JOINS

JOINS

- A SQL Join statement is used to combine data or rows from two or more tables based on a common field between them.
- **JOIN** Keyword is used in SQL queries for joining two or more tables.
- Minimum required condition for joining table, is (n-1) where n, is number of tables.
- A table can also join to itself, which is known as, Self Join.

TYPES OF JOINS

- Inner Join
  - Equi Join
  - Non Equi Join
- Outer Join
  - Left-outer Join
  - Right-outer Join
  - Full-outer Join
- Cross Join
- Self Join
- Natural Join

EASY SHOP DATABASE
Relation: PRODUCT

P_ID	P_NAME	BRAND	PRICE
100	Camera	Nikon	8900
101	Television	Samsung	24500
102	Refrigerator	LG	56000
103	Laptop	Sony	75400
104	Mobile	HTC	62900
106	Television	LG	32860

EASY SHOP DATABASE RELATION - ORDERS

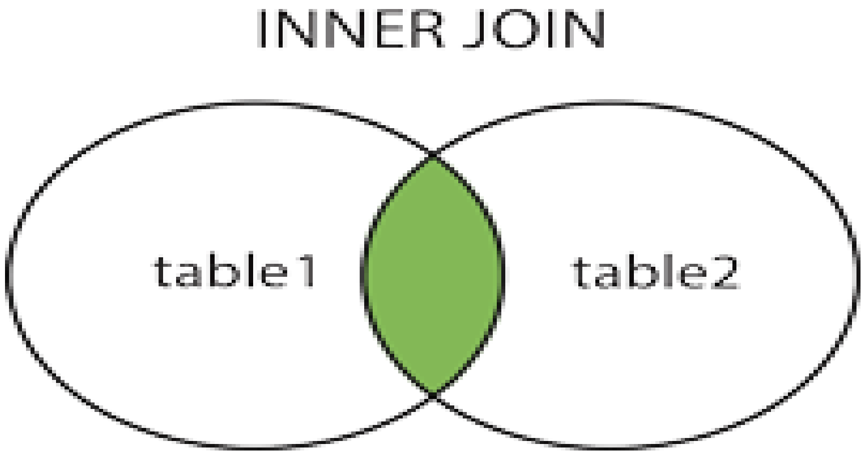
ORDER_ID	P_ID	CUST_NAME
6001	103	Girias
6002	102	Sharptronics
6003	101	BEA
6004	105	Sharptronics
6005	104	Girias
6006	106	MGB

INNER JOIN

Returns records that have matching values in both tables.

Syntax:

SELECT column-name-list FROM table_1
INNER JOIN table_2
ON table_1.common_column=table_2.common_column;



Note: We can simply use **JOIN** instead of INNER JOIN, both are same.

INNER JOIN - EQUI JOIN

- An equijoin is such a join which performs against a join condition containing an equality operator.
- It combines rows of one table associated with one or more rows in another table based on the equality of column values or expressions.

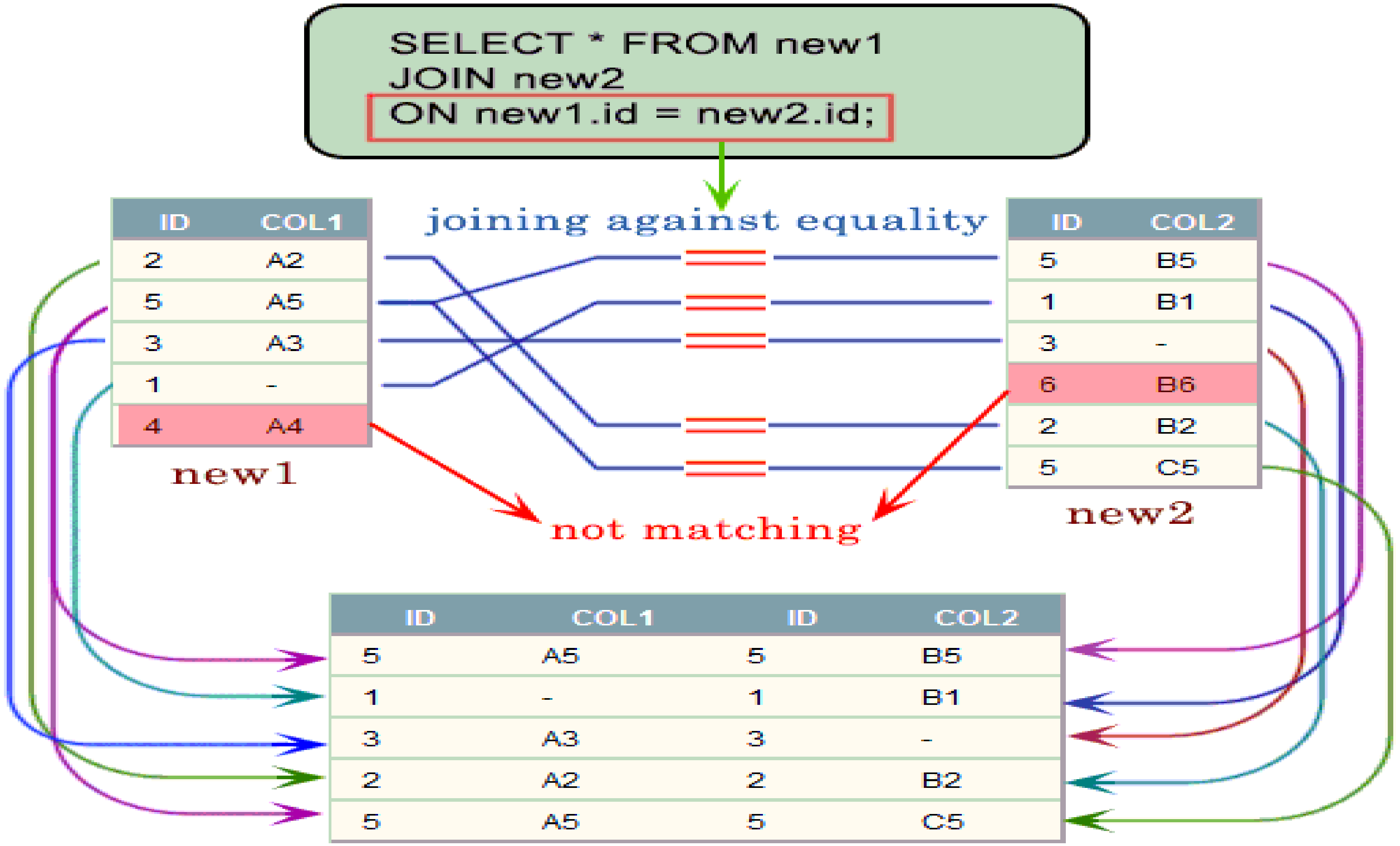
Syntax:

SELECT column-name-list FROM table_1 **JOIN** table_2 **ON** table_1.common_column=table_2.common_column;

(OR)

SELECT column-name-list FROM table_1,table_2 **WHERE** table_1.common_column=table_2.common_column;

INNER JOIN - EQUI JOIN - EXAMPLE



```
SELECT * FROM new1
JOIN new2
ON new1.id = new2.id;
```

new1

ID	COL1
2	A2
5	A5
3	A3
1	-
4	A4

new2

ID	COL2
5	B5
1	B1
3	-
6	B6
2	B2
5	C5

ID	COL1	ID	COL2
5	A5	5	B5
1	-	1	B1
3	A3	3	-
2	A2	2	B2
5	A5	5	C5

Here the **ON** clause is based on the equality condition “ = ”. Hence it is called equi join.

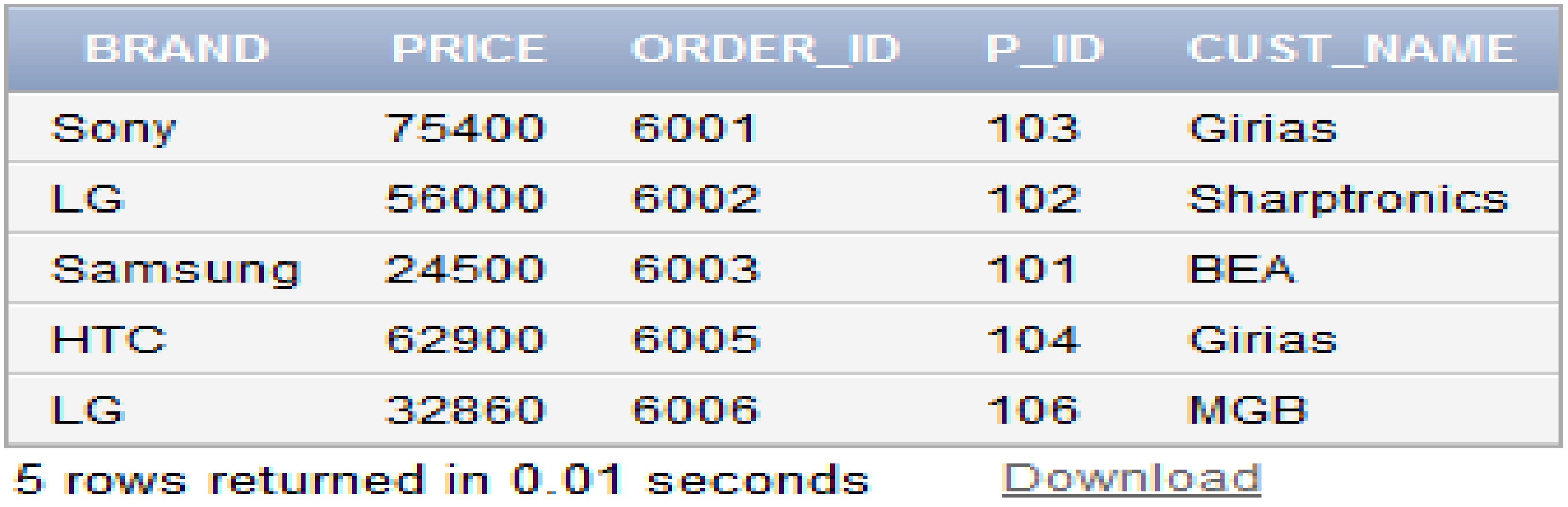
INNER JOIN - EQUI JOIN - EXAMPLE

The Manager of the Easy Shop wants to know the orders placed by his customers with brand name and price.

SELECT Product.Brand, Product.Price, Orders.*

FROM Product, Orders

WHERE Product.P_id=Orders.P_id;



BRAND	PRICE	ORDER_ID	P_ID	CUST_NAME
Sony	75400	6001	103	Girias
LG	56000	6002	102	Sharptronics
Samsung	24500	6003	101	BEA
HTC	62900	6005	104	Girias
LG	32860	6006	106	MGB

5 rows returned in 0.01 seconds   Download

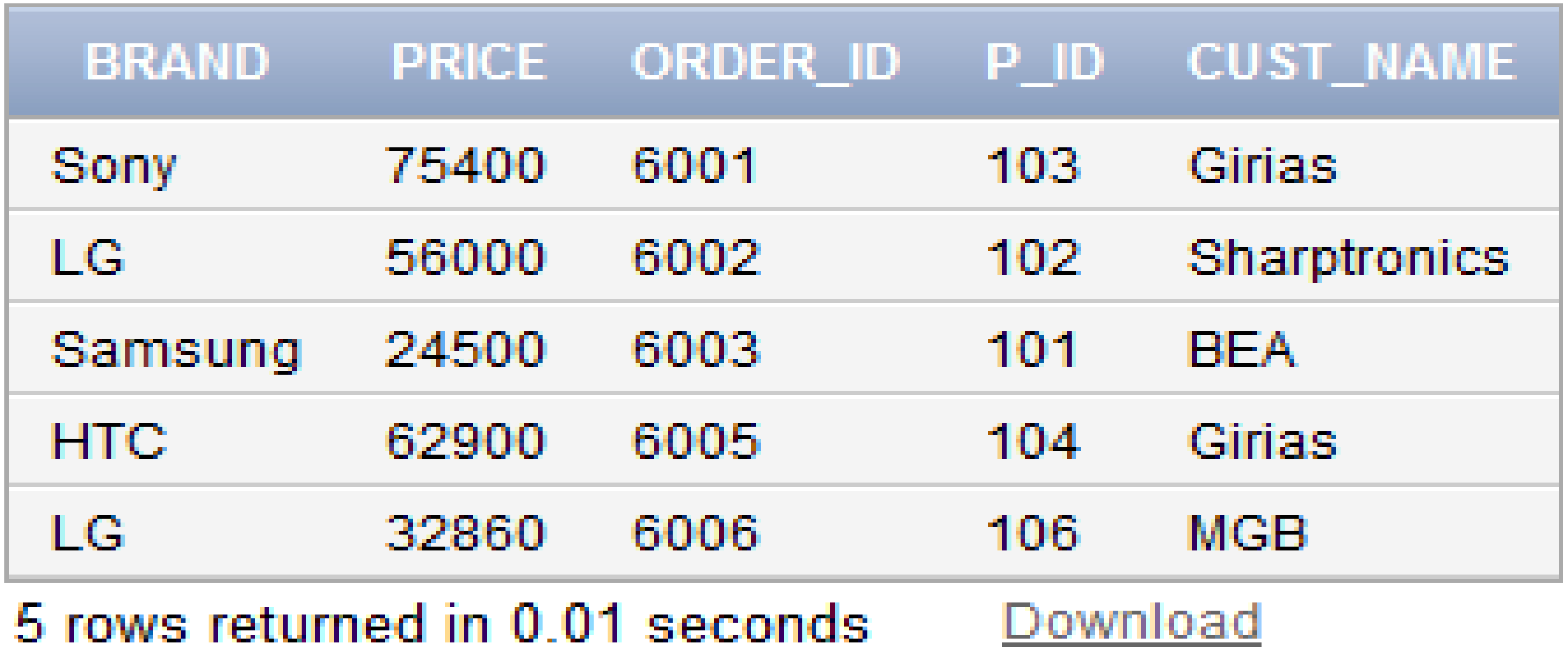
JOINS – ON CLAUSE

- ON clause can be used to join columns that have different names.
- Use the ON clause to specify conditions or specify columns to join.
- The join condition is separated from other search conditions.
- This is the most easiest and widely used form of the join clauses.

