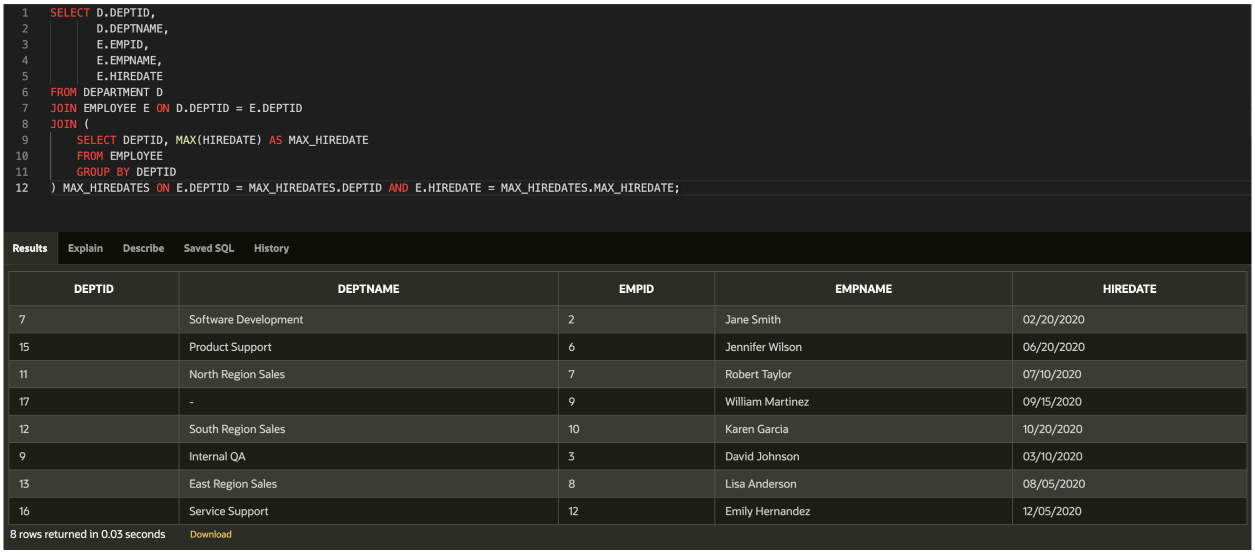


DEPTID	DEPTNAME	HIERARCHY_LEVEL	AVERAGE_SALARY
1	Engineering	1	-
17	-	1	75000
2	Marketing	2	-
3	Development	2	-
4	Quality Assurance	2	-
5	Sales	3	-
6	Customer Support	3	-
7	Software Development	3	77500
8	Hardware Development	3	-
9	Internal QA	3	60000

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.01 seconds

33) Retrieve departments along with the employee with the longest tenure in each department.

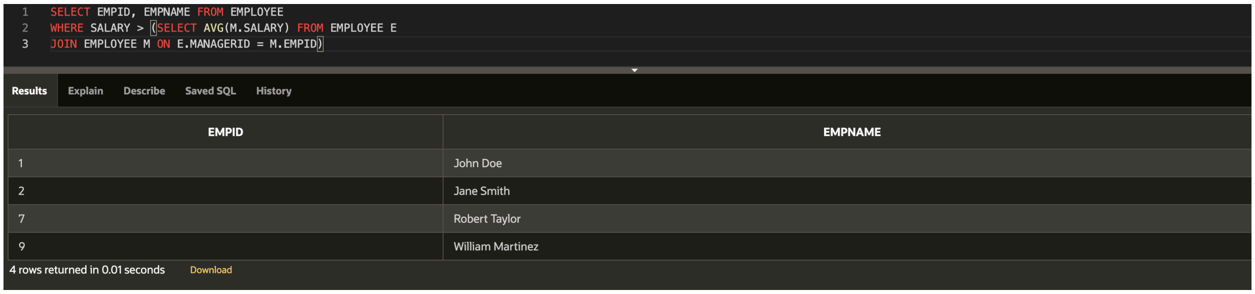


```sql
SELECT D.DEPTID,
       D.DEPTNAME,
       E.EMPID,
       E.EMPNAME,
       E.HIREDATE
FROM DEPARTMENT D
JOIN EMPLOYEE E ON D.DEPTID = E.DEPTID
JOIN (
    SELECT DEPTID, MAX(HIREDATE) AS MAX_HIREDATE
    FROM EMPLOYEE
    GROUP BY DEPTID
) MAX_HIREDATES ON E.DEPTID = MAX_HIREDATES.DEPTID AND E.HIREDATE = MAX_HIREDATES.MAX_HIREDATE;
```

