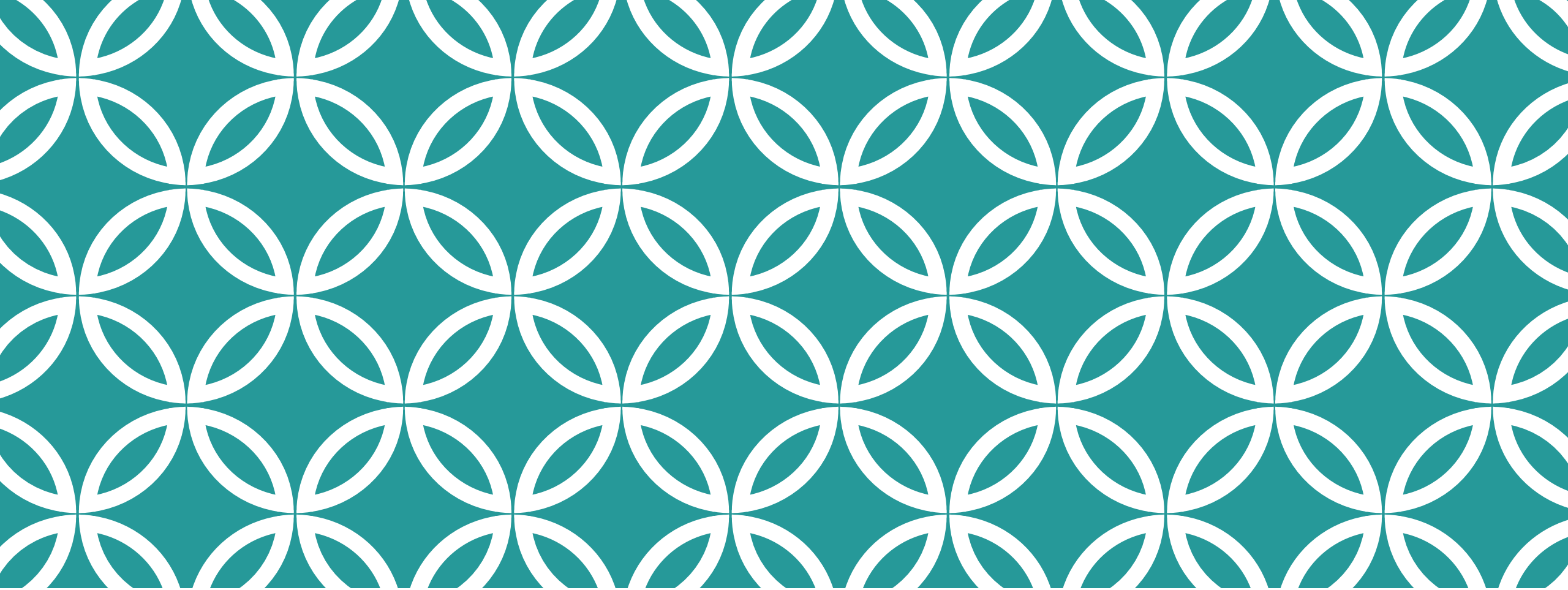


CHAPTER 11: NORMALIZATION & JOINS

2023

CHAPTER 11: NORMALIZATION & JOINS

Content:

1. What is normalization
2. Relationships
3. Joins
4. inner join
5. left join
6. right join
7. outer join
8. full join

WHAT IS NORMALIZATION?

Database normalization is the process of structuring a relational database in accordance with a series of so-called normal forms in order to reduce data redundancy and improve data integrity.

NORMALIZATION

First normal form

- Eliminate repeating groups in individual tables.
- Create a separate table for each set of related data.
- Create separate fields for combined parts
- Identify each set of related data with a primary key.

Second normal form

- Create separate tables for sets of values that apply to multiple records.
- Relate these tables with a foreign key.

Third normal form

- Eliminate fields that do not depend on the key.

