Database Design and Normalization

Chapter 10
Formal Design of Database

- Data about the organization that is to be stored
- Rules of organization
- Decision on the number of relations and their attributes (too few or too many)
Example of Anomalies and Errors

- Errors arise in updates. ASSN (Name, Room#, Tel)

**Rule:** Tel# is associated with Room#, not with the name of the person.

Smith has moved from Room 49 to Room 46

**Update** ASSN

**Set** Room = 46

**WHERE** Room=49, Name = ‘Smith’

But (Smith, 46, 3512) is wrong. Tel# 3512 is associated with Room 49.

Problem: Relation ASSN has too many attributes.
Better Design

R1(Name, Room)
R2(Room, Tel.)

To get tel# of Smith perform join between R1 and R2.

This join is lossless.
Example of Anomalies and Errors

Update Example: Donor(Name, address, year, amount)
Smith, 222 Grant, 1959, 10
Smith, 222 Grant, 1957, 20

Update Donor
Set Address: 351 State St,
WHERE Address = 222 Grant

System has no memory and update will be applied only to the first tuple.
The Evils of Redundancy

- Redundancy is at the root of several problems associated with relational schemas:
  - redundant storage, insert/delete/update anomalies
- Integrity constraints, in particular functional dependencies, can be used to identify schemas with such problems and to suggest refinements.
- Main refinement technique: decomposition (replacing ABCD with, say, AB and BCD, or ACD and ABD).
- Decomposition should be used judiciously:
  - Is there reason to decompose a relation?
  - What problems (if any) does the decomposition cause?
Join with Loss of Information

Example

STORE(Dept, items, color)
R1(Dept, item)
R2(item, color)

- R1 and R2 simple projection of STORE

Query Find a Dept. which do not sell red items

Ans1 Use STORE relation
Join with Loss of Information

STORE

<table>
<thead>
<tr>
<th>Dept</th>
<th>item</th>
<th>color</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Pencil</td>
<td>Red</td>
</tr>
<tr>
<td>D5</td>
<td>Pencil</td>
<td>Black</td>
</tr>
<tr>
<td>D3</td>
<td>Pen</td>
<td>Black</td>
</tr>
</tbody>
</table>

R1

<table>
<thead>
<tr>
<th>Dept</th>
<th>item</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Pencil</td>
</tr>
<tr>
<td>D5</td>
<td>Pencil</td>
</tr>
<tr>
<td>D3</td>
<td>Pen</td>
</tr>
</tbody>
</table>

R2

<table>
<thead>
<tr>
<th>item</th>
<th>color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pencil</td>
<td>Red</td>
</tr>
<tr>
<td>Pencil</td>
<td>Black</td>
</tr>
<tr>
<td>Pen</td>
<td>Black</td>
</tr>
</tbody>
</table>

These designs are not the same semantically
Normalization (Formal Design of DB/Schema)

Normalization is a relation decomposition (projection operation) process which is lossless.

- No artificial constraints
- Freedom from undesirable anomalies due to insertion, update and deletion.
- Restructuring of data is minimized when extensions are made.
Rules of Organization

1) Functional dependency (FD) among date items.
2) Multivalued dependency (MVD)
3) Misc

Strategy of DB Design

1) Recognize all FD’s in the DB
2) Use the FD’s to obtain the optimal design of DB
Functional Dependencies

1) Functional dependency (FD) among date items.
2) Multivalued dependency (MVD)
3) Misc

Strategy of DB Design

1) Recognize all FD’s in the DB
2) Use the FD’s to obtain the optimal design of DB
Definition: Given a table $T$ containing at least two attributes, say $A$ and $B$, we say $A \rightarrow B$ (A functionally determines B) if and only if it is the intent of the designer, for any set of rows in the $T$, that two rows in $T$ cannot agree in value for $A$ and disagree in value for $B$.

FDs are determined from understanding the data items and rules of the enterprise.
Functional Dependencies (FDs)

- A functional dependency $X \rightarrow Y$ holds over relation $R$ if, for every allowable instance $r$ of $R$:
  - $t_1 \in r$, $t_2 \in r$, $\pi_X(t_1) = \pi_X(t_2)$ implies $\pi_Y(t_1) = \pi_Y(t_2)$
  - i.e., given two tuples in $r$, if the $X$ values agree, then the $Y$ values must also agree. ($X$ and $Y$ are sets of attributes.)

- An FD is a statement about all allowable relations.
  - Must be identified based on semantics of application.
  - Given some allowable instance $r_1$ of $R$, we can check if it violates some FD $f$, but we cannot tell if $f$ holds over $R$!

- $K$ is a candidate key for $R$ means that $K \rightarrow R$
  - However, $K \rightarrow R$ does not require $K$ to be minimal!
Example: Constraints on Entity Set

- Consider relation obtained from Hourly_Emps:
  - Hourly_Emps (ssn, name, lot, rating, hrly_wages, hrs_worked)

- **Notation**: We will denote this relation schema by listing the attributes: SNLRWH
  - This is really the set of attributes \{S,N,L,R,W,H\}.
  - Sometimes, we will refer to all attributes of a relation by using the relation name. (e.g., Hourly_Emps for SNLRWH)

- **Some FDs on Hourly_Emps**:
  - ssn is the key: S \rightarrow SNLRWH
  - rating determines hrly_wages: R \rightarrow W
Example (Contd.)

