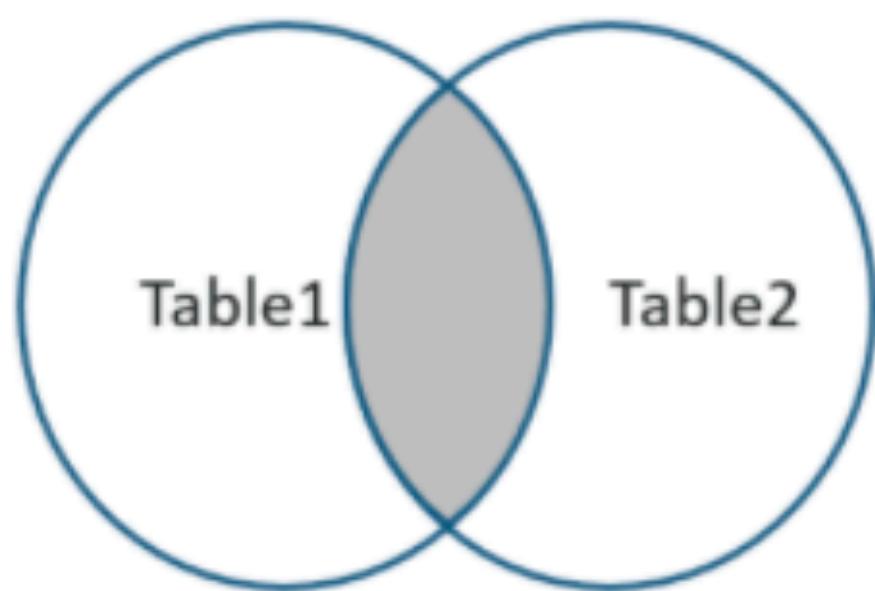
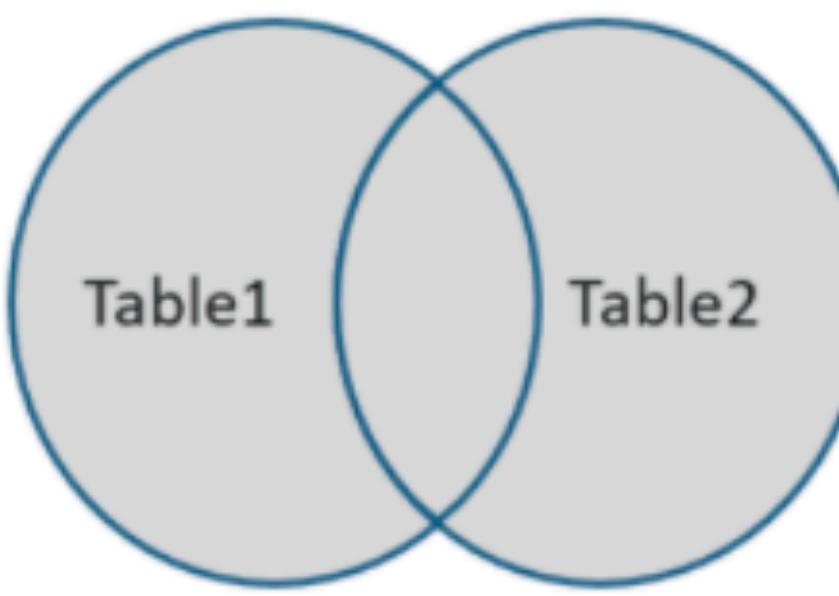