NORMALIZATION

. Unnormalized table:

Student#	Advisor	Adv-Room	Class1	Class2	Class3
1022	Jones	412	101-07	143-01	159-02
4123	Smith	216	101-07	143-01	179-04

Student#	Advisor	Adv-Room	Class#
1022	Jones	412	101-07
1022	Jones	412	143-01
1022	Jones	412	159-02
4123	Smith	216	101-07
4123	Smith	216	143-01
4123	Smith	216	179-04

Students:

Student#	Advisor	Adv-Room
1022	Jones	412
4123	Smith	216

Registration:

Student#	Class#
1022	101-07
1022	143-01
1022	159-02
4123	101-07
4123	143-01
4123	179-04

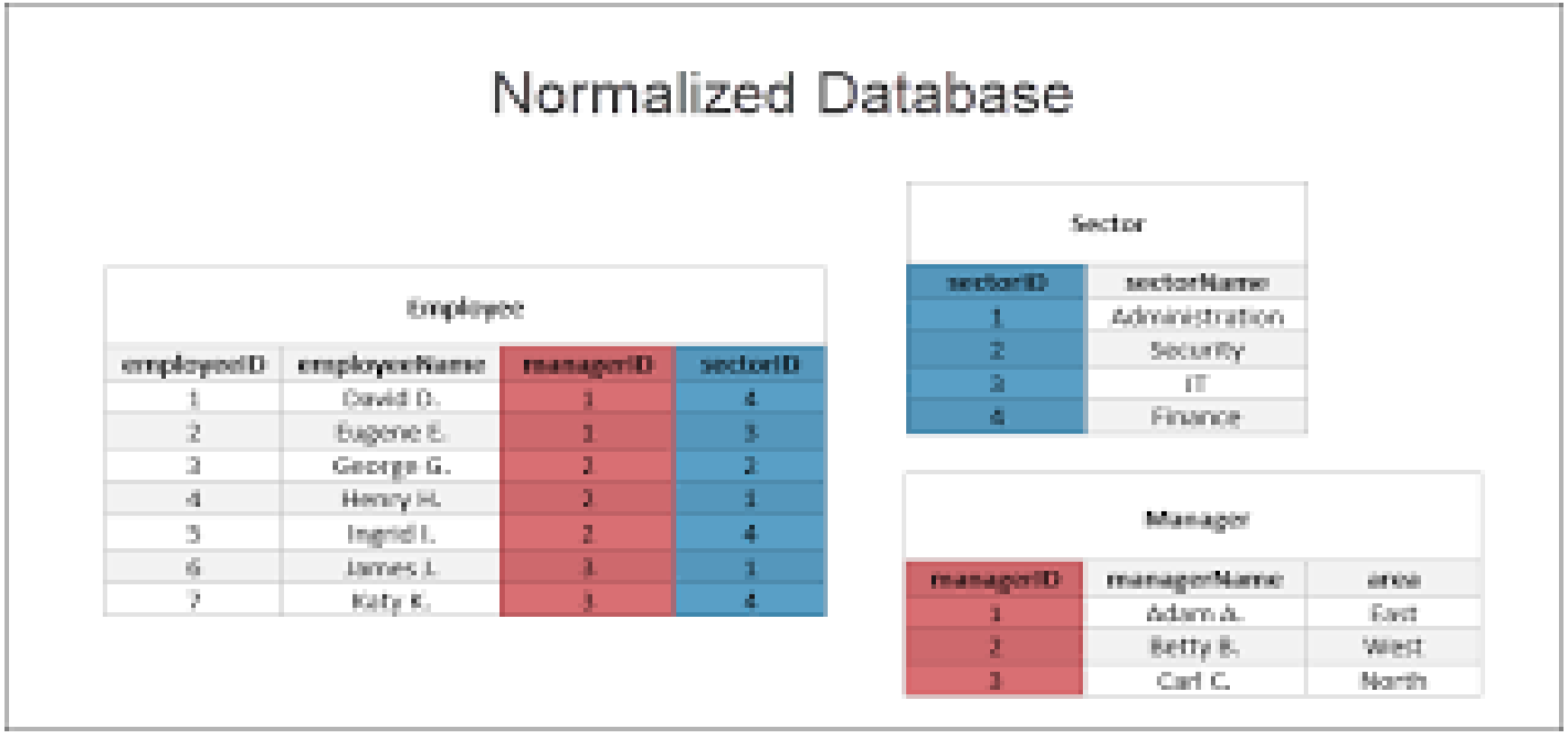
Practice NORMALISATION

1. *Observe the table below, Explain what normalization process you would apply to it to have a normalized dataset.*

