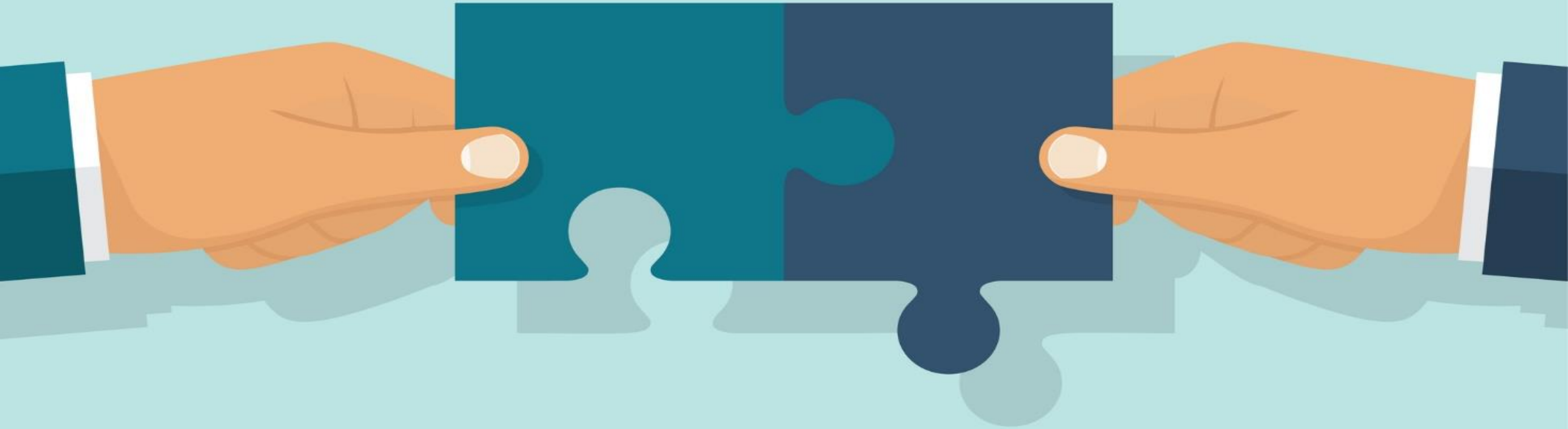


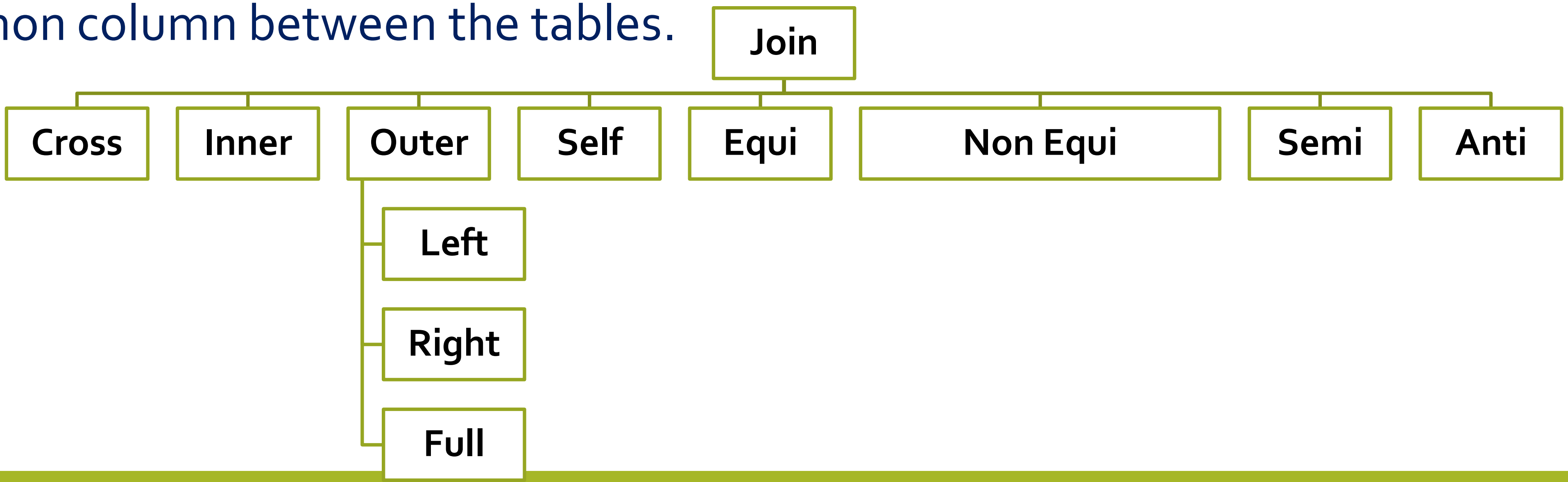
JOIN

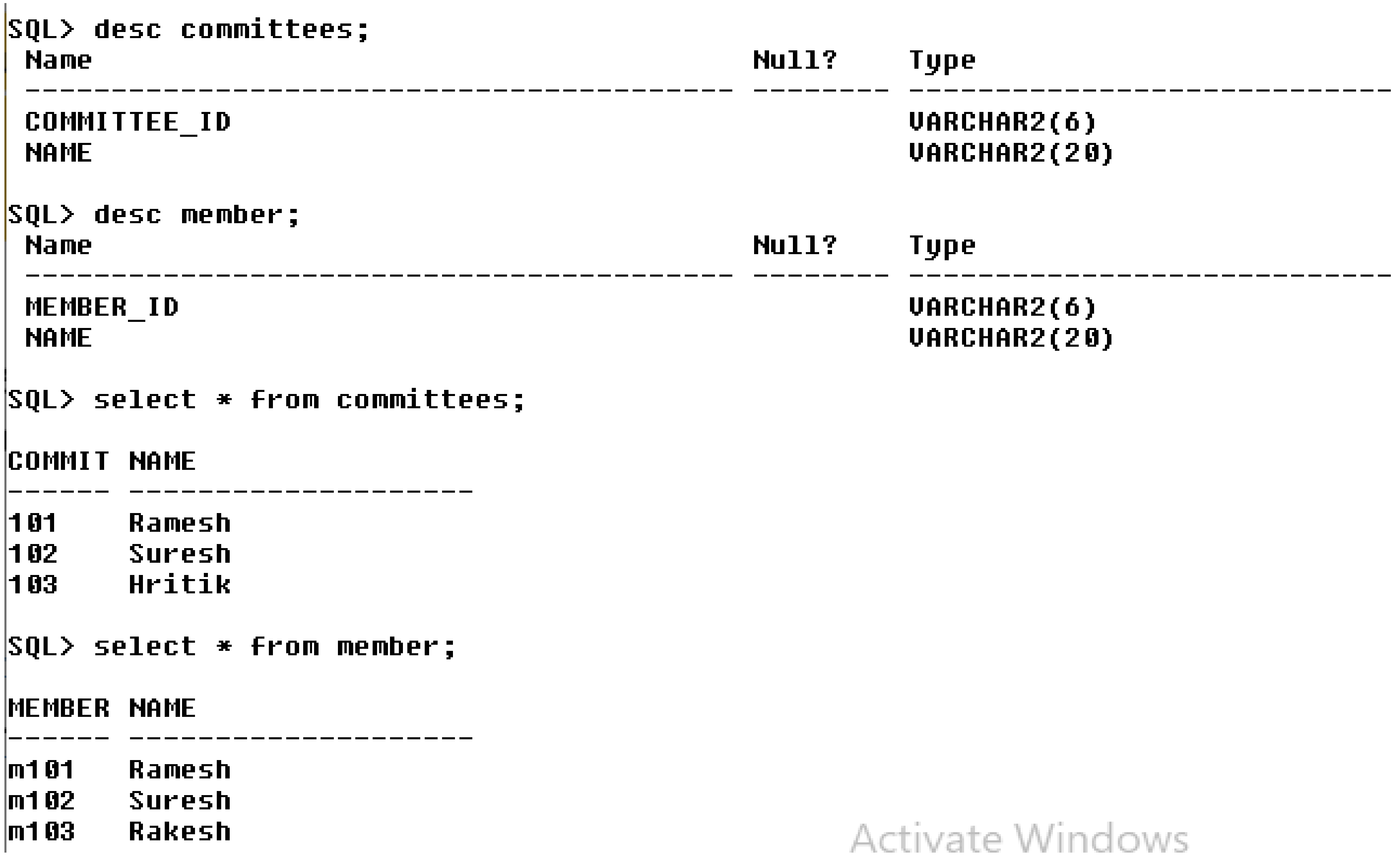
JOIN

Table1: committees	**Table2: member**
committee_id	member_id
Name	Name

Find the name in committee who is a member.
Both the tables need to be queried

- Method of linking data between one or more tables based on values of the common column between the tables.