INNER JOIN, LEFT JOIN, RIGHT JOIN, CROSS JOIN

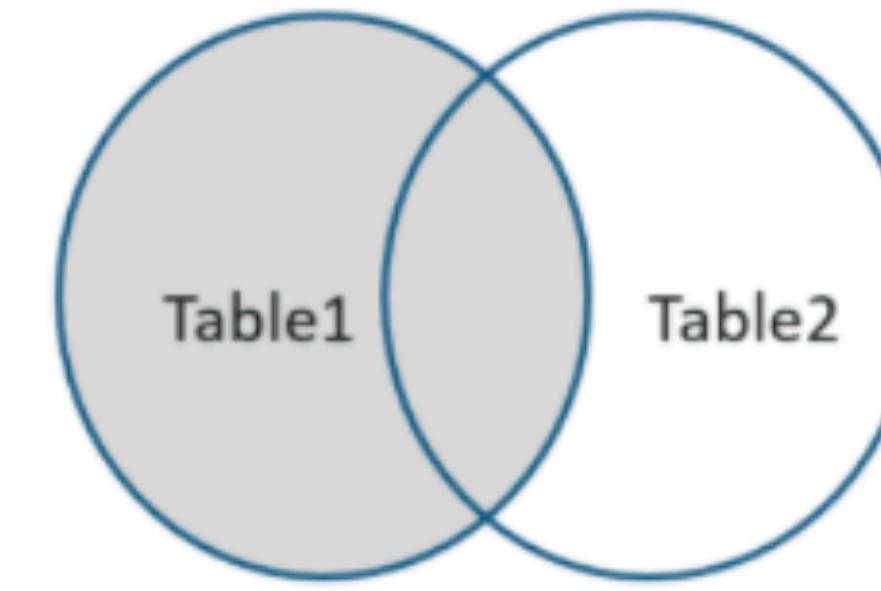
Ms. VIDHYA.S
Assistant Professor(Sr. Gr.)
Department of Computer Science and Engineering
Amrita School of Engineering
Amrita Vishwa Vidyapeetham
Coimbatore



