Department of Software Engineering
Mehran University of Engineering and Technology, Jamshoro

Course: SW215 – Database System			
Instructor	Ms Shafiya Qadeer	**Practical/Lab No.**	07
Date:	22-02-2021	**CLOs**	2
Signature		**Assessment Score**	2 Marks

Topic	**To become familiar with Joins**
Objectives	- To become familiar with Joining Multiple Tables

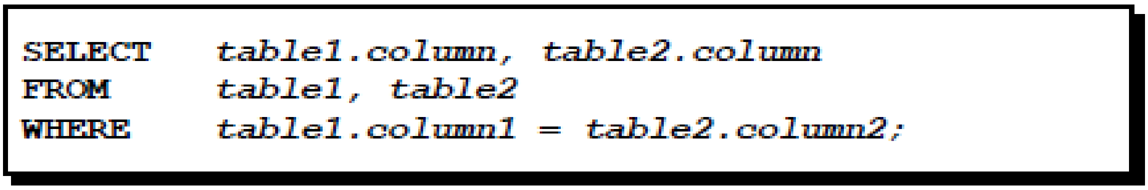
Lab Discussion: Theoretical concepts and Procedural steps

JOINS

- A JOIN clause is used to combine rows from two or more tables.
- It is used for combining column from two or more tables by using values common to both tables.

Use a Join to query data from more than one table

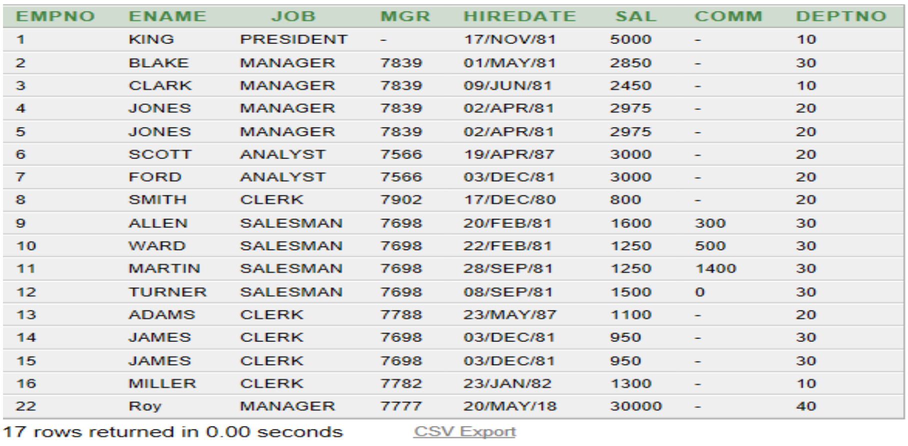
SYNTAX



```
SELECT    table1.column, table2.column
FROM      table1, table2
WHERE     table1.column1 = table2.column2;
```

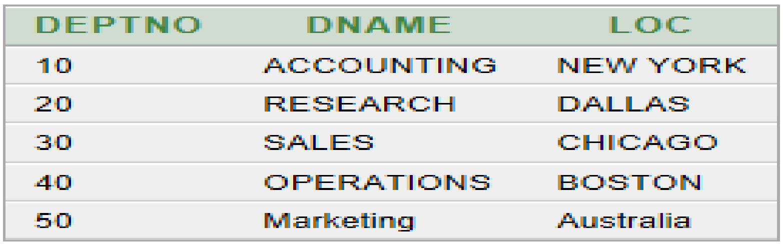
- Oracle Database performs a join whenever multiple tables appear in the FROM clause of the query. The select list of the query can select any columns from any of these tables.
- Rows in one table can be joined to rows in another table according to common values existing in corresponding columns (usually primary and foreign key columns).

EXAMPLE:



EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	KING	PRESIDENT	-	17/NOV/81	5000	-	10
2	BLAKE	MANAGER	7839	01/MAY/81	2850	-	30
3	CLARK	MANAGER	7839	09/JUN/81	2450	-	10
4	JONES	MANAGER	7839	02/APR/81	2975	-	20
5	JONES	MANAGER	7839	02/APR/81	2975	-	20
6	SCOTT	ANALYST	7566	19/APR/87	3000	-	20
7	FORD	ANALYST	7566	03/DEC/81	3000	-	20
8	SMITH	CLERK	7902	17/DEC/80	800	-	20
9	ALLEN	SALESMAN	7698	20/FEB/81	1600	300	30
10	WARD	SALESMAN	7698	22/FEB/81	1250	500	30
11	MARTIN	SALESMAN	7698	28/SEP/81	1250	1400	30
12	TURNER	SALESMAN	7698	08/SEP/81	1500	0	30
13	ADAMS	CLERK	7788	23/MAY/87	1100	-	20
14	JAMES	CLERK	7698	03/DEC/81	950	-	30
15	JAMES	CLERK	7698	03/DEC/81	950	-	30
16	MILLER	CLERK	7782	23/JAN/82	1300	-	10
22	Roy	MANAGER	7777	20/MAY/18	30000	-	40

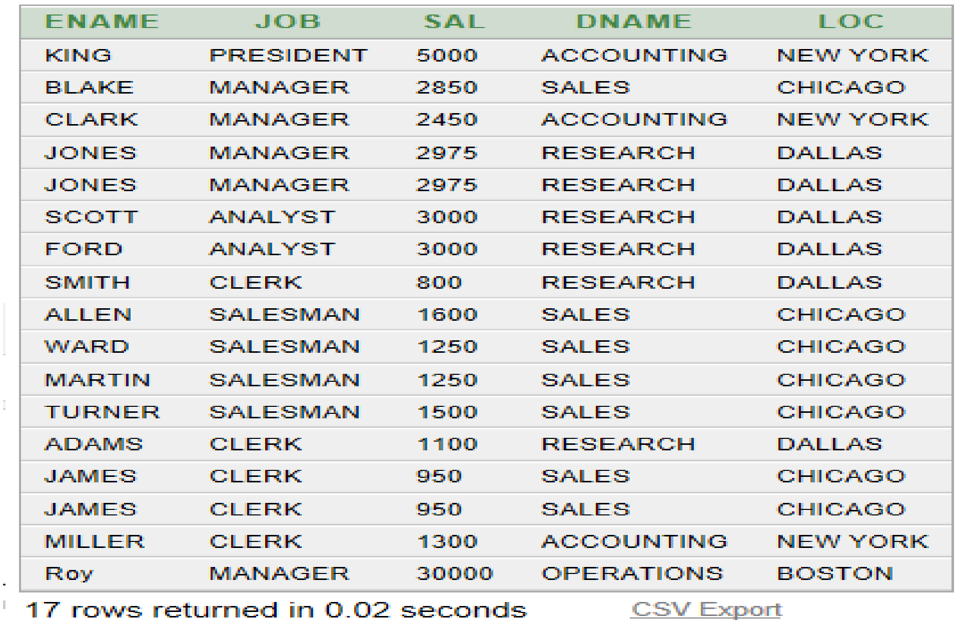
17 rows returned in 0.00 seconds CSV Export



DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON
50	Marketing	Australia

SELECT emp.ename, emp.job, emp.sal, dept.dname, dept.loc

FROM emp , dept WHERE emp.deptno = dept.deptno;



ENAME	JOB	SAL	DNAME	LOC
KING	PRESIDENT	5000	ACCOUNTING	NEW YORK
BLAKE	MANAGER	2850	SALES	CHICAGO
CLARK	MANAGER	2450	ACCOUNTING	NEW YORK
JONES	MANAGER	2975	RESEARCH	DALLAS
JONES	MANAGER	2975	RESEARCH	DALLAS
SCOTT	ANALYST	3000	RESEARCH	DALLAS
FORD	ANALYST	3000	RESEARCH	DALLAS
SMITH	CLERK	800	RESEARCH	DALLAS
ALLEN	SALESMAN	1600	SALES	CHICAGO
WARD	SALESMAN	1250	SALES	CHICAGO
MARTIN	SALESMAN	1250	SALES	CHICAGO
TURNER	SALESMAN	1500	SALES	CHICAGO
ADAMS	CLERK	1100	RESEARCH	DALLAS
JAMES	CLERK	950	SALES	CHICAGO
JAMES	CLERK	950	SALES	CHICAGO
MILLER	CLERK	1300	ACCOUNTING	NEW YORK
Roy	MANAGER	30000	OPERATIONS	BOSTON

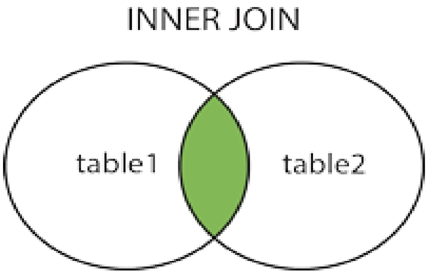
17 rows returned in 0.02 seconds CSV Export

Types Of Joins

INNER JOIN

- Retrieves records that have common values in both tables.