INNER JOIN - EQUI JOIN - EXAMPLE

USING JOIN KEYWORD:

```
SELECT Product.Brand, Product.Price, Orders.*
FROM Product JOIN Orders
ON Product.P_id=Orders.P_id;
```



BRAND	PRICE	ORDER_ID	P_ID	CUST_NAME
Sony	75400	6001	103	Girias
LG	56000	6002	102	Sharptronics
Samsung	24500	6003	101	BEA
HTC	62900	6005	104	Girias
LG	32860	6006	106	MGB

5 rows returned in 0.01 seconds   Download

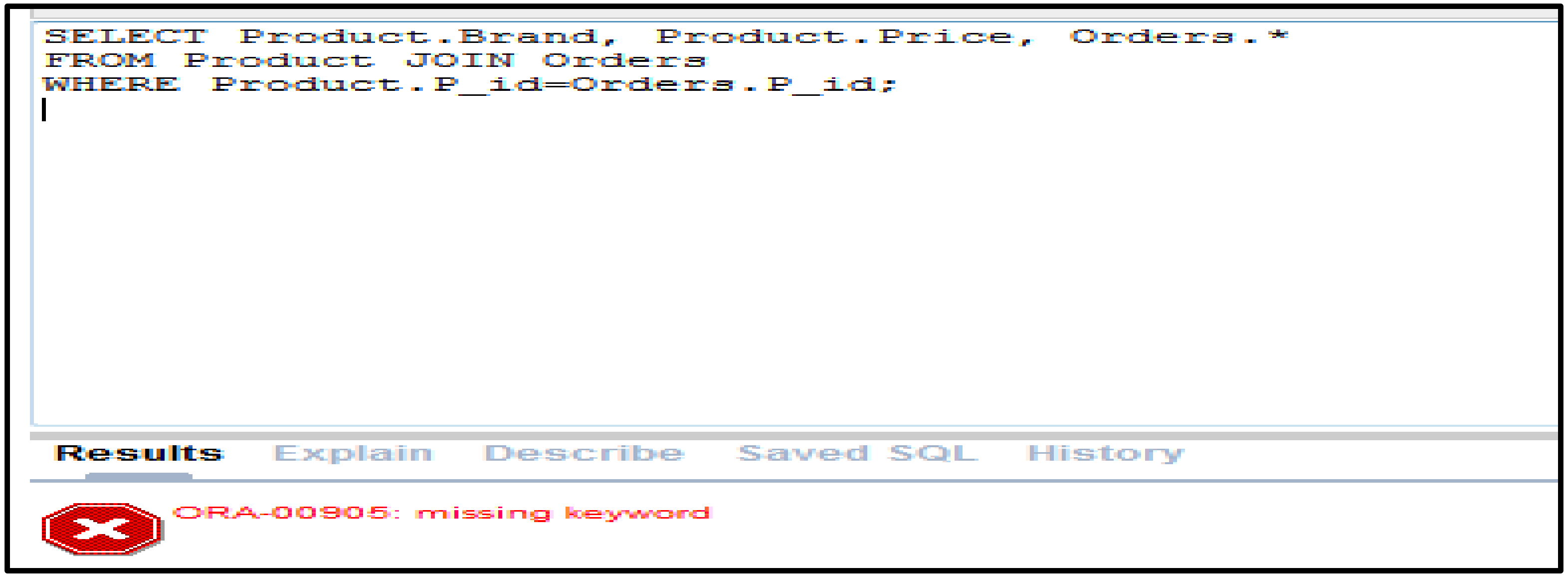
INNER JOIN - EQUI JOIN - EXAMPLE

USING JOIN KEYWORD:

SELECT Product.Brand, Product.Price, Orders.*

FROM Product JOIN Orders

WHERE Product.P_id=Orders.P_id;



```
SELECT Product.Brand, Product.Price, Orders.*
FROM Product JOIN Orders
WHERE Product.P_id=Orders.P_id;
```

```
ORA-00905: missing keyword
```

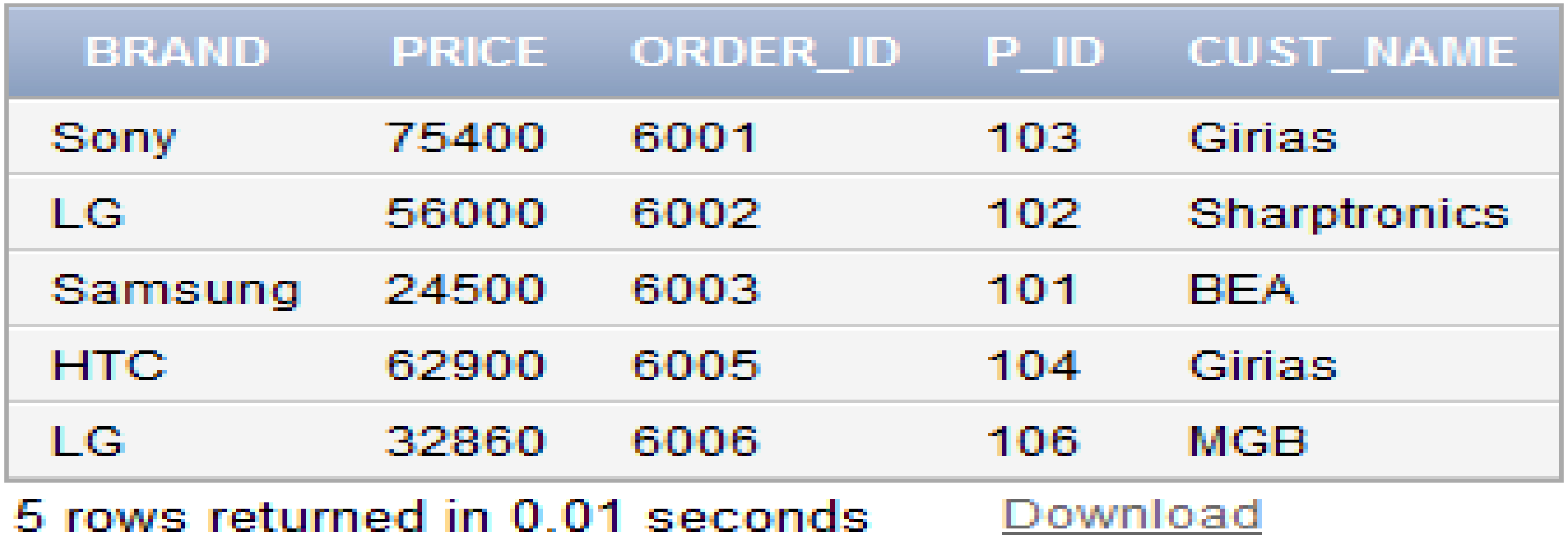
<u>Note:</u>

When Using JOIN keyword in Query we have to use ON clause or USING clause otherwise it throws error.

INNER JOIN - EQUI JOIN - EXAMPLE

USING Alias Name:

```
SELECT P.Brand, P.Price, O.*
FROM Product P JOIN Orders O
ON P.P_id=O.P_id;
```



BRAND	PRICE	ORDER_ID	P_ID	CUST_NAME
Sony	75400	6001	103	Girias
LG	56000	6002	102	Sharptronics
Samsung	24500	6003	101	BEA
HTC	62900	6005	104	Girias
LG	32860	6006	106	MGB

5 rows returned in 0.01 seconds Download

JOINS – USING CLAUSE

- USING Clause is used to match only one column when more than one column matches.
- It should not have a qualifier(table name or Alias) in the referenced columns.

Syntax:

SELECT Column_Names

FROM Table_1 **JOIN** Table_2

USING (Common_Column);

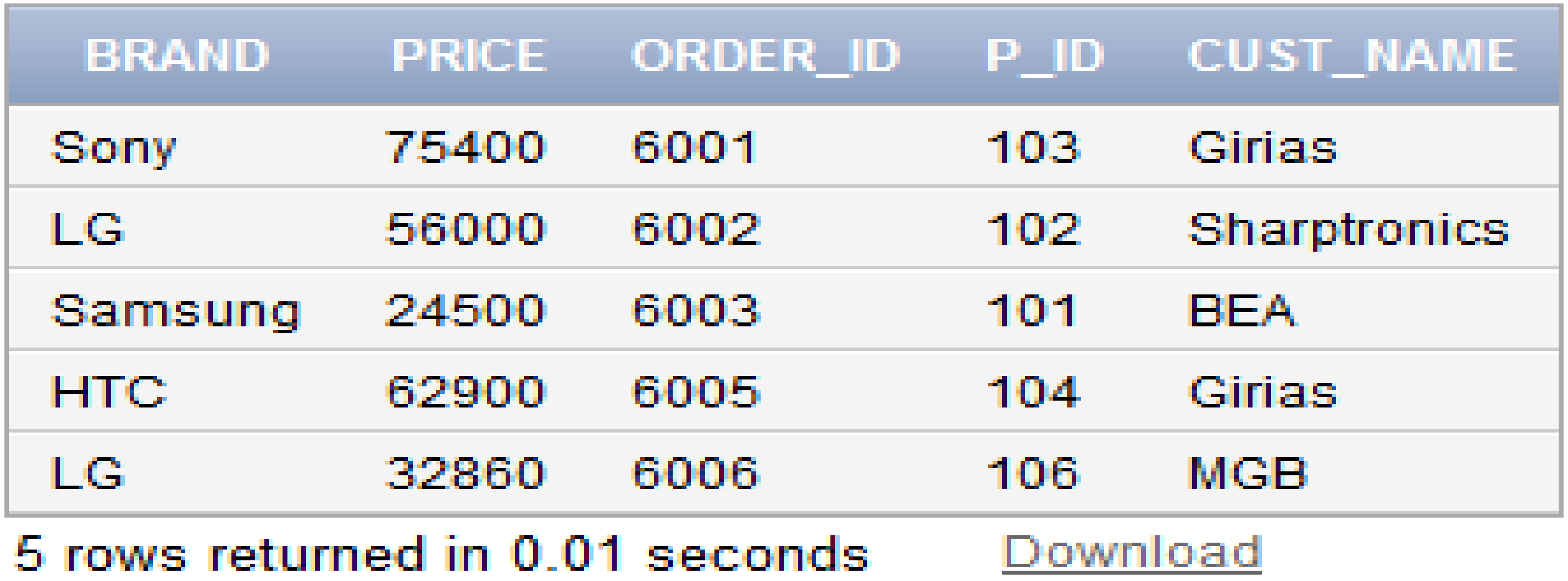
INNER JOIN - EQUI JOIN - EXAMPLE

Using ‘USING’ Keyword:

SELECT Brand,Price,Order_ID,P_ID, CUST_Name

FROM Product JOIN Orders

USING (P_id);



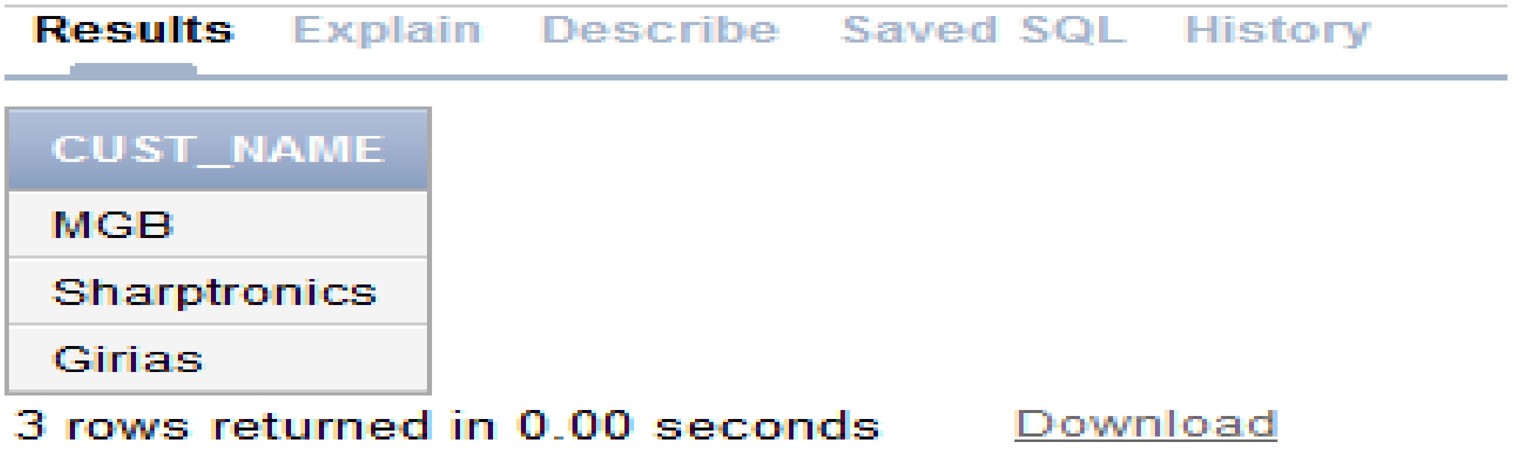
BRAND	PRICE	ORDER_ID	P_ID	CUST_NAME
Sony	75400	6001	103	Girias
LG	56000	6002	102	Sharptronics
Samsung	24500	6003	101	BEA
HTC	62900	6005	104	Girias
LG	32860	6006	106	MGB

5 rows returned in 0.01 seconds Download

INNER JOIN - EQUI JOIN - EXAMPLE

The management of the Easy Shop wants to know the name of the customers whose total bill amount is more than 25000.

