

# INTRODUCTION TO SQL

## **Join**

---

# JOINS in SQL

Connect two or more tables

## Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

## Company

CName	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

What is the Connection between them ?

# TYPES OF JOINS

- 1 • INNER JOIN
- 2 • LEFT OUTER JOIN
- 3 • RIGHT OUTER JOIN
- 4 • FULL OUTER JOIN

# Inner Join

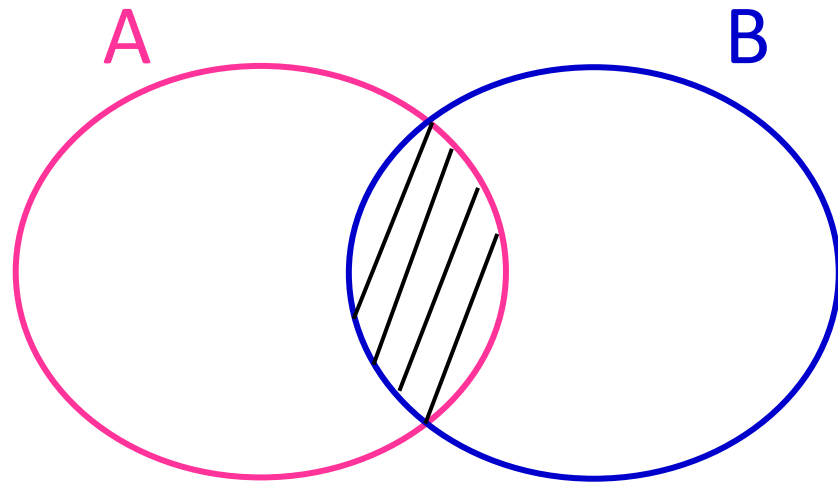
A **Inner Join** is a join operation that joins two tables by their common column. This operation is similar to the set intersection of two tables.

Select \* from A

**INNER JOIN**

B

**ON** A.ID=B.ID



# Example of INNER JOIN

## Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

Cname	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

```
SELECT Product.PName, Product.Price
FROM Product INNER JOIN Company
ON Product.Manufacturer=Company.CName
WHERE Company.Country='Japan'
AND Product.Price <= 200
```

PName	Price
SingleTouch	\$149.99

# Outer Join

An **Outer Join** is a join operation that includes rows that have a match, plus rows that do not have a match in the other table.

- Used when rows from one table should be part of the result there are no related rows in a second table
- Direction must be specified
  - Left/Right specify which table has the rows which should always be included
  - Full specifies that rows from **both** tables should be included even if no match between rows

# Left Outer Join

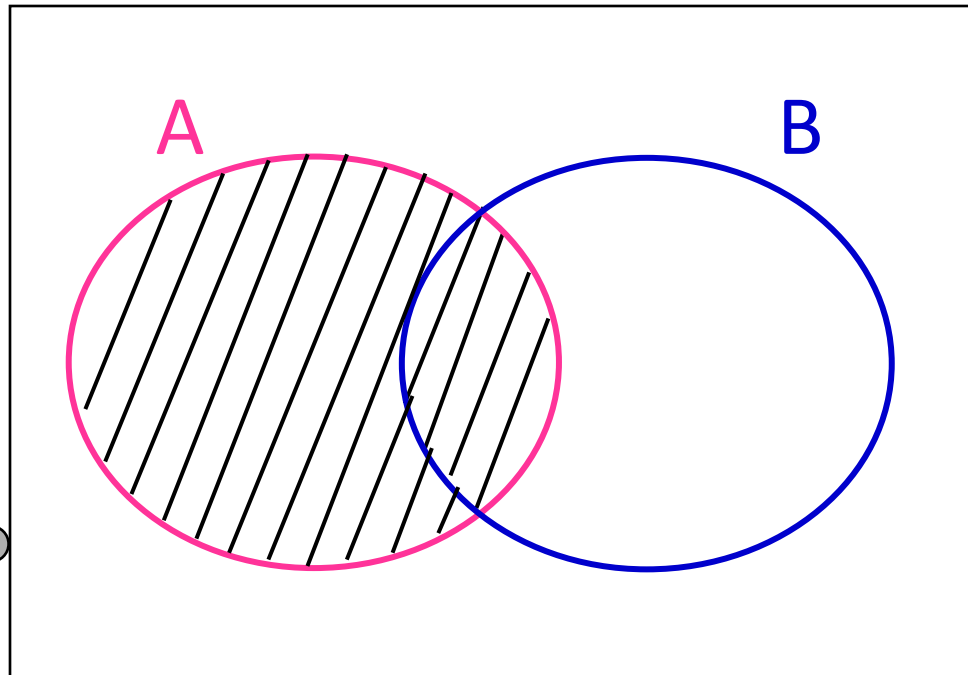
**Left Outer Join** returns all matched rows, plus all unmatched rows from the table on the left of the join clause (use nulls in fields of non-matching tuples)