- Returns records that have matching values in both tables



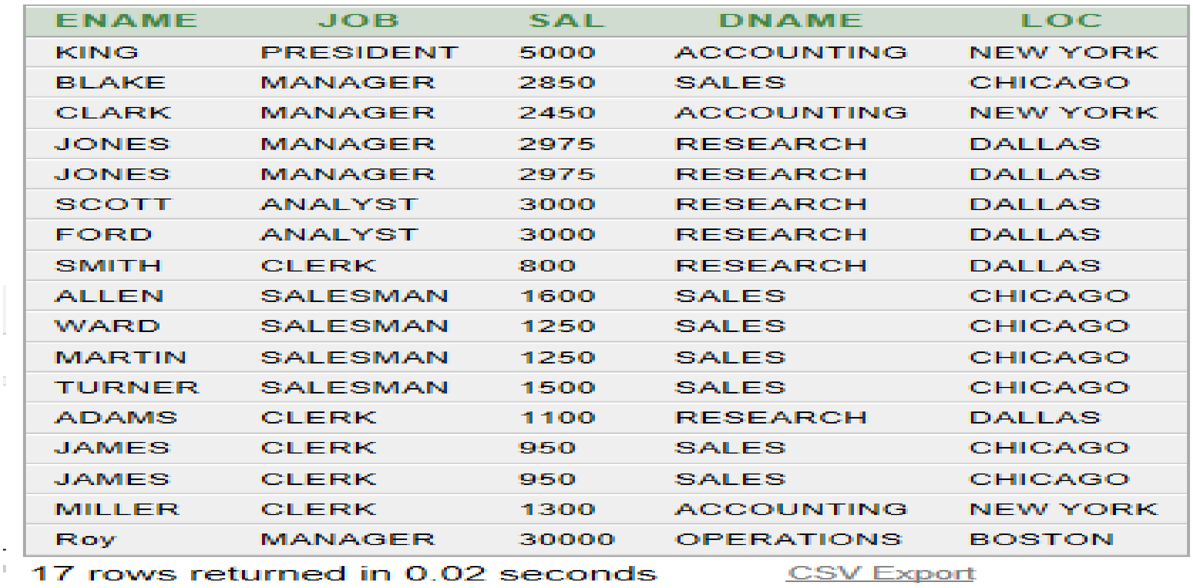
- **Syntax:**

  SELECT column_name(s)

  FROM table1

  INNER JOIN table2 ON

  table1.column_name = table2.column_name;



ENAME	JOB	SAL	DNAME	LOC
KING	PRESIDENT	5000	ACCOUNTING	NEW YORK
BLAKE	MANAGER	2850	SALES	CHICAGO
CLARK	MANAGER	2450	ACCOUNTING	NEW YORK
JONES	MANAGER	2975	RESEARCH	DALLAS
JONES	MANAGER	2975	RESEARCH	DALLAS
SCOTT	ANALYST	3000	RESEARCH	DALLAS
FORD	ANALYST	3000	RESEARCH	DALLAS
SMITH	CLERK	800	RESEARCH	DALLAS
ALLEN	SALESMAN	1600	SALES	CHICAGO
WARD	SALESMAN	1250	SALES	CHICAGO
MARTIN	SALESMAN	1250	SALES	CHICAGO
TURNER	SALESMAN	1500	SALES	CHICAGO
ADAMS	CLERK	1100	RESEARCH	DALLAS
JAMES	CLERK	950	SALES	CHICAGO
JAMES	CLERK	950	SALES	CHICAGO
MILLER	CLERK	1300	ACCOUNTING	NEW YORK
Roy	MANAGER	30000	OPERATIONS	BOSTON

17 rows returned in 0.02 seconds CSV Export

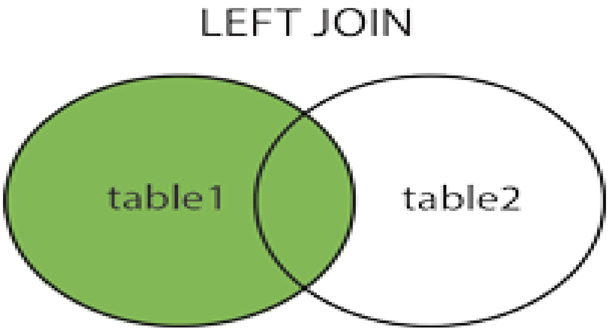
OUTER JOINS:

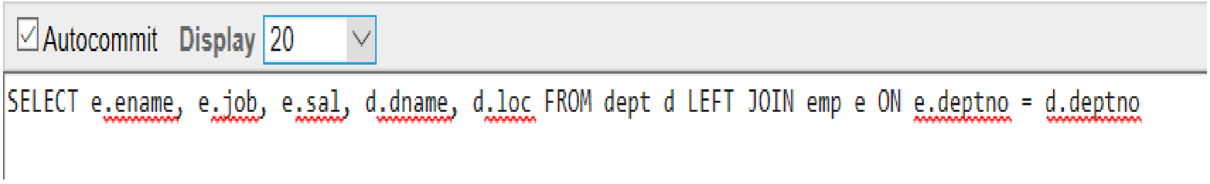
- Outer Join includes records of a table In O/P when there is no matching record in the other table.
- Types:
- Left Outer Join
- Right Outer Join
- Full Outer Join

LEFT OUTER JOIN:

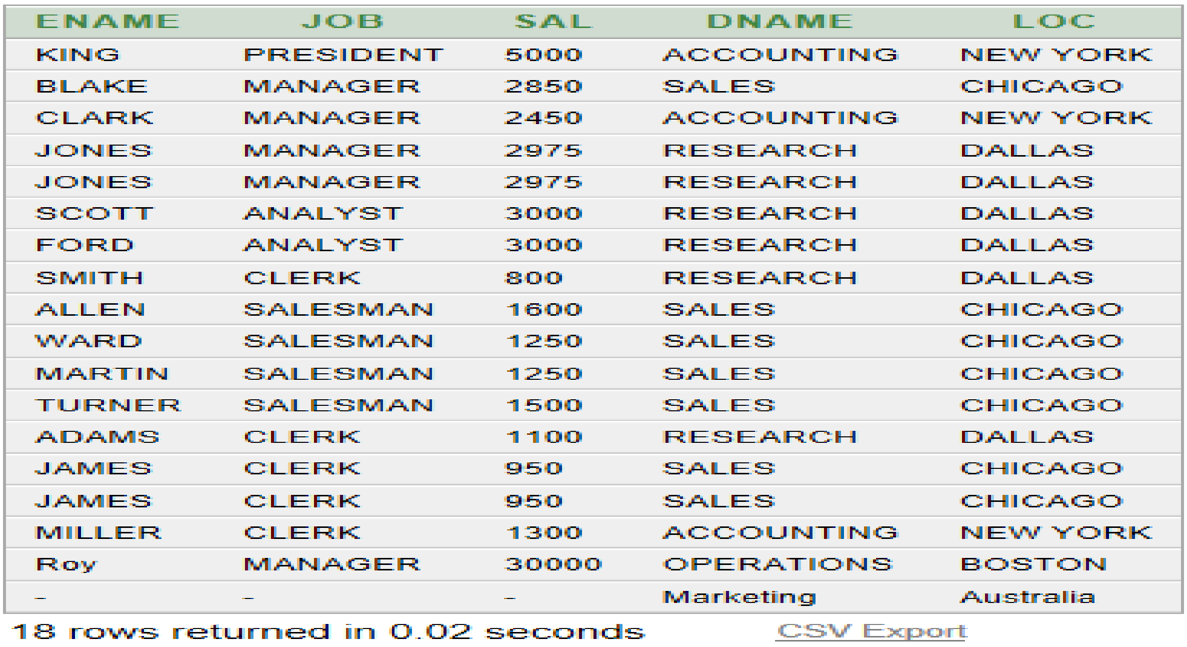
- The LEFT JOIN keyword returns all records from the left table (table1), and the matched records from the right table (table2). The result is NULL from the right side, if there is no match.

- LEFT JOIN is called LEFT OUTER JOIN.



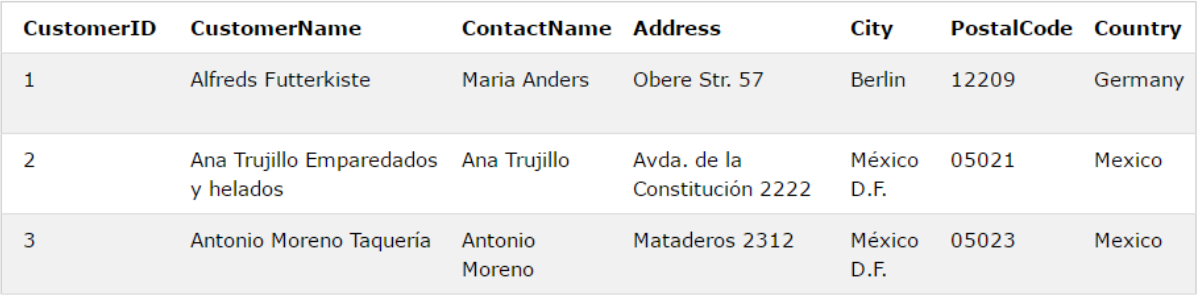

☑ Autocommit Display 20

```
SELECT e.ename, e.job, e.sal, d.dname, d.loc FROM dept d LEFT JOIN emp e ON e.deptno = d.deptno
```

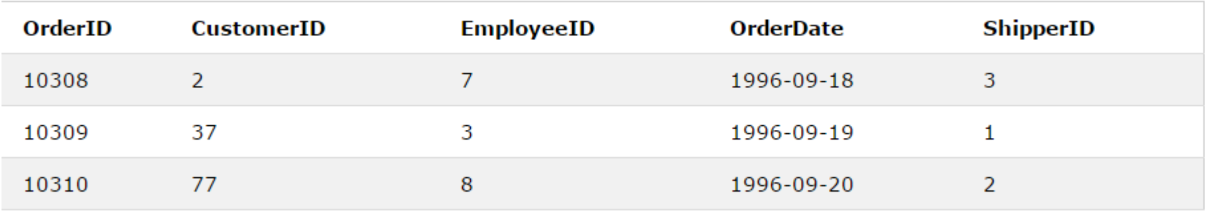


ENAME	JOB	SAL	DNAME	LOC
KING	PRESIDENT	5000	ACCOUNTING	NEW YORK
BLAKE	MANAGER	2850	SALES	CHICAGO
CLARK	MANAGER	2450	ACCOUNTING	NEW YORK
JONES	MANAGER	2975	RESEARCH	DALLAS
JONES	MANAGER	2975	RESEARCH	DALLAS
SCOTT	ANALYST	3000	RESEARCH	DALLAS
FORD	ANALYST	3000	RESEARCH	DALLAS
SMITH	CLERK	800	RESEARCH	DALLAS
ALLEN	SALESMAN	1600	SALES	CHICAGO
WARD	SALESMAN	1250	SALES	CHICAGO
MARTIN	SALESMAN	1250	SALES	CHICAGO
TURNER	SALESMAN	1500	SALES	CHICAGO
ADAMS	CLERK	1100	RESEARCH	DALLAS
JAMES	CLERK	950	SALES	CHICAGO
JAMES	CLERK	950	SALES	CHICAGO
MILLER	CLERK	1300	ACCOUNTING	NEW YORK
Roy	MANAGER	30000	OPERATIONS	BOSTON
-	-	-	Marketing	Australia