Select O.Cust_Name
From Product P Join Orders O
ON P.P_ID = O.P_ID
Group by O.Cust_Name
Having SUM(P.Price) > 25000;



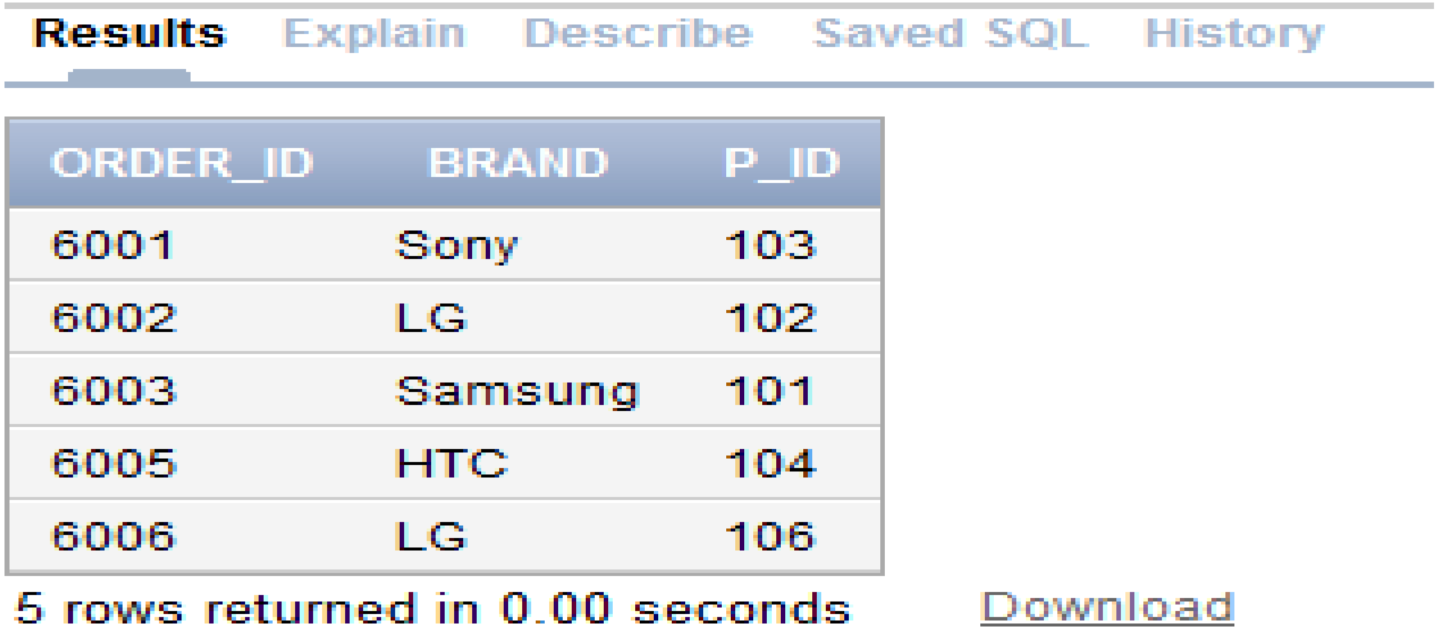
CUST_NAME
MGB
Sharptronics
Girias

3 rows returned in 0.00 seconds

INNER JOIN - EQUI JOIN - EXAMPLE

Write SQL query to find the order_id of the products. Also give their respective P_ID and Brand.

Select Order_ID, Brand, P_ID
From Product Join Orders
USING (P_ID);



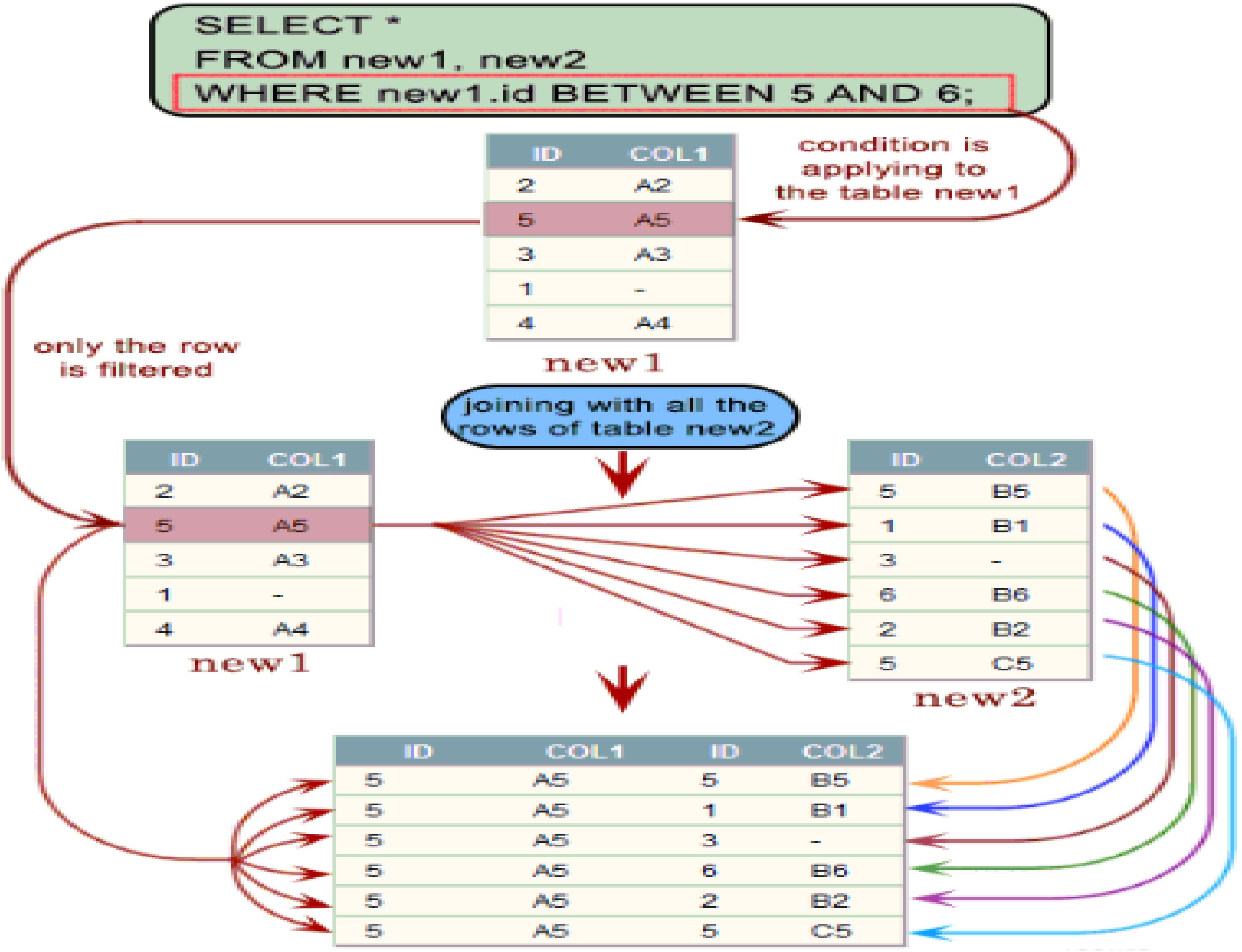
ORDER_ID	BRAND	P_ID
6001	Sony	103
6002	LG	102
6003	Samsung	101
6005	HTC	104
6006	LG	106

5 rows returned in 0.00 seconds

INNER JOIN - NON-EQUI JOIN

- The nonequijoins is such a join which match column values from different tables based on an inequality (instead of the equal sign like >, <, >=, <=) expression.
- The value of the join column in each row in the source table is compared to the corresponding values in the target table.
- A match is found if the expression based on an inequality operator used in the join, evaluates to true.
- Simply, If the WHERE or ON clause is based on a non quality condition (<, >, >=, <=, …). It is called non-equi join.

INNER JOIN – NON EQUI JOIN - EXAMPLE



```
SELECT *
FROM new1, new2
WHERE new1.id BETWEEN 5 AND 6;
```

new1

ID	COL1
2	A2
5	A5
3	A3
1	-
4	A4

new1

ID	COL1
2	A2
5	A5
3	A3
1	-
4	A4

new2

ID	COL2
5	B5
1	B1
3	-
6	B6
2	B2
5	C5

ID	COL1	ID	COL2
5	A5	5	B5
5	A5	1	B1
5	A5	3	-
5	A5	6	B6
5	A5	2	B2
5	A5	5	C5

condition is applying to the table new1
only the row is filtered
joining with all the rows of table new2

INNER JOIN – NON EQUI JOIN - EXAMPLE

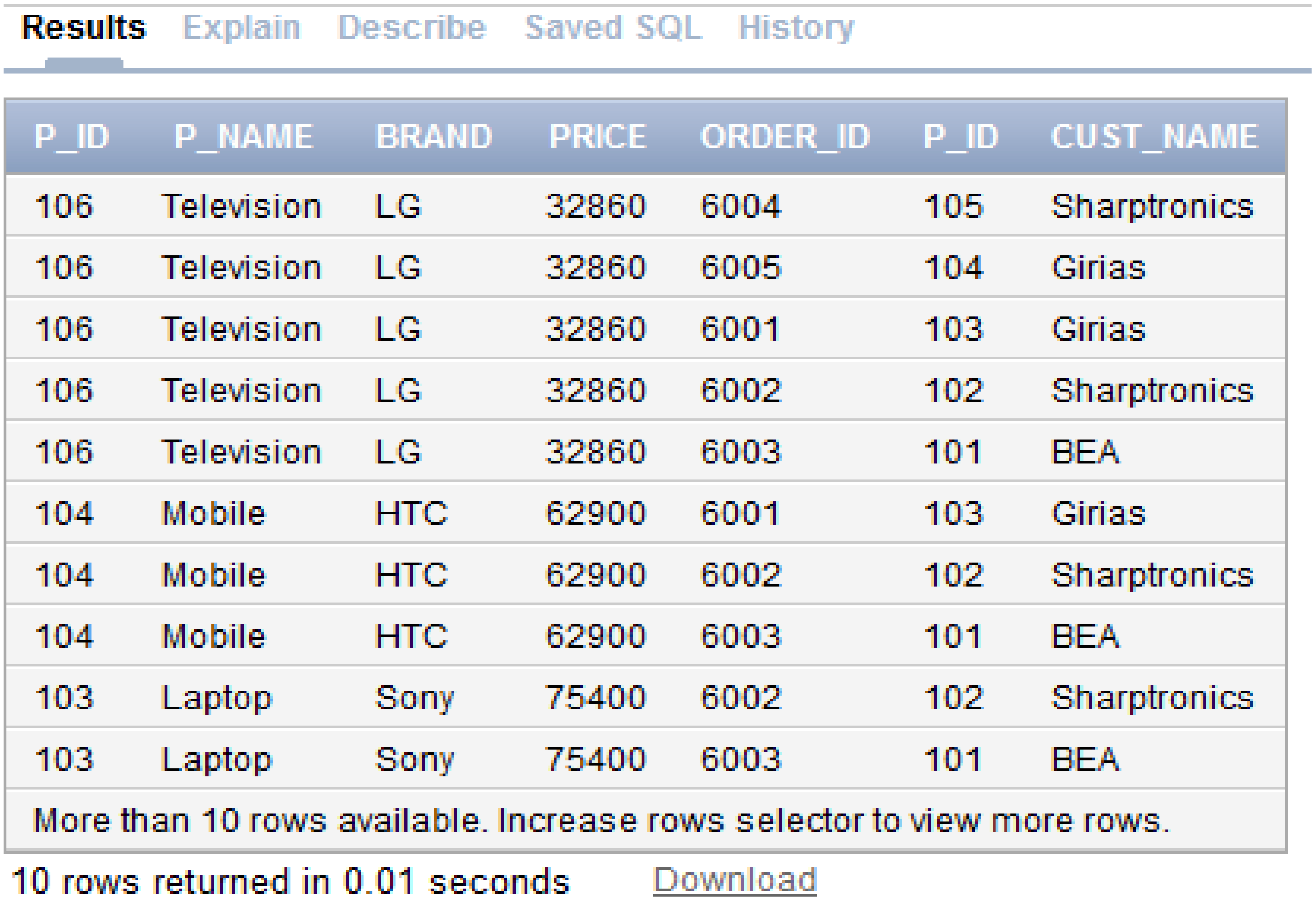
Display the details from Product and Orders tables, if Product ID in Product table is greater than Product ID in Orders table.

```
SELECT *
FROM Product P JOIN Orders O
ON P.P_ID > O.P_ID;
```

