JOINS in SQL - How they work? with examples

What is a JOIN?

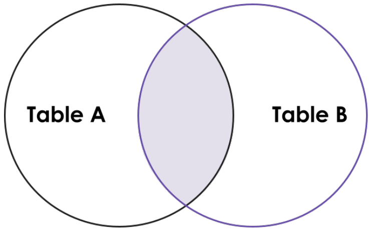
A JOIN is used to combine rows from two or more tables, based on a related column between them.

Why do we need JOINs?

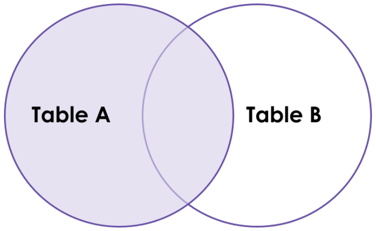
The purpose of JOINs in SQL is to access data from multiple tables based on logical relationships between them.

Types of JOIN

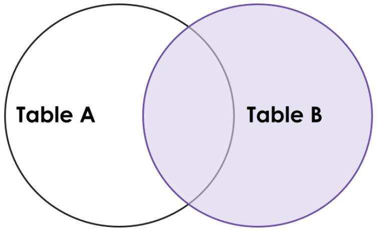
INNER JOIN



LEFT JOIN



RIGHT JOIN



Examples	Concept
Example-1	Inner Join
Example-2	Left Join
Example-3	Right Join
Example-4	Self Join
Example-5	Union
Example-6	Union All
Example-7	Cross Join

The fundamentals for mapping

Understanding how mapping works is essential to understand the different kinds of joins that can be done in SQL

Types of Mapping

i. **One to One mapping (1:1)**

table_A	
column_1	column_2
1	x
2	y
3	z
4	w
5	o

table_B	
column_3	column_4
1	a
2	b
3	c
4	d
5	e

join result

column_1	column_2	column_3	column_4
1	x	1	a
2	y	2	b
3	z	3	c
4	w	4	d
5	o	5	e

ii. **One to Many mapping (1:*)**

table_A	
column_1	column_2
1	x
2	y
3	z
4	w
5	o

table_B	
column_3	column_4
1	a
1	b
3	c
3	d
5	e

join result

column_1	column_2	column_3	column_4
1	x	1	a
1	x	1	b
3	z	3	c
3	z	3	d
5	o	5	e

iii. Many to One mapping (*:1)

table_A	
column_1	column_2
1	x
1	y
2	z
2	w
5	o

table_B	
column_3	column_4
1	a
2	b
3	c
4	d
5	e

column_1	column_2	column_3	column_4
1	x	1	a
1	y	1	a
2	z	2	b
2	w	2	b
5	o	5	e

iv. Many to Many mapping (*:*)

table_A	
column_1	column_2
1	x
1	y
2	z
2	w
5	o

table_B	
column_3	column_4
1	a
1	b
1	c
5	d
5	e

column_1	column_2	column_3	column_4
1	x	1	a
1	x	1	b
1	x	1	c
1	y	1	a
1	y	1	b
1	y	1	c
5	o	5	d
5	o	5	e

1 Find all the students who have football and cricket as their hobby.

stud_details		
stud_id	name	hobby_id
1001	John	1
1002	Marcus	4
1003	Robert	0
1004	Luke	5
1005	Ryan	5

hobby_details	
hobby_id	hobby
1	Cycling
2	Swimming
3	Running
4	Cricket
5	Football

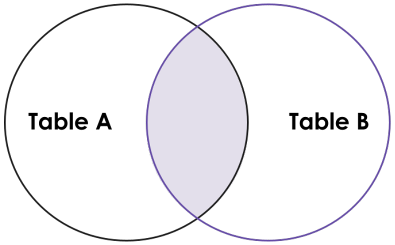
Let's first find the matching records using the common column - hobby_id

stud_details		
stud_id	name	hobby_id
1001	John	1
1002	Marcus	4
1003	Robert	0
1004	Luke	5
1005	Ryan	5

hobby_details	
hobby_id	hobby
1	Cycling
2	Swimming
3	Running
4	Cricket
5	Football

Retrieve the matching records from both tables.

stud_id	name	hobby_id	hobby
1001	John	1	Cycling
1002	Marcus	4	Cricket
1004	Luke	5	Football
1005	Ryan	5	Football



Now, let's find the students with cricket and footbal as hobby

INNER JOIN gets the common records from both tables

stud_id	name	hobby_id	hobby
1002	Marcus	4	Cricket
1004	Luke	5	Football
1005	Ryan	5	Football

