Joins
&
Operators

Prepared For: CS527

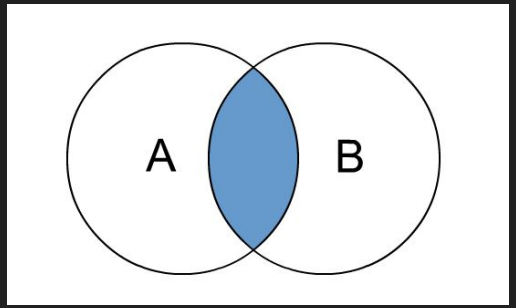
Prepared By: Josh Levine, Pranav Shivkumar, Pratik Mistry, Shounak Rangwala, Swapnil Kamate, Vikhyat Dhamija

Inner Join

- Will select records that have matching values in both tables.

- If there are records in table A that do not have matches in table B or vice-versa, then those rows will not be selected.

- Example: *Select a.*, b.* from a*
  Inner Join b On (a.id = b.id);



Query:

Select products., aisles.* from products Inner Join aisles On (products.aisle_id = aisles.aisle_id);*

product_id	product_name	aisle_id	department_id
1	Chocolate Sandwich	1	19
2	Salt	2	13
3	Tea	3	7
4	Sauce	2	1
5	Pizza	10	5

Products

aisle_id	aisle
1	Dessert
2	Seasons
3	Drinks
4	Medicine
5	Food

Aisles

Query:

Select products., aisles.* from products Inner Join aisles On (products.aisle_id = aisles.aisle_id);*

product_id	product_name	aisle_id	department_id	aisle_id	aisle
1	Chocolate Sandwich	1	19	1	Dessert
2	Salt	2	13	2	Seasons
3	Tea	3	7	3	Drinks
4	Sauce	2	1	2	Seasons

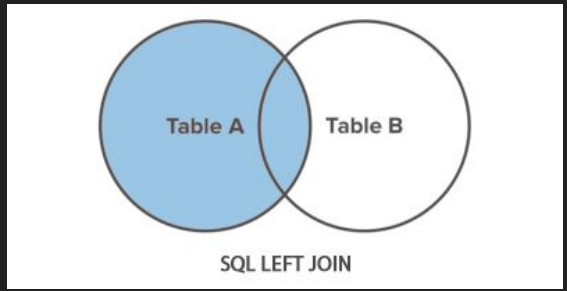
Inner Join Result Table

Left Outer Join

- Will select ALL records from the left table (Table A), and the records that match in right table (Table B).

- If there is no match, the result from the right table is NULL.

- Example: *Select a.*, b.* from a*
  Left Join b On (a.id = b.id);



Query:

Select products., aisles.* from products Left Join aisles On (products.aisle_id = aisles.aisle_id);*

product_id	product_name	aisle_id	department_id
1	Chocolate Sandwich	1	19
2	Salt	2	13
3	Tea	3	7
4	Sauce	2	1
5	Pizza	10	5

Products

aisle_id	aisle
1	Dessert
2	Seasons
3	Drinks
4	Medicine
5	Food

Aisles

Query:

Select products., aisles.* from products Left Join aisles On (products.aisle_id = aisles.aisle_id);*

product_id	product_name	aisle_id	department_id	aisle_id	aisle
1	Chocolate Sandwich	1	19	1	Dessert
2	Salt	2	13	2	Seasons
3	Tea	3	7	3	Drinks
4	Sauce	2	1	2	Seasons
5	Pizza	10	5	NULL	NULL

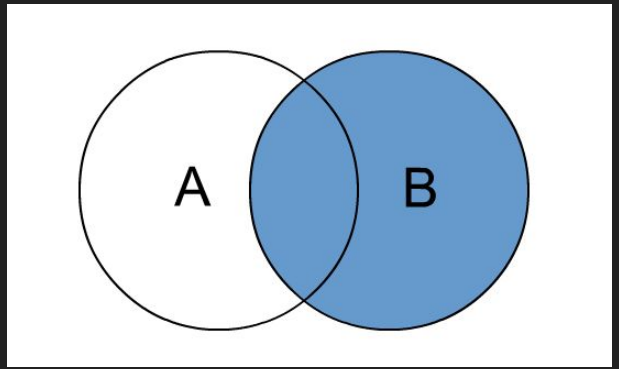
Left Join Result Table

Right Outer Join

- Will select records from the right table (Table B), and records that match from the left table (Table A).