Project Code	Project Name	Project Manager	Project Budget	Employee No.	Employee Name	Department No.	Department Name	Hourly Rate
PC010	Reservation System	Mr. Ajay	120500	S100	Mohan	D03	Database	21.00
PC010	Reservation System	Mr. Ajay	120500	S101	Vipul	D02	Testing	16.50
PC010	Reservation System	Mr. Ajay	120500	S102	Riyaz	D01	IT	22.00
PC011	HR System	Mrs. Charu	500500	S103	Pavan	D03	Database	18.50
PC011	HR System	Mrs. Charu	500500	S104	Jitendra	D02	Testing	17.00
PC011	HR System	Mrs. Charu	500500	S315	Pooja	D01	IT	23.50
PC012	Attendance System	Mr. Rajesh	710700	S137	Rahul	D03	Database	21.50
PC012	Attendance System	Mr. Rajesh	710700	S218	Avneesh	D02	Testing	15.50
PC012	Attendance System	Mr. Rajesh	710700	S109	Vikas	D01	IT	20.50

Practice NORMALISATION

1. *Observe the table below, Explain what normalization process was applied to it to have a normalized dataset.*



Practice NORMALISATION

1. *Observe the table below, Explain what normalization process you would apply to it, to have a normalized dataset.*

Orders

OrderNum	OrderDate	PartNum	Description	NumOrdered	QuotedPrice
21608	10/20/2010	AT94	Iron	11	$21.95
21610	10/20/2010	DR93	Gas Range	1	$495.00
21610	10/20/2010	DW11	Washer	1	$399.99
21613	10/21/2010	KL62	Dryer	4	$329.95
21614	10/21/2010	KT03	Dishwasher	2	$595.00
21617	10/23/2010	BV06	Home Gym	2	$794.95
21617	10/23/2010	CD52	Microwave Oven	4	$150.00
21619	10/23/2010	DR93	Gas Range	1	$495.00
21623	10/23/2010	KV29	Treadmill	2	$1290.00

Practice NORMALISATION

1. *Observe the table below, Explain what normalization process you would apply to it, to have a normalized dataset.*

Table 3. Unnormalized Table Employee

E#	EName	ESalary			
		te_Value	te_ValidTime		te_Gran
			te_Start	te_End	
1	John	2000	03/01/2000	12/01/2003	dd
			01/01/2004	12/01/2005	
		2200	12/01/2003	01/01/2004	dd
		2500	12/01/2005	12/31/9999	dd
2	Peter	2200	03/01/2000	06/01/2004	dd
			01/01/2005	08/01/2005	
		2500	06/01/2004	01/01/2005	dd
			08/01/2005	12/31/9999	

TYPES OF RELATIONSHIPS

One to One

One to many

Many to Many

ONE TO ONE

For every record in the first table one and only one record exists in the second table.
These are not common in relational databases

ONE TO MANY

This is a far more common relationship. Each record in the first table (the parent) is related to one or more records in the second table (the child). Each record in the second table is related to one and only one record in the first table.

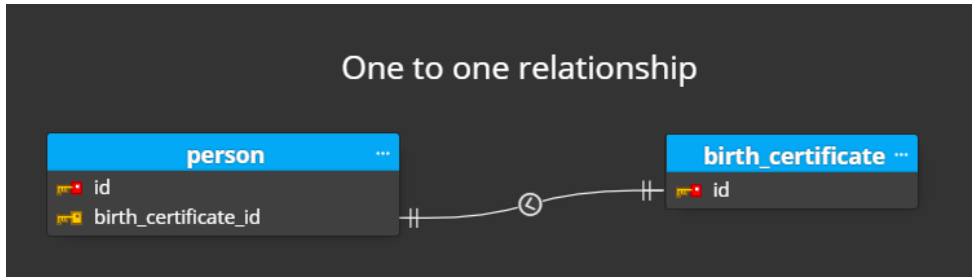
MANY TO MANY

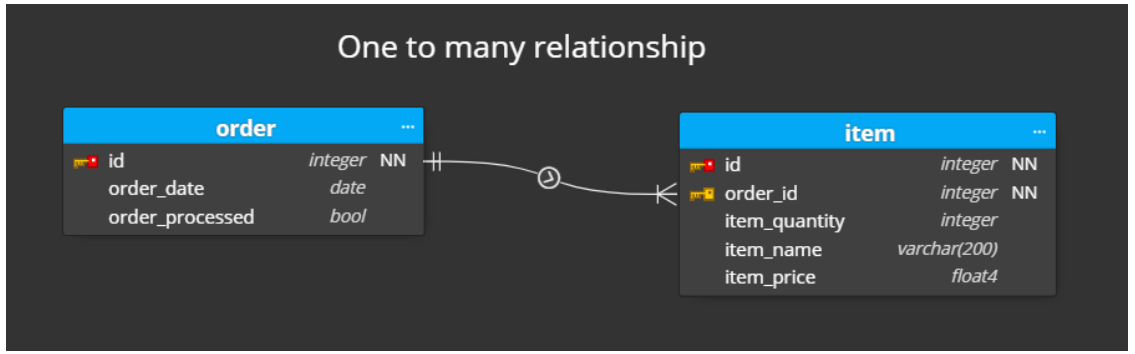
In a many to many arrangements each record in both tables can be related to zero one or many records in the other table.