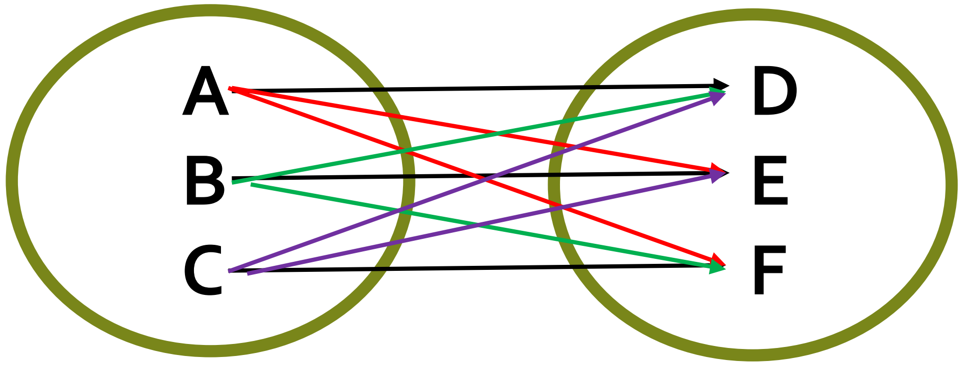
```
SQL> desc committees;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 COMMITTEE_ID                                       VARCHAR2(6)
 NAME                                               VARCHAR2(20)

SQL> desc member;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 MEMBER_ID                                          VARCHAR2(6)
 NAME                                               VARCHAR2(20)

SQL> select * from committees;

COMMIT NAME
------ --------------------
101    Ramesh
102    Suresh
103    Hritik

SQL> select * from member;

MEMBER NAME
------ --------------------
m101   Ramesh
m102   Suresh
m103   Rakesh
```

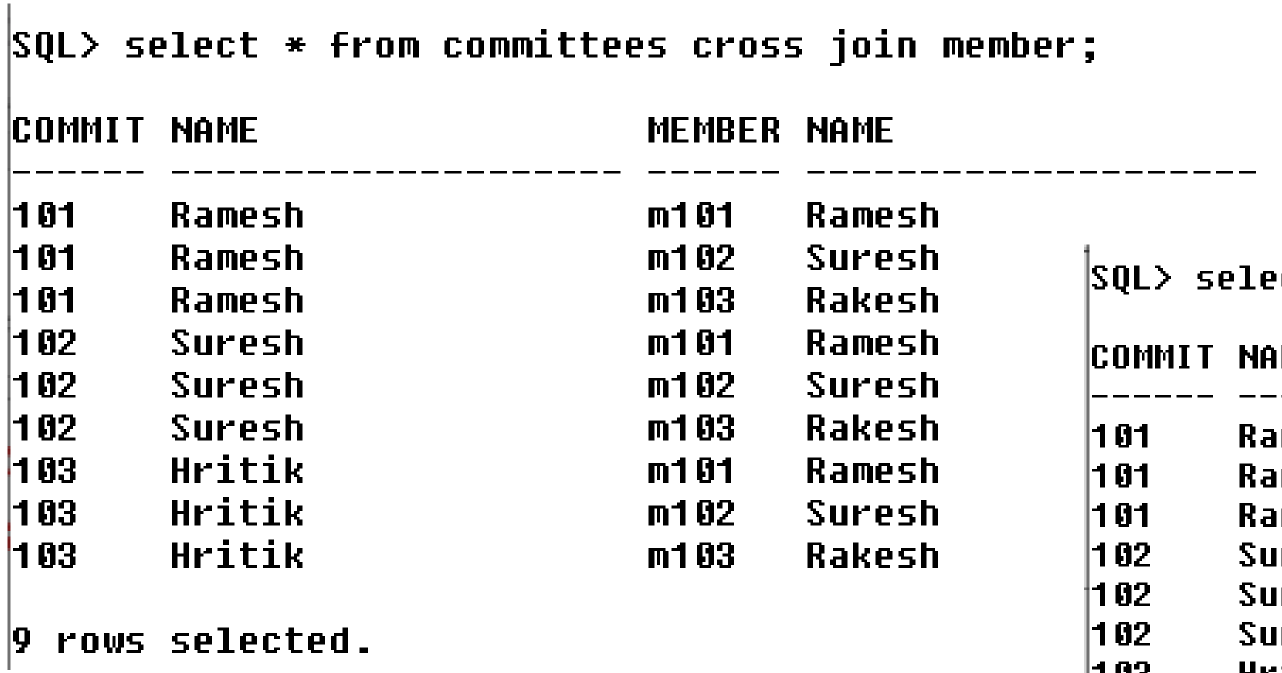
CROSS JOIN

Cartesian product of rows from the joined tables (NO CONDITION).
Combines each row from the first table with every row from the right table.

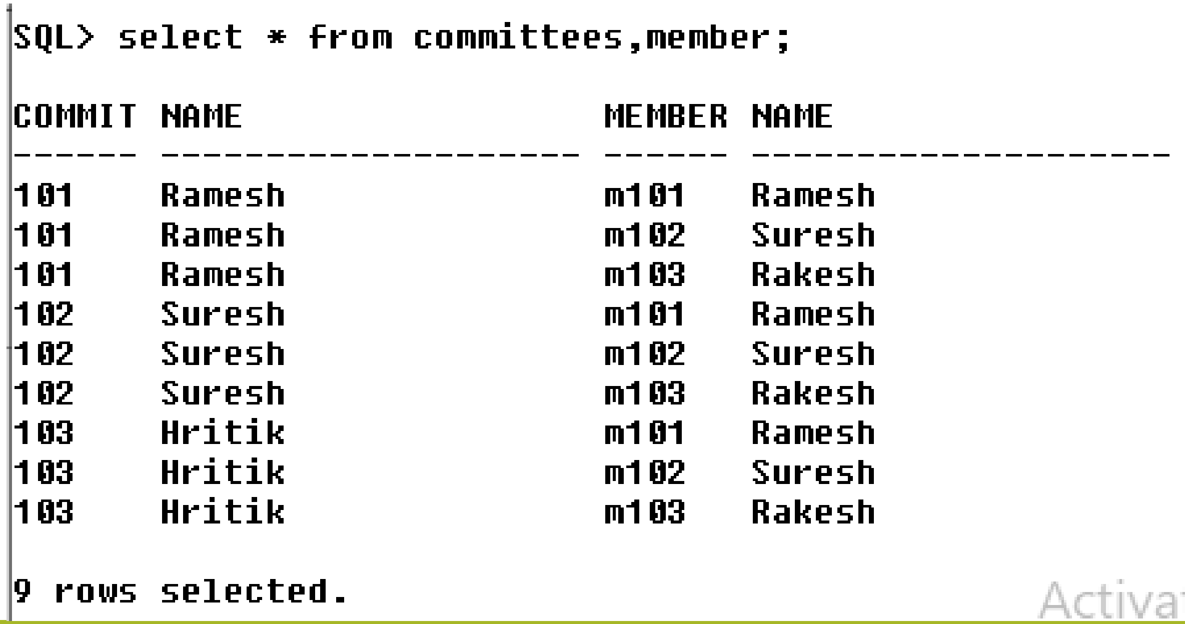
SELECT select_list FROM table_1 CROSS JOIN table_2;

select select_list from table1,table2;





```
SQL> select * from committees cross join member;

COMMIT NAME                           MEMBER NAME
------ ------------------------------ ------ ------------------------------
101    Ramesh                         m101   Ramesh
101    Ramesh                         m102   Suresh
101    Ramesh                         m103   Rakesh
102    Suresh                         m101   Ramesh
102    Suresh                         m102   Suresh
102    Suresh                         m103   Rakesh
103    Hritik                         m101   Ramesh
103    Hritik                         m102   Suresh
103    Hritik                         m103   Rakesh

9 rows selected.
```



