Using Joins for Everyday Life Problems

Nate Sampo and Camille Xue

Types of Joins
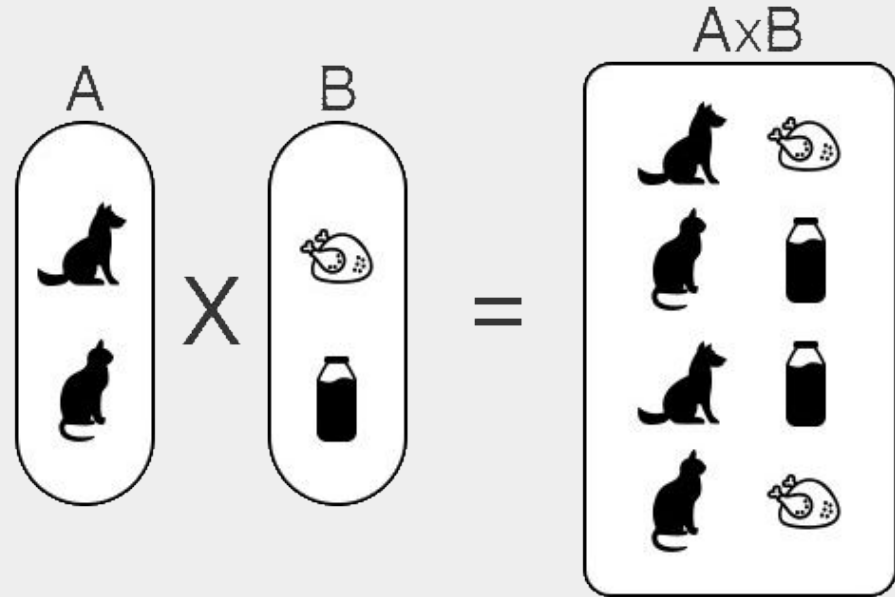
- **Cross**
- **Inner**
- **Self Join**
- **Left outer**
- **Right outer**
- **Full outer**

Inner Joins: Game Server

1. You want to setup a fun game

2. Make sure the right actions happen at the right time

3. Keep track of players

4. Mitigate latency by matching up players based on their region

Cross Join

- Cartesian product
- Creates every permutation
- Useful for setting up other joins

Game Server: Server Routine

Minutes Elapsed

Minute
0
1
2
3
4
5
...

Server Routine Per Minute

Seconds	Action
1	Spawn Enemy
10	Move Obstacle
30	Give 100 Gold
50	Spawn Boss

Server Routine for Entire Game

Minute	Seconds	Action
0	1	Spawn Enemy
0	10	Move Obstacle
0	30	Give 100 Gold
0	50	Spawn Boss
1	1	Spawn Enemy
1	10	Move Obstacle
1	30	Give 100 Gold
1	50	Spawn Boss
2	1	Spawn Enemy
2	10	Move Obstacle
2	30	Give 100 Gold
2	50	Spawn Boss
...	...	...

SELECT *
FROM minutes **CROSS JOIN** actions

Inner Join

Table A   Table B
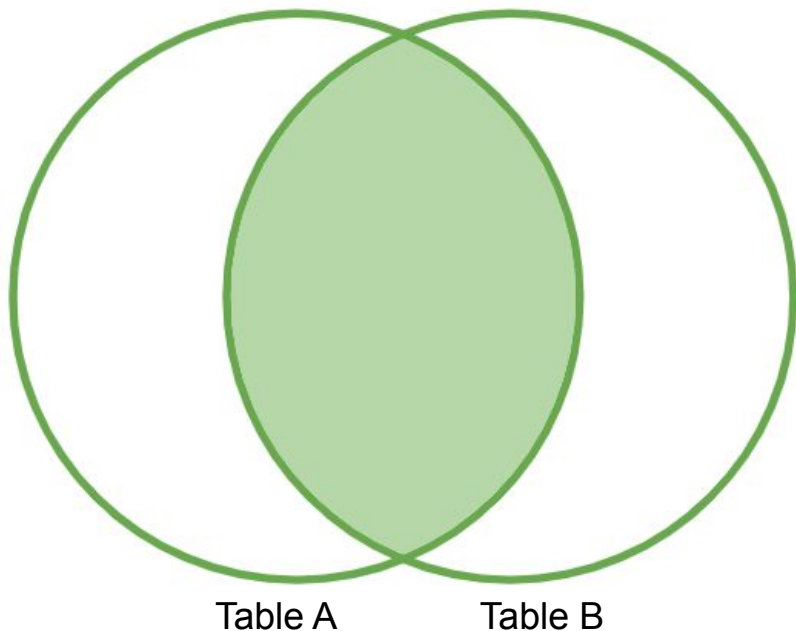
- All rows in both tables where the values meet the join condition
- Equi joins are a subset of Inner joins
  - Natural joins are a subset of equi joins