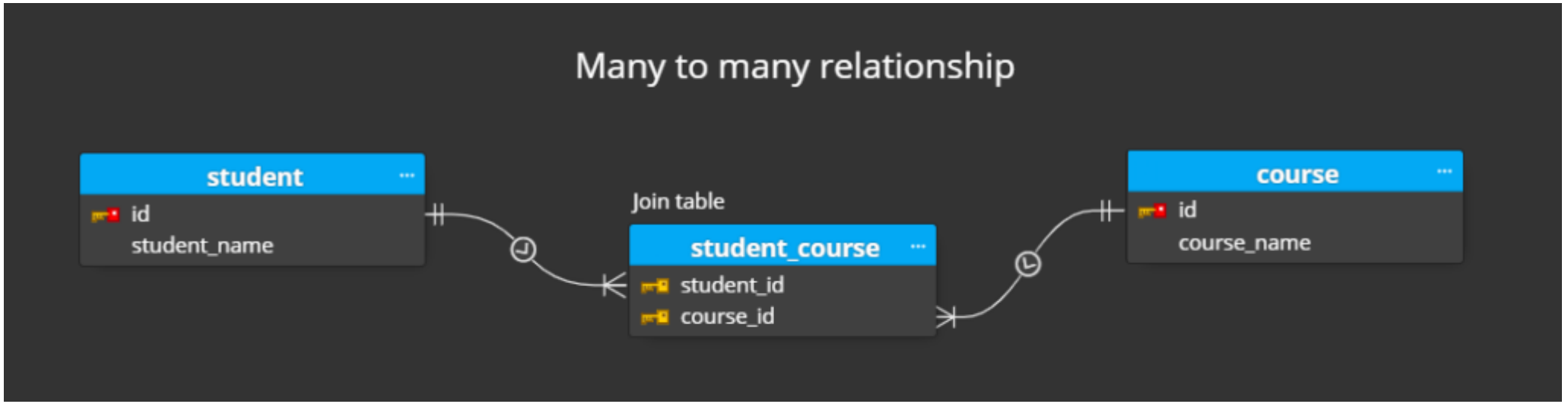
The many to many relationships is broken into

In order to create a many to many relationships the join table must contain the primary keys of both tables joined by the relationship.