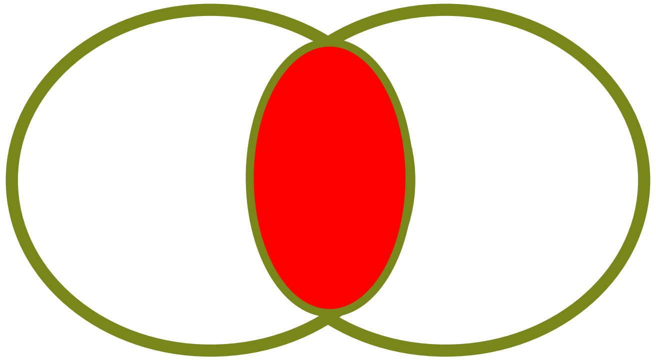
```
SQL> select * from committees,member;

COMMIT NAME                           MEMBER NAME
------ ------------------------------ ------ ------------------------------
101    Ramesh                         m101   Ramesh
101    Ramesh                         m102   Suresh
101    Ramesh                         m103   Rakesh
102    Suresh                         m101   Ramesh
102    Suresh                         m102   Suresh
102    Suresh                         m103   Rakesh
103    Hritik                         m101   Ramesh
103    Hritik                         m102   Suresh
103    Hritik                         m103   Rakesh

9 rows selected.
```

INNER JOIN/Simple join

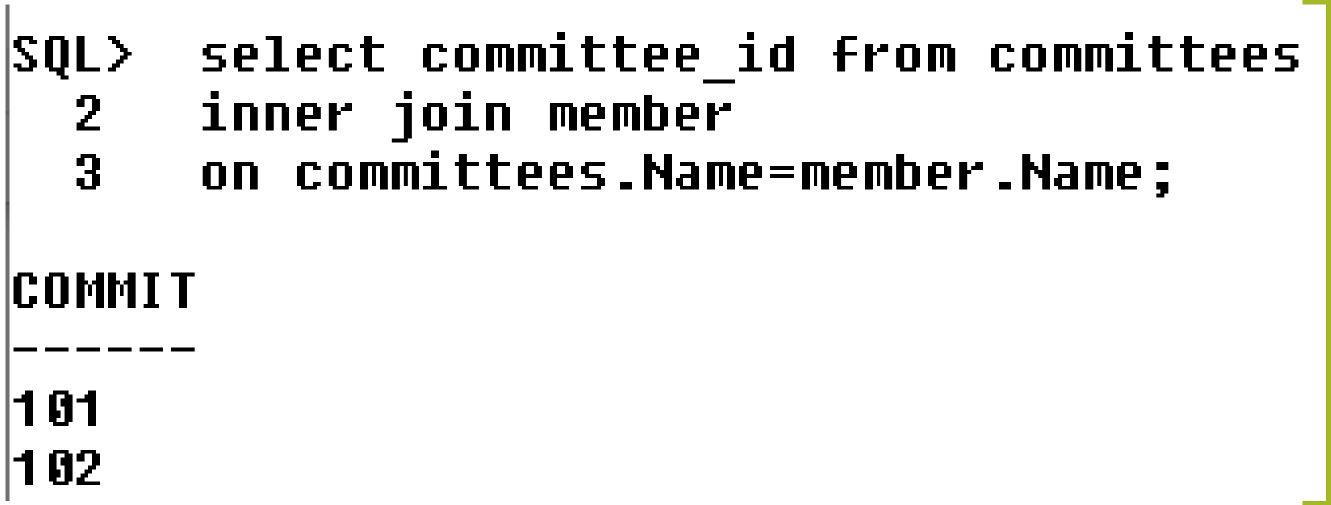
SELECT column_list **FROM** table_1 **INNER JOIN** table_2 **ON** join_condition;

compares each row from the first table with every row from the second table

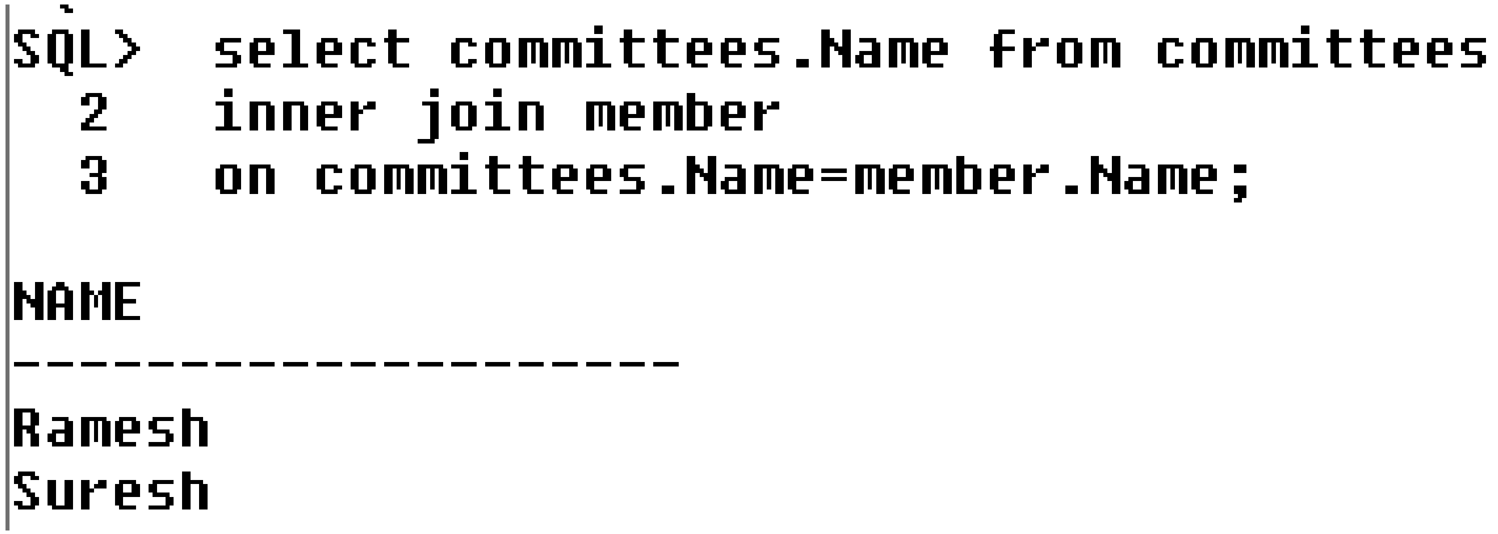
- If values in both rows cause the join condition evaluates to true,
- the inner join clause creates a new row whose column
- contains all columns of the two rows from both tables and include this new row in the final result set.



```
SELECT columns
FROM table1
INNER JOIN table2

ON table1.column = table2.column;
```



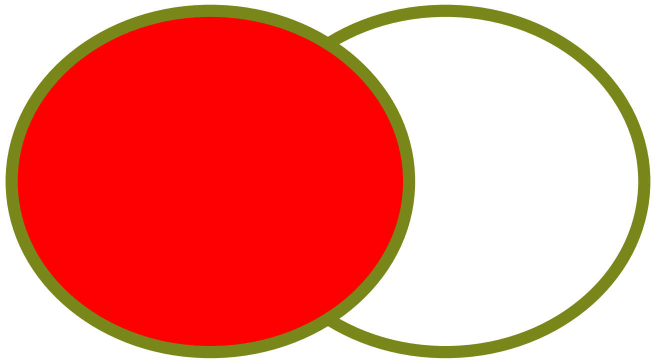
```
SQL>  select committee_id from committees
  2   inner join member
  3   on committees.Name=member.Name;

COMMIT
------
101
102
```



