CS 405G: Introduction to Database Systems

Functional Dependency
Today’s Topic

- Functional Dependency.
- Normalization
- Decomposition
- BCNF
Motivation

- How do we tell if a design is bad, e.g., $WorkOn(EID, Ename, PID, Pname, Hours)$?
- This design has *redundancy*, because the name of an employee is recorded multiple times, once for each project the employee is taking.

<table>
<thead>
<tr>
<th>EID</th>
<th>PID</th>
<th>Ename</th>
<th>Pname</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>1234</td>
<td>10</td>
<td>John Smith</td>
<td>B2B platform</td>
<td>10</td>
</tr>
<tr>
<td>1123</td>
<td>9</td>
<td>Ben Liu</td>
<td>CRM</td>
<td>40</td>
</tr>
<tr>
<td>1234</td>
<td>9</td>
<td>John Smith</td>
<td>CRM</td>
<td>30</td>
</tr>
<tr>
<td>1023</td>
<td>10</td>
<td>Susan Sidhuk</td>
<td>B2B platform</td>
<td>40</td>
</tr>
</tbody>
</table>

Update anomaly
Insert anomaly: Bar not taking classes
Delete anomaly: Bart drops all classes
Why redundancy is bad?

- Waste disk space.
- What if we want to perform update operations to the relation
  - INSERT a new project that no employee has been assigned to it yet.
  - UPDATE the name of “John Smith” to “John L. Smith”
  - DELETE the last employee who works for a certain project

<table>
<thead>
<tr>
<th>EID</th>
<th>PID</th>
<th>Ename</th>
<th>Pname</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>1234</td>
<td>10</td>
<td>John Smith</td>
<td>B2B platform</td>
<td>10</td>
</tr>
<tr>
<td>1123</td>
<td>9</td>
<td>Ben Liu</td>
<td>CRM</td>
<td>40</td>
</tr>
<tr>
<td>1234</td>
<td>9</td>
<td>John Smith</td>
<td>CRM</td>
<td>30</td>
</tr>
<tr>
<td>1023</td>
<td>10</td>
<td>Susan Sidhuk</td>
<td>B2B platform</td>
<td>40</td>
</tr>
</tbody>
</table>
Functional Dependency

- FDs are **constraints** that are derived from the *meaning* and *interrelationships* of the data attributes.

- Functional dependencies (FDs) are used to specify *formal measures* of the "goodness" of relational designs.

- FDs and keys are used to define **normal forms** for relations.
**Functional dependencies**

- A functional dependency (FD) has the form $X \rightarrow Y$, where $X$ and $Y$ are sets of attributes in a relation $R$.
- $X \rightarrow Y$ means that whenever two tuples in $R$ agree on all the attributes in $X$, they must also agree on all attributes in $Y$.
- $t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]$.

<table>
<thead>
<tr>
<th>$X$</th>
<th>$Y$</th>
<th>$Z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>c</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>d</td>
</tr>
</tbody>
</table>

Must be “b”  
Could be anything, e.g. d
FD examples

Address (street_address, city, state, zip)

- street_address, city, state -> zip
- zip -> city, state
- zip, state -> zip?
  - This is a trivial FD
  - Trivial FD: LHS ⊇ RHS
- zip -> state, zip?
  - This is non-trivial, but not completely non-trivial
  - Completely non-trivial FD: LHS ∩ RHS = ?
Functional Dependencies

- An FD is a property of the attributes in the schema R
- The constraint must hold on every relation instance r(R)
- If K is a key of R, then K functionally determines all attributes in R (since we never have two distinct tuples with t1[K]=t2[K])
Keys redefined using FD’s

Let $\text{attr}(R)$ be the set of all attributes of $R$, a set of attributes $K$ is a (candidate) $\text{key}$ for a relation $R$ if

- $K \rightarrow \text{attr}(R) - K$, and
  - That is, $K$ is a “super key”
- No proper subset of $K$ satisfies the above condition
  - That is, $K$ is minimal (full functional dependent)
- $\text{Address (street\_address, city, state, zip)}$
  - \{street\_address, city, state, zip\} Super key
  - \{street\_address, city, zip\} Super key
  - \{street\_address, zip\} Key
  - \{zip\} Non-key
Reasoning with FD’s

Given a relation $R$ and a set of FD’s $F$

- Does another FD follow from $F$?
  - Are some of the FD’s in $F$ redundant (i.e., they follow from the others)?
- Is $K$ a key of $R$?
  - What are all the keys of $R$?
Attribute closure

