Lab #2: More Relational Data Processing

Everything Data
CompSci 216 Spring 2015
Announcements (Mon. Jan. 26)

• Team assignment slightly updated because of drop/add
  – If you didn’t receive an email from me, your team assignment remains unchanged

• You should been very comfortable with your VM now
  – If not, come to our office hours to resolve any setup issues
Seat assignment

See course website (under “Schedule”) for team assignment

Front of room

1
2
3
4
5

6
7
8
9
10

Back of room

11
12
13
14
15

16
17
18
19
20
Format of this lab

• Class discussion of Homework #2 + a few more SQL tricks (15 min.)
• Team discussion of Homework #2, Part 2 (10 min.)
  – Share your claims and counterarguments
• Lab #2, with team challenge! (40 min.)
  – Win prizes and extra credit!
• Summarize lessons learned (10 min.)
Homework #2, Part 2:

Price vs. Pelosi

- There is a Republican Representative whose last name is also Price
  - The filtering condition in the query shown in Lecture #2 wasn’t precise enough

- Once we make the filtering condition precise, the result becomes expected:
  - David Price vs. Pelosi: 86%
  - Butterfield vs. Pelosi: 81%
Homework #2, Part 1 (A-D)

• Find females born in/after 1980: filter
• Find NC senators: join
• Count House votes by record value: join, filter, group-by
• Which Dems voted against funding the government in 2014?
  – A 4-way join: remember the connections (join conditions)!
  – Use LIKE or ~ for string pattern matching
Homework #2, Part 1 (E)

Breakdown of votes by party
• Key: given vote_id and person_id, find the person’s party at the time of the vote
... (SELECT DISTINCT party
    FROM person_roles
    WHERE person_id = person_votes.person_id
    AND start_date <= votes.date AND votes.date <= end_date) ...

– Govtrack data isn’t clean: multiple roles may be active at the same time!

– DISTINCT option for SELECT eliminates duplicates
  • DISTINCT is also available for aggregation functions
    – E.g., COUNT(DISTINCT party) = # distinct parties in a group
Some tips for writing SQL

*Top-down*
- First write the “outer” query
- Then fill in the blanks with subqueries

*Bottom-up*
- For each output row, think about what evidence (rows from input tables) you need to prove it
  - That will give you the tables to join in *FROM*

*Think logically, on whole sets*
Queries too slow?

• **Indexing** may help
  – Index on a column speeds up looking up rows with a particular value (or within a range of values) for that column
  – E.g., `CREATE INDEX ON persons_roles(person_id);`

• **Note:** if a column is **PRIMARY KEY** or **UNIQUE**, the database system will automatically create an index on it

Image credit: http://2.bp.blogspot.com/-otmbPSD4G5k/T2C0dhJZUiI/AAAAAAAAAGQ/zbzKCD3fCCM/s1600/indexing.png
Team work (10 min.)

• Discuss Homework #2, Part 3 with your teammates
  – Share your claims and counterarguments
• At the end of this period, I will give the signal to start the challenge
Team claim-checking challenge

• 4 claims, some chosen from your submissions, some prepared by us
• For each claim
  – Is it technically “correct”?  
  – Is it misleading? Why?  
  – Show us your supporting queries for both

• Each claim successfully checked = extra credit worth 1% of a homework  
• The first team to finish checking all claims gets a prize
Challenge

• On my mark, go to Lab #2 page
• Take just a little time to go over Part 1—it will help you!
• Once your team finishes checking a claim, raise your hands and yell “CHECK ⟨claim #⟩!”
  – One of the course staff will come and verify your answer
• If you finish early, try Part 3 for fun
  – To impress us; no extra credit though
Lessons learned

• With big data, it’s easy to find something to support your agenda
• It’s easy to make mistakes
• Knowing SQL doesn’t mean you know how to query a dataset

…lies, d—ned lies, and statistics…
— Mark Twain
More lessons learned

• Behold the power of simple but flexible data model + “declarative” queries
  – Data = collection of tables
  – Queries = compositions of filter, group-by aggregation, join, sort, subqueries, …

• Physical data independence
When to use a SQL database

- Pros and cons?
- Despite its age, SQL is still your best bet for querying your structured data
Finally

• Remember to submit team.txt by midnight
• Sample solutions to Lab #2 will be posted by tomorrow morning
  – Take some time on your own to go through the sample solutions