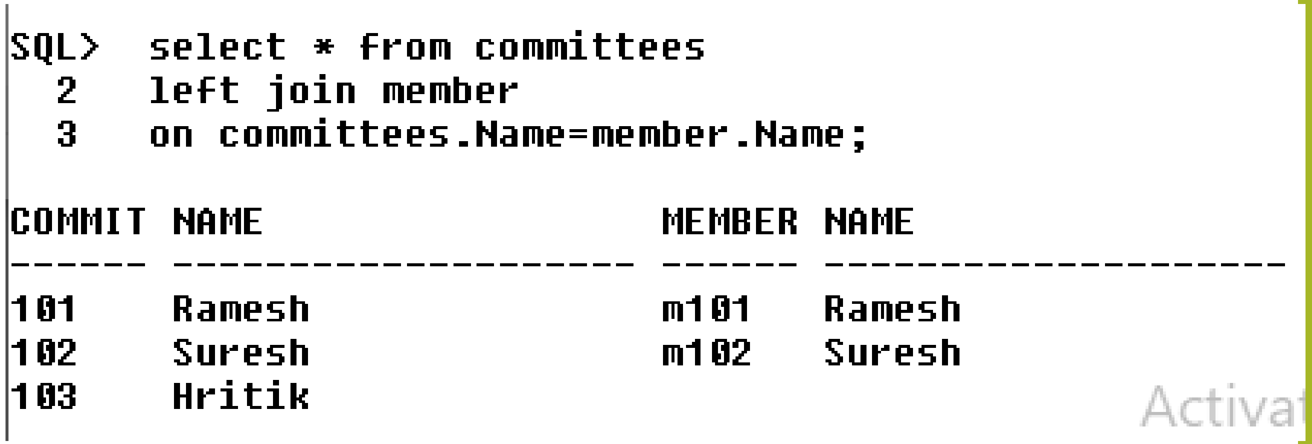
```
SQL>  select committees.Name from committees
  2   inner join member
  3   on committees.Name=member.Name;

NAME
--------------------
Ramesh
Suresh
```

Left Join

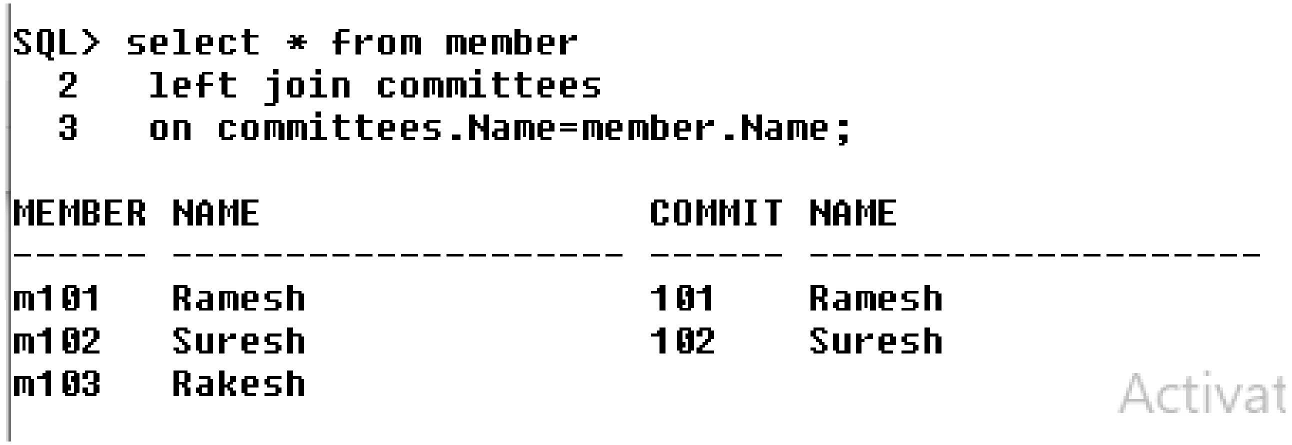
- selects data starting from the left table
- For each row in the left table,
  - the left join compares with every row in the right table
  - If the values in the two rows cause the join condition evaluates to true
  - the left join creates a new row whose columns contain all columns of the rows in both tables and includes this row in the result set.
  - In case there is no matching rows from the right table found, NULLs are used for columns of the row from the right table in the final result set



SELECT columns
FROM table1
LEFT [OUTER] JOIN table2
ON condition;

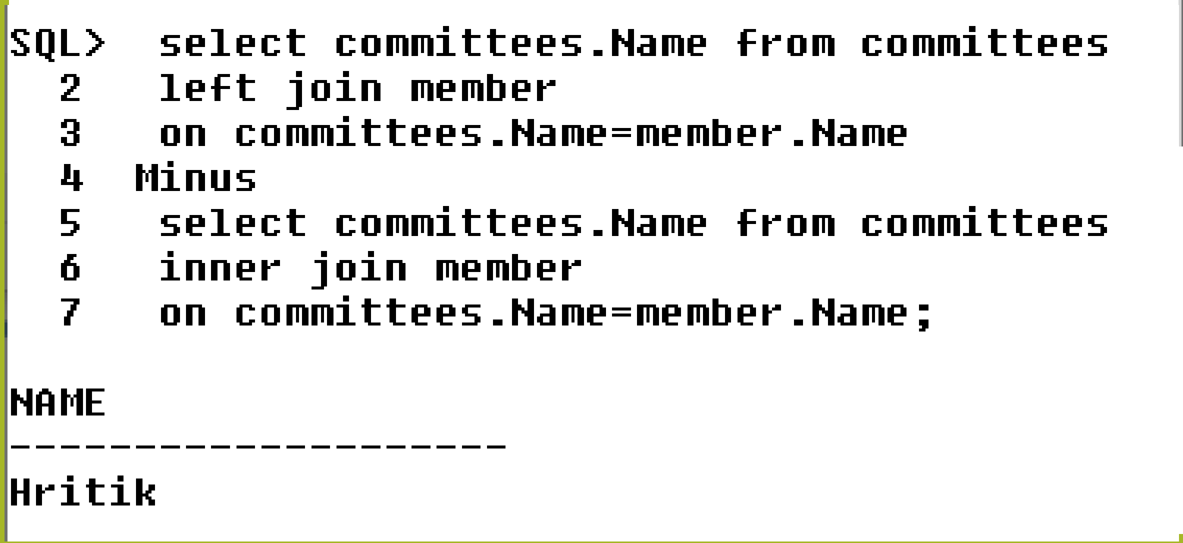


```
SQL>  select * from committees
  2   left join member
  3   on committees.Name=member.Name;

COMMIT NAME                 MEMBER NAME
------ -------------------- ------ --------------------
101    Ramesh               m101   Ramesh
102    Suresh               m102   Suresh
103    Hritik
```



```
SQL> select * from member
  2   left join committees
  3   on committees.Name=member.Name;

MEMBER NAME                 COMMIT NAME
------ -------------------- ------ --------------------
m101   Ramesh               101    Ramesh
m102   Suresh               102    Suresh
m103   Rakesh
```

