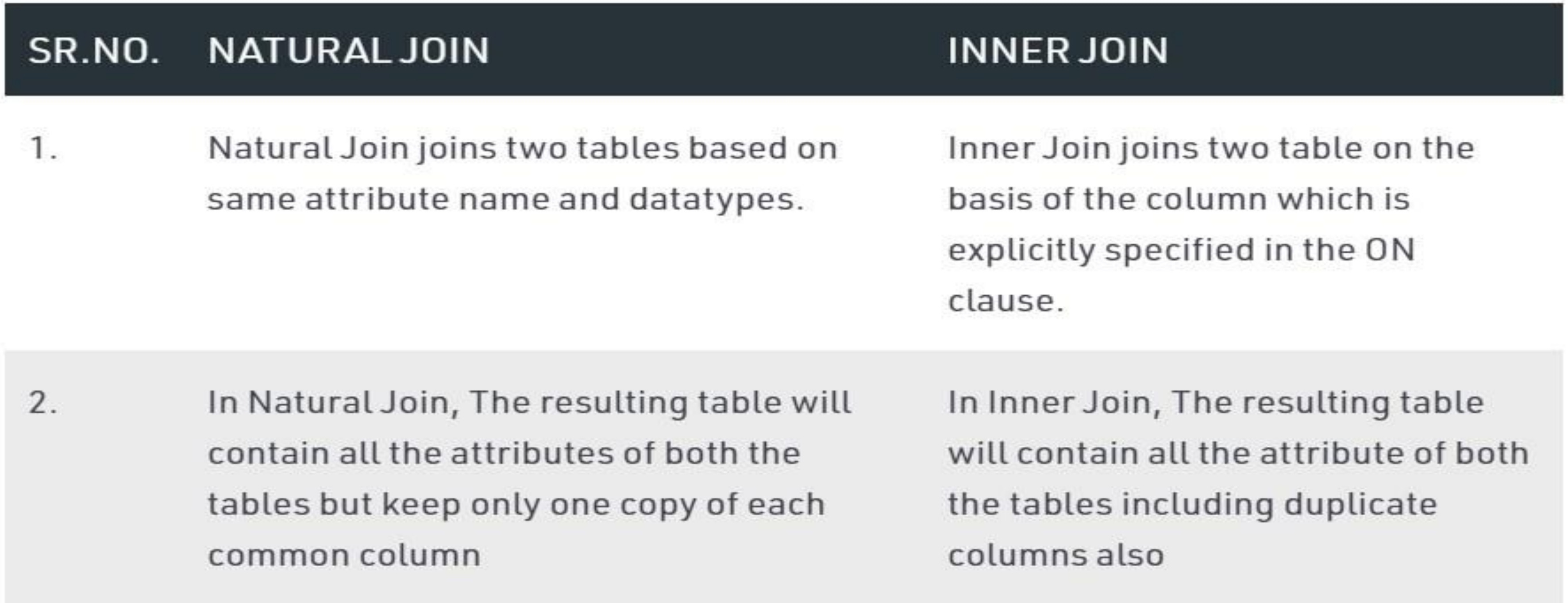
DATABASE SYSTEMS

JOINS

By Sana Faiz
Sana.faiz.muet83@gmail.com

INNER JOIN

- Inner join and natural join are almost same but there is a slight difference between them. The difference is in natural join no need to specify condition but in inner join condition is obligatory. If we do specify the condition in inner join , it resultant tables is like a cartesian product.



SR.NO.	NATURAL JOIN	INNER JOIN
1.	Natural Join joins two tables based on same attribute name and datatypes.	Inner Join joins two table on the basis of the column which is explicitly specified in the ON clause.
2.	In Natural Join, The resulting table will contain all the attributes of both the tables but keep only one copy of each common column	In Inner Join, The resulting table will contain all the attribute of both the tables including duplicate columns also

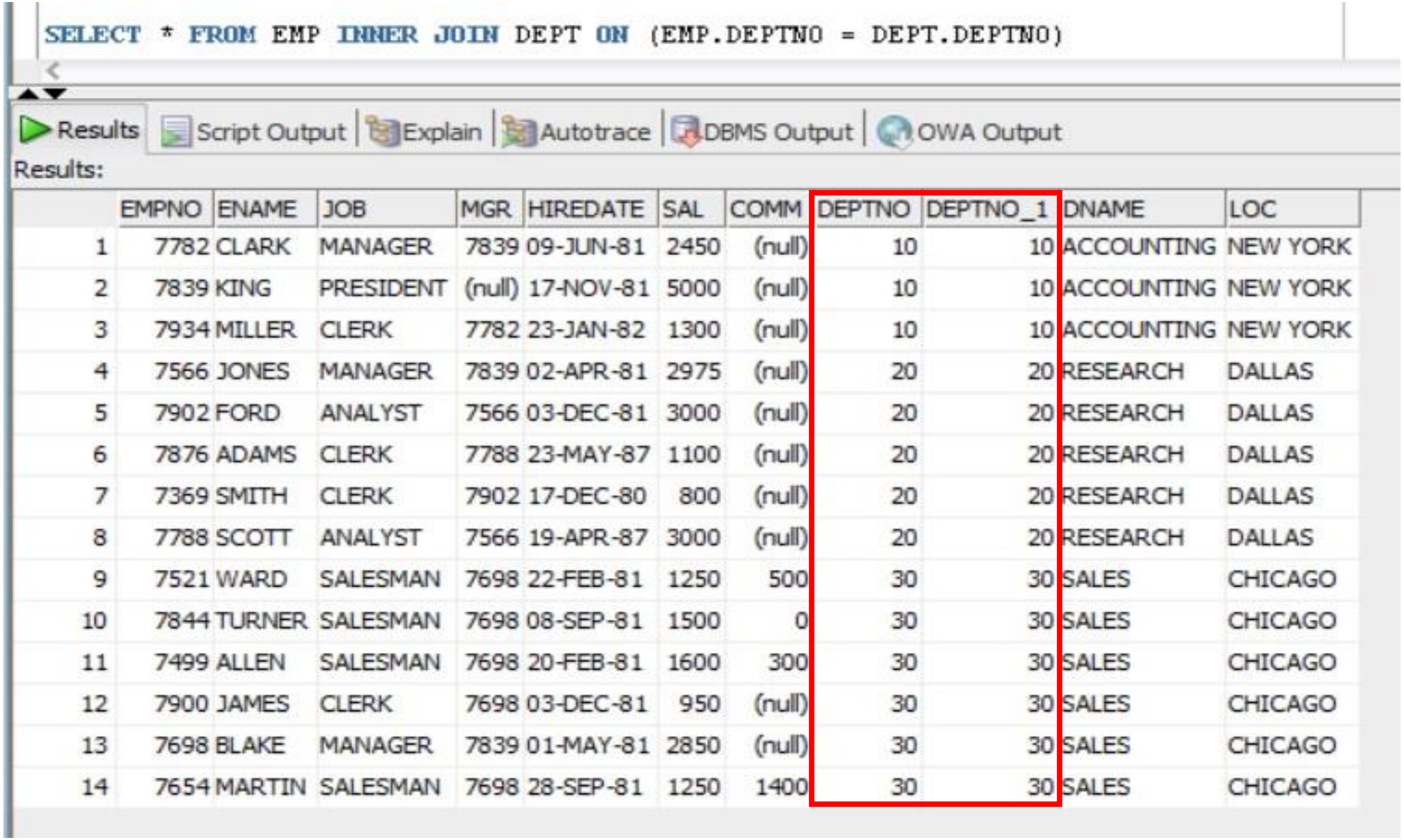
JOIN METHOD - ANSI Syntax (SQL/99 JOIN Approach)

EQUI-JOIN FORMATION THROUGH INNER JOIN

- EQUI-JOIN can be formed through INNER JOIN by using the ON clause.

SYNTAX:

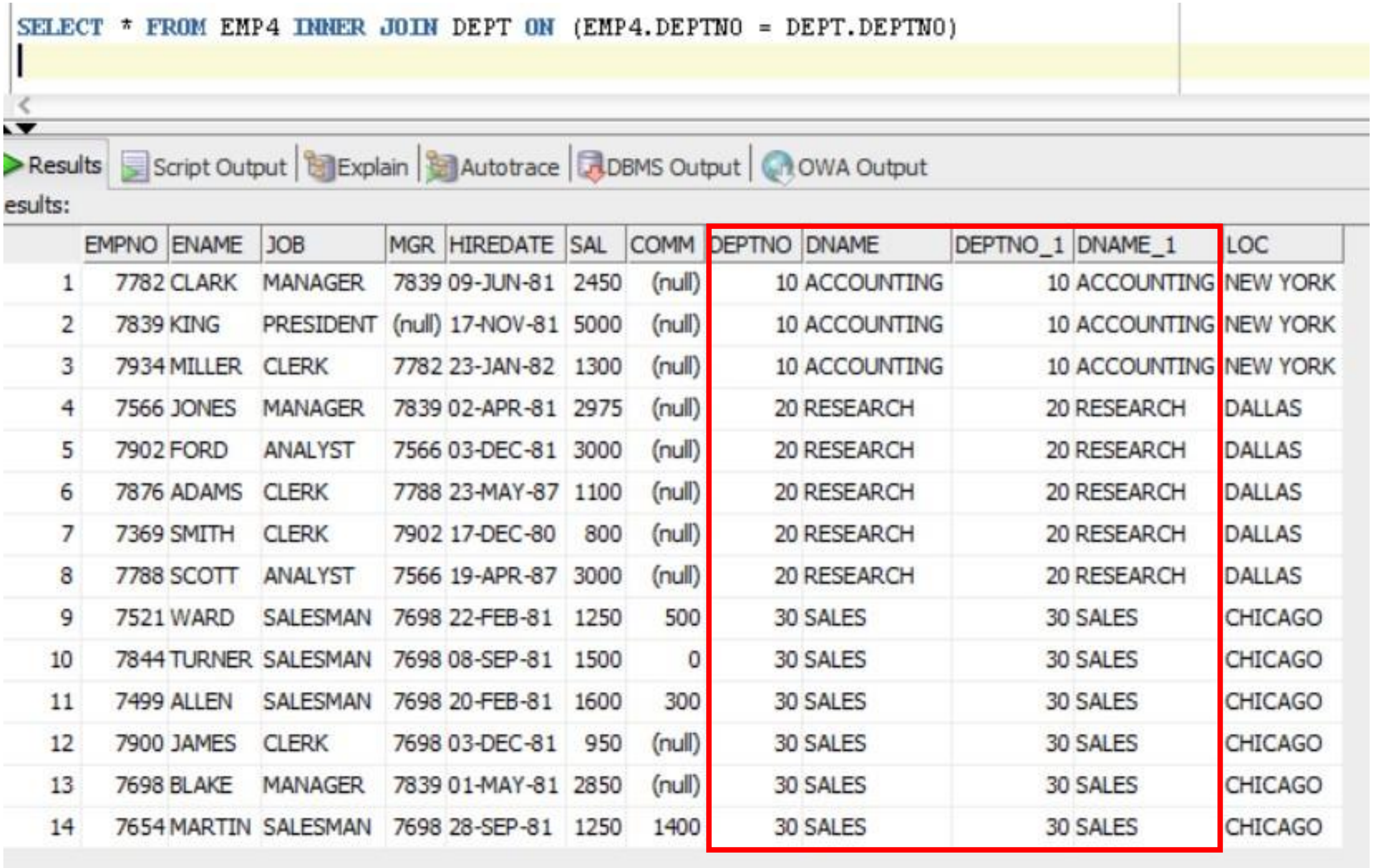
```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON (table1.column_name = table2.column_name) ;


SELECT column_name(s)
FROM (( table1
INNER JOIN table2 ON table1.column_name = table2.column_name )
INNER JOIN table3 ON table2.column_name = table3.column_name ) ;
```



```
SELECT * FROM EMP INNER JOIN DEPT ON (EMP.DEPTNO = DEPT.DEPTNO)
```

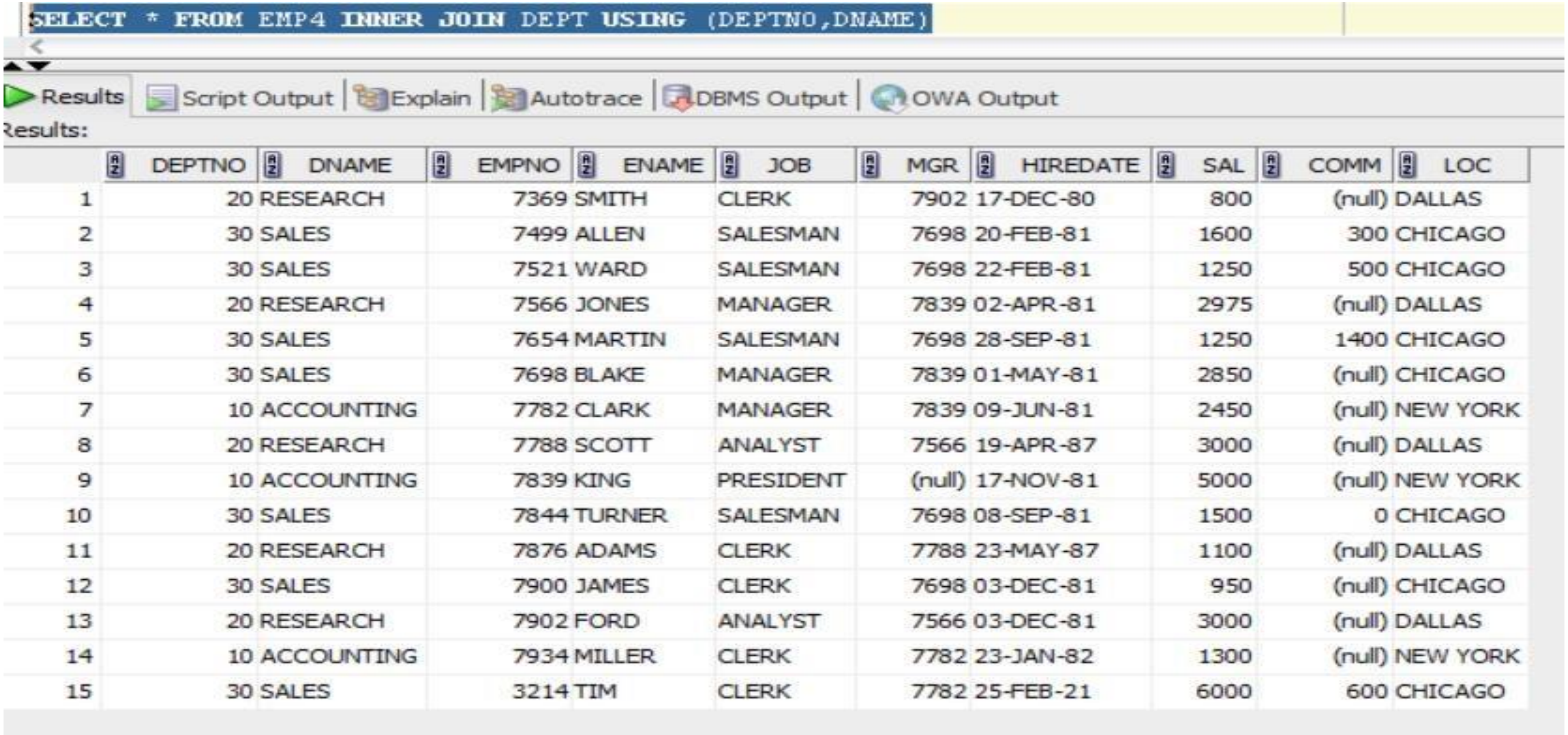
Results:

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DEPTNO_1	DNAME	LOC
1	7782	CLARK	MANAGER	7839	09-JUN-81	2450	(null)	10	10	ACCOUNTING	NEW YORK
2	7839	KING	PRESIDENT	(null)	17-NOV-81	5000	(null)	10	10	ACCOUNTING	NEW YORK
3	7934	MILLER	CLERK	7782	23-JAN-82	1300	(null)	10	10	ACCOUNTING	NEW YORK
4	7566	JONES	MANAGER	7839	02-APR-81	2975	(null)	20	20	RESEARCH	DALLAS
5	7902	FORD	ANALYST	7566	03-DEC-81	3000	(null)	20	20	RESEARCH	DALLAS
6	7876	ADAMS	CLERK	7788	23-MAY-87	1100	(null)	20	20	RESEARCH	DALLAS
7	7369	SMITH	CLERK	7902	17-DEC-80	800	(null)	20	20	RESEARCH	DALLAS
8	7788	SCOTT	ANALYST	7566	19-APR-87	3000	(null)	20	20	RESEARCH	DALLAS
9	7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30	30	SALES	CHICAGO
10	7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30	30	SALES	CHICAGO
11	7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30	30	SALES	CHICAGO
12	7900	JAMES	CLERK	7698	03-DEC-81	950	(null)	30	30	SALES	CHICAGO
13	7698	BLAKE	MANAGER	7839	01-MAY-81	2850	(null)	30	30	SALES	CHICAGO
14	7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30	30	SALES	CHICAGO



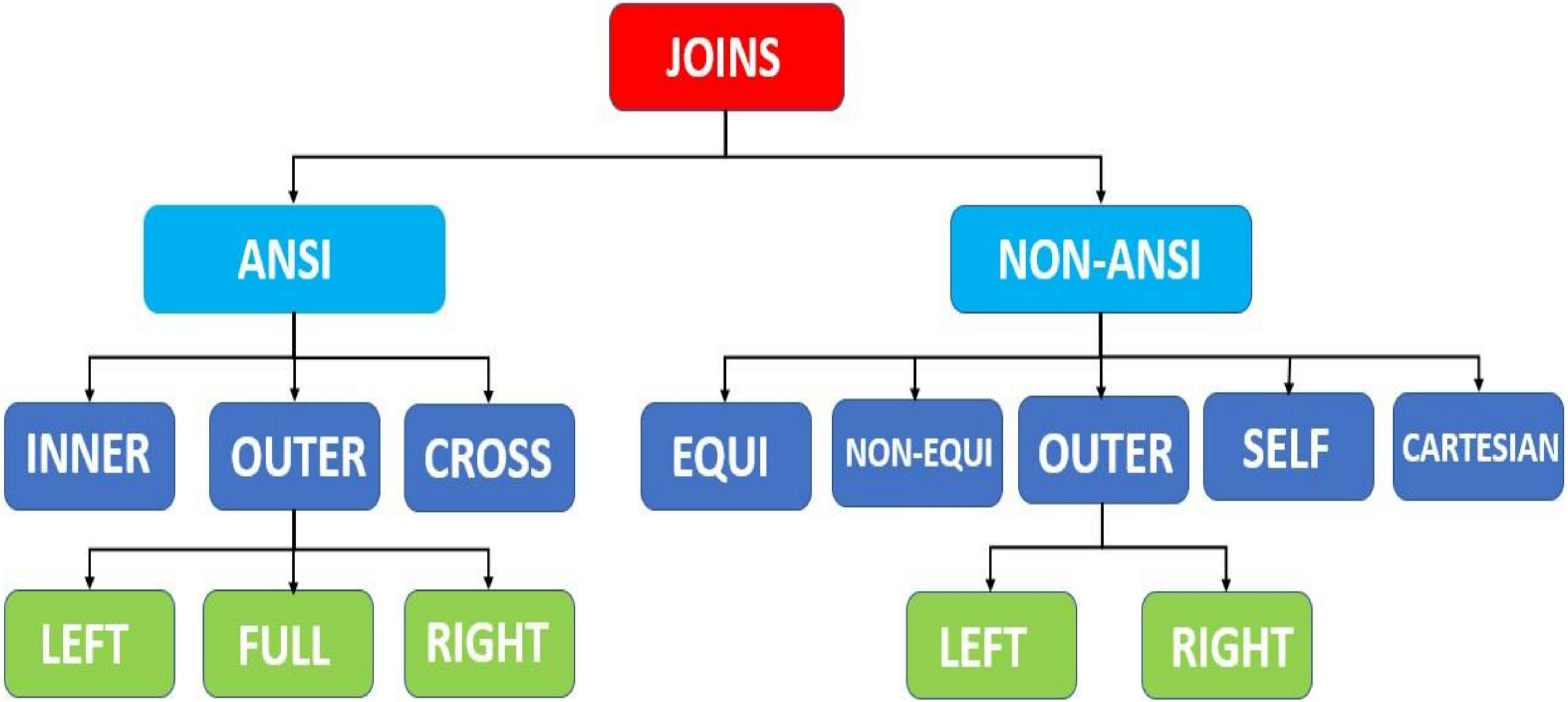
```
SELECT * FROM EMP4 INNER JOIN DEPT ON (EMP4.DEPTNO = DEPT.DEPTNO)
```

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DNAME	DEPTNO_1	DNAME_1	LOC
1	7782	CLARK	MANAGER	7839	09-JUN-81	2450	(null)	10	ACCOUNTING	10	ACCOUNTING	NEW YORK
2	7839	KING	PRESIDENT	(null)	17-NOV-81	5000	(null)	10	ACCOUNTING	10	ACCOUNTING	NEW YORK
3	7934	MILLER	CLERK	7782	23-JAN-82	1300	(null)	10	ACCOUNTING	10	ACCOUNTING	NEW YORK
4	7566	JONES	MANAGER	7839	02-APR-81	2975	(null)	20	RESEARCH	20	RESEARCH	DALLAS
5	7902	FORD	ANALYST	7566	03-DEC-81	3000	(null)	20	RESEARCH	20	RESEARCH	DALLAS
6	7876	ADAMS	CLERK	7788	23-MAY-87	1100	(null)	20	RESEARCH	20	RESEARCH	DALLAS
7	7369	SMITH	CLERK	7902	17-DEC-80	800	(null)	20	RESEARCH	20	RESEARCH	DALLAS
8	7788	SCOTT	ANALYST	7566	19-APR-87	3000	(null)	20	RESEARCH	20	RESEARCH	DALLAS
9	7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30	SALES	30	SALES	CHICAGO
10	7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30	SALES	30	SALES	CHICAGO
11	7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30	SALES	30	SALES	CHICAGO
12	7900	JAMES	CLERK	7698	03-DEC-81	950	(null)	30	SALES	30	SALES	CHICAGO
13	7698	BLAKE	MANAGER	7839	01-MAY-81	2850	(null)	30	SALES	30	SALES	CHICAGO
14	7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30	SALES	30	SALES	CHICAGO



```sql
SELECT * FROM EMP4 INNER JOIN DEPT USING (DEPTNO,DNAME)
```

	DEPTNO	DNAME	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	LOC
1	20	RESEARCH	7369	SMITH	CLERK	7902	17-DEC-80	800	(null)	DALLAS
2	30	SALES	7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	CHICAGO
3	30	SALES	7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	CHICAGO
4	20	RESEARCH	7566	JONES	MANAGER	7839	02-APR-81	2975	(null)	DALLAS
5	30	SALES	7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	CHICAGO
6	30	SALES	7698	BLAKE	MANAGER	7839	01-MAY-81	2850	(null)	CHICAGO
7	10	ACCOUNTING	7782	CLARK	MANAGER	7839	09-JUN-81	2450	(null)	NEW YORK
8	20	RESEARCH	7788	SCOTT	ANALYST	7566	19-APR-87	3000	(null)	DALLAS
9	10	ACCOUNTING	7839	KING	PRESIDENT	(null)	17-NOV-81	5000	(null)	NEW YORK
10	30	SALES	7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	CHICAGO
11	20	RESEARCH	7876	ADAMS	CLERK	7788	23-MAY-87	1100	(null)	DALLAS
12	30	SALES	7900	JAMES	CLERK	7698	03-DEC-81	950	(null)	CHICAGO
13	20	RESEARCH	7902	FORD	ANALYST	7566	03-DEC-81	3000	(null)	DALLAS
14	10	ACCOUNTING	7934	MILLER	CLERK	7782	23-JAN-82	1300	(null)	NEW YORK
15	30	SALES	3214	TIM	CLERK	7782	25-FEB-21	6000	600	CHICAGO



JOINS

ANSI

NON-ANSI

INNER

OUTER

CROSS

LEFT

FULL

RIGHT

EQUI

NON-EQUI

OUTER

SELF

CARTESIAN

LEFT

RIGHT

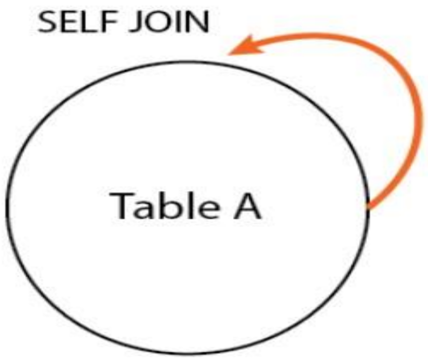
SELF-JOIN – TRADITIONAL METHOD

- A self join is a join in which a table is joined with itself (which is also called Unary relationships), especially when the table has a FOREIGN KEY which references its own PRIMARY KEY.

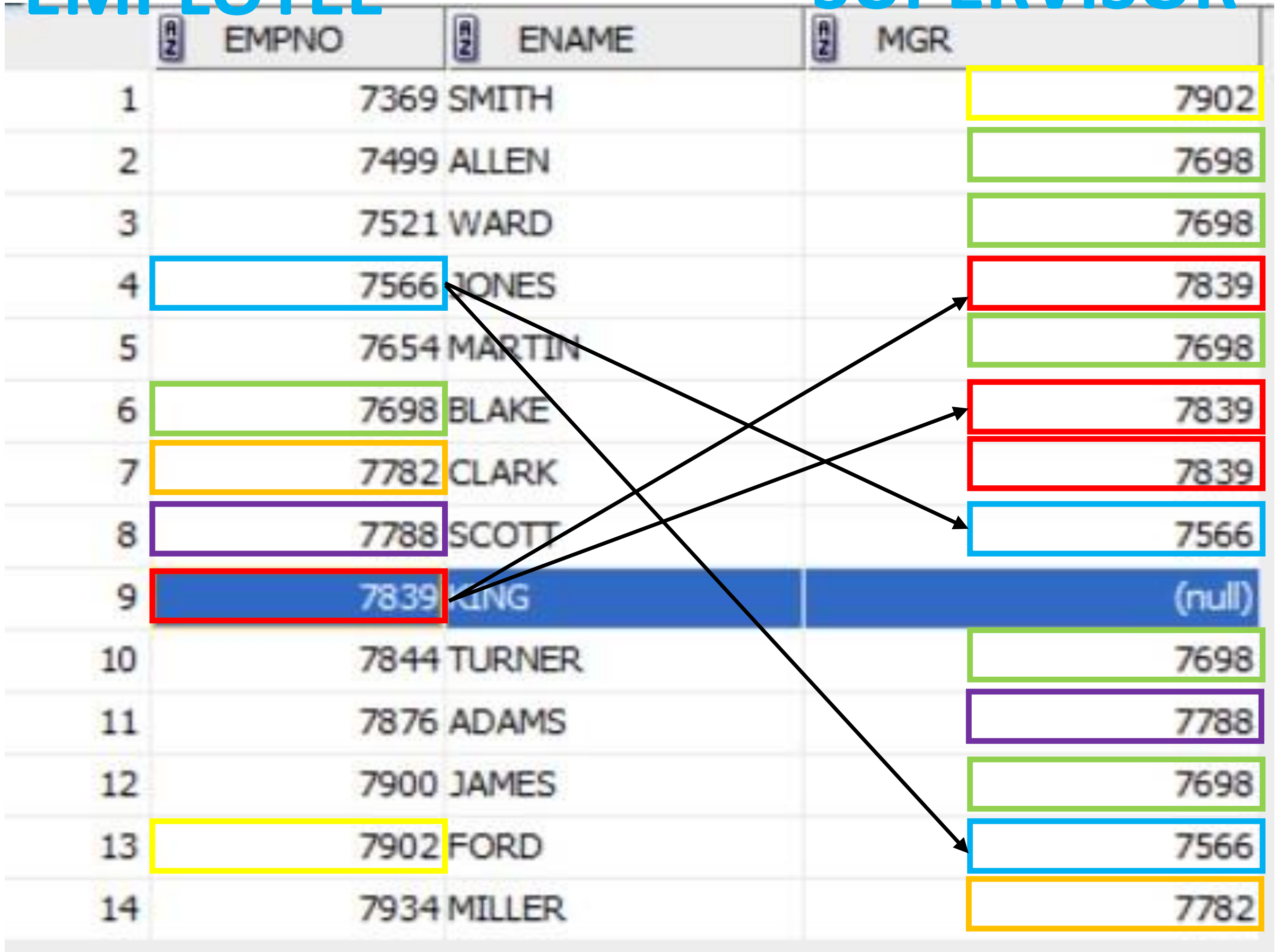
- To join a table to itself means that each row of the table is combined with itself and with every other row of the table.

- The self join can be viewed as a join of two copies of the same table. The table is not actually copied, but SQL performs the command as though it were.

- The syntax of the command for joining a table to itself is almost same as that for joining two different tables. To distinguish the column names from one another, aliases for are used, since both the tables have the same name. Table name aliases are defined in the FROM clause of the SELECT statement.