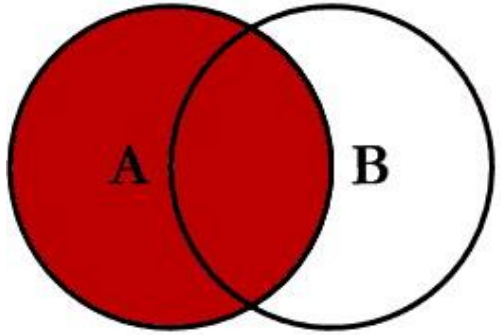
RELATIONSHIPS EXAMPLES



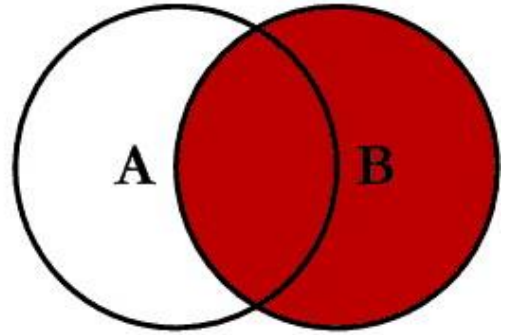




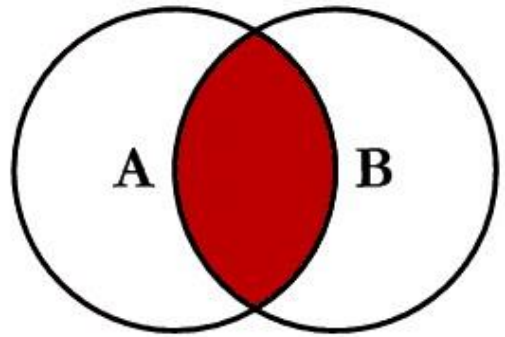
SQL JOINS



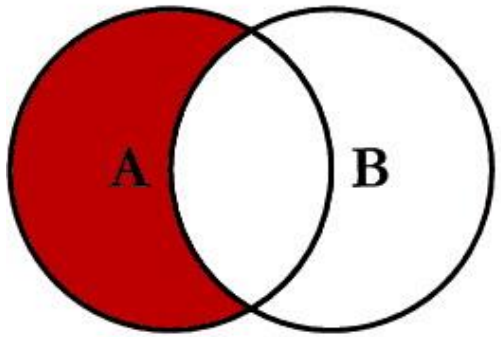

```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



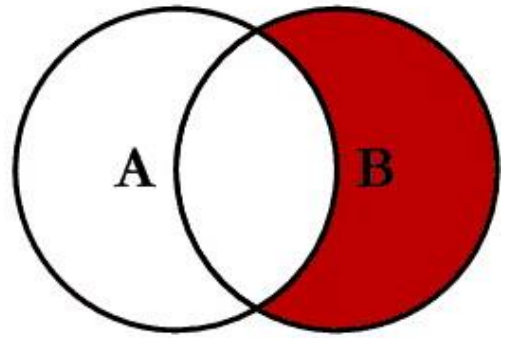

```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



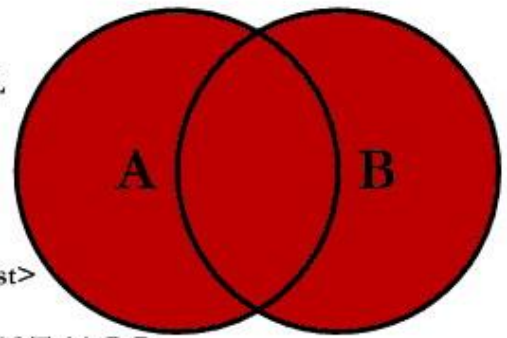

```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



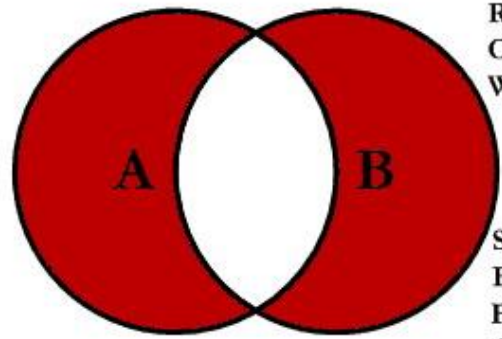

```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```




```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```




```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```




```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

DIFFERENT TYPES OF SQL JOINS

(INNER) JOIN:

Returns records that have matching values in both tables

LEFT (OUTER) JOIN:

Returns all records from the left table and the matched records from the right table

RIGHT (OUTER) JOIN:

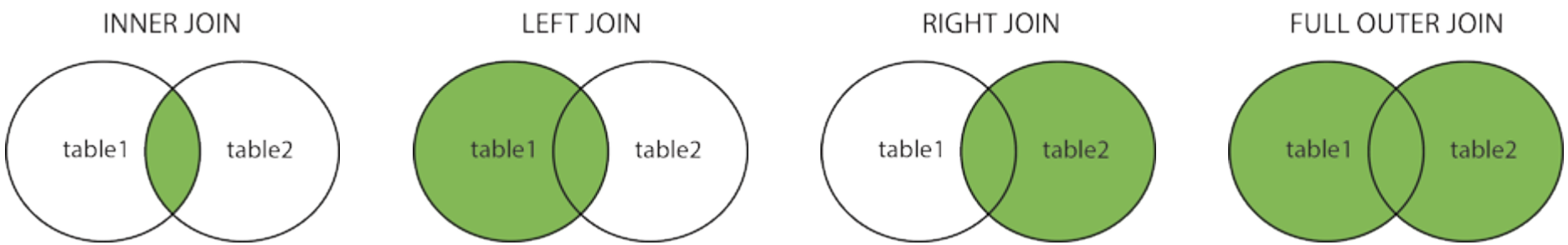
Returns all records from the right table and the matched records from the left table

FULL (OUTER) JOIN:

Returns all records when there is a match in either left or right table

JOINS