DEPTID	DEPTNAME	EMPID	EMPNAME	HIREDATE
7	Software Development	2	Jane Smith	02/20/2020
15	Product Support	6	Jennifer Wilson	06/20/2020
11	North Region Sales	7	Robert Taylor	07/10/2020
17	-	9	William Martinez	09/15/2020
12	South Region Sales	10	Karen Garcia	10/20/2020
9	Internal QA	3	David Johnson	03/10/2020
13	East Region Sales	8	Lisa Anderson	08/05/2020
16	Service Support	12	Emily Hernandez	12/05/2020

8 rows returned in 0.03 seconds

34) Retrieve employees who have a salary higher than the average salary of employees managed by them.

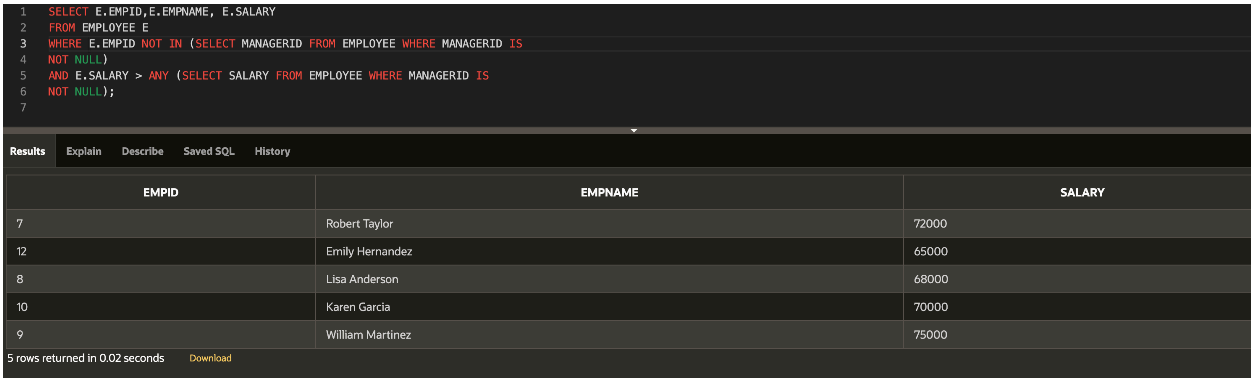


```sql
SELECT EMPID, EMPNAME FROM EMPLOYEE
WHERE SALARY > (SELECT AVG(M.SALARY) FROM EMPLOYEE E
JOIN EMPLOYEE M ON E.MANAGERID = M.EMPID)
```

EMPID	EMPNAME
1	John Doe
2	Jane Smith
7	Robert Taylor
9	William Martinez

4 rows returned in 0.01 seconds

35) Retrieve employees who are not managers but have a higher salary than at least one manager.