(OR)

```
SELECT *
FROM Product P, Orders O
WHERE P.P_ID > O.P_ID;
```

INNER JOIN – NON EQUI JOIN - EXAMPLE



P_ID	P_NAME	BRAND	PRICE	ORDER_ID	P_ID	CUST_NAME
106	Television	LG	32860	6004	105	Sharptronics
106	Television	LG	32860	6005	104	Girias
106	Television	LG	32860	6001	103	Girias
106	Television	LG	32860	6002	102	Sharptronics
106	Television	LG	32860	6003	101	BEA
104	Mobile	HTC	62900	6001	103	Girias
104	Mobile	HTC	62900	6002	102	Sharptronics
104	Mobile	HTC	62900	6003	101	BEA
103	Laptop	Sony	75400	6002	102	Sharptronics
103	Laptop	Sony	75400	6003	101	BEA

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

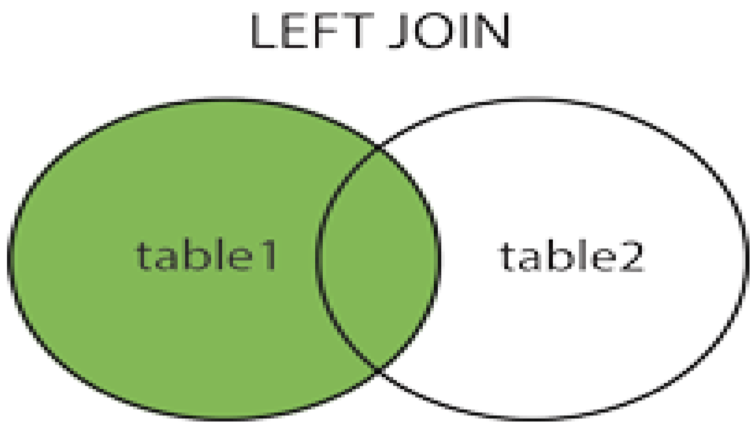
10 rows returned in 0.01 seconds Download

LEFT OUTER JOIN

- Returns all the rows of the table on the left side of the join and matching rows for the table on the right side of join.
- The rows for which there is no matching row on right side, the result-set will contain *null*.
- LEFT JOIN is also known as LEFT OUTER JOIN.

LEFT OUTER JOIN

SELECT column-name-list FROM table_1
LEFT JOIN table_2
ON
table_1.common_column=table_2.common_column;



Note: We can also use **LEFT OUTER JOIN** instead of LEFT JOIN, both are same.

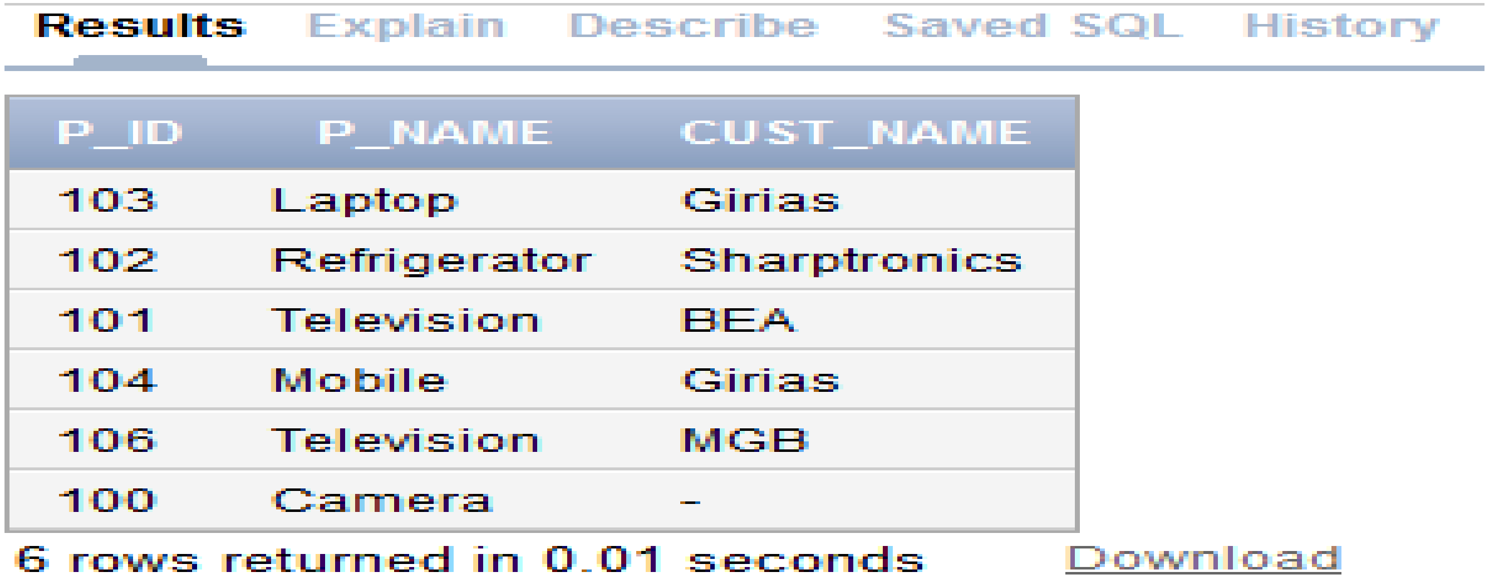
LEFT OUTER JOIN - EXAMPLE

List all the products along with the customer names who ordered it. The list should contain all the products even if it is not ordered by any customer.

SELECT P.P_ID, P.P_Name, O.Cust_Name

FROM Product P LEFT JOIN Orders O

ON P.P_id=O.P_id;



P_ID	P_NAME	CUST_NAME
103	Laptop	Girias
102	Refrigerator	Sharptronics
101	Television	BEA
104	Mobile	Girias
106	Television	MGB
100	Camera	-

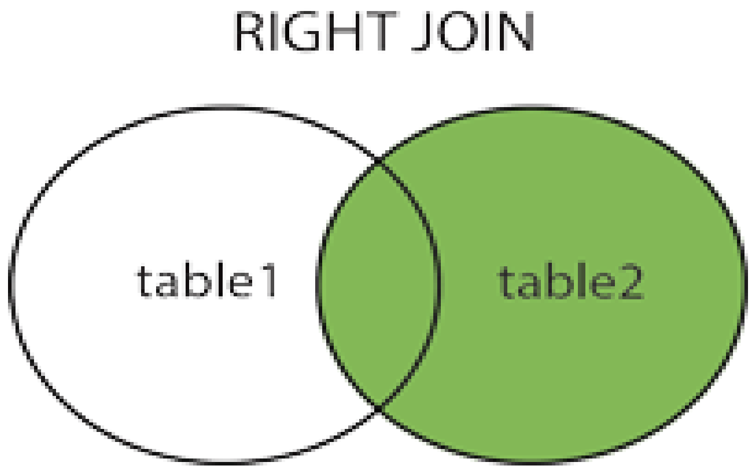
6 rows returned in 0.01 seconds

RIGHT OUTER JOIN

- RIGHT JOIN is similar to LEFT JOIN. This join returns all the rows of the table on the right side of the join and matching rows for the table on the left side of join.
- The rows for which there is no matching row on left side, the result-set will contain null.
- RIGHT JOIN is also known as RIGHT OUTER JOIN.

RIGHT OUTER JOIN

SELECT column-name-list FROM table_1
RIGHT JOIN table_2
ON
table_1.common_column=table_2.common_column;



Note: We can also use **RIGHT OUTER JOIN** instead of RIGHT JOIN, both are same.

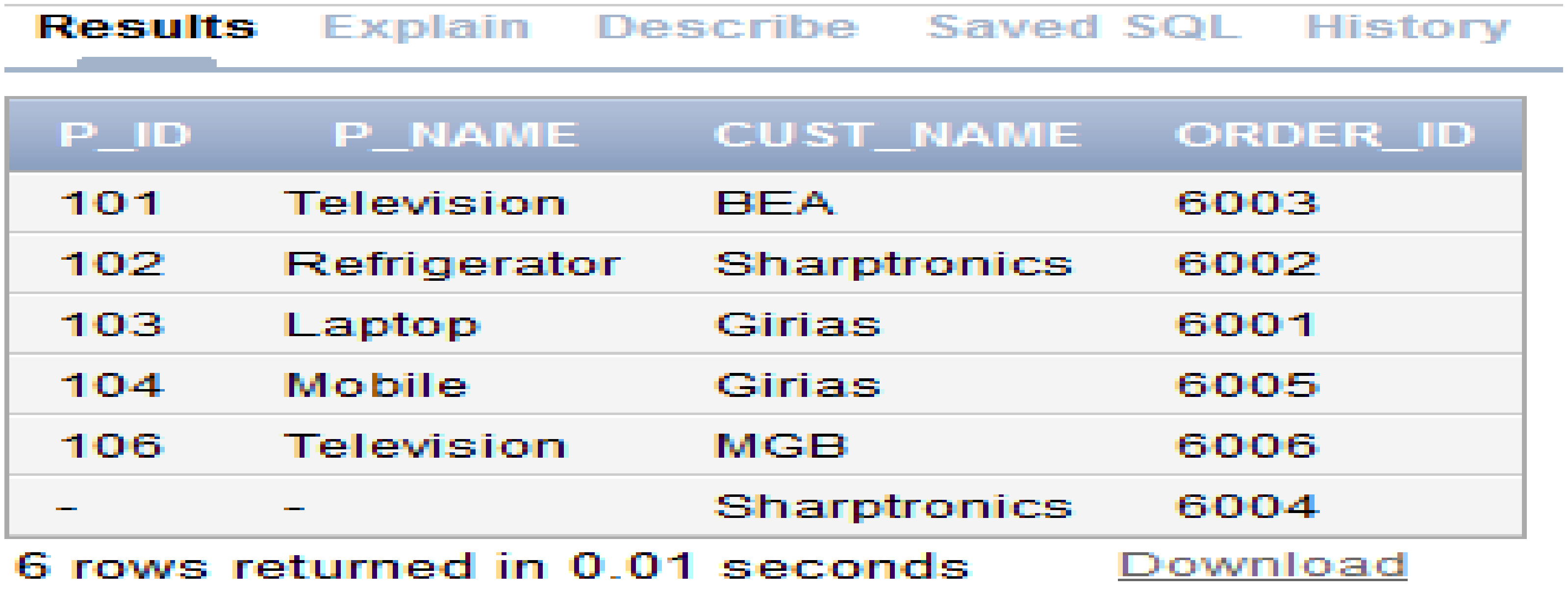
RIGHT OUTER JOIN - EXAMPLE

Identify the Product Name for every order. Display P_ID, P_Name, Order_ID and the name of the customers.

SELECT P.P_ID, P.P_Name, O.Cust_Name,O.Order_ID

FROM Product P RIGHT JOIN Orders O

ON P.P_id=O.P_id;



Results Explain Describe Saved SQL History

P_ID	P_NAME	CUST_NAME	ORDER_ID
101	Television	BEA	6003
102	Refrigerator	Sharptronics	6002
103	Laptop	Girias	6001
104	Mobile	Girias	6005
106	Television	MGB	6006
-	-	Sharptronics	6004

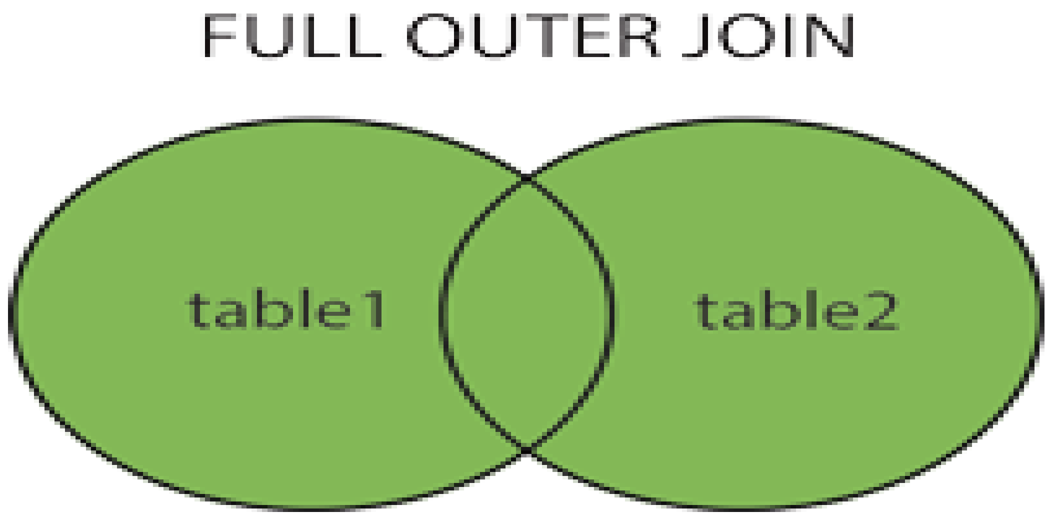
6 rows returned in 0.01 seconds Download

FULL OUTER JOIN

- FULL JOIN creates the result-set by combining result of both LEFT JOIN and RIGHT JOIN.
- The result-set will contain all the rows from both the tables. The rows for which there is no matching, the result-set will contain NULL values.