```
SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate
FROM Orders
INNER JOIN Customers
ON Orders.CustomerID=Customers.CustomerID;
```


INNER JOIN
table1
table2
LEFT JOIN
table1
table2
RIGHT JOIN
table1
table2
FULL OUTER JOIN
table1
table2

INNER JOIN

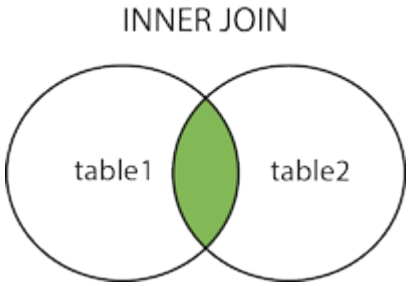
The INNER JOIN keyword selects all rows from both the tables as long as the condition satisfied.

This keyword will create the result-set by combining all rows from both the tables where the condition satisfies i.e value of the common field will be same.

table1: First table.

table2: Second table

matching column: Column common to both the tables.



Note: We can also write JOIN instead of INNER JOIN. JOIN is same as INNER JOIN.

INNER JOIN

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```

```
SELECT Orders.OrderID, Customers.CustomerName
FROM Orders
INNER JOIN Customers ON Orders.CustomerID =
Customers.CustomerID;
```

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.

table1: First table.

table2: Second table

matching_column: Column common to both the tables.

FULL OUTER JOIN

```
SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2
ON table1.column_name = table2.column_name
WHERE condition;
```

```
SELECT Customers.CustomerName, Orders.OrderID
FROM Customers
FULL OUTER JOIN Orders ON Customers.CustomerID=Orders.Custome
rID
ORDER BY Customers.CustomerName;
```

LEFT JOIN

This 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.

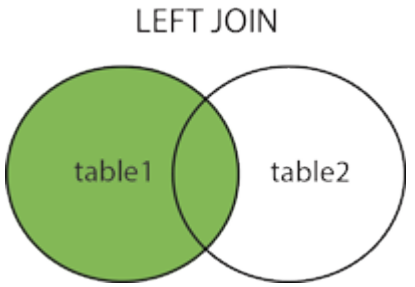


table1: First table.

table2: Second table

matching_column: Column common to both the tables.

We can also use LEFT OUTER JOIN instead of LEFT JOIN both are same.

LEFT JOIN

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```

```
SELECT Customers.CustomerName, Orders.OrderID
FROM Customers
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID
ORDER BY Customers.CustomerName;
```

RIGHT 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.

table1: First table.

table2: Second table

matching_column: Column common to both the tables.

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

RIGHT JOIN

```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```

```
SELECT Orders.OrderID,  Employees.LastName
Employees.FirstName
FROM Orders
RIGHT JOIN Employees ON Orders.EmployeeID =
Employees.EmployeeID
ORDER BY Orders.OrderID;
```

