Lecture 7:
- Quiz 3
- Intro to SQL

Announcements:
- HW 2 due Tues
- Look over the DB connection guide (course webpage)
- Note: I will be using mariadb-shell in class
**Relationships among operations**

Some examples of relational algebra equivalences (not comprehensive)

1. \(R \bowtie_C S = S \bowtie_C R\) ... commutative, also: \(\cup, \cap, \times\)

2. \(R \cap S = R \setminus (R \setminus S)\)

3. \(R \bowtie_C S = \sigma_C(R \times S)\)

4. \(\sigma_{C_1 \land C_2}(R) = \sigma_{C_2}(\sigma_{C_1}(R)) = \sigma_{C_1}(\sigma_{C_2}(R))\)

5. \(\sigma_{C_1 \lor C_2}(R) = \sigma_{C_1}(R) \cup \sigma_{C_2}(R)\)

6. \(\sigma_{C_1}(R) \bowtie_{C_2} S = \sigma_{C_1}(R \bowtie_{C_2} S)\) ... if \(C_1\) mentions only attributes of \(R\)

7. \(\pi_A(\sigma_C(R)) = \sigma_C(\pi_A(R))\) ... if \(C\) only mentions attributes in \(A\)

(*) the 6th equivalence is called "pushing a select"

There are additional operations/extensions we won’t cover

- semi-join (like natural join, but doesn’t remove duplicate columns)
- anti-join (non matches)
- divide (\(R \div S\) finds all \(R\) tuples, w.r.t. non \(S\) atts, that match all \(S\) tuples)
- grouping and aggregation
Intro to the Structured Query Language (SQL)

SQL standardization

- A series of standards: '86, '89, '92 (SQL2), '99 (SQL3), ... 2016
- DBMS products differ in how much they support
- Many implement extra features (extensions)

SQL is a “declarative” command-based language

- In general, this means that you say what you want to happen
- Not how to perform it (the DBMS figures it out)
- commands such as: CREATE, INSERT, SELECT, etc

SQL largely case insensitive

- Various conventions (lowercase, uppercase, camel case, etc.)
- Keywords usually in all uppercase, ids in all lowercase
- Some systems allow for case-sensitive names (at least partially)
General note on formatting and style conventions:

- use one ... i.e., use consistent formatting and naming
- there isn’t one true standard, and each standard is just a preference
- use the one that exists ... i.e., don’t create your own if unneeded

My general (current) preference: ... examples as we go

- use lowercase names with words separated by _’s (e.g., first_name)
- use meaningful names, avoid abbreviations
- use singular (not plural) table and attribute names
- indent 2-4 spaces (but use one consistently)
- add id to the end of simple primary keys when it makes sense
CS Database Server Basics

See the db-server info sheet on the course webpage ...

Option 1: Connect to ada, then connect to DB server (see diagram below)

Option 2: Connect directly to DB server (install mysql/mariadb shell)

To connect directly or to ada from off-campus, must use the GU VPN

• requires setting up VPN connection on your machine (see info sheet)

Some prefer to use MySQL workbench and/or VS Code extensions

• similar idea, just a GUI “veneer” ... better to learn command line first