18 rows returned in 0.02 seconds CSV Export



CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

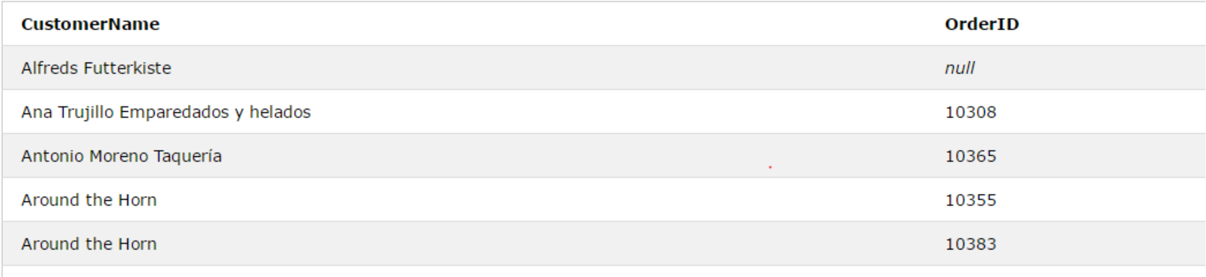


OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10308	2	7	1996-09-18	3
10309	37	3	1996-09-19	1
10310	77	8	1996-09-20	2

SELECT Customers.CustomerName, Orders.OrderID

FROM Customers LEFT JOIN Orders

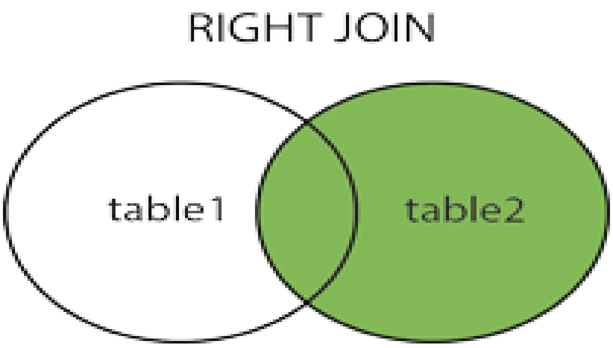
ON Customers.CustomerID=Orders.CustomerID;4

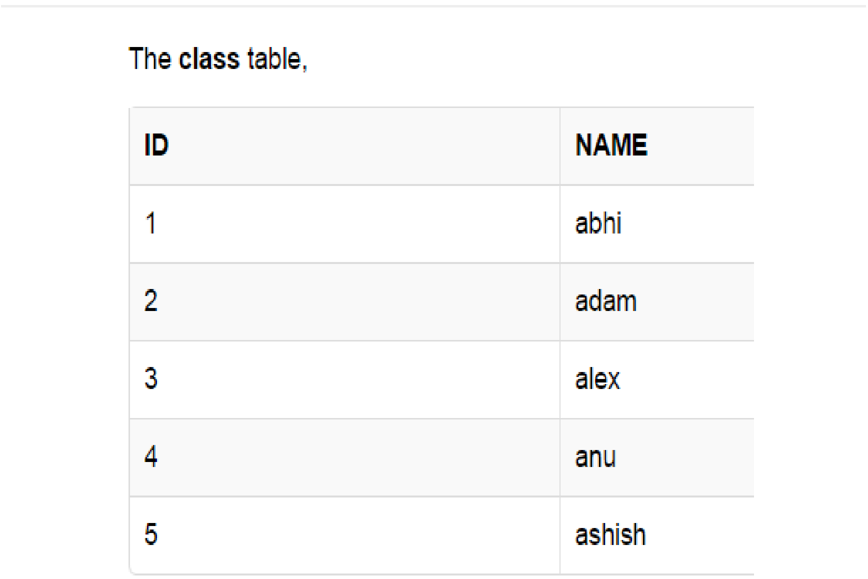


CustomerName	OrderID
Alfreds Futterkiste	*null*
Ana Trujillo Emparedados y helados	10308
Antonio Moreno Taquería	10365
Around the Horn	10355
Around the Horn	10383

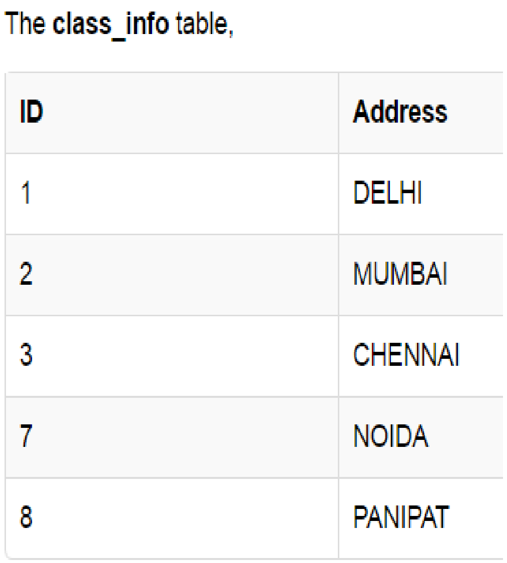
RIGHT JOIN

- The RIGHT JOIN keyword returns all records from the right table (table2), and the matched records from the left table (table1). The result is NULL from the left side, when there is no match.
- RIGHT JOIN is called RIGHT OUTER JOIN.