- If there is no match, the result from the left table in NULL.

- Example: *Select a.*, b.* from a*
  Right Join b On (a.id = b.id);



Query:

Select products., aisles.* from products Right Join aisles On (products.aisle_id = aisles.aisle_id);*

product_id	product_name	aisle_id	department_id
1	Chocolate Sandwich	1	19
2	Salt	2	13
3	Tea	3	7
4	Sauce	2	1
5	Pizza	10	5

Products

aisle_id	aisle
1	Dessert
2	Seasons
3	Drinks
4	Medicine
5	Food

Aisles

Query:

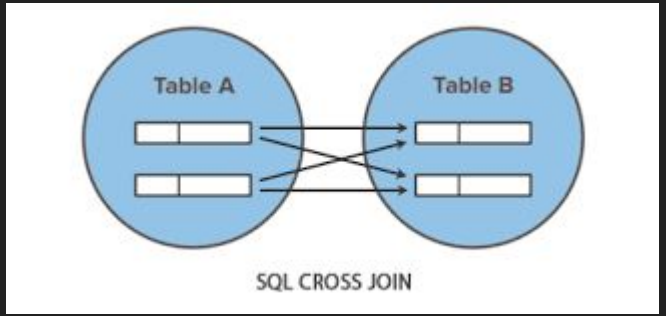
Select products., aisles.* from products Right Join aisles On (products.aisle_id = aisles.aisle_id);*

product_id	product_name	aisle_id	department_id	aisle_id	aisle
1	Chocolate Sandwich	1	19	1	Dessert
2	Salt	2	13	2	Seasons
3	Tea	3	7	3	Drinks
4	Sauce	2	1	2	Seasons
NULL	NULL	NULL	NULL	5	Food

Right Join Result Table

Cross Join

- It produces a result set which is the number of rows in the first table multiplied by the number of rows in the second table. The result is also called Cross Product or Cartesian Product.
- This result set is produced when no “WHERE” clause is used in the join
- A common use for a cross join is to obtain all combinations of items.
- Example: *Select a.*, b.* from a*
  Cross Join b On (a.id = b.id);



Query:

Select p.product_name, p.product_id, a.aisle from products p Cross Join aisles a

product_id	product_name	aisle_id	department_id
1	Chocolate Sandwich	1	19
2	Salt	2	13

Products

aisle_id	aisle
1	Dessert
2	Seasons

Aisles

Query:

Select p.product_name, p.product_id, a.aisle from products p Cross Join aisles a

product_id	product_name	aisle
1	Chocolate Sandwich	Dessert
1	Chocolate Sandwich	Seasons
2	Salt	Dessert
2	Salt	Seasons

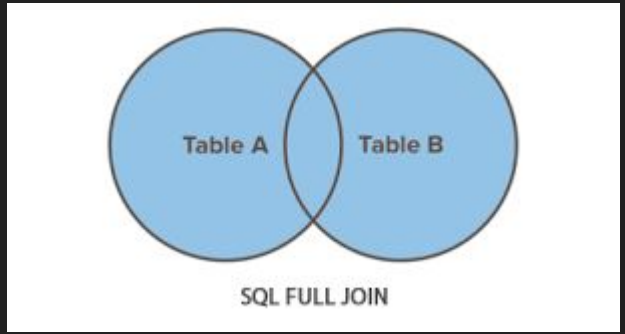
Cross Join Result Table

Full Join

- Full Join returns all matching records from both tables whether the other table matches or not.

- Full Join can return very large data sets.