Find Name of person in committee who is not a member

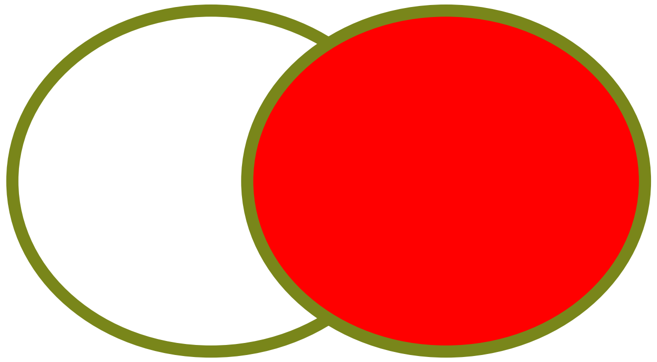


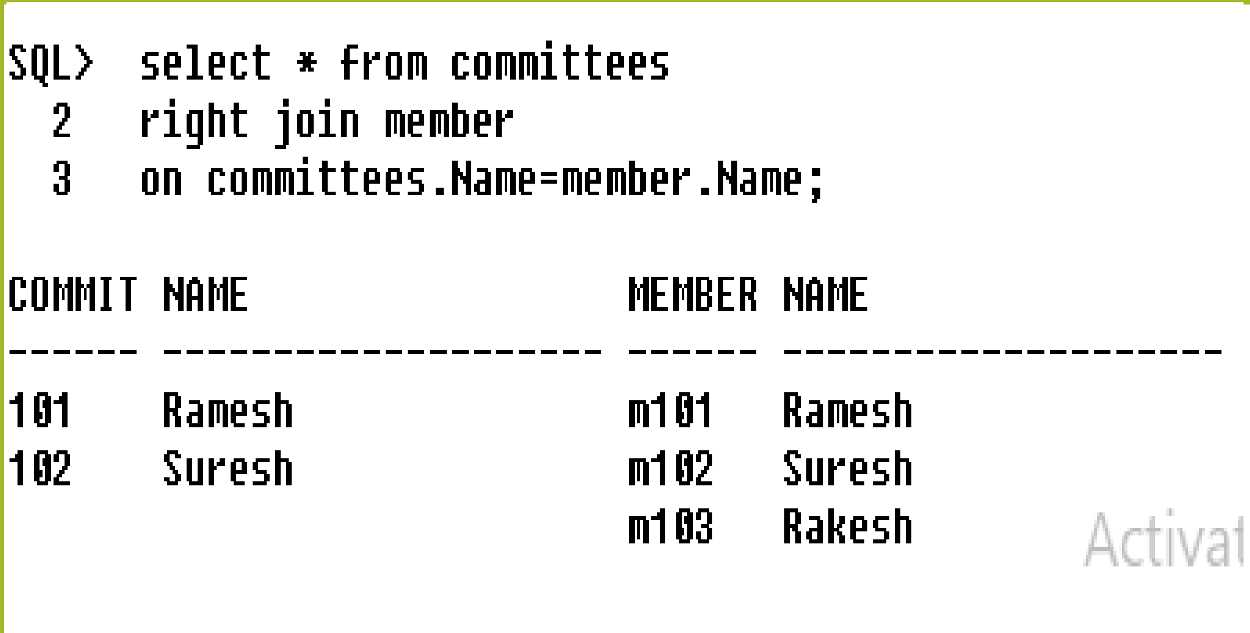
```
SQL>  select committees.Name from committees
  2   left join member
  3   on committees.Name=member.Name
  4  Minus
  5   select committees.Name from committees
  6   inner join member
  7   on committees.Name=member.Name;

NAME
--------------------
Hritik
```

Right Outer Join

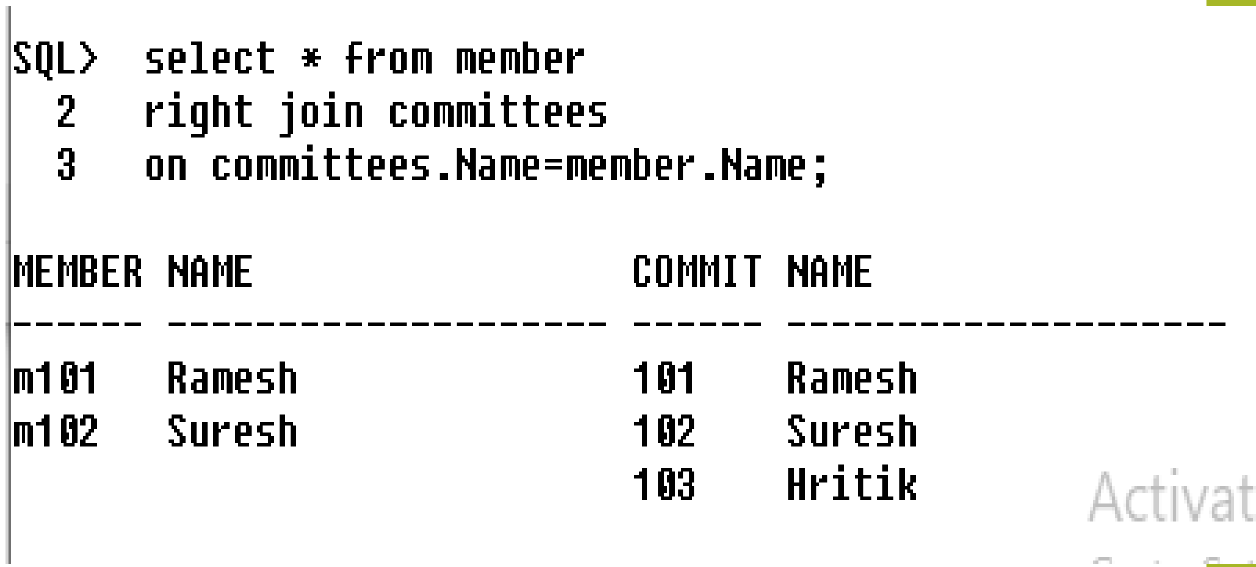
- returns all rows from the right-hand table specified in the ON condition
- and only those rows from the other table where the join condition is met.

```
SELECT columns
FROM table1
RIGHT [OUTER] JOIN table2
ON condition;
```



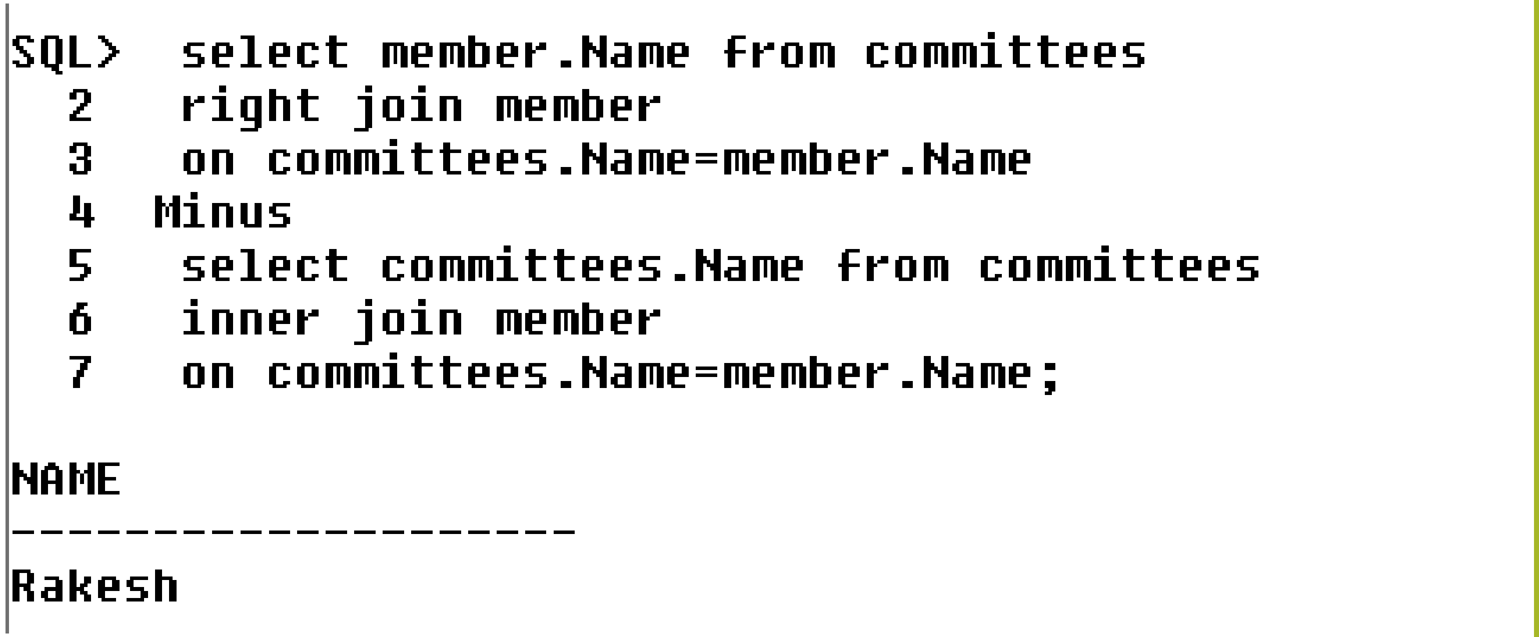


```
SQL>  select * from committees
  2   right join member
  3   on committees.Name=member.Name;

COMMIT NAME                 MEMBER NAME
------ -------------------- ------ --------------------
101    Ramesh               m101   Ramesh
102    Suresh               m102   Suresh
                            m103   Rakesh
```



```
SQL>  select * from member
  2   right join committees
  3   on committees.Name=member.Name;

MEMBER NAME                 COMMIT NAME
------ -------------------- ------ --------------------
m101   Ramesh               101    Ramesh
m102   Suresh               102    Suresh
                            103    Hritik
```

