CS143: Relational Algebra

Book Chapters

(5th) Chapters 2.2-3 (6th) Chapter 2.5-6 (7th) Chapter 2.5-6

Things to Learn

- Relational algebra
 - Select, Project, Join, \ldots

Database query language

• Data Manipulation Language (DML): A language to query and update relations

What is a query?

- Oxford English Dictionary: A question, especially one addressed to an official or organization
- Database jargon for question (complex word for simple concept)
- Questions to get answers from a database
 - Example: Get the students who are taking all CS classes but no Physics class
- Some queries are easy to pose, some are not
- Some queries are easy for DBMS to answer, some are not

Relational query languages

- Relational algebra (formal), SQL (practical)
- Relational Query:
 - A query is executed against input relations and produces an output relation Input relations \longrightarrow [query] \longrightarrow Output relation
 - Very useful: "Piping" is possible

Relational Algebra

 $\textit{Input relations (set)} \longrightarrow \boxed{\texttt{query}} \longrightarrow \textit{Output relation (set)}$

- Set semantics. no duplicate tuples. duplicates are eliminated
- In contrast, multiset semantics for SQL (performance reason)

Examples to Use

- School information
 - Student(sid, name, addr, age, GPA)

sid	name	addr	age	GPA
301	John	183 Westwood	19	2.1
303	Elaine	301 Wilshire	17	3.9
401	James	183 Westwood	17	3.5
208	Esther	421 Wilshire	20	3.1

- Class(dept, <u>cnum</u>, <u>sec</u>, unit, title, instructor)

dept	cnum	sec	unit	title	instructor
CS	112	01	03	Modeling	Dick Muntz
\mathbf{CS}	143	01	04	DB Systems	John Cho
\mathbf{EE}	143	01	03	Signal	Dick Muntz
ME	183	02	05	Mechanics	Susan Tracey

- Enroll(sid, dept, cnum, sec)

sid	dept	cnum	sec
301	CS	112	01
301	CS	143	01
303	EE	143	01
303	CS	112	01
401	CS	112	01

Simplest query: relation name

• Query 1: All students

SELECT operator

Select all tuples satisfying a condition

• Query 2: Students with age < 18

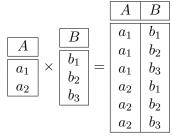
- Query 3: Students with GPA > 3.7 and age < 18
- Notation: $\sigma_C(R)$
 - Filters out rows in a relation
 - C: A boolean expression with attribute names, constants, comparisons (>, ≤, ≠, ...) and connectives (∧, ∨, ¬)
 - $-\ R$ can be either a relation or a result from another operator

PROJECT operator

- Query 4: sid and GPA of all students
- Query 5: All departments offering classes
 - Relational algebra removes duplicates (set semantics)
 - SQL does not (multiset or bag semantics)
- Notation: $\pi_A(R)$
 - Filters out columns in a relation
 - A: a set of attributes to keep
- Query 6: sid and GPA of all students with age < 18
 - We can "compose" multiple operators
- Q: Is it ever useful to compose two projection operators next to each other?
- Q: Is it ever useful to compose two selection operators next to each other?

CROSS PRODUCT (CARTESIAN PRODUCT) operator

• Example: $R \times S$



- Concatenation of tuples from both relations
- One result tuple for each pair of tuples in R and S
- If column names conflict, prefix with the table name
- Notation: $R_1 \times R_2$

 $-R_1 \times R_2 = \{t \mid t = \langle t_1, t_2 \rangle \text{ for } t_1 \in R_1 \text{ and } t_2 \in R_2 \}$

- Q: Looks odd to concatenate unrelated tuples. Why use $\times?$
- Query 7: Names of students who take CS courses

- Explanation: start with the query requiring sid, not name

• **Q:** Can we write it differently?

- Benefit of RDBMS. It figures out the best way to compute.

• Q: If |R| = r and |S| = s, what is $|R \times S|$?

NATURAL JOIN operator

- Example: Student \bowtie Enroll
 - Shorthand for $\sigma_{Student.sid=Enroll.sid}$ (Student × Enroll)
- Notation: $R_1 \bowtie R_2$

- Concatenate tuples horizontally
- Enforce equality on common attributes
- We may assume only one copy of the common attributes are kept
- Query 8: Names of students who take CS classes (Same as before)
- Query 9: Names of students taking classes offered by "Dick Muntz"
- Natural join: The most natural way to join two tables

RENAME operator

- Query 10: Find the pairs of student names who live in the same address.
- What about $\pi_{name,name}(\sigma_{addr=addr}(\text{Student} \times \text{Student}))?$
- Notation: $\rho_S(R)$ rename R to S
- Notation: $\rho_{S(A1',A2')}(R)$ for R(A1,A2) rename R(A1,A2) to S(A1',A2')
- Q: Is $\pi_{Student.name,S.name}(\sigma_{Student.addr=S.addr}(Student \times \rho_S(Student)))$ really correct?
 - How many times (John, James) returned?

UNION operator

- Query 11: Find all student and instructor names.
 - **Q:** Can we do it with cross product or join?
- Notation: $R \cup S$
 - Union of tuples from R and S
 - The schemas of R and S should be the same
 - No duplicate tuples in the result

DIFFERENCE operator

- Query 12: Find the courses (dept, cnum, sec) that no student is taking - How can we find the courses that at least one student is taking?
- Notation: R S
 - Schemas of R and S must match exactly
- Query 13: What if we want to get the titles of the courses?

- Very common. To match schemas, we lose information. We have to join back.

INTERSECT operator

- Query 14: Find the instructors who teach both CS and EE courses
 - **Q:** Can we answer this using only selection and projection?
- Notation: $R \cap S = R (R S)$
 - Draw Venn Diagram to verify

More questions

- Q: sids of students who did not take any CS courses?
 - **Q:** Is $\pi_{sid}(\sigma_{title \neq 'CS'}(Enroll))$ correct?

- **Q:** What is its complement?

• General advice: When a query is difficult to write, think in terms of its complement.

Relational algebra: things to remember

- Data manipulation language (query language)
 - Relation \rightarrow algebra \rightarrow relation
- Relational algebra: set semantics, SQL: bag semantics
- Operators: $\sigma, \times, \bowtie, \rho, \cup, -, \cap$
- General suggestion: If difficult to write, consider its complement