CS 4604: Introduction to Database Management Systems

B. Aditya Prakash

Lecture #13: Mid-Term Review
Course Outline

- **Weeks 1–5, 13:** Query/Manipulation Languages
  - Relational Algebra
  - Data definition
  - Programming with SQL

- **Weeks 6–8:** Data Modeling
  - Entity-Relationship (E/R) approach
  - Specifying Constraints
  - Good E/R design

- **Weeks 9–13:** Relational Design
  - The relational model
  - Converting ER to “R”
  - Normalization to avoid redundancy

- **Week 14–15:** Students’ choice
  - Practice Problems
  - XML
  - Query optimization
  - Data mining
Course Outline

- **Weeks 1–5, 13:** Query/Manipulation Languages
  - Relational Algebra
  - Data definition
  - Programming with SQL

- **Weeks 6–8:** Data Modeling
  - Entity-Relationship (E/R) approach
  - Specifying Constraints
  - Good E/R design

- **Weeks 9–13:** Relational Design
  - The relational model
  - Converting ER to “R”
  - Normalization to avoid redundancy

- **Week 14–15:** Students’ choice
  - Practice Problems
  - XML
  - Query optimization
  - Data mining
Basic Operators Covered

- **Remove parts of a single relation:**
  - projection: $\pi_{A,B}(R)$ and SELECT A, B FROM R.
  - selection: $\sigma_C(R)$ and SELECT * FROM R WHERE C.
  - combining projection and selection:
    - $\pi_{A,B}(\sigma_C(R))$
    - SELECT A, B FROM R WHERE C.

- **Set operations** ($R$ and $S$ must have the same attributes, same attribute types, and same order of attributes):
  - union: $R \cup S$ and (R) UNION (S).
  - intersection: $R \cap S$ and (R) INTERSECT (S).
  - difference: $R - S$ and (R) EXCEPT (S).

- **Combine the tuples of two relations:**
  - Cartesian product: $R \times S$ and ... FROM R, S ...
  - Theta-join: $R \bowtie^C S$ and ... FROM R, S WHERE C.
  - Natural join: $R \bowtie S$; in SQL, list the conditions that the common attributes be equal in the WHERE clause.
Renaming

- If two relations have the same attribute, disambiguate the attributes by prefixing the attribute with the name of the relation it belongs to.

- How do we answer the query “Name pairs of students who live at the same address”? Students(Name, Address)
  - We need to take the cross-product of Students with itself?
  - How do we refer to the two “copies” of Students?
  - Use the rename operator.

RA: $\rho_S (A_1, A_2, \ldots, A_n)(R)$ : give R the name S; R has n attributes, which are called A1, A2, \ldots, An in S

SQL: Use the AS keyword in the FROM clause: Students AS Students1 renames Students to Students1.

SQL: Use the AS keyword in the SELECT clause to rename attributes.
Extended Operators

- Powerful operators based on basic relational operators and bag semantics.
- **Sorting**: convert a relation into a list of tuples.
- **Duplicate elimination**: turn a bag into a set by eliminating duplicate tuples.
- **Grouping**: partition the tuples of a relation into groups, based on their values among specified attributes.
- **Aggregation**: used by the grouping operator and to manipulate/combine attributes.
- **Extended projections**: projection on *steroids*.
- **Outerjoin**: extension of joins that make sure every tuple is in the output.
More

- SQL Data Definition Commands
- Constraints
- Triggers
- ...

Prakash 2013
Basic SQL Query

**SELECT [DISTINCT] target-list**

**FROM relation-list**

**WHERE qualification;**

- **Relation-list:** A list of relation names (possibly with range-variable after each name).
- **Target-list:** A list of attributes of relations in relation-list
- **Qualification:** conditions on attributes
- **DISTINCT:** optional keyword for duplicate removal.
  - Default = no duplicate removal!
Representing “Multiplicity”

- Show a many-one relationship by an arrow entering the “one” side.
- Show a one-one relationship by arrows entering both entity sets.
- In some situations, we can also assert “exactly one,” i.e., each entity of one set must be related to exactly one entity of the other set. To do so, we use a rounded arrow.
Weak Entity Sets

- Occasionally, entities of an entity set need “help” to identify them uniquely.
- Entity set $E$ is said to be weak if in order to identify entities of $E$ uniquely, we need to follow one or more many-one relationships from $E$ and include the key of the related entities from the connected entity sets.
- *name* is almost a key for football players, but there might be two with the same name.
- *number* is certainly not a key, since players on two teams could have the same number.
- But *number*, together with the *Team* related to the player by *Plays-on* should be unique.

- Double diamond for *supporting* many-one relationship.
- Double rectangle for the weak entity set.
Weak Entity-Set Rules

- A weak entity set has one or more many-one relationships to other (supporting) entity sets.
  - Not every many-one relationship from a weak entity set need be supporting.

- The key for a weak entity set is its own underlined attributes and the keys for the supporting entity sets.
  - E.g., player-\textit{number} and team-\textit{name} is a key for \textit{Players} in the previous example.
Converting E/R Diagrams to Relational Designs

- **Entity Set** → **Relation**
  - Attribute of Entity Set → Attribute of a Relation

- **Relationship** → **relation** whose attributes are
  - Attribute of the relationship itself
  - Key attributes of the connected entity sets

- **Several special cases:**
  - Weak entity sets.
  - Combining relations (especially for many-one relationships)
  - ISA relationships and subclasses
Exam

- **No** books, no notes, no laptops
- **Only** 1 letter-size page
  - you may use only **one** side

- Duration: 50 mins, during class.

- More or less equal weightage to RA/SQL/E-R