- Given $R$, a set of FD’s $F$ that hold in $R$, and a set of attributes $Z$ in $R$:
  The closure of $Z$ (denoted $Z^+$) with respect to $F$ is the set of all attributes $\{A_1, A_2, \ldots\}$ functionally determined by $Z$ (that is, $Z \rightarrow A_1 A_2 \ldots$)
- Algorithm for computing the closure
  - Start with closure $= Z$
  - If $X \rightarrow Y$ is in $F$ and $X$ is already in the closure, then also add $Y$ to the closure
  - Repeat until no more attributes can be added
A more complex example

WorkOn(EID, Ename, email, PID, Pname, Hours)

- EID -> Ename, email
- email -> EID
- PID -> Pname
- EID, PID -> Hours

(Not a good design, and we will see why later)
Example of computing closure

- F includes:
  - $EID \rightarrow Ename, email$
  - $email \rightarrow EID$
  - $PID \rightarrow Pname$
  - $EID, PID \rightarrow Hours$
- $\{ PID, email \} = ?$
- closure $= \{ PID, email \}$
- $email \rightarrow EID$
  - Add $EID$; closure is now $\{ PID, email, EID \}$
- $EID \rightarrow Ename, email$
  - Add $Ename, email$; closure is now $\{ PID, email, EID, Ename \}$
- $PID \rightarrow Pname$
  - Add $Pname$; close is now $\{ PID, Pname, email, EID, Ename \}$
- $EID, PID \rightarrow hours$
  - Add $hours$; closure is now all the attributes in $WorksOn$
Using attribute closure

Given a relation $R$ and set of FD’s $\mathbf{F}$

- Does another FD $X \rightarrow Y$ follow from $\mathbf{F}$?
  - Compute $X^+$ with respect to $\mathbf{F}$
  - If $Y \subseteq X^+$, then $X \rightarrow Y$ follow from $\mathbf{F}$

- Is $K$ a super key of $R$?
  - Compute $K^+$ with respect to $\mathbf{F}$
  - If $K^+$ contains all the attributes of $R$, $K$ is a super key

- Is a super key $K'$ a key of $R$?
  - Test where $K' = K - \{ a \mid a \in K \}$ is a superkey of $R$ for all possible $a$
Rules of FD’s

- Armstrong’s axioms
  - Reflexivity: If $Y \subseteq X$, then $X \rightarrow Y$
  - Augmentation: If $X \rightarrow Y$, then $XZ \rightarrow YZ$ for any $Z$
  - Transitivity: If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$

- Rules derived from axioms
  - Splitting: If $X \rightarrow YZ$, then $X \rightarrow Y$ and $X \rightarrow Z$
  - Combining: If $X \rightarrow Y$ and $X \rightarrow Z$, then $X \rightarrow YZ$
Using rules of FD’s

Given a relation $R$ and set of FD’s $F$

- Does another FD $X \rightarrow Y$ follow from $F$?
  - Use the rules to come up with a proof

- Example:
  - $F$ includes:
    - $EID \rightarrow Ename, email; email \rightarrow EID; EID, PID \rightarrow Hours,$
    - $Pid \rightarrow Pname$
  - $PID, email \rightarrow hours$?
    - $email \rightarrow EID$ (given in $F$)
    - $PID, email \rightarrow PID, EID$ (augmentation)
    - $PID, EID \rightarrow hours$ (given in $F$)
    - $PID, email \rightarrow hours$ (transitivity)
Example of redundancy

- **WorkOn** ($EID$, $Ename$, $email$, $PID$, $hour$)
- We say $X \rightarrow Y$ is a **partial dependency** if there exist a $X' \subseteq X$ such that $X' \rightarrow Y$
  - e.g. $EID$, $email$-$\rightarrow$ $Ename$, $email$
- Otherwise, $X \rightarrow Y$ is a **full dependency**
  - e.g. $EID$, $PID$-$\rightarrow$ $hours$

<table>
<thead>
<tr>
<th>EID</th>
<th>PID</th>
<th>Ename</th>
<th>email</th>
<th>Pname</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>1234</td>
<td>10</td>
<td>John Smith</td>
<td><a href="mailto:jsmith@ac.com">jsmith@ac.com</a></td>
<td>B2B platform</td>
<td>10</td>
</tr>
<tr>
<td>1123</td>
<td>9</td>
<td>Ben Liu</td>
<td><a href="mailto:blni@ac.com">blni@ac.com</a></td>
<td>CRM</td>
<td>40</td>
</tr>
<tr>
<td>1234</td>
<td>9</td>
<td>John Smith</td>
<td><a href="mailto:jsmith@ac.com">jsmith@ac.com</a></td>
<td>CRM</td>
<td>30</td>
</tr>
<tr>
<td>1023</td>
<td>10</td>
<td>Susan Sidhuk</td>
<td><a href="mailto:ssidhuk@ac.com">ssidhuk@ac.com</a></td>
<td>B2B platform</td>
<td>40</td>
</tr>
</tbody>
</table>

Why is that