FULL OUTER JOIN

SELECT column-name-list FROM table_1
FULL JOIN table_2
ON table_1.common_column=table_2.common_column;

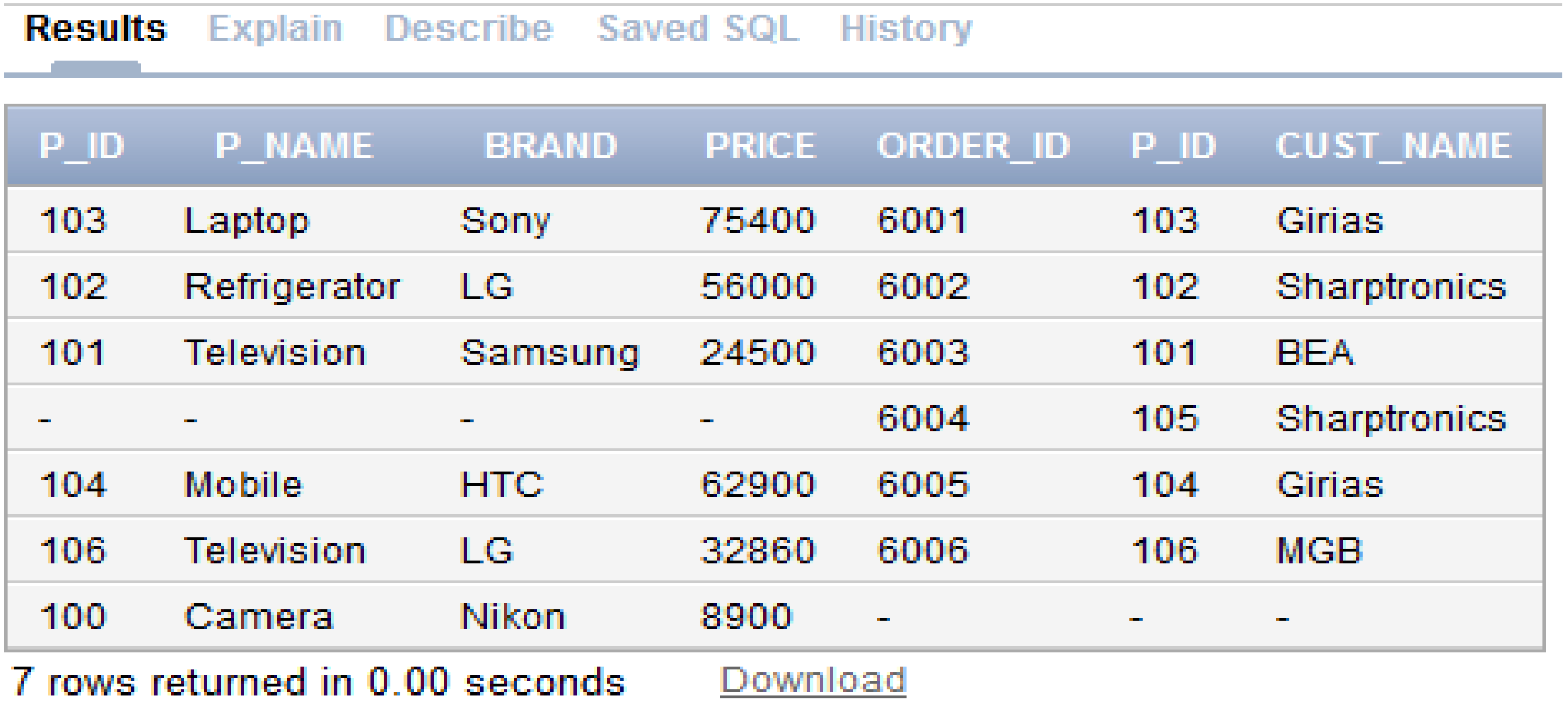


Note: We can also use **FULL OUTER JOIN** instead of FULL JOIN, both are same.

FULL OUTER JOIN - EXAMPLE

Retrieve all the rows from both the tables, even if there is no match.

SELECT *FROM Product P FULL JOIN Orders O ON P.P_id=O.P_id;



P_ID	P_NAME	BRAND	PRICE	ORDER_ID	P_ID	CUST_NAME
103	Laptop	Sony	75400	6001	103	Girias
102	Refrigerator	LG	56000	6002	102	Sharptronics
101	Television	Samsung	24500	6003	101	BEA
-	-	-	-	6004	105	Sharptronics
104	Mobile	HTC	62900	6005	104	Girias
106	Television	LG	32860	6006	106	MGB
100	Camera	Nikon	8900	-	-	-

7 rows returned in 0.00 seconds

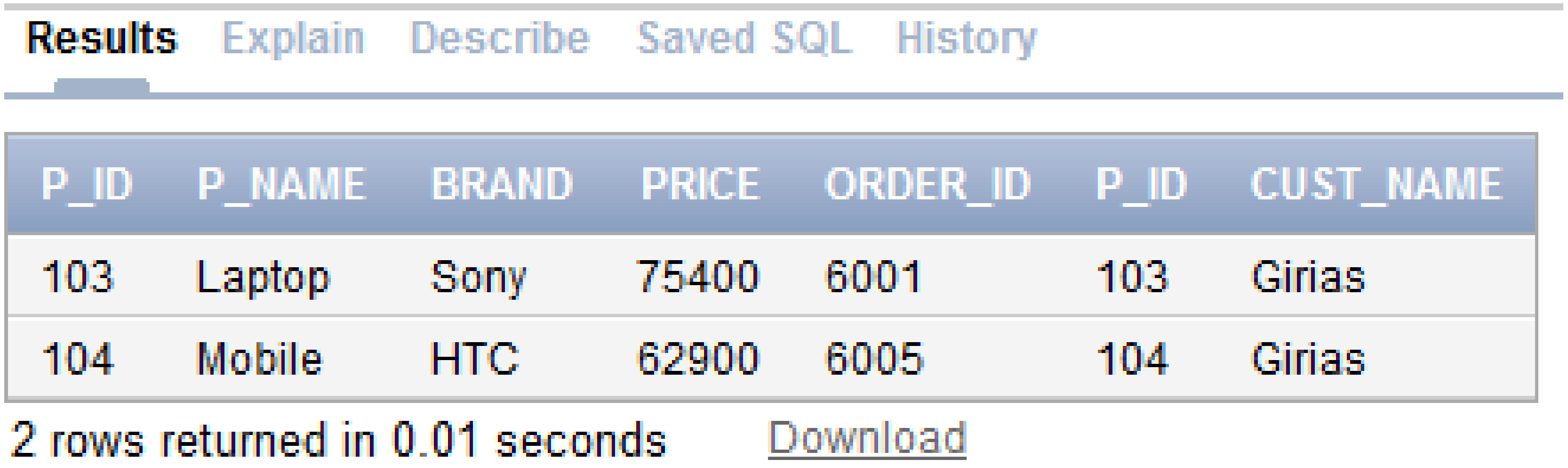
FULL OUTER JOIN - EXAMPLE

Retrieve all the rows from both the tables, even if there is no match. The result should filtered for those customers whose name starts with the letter 'G'.

SELECT *FROM Product P FULL JOIN Orders O

ON P.P_id=O.P_id

WHERE CUST_Name LIKE 'G%';



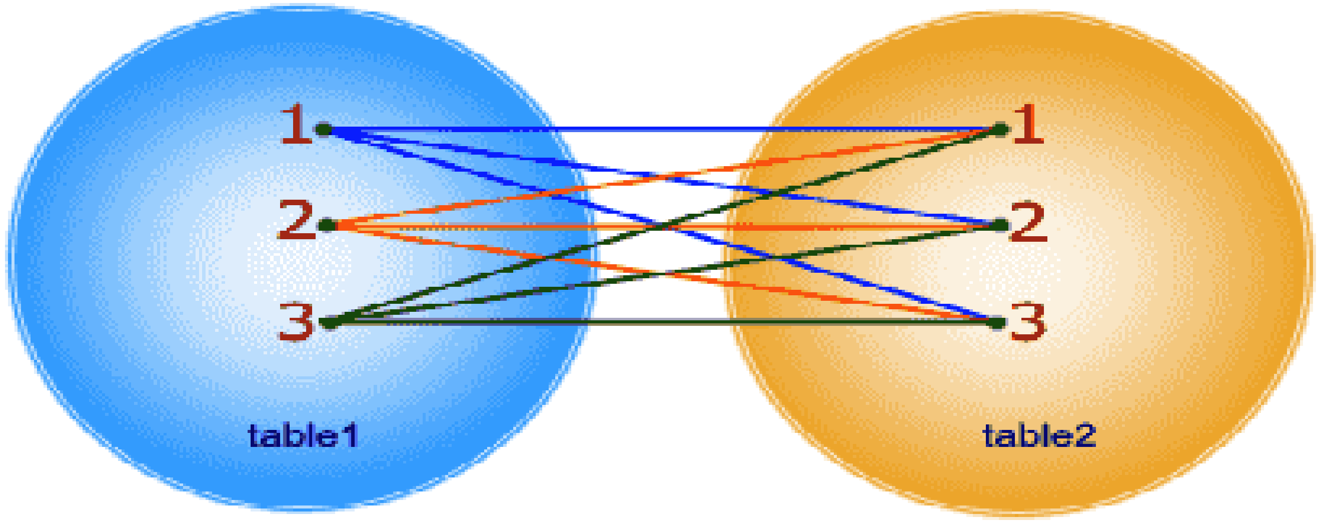
P_ID	P_NAME	BRAND	PRICE	ORDER_ID	P_ID	CUST_NAME
103	Laptop	Sony	75400	6001	103	Girias
104	Mobile	HTC	62900	6005	104	Girias

2 rows returned in 0.01 seconds

CROSS JOIN

- This type of JOIN returns the Cartesian product of rows from the tables in Join.
- It will return a table which consists of records which combines each row from the first table with each row of the second table.