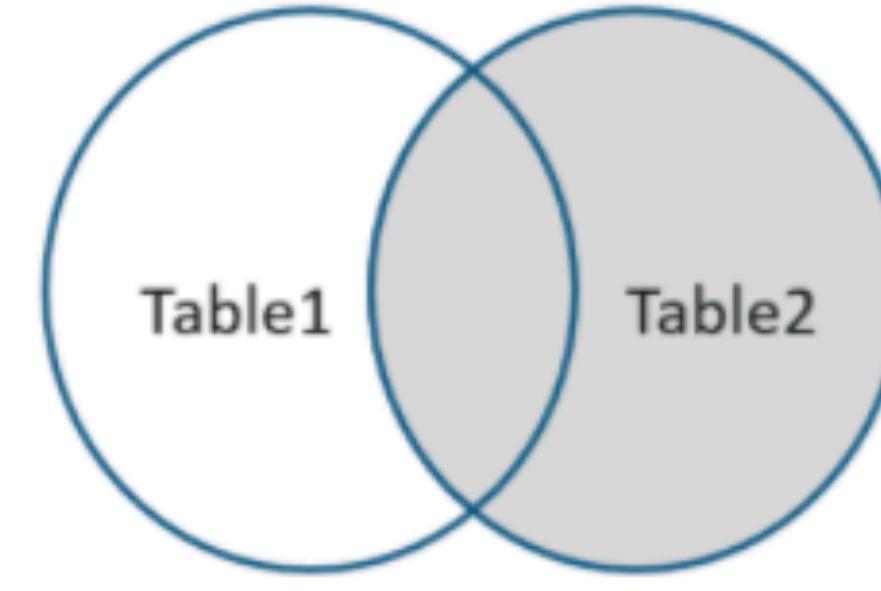
INNER JOIN



FULL JOIN



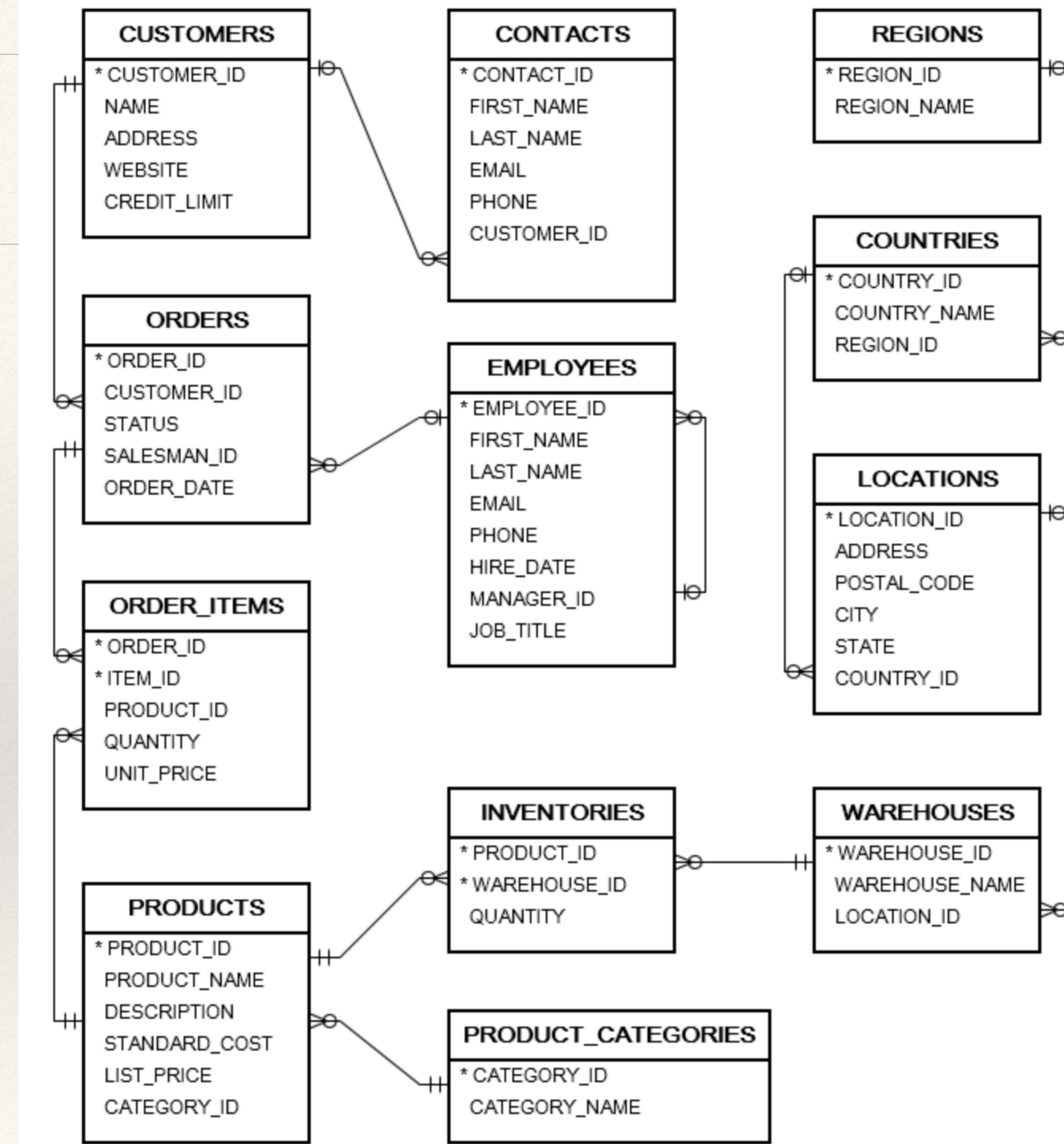
LEFT JOIN

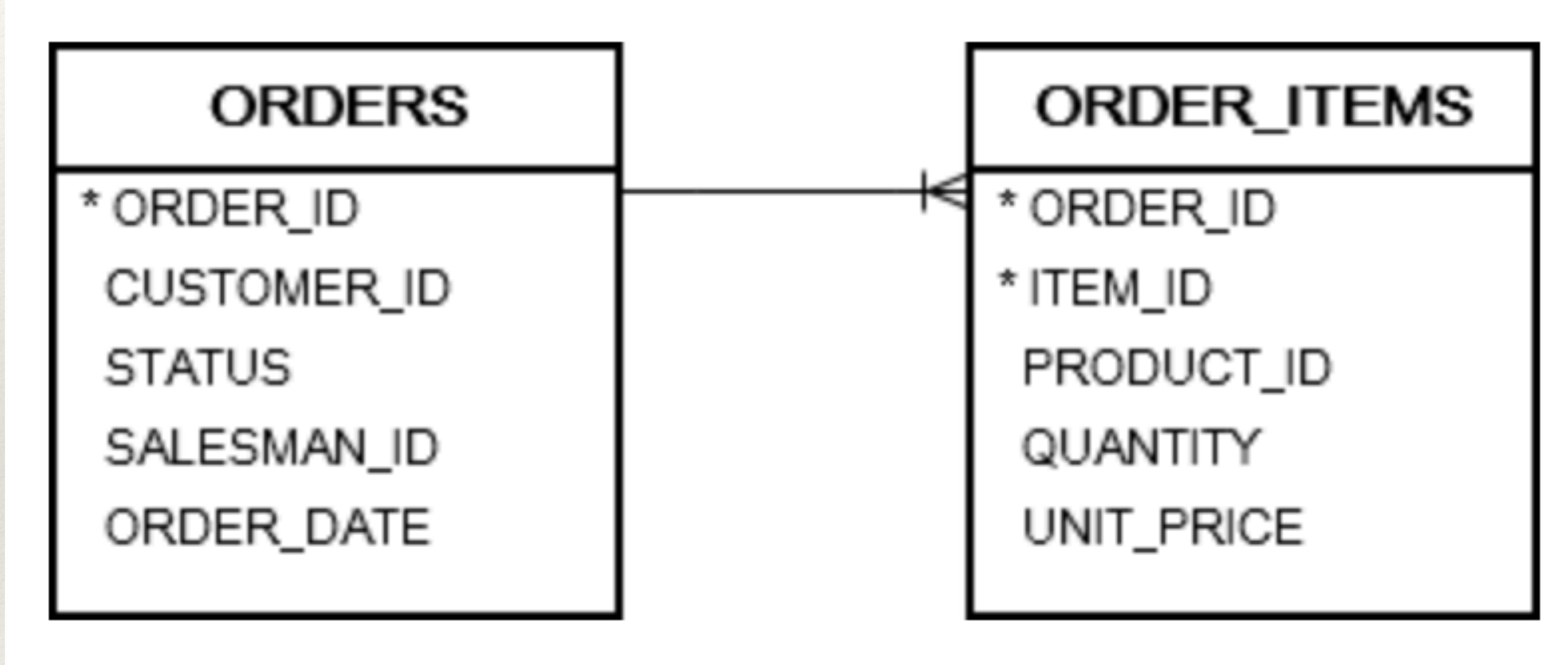


RIGHT JOIN

edureka!

Sample Database





Pic courtesy - oracle tutorial

INNER JOIN

- ❖ The `orders` table stores the order's header information and the `order_items` table stores the order line items.
- ❖ The `orders` table links to the `order_items` table via the `order_id` column. It means that for each row in the `orders` table, you can find one or more rows in the `order_items` with the same values in the `order_id` column.
- ❖ To query data from two or more related tables, you use the `INNER JOIN` clause. The following statement illustrates how to join two tables T1 and T2.

```
SELECT *
FROM T1
INNER JOIN T2 ON join_predicate;
```

INNER JOIN

```
SELECT *
FROM T1
INNER JOIN T2 ON join_predicate;
```

- ❖ First, specify the main table in the **FROM** clause, **T1** in this case.
- ❖ Second, specify the joined table in the **INNER JOIN** clause followed by a **join_predicate**. The joined table is **T2** in the above statement.
- ❖ Third, a join predicate specifies the condition for joining tables. Only rows that satisfy the join predicate are included in the result set.