SELF JOIN

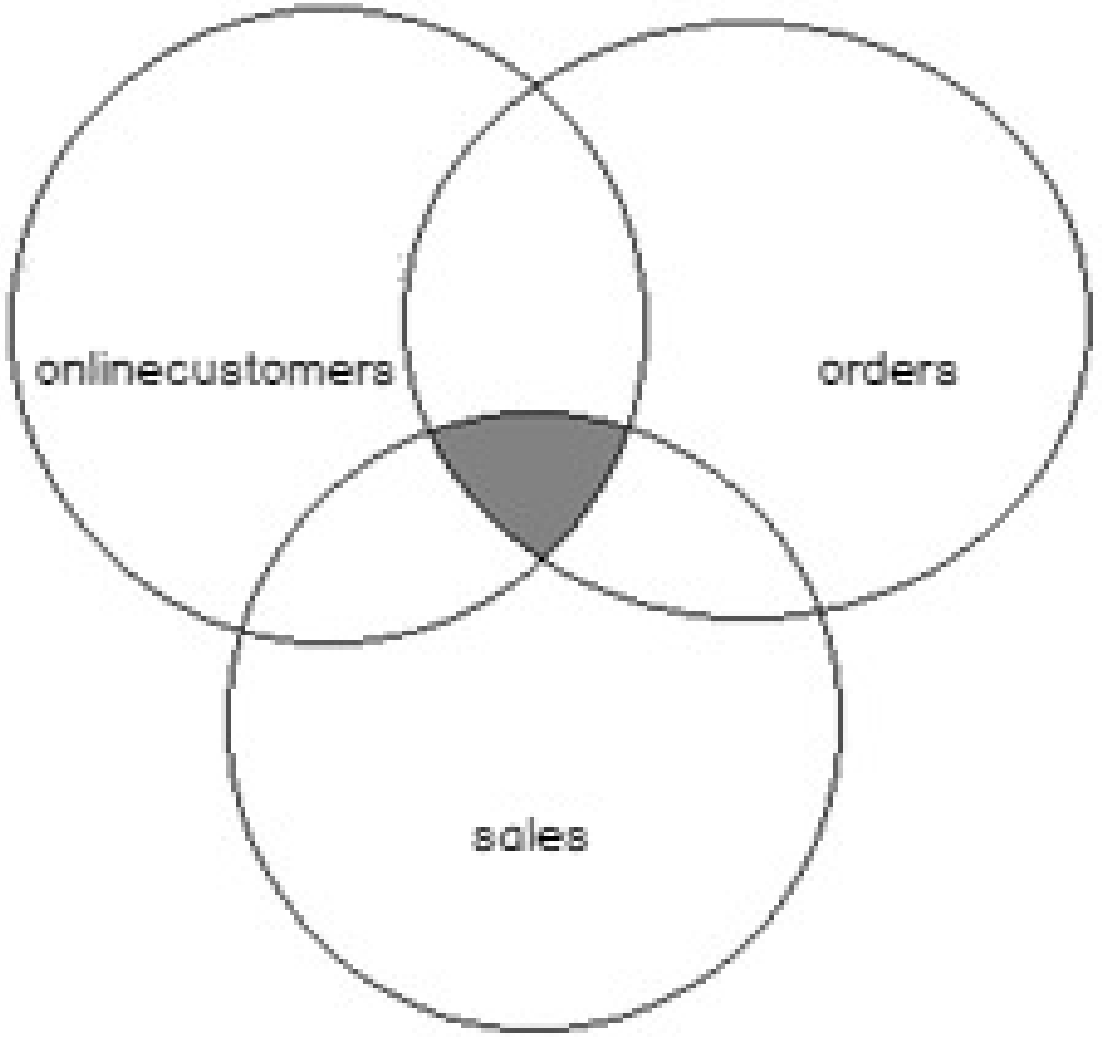
```
SELECT column_name(s)
FROM table1 T1  table1 T2
WHERE condition;
```

```
SELECT A.CustomerName AS CustomerName1 ,
B.CustomerName AS CustomerName2,  A.City
FROM Customers A , Customers B
WHERE A.CustomerID <> B.CustomerID
AND A.City = B.City
ORDER BY A.City;
```

```
SELECT A.CustomerName AS CustomerName1 ,
B.CustomerName AS CustomerName2,  A.City
FROM Customers A
Left join customers B
On a.customerID = b.customerID
```

MULTIPLE JOINS

```
SELECT customerName
customercity  customermail
salestotal

FROM onlinecustomers AS oc

   INNER JOIN

   orders AS o

   ON oc.customerid = o.customerid

   INNER JOIN

   sales AS s

   ON o.orderId = s.orderId
```



MULTIPLE JOINS

customerid	CustomerName	CustomerCity	Customermail
1	Salvador	Philadelphia	tyiptqo.wethls@chttw.org
2	Gilbert	San Diego	rrvyy.wdumos@lklkj.org
3	Ernest	New York	ymuea.pnxkukf@dwv.org
4	Stella	Phoenix	xvsfzp.rjhtni@rdn.com
5	Jorge	Los Angeles	oykbo.vlxopp@nmwhv.org
6	Jerome	San Antonio	wkabc.ofmhetq@gtmh.co
7	Edward	Chicago	wguexiymy.nnbdgpc@juc.co

onlinecustomers AS oc
INNER JOIN
orders AS o
ON oc.customerid = o.customerid

orderId	customerid	ordertotal	discountrate	orderdate
1	3	1910.64	5.49	2019-12-03 00:00:00.000
2	4	150.89	15.33	2019-06-11 00:00:00.000
3	5	912.55	13.74	2019-09-15 00:00:00.000
4	7	418.24	14.53	2019-05-28 00:00:00.000
5	55	512.55	13.74	2019-06-15 00:00:00.000
6	57	118.24	14.53	2019-12-28 00:00:00.000

INNER JOIN
sales AS s
ON o.orderId = s.orderId

salesId	orderId	salestotal
1	3	370.95
2	4	882.13
3	12	370.95
4	13	882.13
5	55	170.95
6	57	382.13