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



CROSS JOIN

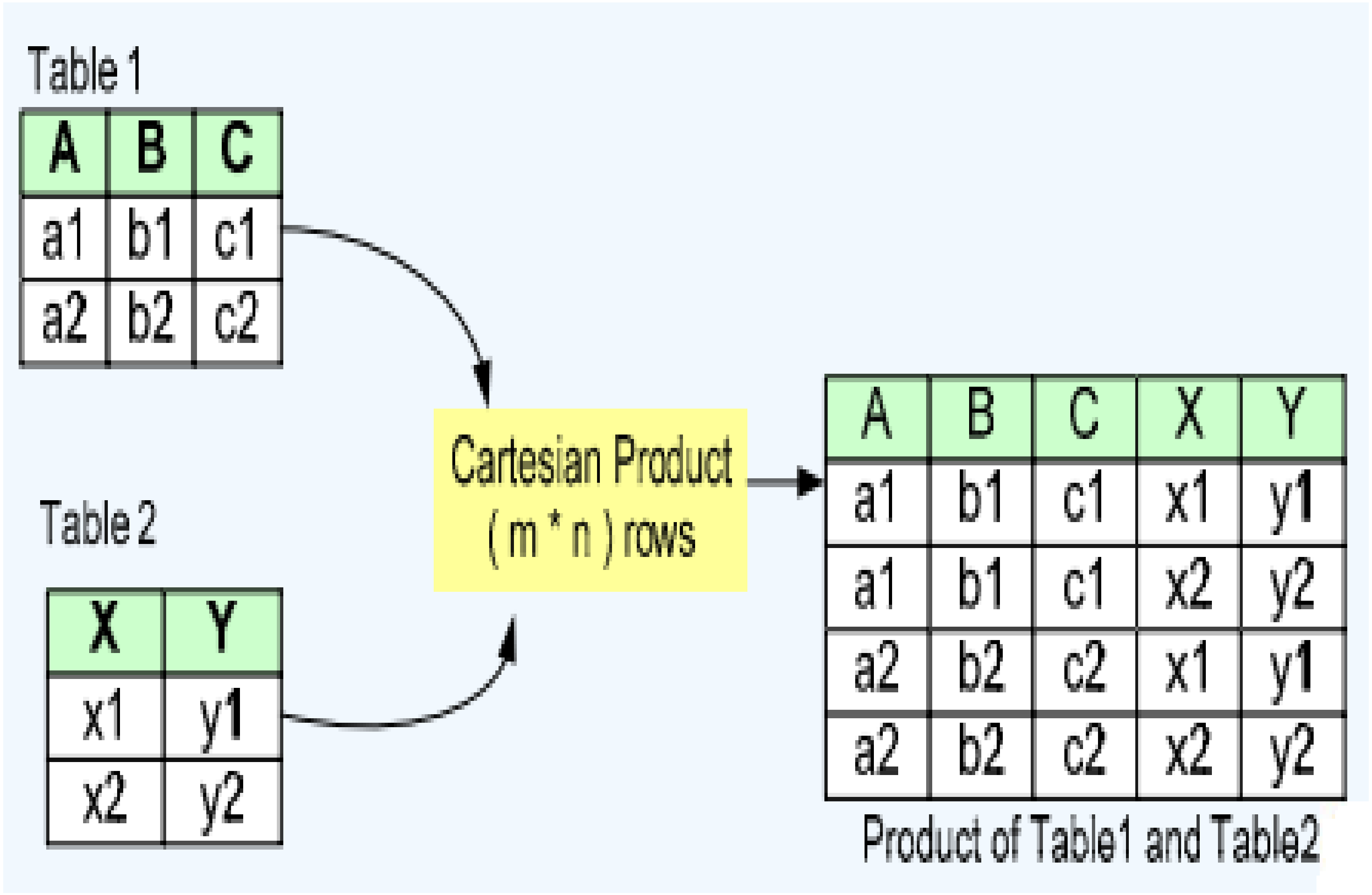


Table 1

A	B	C
a1	b1	c1
a2	b2	c2

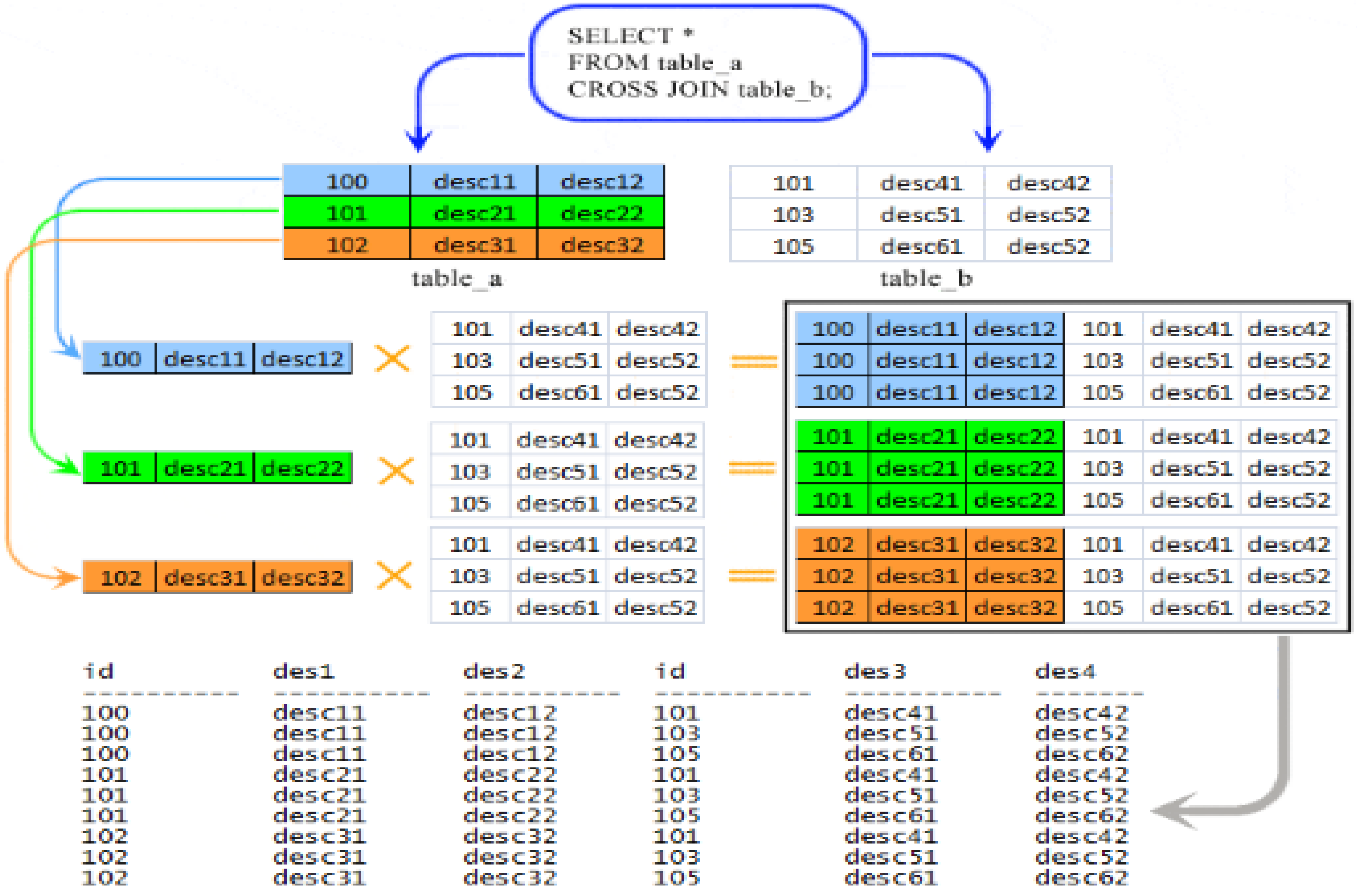
Table 2

X	Y
x1	y1
x2	y2

A	B	C	X	Y
a1	b1	c1	x1	y1
a1	b1	c1	x2	y2
a2	b2	c2	x1	y1
a2	b2	c2	x2	y2

Product of Table1 and Table2

CROSS JOIN



```
SELECT *
FROM table_a
CROSS JOIN table_b;
```

100	desc11	desc12
101	desc21	desc22
102	desc31	desc32

table_a

101	desc41	desc42
103	desc51	desc52
105	desc61	desc52

table_b

100	desc11	desc12

×

101	desc41	desc42
103	desc51	desc52
105	desc61	desc52

101	desc21	desc22

×

101	desc41	desc42
103	desc51	desc52
105	desc61	desc52

102	desc31	desc32

×

101	desc41	desc42
103	desc51	desc52
105	desc61	desc52

=

100	desc11	desc12	101	desc41	desc42
100	desc11	desc12	103	desc51	desc52
100	desc11	desc12	105	desc61	desc52
101	desc21	desc22	101	desc41	desc42
101	desc21	desc22	103	desc51	desc52
101	desc21	desc22	105	desc61	desc52
102	desc31	desc32	101	desc41	desc42
102	desc31	desc32	103	desc51	desc52
102	desc31	desc32	105	desc61	desc52

```
id          des1        des2        id          des3        des4
----------  ----------  ----------  ----------  ----------  -------
100         desc11      desc12      101         desc41      desc42
100         desc11      desc12      103         desc51      desc52
100         desc11      desc12      105         desc61      desc62
101         desc21      desc22      101         desc41      desc42
101         desc21      desc22      103         desc51      desc52
101         desc21      desc22      105         desc61      desc62
102         desc31      desc32      101         desc41      desc42
102         desc31      desc32      103         desc51      desc52
102         desc31      desc32      105         desc61      desc62
```

CROSS JOIN

- In the absence of a WHERE or ON condition the CARTESIAN (CROSS) JOIN will behave like a CARTESIAN PRODUCT. i.e., the number of rows in the result-set is the product of the number of rows of the two tables.
- In the presence of WHERE or ON condition this JOIN will function like a INNER JOIN

CROSS JOIN - EXAMPLE

Write a query to perform Cartesian product between Product and Orders Relations.

SELECT *FROM

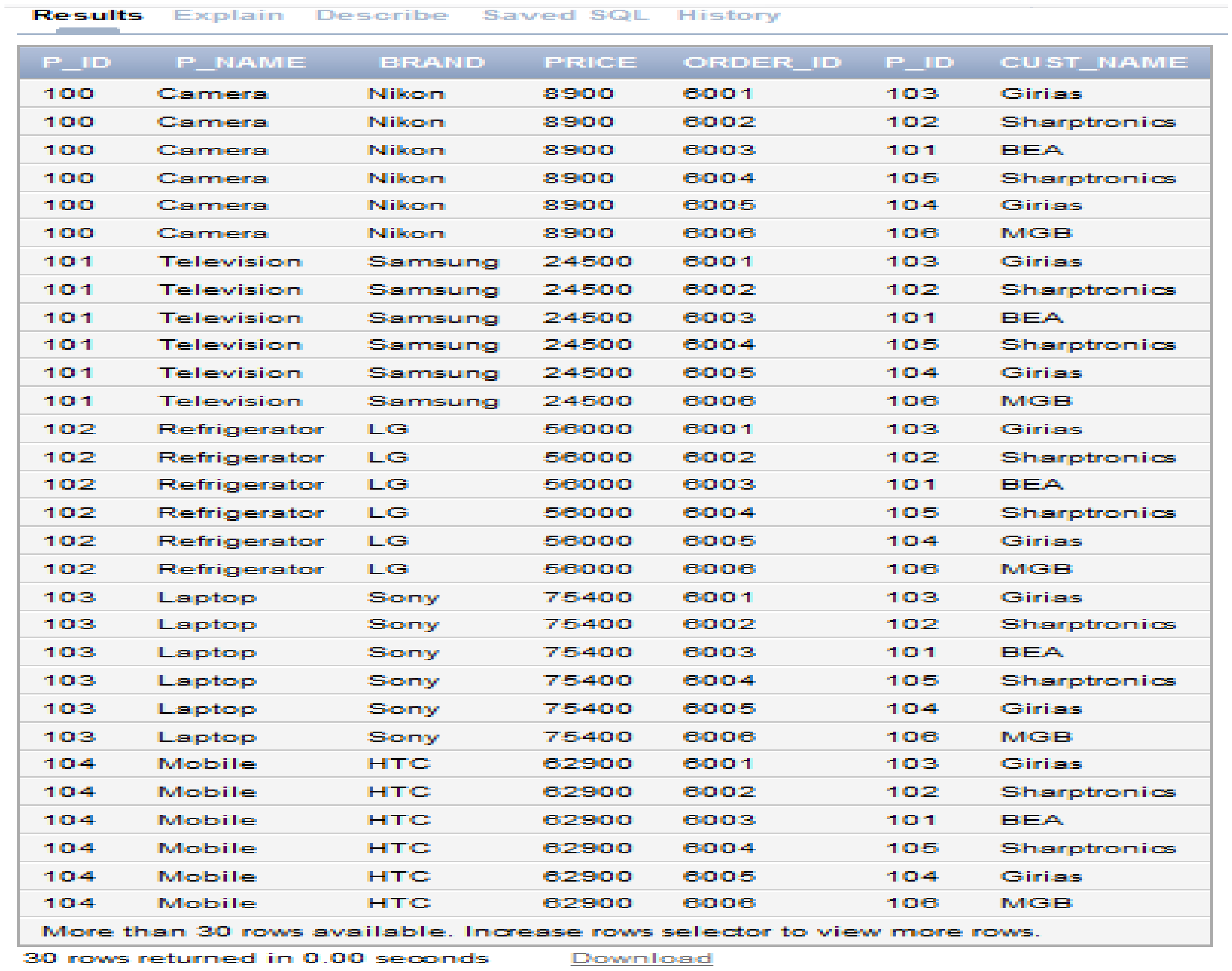
PRODUCT CROSS JOIN ORDERS;

(OR)

SELECT *FROM

PRODUCT,ORDERS;

CROSS JOIN – EXAMPLE - OUTPUT



P_ID	P_NAME	BRAND	PRICE	ORDER_ID	P_ID	CUST_NAME
100	Camera	Nikon	8900	6001	103	Girias
100	Camera	Nikon	8900	6002	102	Sharptronics
100	Camera	Nikon	8900	6003	101	BEA
100	Camera	Nikon	8900	6004	105	Sharptronics
100	Camera	Nikon	8900	6005	104	Girias
100	Camera	Nikon	8900	6006	106	MGB
101	Television	Samsung	24500	6001	103	Girias
101	Television	Samsung	24500	6002	102	Sharptronics
101	Television	Samsung	24500	6003	101	BEA
101	Television	Samsung	24500	6004	105	Sharptronics
101	Television	Samsung	24500	6005	104	Girias
101	Television	Samsung	24500	6006	106	MGB
102	Refrigerator	LG	56000	6001	103	Girias
102	Refrigerator	LG	56000	6002	102	Sharptronics
102	Refrigerator	LG	56000	6003	101	BEA
102	Refrigerator	LG	56000	6004	105	Sharptronics
102	Refrigerator	LG	56000	6005	104	Girias
102	Refrigerator	LG	56000	6006	106	MGB
103	Laptop	Sony	75400	6001	103	Girias
103	Laptop	Sony	75400	6002	102	Sharptronics
103	Laptop	Sony	75400	6003	101	BEA
103	Laptop	Sony	75400	6004	105	Sharptronics
103	Laptop	Sony	75400	6005	104	Girias
103	Laptop	Sony	75400	6006	106	MGB
104	Mobile	HTC	62900	6001	103	Girias
104	Mobile	HTC	62900	6002	102	Sharptronics
104	Mobile	HTC	62900	6003	101	BEA
104	Mobile	HTC	62900	6004	105	Sharptronics
104	Mobile	HTC	62900	6005	104	Girias
104	Mobile	HTC	62900	6006	106	MGB

More than 30 rows available. Increase rows selector to view more rows.