INNER JOIN - Eg

Project Table:

ProjectID	EmplD	ClientID	ProjectName	ProjectStartDate
111	1	3	Project1	2019-04-21
222	2	1	Project2	2019-02-12
333	3	5	Project3	2019-01-10
444	3	2	Project4	2019-04-16
555	5	4	Project5	2019-05-23
666	9	1	Project6	2019-01-12
777	7	2	Project7	2019-07-25
888	8	3	Project8	2019-08-20

Employee Table:

EmplD	EmpFname	EmpLname	Age	EmailID	PhoneNo	Address
1	Vardhan	Kumar	22	vardy@abc.com	9876543210	Delhi
2	Himani	Sharma	32	himani@abc.com	9977554422	Mumbai
3	Aayushi	Shreshth	24	aayushi@abc.com	9977555121	Kolkata
4	Hemanth	Sharma	25	hemanth@abc.com	9876545666	Bengaluru
5	Swatee	Kapoor	26	swatee@abc.com	9544567777	Hyderabad

Client Table:

ClientID	ClientFname	ClientLname	Age	ClientEmailID	PhoneNo	Address	EmplD
1	Susan	Smith	30	susan@adn.com	9765411231	Kolkata	3
2	Mois	Ali	27	mois@jsq.com	9876543561	Kolkata	3
3	Soma	Paul	22	soma@wja.com	9966332211	Delhi	1
4	Zainab	Daginawala	40	zainab@qkq.com	9955884422	Hyderabad	5
5	Bhaskar	Reddy	32	bhaskar@xyz.com	9636963269	Mumbai	2

INNER JOIN - Eg

```
SELECT Employee.EmpID, Employee.EmpFname, Employee.EmpLname, Projects.ProjectID,  
Projects.ProjectName  
FROM Employee  
INNER JOIN Projects ON Employee.EmpID=Projects.EmpID;
```