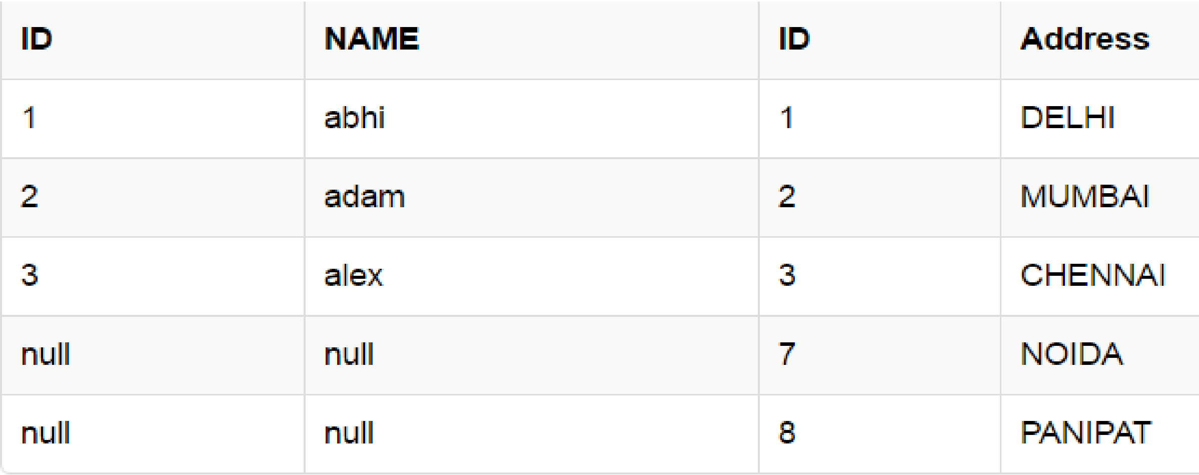

The **class** table,

ID	NAME
1	abhi
2	adam
3	alex
4	anu
5	ashish

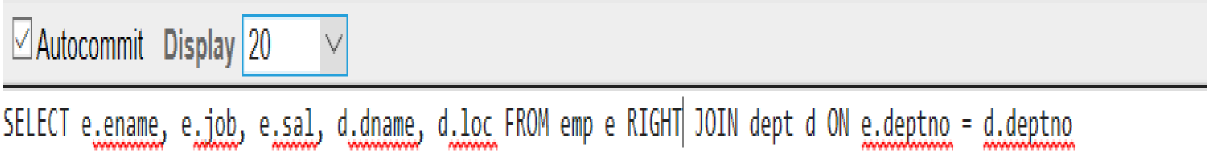

The **class_info** table,

ID	Address
1	DELHI
2	MUMBAI
3	CHENNAI
7	NOIDA
8	PANIPAT

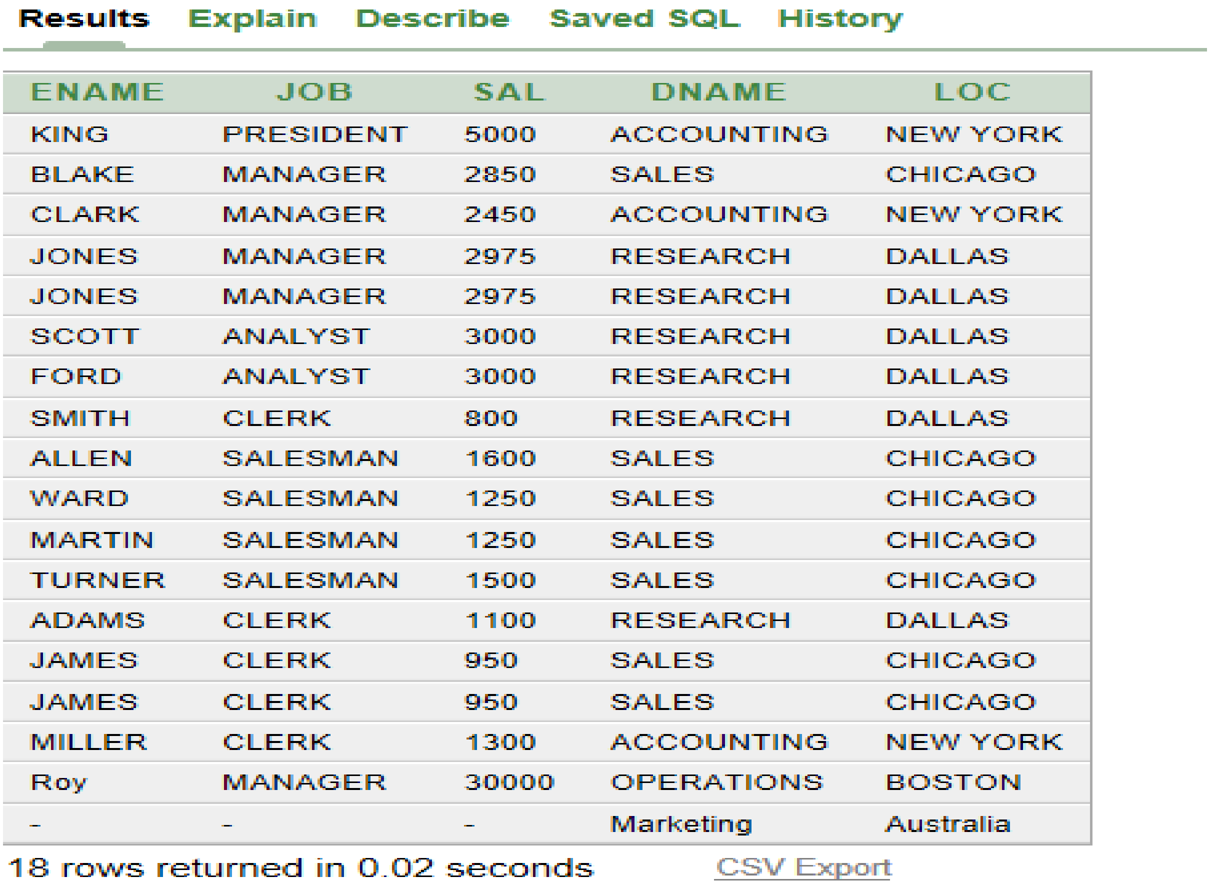
SELECT * FROM class RIGHT OUTER JOIN class_info ON (class.id=class_info.id);



ID	NAME	ID	Address
1	abhi	1	DELHI
2	adam	2	MUMBAI
3	alex	3	CHENNAI
null	null	7	NOIDA
null	null	8	PANIPAT



```
SELECT e.ename, e.job, e.sal, d.dname, d.loc FROM emp e RIGHT JOIN dept d ON e.deptno = d.deptno
```

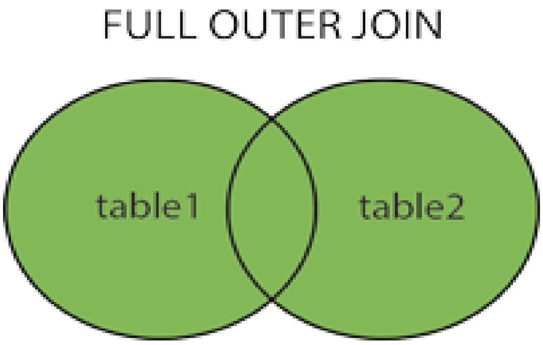


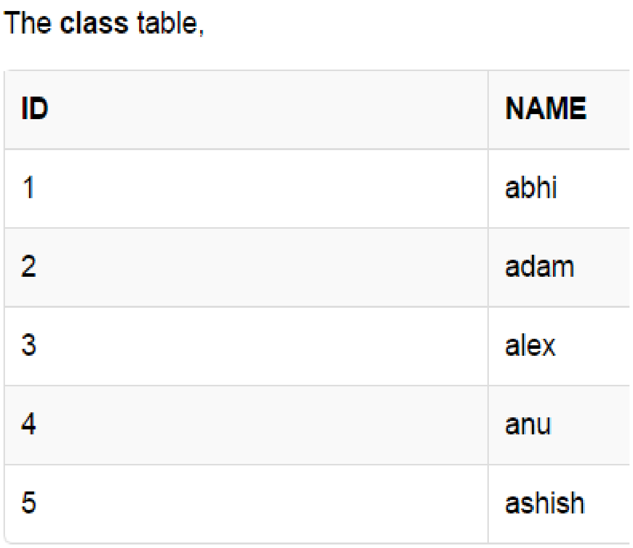
ENAME	JOB	SAL	DNAME	LOC
KING	PRESIDENT	5000	ACCOUNTING	NEW YORK
BLAKE	MANAGER	2850	SALES	CHICAGO
CLARK	MANAGER	2450	ACCOUNTING	NEW YORK
JONES	MANAGER	2975	RESEARCH	DALLAS
JONES	MANAGER	2975	RESEARCH	DALLAS
SCOTT	ANALYST	3000	RESEARCH	DALLAS
FORD	ANALYST	3000	RESEARCH	DALLAS
SMITH	CLERK	800	RESEARCH	DALLAS
ALLEN	SALESMAN	1600	SALES	CHICAGO
WARD	SALESMAN	1250	SALES	CHICAGO
MARTIN	SALESMAN	1250	SALES	CHICAGO
TURNER	SALESMAN	1500	SALES	CHICAGO
ADAMS	CLERK	1100	RESEARCH	DALLAS
JAMES	CLERK	950	SALES	CHICAGO
JAMES	CLERK	950	SALES	CHICAGO
MILLER	CLERK	1300	ACCOUNTING	NEW YORK
Roy	MANAGER	30000	OPERATIONS	BOSTON
-	-	-	Marketing	Australia

18 rows returned in 0.02 seconds

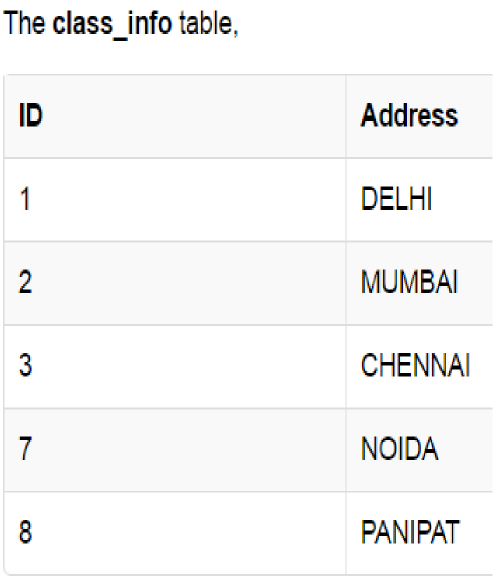
FULL OUTER JOIN

- The FULL OUTER JOIN keyword returns all records from table 1 even if there is no match in table 2 and vice versa.



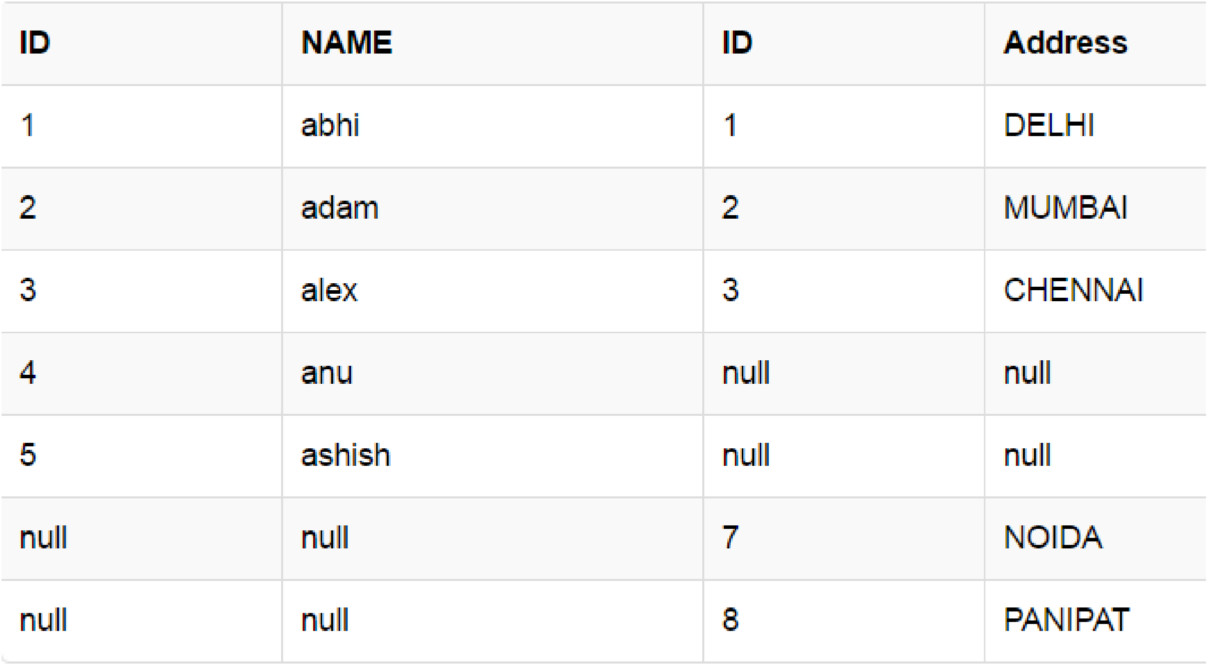

The **class** table,

ID	NAME
1	abhi
2	adam
3	alex
4	anu
5	ashish


The **class_info** table,

ID	Address
1	DELHI
2	MUMBAI
3	CHENNAI
7	NOIDA
8	PANIPAT

SELECT * FROM class FULL OUTER JOIN class_info ON (class.id=class_info.id);