Find the name of member who is not in committee list



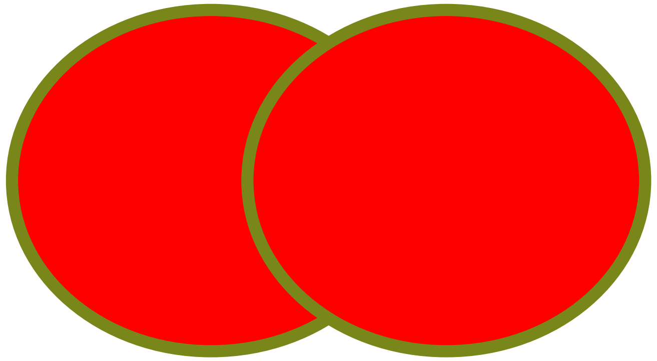
```
SQL>  select member.Name from committees
  2   right join member
  3   on committees.Name=member.Name
  4  Minus
  5   select committees.Name from committees
  6   inner join member
  7   on committees.Name=member.Name;

NAME
--------------------
Rakesh
```

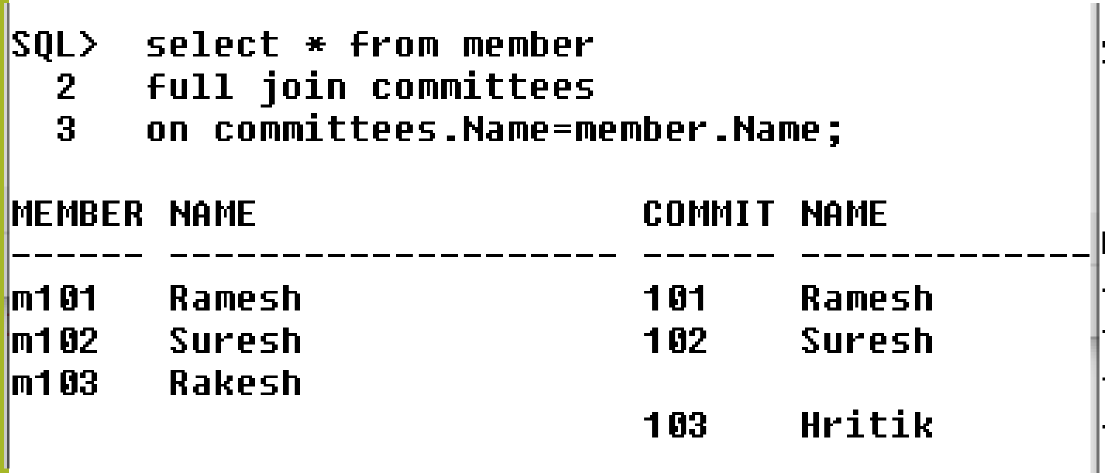
Full outer join

1. The Full Outer Join returns all rows from the left hand table and right hand tab

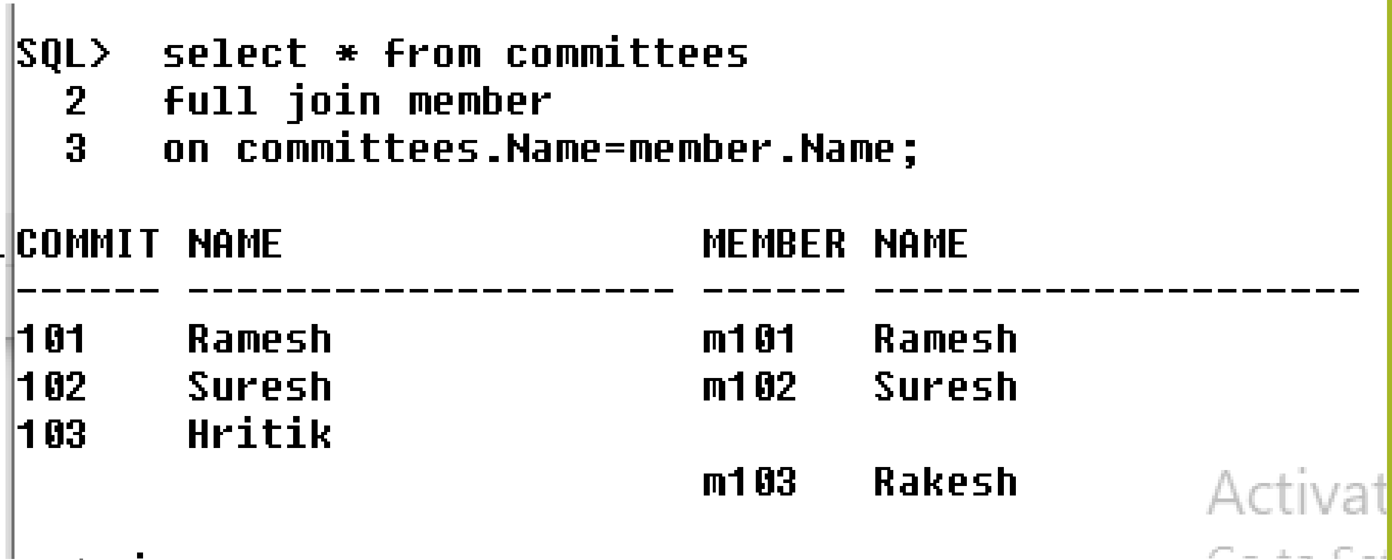
```
SELECT columns
FROM table1
FULL [OUTER] JOIN table2
ON condition;
```

It places NULL where the join condition is not met.

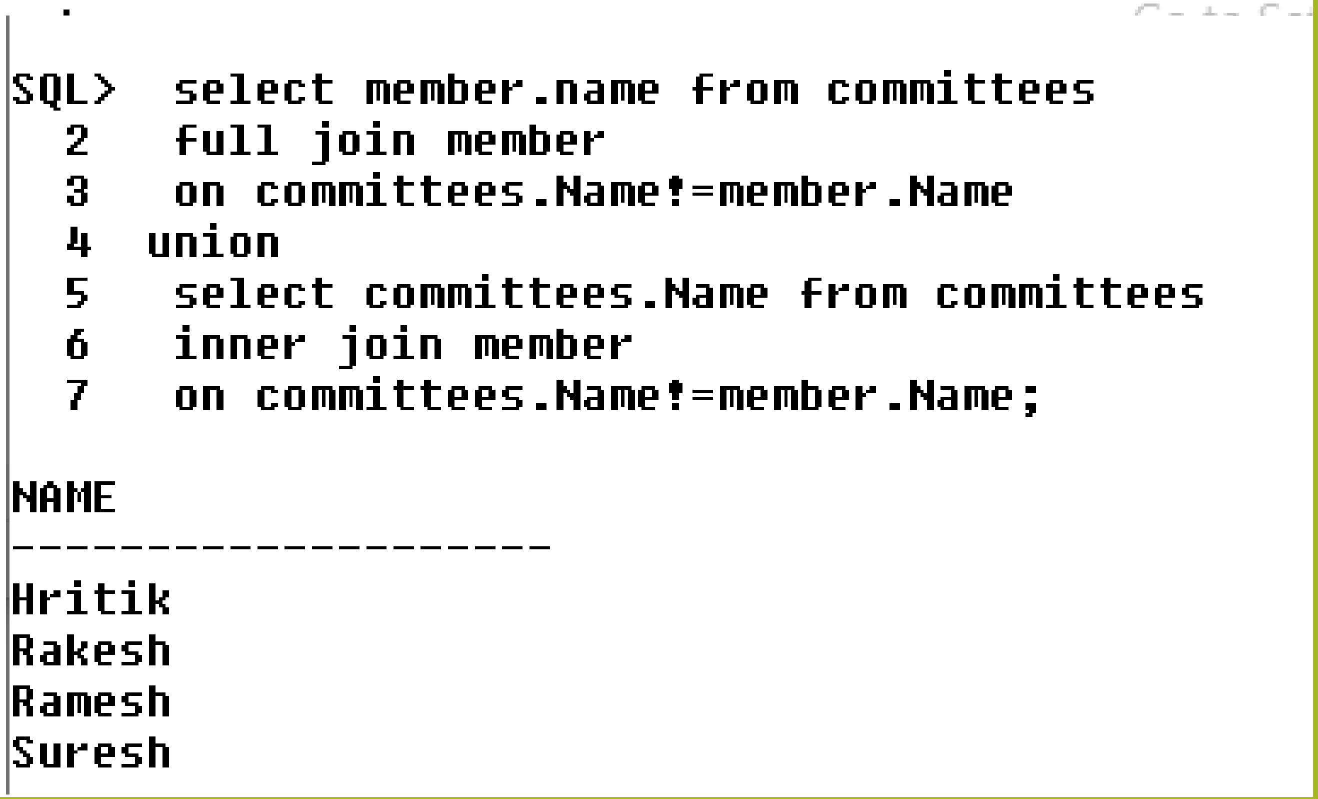




```
SQL>  select * from member
  2   full join committees
  3   on committees.Name=member.Name;

MEMBER NAME                  COMMIT NAME
------ -------------------- ------ -------------
m101   Ramesh               101    Ramesh
m102   Suresh               102    Suresh
m103   Rakesh
                            103    Hritik
```



```
SQL>  select * from committees
  2   full join member
  3   on committees.Name=member.Name;

COMMIT NAME                 MEMBER NAME
------ -------------------- ------ --------------------
101    Ramesh               m101   Ramesh
102    Suresh               m102   Suresh
103    Hritik
                            m103   Rakesh
```



```
SQL>  select member.name from committees
  2   full join member
  3   on committees.Name!=member.Name
  4  union
  5   select committees.Name from committees
  6   inner join member
  7   on committees.Name!=member.Name;

NAME
--------------------
Hritik
Rakesh
Ramesh
Suresh
```