Cross join, but only keep the rows where the join condition is met

```
SELECT teams.teamID, teams.playerID, players.username
FROM teams CROSS JOIN players
WHERE teams.playerID = players.playerID
```

Game Server: Keeping Track of Teams

Players

PlayerID	Username
1	N4tticus
2	H4CKERD00D
3	camilleon
4	SoxFan98
5	xXGamerXx
6	Derpezoid
7	vinny
8	tigereye

Teams

TeamID	PlayerID
1	6
1	2
2	3
2	1
3	5
3	8

Teams with usernames

TeamID	PlayerID	Username
1	6	Derpezoid
1	2	H4CKERD00D
2	3	camilleon
2	1	N4tticus
3	5	xXGamerXx
3	8	tigereye

SELECT teams.teamID, teams.playerID, players.username
FROM teams **INNER JOIN** players **ON**
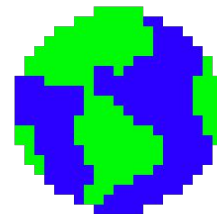teams.playerID = players.playerID

Self Join
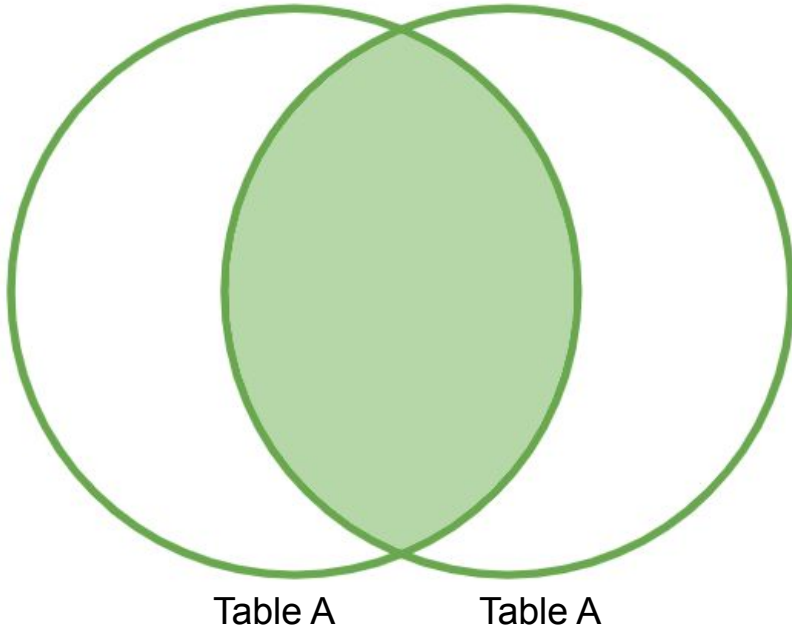
Table A   Table A

- Similar to inner join, where you are joining two tables based on a matching column
- … But the two tables are the same table

Done by making two copies of the table and performing inner join on them

Game Server: Matchmaking

Matchmaking Queue

PlayerID	Name	Region
1	N4tticus	North America
2	H4CKERD00D	Europe
3	camilleon	North America
4	SoxFan98	North America
5	xXGamerXx	Europe
6	Derpezoid	Asia
7	vinny	Asia
8	tigereye	Europe

Potential Teammates

| PlayerID	Name	PlayerID	Name	Region
1	N4tticus	3	camilleon	North America
1	N4tticus	4	SoxFan98	North America
3	camilleon	4	SoxFan98	North America
2	H4CKERD00D	5	xXGamerXx	Europe
2	H4CKERD00D	8	tigereye	Europe
5	xXGamerXx	8	tigereye	Europe
6	Derpezoid	7	vinny	Asia