How can I write a query to get this answer programmatically?

```
SELECT stu.*, hob.hobby
FROM
   stud_details as stu INNER JOIN hobby_details as hob
   ON stu.hobby_id = hob.hobby_id
WHERE hob.hobby IN ('Cricket', 'Football');
```

2 Find the total number of hobbies each student has

stud_details

stud_id	name	age
1001	John	10
1002	Marcus	11
1003	Robert	9
1004	Luke	10
1005	Ryan	11
1006	Shaw	9

hobby_details

stud_id	hobby
1001	Cycling
1001	Painting
1002	Swimming
1000	Running
1004	Football
1005	Football
1005	Rugby

Let's combine the two tables to find out more about the students and their interests.

stud_details

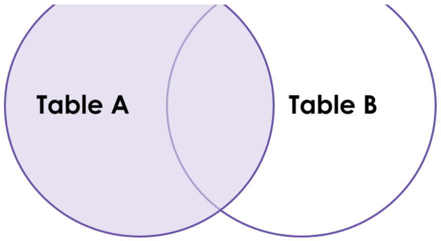
stud_id	name	age
1001	John	10
1002	Marcus	11
1003	Robert	9
1004	Luke	10
1005	Ryan	11
1006	Shaw	9

hobby_details

stud_id	hobby
1001	Cycling
1001	Painting
1002	Swimming
1000	Running
1004	Football
1005	Football
1005	Rugby

stud_id	name	age	hobby
1001	John	10	Cycling
1001	John	10	Painting

1002	Marcus	11	Swimming
1003	Robert	9	*Null*
1004	Luke	10	Football
1005	Ryan	11	Football
1005	Ryan	11	Rugby
1006	Shaw	9	*Null*



Now let's find the number of hobbies for each student.

stud_id	name	age	no_of_hobbies
1001	John	10	2
1002	Marcus	4	1
1003	Robert	3	0
1004	Luke	6	1
1005	Ryan	6	1
1006	Chris	5	0

How can I write a query to get this answer programmatically?

```
SELECT stu.*, count(hob.hobby) as no_of_hobbies
FROM
  stud_details as stu LEFT JOIN hobby_details as hob
  ON stu.stud_id=hob.stud_id
GROUP BY stu.stud_id, stu.name, stu.age;
```

3 **Find the total number of students for each hobby.**

stud_details

stud_id	name	hobby_id
1001	John	1
1001	John	2
1002	Marcus	10
1003	Robert	6
1003	Robert	5
1004	Luke	6
1005	Ryan	6

hobby_details

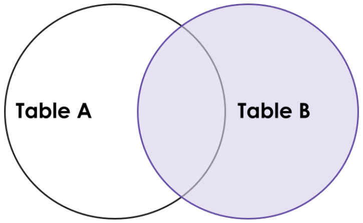
hobby_id	hobby
1	Cycling
2	Painting
3	Swimming
4	Running
5	Cricket
6	Football
7	Rugby

Let's join the two tables to find the number of students for each hobby.

stud_id	name	hobby_id
1001	John	1
1001	John	2
1002	Marcus	10
1003	Robert	6
1003	Robert	5
1004	Luke	6
1005	Ryan	6

hobby_id	hobby
1	Cycling
2	Painting
3	Swimming
4	Running
5	Cricket
6	Football
7	Rugby

stud_id	name	hobby_id	hobby
1001	John	1	Cycling
1001	John	2	Painting
Null	*Null*	3	Swimming
Null	*Null*	4	Running
1003	Robert	5	Cricket
1003	Robert	6	Football
1004	Luke	6	Football
1005	Ryan	6	Football



Null	*Null*	7	Rugby

Now, let's find the number of students for each hobby.

hobby_id	hobby	no_of_students
1	Cycling	1
2	Painting	1
3	Swimming	0
4	Running	0
5	Cricket	1
6	Football	3
7	Rugby	0

How can I write a query to get this answer programmatically?

```
SELECT hob.hobby_id, hob.hobby, COUNT(hob.hobby) as no_of_students
FROM
  hobby_details as hob RIGHT JOIN stud_details as stu
  ON hob.hobby_id=stu.hobby_id
GROUP BY hob.hobby_id;
```

NOTE:

The LEFT JOIN includes all records from the left side and matched rows from the right table, whereas RIGHT JOIN returns all rows from the right side and matched rows from the left table

4 Write a query to find the representative details of each student.

stud_details

id	name	age	representative_id
1001	John	18	1003
1002	Marcus	20	1003
1003	Robert	50	Null
1004	Luke	15	1005
1005	Ryan	45	Null
1006	Chris	19	1003

stud_details

id	name	age	representative_id
1001	John	18	1003
1002	Marcus	20	1003
1003	Robert	50	*Null*
1004	Luke	15	1005
1005	Ryan	45	*Null*
1006	Chris	19	1003

