

# Database Lab (CSE3104)

## SQL Join

---



# What is SQL Join?



# Join?



Statement or a Clause



Accumulate/Combine data or tuples from two or more entities



Should be an identical attribute to perform join between tables

Orders
<u>OrderID</u>
OrderDateTime
CustomerID
Tax
Shipping



OrderItems
<u>OrderItemID</u>
OrderID
ProductID
Quantity



OrderItems
<u>OrderItemID</u>
OrderID
ProductID
Quantity



ProductTypes
<u>ProductTypeID</u>
ProductTypeName



## Join?



SELECT A, B, E FROM *tableA*, *tableB* WHERE *tableA.AID* = *tableB.D*

tableA			
AID	A	B	C
1	a	11	aa
2	b	12	bb
3	c	13	cc
4	d	14	dd
5	e	15	ee
6	f	16	ff

tableB		
BID	D	E
100	5	c
101	4	c
102	4	d
103	1	f
104	2	a



A	B	E
a	11	f
b	12	a
d	14	c
d	14	d
e	15	c

# **Types of Join**



## Types of Join



Inner Join



Left Join



Right Join



Full Join



Self Join

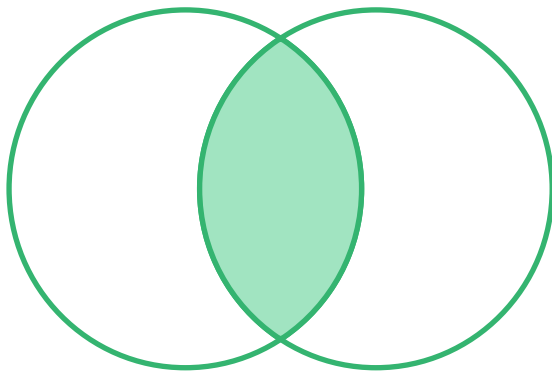


Cartesian Join



## Inner Join (EQUIJOIN)

- ✓ Select all the tuples from the tables as long as the specified conditions satisfied
- ✓ Column values for each matched pair of rows of two or more tables are combined

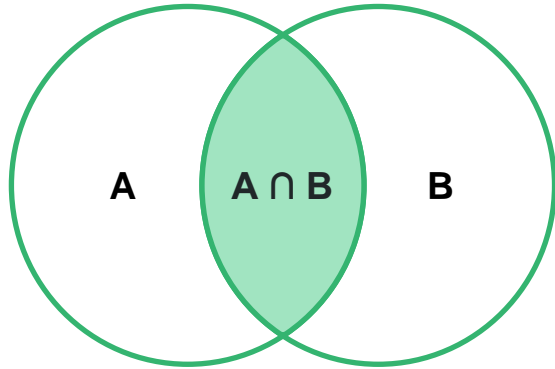




## Inner Join (Representation)



Common part of the two sets or entities



```
SELECT column1, column2, ..., columnn
FROM firstTable INNER JOIN secondTable
ON firstTable.columnName = secondTable.columnName
WHERE conditions
GROUP BY .....
ORDER BY ..... ;
```





## Inner Join (Example)



SELECT A, B, E FROM *tableA* INNER JOIN *tableB* ON *tableA*.AID = *tableB*.D

tableA			
AID	A	B	C
1	a	11	aa
2	b	12	bb
3	c	13	cc
4	d	14	dd
5	e	15	ee
6	f	16	ff

tableB		
BID	D	E
100	5	c
101	4	c
102	4	d
103	1	f
104	2	a



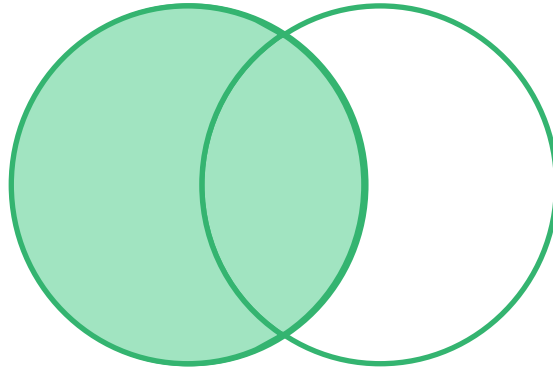
A	B	E
a	11	f
b	12	a
d	14	c
d	14	d
e	15	c