SELECT first.playerID, first.name, second.playerID, second.name, second.region
FROM players first **INNER JOIN** players second **ON** first.region = second.region
WHERE first.playerID < second.playerID

Outer Joins: Party Planning

1. You want to plan the best party

2. Make sure all your friends are happy: DON'T INVITE THEIR ENEMIES

3. Make sure the food is amazing: CHECK FOR FOOD ALLERGIES

Left Outer Join
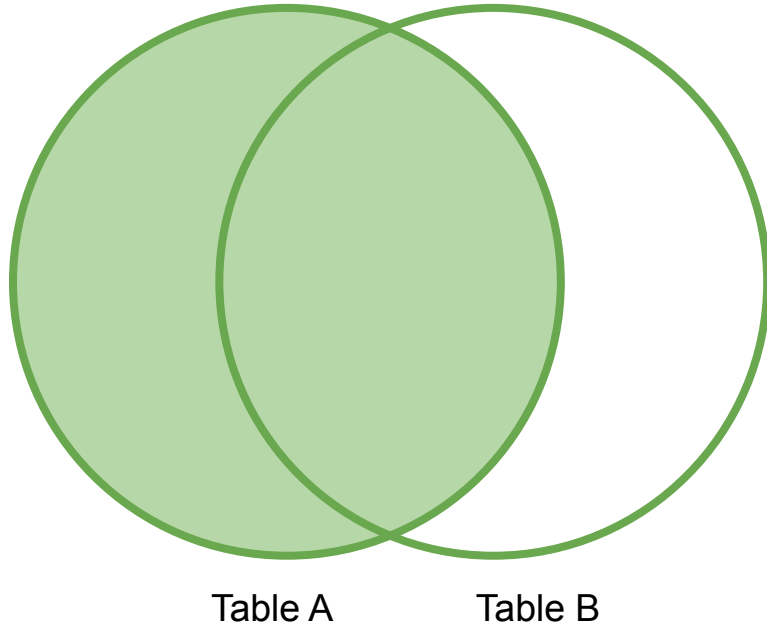
- All of the rows in Table A
- If there are no matches for ON clause, the row column value will be NULL

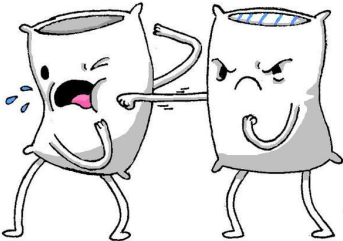Inner join + All values in Table A that don't have matches in B

Table A   Table B

Planning a Poppin' Party: WHO TO INVITE

Friends to Invite

Name	Favorite Food
Zuko	Burritos
Ariel	French Fries
Aang	Kale

Potential Party Problems

Name	Allergies	Enemies
Zuko	Dairy	Aang
Elsa	Pepper	Hans
Ariel	Shellfish	Ursula

```
SELECT *
FROM friends
LEFT OUTER JOIN problems ON
friends.Name = problems.Name;
```

Poppin' Party Invite Guide

Name	Favorite Food	Name	Allergies	Enemies
Zuko	Burritos	Zuko	Dairy	Aang
Ariel	French Fries	Ariel	Shellfish	Ursula
Aang	Kale	NULL	NULL	NULL

Right Outer Join

Table A   Table B
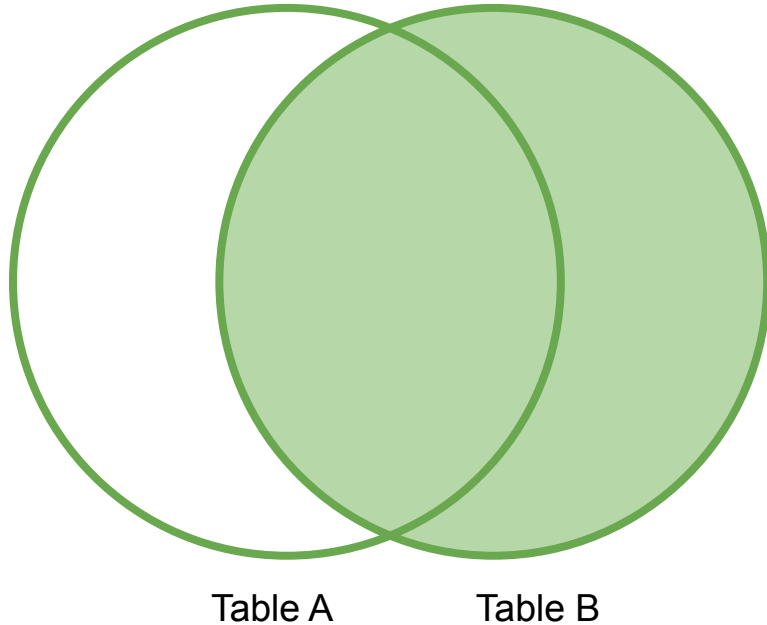
- All of the rows in Table B
- If there are no matches for ON clause, the row column value will be NULL