Select \* from A

**LEFT OUTER  
JOIN**

B

**ON** A.ID=B.ID



# Left Outer Join (Example)

---

Sailor			
<u>sid</u>	sname	rating	age
22	Dustin	7	45.0
31	Lubber	8	55.5
95	Bob	3	63.5

Reserves		
<u>sid</u>	<u>bid</u>	<u>day</u>
22	101	10/10/96
95	103	11/12/96

**SELECT** s.sid, s.name, r.bid  
**FROM** Sailors s **LEFT OUTER JOIN** Reserves r  
**ON** s.sid = r.sid

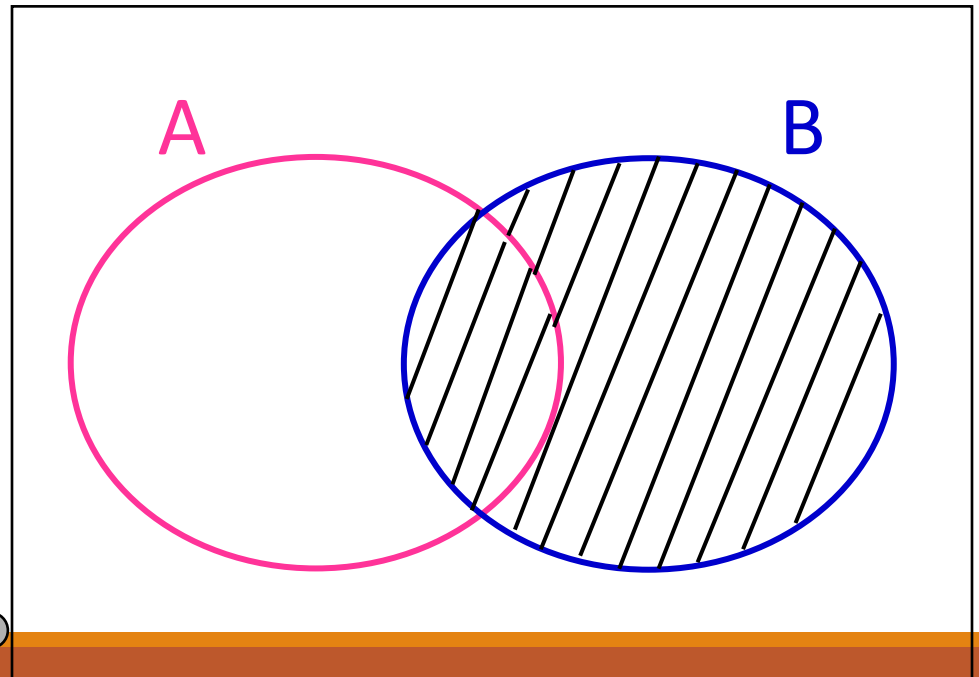
r.bid	s.name	s.sid
101	Dustin	22
103	Lubber	95
NULL	Bob	31



# Right Outer Join

**Right Outer Join** returns all matched rows, plus all unmatched rows from the table on the right of the join clause (use nulls in fields of non-matching tuples)

Select \* from A  
**RIGHT OUTER**  
**JOIN**  
B  
**ON** A.ID=B.ID



# Right Outer Join (EXAMPLE)

Reserves		
<u>sid</u>	<u>bid</u>	<u>day</u>
22	101	10/10/96
95	103	11/12/96

Boats		
<u>bid</u>	bname	color
101	Interlake	blue
102	Interlake	red
103	Clipper	green
104	Marine	red

**SELECT** r.sid, b.bid, b.name  
**FROM** Reserves r **RIGHT OUTER JOIN** Boats b  
**ON** r.bid = b.bid

r.sid	b.bid	b.name
22	101	Interlake
NULL	102	Interlake
95	103	Clipper
NULL	104	Marine