- **Problems due to R→ W:**
  - **Update anomaly:** Can we change W in just the 1st tuple of SNLRWH?
  - **Insertion anomaly:** What if we want to insert an employee and don’t know the hourly wage for his rating?
  - **Deletion anomaly:** If we delete all employees with rating 5, we lose the information about the wage for rating 5!

<table>
<thead>
<tr>
<th>S</th>
<th>N</th>
<th>L</th>
<th>R</th>
<th>W</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>123-22-3666</td>
<td>Attishoo</td>
<td>48</td>
<td>8</td>
<td>10</td>
<td>40</td>
</tr>
<tr>
<td>231-31-5368</td>
<td>Smiley</td>
<td>22</td>
<td>8</td>
<td>10</td>
<td>30</td>
</tr>
<tr>
<td>131-24-3650</td>
<td>Smethurst</td>
<td>35</td>
<td>5</td>
<td>7</td>
<td>30</td>
</tr>
<tr>
<td>434-26-3751</td>
<td>Guldu</td>
<td>35</td>
<td>5</td>
<td>7</td>
<td>32</td>
</tr>
<tr>
<td>612-67-4134</td>
<td>Madayan</td>
<td>35</td>
<td>8</td>
<td>10</td>
<td>40</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S</th>
<th>N</th>
<th>L</th>
<th>R</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>123-22-3666</td>
<td>Attishoo</td>
<td>48</td>
<td>8</td>
<td>40</td>
</tr>
<tr>
<td>231-31-5368</td>
<td>Smiley</td>
<td>22</td>
<td>8</td>
<td>30</td>
</tr>
<tr>
<td>131-24-3650</td>
<td>Smethurst</td>
<td>35</td>
<td>5</td>
<td>30</td>
</tr>
<tr>
<td>434-26-3751</td>
<td>Guldu</td>
<td>35</td>
<td>5</td>
<td>32</td>
</tr>
<tr>
<td>612-67-4134</td>
<td>Madayan</td>
<td>35</td>
<td>8</td>
<td>40</td>
</tr>
</tbody>
</table>
Refining an ER Diagram

- **1st diagram translated:**
  - Workers(S,N,L,D,S)
  - Departments(D,M,B)
    - Lots associated with workers.

- Suppose all workers in a dept are assigned the same lot: \( D \rightarrow L \)

- Redundancy; fixed by:
  - Workers2(S,N,D,S)
  - Dept_Lots(D,L)

- Can fine-tune this:
  - Workers2(S,N,D,S)
  - Departments(D,M,B,L)
**Reasoning About FDs**

- Given some FDs, we can usually infer additional FDs:
  - \( \text{ssn} \rightarrow \text{did}, \text{did} \rightarrow \text{lot} \) implies \( \text{ssn} \rightarrow \text{lot} \)

- An **FD \( f \) is implied by** a set of FDs \( F \) if \( f \) holds whenever all FDs in \( F \) hold.
  - \( F^+ = \text{closure of } F \) is the set of all FDs that are implied by \( F \).

- **Armstrong’s Axioms** (\( X, Y, Z \) are sets of attributes):
  - **Reflexivity**: If \( Y \subseteq X \), then \( X \rightarrow Y \), e.g. \( AB \rightarrow A \)
  - **Augmentation**: If \( X \rightarrow Y \), then \( XZ \rightarrow YZ \) for any \( Z \)
  - **Transitivity**: If \( X \rightarrow Y \) and \( Y \rightarrow Z \), then \( X \rightarrow Z \)

- **These are sound and complete inference rules for FDs** (i.e. no other rules of implications can be added to increase their effectiveness)
Reasoning About FDs (Contd.)

- Couple of additional rules (that follow from AA):
  - Union: If \( X \rightarrow Y \) and \( X \rightarrow Z \), then \( X \rightarrow YZ \)
  - Decomposition: If \( X \rightarrow YZ \), then \( X \rightarrow Y \) and \( X \rightarrow Z \)

- Example: Contracts(\(cid, sid, fid, did, pid, qty, value\)), and:
  - C is the key: \( C \rightarrow CSJDPQV\)
  - Project purchases each part using single contract: \( JP \rightarrow C\)
  - Dept purchases at most one part from a supplier: \( SD \rightarrow P\)

- \( JP \rightarrow C, C \rightarrow CSJDPQV \) imply \( JP \rightarrow CSJDPQV\)
- \( SD \rightarrow P \) implies \( SDJ \rightarrow JP\)
- \( SDJ \rightarrow JP, JP \rightarrow CSJDPQV \) imply \( SDJ \rightarrow CSJDPQV\)
Closure, Cover and Minimal Cover

- Closure of set of FDs: Given a set $F$ of FDs on attributes of a table $T$, the closure of $F$, $F^+$, is the set of all FDs implied by $F$.

- Computing the closure of a set of FDs can be expensive. (Size of closure is exponential in # attrs!)