Joins (cont.)

Who likes pizza *and* beer?

Try a three way join (persons x favourite_foods x favourite_foods) with the ON condition ensuring:

- all three person_ids match
- favourite_foods.food (first copy) should equal ‘Pizza’
- favourite_foods.food (second copy) should equal ‘Beer’

Example:

```
SELECT first_name, last_name, f1.food, f2.food
FROM

persons AS p
JOIN favourite_foods AS f1
JOIN favourite_foods AS f2
ON

p.person_id = f1.person_id
AND f1.person_id = f2.person_id
AND f1.food = ‘Pizza’
AND f2.food = ‘Beer’;
```

Note no quotes with AS when used like this, because we’re using it in a different way.

Types of JOIN

We have been using the INNER JOIN (implicitly, by leaving out inner). There are a few other types of JOIN, like OUTER, LEFT, RIGHT, and we may see these later.

Movies example

Three tables:

- movies
- actors (info about each actor in the database)

- castings (actor-movie combinations)

List the max score obtained by any film released during the 60s:

```
SELECT MAX(score)

FROM movies

WHERE yr BETWEEN 1960 AND 1969;
```

List for each year the number of films released that year and the max, min, and average score obtained:

```
SELECT yr, COUNT(*), MIN(score), AVG(score), MAX(score)

FROM movies

GROUP BY yr;
```

List the ids of all films starring Cary Grant:

We will need to use a join here, since we need both the actors and castings tables.

```
SELECT movieid
FROM

actors JOIN castings
ON id = actorid

WHERE name = ‘Cary Grant’;
```