- Example: *Select a.*, b.* from a*
  Full Join b On (a.id = b.id);



Query:

Select products.*, aisles.* from products Full Join aisles On (products.aisle_id = aisles.aisle_id);

product_id	product_name	aisle_id	department_id
1	Chocolate Sandwich	1	19
2	Salt	2	13
3	Tea	3	7
4	Sauce	2	1
5	Chicken	5	9

Products

aisle_id	aisle
1	Dessert
2	Seasons
3	Drinks
4	Medicine

Aisles

Query:

Select products.*, aisles.* from products Full Join aisles On (products.aisle_id = aisles.aisle_id);

product_id	product_name	aisle_id	department_id	aisle_id	aisle
1	Chocolate Sandwich	1	19	1	Dessert
2	Salt	2	13	2	Seasons
3	Tea	3	7	3	Drinks
4	Sauce	2	1	2	Seasons
5	Chicken	5	9	NULL	NULL
NULL	NULL	NULL	NULL	4	Medicine

Full Join Result Table

Self Join

- Is used to join a table to itself as if the table were two tables.

- They are also useful for comparisons within a table also when modeling hierarchies.

- Example:

```
Select a.*, b.*
From table_ab a, table_ab b
Where a.id = b.id;
```

Query:

Select a.product_id, b.product_name From products a, products b where a.aisle_id = b.aisle_id;

product_id	product_name	aisle_id	department_id
1	Chocolate Sandwich	1	19
2	Salt	2	13
3	Tea	3	7
4	Sauce	2	1
5	Chicken	5	9

Products

Query:

Select a.product_id, b.product_name From products a, products b where a.aisle_id = b.aisle_id;

product_id	product_name
2	Salt
4	Sauce

Self Join Result Table

Exists - Operator

- Used to test for the existence of any record in a subquery.

- Will return true if the subquery returns one or more records.

- Example:

```
Select a.*
From a
Where Exists (Select id from b where id ⇔ condition);
```

Query:

*Select * From products Where Exists (Select aisle_id from aisles where aisle_id > 2);*

product_id	product_name	aisle_id	department_id
1	Chocolate Sandwich	1	19
2	Salt	2	13
3	Tea	3	7
4	Sauce	2	1

Products

aisle_id	aisle
1	Dessert
2	Seasons
3	Drinks
4	Medicine

Aisles

Query:

*Select * From products Where Exists (Select aisle_id from aisles where aisle_id > 2);*

product_id	product_name	aisle_id	department_id
3	Tea	3	7

Result Table after using EXISTS Operator

Any - Operator

- Is used with a Where or Having clause.
  - Having: used for aggregate functions.

- Returns true if any of the subquery values meet the condition.

- Example:

```
Select a.*
From a
Where id = Any (Select id from b where ⇔ condition);
```

Query:

*Select * From products Where aisle_id = Any (Select aisle_id from aisles where aisle_id > 1);*

product_id	product_name	aisle_id	department_id
1	Chocolate Sandwich	1	19
2	Salt	2	13
3	Tea	3	7

Products

aisle_id	aisle
1	Dessert
2	Seasons
3	Drinks
4	Medicine

Aisles

Query:

*Select * From products Where aisle_id = Any (Select aisle_id from aisles where aisle_id > 1);*

product_id	product_name	aisle_id	department_id
2	Salt	2	13
3	Tea	3	7

Result Table after using ANY Operator

All - Operator

- Also used with a Where or Having clause.

- Returns true if all of the subquery values meet the condition.

- Example:

```
Select a.*
From a
Where id = ALL (Select id from b where value ⇔ condition);
```

Query:

*Select * From products Where aisle_id = ALL (Select aisle_id from aisles where aisle_id = 2);*

product_id	product_name	aisle_id	department_id
1	Chocolate Sandwich	1	19
2	Salt	2	13
3	Tea	3	7
4	Sauce	2	1

Products

aisle_id	aisle
1	Dessert
2	Seasons
3	Drinks
4	Medicine

Aisles

Query:

*Select * From products Where aisle_id = ALL (Select aisle_id from aisles where aisle_id = 2);*

product_id	product_name	aisle_id	department_id
2	Salt	2	13
4	Sauce	2	1

Result Table after using ALL Operator

Thank You