Session 4

Last time

We saw:

- OLAP vs OLTP databases
- ERD and how to generate it in pgAdmin, also generating the SQL that creates the schema
- insertion, update and deletion anomalies
- the problem with `NULL` values and why it's best to avoid them
- normalization
- normal forms: 1NF, 2NF, 3NF

We started working on the normalization of the trees table

Taking the column `domain` as example, let's first make sure that the column is a good candidate for normalization

`domain` is a categorical column, with null values and many uniques. Inserts, updates and deletion anomalies.

The process has the following steps:

- create a new `domains` table (id, domain)
- import UNIQUE and NOT NULL domain values from trees to the table `domains`. Best if values are sorted alphabetically.
- add a `domain_id` key in the trees table
- reconcile the foreign key with the primary key: `trees.domain_id` with the `domain.id`
- add a constraint on that key so that it is a foreign key in the trees table referencing the domains table
- check that all trees records have a valid `domain_id`
- drop the `domain` column in trees

Today

First a PC and Mac

- where is PostgreSQL installed?
- how to add PostgreSQL to the PATH
- starting, stopping and monitoring PostgreSQL with `brew` or `pg_ctl`
- location of `.psqlrc`, `pg_hba`, `postgresql.conf`
- create `.pgpass` to avoid having to type your password each time

Practice

All practices are graded.

https://forms.gle/GBGffeAT1sLnYeTk9

- connect to your local server with `psql` or in `pgAdmin`
- new dataset : Spotify songs : 366 records. csv file. no need to USE pgAdmin restore or pg_restore.
- Your task is to normalize the artist column following the process above

- If you're done before the break (11h30), read:
    - how to setup `.psqlrc` : https://www.crunchydata.com/postgres-tips
    - or choose something to read in the documentation: https://www.postgresql.org/docs/current/sql.html

Window functions and CTEs

We'll use the WorldHits dataset to learn about:

- window functions : ROW_NUMBER(), RANK(), ... , OVER(Partition by ...)
- Common Table Expressions : WITH ... AS (sql) SELECT ....

EOD

- everybody is on board : PostgreSQL running, restoring SQL dump files , `psql` to connect
- you can normalize a column from a table
- you can write queries using window function and CTEs