EMPLOYEE SUPERVISOR



	EMPNO	ENAME	MGR
1	7369	SMITH	7902
2	7499	ALLEN	7698
3	7521	WARD	7698
4	7566	JONES	7839
5	7654	MARTIN	7698
6	7698	BLAKE	7839
7	7782	CLARK	7839
8	7788	SCOTT	7566
9	7839	KING	(null)
10	7844	TURNER	7698
11	7876	ADAMS	7788
12	7900	JAMES	7698
13	7902	FORD	7566
14	7934	MILLER	7782

Unary relationship to emp

How the employees are related to themselves:

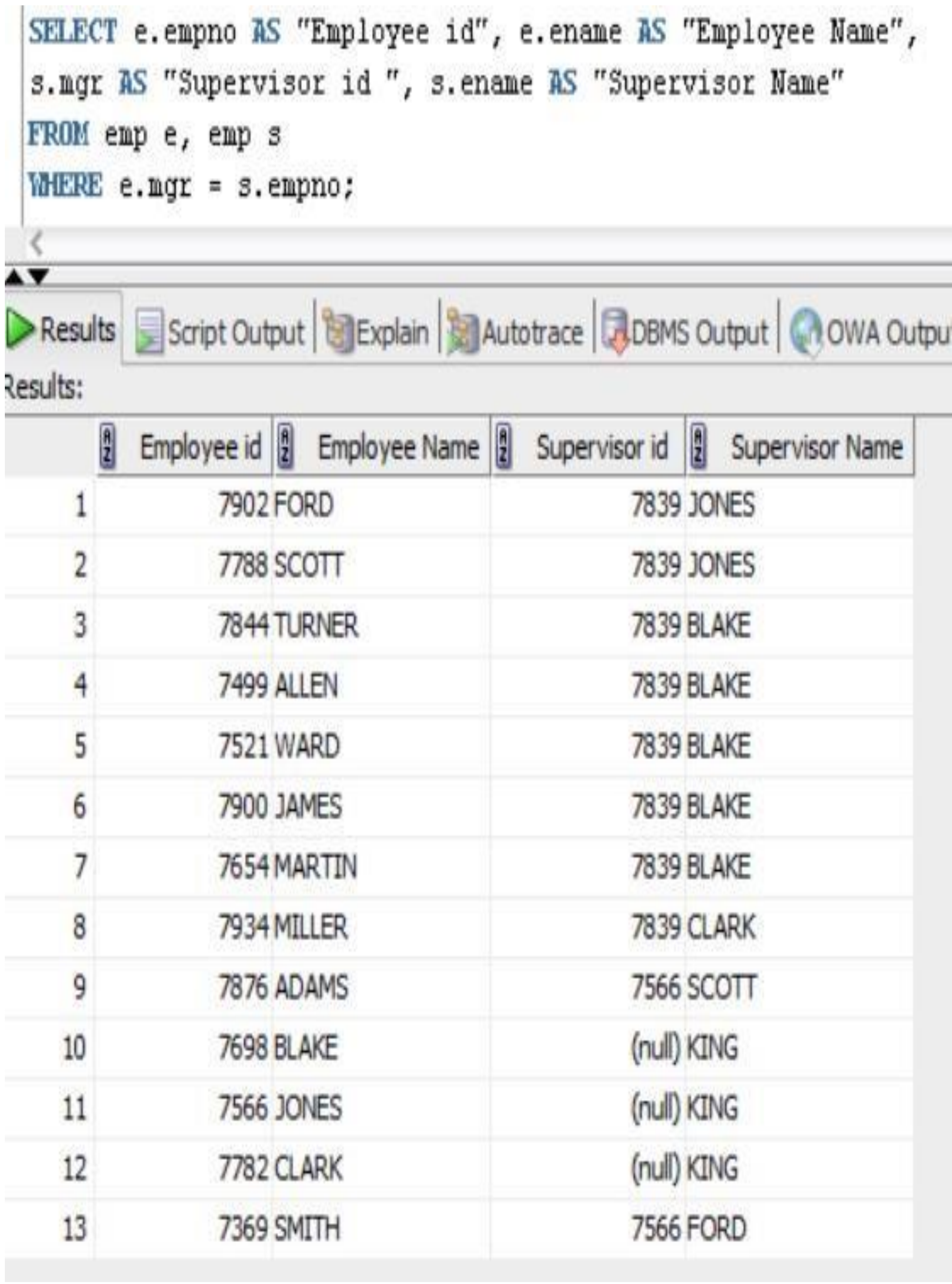
- An employee may report to another employee (supervisor).
- An employee may supervise himself ,to many employees (subordinates).

SYNTAX:

```
SELECT a.column_name, b.column_name [,….]
FROM table1 a, table1 b
WHERE a.common_column = b.common_column ;
```

EXAMPLE:

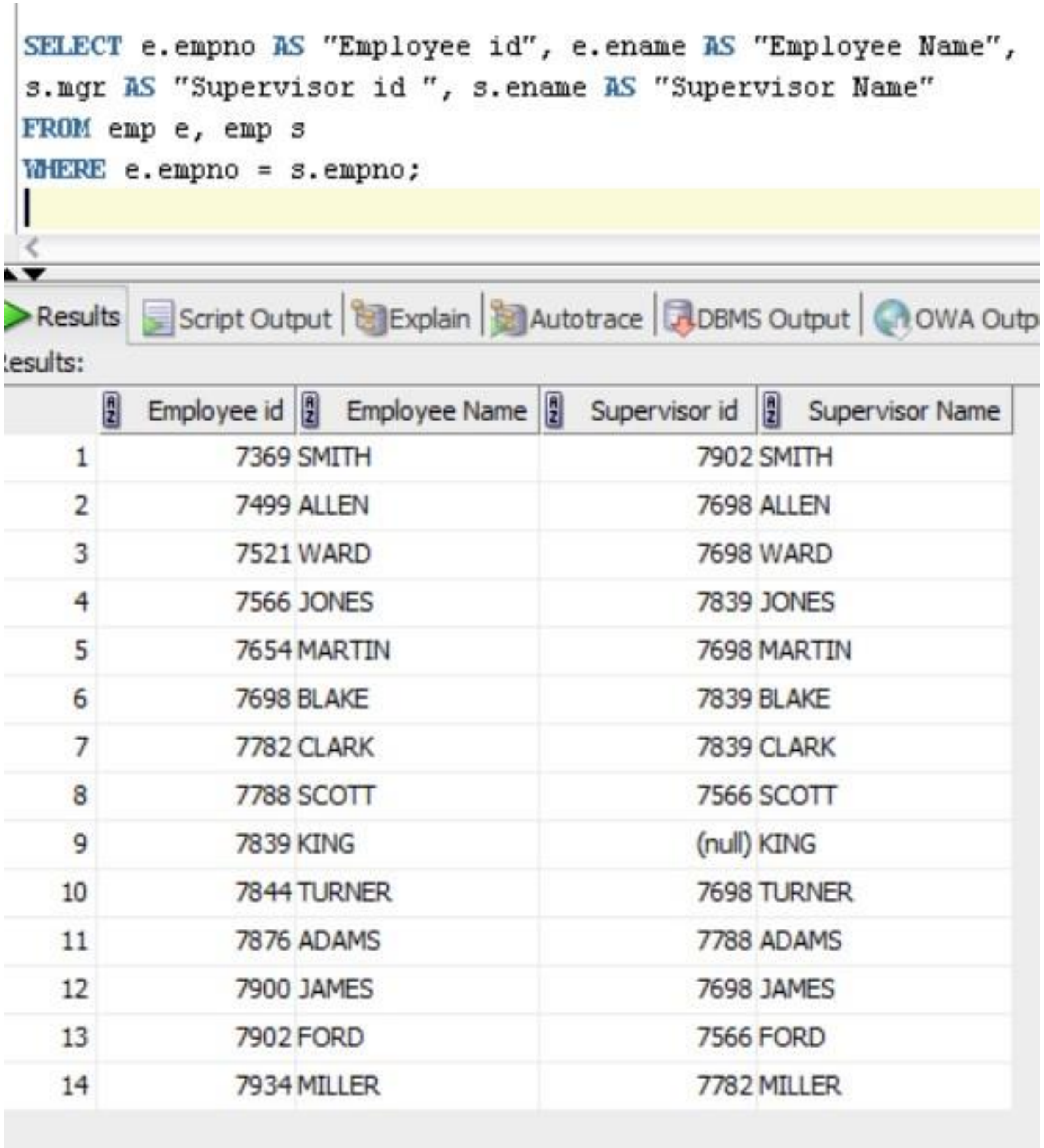
```
SELECT e.empno AS "Employee id", e.ename AS
"Employee Name", s.mgr AS "Supervisor id ", s.ename AS
"Supervisor Name"
FROM emp e, emp s
WHERE e.mgr = s.empno;
```



```
SELECT e.empno AS "Employee id", e.ename AS "Employee Name",
s.mgr AS "Supervisor id ", s.ename AS "Supervisor Name"
FROM emp e, emp s
WHERE e.mgr = s.empno;
```

Results:

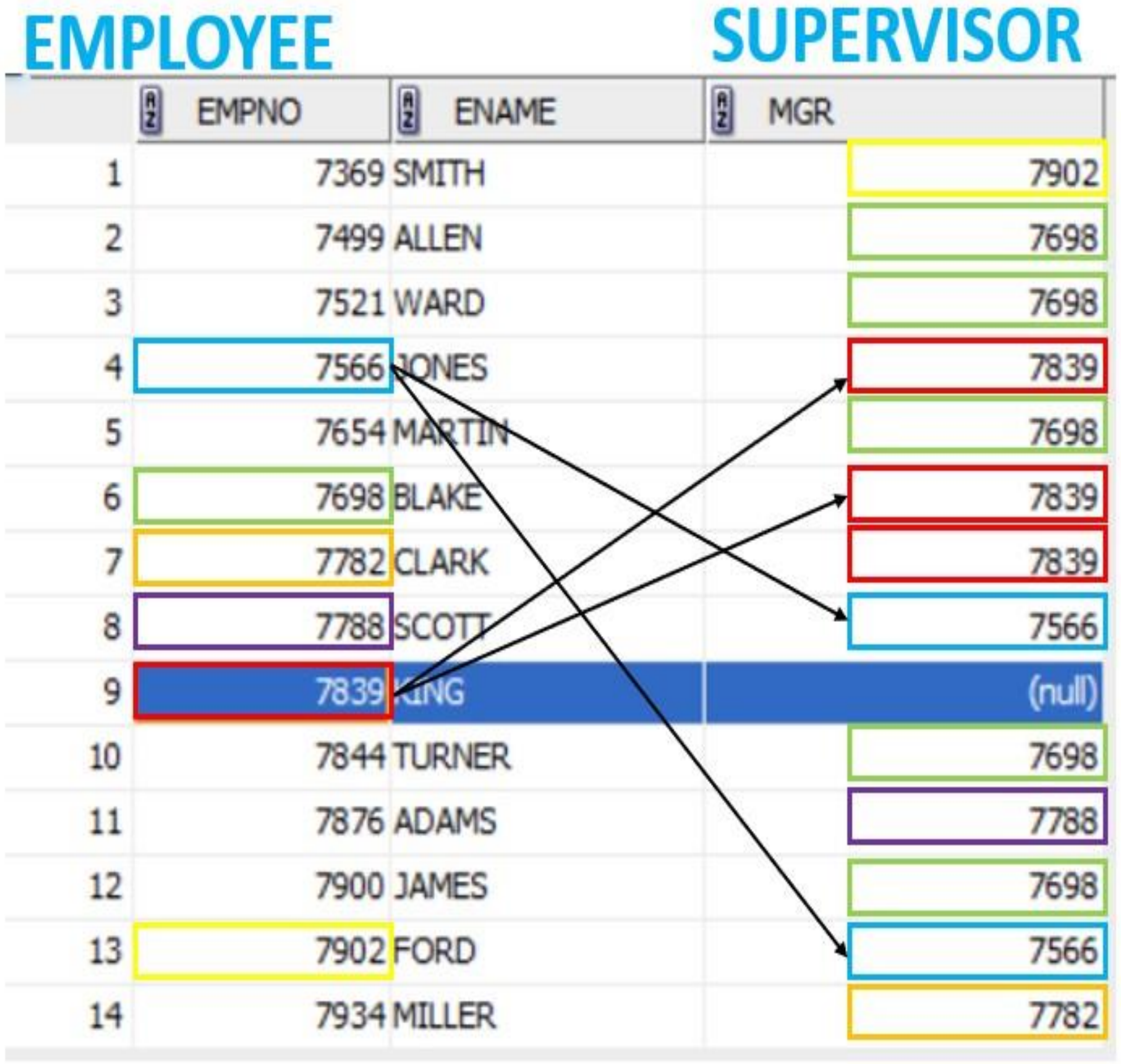
	Employee id	Employee Name	Supervisor id	Supervisor Name
1	7902	FORD	7839	JONES
2	7788	SCOTT	7839	JONES
3	7844	TURNER	7839	BLAKE
4	7499	ALLEN	7839	BLAKE
5	7521	WARD	7839	BLAKE
6	7900	JAMES	7839	BLAKE
7	7654	MARTIN	7839	BLAKE
8	7934	MILLER	7839	CLARK
9	7876	ADAMS	7566	SCOTT
10	7698	BLAKE	(null)	KING
11	7566	JONES	(null)	KING
12	7782	CLARK	(null)	KING
13	7369	SMITH	7566	FORD



```
SELECT e.empno AS "Employee id", e.ename AS "Employee Name",
s.mgr AS "Supervisor id ", s.ename AS "Supervisor Name"
FROM emp e, emp s
WHERE e.empno = s.empno;
```

esults:

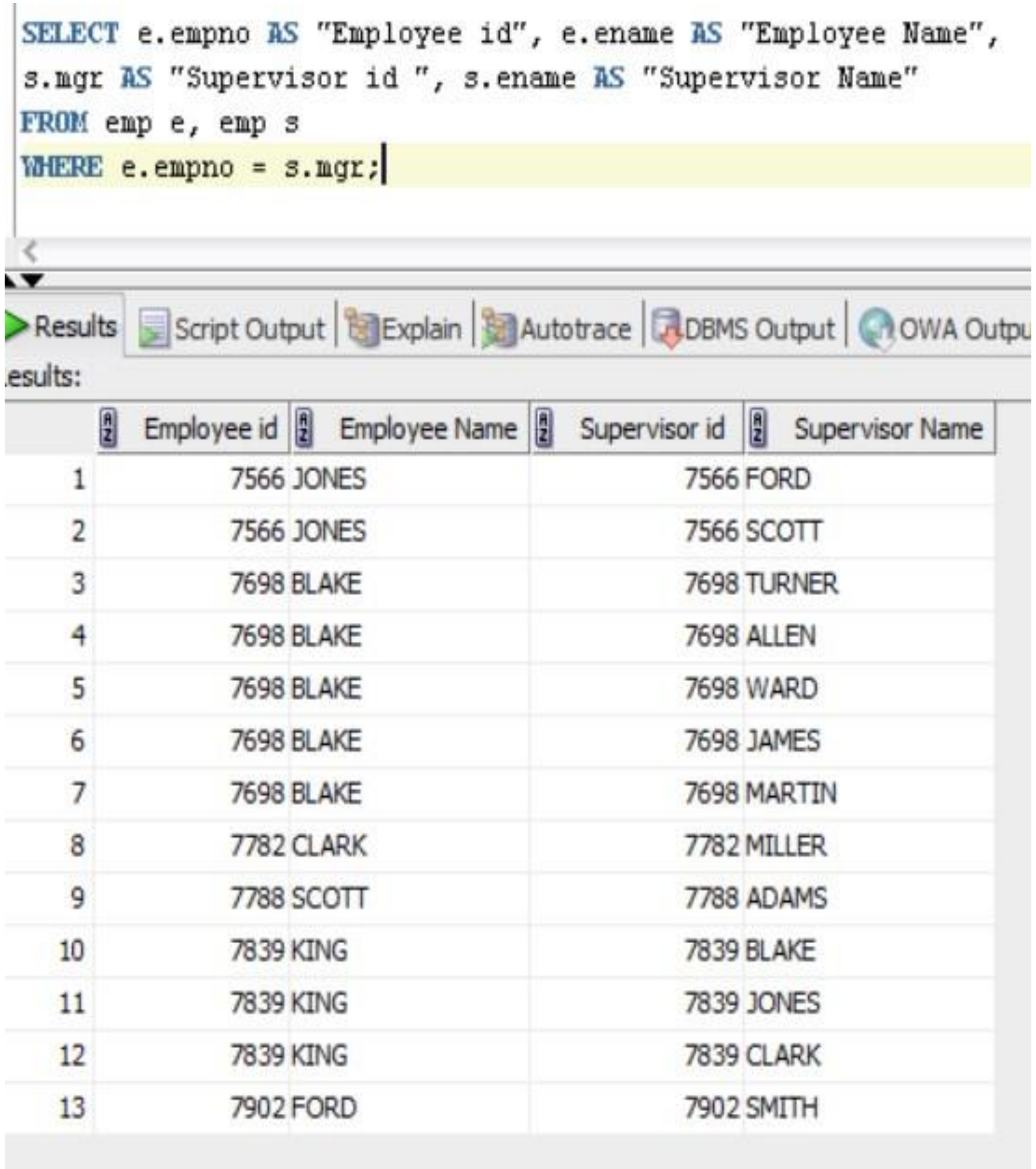
	Employee id	Employee Name	Supervisor id	Supervisor Name
1	7369	SMITH	7902	SMITH
2	7499	ALLEN	7698	ALLEN
3	7521	WARD	7698	WARD
4	7566	JONES	7839	JONES
5	7654	MARTIN	7698	MARTIN
6	7698	BLAKE	7839	BLAKE
7	7782	CLARK	7839	CLARK
8	7788	SCOTT	7566	SCOTT
9	7839	KING	(null)	KING
10	7844	TURNER	7698	TURNER
11	7876	ADAMS	7788	ADAMS
12	7900	JAMES	7698	JAMES
13	7902	FORD	7566	FORD
14	7934	MILLER	7782	MILLER



EMPLOYEE

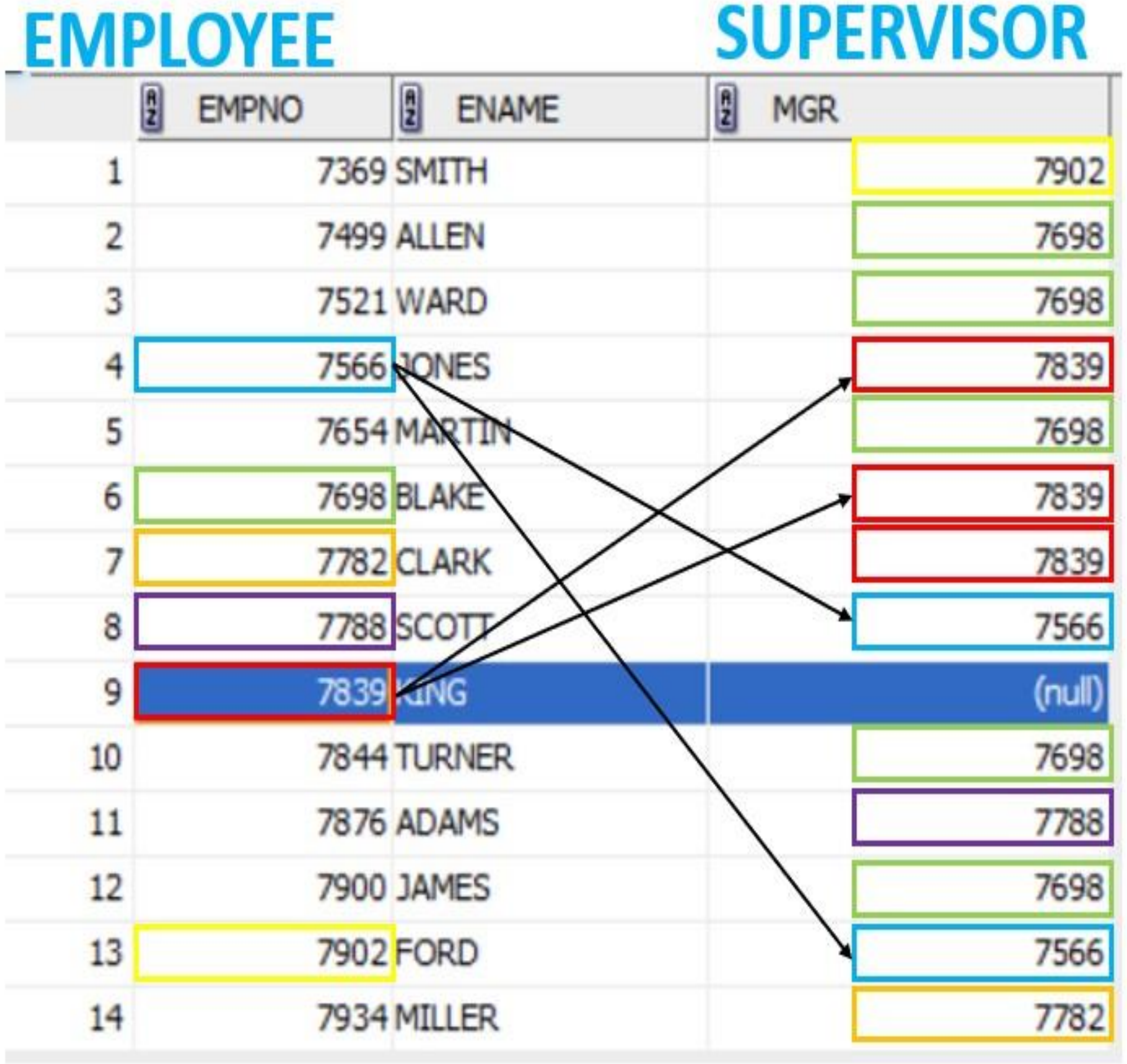
SUPERVISOR

	EMPNO	ENAME	MGR
1	7369	SMITH	7902
2	7499	ALLEN	7698
3	7521	WARD	7698
4	7566	JONES	7839
5	7654	MARTIN	7698
6	7698	BLAKE	7839
7	7782	CLARK	7839
8	7788	SCOTT	7566
9	7839	KING	(null)
10	7844	TURNER	7698
11	7876	ADAMS	7788
12	7900	JAMES	7698
13	7902	FORD	7566
14	7934	MILLER	7782



```
SELECT e.empno AS "Employee id", e.ename AS "Employee Name",
s.mgr AS "Supervisor id ", s.ename AS "Supervisor Name"
FROM emp e, emp s
WHERE e.empno = s.mgr;
```

	Employee id	Employee Name	Supervisor id	Supervisor Name
1	7566	JONES	7566	FORD
2	7566	JONES	7566	SCOTT
3	7698	BLAKE	7698	TURNER
4	7698	BLAKE	7698	ALLEN
5	7698	BLAKE	7698	WARD
6	7698	BLAKE	7698	JAMES
7	7698	BLAKE	7698	MARTIN
8	7782	CLARK	7782	MILLER
9	7788	SCOTT	7788	ADAMS
10	7839	KING	7839	BLAKE
11	7839	KING	7839	JONES
12	7839	KING	7839	CLARK
13	7902	FORD	7902	SMITH



	EMPNO	ENAME	MGR
1	7369	SMITH	7902
2	7499	ALLEN	7698
3	7521	WARD	7698
4	7566	JONES	7839
5	7654	MARTIN	7698
6	7698	BLAKE	7839
7	7782	CLARK	7839
8	7788	SCOTT	7566
9	7839	KING	(null)
10	7844	TURNER	7698
11	7876	ADAMS	7788
12	7900	JAMES	7698
13	7902	FORD	7566
14	7934	MILLER	7782

JOIN METHOD - ANSI Syntax (SQL/99 JOIN Approach)

SELF JOIN – JOIN METHOD

SYNTAX:

SELECT column_names

FROM table1 t1 JOIN table1 t2 ON

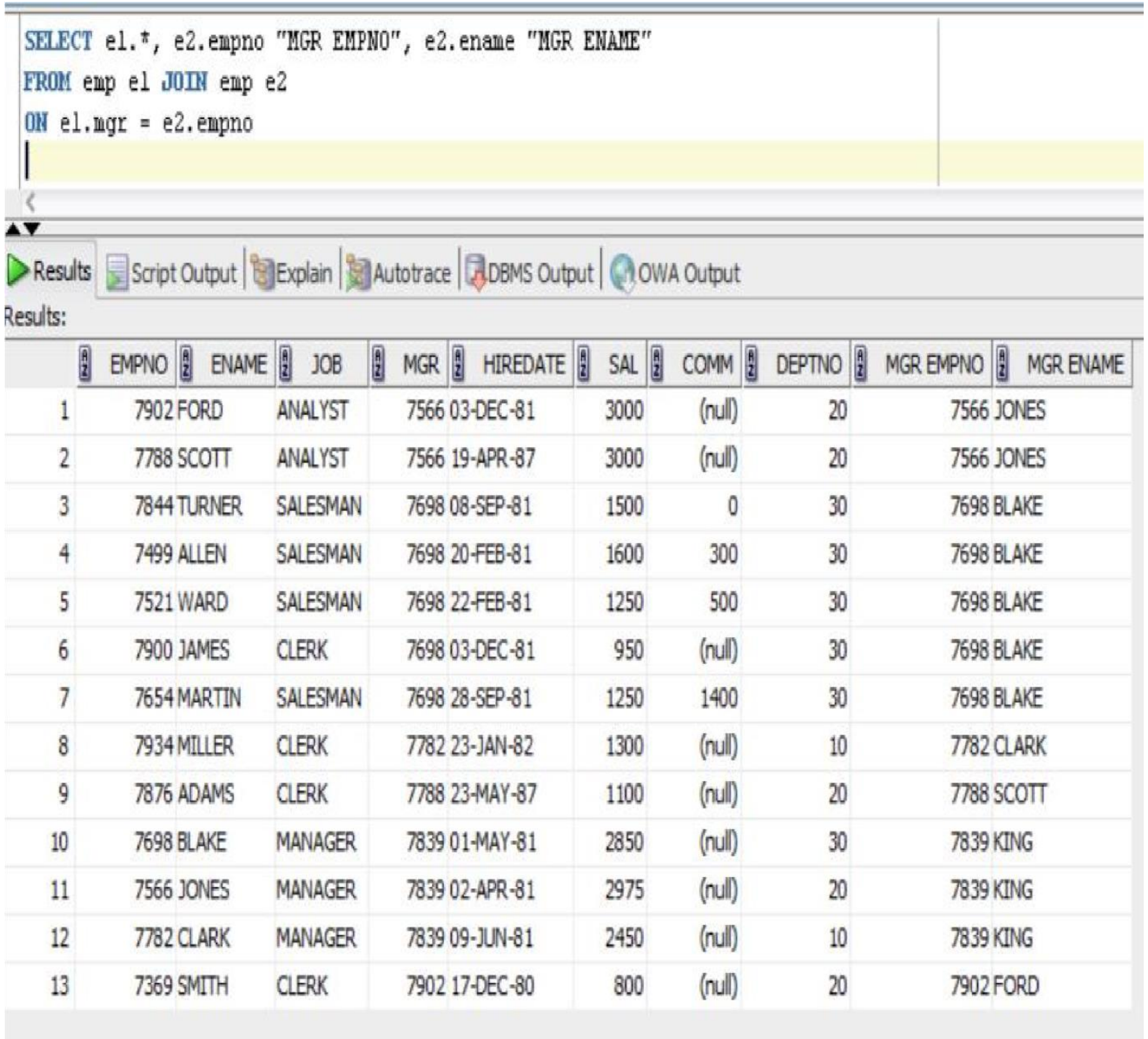
t1.columnname = t2.columnname

EXAMPLE:

SELECT e1.*, e2.empno "MGR EMPNO", e2.ename "MGR ENAME"

FROM emp e1 JOIN emp e2 ON

e1.mgr = e2.empno



```
SELECT e1.*, e2.empno "MGR EMPNO", e2.ename "MGR ENAME"
FROM emp e1 JOIN emp e2
ON e1.mgr = e2.empno
```

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	MGR EMPNO	MGR ENAME
1	7902	FORD	ANALYST	7566	03-DEC-81	3000	(null)	20	7566	JONES
2	7788	SCOTT	ANALYST	7566	19-APR-87	3000	(null)	20	7566	JONES
3	7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30	7698	BLAKE
4	7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30	7698	BLAKE
5	7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30	7698	BLAKE
6	7900	JAMES	CLERK	7698	03-DEC-81	950	(null)	30	7698	BLAKE
7	7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30	7698	BLAKE
8	7934	MILLER	CLERK	7782	23-JAN-82	1300	(null)	10	7782	CLARK
9	7876	ADAMS	CLERK	7788	23-MAY-87	1100	(null)	20	7788	SCOTT
10	7698	BLAKE	MANAGER	7839	01-MAY-81	2850	(null)	30	7839	KING
11	7566	JONES	MANAGER	7839	02-APR-81	2975	(null)	20	7839	KING
12	7782	CLARK	MANAGER	7839	09-JUN-81	2450	(null)	10	7839	KING
13	7369	SMITH	CLERK	7902	17-DEC-80	800	(null)	20	7902	FORD

JOIN METHOD - ANSI Syntax (SQL/99 JOIN Approach)

SELF JOIN – JOIN METHOD

SYNTAX:

SELECT column_names

FROM table1 t1 JOIN table1 t2

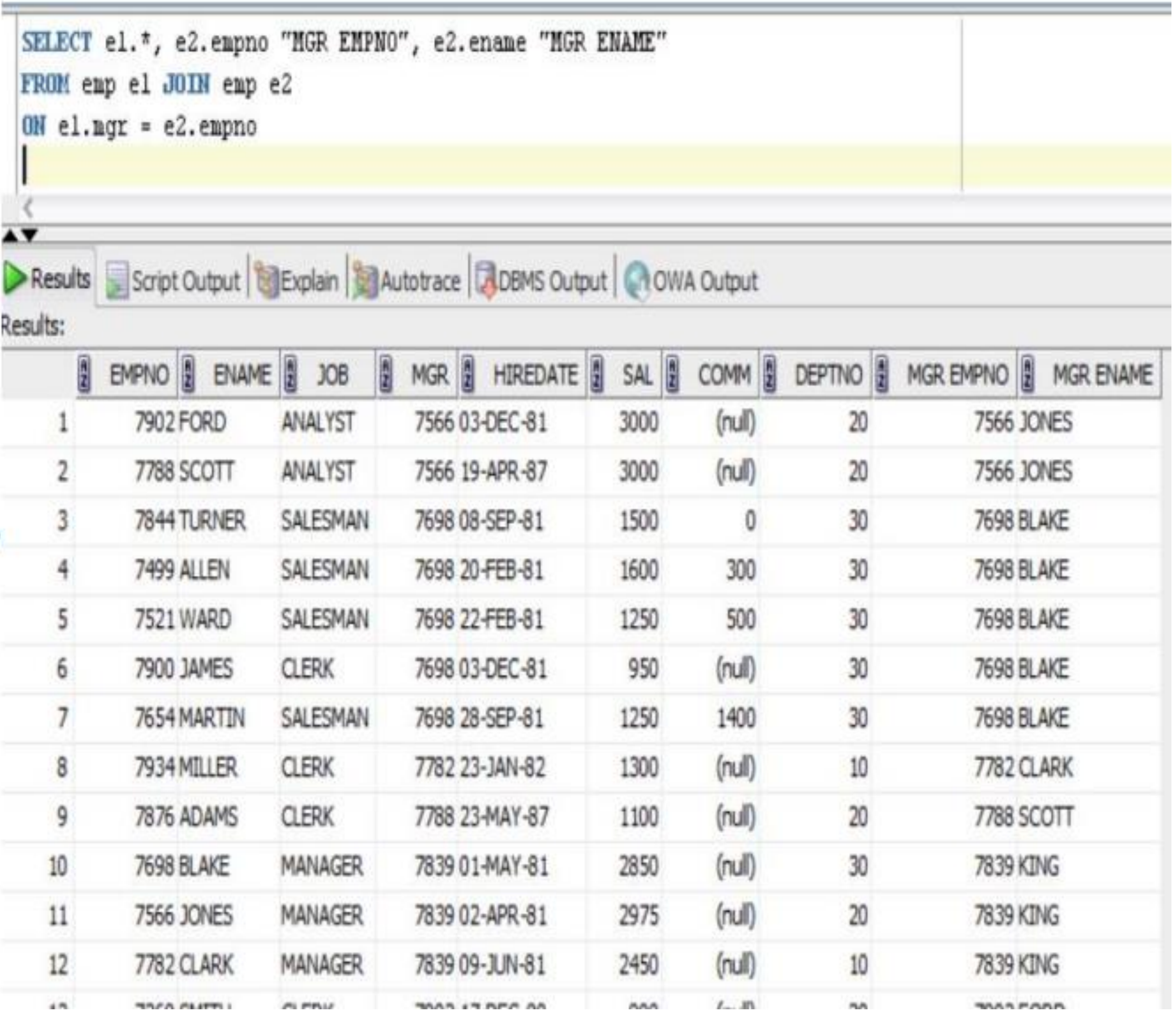
ON t1.columnname = t2.columnname

EXAMPLE:

SELECT e1.*, e2.empno "MGR EMPNO", e2.ename "MGR ENAME"

FROM emp e1 JOIN emp e2

ON e1.mgr = e2.empno



```
SELECT e1.*, e2.empno "MGR EMPNO", e2.ename "MGR ENAME"
FROM emp e1 JOIN emp e2
ON e1.mgr = e2.empno
```

Results:

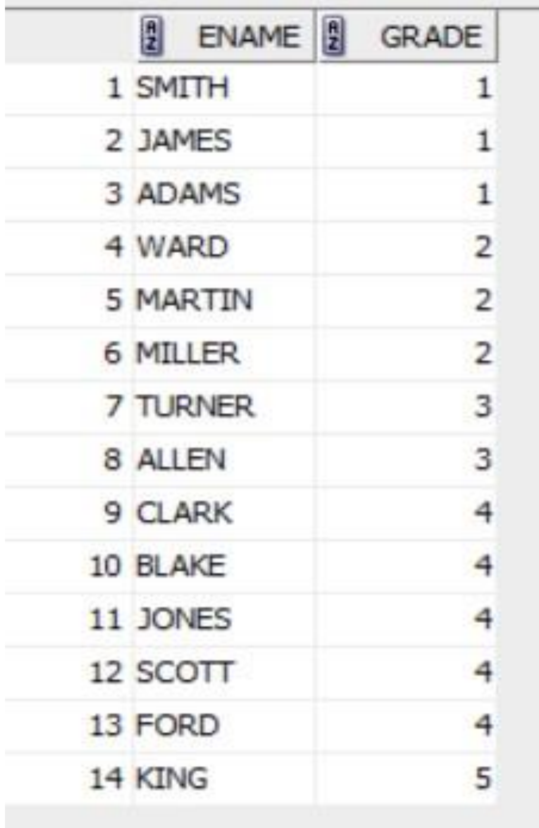
	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	MGR EMPNO	MGR ENAME
1	7902	FORD	ANALYST	7566	03-DEC-81	3000	(null)	20	7566	JONES
2	7788	SCOTT	ANALYST	7566	19-APR-87	3000	(null)	20	7566	JONES
3	7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30	7698	BLAKE
4	7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30	7698	BLAKE
5	7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30	7698	BLAKE
6	7900	JAMES	CLERK	7698	03-DEC-81	950	(null)	30	7698	BLAKE
7	7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30	7698	BLAKE
8	7934	MILLER	CLERK	7782	23-JAN-82	1300	(null)	10	7782	CLARK
9	7876	ADAMS	CLERK	7788	23-MAY-87	1100	(null)	20	7788	SCOTT
10	7698	BLAKE	MANAGER	7839	01-MAY-81	2850	(null)	30	7839	KING
11	7566	JONES	MANAGER	7839	02-APR-81	2975	(null)	20	7839	KING
12	7782	CLARK	MANAGER	7839	09-JUN-81	2450	(null)	10	7839	KING
13	7369	SMITH	CLERK	7902	17-DEC-80	800	(null)	20	7902	FORD

SYNTAX:

```
SELECT *
FROM table_name1, table_name2
WHERE table_name1.column [ > | < | >= | <= ] table_name2.column ;
```

EXAMPLE:

```
Select ename, grade
From emp, salgrade
Where emp.sal >= salgrade.losal and emp.sal <= salgrade.hisal ;

Select ename, grade
From emp, salgrade
Where sal BETWEEN losal AND hisal ;
```



	ENAME	GRADE
1	SMITH	1
2	JAMES	1
3	ADAMS	1
4	WARD	2
5	MARTIN	2
6	MILLER	2
7	TURNER	3
8	ALLEN	3
9	CLARK	4
10	BLAKE	4
11	JONES	4
12	SCOTT	4
13	FORD	4
14	KING	5

NONEQUI JOIN-JOIN METHOD

SYNTAX:

SELECT *

FROM table_name1 JOIN table_name2

ON table_name1.column [> | < | >= | <=]

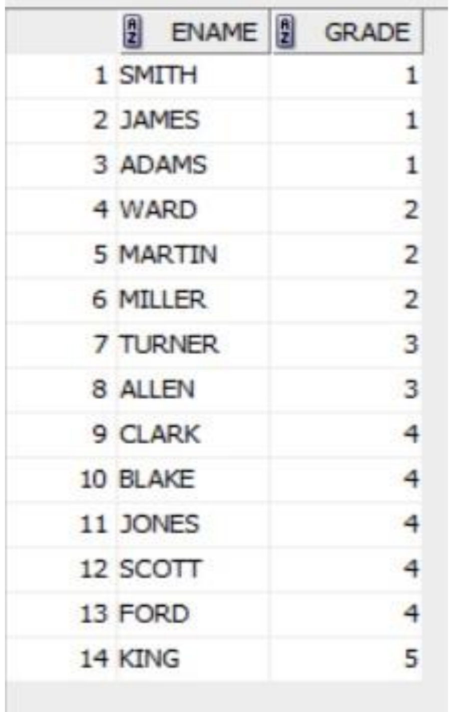
table_name2.column ; **EXAMPLE:**

SELECT ename, grade

FROM emp JOIN salgrade

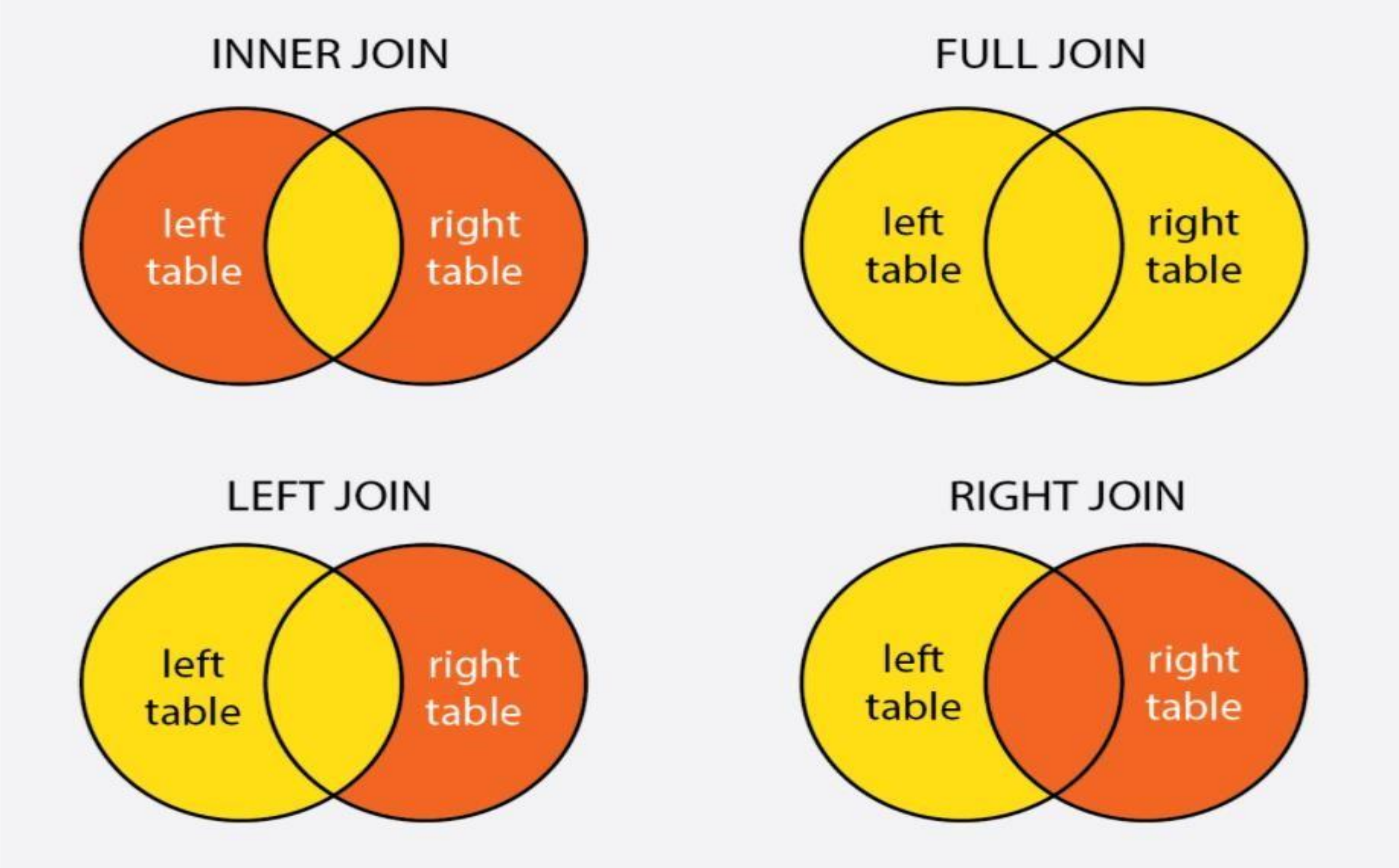
ON sal BETWEEN losal AND hisal ;

USING and NATURAL JOIN can not be used to form a non-equi join, as both of these do not involve specification of join conditions.



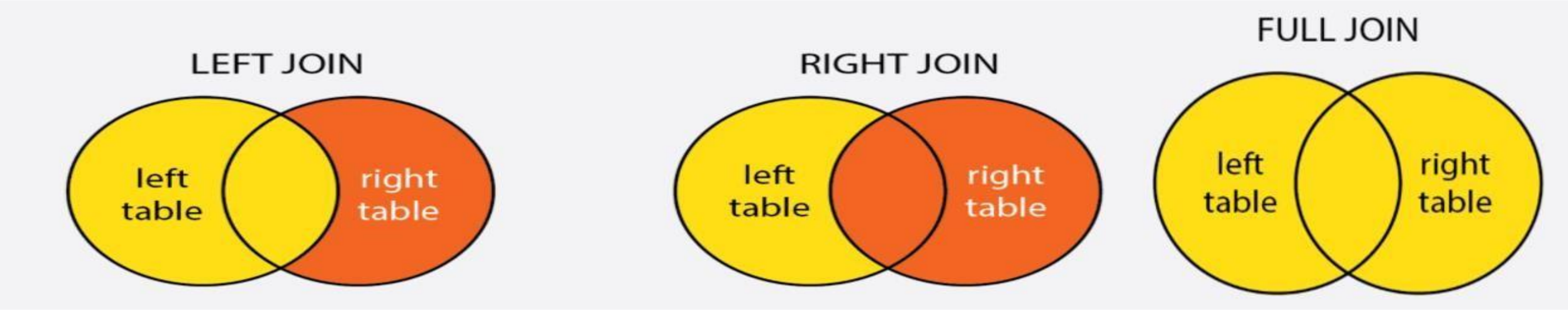
	ENAME	GRADE
1	SMITH	1
2	JAMES	1
3	ADAMS	1
4	WARD	2
5	MARTIN	2
6	MILLER	2
7	TURNER	3
8	ALLEN	3
9	CLARK	4
10	BLAKE	4
11	JONES	4
12	SCOTT	4
13	FORD	4
14	KING	5

INNER JOIN versus OUTER JOIN



OUTER JOIN

- An OUTER JOIN is used to return all rows that exist in one table, even though corresponding rows do not exist in the joined table.
- When performing an inner join, rows from either table that are unmatched in the other table are not returned. However, in an outer join, unmatched rows in one or both tables can be returned. There are three types of outer joins:
  1. LEFT JOIN returns only unmatched rows from the left table.
  2. RIGHT JOIN returns only unmatched rows from the right table.
  3. FULL OUTER JOIN returns unmatched rows from both tables.



