JOINS

What is JOIN?

A join is an SQL operation performed to **establish a connection between two or more database tables Based on matching columns**, thereby creating a Relationship between the tables. Most complex queries in an SQL database management system involve join commands.

Types of JOIN

- Equijoins
- Non-Equi Joins
- Self Joins
- Cross Join / Cartesian Products
- Inner Join
- Outer Joins
  - Left Outer Join
  - Right Outer Join
  - Full Outer Join
- Natural Join
- Anti Join
- Semi Join

Sample Table:

Products

P_ID	P_NAME	S_NAME	PRICE
100	camera	nikon	300
101	TV	onida	100
102	fridge	vediocon	150
103	ipod	apple	75
104	mobile	nokia	50

order_items

ORDER_ID	P_ID	UNITS	CUS_NAME
5100	104	30	infosys
5101	102	5	satyam
5102	103	25	wipro
5103	101	10	tcs
5104	105	20	cts

Equijoins

- An equijoin is a join with a join condition containing an equality operator. This is represented by (=) sign. This join retrieves information by using equality condition.
- EQUI JOIN can be performed 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)]
```

Example: without JOIN keyword

Display the details from Product and order table .

SELECT *

FROM products, order_items

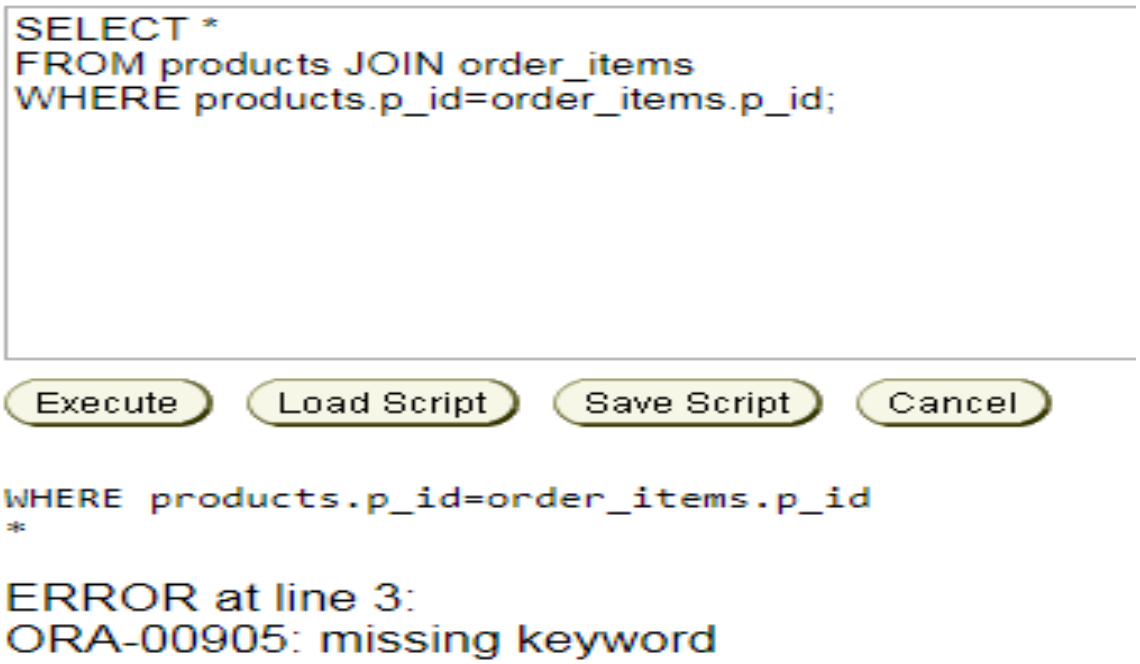
WHERE products.p_id=order_items.p_id;

P_ID	P_NAME	S_NAME	PRICE	ORDER_ID	P_ID	UNITS	CUS_NAME
104	mobile	nokia	50	5100	104	30	infosys
102	fridge	vediocon	150	5101	102	5	satyam
103	ipod	apple	75	5102	103	25	wipro
101	TV	onida	100	5103	101	10	tcs

When Using JOIN keyword in Query we have to use ON clause or USING clause.

Note:

When **where** clause is used with JOIN keyword it throws an error.



Example: with JOIN Keyword

Display the details from Product and order table .

SELECT *

FROM products **JOIN** order_items

ON products.p_id=order_items.p_id**;**

(OR)

With table alias name:

SELECT *

FROM products **p JOIN** order_items **o**

ON p.p_id=o.p_id;

P_ID	P_NAME	S_NAME	PRICE	ORDER_ID	P_ID	UNITS	CUS_NAME
104	mobile	nokia	50	5100	104	30	infosys
102	fridge	vediocon	150	5101	102	5	satyam
103	ipod	apple	75	5102	103	25	wipro
101	TV	onida	100	5103	101	10	tcs

Non-Equijoin

An non-equijoin is an inner join statement that uses an unequal operation (i.e.: <>, >, <, !=, BETWEEN, etc.) to match rows from different tables.

Syntax:

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

Example: Non equiJOIN

Display the details from product and order_items tables, if Product id of product table is greater than product id in order_items table.

SELECT *

FROM products **p JOIN** order_items **o**

ON p.p_id > o.p_id**;**

(OR)

SELECT *

FROM products **p,** order_items **o**

WHERE p.p_id>o.p_id;

P_ID	P_NAME	S_NAME	PRICE	ORDER_ID	P_ID	UNITS	CUS_NAME
104	mobile	nokia	50	5102	103	25	wipro
104	mobile	nokia	50	5101	102	5	satyam
104	mobile	nokia	50	5103	101	10	tcs
103	ipod	apple	75	5101	102	5	satyam
103	ipod	apple	75	5103	101	10	tcs
102	fridge	vediocon	150	5103	101	10	tcs

Self Join

- A self join is such a join in which a table is joined with itself(which is also called Unary relationships). For example, when you require details about an employee and his manager (also an employee).
- Especially when the table has a FOREIGN KEY which references its own PRIMARY KEY. To join a table 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.

Self Join

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 the actual the table name are used, since both the tables have the same name. Table name aliases are defined in the FROM clause of the SELECT statement.

select * from emp;

EMP_ID	EMP_NAME	DT_OF_JOIN	EMP_SUPERVISOR
20051	Flora	04-JAN-16	
20073	Mani	09-FEB-15	20051
20064	Kavi	23-OCT-09	20073
20069	Anu	03-DEC-12	20051
20055	Divya	01-MAR-07	20073

FROM table1 a, table1 b

WHERE a.common_filed = b.common_field;

Example 1

Display Emp_id, emp_name, supervisor ID and Supervisor name from the given relation.

SELECT a.emp_id "Emp_Id", a.emp_name "Emp_name", b.emp_id "supervisor ID", b.emp_name "supervisor name"
FROM emp a, emp b
WHERE a.emp_supervisor=b.emp_id;

Emp_Id	Emp_name	supervisor ID	supervisor name
20069	Anu	20051	Flora
20073	Mani	20051	Flora
20055	Divya	20073	Mani
20064	Kavi	20073	Mani

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.
- An alternative way of achieving the same result is to use column names separated by commas after SELECT and mentioning the table names involved, after a FROM clause.
- A Cross Join or Cartesian join or Cartesian product is a join of every row of one table to every row of another table.

Syntax

SELECT *

FROM table1

CROSS JOIN table2**;**

Perform Cross join between products and order_items table.

