Databases for Data Science

Lecture 03 · 2022-09-08

Course Logistics

Assignment 1 is now available on Canvas.

- Due date: end of day 9/16 (next Friday)

Topics for the next few lectures have been adjusted.

Today

- Query exercises
- Common table expressions
- Investigating performance

Warm-up

- Connect VSCode to CS1.
- Create your `assignment_1` folder.

Exercise: `WHERE`
[using `booking_dates`]

- Show all the arrests before 2000.
- Show all the arrests between 2000 and 2005.

Exercise: `SELECT DISTINCT` [using `booking_dates`]

- What release codes are listed in the database?
- List the people released for `'TIME SERVED'`.

Matching multiple values

`WHERE ... IN (...)`: match any of a list of values.

```
select * from person
where
    name='Alex' or
    name='Blake' or
    name='Charlie';
```

```
select * from person
where name in ('Alex', 'Blake', 'Charlie');
```

Exercise

[using `booking_dates`]

- List the people released for `'STATE HOSPITAL' or 'DECEASED'`.

Exercise: Answering questions with data

[using `charges`]

Which charges are related to marijuana use?

- How could we find this information in the DB?
- What could we learn *without* looking at the DB, and how?

Exercise: Counting
[using `charges`]

- How many courts are listed?

(...are we sure about that?)

Exercise: Counting

[using `charges`]

- How many times has each court tried a marijuana charge?

Problem: we keep having to clean the column.

How can we avoid this sort of thing?

Common Table Expressions

Using `WITH tablename AS (...)`, define a temporary table.

```sql
-- Subquery
SELECT
    employment.job_title,
    employment.salary,
    averages.salary as average_salary
FROM (
    select
        job_title,
        AVG(salary) as salary
    from employment
    group by job_title
) averages JOIN employment
ON averages.job_title = employment.job_title;
```

```sql
-- Common table expression
WITH averages AS (
    select
        job_title,
        AVG(salary) as salary
    from employment
    group by job_title
)
SELECT
    employment.job_title,
    employment.salary,
    averages.salary as average_salary
FROM averages JOIN employment
ON averages.job_title = employment.job_title;
```

Common Table Expressions

Why is this useful?

Exercise: Counting and CTEs
[using `charges`]

- (again) How many times has each court tried a marijuana charge?

Exercise: Counting and CTEs

[using `charges`]

- Which courts haven't tried a marijuana charge?

`COUNT` will ignore empty groups.

Can you think of two approaches to find the courts with zero counted charges?

Exercise: Counting and CTEs
[using `charges`]

- How many courts have tried a marijuana charge, and how many have not?

Asking useful questions

Questions about the dataset

- Why are there more charges than bookings?
- Why are there fewer booking_dates than bookings?
- What do the column names mean?

Questions of methodology

- What signifies homelessness? Lack of address?
- Can homeless status be given by the data?

Questions answerable in the dataset

- Who has been arrested over ten times?
- How many arrests are there by race, gender, or ethnicity?

Can you think of more questions?

1. About the dataset
2. About methodology
3. Within the dataset

Investigating performance

`explain` prints the *query plan*: how Postgres will execute a query

`explain analyze` runs the query and shows execution times

More datasets

- `imdb`
- `yelp`
- `sentiment140`

Next time

- Create, update, delete
- Database design
- Normalization
- SQL and Python