ID	NAME	ID	Address
1	abhi	1	DELHI
2	adam	2	MUMBAI
3	alex	3	CHENNAI
4	anu	null	null
5	ashish	null	null
null	null	7	NOIDA
null	null	8	PANIPAT

PERFORMING OUTER JOINS USING THE (+) SYMBOL

- Oracle provides a special outer join operator (the + symbol) that is shorthand for performing OUTER JOINS.
- The + symbol is placed on the side of the table (the one which is allowed to contain empty or null).
- + symbol denotes the optional table in the join that contain null values.
- **LEFT OUTER JOIN**

  ```
  SELECT *
  FROM A, B
  WHERE A.column = B.column(+) ;
  ```

  B Table contains null values.
- **RIGHT OUTER JOIN**

  ```
  SELECT *
  FROM A, B
  ```

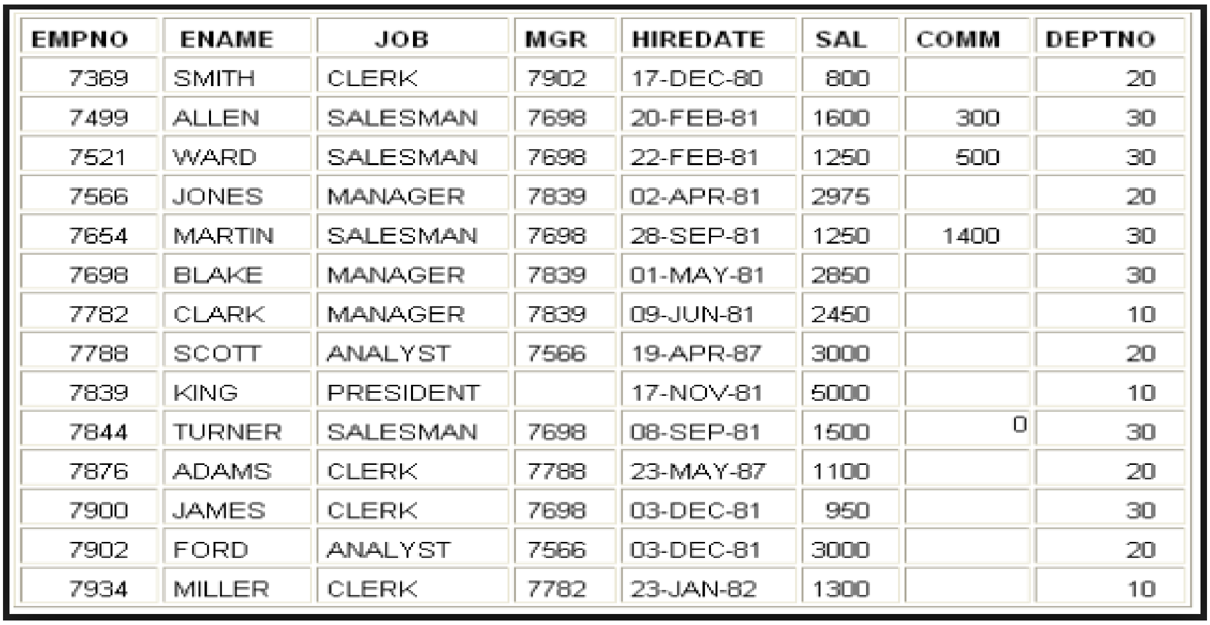
WHERE A.column(+) = B.column ;

Table A contains null values.

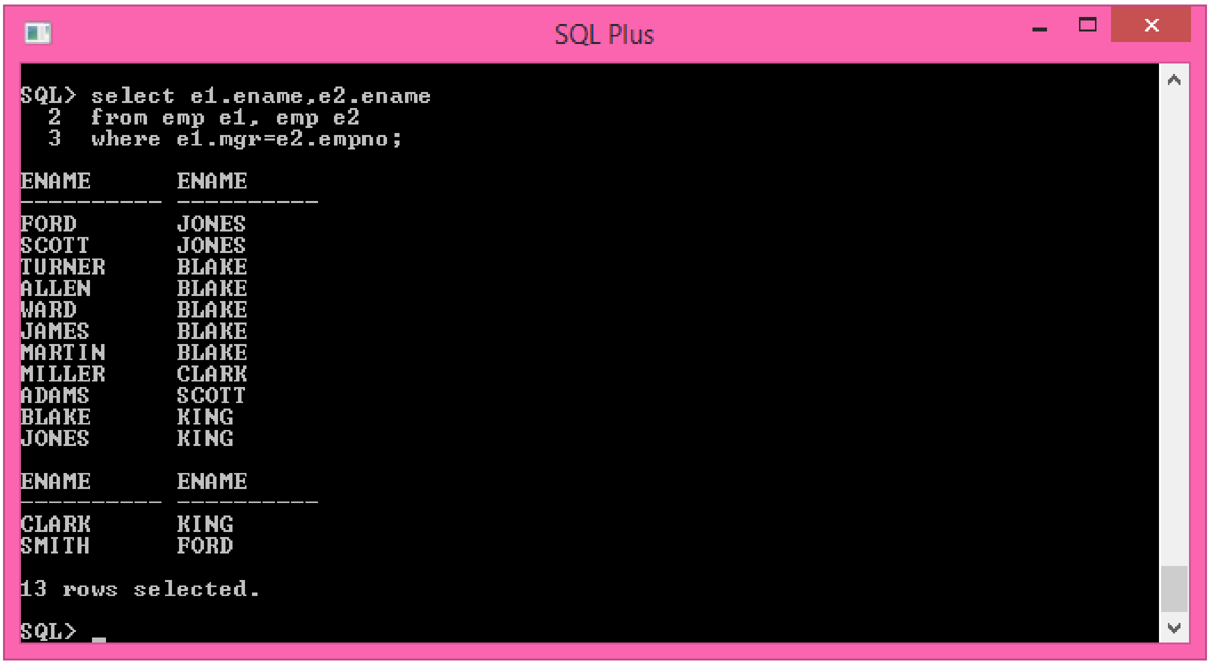
Self JOIN

- A self JOIN is a regular join, but the table is joined with itself.

Joining a table to itself is called self join.



EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

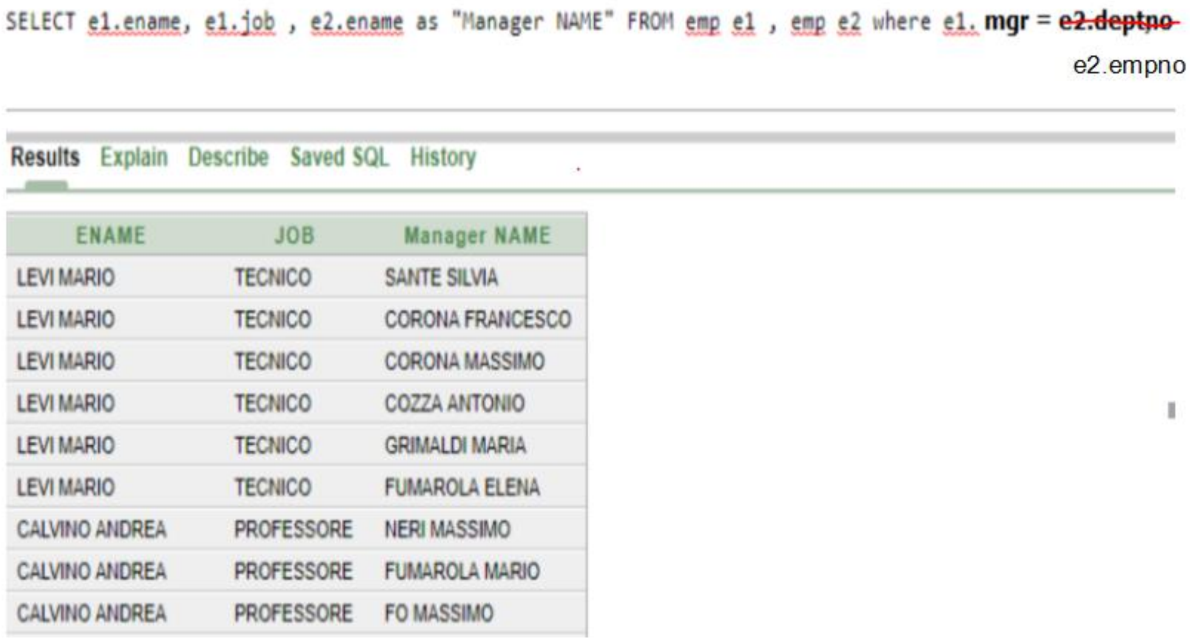


```
SQL> select e1.ename,e2.ename
  2  from emp e1, emp e2
  3  where e1.mgr=e2.empno;

ENAME      ENAME
---------- ----------
FORD       JONES
SCOTT      JONES
TURNER     BLAKE
ALLEN      BLAKE
WARD       BLAKE
JAMES      BLAKE
MARTIN     BLAKE
MILLER     CLARK
ADAMS      SCOTT
BLAKE      KING
JONES      KING

ENAME      ENAME
---------- ----------
CLARK      KING
SMITH      FORD

13 rows selected.

SQL> _
```

EXAMPLE

Find the name of King's manager ?