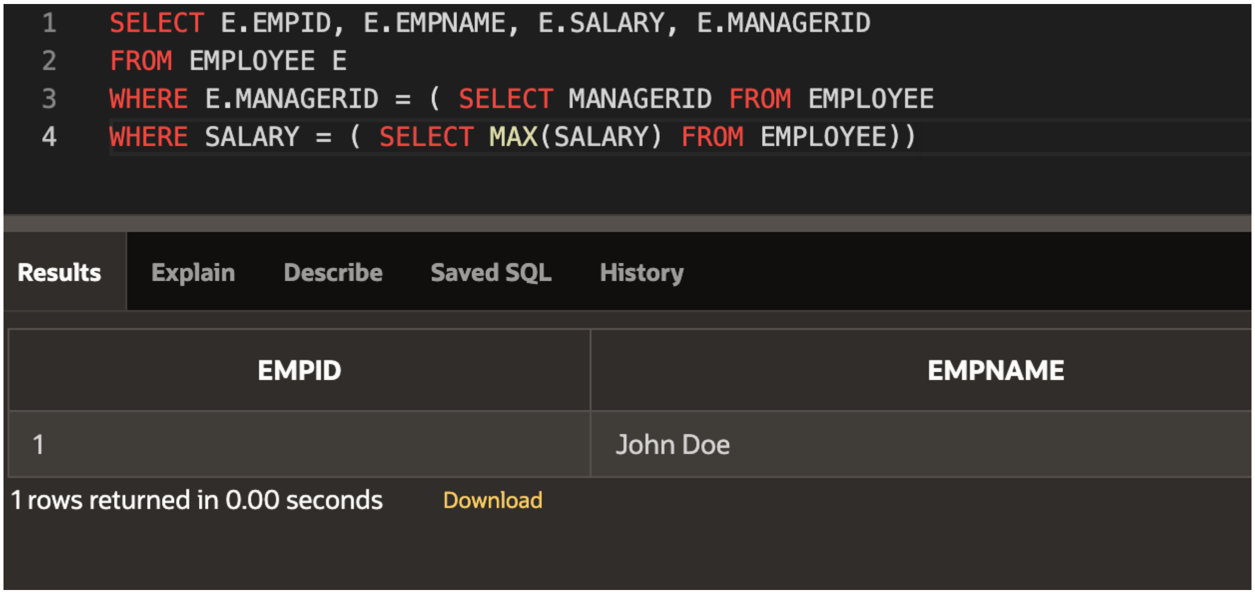


```sql
SELECT E.EMPID,E.EMPNAME, E.SALARY
FROM EMPLOYEE E
WHERE E.EMPID NOT IN (SELECT MANAGERID FROM EMPLOYEE WHERE MANAGERID IS
NOT NULL)
AND E.SALARY > ANY (SELECT SALARY FROM EMPLOYEE WHERE MANAGERID IS
NOT NULL);
```

EMPID	EMPNAME	SALARY
7	Robert Taylor	72000
12	Emily Hernandez	65000
8	Lisa Anderson	68000
10	Karen Garcia	70000
9	William Martinez	75000

5 rows returned in 0.02 seconds

36) Retrieve employees who have the same manager as the employee with the highest salary.

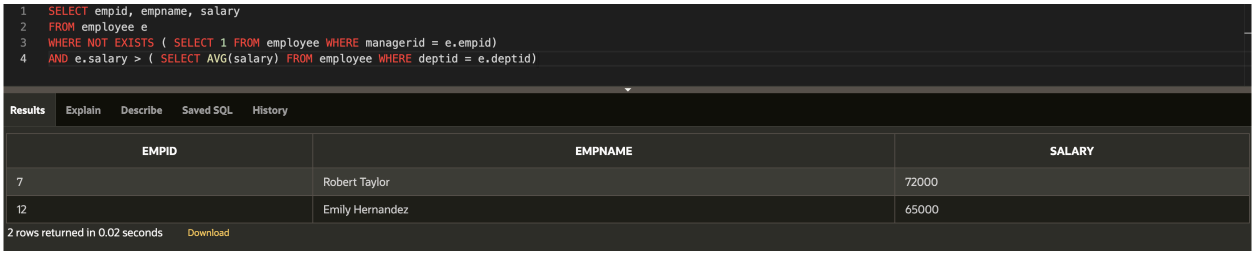


```sql
SELECT E.EMPID, E.EMPNAME, E.SALARY, E.MANAGERID
FROM EMPLOYEE E
WHERE E.MANAGERID = ( SELECT MANAGERID FROM EMPLOYEE
WHERE SALARY = ( SELECT MAX(SALARY) FROM EMPLOYEE))
```

EMPID	EMPNAME
1	John Doe

1 rows returned in 0.00 seconds

37) Retrieve employees who do not have any subordinates and have a salary higher than the average salary in their department.



```sql
SELECT empid, empname, salary
FROM employee e
WHERE NOT EXISTS ( SELECT 1 FROM employee WHERE managerid = e.empid)
AND e.salary > ( SELECT AVG(salary) FROM employee WHERE deptid = e.deptid)
```

EMPID	EMPNAME	SALARY
7	Robert Taylor	72000
12	Emily Hernandez	65000

2 rows returned in 0.02 seconds