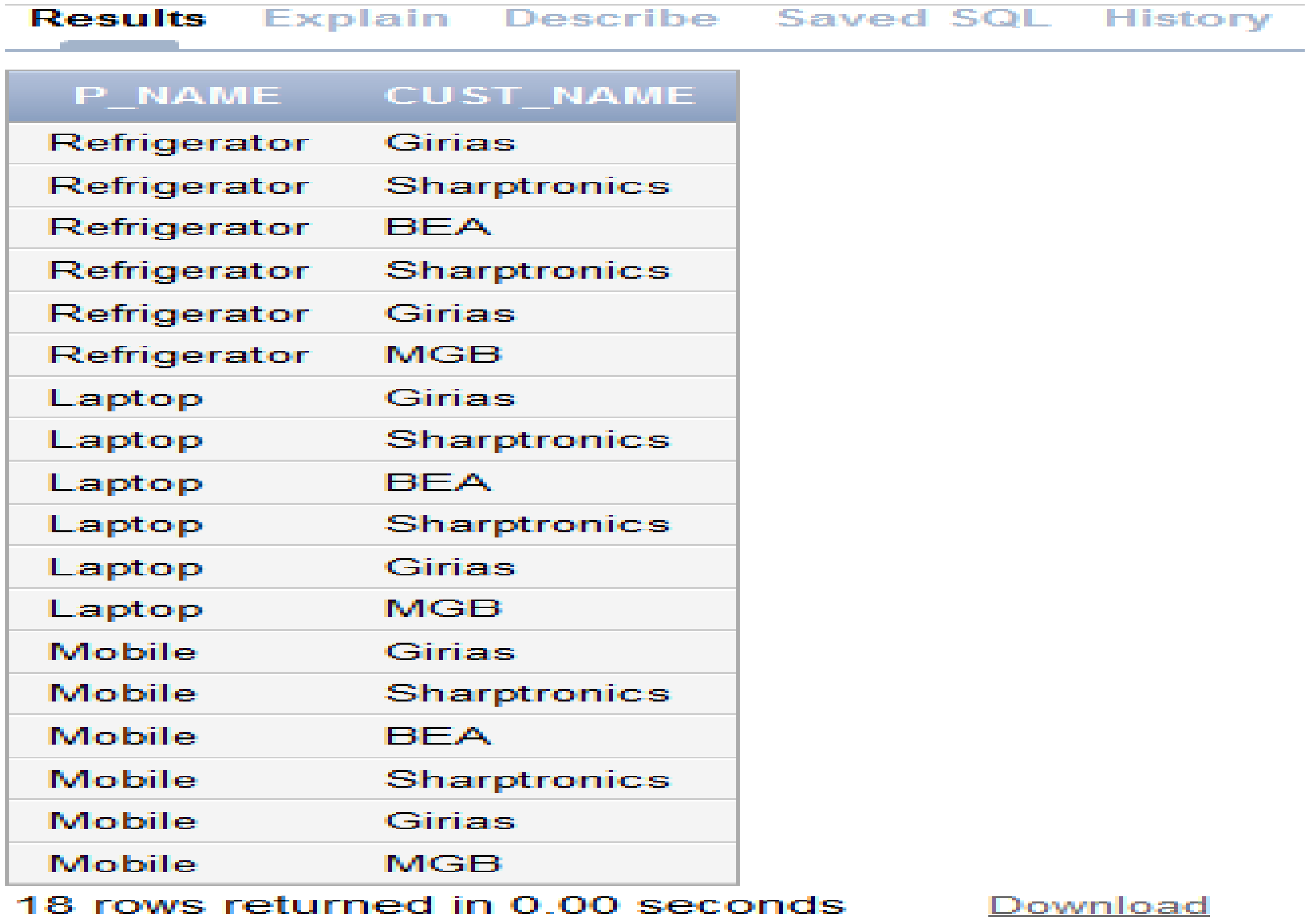
30 rows returned in 0.00 seconds   Download

CROSS JOIN - EXAMPLE

Write a query to perform Cartesian product between Product and Orders Relations. The result should filtered for those products whose price ranges from 50000 to 80000 with the name of the product and customers.

```
SELECT Product.P_Name, Orders.Cust_Name FROM
PRODUCT CROSS JOIN ORDERS
WHERE Product.Price BETWEEN 50000 AND 80000;
```

CROSS JOIN – EXAMPLE - OUTPUT



P_NAME	CUST_NAME
Refrigerator	Girias
Refrigerator	Sharptronics
Refrigerator	BEA
Refrigerator	Sharptronics
Refrigerator	Girias
Refrigerator	MGB
Laptop	Girias
Laptop	Sharptronics
Laptop	BEA
Laptop	Sharptronics
Laptop	Girias
Laptop	MGB
Mobile	Girias
Mobile	Sharptronics
Mobile	BEA
Mobile	Sharptronics
Mobile	Girias
Mobile	MGB

18 rows returned in 0.00 seconds

NATURAL JOIN

- Natural Join is a type of Inner join which is based on column having same name and same datatype present in both the tables to be joined.
- The join creates, by using the NATURAL JOIN keywords.
- When specifying columns that are involved in the natural join, do not qualify the column name with a table name or table alias.

NOTE:

Don't use ON or USING clause in a natural join.

Syntax:

SELECT * FROM

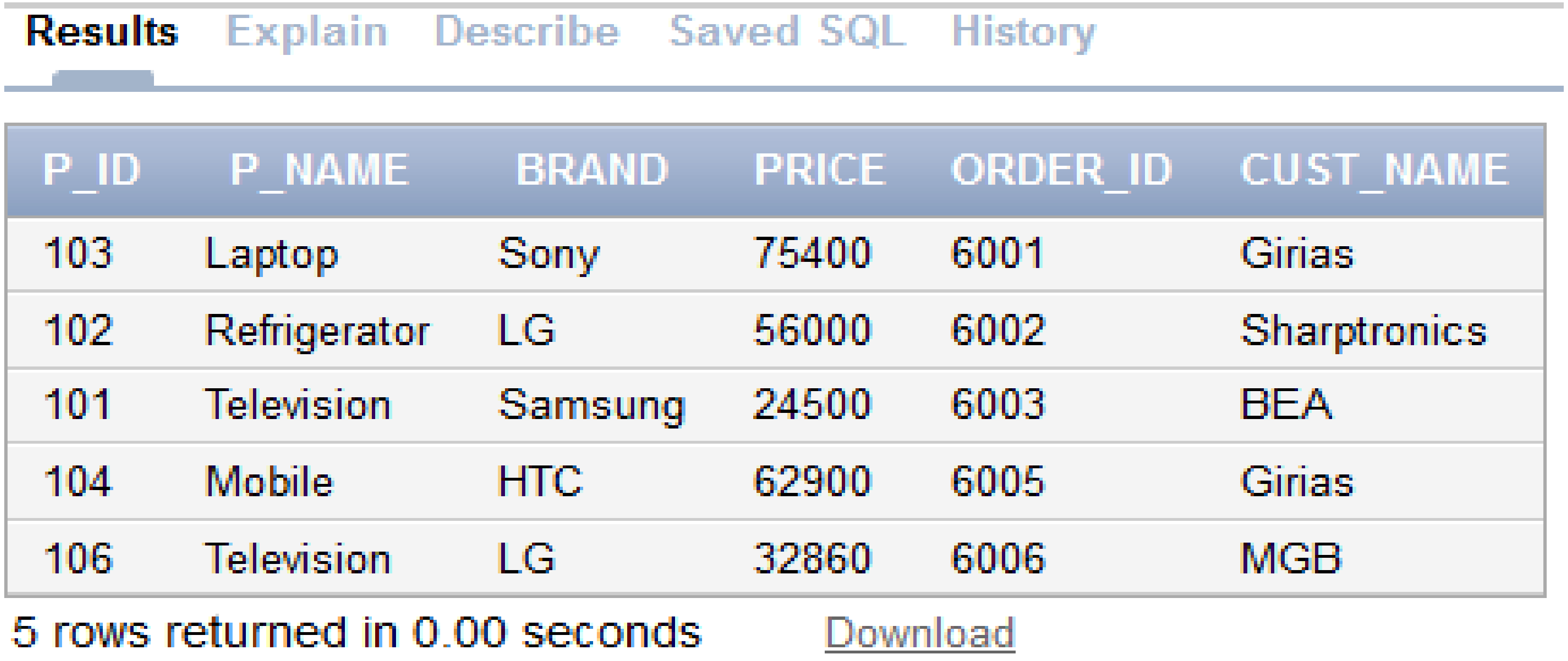
table-name1 **NATURAL JOIN** table-name2;

NATURAL JOIN - EXAMPLE

Perform Natural Join between Product and Orders Relations.

SELECT * FROM

Product NATURAL JOIN Orders;



Results Explain Describe Saved SQL History

P_ID	P_NAME	BRAND	PRICE	ORDER_ID	CUST_NAME
103	Laptop	Sony	75400	6001	Girias
102	Refrigerator	LG	56000	6002	Sharptronics
101	Television	Samsung	24500	6003	BEA
104	Mobile	HTC	62900	6005	Girias
106	Television	LG	32860	6006	MGB

5 rows returned in 0.00 seconds Download

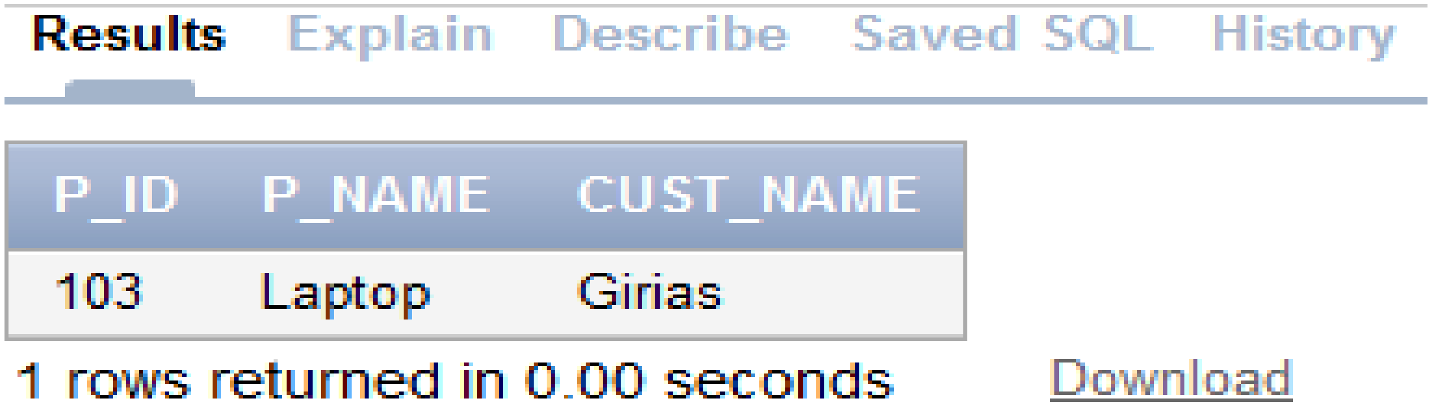
NATURAL JOIN - EXAMPLE

Perform Natural Join between Product and Orders Relations. Display the ID and Name of the products along with the customer name for which the price exceeds 75000.

SELECT P_ID, P_Name, Cust_Name FROM

Product NATURAL JOIN Orders

WHERE Price >75000;



P_ID	P_NAME	CUST_NAME
103	Laptop	Girias

1 rows returned in 0.00 seconds

NATURAL JOIN

Points to be Noted:

- NATURAL JOIN and USING Clause are mutually exclusive.
- NATURAL JOIN uses all the columns with matching names and datatypes to join the tables.
- The USING Clause can be used to specify only those columns that should be used for an EQUIJOIN.

SELF JOIN

- As the name signifies, in SELF JOIN a table is joined to itself.
- That is, each row of the table is joined with itself and all other rows depending on some conditions.
- In other words we can say that it is a join between two copies of the same table.
- A self join is useful for comparing rows within a table or querying hierarchical data.
- In addition, it uses the table alias to give the table different names in the same query.