- Find King in the emp table
- Find the manager number for king (mgr = 101)
- Find the name of manager with empno 101
- How to solve this query?
  - Use Self Joins



```
SELECT e1.ename, e1.job , e2.ename as "Manager NAME" FROM emp e1 , emp e2 where e1. mgr = ~~e2.deptno~~
e2.empno
```

ENAME	JOB	Manager NAME
LEVI MARIO	TECNICO	SANTE SILVIA
LEVI MARIO	TECNICO	CORONA FRANCESCO
LEVI MARIO	TECNICO	CORONA MASSIMO
LEVI MARIO	TECNICO	COZZA ANTONIO
LEVI MARIO	TECNICO	GRIMALDI MARIA
LEVI MARIO	TECNICO	FUMAROLA ELENA
CALVINO ANDREA	PROFESSORE	NERI MASSIMO
CALVINO ANDREA	PROFESSORE	FUMAROLA MARIO
CALVINO ANDREA	PROFESSORE	FO MASSIMO

EQUI JOINS

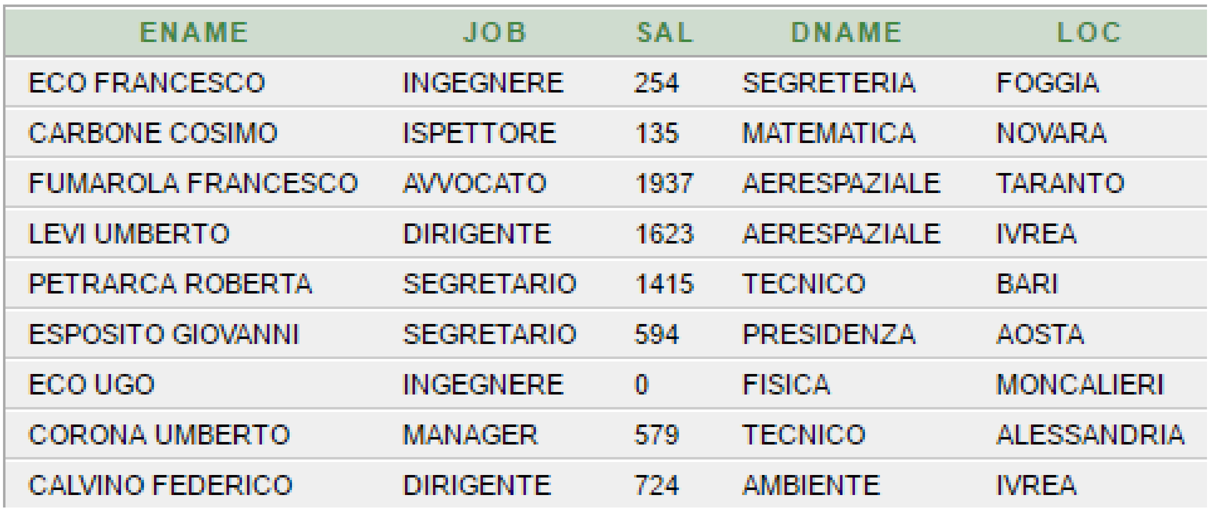
- Returns records that have matching values in both tables.
- SQL EQUI JOIN performs a JOIN against equality or matching column(s) values of the associated tables.
- An equal sign (=) is used as comparison operator in the where clause to refer equality.
- You may also perform EQUI JOIN by using JOIN keyword followed by ON keyword and then specifying names of the columns along with their associated tables to check equality.

SYNTAX:

```
SELECT column_list
```

```
FROM table1, table2....
WHERE table1.column_name =
table2.column_name;
OR
SELECT *
FROM table1
JOIN table2
[ON (join_condition)]
```

```
SELECT emp.ename, emp.job, emp.sal, dept.dname, dept.loc
FROM emp  , dept WHERE emp.deptno = dept.deptno;
```



ENAME	JOB	SAL	DNAME	LOC
ECO FRANCESCO	INGEGNERE	254	SEGRETERIA	FOGGIA
CARBONE COSIMO	ISPETTORE	135	MATEMATICA	NOVARA
FUMAROLA FRANCESCO	AVVOCATO	1937	AERESPAZIALE	TARANTO
LEVI UMBERTO	DIRIGENTE	1623	AERESPAZIALE	IVREA
PETRARCA ROBERTA	SEGRETARIO	1415	TECNICO	BARI
ESPOSITO GIOVANNI	SEGRETARIO	594	PRESIDENZA	AOSTA
ECO UGO	INGEGNERE	0	FISICA	MONCALIERI
CORONA UMBERTO	MANAGER	579	TECNICO	ALESSANDRIA
CALVINO FEDERICO	DIRIGENTE	724	AMBIENTE	IVREA

- Inner join can have equality (=) and other operators (like <,>,<>) in the join condition.

- Equi join only have equality (=) operator in the join condition.

- Equi join can be an Inner join, Left Outer join, Right Outer join.

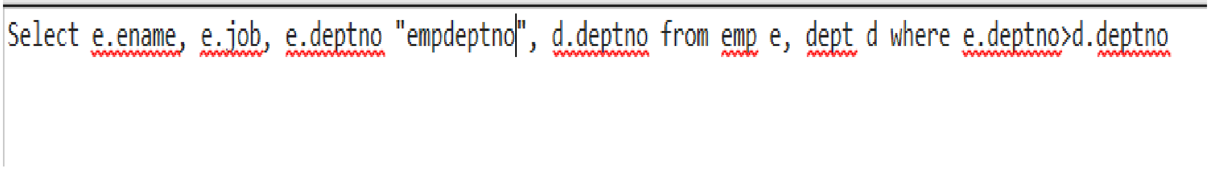
(Almost every join is an equijoin, because the condition for matching rows is based on the equality of two values)

NON-EQUI JOINS

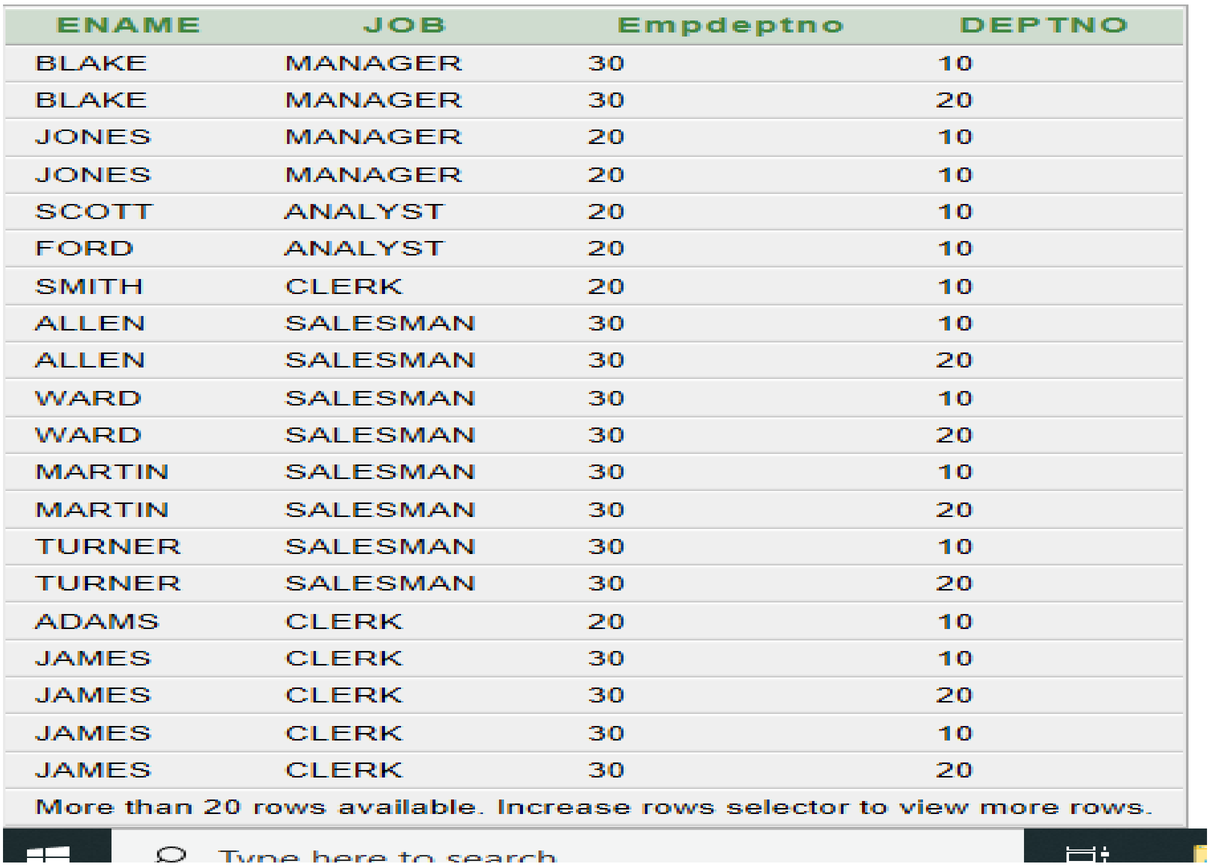
- The SQL NON EQUI JOIN uses comparison operator instead of the equal sign like >, <, >=, <= along with conditions.