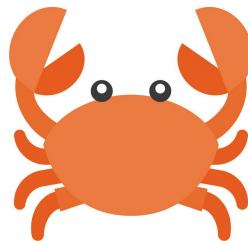Inner join + All values in Table B that don't have matches in A

Planning a Poppin' Party: FOOD

Friends to Invite

Name	Favorite Food
Zuko	Burritos
Ariel	French Fries
Aang	Kale

Potential Party Problems

Name	Allergies	Enemies
Zuko	Dairy	Aang
Elsa	Pepper	Hans
Ariel	Shellfish	Ursula

Party Food Guide

Name	Favorite Food	Name	Allergies	Enemies
Zuko	Burritos	Zuko	Dairy	Aang
NULL	NULL	Elsa	Pepper	Hans
Ariel	French Fries	Ariel	Shellfish	Ursula

```
SELECT *
FROM friends
RIGHT OUTER JOIN problems ON
friends.Name = problems.Name;
```

Full Outer Join
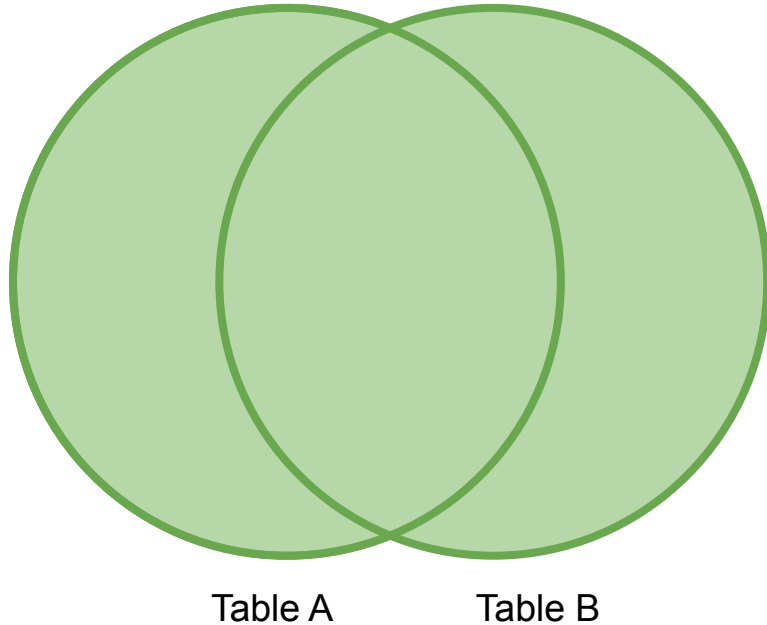
- All of the rows in Table A & B
- If there are no matches for ON clause, the row column value will be NULL

Inner join + All values in Table B and Table A that don't have matches

Table A   Table B

Planning a Poppin' Party: ULTIMATE GUIDE

Friends to Invite

Name	Favorite Food
Zuko	Burritos
Ariel	French Fries
Aang	Kale

Potential Party Problems

Name	Allergies	Enemies
Zuko	Dairy	Aang
Elsa	Pepper	Hans
Ariel	Shellfish	Ursula

Party Food Guide

```
SELECT *
FROM friends
RIGHT OUTER JOIN problems ON
friends.Name = problems.Name;
```

Name	Favorite Food	Name	Allergies	Enemies
Zuko	Burritos	Zuko	Dairy	Aang
Ariel	French Fries	Ariel	Shellfish	Ursula
Aang	Kale	NULL	NULL	NULL
NULL	NULL	Elsa	Pepper	Hans

Thank you for listening :)