Discussion 11

SQL 💀

Antonio Kam

anto [at] berkeley [dot] edu

All slides can be found on

`teaching.rouxl.es`

Announcements

- Scheme Project
  - Checkpoint 2 due on Friday
  - Project due Tuesday
  - Good luck 🥲

Notes from last section

- What's your favorite tv show/movie?
  - HTTYD
- is it pronounced squeel or sequel?
  - sequel
- have you tried ascension? It's also a deckbuilding game similar to dominion.
  - have never tried it, will look around because dominion is fun
- do you use reddit? i found the reddit question hilarious bc i am an avid redditor unfortunately
  - i used to, but stopped after the protests
- favorite pokemon?
  - mega altaria

Temperature Check 🌡️

- SQL

SQL

Select Statements and Queries

Select Statements

- You can experiment with all of this on sql.cs61a.org
- If we have a pre-existing table (for example, the `records` table), we can grab values from that table using a `FROM` clause
- Using `*` will select all columns from a table

```
SELECT [columns] FROM [tables] WHERE [condition] ORDER BY [criteria] LIMIT [number];
```

Demo:

```
SELECT * FROM records WHERE title = "Programmer";
SELECT name, salary FROM records WHERE division = "Accounting"
  ORDER BY salary DESC LIMIT 5;
```

Worksheet!

Joins

Joins

- Sometimes, people might store data in multiple tables
- It's hard to access data from both these tables
- That's where *joins* come in!

```
SELECT * FROM records, meetings; -- can select multiple tables
```

Ambiguous Joins

- Tables might have overlapping column names
- We need a way to distinguish between these columns
  - Especially if you need to join a table with itself (useful if you want to compare 2 people with each other)
- Use the `as` keyword

Ambiguous Joins

```
SELECT a.name, a.title FROM records AS a, records AS b
  WHERE a.name = "Louis Reasoner" AND a.supervisor = b.name;
```

Aggregation 

Aggregation

- Aggregation tends to be useful when you have multiple groups, and you want to group by certain pieces of data.
- You can also combine multiple rows into 1 with aggregation
  - `SELECT COUNT(*) FROM RECORDS;`
  - `SELECT name, MAX(salary) FROM RECORDS;`
- `GROUP BY` will allow you to perform these aggregation functions on specific groups
  - `SELECT division, MIN(salary) FROM records GROUP BY division;`
- `WHERE` statements for `GROUP BY`s uses the `HAVING` clause
  - `HAVING` filters out entire groups
  - You can have both `WHERE` and `HAVING` in the same statement

Results from last section (`links.rouxl.es/disc`)

- Discussion: 44%
- Lab: 56%

Mental Health Resources

- CAPS:
  - If you need to talk to a professional, please call CAPS at 510-642-9494.
- After Hours Assistance
  - For any assistance after hours, details on what to do can be found at this link

Anonymous Feedback Form

`links.rouxl.es/feedback`

Thanks for coming! 🥳

Please give me feedback on what to improve!