- **SYNTAX**

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

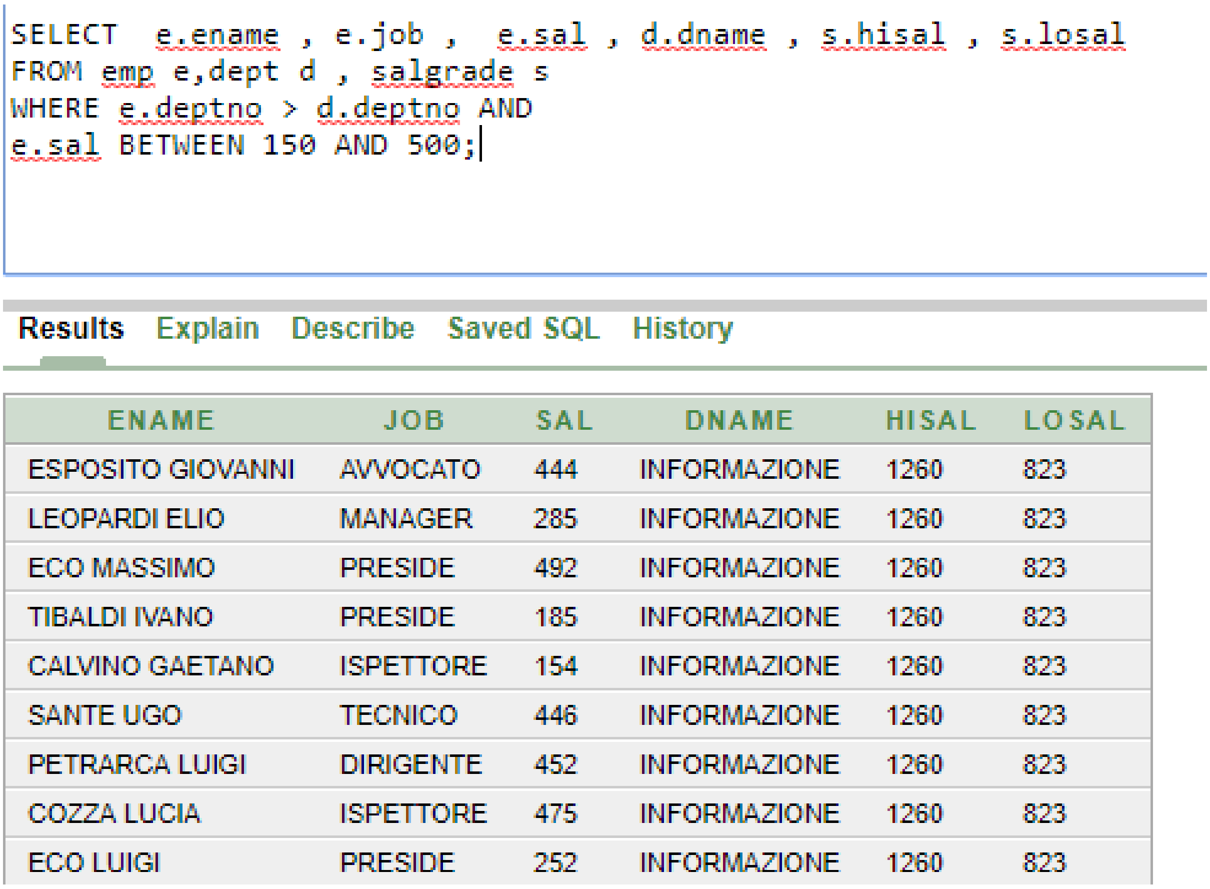


```
Select e.ename, e.job, e.deptno "empdeptno", d.deptno from emp e, dept d where e.deptno>d.deptno
```



ENAME	JOB	Empdeptno	DEPTNO
BLAKE	MANAGER	30	10
BLAKE	MANAGER	30	20
JONES	MANAGER	20	10
JONES	MANAGER	20	10
SCOTT	ANALYST	20	10
FORD	ANALYST	20	10
SMITH	CLERK	20	10
ALLEN	SALESMAN	30	10
ALLEN	SALESMAN	30	20
WARD	SALESMAN	30	10
WARD	SALESMAN	30	20
MARTIN	SALESMAN	30	10
MARTIN	SALESMAN	30	20
TURNER	SALESMAN	30	10
TURNER	SALESMAN	30	20
ADAMS	CLERK	20	10
JAMES	CLERK	30	10
JAMES	CLERK	30	20
JAMES	CLERK	30	10
JAMES	CLERK	30	20

More than 20 rows available. Increase rows selector to view more rows.



```
SELECT  e.ename , e.job ,  e.sal , d.dname , s.hisal , s.losal
FROM emp e,dept d , salgrade s
WHERE e.deptno > d.deptno AND
e.sal BETWEEN 150 AND 500;
```

ENAME	JOB	SAL	DNAME	HISAL	LOSAL
ESPOSITO GIOVANNI	AVVOCATO	444	INFORMAZIONE	1260	823
LEOPARDI ELIO	MANAGER	285	INFORMAZIONE	1260	823
ECO MASSIMO	PRESIDE	492	INFORMAZIONE	1260	823
TIBALDI IVANO	PRESIDE	185	INFORMAZIONE	1260	823
CALVINO GAETANO	ISPETTORE	154	INFORMAZIONE	1260	823
SANTE UGO	TECNICO	446	INFORMAZIONE	1260	823
PETRARCA LUIGI	DIRIGENTE	452	INFORMAZIONE	1260	823
COZZA LUCIA	ISPETTORE	475	INFORMAZIONE	1260	823
ECO LUIGI	PRESIDE	252	INFORMAZIONE	1260	823

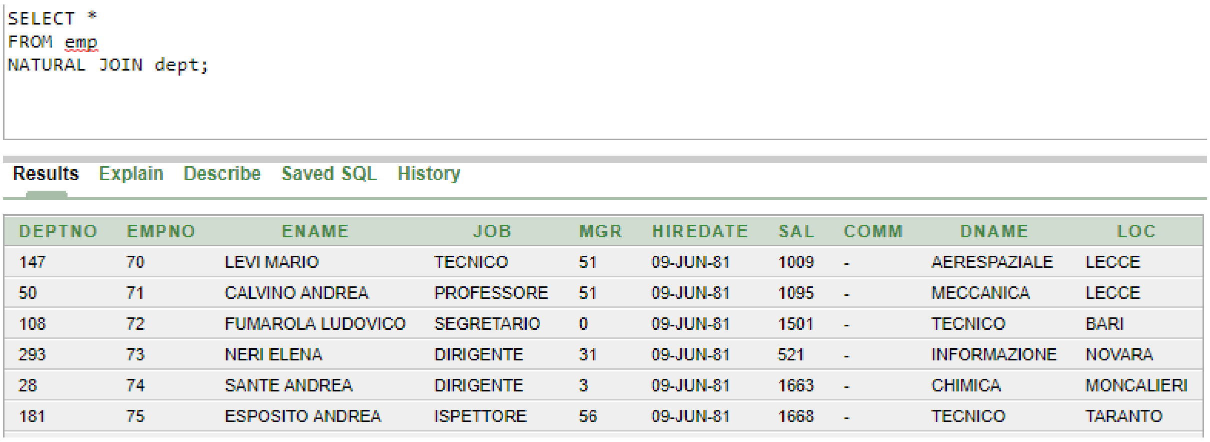
NATURAL JOINS

- The SQL NATURAL JOIN is a type of EQUI JOIN and is structured in such a way that, columns with the same name of associated tables(“deptno” column in emp and dept table) will appear once only.
- A NATURAL JOIN is a JOIN operation that creates join clause for you based on the common columns in the two tables being joined. Common columns are columns that have the same name in both tables.
- A NATURAL JOIN can be an INNER join, a LEFT OUTER join, or a RIGHT OUTER join. The default is INNER join.

Syntax:

```
SELECT *
FROM table1
NATURAL JOIN table2;
```

Don’t use ON clause in a natural join.



```
SELECT *
FROM emp
NATURAL JOIN dept;
```

DEPTNO	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DNAME	LOC
147	70	LEVI MARIO	TECNICO	51	09-JUN-81	1009	-	AERESPAZIALE	LECCE
50	71	CALVINO ANDREA	PROFESSORE	51	09-JUN-81	1095	-	MECCANICA	LECCE
108	72	FUMAROLA LUDOVICO	SEGRETARIO	0	09-JUN-81	1501	-	TECNICO	BARI
293	73	NERI ELENA	DIRIGENTE	31	09-JUN-81	521	-	INFORMAZIONE	NOVARA
28	74	SANTE ANDREA	DIRIGENTE	3	09-JUN-81	1663	-	CHIMICA	MONCALIERI
181	75	ESPOSITO ANDREA	ISPETTORE	56	09-JUN-81	1668	-	TECNICO	TARANTO

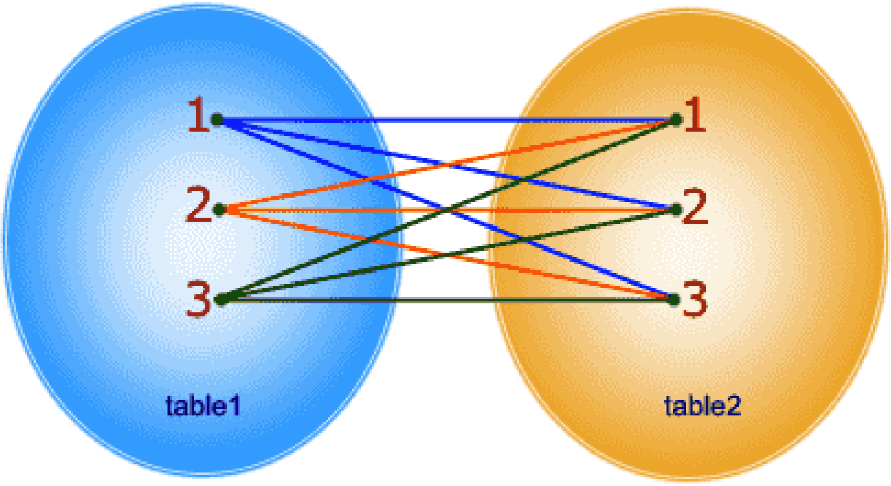
CROSS JOIN

- The SQL CROSS JOIN produces a result set which is the number of rows in the first table multiplied by the number of rows in the second table if no WHERE clause is used along with CROSS JOIN.This kind of result is called as Cartesian Product.
- If WHERE clause is used with CROSS JOIN, it functions like an INNER JOIN.

