Join

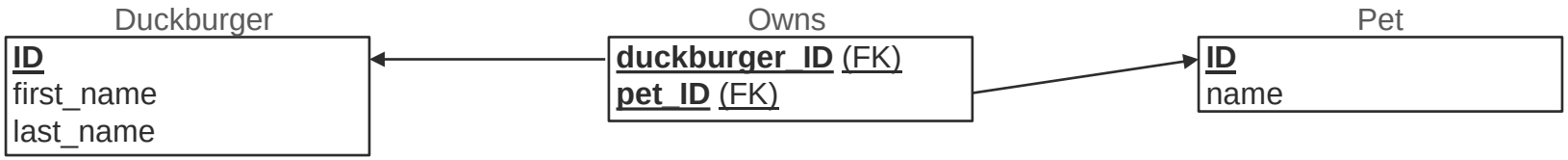
Inner Join, Left Join, Right Join

Inner Join

- This technique is very similary to using where statements in specifying the conditions for joined queries.
  - We will use the same example as the one we used for the where statement.
- Left join and right join work slightly differently.

Inner Join

- We need to fetch the first name and last name for each of Bolivar's owners.
- Let's first take another look at our relational model.



- The owns table contains two foreign keys referencing the duckburger and pet tables respectively.

Inner Join

- We notice that the required columns, the owners' first and last names, are stored in the duckburger table.
- Furthermore, we notice that the first information, Bolivar's name is stored in the pets table.
- The ownership information is in the owns table.
- Conclusion: We need all three tables to complete the query.

Inner Join

- Now, completing the query is purely mechanical.
- Remember that the foreign keys always point to the primary key of another table.
- With this information, we can write the join conditions to join our tables:
- The foreign key duckburger_ID of the owns table connects to the primary key id of the duckburger table.
- The foreign key pet_id connects to the primary key id of the pets table.
- The last thing we need is an additional condition specifying that we are only interested in the owners of Bolivar.

Inner Join

```
MariaDB [duckburg]> select first_name, last_name
    -> from duckburger
    -> inner join owns on owns.duckburger_id = duckburger.id
    -> inner join pet on owns.pet_id = pet_id
    -> where pet.name = "Bolivar";
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| Donald     | Duck      |
| Huey       | Duck      |
+------------+-----------+
```

New skills:

- Syntax for inner_join and on
- The join condition is the same as when using where

Note:

- The order in which you join the tables does not matter.
- The primary and foreign keys can be on either side of the equals symbol.
- If the same column name is used in separate tables, we must specify the table we want to refer to (here for id).

Left Join / Righ Join

- Both of these operations fetch all values from the other table even if they have not been included in the join.
  - Compare to the where conditions and inner join that only include the joined values.
- Left join includes all values from the left table.
- Right join includes all values from the right table.
  - Easy to notice if the tables are mirrored.

Left Join / Righ Join

- In the previous examples we have queried for the owners of Bolivar.
- Before we test left join and right join, we will first create an inner join query that lists all duckburgers:
  - Their first name
  - Their last name
  - The name of their pet

Left Join / Righ Join

```
MariaDB [duckburg]> select first_name, last_name
    -> from duckburger
    -> inner join owns on owns.duckburger_id = duckburger.id
    -> inner join pet on owns.pet_id = pet_id;
+------------+-----------+-----------+
| first_name | last_name | name      |
+------------+-----------+-----------+
| Donald     | Duck      | Bolivar   |
| Huey       | Duck      | Bolivar   |
| Mickey     | Mouse     | Pluto     |
| Magica     | De Spell  | Ratface   |
+------------+-----------+-----------+
```

Scrooge McDuck is missing from the results as he does not own a pet. This is because inner join and where only include the values that are part of the join condition, not records where one more more of the required values are missing.

Left Join / Righ Join

- Now we will create a query that lists all duckburgers by their first name and last name, regardless of whether the person has a pet. If they do, the pet name is listed as well. Otherwise the pet name field shows "NULL".

Left Join / Righ Join

```
MariaDB [duckburg]> select first_name, last_name, name
    -> from duckburger
    -> left join owns on owns.duckburger_id = duckburger.id
    -> left join pet on owns.pet_id = pet_id;
+------------+-----------+-----------+
| first_name | last_name | name      |
+------------+-----------+-----------+
| Donald     | Duck      | Bolivar   |
| Scrooge    | McDuck    | NULL      |
| Huey       | Duck      | Bolivar   |
| Mickey     | Mouse     | Pluto     |
| Magica     | De Spell  | Ratface   |
+------------+-----------+-----------+
```

- Now also Scrooge McDuck is listed.
- The duckburger table is marked on the left side of the left join statement.

Left Join / Righ Join

```
MariaDB [duckburg]> select first_name, last_name, name
    -> from duckburger
    -> right join owns on owns.duckburger_id = duckburger.id
    -> left join pet on owns.pet_id = pet_id;
+------------+-----------+-----------+
| first_name | last_name | name      |
+------------+-----------+-----------+
| Donald     | Duck      | Bolivar   |
| Scrooge    | McDuck    | NULL      |
| Huey       | Duck      | Bolivar   |
| Mickey     | Mouse     | Pluto     |
| Magica     | De Spell  | Ratface   |
+------------+-----------+-----------+
```

Here the same operation was performed mirrored using right join where the duckburger table is marked on the rigth side of the statement.