Review Session Quiz 1
Different Topics Covered at this Point

- Relational Algebra
- Relational Model
- Entity-Relationship Model
- Tuple Relational Calculus (Not on test)
- Sql 1
- Sql 2
- Constraints
Relational Algebra

Basic Operators
1. Select (σ)
2. Project (π)
3. Union (∪)
4. Set difference (−)
5. Cartesian product (X)
6. Rename (ρ)

Advanced Operators
1. Natural Join (⋈)
2. Division (÷)
3. Aggregation (g)
4. Outer Joins (⨝⨝⨝⨝)
Example Question

Schema:

- Product(Maker, model, type)
- PC(Model, speed, ram, harddrive, screen, price)
- Laptops(model, speed, ram, harddrive, screen, price)
- Printer(model, color, type, price)

3. Using no aggregate function, list the manufacture that makes the pc with the fastest processor?
Example Question

Schema:

- Product(Maker, model, type)
- PC(Model, speed, ram, harddrive, screen, price)
- Laptops(model, speed, ram, harddrive, screen, price)
- Printer(model, color, type, price)

3. Using no aggregate function, list the manufacture that makes the pc with the fastest processor?

$$\Pi_{\text{maker}}(\Pi_{\text{model}}(\text{PC}) - \Pi_{\text{PC.model}}(\sigma_{\text{PC.speed} < \text{PC2.speed}}(\text{PC} \times \rho_{\text{PC2}}(\text{PC})))) \bowtie \text{Product}$$
Example Question 2

Schema:

- Product(Maker, model, type)
- PC(Model, speed, ram, harddrive, screen, price)
- Laptops(model, speed, ram, harddrive, screen, price)
- Printer(model, color, type, price)

6. Which manufacture makes the most unique laptop models?
Questions on Relational Algebra?

Basic Operators
1. Select (σ)
2. Project (π)
3. Union (∪)
4. Set difference (–)
5. Cartesian product (X)
6. Rename (ρ)

Advanced Operators
1. Natural Join (⋈)
2. Division (÷)
3. Aggregation (g)
4. Outer Joins (⟕⟗⟖)
Entity Relationship Diagrams

Basic Terms and Concepts
1. Entities
2. Relationships

Design Constraints
1. Entity Sets vs Attributes
2. Entity Sets vs Relationship Sets
3. Relationship Cardinalities
4. Weak Entity Sets
5. Inheritance

Types of keys
1. Candidate Keys
2. Super Keys
3. Primary Keys

Translating ER Diagrams To Relational Sets
Questions on Tuple Relational Calculus?

Formula Set up:
1. Query has the form: \( \{ T \mid p(T) \} \)
2. Answer is the set of all tuples \( T \) for which the formula \( p(T) \) evaluates to true.

Available Operations
1. \( \neg p, p \lor q, p \land q \)
   1. \( P \) and \( Q \) are formulas
2. \( \exists R(p(R)) \)
   1. \( R \) is a tuple variable
3. \( \forall R(p(R)) \)
   1. \( R \) is a tuple variable
SQL

• Basic Structure

• Other Ideas
  • Distinct Clause
  • Natural Join
  • Rename Operator
  • Aggregate Functions
    • Group By
  • Nested Sub Queries

\[
\text{select } A_1, A_2, ..., A_n \\
\text{from } r_1, r_2, ..., r_m \\
\text{where } P
\]

$A_i$ represents an attribute
$R_i$ represents a relation
$P$ is a predicate.
Example Question

3. Display the names of all students along with their advisor's name. Note that not every student has an advisor.
Example Question

advisor(s.ID, i.ID)
classroom(building, room_number, capacity)
course(course_id, title, dept_name, credits)
department(dept_name, building, budget)
instructor(ID, name, dept_name, salary)
section(course_id, sec_id, semester, year, building, room_number, time_slot_id)
student(ID, name, dept_name, tot_cred)
takes(ID, course_id, sec_id, semester, year, grade)
teaches(ID, course_id, sec_id, semester, year)

3. Display the names of all students along with their advisor’s name. Note that not every student has an advisor.

    with advise as
    (select s.id as id, i.id from advisor)
    
    select student.name, instructor.name
    from student left outer join (advise inner join instructor on advise.i_id = instructor.id)"
Example Question

advisor(s.ID, i.ID)
classroom(building, room_number, capacity)
course(course.id, title, dept.name, credits)
department(dept.name, building, budget)
instructor(ID, name, dept.name, salary)
section(course.id, sec.id, semester, year, building, room_number, time.slot.id)
student(ID, name, dept.name, tot.cred)
takes(ID, course.id, sec.id, semester, year, grade)
teaches(ID, course.id, sec.id, semester, year)

4. The university needs to determine whether a student is eligible for graduation. Here are the following requirements for graduation:

- the student has taken at least 130 credits
- the student has taken courses in at least five different departments
- the student has taken at least seven courses in a single department. This requirement is not restricted to the department associated with their student ID.

Find all students who are eligible to graduate.
Example Question Cont.

• Break the problem up into sub parts

```sql
with breadth as
  (select id, count(distinct dept_name) as dept_breadth
   from course
   inner join takes on course.course_id = takes.course_id
   inner join student on student.id = takes.id
   group by id)

with depth as
  (select id, count(distinct course_id) as dept_depth
   from course
   inner join takes on course.course_id = takes.course_id
   inner join student on student.id = takes.id
   group by id, depart_name)

with credits as
  (select id from student where total_creds > 130)

select id
from credits
inner join depth on credits.id = depth.id
inner join breadth on breadth.id = depth.id
where dept_depth > 6 and dept_breadth > 4
```
Constraints

Integrity Constraints
1. Primary Key
2. Attribute Constraint
3. Referential Integrity Constraint
4. Global Constraints