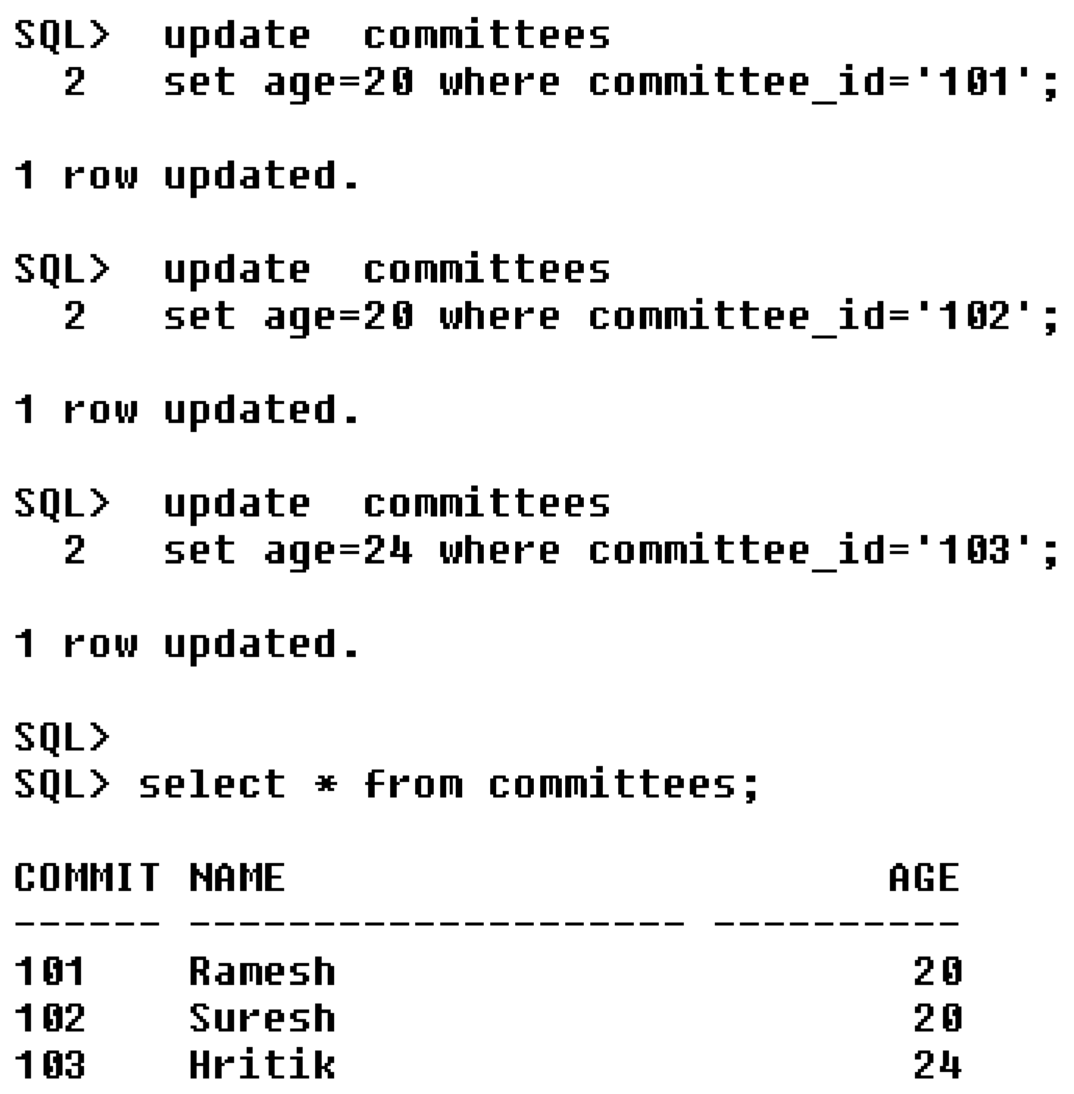
Find the name of persion who is either in committee or in member list.

Self Join

- Self Join is a specific type of Join.
- In Self Join, a table is joined with itself (Unary relationship).
- A self join simply specifies that each rows of a table is combined with itself and every other row of the table.

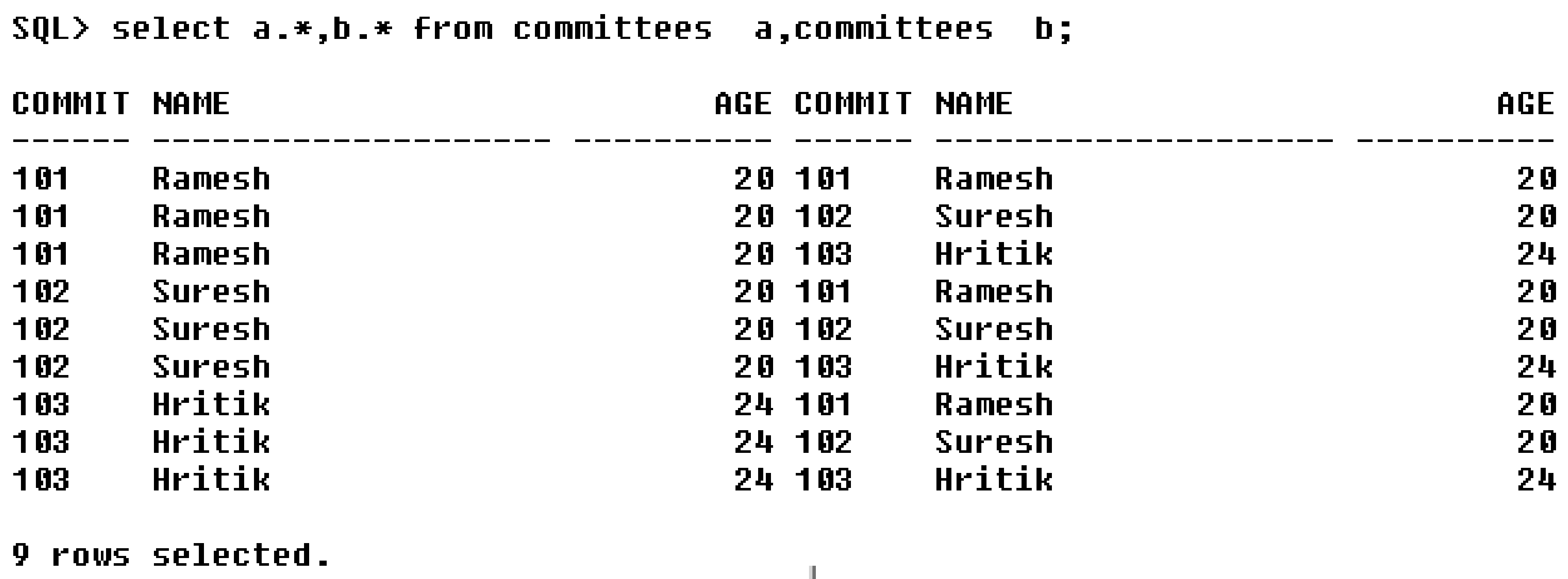
- alter table committees

  add (age number);