```
SELECT *
FROM products
CROSS JOIN order_items ;
```

(OR)

```
SELECT *
FROM products , order_items;
```

Output:[Cross Join]

P_ID	P_NAME	S_NAME	PRICE	ORDER_ID	P_ID	UNITS	CUS_NAME
100	camera	nikon	300	5100	104	30	infosys
100	camera	nikon	300	5101	102	5	satyam
100	camera	nikon	300	5102	103	25	wipro
100	camera	nikon	300	5103	101	10	tcs
100	camera	nikon	300	5104	105	20	cts
101	TV	onida	100	5100	104	30	infosys
101	TV	onida	100	5101	102	5	satyam
101	TV	onida	100	5102	103	25	wipro
101	TV	onida	100	5103	101	10	tcs
101	TV	onida	100	5104	105	20	cts
102	fridge	vediocon	150	5100	104	30	infosys
102	fridge	vediocon	150	5101	102	5	satyam
102	fridge	vediocon	150	5102	103	25	wipro
102	fridge	vediocon	150	5103	101	10	tcs
102	fridge	vediocon	150	5104	105	20	cts
103	ipod	apple	75	5100	104	30	infosys
103	ipod	apple	75	5101	102	5	satyam
103	ipod	apple	75	5102	103	25	wipro
103	ipod	apple	75	5103	101	10	tcs
103	ipod	apple	75	5104	105	20	cts
104	mobile	nokia	50	5100	104	30	infosys
104	mobile	nokia	50	5101	102	5	satyam
104	mobile	nokia	50	5102	103	25	wipro
104	mobile	nokia	50	5103	101	10	tcs

Inner Join

The INNER JOIN selects all rows from both participating tables as long as there is a match between the columns. An SQL INNER JOIN is same as JOIN clause, combining rows from two or more tables.

Syntax:

SELECT a.column_name, b.column_name...
FROM table1 a
INNER JOIN table2 b
ON a.column_name =b.column_name;

(OR)

SELECT table1.column_name, table2.column_name...
FROM table1
JOIN table2
ON table1.column_name = table2.column_name;

Inner Join Syntax

(OR)