OUTER JOIN-TRADITIONAL METHOD

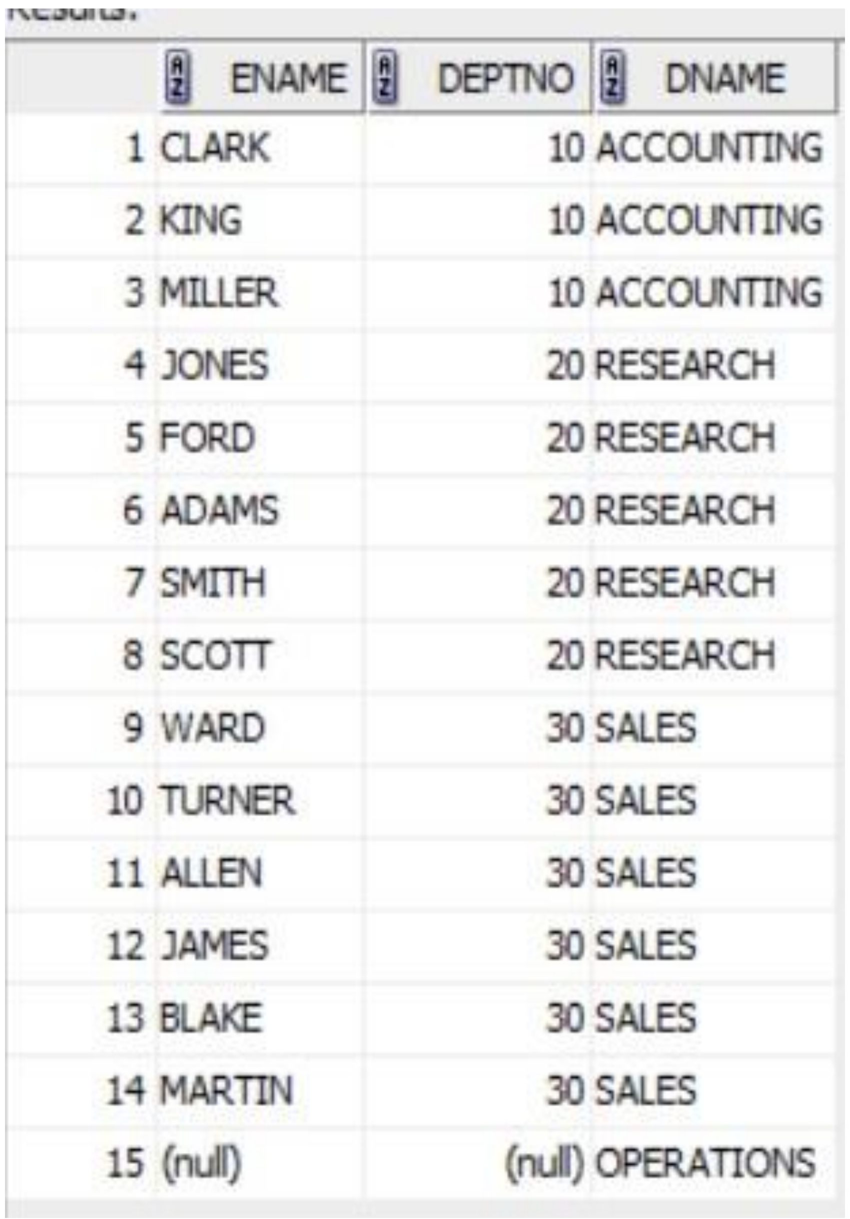
- The OUTER JOIN returns all rows from both the participating tables which satisfy the join condition along with rows which do not satisfy the join condition.
- The (+) symbol is used to denote an OUTER JOIN in a query.
- The (+) is placed at the end of the table name in the WHERE clause.
- The table with the (+) should be the table that does not have matching rows(i.e., the table having deficit information).
- The (+) symbol can only be used with one of the tables in the join condition.
- The output of the join depends on the placement of the (+) symbol.
- If the (+) symbol is placed with the table on the right-hand side of the join condition, then the join is called RIGHT OUTER JOIN.
- If the (+) symbol is placed with the table on the left-hand side of the join condition, then the join is called LEFT OUTER JOIN.

SYNTAX:

```
SELECT select_list
FROM table1 , table2 WHERE conditions (+) ;
```

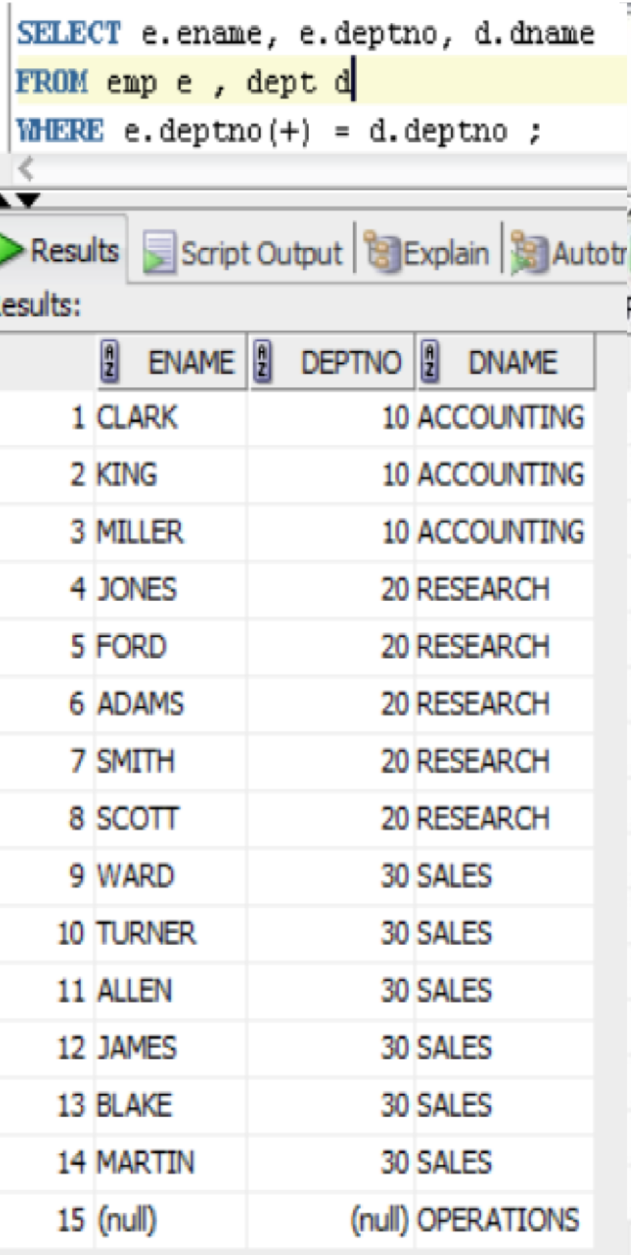
EXAMPLE:

```
SELECT e.ename, e.deptno, d.dname
FROM emp e , dept d
WHERE e.deptno(+) = d.deptno ;
```



Results:

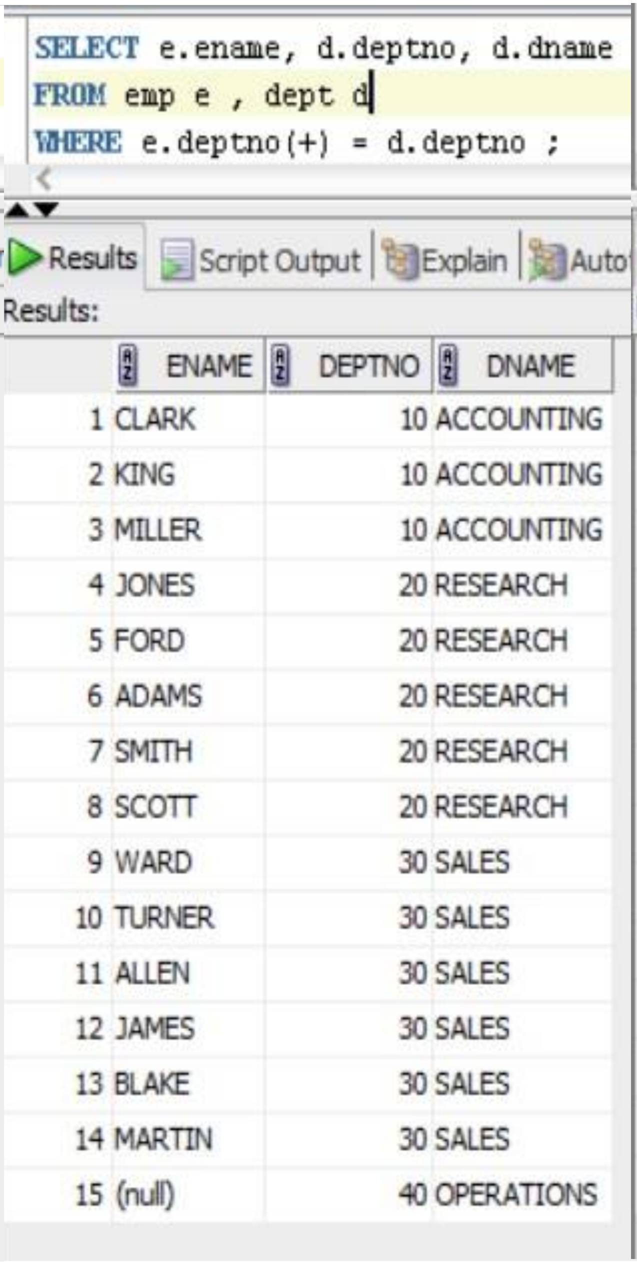
	ENAME	DEPTNO	DNAME
1	CLARK	10	ACCOUNTING
2	KING	10	ACCOUNTING
3	MILLER	10	ACCOUNTING
4	JONES	20	RESEARCH
5	FORD	20	RESEARCH
6	ADAMS	20	RESEARCH
7	SMITH	20	RESEARCH
8	SCOTT	20	RESEARCH
9	WARD	30	SALES
10	TURNER	30	SALES
11	ALLEN	30	SALES
12	JAMES	30	SALES
13	BLAKE	30	SALES
14	MARTIN	30	SALES
15	(null)	(null)	OPERATIONS



```sql
SELECT e.ename, e.deptno, d.dname
FROM emp e , dept d
WHERE e.deptno(+) = d.deptno ;
```

esults:

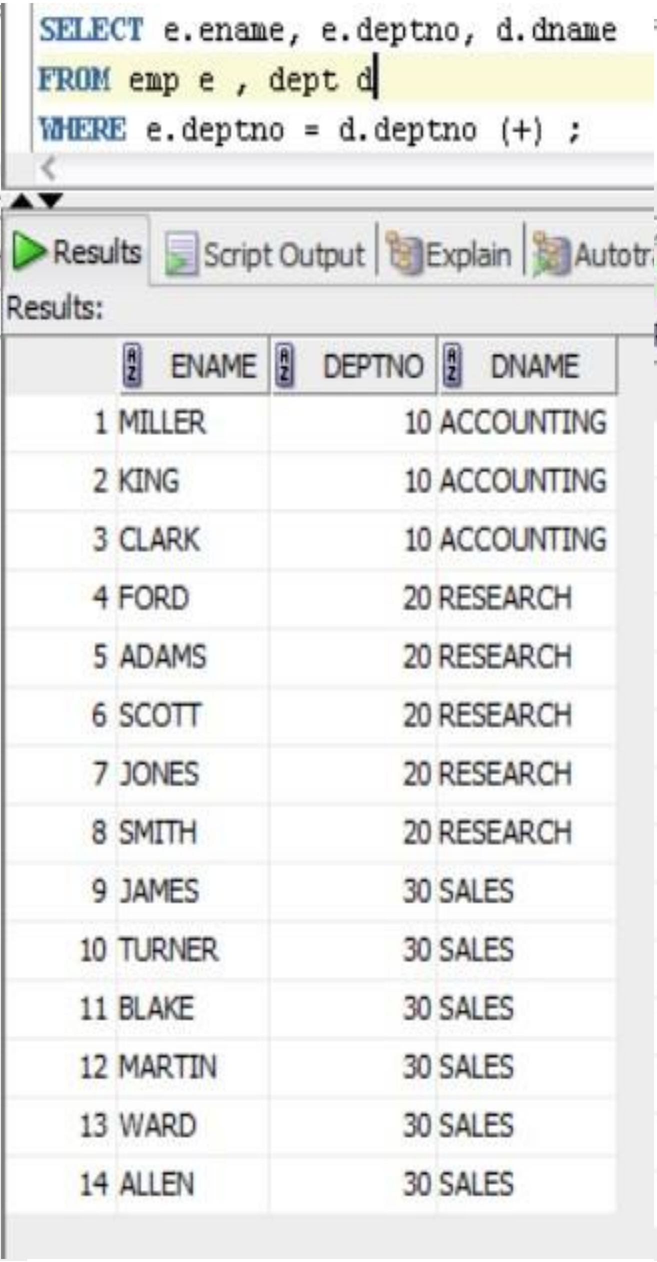
	ENAME	DEPTNO	DNAME
1	CLARK	10	ACCOUNTING
2	KING	10	ACCOUNTING
3	MILLER	10	ACCOUNTING
4	JONES	20	RESEARCH
5	FORD	20	RESEARCH
6	ADAMS	20	RESEARCH
7	SMITH	20	RESEARCH
8	SCOTT	20	RESEARCH
9	WARD	30	SALES
10	TURNER	30	SALES
11	ALLEN	30	SALES
12	JAMES	30	SALES
13	BLAKE	30	SALES
14	MARTIN	30	SALES
15	(null)	(null)	OPERATIONS



```sql
SELECT e.ename, d.deptno, d.dname
FROM emp e , dept d
WHERE e.deptno(+) = d.deptno ;
```

Results:

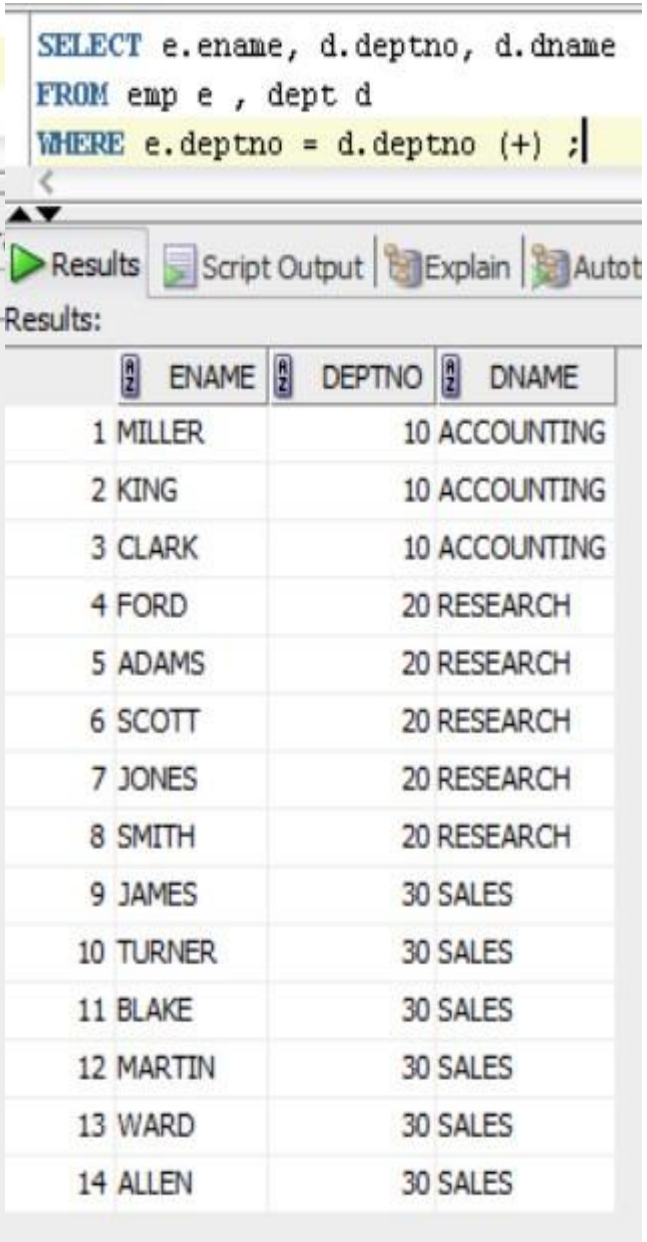
	ENAME	DEPTNO	DNAME
1	CLARK	10	ACCOUNTING
2	KING	10	ACCOUNTING
3	MILLER	10	ACCOUNTING
4	JONES	20	RESEARCH
5	FORD	20	RESEARCH
6	ADAMS	20	RESEARCH
7	SMITH	20	RESEARCH
8	SCOTT	20	RESEARCH
9	WARD	30	SALES
10	TURNER	30	SALES
11	ALLEN	30	SALES
12	JAMES	30	SALES
13	BLAKE	30	SALES
14	MARTIN	30	SALES
15	(null)	40	OPERATIONS



```sql
SELECT e.ename, e.deptno, d.dname
FROM emp e , dept d
WHERE e.deptno = d.deptno (+) ;
```

Results:

	ENAME	DEPTNO	DNAME
1	MILLER	10	ACCOUNTING
2	KING	10	ACCOUNTING
3	CLARK	10	ACCOUNTING
4	FORD	20	RESEARCH
5	ADAMS	20	RESEARCH
6	SCOTT	20	RESEARCH
7	JONES	20	RESEARCH
8	SMITH	20	RESEARCH
9	JAMES	30	SALES
10	TURNER	30	SALES
11	BLAKE	30	SALES
12	MARTIN	30	SALES
13	WARD	30	SALES
14	ALLEN	30	SALES



```sql
SELECT e.ename, d.deptno, d.dname
FROM emp e , dept d
WHERE e.deptno = d.deptno (+) ;
```

Results:

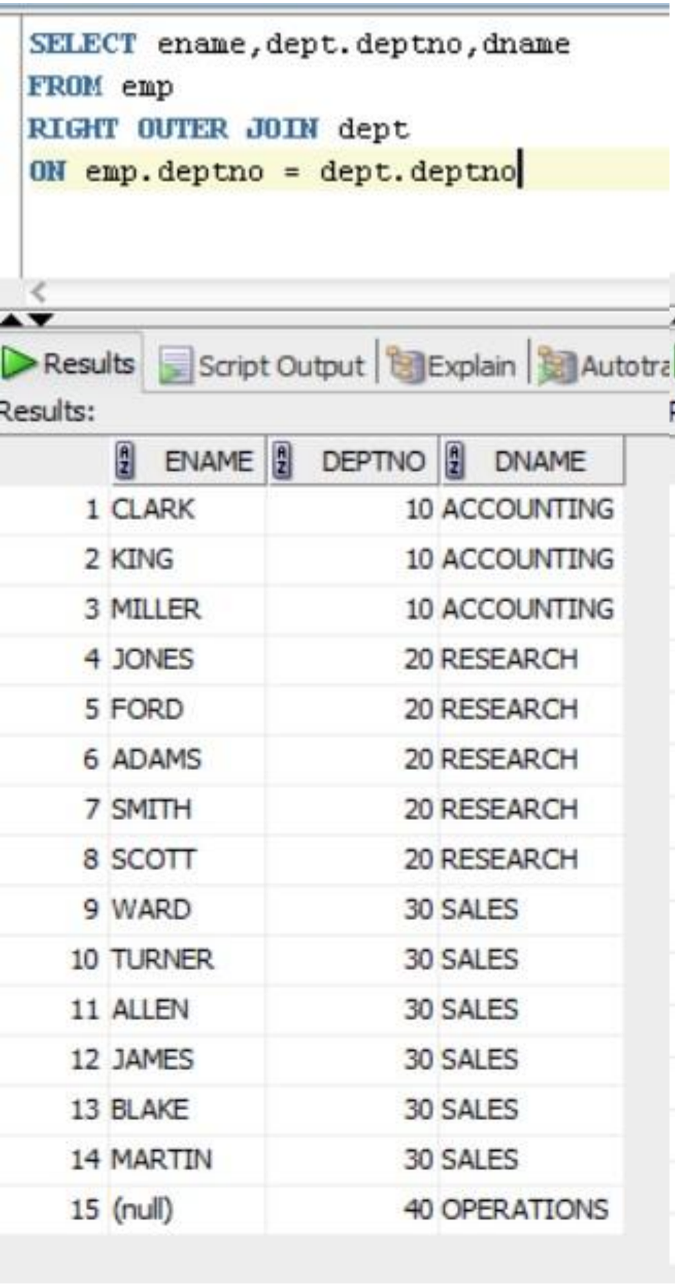
	ENAME	DEPTNO	DNAME
1	MILLER	10	ACCOUNTING
2	KING	10	ACCOUNTING
3	CLARK	10	ACCOUNTING
4	FORD	20	RESEARCH
5	ADAMS	20	RESEARCH
6	SCOTT	20	RESEARCH
7	JONES	20	RESEARCH
8	SMITH	20	RESEARCH
9	JAMES	30	SALES
10	TURNER	30	SALES
11	BLAKE	30	SALES
12	MARTIN	30	SALES
13	WARD	30	SALES
14	ALLEN	30	SALES

JOIN METHOD - ANSI Syntax (SQL/99 JOIN Approach)

OUTER JOIN-JOIN METHOD

SYNTAX:

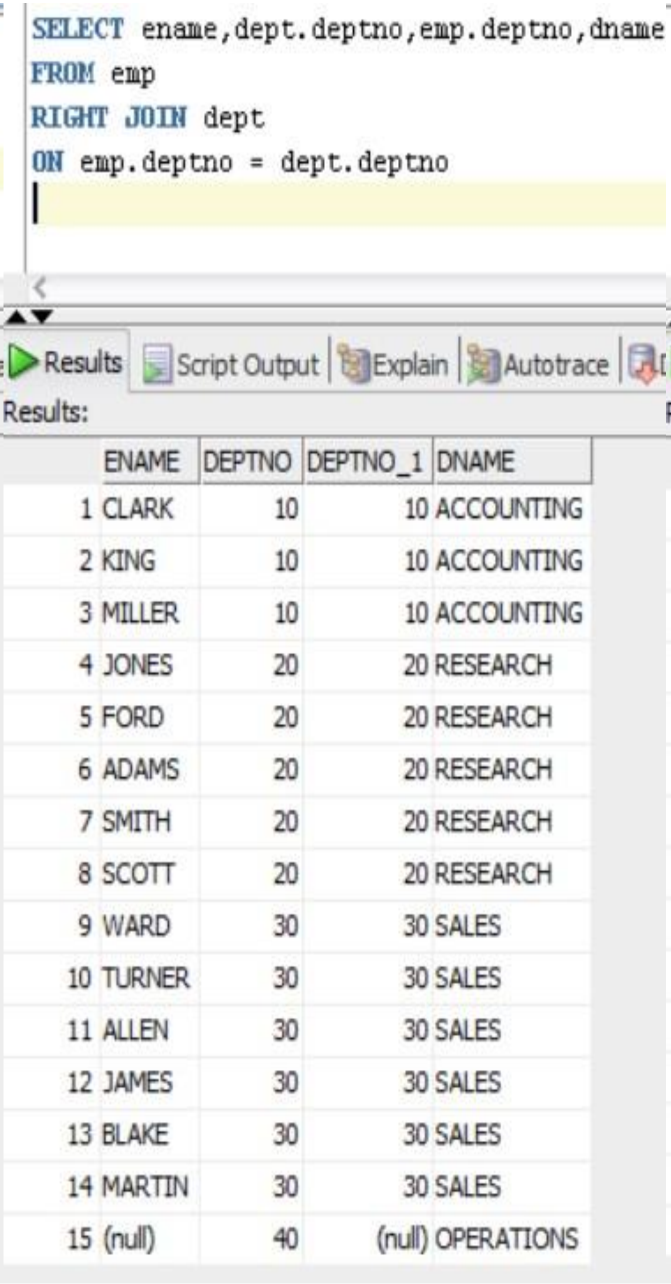
```
SELECT column_names
FROM table1
FULL OUTER JOIN | RIGHT OUTER JOIN |LEFT OUTER JOIN table2
ON table1.column_name = table2.column_name
[WHERE condition] ;
```



```sql
SELECT ename,dept.deptno,dname
FROM emp
RIGHT OUTER JOIN dept
ON emp.deptno = dept.deptno
```

Results:

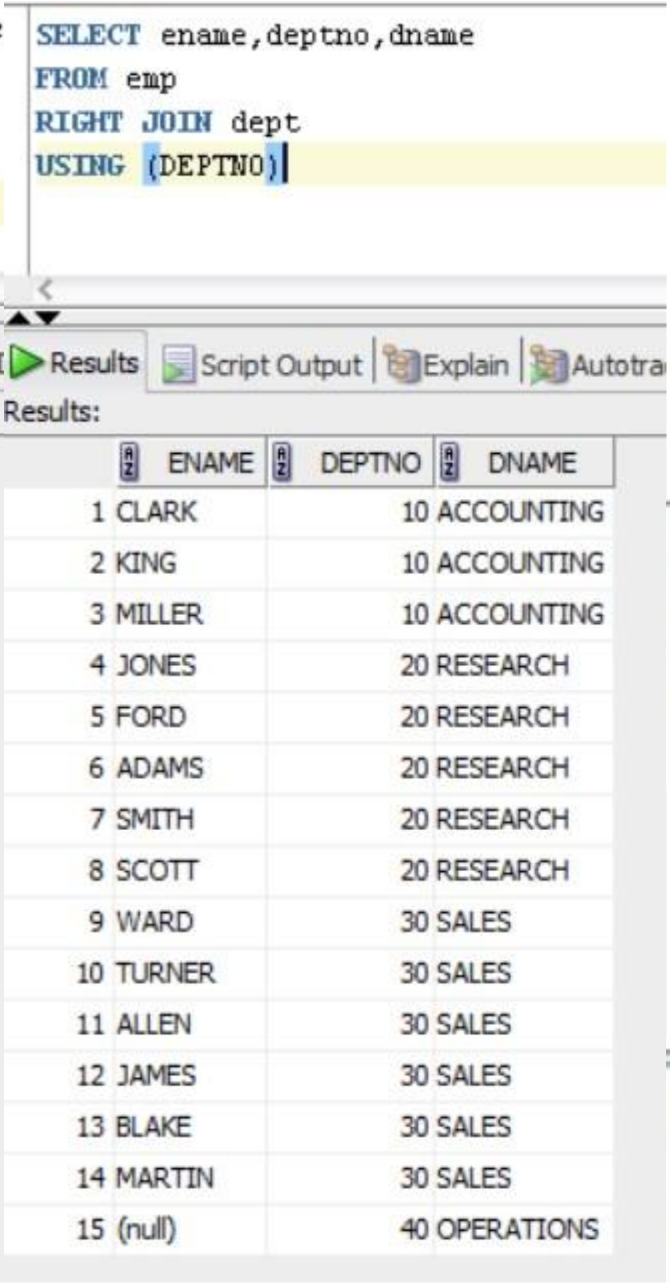
	ENAME	DEPTNO	DNAME
1	CLARK	10	ACCOUNTING
2	KING	10	ACCOUNTING
3	MILLER	10	ACCOUNTING
4	JONES	20	RESEARCH
5	FORD	20	RESEARCH
6	ADAMS	20	RESEARCH
7	SMITH	20	RESEARCH
8	SCOTT	20	RESEARCH
9	WARD	30	SALES
10	TURNER	30	SALES
11	ALLEN	30	SALES
12	JAMES	30	SALES
13	BLAKE	30	SALES
14	MARTIN	30	SALES
15	(null)	40	OPERATIONS



```sql
SELECT ename,dept.deptno,emp.deptno,dname
FROM emp
RIGHT JOIN dept
ON emp.deptno = dept.deptno
```

Results:

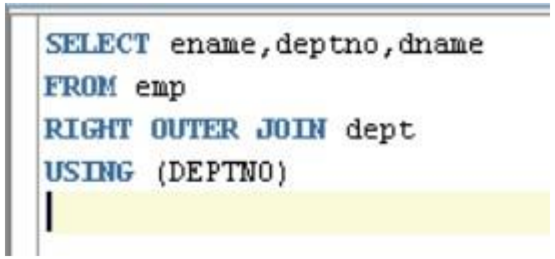
	ENAME	DEPTNO	DEPTNO_1	DNAME
1	CLARK	10	10	ACCOUNTING
2	KING	10	10	ACCOUNTING
3	MILLER	10	10	ACCOUNTING
4	JONES	20	20	RESEARCH
5	FORD	20	20	RESEARCH
6	ADAMS	20	20	RESEARCH
7	SMITH	20	20	RESEARCH
8	SCOTT	20	20	RESEARCH
9	WARD	30	30	SALES
10	TURNER	30	30	SALES
11	ALLEN	30	30	SALES
12	JAMES	30	30	SALES
13	BLAKE	30	30	SALES
14	MARTIN	30	30	SALES
15	(null)	40	(null)	OPERATIONS