Output:

EmpID	EmpFname	EmpLname	ProjectID	ProjectName
1	Vardhan	Kumar	111	Project1
2	Himani	Sharma	222	Project2
3	Aayushi	Shreshth	333	Project3
3	Aayushi	Shreshth	444	Project4
5	Swatee	Kapoor	555	Project5

INNER JOIN

- ❖ The query returns a result set by combining column values of both tables T1 and T2 based on the join predicate.
- ❖ It compares each row of table T1 with rows of table T2 to find all pairs of rows that satisfy the join predicate.
- ❖ Whenever the join predicate is satisfied by matching non-NULL values, column values for each matching pair of rows of T1 and T2 tables are combined into a row in the result set.

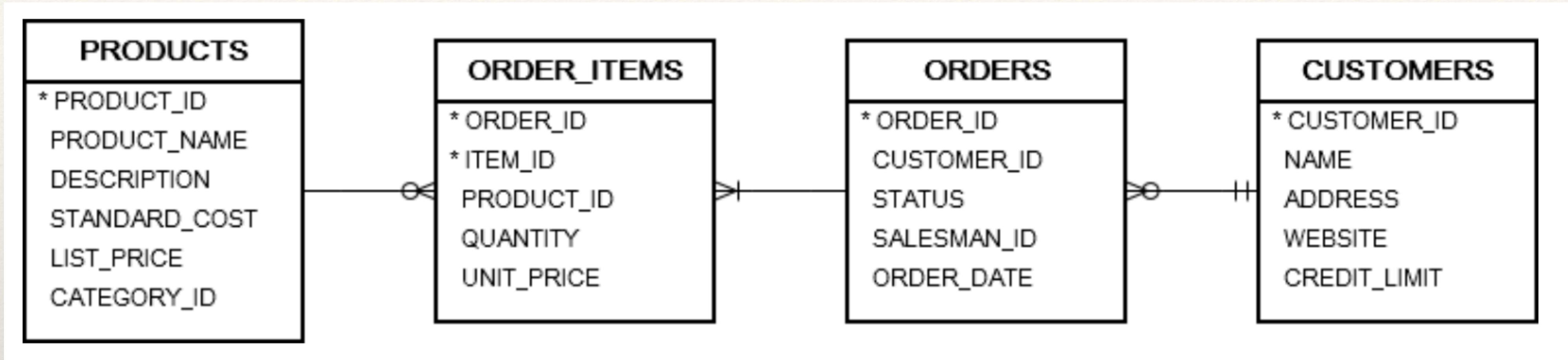
INNER JOIN with USING clause

- ❖ Besides the ON clause, it is possible to use the USING clause to specify which columns to test for equality when joining tables.

```
SELECT * FROM orders  
INNER JOIN order_items USING( order_id )  
ORDER BY order_date DESC;
```

ORDER_ID	CUSTOMER_ID	STATUS	SALESMAN_ID	ORDER_DATE	ITEM_ID	PRODUCT_ID	QUANTITY	UNIT_PRICE
88	6	Shipped	61	01-NOV-17	2	11	106	2015.11
88	6	Shipped	61	01-NOV-17	1	278	139	677.99
94	1	Shipped	62	27-OCT-17	6	4	111	2699.99
94	1	Shipped	62	27-OCT-17	5	181	143	999.99
94	1	Shipped	62	27-OCT-17	8	258	73	57.98
94	1	Shipped	62	27-OCT-17	4	80	133	564.89
94	1	Shipped	62	27-OCT-17	7	172	37	358.49
94	1	Shipped	62	27-OCT-17	2	186	146	1449.98
94	1	Shipped	62	27-OCT-17	3	218	86	1388.89
94	1	Shipped	62	27-OCT-17	9	12	33	824.98
94	1	Shipped	62	27-OCT-17	1	255	38	90.99
1	4	Pending	56	15-OCT-17	10	64	147	525.99

INNER JOIN - joining multiple tables



```
SELECT name AS customer_name, order_id, order_date, item_id, quantity, unit_price
FROM orders
INNER JOIN order_items USING(order_id)
INNER JOIN customers USING(customer_id)
ORDER BY order_date DESC, order_id DESC, item_id ASC;
```