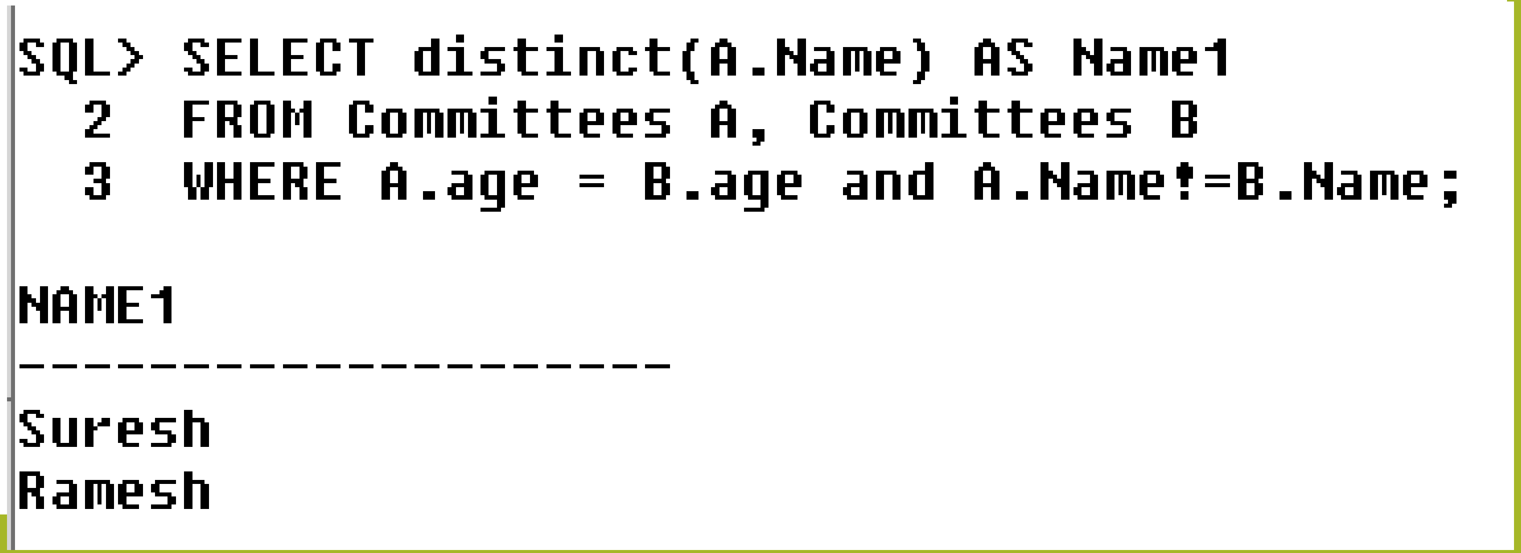


```
SQL>  update  committees
  2   set age=20 where committee_id='101';

1 row updated.

SQL>  update  committees
  2   set age=20 where committee_id='102';

1 row updated.

SQL>  update  committees
  2   set age=24 where committee_id='103';

1 row updated.

SQL>
SQL> select * from committees;

COMMIT NAME                        AGE
------ -------------------- ----------
101    Ramesh                       20
102    Suresh                       20
103    Hritik                       24
```

Find the name of person in committees having same age



```
SQL> select a.*,b.* from committees  a,committees  b;

COMMIT NAME                        AGE COMMIT NAME                        AGE
------ -------------------- ---------- ------ -------------------- ----------
101    Ramesh                       20 101    Ramesh                       20
101    Ramesh                       20 102    Suresh                       20
101    Ramesh                       20 103    Hritik                       24
102    Suresh                       20 101    Ramesh                       20
102    Suresh                       20 102    Suresh                       20
102    Suresh                       20 103    Hritik                       24
103    Hritik                       24 101    Ramesh                       20
103    Hritik                       24 102    Suresh                       20
103    Hritik                       24 103    Hritik                       24

9 rows selected.
```



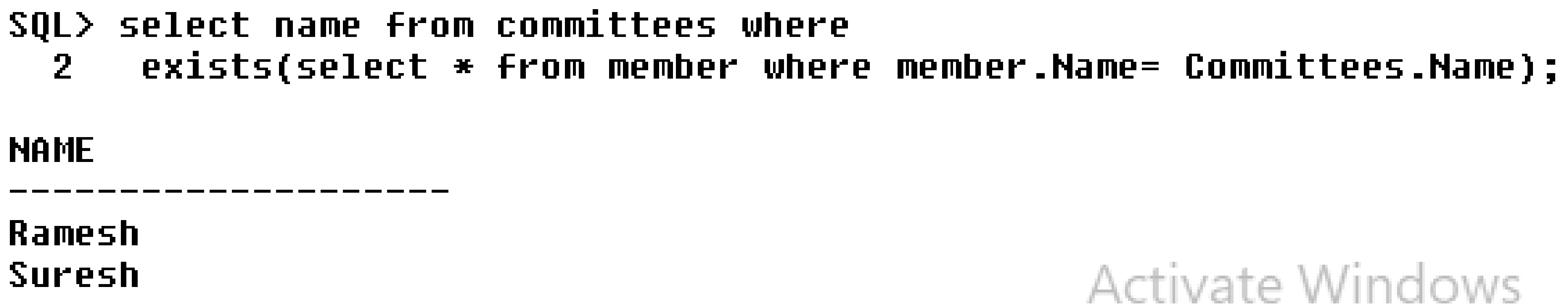
```
SQL> SELECT distinct(A.Name) AS Name1
  2  FROM Committees A, Committees B
  3  WHERE A.age = B.age and A.Name!=B.Name;

NAME1
--------------------
Suresh
Ramesh
```

Equi and Non Equi

- EQUI JOIN creates a JOIN for equality or matching column(s) values of the relative tables.
- EQUI JOIN also create JOIN by using JOIN with ON and then providing the names of the columns with their relative tables to check equality using equal sign (=).
- **NON EQUI JOIN :**
- NON EQUI JOIN performs a JOIN using comparison operator other than equal(=) sign like >, <, >=, <= with conditions.

Semi

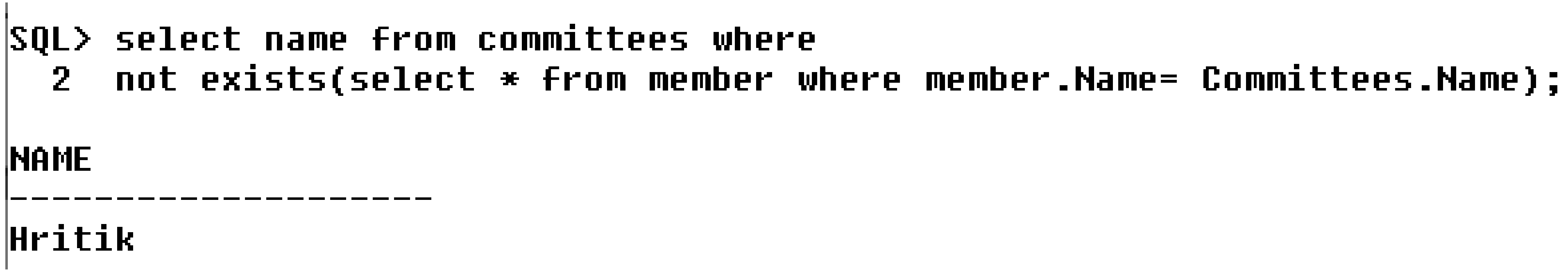
- Semi-join is introduced in Oracle 8.o.
- It provides an efficient method of performing a WHERE EXISTS sub-query.
- A semi-join returns one copy of each row in first table for which at least one match is found.
- Semi-joins are written using the EXISTS construct.



```
SQL> select name from committees where
  2   exists(select * from member where member.Name= Committees.Name);

NAME
--------------------
Ramesh
Suresh
```

Anti

- Anti-join is used to make the queries run faster. It is a very powerful SQL construct Oracle offers for faster queries.
- Anti-join between two tables returns rows from the first table where no matches are found in the second table. It is opposite of a semi-join. An anti-join returns one copy of each row in the first table for which no match is found.
- Anti-joins are written using the NOT EXISTS or NOT IN constructs.



```
SQL> select name from committees where
  2  not exists(select * from member where member.Name= Committees.Name);

NAME
--------------------
Hritik
```