SELF JOIN

Syntax:

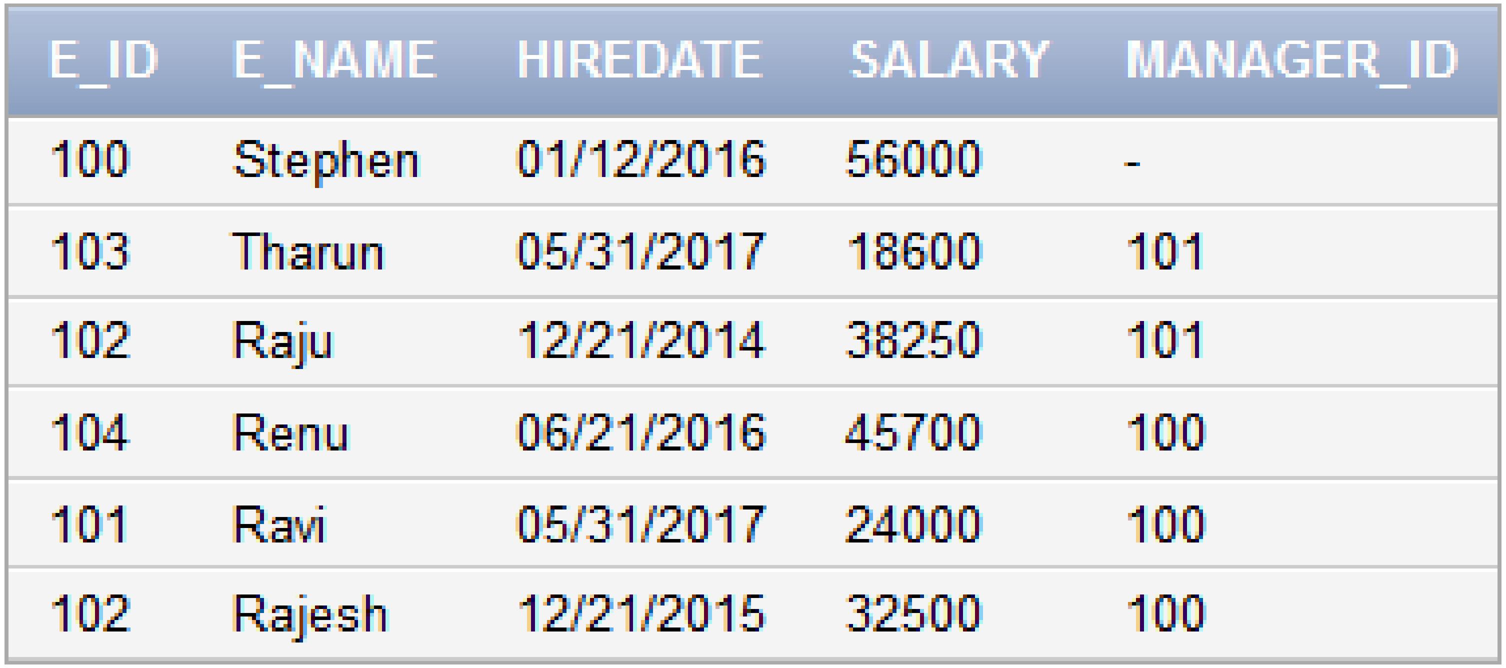
SELECT Columns_List
FROM **table_name A, table_name B**
WHERE condition;

(OR)

SELECT Columns_List
FROM **table_name A JOIN table_name B**
ON condition;

SELF JOIN

RELATION: EMPLOYEE



E_ID	E_NAME	HIREDATE	SALARY	MANAGER_ID
100	Stephen	01/12/2016	56000	-
103	Tharun	05/31/2017	18600	101
102	Raju	12/21/2014	38250	101
104	Renu	06/21/2016	45700	100
101	Ravi	05/31/2017	24000	100
102	Rajesh	12/21/2015	32500	100

SELF JOIN - EXAMPLE

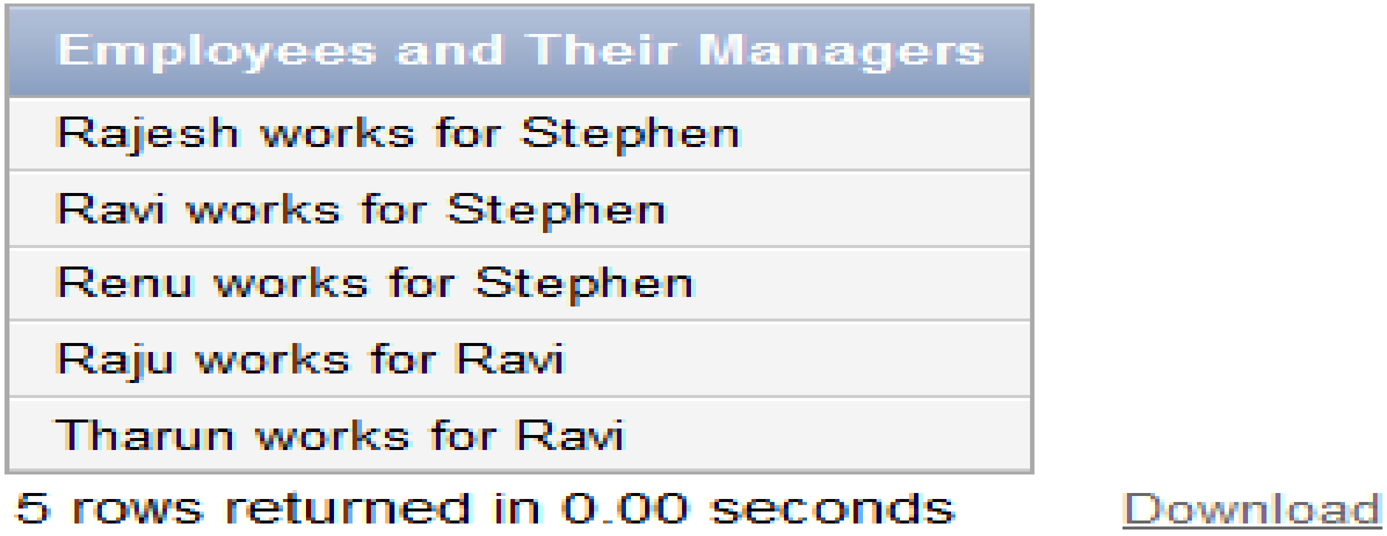
Write a query to return the name of each employee along with the name of the employee's manager.

SELECT e1.E_Name||' works for '||e2. E_Name

"Employees and Their Managers"

FROM Employee e1, Employee e2

WHERE e1.manager_id = e2.employee_id;



Employees and Their Managers
Rajesh works for Stephen
Ravi works for Stephen
Renu works for Stephen
Raju works for Ravi
Tharun works for Ravi

5 rows returned in 0.00 seconds Download

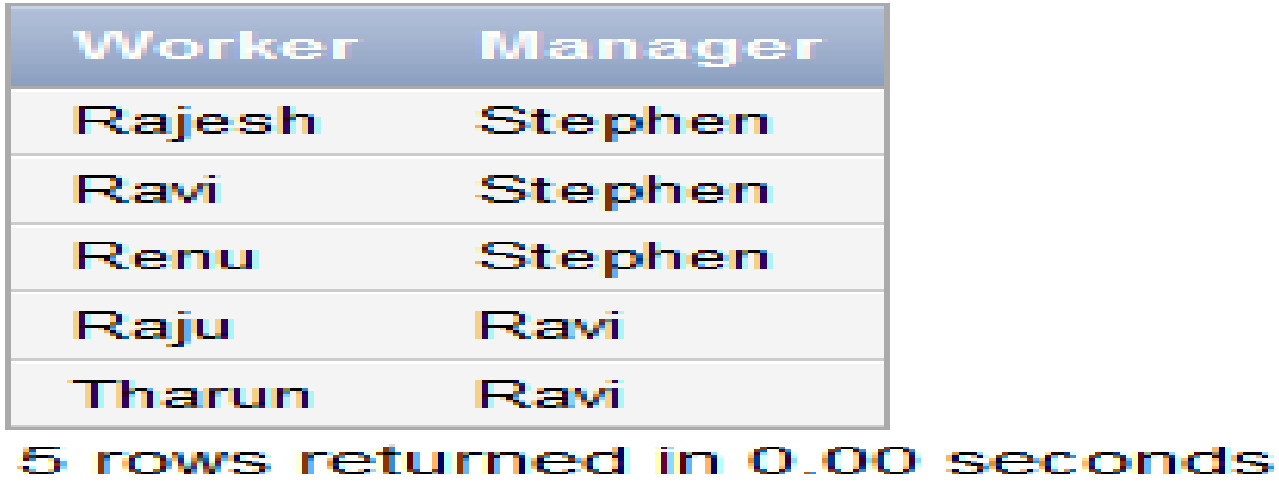
SELF JOIN - EXAMPLE

Self-Joins Using the ON Clause

SELECT e1.E_Name "Worker", e2.E_Name "Manager"

FROM Employee e1 JOIN Employee e2

ON e1.manager_id = e2.e_id;



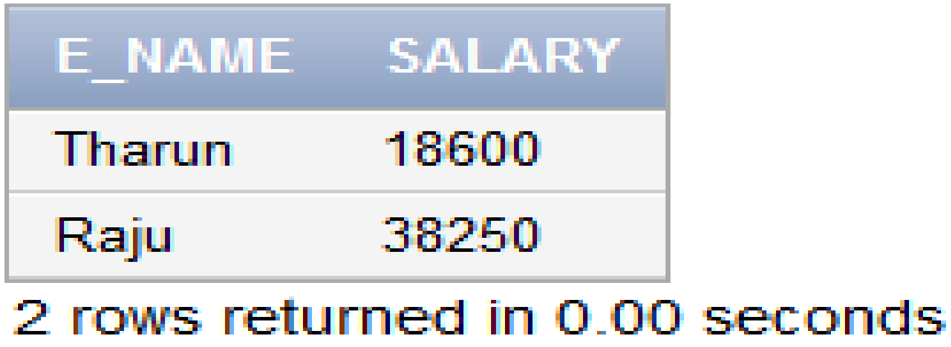
Worker	Manager
Rajesh	Stephen
Ravi	Stephen
Renu	Stephen
Raju	Ravi
Tharun	Ravi

5 rows returned in 0.00 seconds

SELF JOIN - EXAMPLE

List all the employees with Name and Salary who are working under the same Manager as that of Raju.

```
Select e1.E_name,e1.Salary from employee e1,employee e2
WHERE e1.Manager_id=e2.Manager_id
AND e2.E_name='Raju';
```



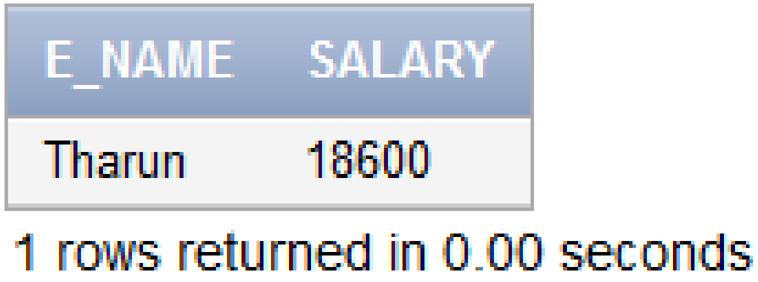
E_NAME	SALARY
Tharun	18600
Raju	38250

2 rows returned in 0.00 seconds

SELF JOIN - EXAMPLE

List all the employees with Name and Salary who are working under the same Manager as that of Raju. The Result set should exclude Raju's information.

```
Select e1.E_name,e1.Salary from employee e1,employee e2
WHERE e1.Manager_id=e2.Manager_id
AND e2.E_name='Raju' AND e1.E_name!='Raju';
```



E_NAME	SALARY
Tharun	18600

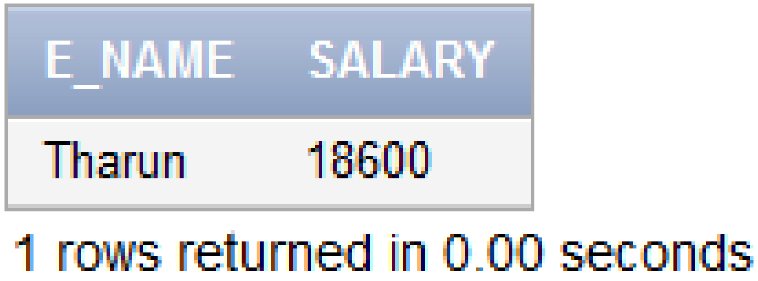
```
1 rows returned in 0.00 seconds
```

SELF JOIN - EXAMPLE

List all the employees with Name and Salary who are working under the same Manager as that of Raju. The Result set should exclude Raju’s information.

USING JOIN Keyword:

```
Select e1.E_name,e1.Salary from employee e1 JOIN employee e2
ON e1.Manager_id=e2.Manager_id
AND e2.E_name='Raju' AND e1.E_name!='Raju';
```



E_NAME	SALARY
Tharun	18600

1 rows returned in 0.00 seconds