COMS W4111: Introduction to Databases — Week 2 Notes

Class Meeting Date: September 17, 2026 (Week 2) **Instructor:** Prof. Oktie Hassanzadeh
Course Name: COMS W4111 (Introduction to Databases) **Note-Taker:** Panfeng Jiang

Course Administration & Important Announcements

- **HW1 (Domain & Conceptual Model):** Opened September 17, 2026; Due **September 27, 2026 at 11:59 PM**.
- **HW1 Structure (4 Parts):** (1) Choose Domain, (2) ER Model in Diagram Editor, (3) Written Write-up, (4) Explain Your Decision (5 sentences).
- **Next Quiz Deadline:** Due **Thursday, Sep 24, 2026 at 7:00 PM**.
- **AI Usage Policy:** Allowed and encouraged for drafting/critiquing models, but all write-ups must end with an AI usage disclosure.

1. Entity-Relationship (ER) Model Basics

Core Elements: Entity Sets, Relationship Sets & Attributes

- **Entity & Entity Set:** An *entity* is a distinguishable real-world object (e.g., a specific instructor). An *entity set* is a collection of entities of the same type sharing properties (e.g., all instructors).
- **Relationship Set:** A mathematical association among $n \geq 2$ entities: $\{\,(e_1, e_2, \ldots, e_n) \mid e_1 \in E_1, e_2 \in E_2, \ldots, e_n \in E_n\,\}$. Concretely, `advisor` associates `student` entities with `instructor` entities.
- **Attributes:** Descriptive properties possessed by members of an entity set (e.g., `instructor = (ID, name, salary)`).

Standard Visual Notation in ER Diagrams

- **Rectangles:** Represent Entity Sets (e.g., `instructor`, `student`).
- **Diamonds:** Represent Relationship Sets (e.g., `advisor`).
- **Ellipses:** Represent Attributes, with the **Primary Key Attribute Underlined** (e.g., `ID`).

2. Visual Notations & Cardinality Constraints

Comparing Design Notations

Notation	Key Strength	Where You Meet It
Textbook ER	Highly precise; explicit diamond relationships	Silberschatz textbook (DSC7) & exams
Crow's Foot	Compact; reads cardinality at a glance	Industry tools, Lucidchart, Vertabelo, draw.io
UML Class Diagram	Integrates directly with software design	Software engineering teams

Crow's Foot Cardinality Symbols

Read each end of a relationship line on its own:

- **|| (Two bars):** Exactly One   **O| (Circle + Bar):** Zero or One (Optional)
- **|< (Bar + Crow's Foot):** One or More   **O< (Circle + Crow's Foot):** Zero or More

3. Keys in Practice: Natural, Surrogate & Composite

Key Taxonomy Comparison

Key	Kind	Source / Origin	Properties & Usage
`UNI`	Surrogate	Assigned by Columbia University	Stable, system-generated, no business meaning
`CUID`	Surrogate	Assigned on ID card	Stable, system-generated
`email`	Natural	Domain fact	Self-describing, but can change over time
`(dept, sec, sem, yr)`	Natural, Composite	Built from section facts	Multiple attributes combined

Primary Key Immutability Principle

Primary keys are strongly recommended to be **immutable**. If a natural key (like `email`) changes, that change must cascade through every foreign key in every junction table. Surrogate keys avoid this cascading cost.

4. Relational Algebra: Core Focus & Mechanics

Relational algebra is a procedural query language consisting of operations that take one or two relations as input and produce a new relation as output.

The Closure Property & Expression Composition

Because the result of any algebraic operation is itself a relation, operations can be nested and composed into complex expressions: $\pi_{\text{name}}\left(\sigma_{\text{dept\_name}=\text{'Physics'}}(\text{instructor})\right)$. This **closure property** allows building modular, verifiable query logic step-by-step.

1. Selection (σ): Filtering Tuples

- **Definition:** $\sigma_P(r)$ selects tuples from relation r that satisfy predicate P. Filters **rows**.
- **SQL Equivalent:** Corresponds directly to the `WHERE` clause.
- **Example:** $\sigma_{\text{salary}>80000}(\text{instructor})$ returns all instructor rows earning over \$80,000.

2. Projection (π): Selecting Attributes & Duplicate Elimination

- **Definition:** $\pi_{A_1,A_2,\ldots,A_k}(r)$ returns relation r restricted to attributes $A_1,\ldots,A_k$. Filters **columns**.

Set Semantics vs. Bag Semantics (Exam Focus)

- **Relational Algebra (π):** Uses **Set Semantics**. Duplicate rows are **automatically eliminated** ($\pi_{\text{dept\_name}}(\text{instructor})$ returns 7 departments for 12 instructors).
- **SQL (`SELECT`):** Uses **Bag Semantics**. SQL retains duplicates by default for performance (`SELECT dept_name FROM instructor` returns 12 rows). You MUST write `SELECT DISTINCT` to mimic algebraic projection.

3. The Join Operator ($\bowtie$): Definition & Optimizer Freedom

Join Definition: Selection over Cartesian Product

The join operator is **defined** as a selection over a Cartesian product:

$$r \bowtie_\theta s \;=\; \sigma_\theta(r \times s)$$

It is not a fundamental new capability, but a shorthand for a pattern used constantly.

- **Why the Definition Matters for Query Optimization:** Because $r \bowtie_\theta s$ is algebraically defined as $\sigma_\theta(r \times s)$, the **Query Optimizer** is free to execute it using any physical algorithm that yields the correct result — including *Nested-Loop Join*, *Hash Join*, or *Sort-Merge Join*.
- **Comma Syntax vs. Explicit `JOIN ... ON`:**
  - *Implicit (Comma):* `SELECT * FROM instructor i, teaches t WHERE i.ID = t.ID;`
  - *Explicit (Preferred):* `SELECT * FROM instructor i JOIN teaches t ON i.ID = t.ID;`
- **Best Practice Rule:** Always write `JOIN ... ON`. Omitting the `ON` predicate results in an immediate syntax error, preventing accidental and costly Cartesian products.

4. RelAX: Online Relational Algebra Calculator

To practice algebraic expressions interactively, the course recommends the open-source browser tool **RelAX** (`dbis-uibk.github.io/relax`) using the *Silberschatz - UniversityDB* dataset.