Today

- Quiz 7
- Normalization (cont)

Assignments

- HW7 out
- Proj 1 out
Enforcing functional dependencies

For our table

\[ \text{Emp}(\text{eid}, \text{name, dept, deptname}) \]
- with key eid
- and FD dept → deptname

Q: Although eid is the key for this table ... is it still possible for there to be 2 names for the same department?

- YES!
- The DBMS can enforce keys, but not non-key FDs

What are possible non-key, non-trivial FDs in these examples?

\[ \text{Customer}(\text{cid, address, city, state, zip}) \]
- zip → state
- address, city, state → zip

\[ \text{Enrollment}(\text{student_id, class_id, instructor_id, student_name, instructor_name}) \]
- instructor_id → instructor_name
- student_id → student_name
Not all redundancy is covered by FDs

- Lets say we split employee into two tables:
  
  \[
  \text{Emp}_1(ssn, \text{name}, \text{salary}, \text{birthdate})
  \]
  
  \[
  \text{Emp}_2(ssn, \text{name}, \text{address})
  \]

- name stored redundantly, same employee can have multiple names, etc.

Q: How can we fix this redundancy? ... remove name from a table

Example decomposition based on FDs

- The table:
  
  \[
  \text{Emp}(ssn, \text{name}, \text{birthdate}, \text{address}, dnum, dname, dmgr)
  \]
  
  \[
  \quad \text{dnum} \rightarrow dname, dmgr
  \]

- We can move the non-trivial FDs into their own table
  
  \[
  \text{Dept}(dnum, dname, dmgr)
  \]

- Where the original becomes:
  
  \[
  \text{Emp}(ssn, \text{name}, \text{birthdate}, \text{address}, dnum)
  \]
  
  \[
  \quad \text{and Emp.dnum is an FK to Dept.dnum}
  \]
Normalization Based on FDs

1. Identify all FDs
   - FDs implied by the key
   - FDs not (directly) implied by the keys (the “troublesome” ones)

2. Generate one or more new tables from the troublesome FDs
   - each new table should only have FDs implied by the key

3. Remove attributes from the original table that are functionally dependent on troublesome FDs

4. Specify appropriate FKs to these new tables
Reasoning about Functional Dependencies

- For the table:
  \[ \text{EmpDept}(\text{eid}, \text{name}, \text{dept\_id}, \text{dept\_name}) \]

- Two natural FDs are:
  \[
  \begin{align*}
  &\text{eid} \to \text{dept\_id} \quad \text{a key FD} \\
  &\text{dept\_id} \to \text{dept\_name} \quad \text{a non-key FD}
  \end{align*}
  \]

- These two FDs imply: \( \text{eid} \to \text{dept\_name} \)
  \[
  \begin{align*}
  &\text{if two rows agree on \text{eid}, then by } \text{eid} \to \text{dept\_id} \text{ they agree on } \text{dept\_id} \\
  &\text{if they agree on } \text{dept\_id}, \text{ then by } \text{dept\_id} \to \text{dept\_name} \text{ they agree on } \text{dept\_name}
  \end{align*}
  \]

The FDs implied by a given set \( F \) of FDs is called the \textit{closure of } F \ldots denoted \( F^+ \).

FD Inference Rules

- \( F^+ \) can be computed using these axioms
  \[
  \begin{align*}
  &\text{Reflexivity: if } Y \subseteq X, \text{ then } X \to Y \\
  &\text{Augmentation: if } X \to Y, \text{ then } XZ \to YZ \text{ for any } Z \\
  &\text{Transitivity: if } X \to Y \text{ and } Y \to Z \text{ then } X \to Z
  \end{align*}
  \]

- Apply these rules repeatedly to \( F \) until we no longer produce any new FDs

- This is a “sound” and “complete” inference procedure
  \[
  \begin{align*}
  &\text{Soundness: Only FDs in } F^+ \text{ are generated} \\
  &\text{Completeness: Generates all FDs in } F^+
  \end{align*}
  \]
Exercise

Given the FD's \{A \rightarrow C, B \rightarrow C, CD \rightarrow E\}

Prove \(AD \rightarrow E\)

1. \(A \rightarrow C\)  
   \hspace{1cm} Given

2. \(AD \rightarrow CD\)  
   \hspace{1cm} Augmentation

3. \(CD \rightarrow E\)  
   \hspace{1cm} Given

4. \(AD \rightarrow E\)  
   \hspace{1cm} Transitivity