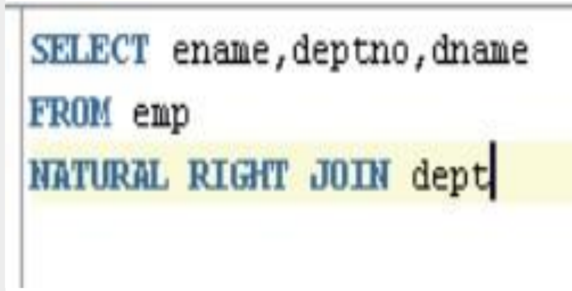
```sql
SELECT ename,deptno,dname
FROM emp
RIGHT JOIN dept
USING (DEPTNO)
```

Results:

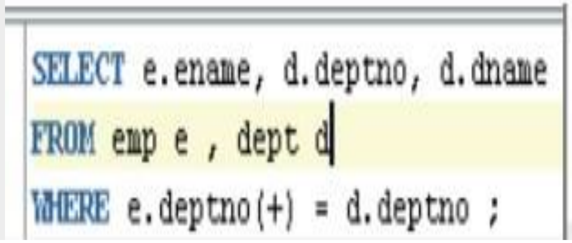
	ENAME	DEPTNO	DNAME
1	CLARK	10	ACCOUNTING
2	KING	10	ACCOUNTING
3	MILLER	10	ACCOUNTING
4	JONES	20	RESEARCH
5	FORD	20	RESEARCH
6	ADAMS	20	RESEARCH
7	SMITH	20	RESEARCH
8	SCOTT	20	RESEARCH
9	WARD	30	SALES
10	TURNER	30	SALES
11	ALLEN	30	SALES
12	JAMES	30	SALES
13	BLAKE	30	SALES
14	MARTIN	30	SALES
15	(null)	40	OPERATIONS



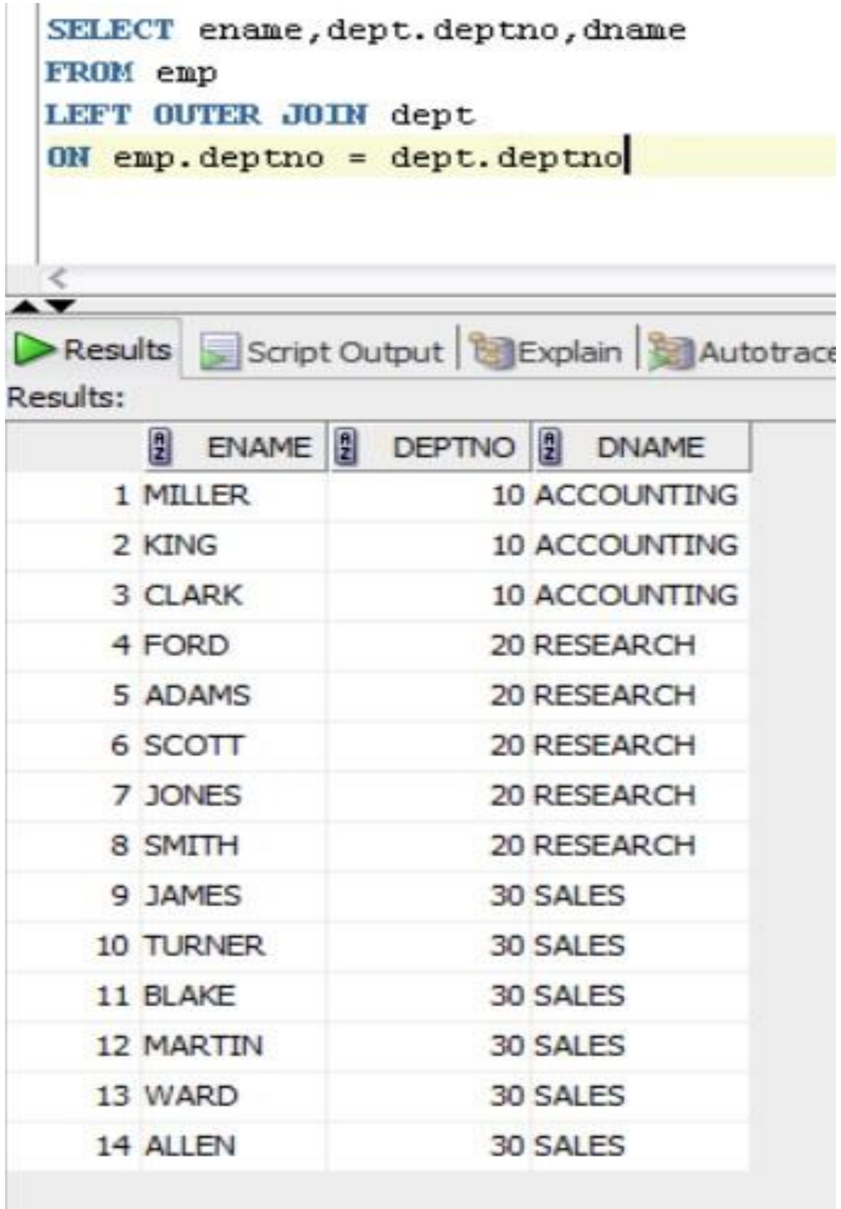
```sql
SELECT ename,deptno,dname
FROM emp
RIGHT OUTER JOIN dept
USING (DEPTNO)
```



```sql
SELECT ename,deptno,dname
FROM emp
NATURAL RIGHT JOIN dept
```



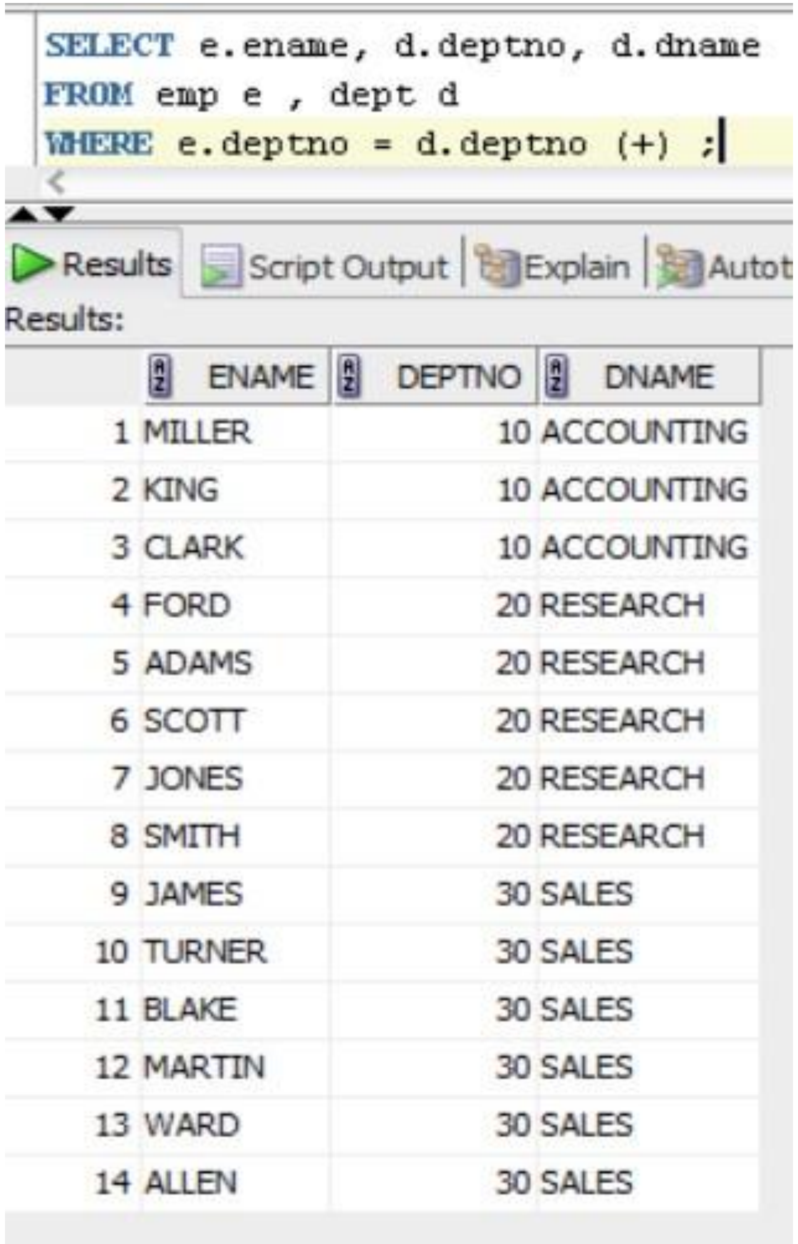
```sql
SELECT e.ename, d.deptno, d.dname
FROM emp e , dept d
WHERE e.deptno(+) = d.deptno ;
```



```sql
SELECT ename,dept.deptno,dname
FROM emp
LEFT OUTER JOIN dept
ON emp.deptno = dept.deptno
```

Results:

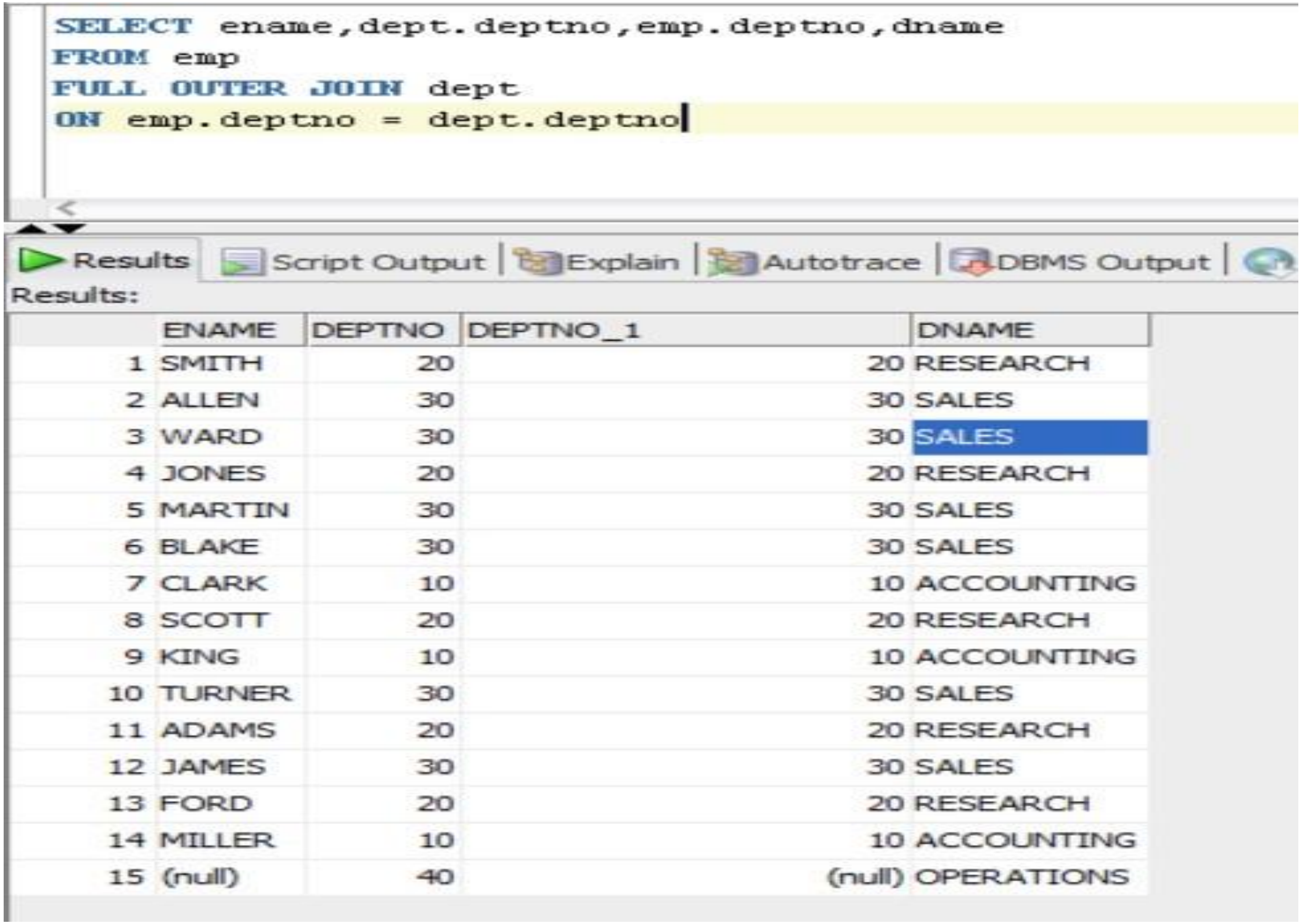
	ENAME	DEPTNO	DNAME
1	MILLER	10	ACCOUNTING
2	KING	10	ACCOUNTING
3	CLARK	10	ACCOUNTING
4	FORD	20	RESEARCH
5	ADAMS	20	RESEARCH
6	SCOTT	20	RESEARCH
7	JONES	20	RESEARCH
8	SMITH	20	RESEARCH
9	JAMES	30	SALES
10	TURNER	30	SALES
11	BLAKE	30	SALES
12	MARTIN	30	SALES
13	WARD	30	SALES
14	ALLEN	30	SALES



```sql
SELECT e.ename, d.deptno, d.dname
FROM emp e , dept d
WHERE e.deptno = d.deptno (+) ;
```

Results:

	ENAME	DEPTNO	DNAME
1	MILLER	10	ACCOUNTING
2	KING	10	ACCOUNTING
3	CLARK	10	ACCOUNTING
4	FORD	20	RESEARCH
5	ADAMS	20	RESEARCH
6	SCOTT	20	RESEARCH
7	JONES	20	RESEARCH
8	SMITH	20	RESEARCH
9	JAMES	30	SALES
10	TURNER	30	SALES
11	BLAKE	30	SALES
12	MARTIN	30	SALES
13	WARD	30	SALES
14	ALLEN	30	SALES



```sql
SELECT ename,dept.deptno,emp.deptno,dname
FROM emp
FULL OUTER JOIN dept
ON emp.deptno = dept.deptno
```

Results:

	ENAME	DEPTNO	DEPTNO_1	DNAME
1	SMITH	20	20	RESEARCH
2	ALLEN	30	30	SALES
3	WARD	30	30	SALES
4	JONES	20	20	RESEARCH
5	MARTIN	30	30	SALES
6	BLAKE	30	30	SALES
7	CLARK	10	10	ACCOUNTING
8	SCOTT	20	20	RESEARCH
9	KING	10	10	ACCOUNTING
10	TURNER	30	30	SALES
11	ADAMS	20	20	RESEARCH
12	JAMES	30	30	SALES
13	FORD	20	20	RESEARCH
14	MILLER	10	10	ACCOUNTING
15	(null)	40	(null)	OPERATIONS