02a_intro_sql

Introduction to SQL

Relational Model

- **DBMS**: Database Management System
- **Database** (DB): set of *relations*
  - *relation* = *table*
- **Relation**: set of *tuples*
  - *tuple* = *row* = *record*
- **Tuple**: set of attributes
  - *attribute* = *column*
- All tuples in a given relation have a fix set of attributes
- Most common terms: *table*, *row*, and *column*

Contacts DB

- Think of a contacts DB on a phone
- 2 tables:
  - `contact(CONTACT_ID, name, phone, address, email)`
  - `call(CALL_ID, phone, date, time, contact_id*)`

Creating the contacts DB

1. Run the `contact.sql` file to create the DB
   - the SQL commands included in that file will be explained later
   - open the file in DataGrip, and run the file in an existing session, or create a new session
2. To check the results, you can use the *Database Explorer* tab, or run the following queries:

```
SELECT *
FROM contacts.contact;
```

```
SELECT *
FROM contacts.call;
```

3. To avoid prefixing the table names with the schema name all the time, we can run the following command:

```
SET search_path TO contacts;

SELECT *
FROM contact;

SELECT *
FROM call;
```

Single Table Queries

1. Find John Doe's email address

```
SELECT email
FROM contact
WHERE name = 'John Doe';
```

2. Find the contacts without a phone number

```
SELECT *
FROM contact
WHERE phone IS NULL;
```

3. Find the contacts with a phone number

```
SELECT *
FROM contact
WHERE phone IS NOT NULL;
```

With Aggregate functions

4. Count the number of rows in the `call` table

```
SELECT COUNT(*)
FROM call;
```

5. Count the number of rows in the `call` table with non-null values for `contact_id`

```
SELECT COUNT(contact_id)
FROM call;
```

6. Count the number of rows in the `call` table with null values for `contact_id`

```
SELECT COUNT(*)
FROM call
WHERE contact_id IS NULL;
```