Syntax:

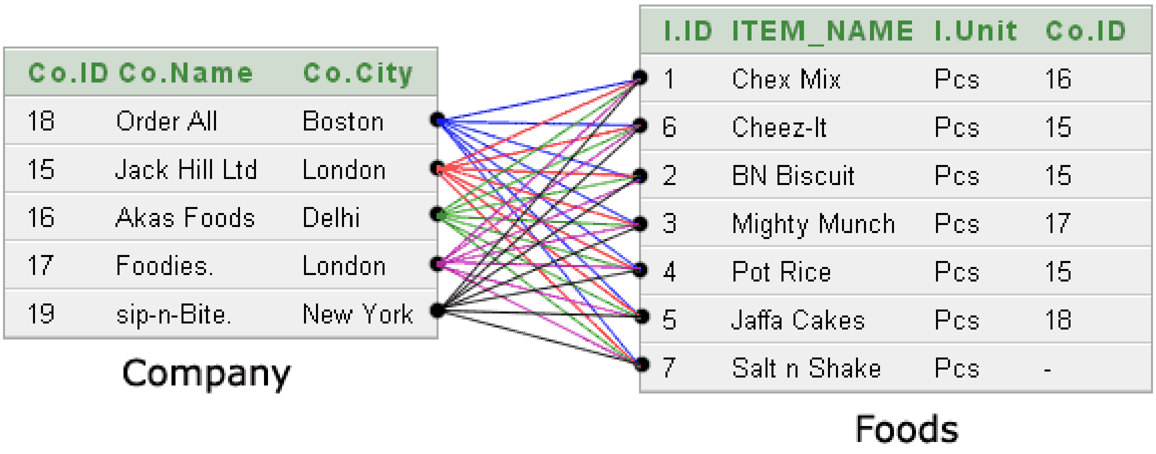
```
SELECT *
FROM table1
CROSS JOIN table2;
```

```
SELECT * FROM table1 CROSS JOIN  table2;
```



In CROSS JOIN, each row from 1st table joins with all the rows of another table.
If 1st table contain x rows and y rows in 2nd one the result set will be x * y rows.

```
SELECT foods.item_name,foods.item_unit,
company.company_name,company.company_city
FROM foods
CROSS JOIN company;
```

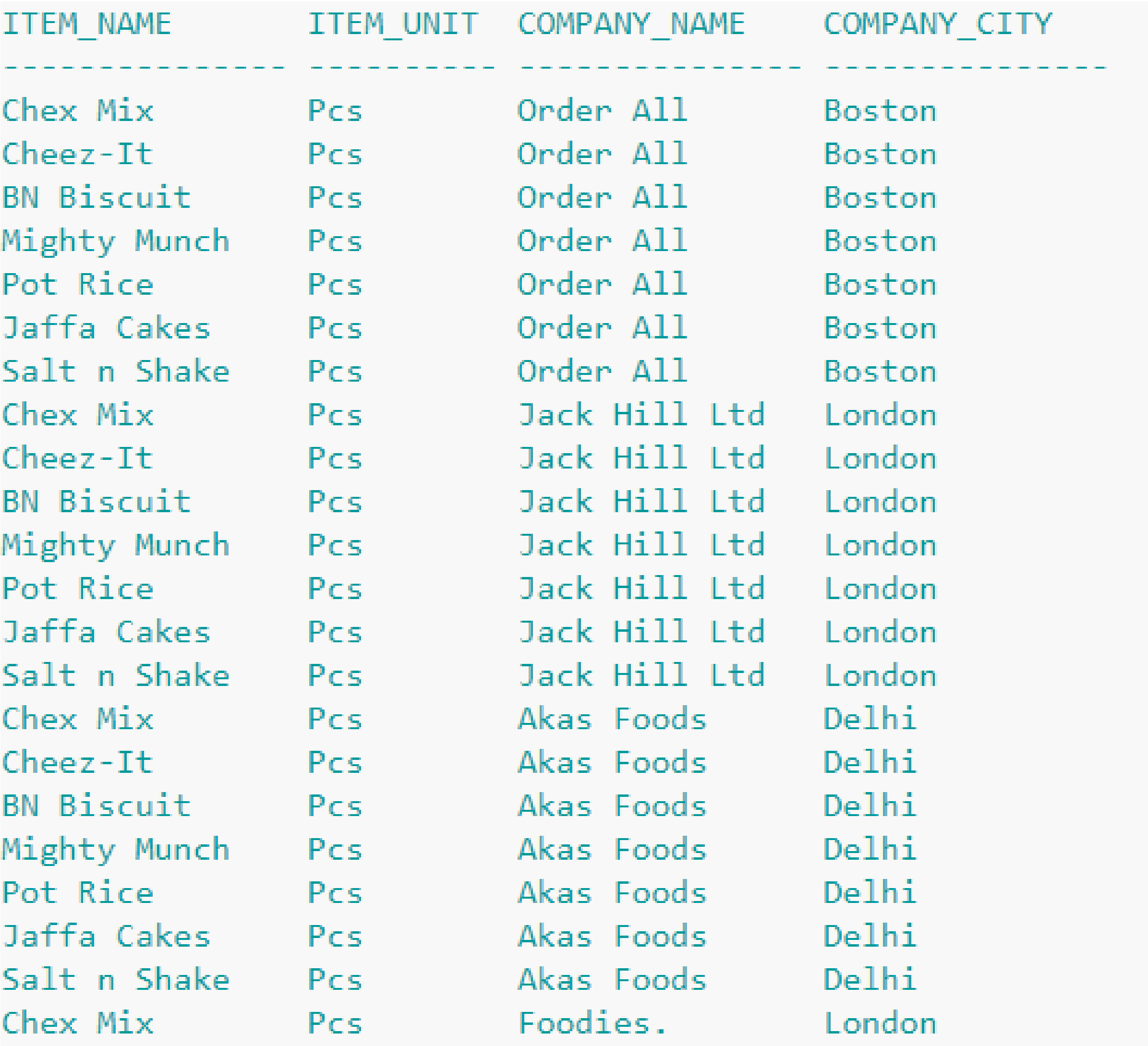


Co.ID	Co.Name	Co.City
18	Order All	Boston
15	Jack Hill Ltd	London
16	Akas Foods	Delhi
17	Foodies.	London
19	sip-n-Bite.	New York

Company

I.ID	ITEM_NAME	I.Unit	Co.ID
1	Chex Mix	Pcs	16
6	Cheez-It	Pcs	15
2	BN Biscuit	Pcs	15
3	Mighty Munch	Pcs	17
4	Pot Rice	Pcs	15
5	Jaffa Cakes	Pcs	18
7	Salt n Shake	Pcs	-

Foods

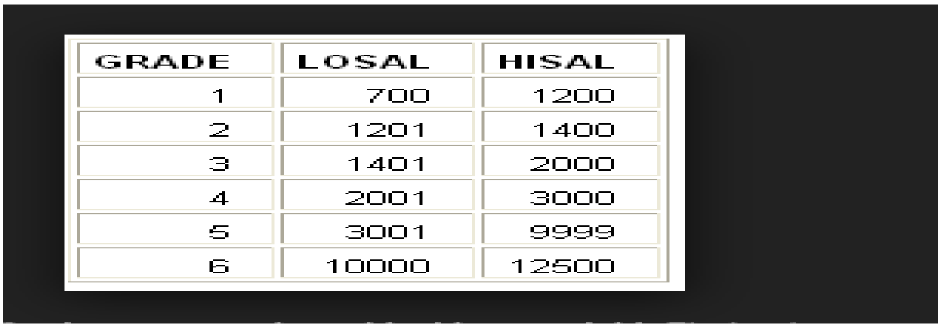


```
ITEM_NAME        ITEM_UNIT  COMPANY_NAME     COMPANY_CITY
---------------- ---------- ---------------- ----------------
Chex Mix         Pcs        Order All        Boston
Cheez-It         Pcs        Order All        Boston
BN Biscuit       Pcs        Order All        Boston
Mighty Munch     Pcs        Order All        Boston
Pot Rice         Pcs        Order All        Boston
Jaffa Cakes      Pcs        Order All        Boston
Salt n Shake     Pcs        Order All        Boston
Chex Mix         Pcs        Jack Hill Ltd    London
Cheez-It         Pcs        Jack Hill Ltd    London
BN Biscuit       Pcs        Jack Hill Ltd    London
Mighty Munch     Pcs        Jack Hill Ltd    London
Pot Rice         Pcs        Jack Hill Ltd    London
Jaffa Cakes      Pcs        Jack Hill Ltd    London
Salt n Shake     Pcs        Jack Hill Ltd    London
Chex Mix         Pcs        Akas Foods       Delhi
Cheez-It         Pcs        Akas Foods       Delhi
BN Biscuit       Pcs        Akas Foods       Delhi
Mighty Munch     Pcs        Akas Foods       Delhi
Pot Rice         Pcs        Akas Foods       Delhi
Jaffa Cakes      Pcs        Akas Foods       Delhi
Salt n Shake     Pcs        Akas Foods       Delhi
Chex Mix         Pcs        Foodies.         London
```

Lab Tasks

1. Write a query to display the employee name, department number, dept name and dept location for all the employees.
2. Create a unique listing of all jobs that are in dept 30. Include the location of department in the output.
3. Write a query to display the employee name, emp sal, department name, dept location of all employees who earn a commission.
4. Write a query that lists empno, ename, mgr id and dname from emp and dept table.

5. Write a query that returns all the employees, their related information and dept name and even those department where no employee is working.
6. Write a query to find the grade of employees based on their high and low salaries



GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999
6	10000	12500