Now, let's find the representative's details by referring to the same table.

stud_details

id	name	age	representative_id
1001	John	18	1003
1002	Marcus	20	1003
1003	Robert	50	*Null*
1004	Luke	15	1005
1005	Ryan	45	*Null*
1006	Chris	19	1003

stud_details

id	name	age	representative_id
1001	John	18	1003
1002	Marcus	20	1003
1003	Robert	50	Null
1004	Luke	15	1005
1005	Ryan	45	Null
1006	Chris	19	1003

Let's use the above reference to find the representative details of each student.

name	age	rep_name	rep_age
John	18	Robert	50
Marcus	20	Robert	50
Luke	15	Ryan	45

Chris	19	Robert	50

Each occurrence of the table must be given a different alias.

Each column reference must be preceded with an appropriate table alias.

How can I write a query to get this answer programmatically?

```
SELECT stu.name , stu.age, rep.name AS rep_name, rep.age AS rep_age
FROM
   student_details AS stu INNER JOIN student_details AS rep
   ON stu.representative_id = rep.id;
```

5 Find all the products that are in stock.

truck_accessories			
product_id	name	quantity	cost_per_unit
1001	Gear box	5	3000
1002	Tyre	10	800
1003	Windshield	20	500
1004	Steering	10	1500
1005	Tank	0	2000

car_accessories			
product_id	name	quantity	cost_per_unit
1001	Horn	20	250
1002	Wheel	10	1100
1003	Steering	0	1200
1004	Pedal	8	600
1005	Gear box	12	2000

Let's find all the accessories that are in stock for the truck.

name
Gear box
Tyre
Windshield
Steering

Let's find all the accessories that are in stock for the car.

name
Horn
Wheel
Pedal
Gear box

Now, let's merge the data to find all the available accessories.

name
Gear box
Tyre
Windshield
Steering
Horn
Wheel
Pedal

How can I write a query to get this answer programmatically?

```
SELECT name FROM truck_accessories WHERE quantity >= 1
   UNION
SELECT name FROM car_accessories WHERE quantity >= 1;
```

name
Tyre
Windshield
Steering
Tank
Horn
Wheel
Pedal
Gear box

6 Find out which products cost more than $1,000 per unit.

truck_accessories			
product_id	name	quantity	cost_per_unit
1001	Gear box	5	3000
1002	Tyre	10	800
1003	Windshield	20	500
1004	Steering	10	1500
1005	Tank	0	2000

car_accessories			
product_id	name	quantity	cost_per_unit
1001	Horn	20	250
1002	Wheel	10	1100
1003	Steering	0	1200
1004	Pedal	8	600
1005	Gear box	12	2000

Let's look for all the truck accessories that cost more than $1,000.

name
Gear box
Steering
Tank

Let's look for all the car accessories that cost more than $1,000.

name
Wheel
Steering
Gear box

Now, let's merge the data to find all the available accessories.

name

Gear box
Steering
Tank
Wheel
Steering
Gear box

How can I write a query to get this answer programmatically?

```
SELECT name FROM truck_accessories WHERE cost_per_unit > 1000
   UNION ALL
SELECT name FROM car_accessories WHERE cost_per_unit > 1000;
```

name
Gear box
Steering
Tank
Wheel
Steering
Gear box

7 Find all of the possible car and colour combinations.

car_details

car_id	name	color_id
1001	Toyota	1
1002	Honda	2
1003	Renault	3
1004	Hyundai	4

color_details

color_id	color
1	Black
2	Red
3	Blue
4	Silver

Let's find all the possible combinations.

car_id	name	color_id	color_id	color
1001	Toyota	1	1	Black
1002	Honda	2	1	Black
1003	Renault	3	1	Black
1004	Hyundai	4	1	Black
1001	Toyota	1	2	Red
1002	Honda	2	2	Red
1003	Renault	3	2	Red
1004	Hyundai	4	2	Red
1001	Toyota	1	3	Blue
1002	Honda	2	3	Blue
1003	Renault	3	3	Blue
1004	Hyundai	4	3	Blue
1001	Toyota	1	4	Silver
1002	Honda	2	4	Silver
1003	Renault	3	4	Silver
1004	Hyundai	4	4	Silver

Note:

❑ Table A has m number of rows

❑ Table B has n number of rows

❑ The output will consist of mxn number of rows

How can I write a query to get this answer programmatically?

```
SELECT *
FROM
    car_details CROSS JOIN color_details ;
```