03a_joins

Queries with more than 1 table

Contacts DB

```
SET search_path TO contacts;
```

1. *Cartesian product*: match every of a table with every row of the other table
   - gives too many rows
   - rarely used because it doesn't usually give useful information
   - and it can become very inefficient if the tables involved are large
     - the number of rows in the results will be the multiplication of the number of rows in each table involved

```
SELECT *
FROM call,
     contact;
```

2. *Join*: keep only the matching rows, by "following" the foreign key from `call` to `contact`
   - this is the old way to do a join (a cartesian product followed by a `WHERE` condition)

```
SELECT *
FROM call,
     contact
WHERE call.contact_id = contact.contact_id;
```

Joins

Inner Join

3. A more modern way to express a join in SQL
   - there are many types of joins, the most common kind is an `INNER JOIN`

```
SELECT *
FROM call
         INNER JOIN contact
```

```
                    ON call.contact_id = contact.contact_id;
```

Natural join

4. Join on columns with the same names, using the = operator, and by removing duplicate columns
   - note that the columns are in a different order
   - it's not recommended (even discouraged) to use natural joins because the join condition is not specified and might have consequences difficult to predict, especially on the long term if the database is modified after the queries have been written

```
SELECT *
FROM call
         NATURAL JOIN contact;
```

- The phone call with ID 2 doesn't match any contact, therefore it is not listed in the results
- Whenever we use = with `NULL`, it's always false, hence call 3 is not in the results

Outer Joins

5. Use a *left outer join* to keep the phone calls not matching any contact
   - it is like an inner join, but rows on the left not matching anything on the right will be kept
   - use a *right outer join* to keep the contacts not matching any phone call
   - or a *full outer join* to keep rows not matching on both sides

```
SELECT *
FROM call
         LEFT OUTER JOIN contact
                         ON call.contact_id = contact.contact_id;
```

```
SELECT *
FROM call
         RIGHT OUTER JOIN contact
                         ON call.contact_id = contact.contact_id;
```

```
SELECT *
FROM call FULL OUTER JOIN contact
    ON call.contact_id = contact.contact_id;
```