Practice JOINS

HR

Employee

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	manager_id	department_id
100	Steven	King	steven.king@sqltutorial.org	515.123.4567	1987/06/17	4	24000	0	9
101	Neena	Kochhar	neena.kochhar@sqltutorial.org	515.123.4568	1989/09/21	5	17000	100	9
102	Lex	De Haan	lex.de haan@sqltutorial.org	515.123.4569	1993/01/13	5	17000	100	9
103	Alexander	Hunold	alexander.hunold@sqltutorial.org	590.423.4567	1990/01/03	9	9000	102	6

Country

country_id	country_name	region_id
AR	Argentina	2
AU	Australia	3
BE	Belgium	1
BR	Brazil	2
CA	Canada	2

Department

department_id	department_name	location_id
1	Administration	1700
2	Marketing	1800
3	Purchasing	1700
4	Human Resources	2400

Dependant

dependent_id	first_name	last_name	relationship	employee_id
1	Penelope	Gietz	Child	206
2	Nick	Higgins	Child	205
3	Ed	Whalen	Child	200
4	Jennifer	King	Child	100
5	Johnny	Kochhar	Child	101

Jobs

job_id	job_title	min_salary	max_salary
1	Public Accountant	4200	9000
2	Accounting Manager	8200	16000
3	Administration Assistant	3000	6000

Location

location_id	street_address	postal_code	city	state_province	country_id
1400	2014 Jabberwocky Rd	26192	Southlake	Texas	US
1500	2011 Interiors Blvd	99236	South San Francisco	California	US
1700	2004 Charade Rd	98199	Seattle	Washington	US

Region

region_id	region_name
1	Europe
2	Americas
3	Asia
4	Middle East and Africa

Practice JOINS – RESTAURANT

Sales

customer_id	order_date	product_id
A	2021-01-01	1
A	2021-01-01	2
A	2021-01-07	2
A	2021-01-10	3
A	2021-01-11	3
A	2021-01-11	3
B	2021-01-01	2
B	2021-01-02	2
B	2021-01-04	1

Menu

product_id	product_name	price
1	sushi	10
2	curry	15
3	ramen	12

Members

customer_id	join_date
A	2021-01-07
B	2021-01-09

1. How much money did each member spend at the restaurant?
2. What is the most purchased item on the menu and how many times was it purchased by all customers?
3. What is the total items and amount spent for each member before they became a member? **

Practice JOINS - RESTAURANT

```
CREATE SCHEMA restaurant;
USE restaurant;

CREATE TABLE sales (
  "customer_id" VARCHAR(1)
  "order_date" DATE
  "product_id" INTEGER
);

INSERT INTO sales
  ("customer_id"  "order_date"  "product_id")
VALUES
  ( A    2021-01-01    1 )
  ( A    2021-01-01    2 )
  ( A    2021-01-07    2 )
  ( A    2021-01-10    3 )
  ( A    2021-01-11    3 )
  ( A    2021-01-11    3 )
  ( B    2021-01-01    2 )
  ( B    2021-01-02    2 )
  ( B    2021-01-04    1 )
  ( B    2021-01-11    1 )
  ( B    2021-01-16    3 )
  ( B    2021-02-01    3 )
  ( C    2021-01-01    3 )
  ( C    2021-01-01    3 )
  ( C    2021-01-07    3 );
```

```
CREATE TABLE menu (
  "product_id" INTEGER
  "product_name" VARCHAR(5)
  "price" INTEGER
);

INSERT INTO menu
  ("product_id"  "product_name"  "price")
VALUES
  ( 1    sushi    10 )
  ( 2    curry    15 )
  ( 3    ramen    12 );


CREATE TABLE members (
  "customer_id" VARCHAR(1)
  "join_date" DATE
);

INSERT INTO members
  ("customer_id"  "join_date")
VALUES
  ( A    2021-01-07 )
  ( B    2021-01-09 );
```

AIRBNB

Practice JOINS

1. *Display each employee's first and last name, as well as the job titles to which they belong.*
2. *Now include the department titles of each employee in the same results.*
3. *Display each employee's first and last name as well as the department and department's city where is located.*
4. *Determine which employees are managers on the employees table show the full name of the employee and the Manger s first name.*
5. *Show all the employees that work in the same department and they order by their department.*
6. *What is the total salary for paid by department?*

Practice ***

1. Show the name, and surname of each employee, together with how many children they have
2. Show the firstname, their salary amount, and the difference between their salary and the min_amount for their position – thus, how much are they getting more than the minimum amount for that role?
3. How many employees are based in each region?