## Left Join

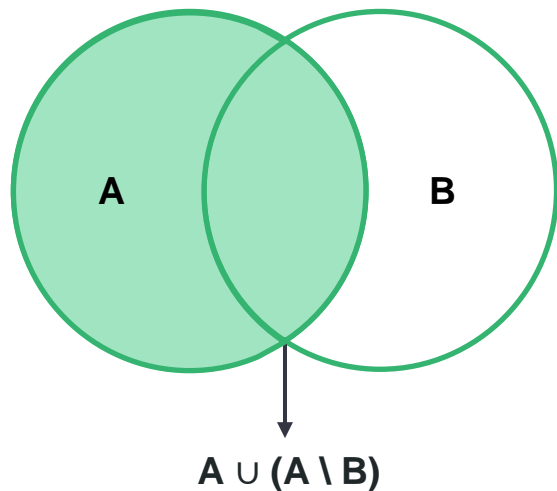


Select all the tuples from the left table and the matched tuples from the right table



## ➡ Left Join (Representation)

✓ Common part of the two sets or entities



```
SELECT column1, column2, ..., columnn
FROM firstTable LEFT JOIN secondTable
ON firstTable.columnName = secondTable.columnName
WHERE conditions
GROUP BY .....
ORDER BY ..... ;
```



## Left Join (Example)



`SELECT A, B, E FROM tableA LEFT JOIN tableB ON tableA.AID = tableB.D`

tableA			
AID	A	B	C
1	a	11	aa
2	b	12	bb
3	c	13	cc
4	d	14	dd
5	e	15	ee
6	f	16	ff

tableB		
BID	D	E
100	5	c
101	4	c
102	4	d
103	1	f
104	2	a



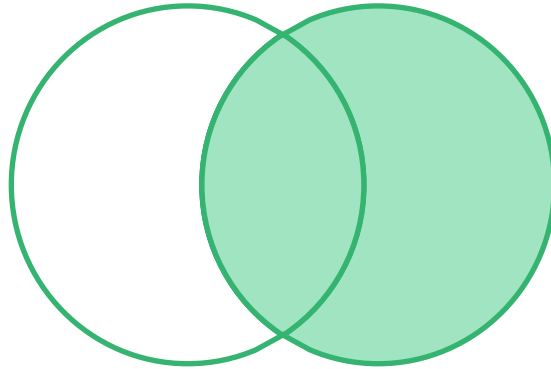
A	B	E
a	11	f
b	12	a
c	13	NULL
d	14	c
d	14	d
e	15	c
f	16	NULL



## Right Join



Select all the tuples from the right table and the matched tuples from the left table

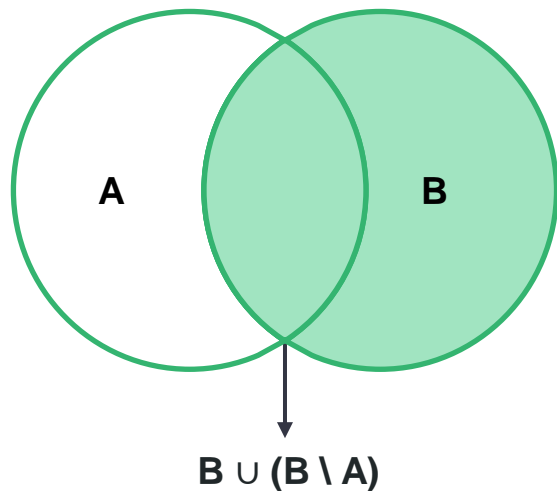




## Right Join (Representation)



Common part of the two sets or entities



```
SELECT column1, column2, ..., columnn
FROM firstTable RIGHT JOIN secondTable
ON firstTable.columnName = secondTable.columnName
WHERE conditions
GROUP BY .....
ORDER BY ..... ;
```



## Right Join (Example)



SELECT A, B, E FROM *tableA* RIGHT JOIN *tableB* ON *tableA*.AID = *tableB*.D

tableA			
AID	A	B	C
1	a	11	aa
2	b	12	bb
3	c	13	cc
4	d	14	dd
5	e	15	ee
6	f	16	ff

tableB		
BID	D	E
100	5	c
101	4	c
102	4	d
103	1	f
104	2	a



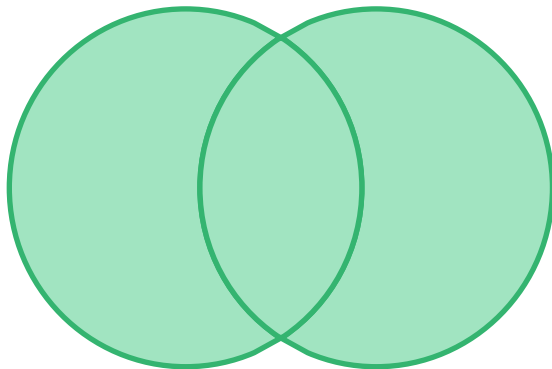
A	B	E
e	15	c
d	14	c
d	14	d
a	11	f
b	12	a



