SQL Joins

Max Masnick

August 7, 2015

What are joins?

Combine two (or more) tables into a single results table

I will go through the most common joins here.

Setup

- We will use two tables for these examples: `friends` and `pets`
- `friends` is a list of your friends
- `pets` is a list of pets, with a foreign key linking them to your `friends`
- One of your friends (Sam) has no pets
- One of the pets (Scales the iguana) is lost, so he's not linked to any friend

friends

friend_id	friend_name
1	John
2	Sarah
3	Rachel
4	Sam

pets

pet_id	owner_id	pet_type	pet_name
1	1	goldfish	Fishy
2	1	goldfish	Nemo
3	1	dog	Fido
4	2	cat	Samwise
5	2	bird	Feathers
6	3	chinchilla	Fuzzy
7	NULL	iguana	Scales

The SQL to set this up is available at
http://masnick.org/projects/sql-joins/

Inner joins

```
SELECT * FROM friends
INNER JOIN pets
ON friends.friend_id = pets.owner_id;
```

- Rows from `friends` that **match up** with at least one row from `pets`.
  - "Match up" defined by `ON friends.friend_id = pets.owner_id`

Inner joins

```
SELECT * FROM friends
INNER JOIN pets
ON friends.friend_id = pets.owner_id;
```

- Results table has columns from **both** `friends` and `pets`

Inner joins

```
SELECT * FROM friends
INNER JOIN pets
ON friends.friend_id = pets.owner_id;
```

- If a friend has multiple pets, there will be multiple rows in the results—one for each of their pets.
- If a friend doesn't have any pets, they won't be included in the results.

Inner joins

```
| friend_id | friend_name | pet_id | owner_id | pet_type   | pet_name |
|-----------|-------------|--------|----------|------------|----------|
| 1         | John        | 1      | 1        | goldfish   | Fishy    |
| 1         | John        | 2      | 1        | goldfish   | Nemo     |
| 1         | John        | 3      | 1        | dog        | Fido     |
| 2         | Sarah       | 4      | 2        | cat        | Kitty    |
| 2         | Sarah       | 5      | 2        | bird       | Feathers |
| 3         | Rachel      | 6      | 3        | chinchilla | Fuzzy    |
```

- Multiple rows for friends with multiple pets
- No Sam (friend)
- No Scales the iguana (pet)

Inner joins – implicit syntax

An alternate way to get the same thing:

```
SELECT * FROM friends, pets
WHERE friends.friend_id = pets.owner_id;
```

Outer joins

- **Left outer join**: all rows from `friends`, all *matching* rows from `pets`
- **Right outer join**: all rows from `pets`, all *matching* rows from `friends`
- **Full outer join**: combines all rows from `friends` and `pets`, regardless of whether they match

Left outer join

```
SELECT * FROM friends
LEFT OUTER JOIN pets
ON friends.friend_id = pets.owner_id
```

- All `friends`, matching `pets` only
- No matching pet → NULL cells
- Pets with no owner are not included

Left outer join

```
| friend_id | friend_name  | pet_id  | owner_id  | pet_type    | pet_name  |
|-----------|--------------|---------|-----------|-------------|-----------|
| 1         | John         | 1       | 1         | goldfish    | Fishy     |
| 1         | John         | 2       | 1         | goldfish    | Nemo      |
| 1         | John         | 3       | 1         | dog         | Fido      |
| 2         | Sarah        | 4       | 2         | cat         | Kitty     |
| 2         | Sarah        | 5       | 2         | bird        | Feathers  |
| 3         | Rachel       | 6       | 3         | chinchilla  | Fuzzy     |
| 4         | Sam          | (null)  | (null)    | (null)      | (null)    |
```

Right outer join

```
SELECT * FROM friends
RIGHT OUTER JOIN pets
ON friends.friend_id = pets.owner_id
```

- Almost the same as left outer join.
- Only differences:
  - `pets` with no owners in results
  - `friends` with no pets are left out

Right outer join

```
| friend_id | friend_name | pet_id | owner_id | pet_type   | pet_name |
|-----------|-------------|--------|----------|------------|----------|
| 1         | John        | 1      | 1        | goldfish   | Fishy    |
| 1         | John        | 2      | 1        | goldfish   | Nemo     |
| 1         | John        | 3      | 1        | dog        | Fido     |
| 2         | Sarah       | 4      | 2        | cat        | Kitty    |
| 2         | Sarah       | 5      | 2        | bird       | Feathers |
| 3         | Rachel      | 6      | 3        | chinchilla | Fuzzy    |
| (null)    | (null)      | 7      | (null)   | iguana     | Scales   |
```

Full outer join

```
SELECT * FROM friends
FULL OUTER JOIN pets
ON friends.friend_id = pets.owner_id;
```

- **Does not exist in MySQL**, but does in PostgreSQL and other databases
- Like left/right outer join, but includes unmatched rows from both tables

Full outer join

friend_id	friend_name	pet_id	owner_id	pet_type	pet_name
1	John	1	1	goldfish	Fishy
1	John	2	1	goldfish	Nemo
1	John	3	1	dog	Fido
2	Sarah	4	2	cat	Kitty
2	Sarah	5	2	bird	Feathers
3	Rachel	6	3	chinchilla	Fuzzy
4	Sam	(null)	(null)	(null)	(null)
(null)	(null)	7	(null)	iguana	Scales

Cross join

```
SELECT * FROM friends
CROSS JOIN pets;
```

- Different from other joins
- No matching between tables
- Takes every row from `friends`, combines it with every row from `pets`
- Click here to see the results table

Links to other resources

- SQL joins article on Wikipedia
- Venn diagrams of SQL joins
- SQL Fiddle