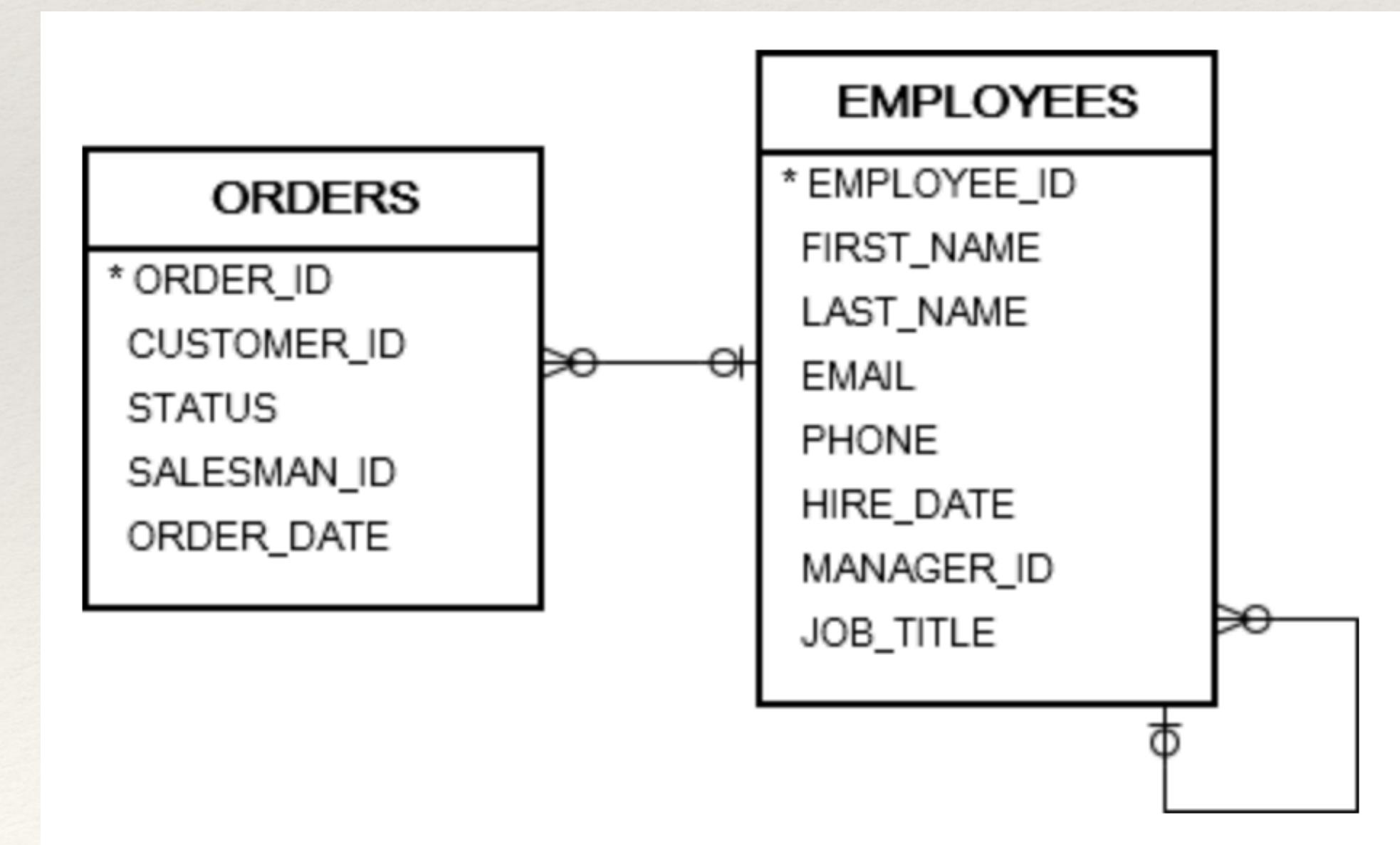
LEFT JOIN

- ❖ In this query, T1 is the left table and T2 is the right table.
- ❖ The query compares each row in the T1 table with rows in the T2 table.
- ❖ If a pair of rows from both T1 and T2 tables satisfy the join predicate, the query combines column values from rows in both tables and includes this row in the result set.
- ❖ In case a row in the T1 table does not have any matching row in the T2 table, the query combines column values from the row in the T1 table with a NULL value for each column in the right table that appears in the SELECT clause.
- ❖ In other words, a left join returns all rows from the left table and matching rows from the right table.

```
SELECT column_list  
FROM T1  
LEFT JOIN T2 ON  
join_predicate;
```