# Full Outer Join

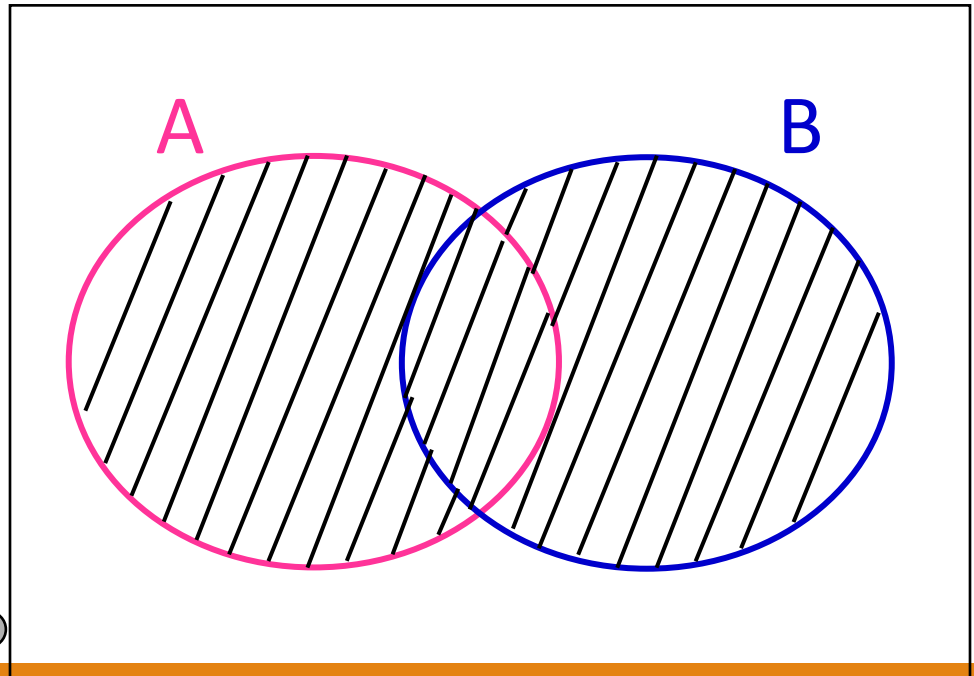
**Full Outer Join** returns all (matched or unmatched) rows from the tables on both sides of the join clause (use nulls in fields of non-matching tuples)

Select \* from A

**FULL OUTER  
JOIN**

B

**ON** A.ID=B.ID



# Full Outer Join

Reserves		
<u>sid</u>	<u>bid</u>	<u>day</u>
22	101	10/10/96
95	103	11/12/96

Boats		
<u>bid</u>	bname	color
101	Interlake	blue
102	Interlake	red
103	Clipper	green
104	Marine	red

**SELECT** r.sid, b.bid, b.name  
**FROM** Reserves r **FULL OUTER JOIN** Boats b  
**ON** r.bid = b.bid

r.sid	b.bid	b.name
22	101	Interlake
NULL	102	Interlake
95	103	Clipper
NULL	104	Marine

**Note:** in the same because bid is a foreign key in reserves, reservations have a couple in b

# Full Outer Join

Reserves		
<u>sid</u>	<u>bid</u>	<u>day</u>
22	101	10/10/96
95	103	11/12/96

Boats		
<u>bid</u>	bname	color
101	Interlake	blue
102	Interlake	red
103	Clipper	green
104	Marine	red

**SELECT** r.sid, b.bid, b.name  
**FROM** Reserves r **FULL OUTER JOIN** Boats b  
**ON** r.bid = b.bid

r.sid	b.bid	b.name
22	101	Interlake
NULL	102	Interlake
95	103	Clipper
NULL	104	Marine

**Note:** in the same because bid is a foreign key in reserves, reservations have a couple in b

# Cross Join

Typically used to generate lots of data quickly  
Match each row from table 1 with every row from table 2

Select \* from  
Table 1

**CROSS JOIN**

Table 2

❖ Result is (table 1 row count)\*  
(table 2 row count)

# CARTESIAN PRODUCT/ CROSS JOIN

**Account Type**

AccountType ID	Name
111	Current
112	Saving
113	Mutual

**Account**

Account#	AccountType ID	Account Amount
11	111	3000
12	112	2000

```
SELECT Account#, AccountTypeID FROM  
Account  
CROSS JOIN  
AccountType
```



Account#	Account TypeID
11	111
11	112
11	113
12	111
12	112
12	113