```
SELECT *
FROM table1
JOIN table2
USING (column_name);
```

Question 1

Display p_id,p_name,order_id and units from the above Relation

SELECT p.p_id, p.p_name, o.p_id, o.order_id, o.units
FROM products p
INNER JOIN order_items o
ON p.p_id=o.p_id;

(OR)

Example: JOIN keyword

SELECT p.p_id, p.p_name, o.p_id, o.order_id, o.units
FROM products p
JOIN order_items o
ON p.p_id=o.p_id;

Outer Joins

An outer join is such a join which is similar to the equijoin, but Oracle will also return non matched rows from the table.

1. LEFT OUTER JOIN
2. RIGHT OUTER JOIN
3. FULL OUTER JOIN

LEFT OUTER JOIN

The LEFT JOIN (OR) LEFT OUTER 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.

Question 1

Display the details from product and order_items table using LEFT OUTER JOIN

```
SELECT *
FROM products p
LEFT OUTER JOIN order_items o
ON p.p_id=o.p_id;
```

(OR)

```
SELECT *
FROM products
LEFT OUTER JOIN order_items
USING (p_id);
```

Question 1[OUTPUT]

P_ID	P_NAME	S_NAME	PRICE	ORDER_ID	P_ID	UNITS	CUS_NAME
104	mobile	nokia	50	5100	104	30	infosys
102	fridge	vediocon	150	5101	102	5	satyam
103	ipod	apple	75	5102	103	25	wipro
101	TV	onida	100	5103	101	10	tcs
100	camera	nikon	300				

RIGHT OUTER 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.

Question 1

Display the details from product and order_items table using RIGHT OUTER JOIN

```
SELECT *
FROM products p
RIGHT OUTER JOIN order_items o
ON p.p_id=o.p_id;
```

(OR)

```
SELECT *
FROM products
RIGHT OUTER JOIN order_items
USING (p_id);
```

Question 1[OUTPUT]

P_ID	P_NAME	S_NAME	PRICE	ORDER_ID	P_ID	UNITS	CUS_NAME
101	TV	onida	100	5103	101	10	tcs
102	fridge	vediocon	150	5101	102	5	satyam
103	ipod	apple	75	5102	103	25	wipro
104	mobile	nokia	50	5100	104	30	infosys
				5104	105	20	cts

FULL OUTER JOIN

The FULL OUTER JOIN keyword return all records when there is a match in either left (table1) or right (table2) table records.

Question 1

Display the details from product and order_items table using FULL OUTER JOIN

```
SELECT *
FROM products
FULL OUTER JOIN order_items
USING (p_id);
```

P_ID	P_NAME	S_NAME	PRICE	ORDER_ID	UNITS	CUS_NAME
104	mobile	nokia	50	5100	30	infosys
102	fridge	vediocon	150	5101	5	satyam
103	ipod	apple	75	5102	25	wipro
101	TV	onida	100	5103	10	tcs
100	camera	nikon	300			
105				5104	20	cts

NATURAL JOIN

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 will appear once only. Result table does not containNULL value.**

Guidelines

- The associated tables have one or more pairs of identically named columns.
- The columns must be the same data type.
- Don’t use ON clause in a natural join.

Question 1

Display the details from product and order_items table using NATURAL OUTER JOIN

SELECT *

FROM products

NATURAL JOIN order_items

P_ID	P_NAME	S_NAME	PRICE	ORDER_ID	UNITS	CUS_NAME
104	mobile	nokia	50	5100	30	infosys
102	fridge	vediocon	150	5101	5	satyam
103	ipod	apple	75	5102	25	wipro
101	TV	onida	100	5103	10	tcs

THANK U