LEFT JOIN

- ❖ The **orders** table stores the sales order header data. It has the **salesman_id** column that references to the **employee_id** column in the employees table.
- ❖ The **salesman_id** column is null-able, meaning that not all orders have a sales employee who is in charge of the orders.



```
SELECT order_id, status, first_name, last_name  
FROM orders  
LEFT JOIN employees ON employee_id =  
salesman_id  
ORDER BY order_date DESC;
```

ORDER_ID	STATUS	FIRST_NAME	LAST_NAME
88	Shipped	Daisy	Ortiz
94	Shipped	Freya	Gomez
1	Pending	Evie	Harrison
14	Shipped	(null)	(null)
17	Shipped	(null)	(null)
15	Shipped	(null)	(null)
36	Shipped	(null)	(null)
57	Shipped	Scarlett	Gibson
28	Canceled	Scarlett	Gibson
29	Shipped	(null)	(null)
30	Shipped	(null)	(null)
31	Canceled	(null)	(null)
60	Shipped	Freva	Gomez

```
SELECT Employee.EmpFname, Employee.EmpLname, Projects.ProjectID, Projects.ProjectName  
FROM Employee  
LEFT JOIN  
ON Employee.EmpID = Projects.EmpID ;
```

Output:

EmpFname	EmpLname	ProjectID	ProjectName
Vardhan	Kumar	111	Project1
Himani	Sharma	222	Project2
Aayushi	Shreshth	333	Project3
Aayushi	Shreshth	444	Project4
Swatee	Kapoor	555	Project5
Hemanth	Sharma	NULL	NULL

LEFT JOIN with USING

```
SELECT name, order_id, status, order_date  
FROM customers  
LEFT JOIN orders USING(customer_id)  
ORDER BY name;
```

- ❖ This statement retrieved all customers and their orders.
- ❖ For the customers who have not placed any orders, null are used for the columns of orders table that appears in the SELECT clause, which are order_id, status, and order_date.

NAME	ORDER_ID	STATUS	ORDER_DATE
3M	(null)	(null)	(null)
ADP	(null)	(null)	(null)
AECOM	24	Shipped	07-SEP-16
AES	(null)	(null)	(null)
AIG	(null)	(null)	(null)
AT&T	(null)	(null)	(null)
AbbVie	2	Shipped	26-APR-15
AbbVie	1	Pending	15-OCT-17
AbbVie	85	Pending	01-DEC-16
AbbVie	63	Shipped	30-JUN-16
Abbott Laboratories	19	Shipped	27-MAY-16
Advance Auto Parts	(null)	(null)	(null)
Aetna	(null)	(null)	(null)
Aflac	75	Shipped	10-FEB-17

Conditions in ON vs WHERE

```
SELECT order_id, status, employee_id, last_name  
FROM orders  
LEFT JOIN employees ON  
    employee_id = salesman_id  
WHERE  
    order_id = 58;
```

ORDER_ID	STATUS	EMPLOYEE_ID	LAST_NAME
58	Shipped	57	Gibson

```
SELECT order_id, status, employee_id, last_name  
FROM orders  
LEFT JOIN employees ON  
    employee_id = salesman_id  
AND order_id = 58;
```

ORDER_ID	STATUS	EMPLOYEE_ID	LAST_NAME
58	Shipped	57	Gibson
71	Shipped	(null)	(null)
50	Pending	(null)	(null)
90	Pending	(null)	(null)
91	Pending	(null)	(null)
1	Pending	(null)	(null)
4	Shipped	(null)	(null)
51	Shipped	(null)	(null)

RIGHT JOIN

```
SELECT Employee.EmpFname, Employee.EmpLname, Projects.ProjectID, Projects.ProjectName  
FROM Employee  
RIGHT JOIN  
Projects ON Employee.EmpID = Projects.EmpID;
```

Output:

EmpFname	EmpLname	ProjectID	ProjectName
Vardhan	Kumar	111	Project1
Himani	Sharma	222	Project2
Aayushi	Shreshth	333	Project3
Aayushi	Shreshth	444	Project4
Swatee	Kapoor	555	Project5
NULL	NULL	666	Project6
NULL	NULL	777	Project7
NULL	NULL	888	Project8

CROSS JOIN

```
SELECT  
    product_id,  
    warehouse_id, quantity  
FROM  
    products  
CROSS JOIN warehouses;
```