## Full Join



Contains all the record from all the entities



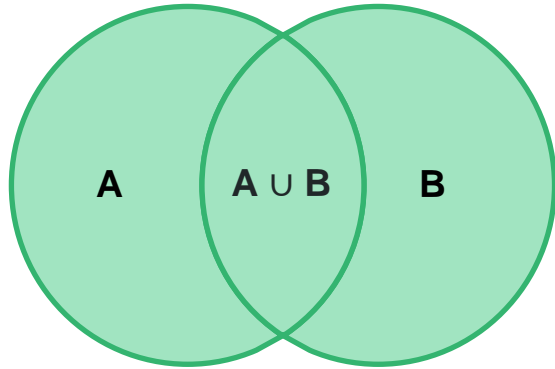




## Full Join (Representation)



Common part of the two sets or entities



```
SELECT column1, column2, ..., columnn
FROM firstTable FULL JOIN secondTable
ON firstTable.columnName = secondTable.columnName
WHERE conditions
GROUP BY .....
ORDER BY ..... ;
```



## Full Join (Example)



SELECT A, B, E FROM *tableA* FULL JOIN *tableB*  
ON *tableA*.AID = *tableB*.D

tableA			
AID	A	B	C
1	a	11	aa
2	b	12	bb
3	c	13	cc
4	d	14	dd
5	e	15	ee
6	f	16	ff

tableB		
BID	D	E
100	5	c
101	4	c
102	4	d
103	1	f
104	2	a



A	B	E
a	11	f
b	12	a
c	13	NULL
d	14	c
d	14	d
e	15	c
f	16	NULL
e	15	c
d	14	c
d	14	d
a	11	f
b	12	a



## Full Join (Exception)



MySQL does not support Full Join



Instead of Full Join, **UNION ALL** can be used

```
SELECT column1, column2, ..., columnn FROM firstTable LEFT JOIN secondTable  
ON firstTable.columnName = secondTable.columnName  
WHERE conditions GROUP BY conditions ORDER BY conditions
```

UNION ALL

```
SELECT column1, column2, ..., columnn FROM firstTable RIGHT JOIN secondTable  
ON firstTable.columnName = secondTable.columnName  
WHERE conditions GROUP BY conditions ORDER BY conditions
```



# Cartesian Join or Cross Join



Returns the Cartesian product of the two entities

```
SELECT column1, column2, ...  
FROM table1, table2 ;
```

table (a)		
ID	A	B
100	a	1
101	b	2

table (b)		
ID	A	B
100	a	1
101	b	2



Cartesian product			
A	B	A	B
a	1	a	1
a	1	b	2
b	2	a	1
b	2	b	2



## Self Join

- ✓ Used to join a table to itself as if the table were two tables
- ✓ It is done by renaming at least one of the two entities
- ✓ **Self Join = Cartesian product + Conditions**

```
SELECT a.column, b.column, ...
FROM table a, table b
WHERE a.commonField = b.commonField
GROUP BY .....
ORDER BY ..... ;
```



## Self Join

```
SELECT a.column, b.column, ...  
FROM example as a, example as b  
WHERE a.B <> b.B
```

example (a)		
ID	A	B
100	a	1
101	b	2

example (b)		
ID	A	B
100	a	1
101	b	2

Cartesian product			
A	B	A	B
a	1	a	1
a	1	b	2
b	2	a	1
b	2	b	2

*condition*

Output			
A	B	A	B
a	1	b	2
b	2	a	1

# Practice Problems

SQL Join

---



## Practice Problems

**Table - Customer**

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

**Table - Order**

OID	DATE	CUSTOMER_ID	AMOUNT
102	2009-10-08 00:00:00	3	3000
100	2009-10-08 00:00:00	3	1500
101	2009-11-20 00:00:00	2	1560
103	2008-05-20 00:00:00	4	2060





# Practice Problems



Find the output of the following SQL Queries

1. `SELECT ID, NAME, AMOUNT, DATE FROM Customers, Orders;`
2. `SELECT ID, NAME, AMOUNT, DATE FROM Customers INNER JOIN Orders  
ON Customers.ID = Orders.Customer_ID;`
3. `SELECT ID, NAME, AMOUNT, DATE FROM Customers LEFT JOIN Orders  
ON Customers.ID = Orders.Customer_ID;`
4. `SELECT ID, NAME, AMOUNT, DATE FROM Customers RIGHT JOIN Orders  
ON Customers.ID = Orders.Customer_ID;`
5. `SELECT ID, NAME, AMOUNT, DATE FROM Customers FULL JOIN Orders  
ON Customers.ID = Orders.Customer_ID;`
6. `SELECT a.ID, b.NAME, a.SALARY FROM Customers a, Customers b  
WHERE a.SALARY < b.SALARY;`



# Practice Problems (Answer)

1

ID	NAME	AMOUNT	DATE
1	Ramesh	3000	2009-10-08 00:00:00
1	Ramesh	1500	2009-10-08 00:00:00
1	Ramesh	1560	2009-11-20 00:00:00
1	Ramesh	2060	2008-05-20 00:00:00
2	Khilan	3000	2009-10-08 00:00:00
2	Khilan	1500	2009-10-08 00:00:00
2	Khilan	1560	2009-11-20 00:00:00
2	Khilan	2060	2008-05-20 00:00:00
3	kaushik	3000	2009-10-08 00:00:00
3	kaushik	1500	2009-10-08 00:00:00
3	kaushik	1560	2009-11-20 00:00:00
3	kaushik	2060	2008-05-20 00:00:00
4	Chaitali	3000	2009-10-08 00:00:00
4	Chaitali	1500	2009-10-08 00:00:00
4	Chaitali	1560	2009-11-20 00:00:00
4	Chaitali	2060	2008-05-20 00:00:00
5	Hardik	3000	2009-10-08 00:00:00
5	Hardik	1500	2009-10-08 00:00:00
5	Hardik	1560	2009-11-20 00:00:00
5	Hardik	2060	2008-05-20 00:00:00
6	Komal	3000	2009-10-08 00:00:00
6	Komal	1500	2009-10-08 00:00:00
6	Komal	1560	2009-11-20 00:00:00
6	Komal	2060	2008-05-20 00:00:00
7	Muffy	3000	2009-10-08 00:00:00
7	Muffy	1500	2009-10-08 00:00:00
7	Muffy	1560	2009-11-20 00:00:00
7	Muffy	2060	2008-05-20 00:00:00

2

ID	NAME	AMOUNT	DATE
3	kaushik	3000	2009-10-08 00:00:00
3	kaushik	1500	2009-10-08 00:00:00
2	Khilan	1560	2009-11-20 00:00:00
4	Chaitali	2060	2008-05-20 00:00:00

3

ID	NAME	AMOUNT	DATE
1	Ramesh	NULL	NULL
2	Khilan	1560	2009-11-20 00:00:00
3	kaushik	3000	2009-10-08 00:00:00
3	kaushik	1500	2009-10-08 00:00:00
4	Chaitali	2060	2008-05-20 00:00:00
5	Hardik	NULL	NULL
6	Komal	NULL	NULL
7	Muffy	NULL	NULL



# Practice Problems (Answer)

4

ID	NAME	AMOUNT	DATE
3	kaushik	3000	2009-10-08 00:00:00
3	kaushik	1500	2009-10-08 00:00:00
2	Khilan	1560	2009-11-20 00:00:00
4	Chaitali	2060	2008-05-20 00:00:00

5

ID	NAME	AMOUNT	DATE
1	Ramesh	NULL	NULL
2	Khilan	1560	2009-11-20 00:00:00
3	kaushik	3000	2009-10-08 00:00:00
3	kaushik	1500	2009-10-08 00:00:00
4	Chaitali	2060	2008-05-20 00:00:00
5	Hardik	NULL	NULL
6	Komal	NULL	NULL
7	Muffy	NULL	NULL
3	kaushik	3000	2009-10-08 00:00:00
3	kaushik	1500	2009-10-08 00:00:00
2	Khilan	1560	2009-11-20 00:00:00
4	Chaitali	2060	2008-05-20 00:00:00

6

ID	NAME	SALARY
2	Ramesh	1500.00
2	kaushik	1500.00
1	Chaitali	2000.00
2	Chaitali	1500.00
3	Chaitali	2000.00
6	Chaitali	4500.00
1	Hardik	2000.00
2	Hardik	1500.00
3	Hardik	2000.00
4	Hardik	6500.00
6	Hardik	4500.00
1	Komal	2000.00
2	Komal	1500.00
3	Komal	2000.00
1	Muffy	2000.00
2	Muffy	1500.00
3	Muffy	2000.00
4	Muffy	6500.00
5	Muffy	8500.00
6	Muffy	4500.00