CIS 451/551

week 5: Normalization
Goal

• formalize notions of clean design
• motto: “One fact in one place”

• deletion anomaly
• insertion anomaly
• update anomaly
brief overview

1. First Normal Form (1NF)
   • all entries are atomic
2. Second Normal Form (2NF)
   • remove partial dependencies
3. Third Normal Form (3NF)
   • remove transitive dependencies
4. Boyce-Codd Normal Form (BCNF)
   • like 3NF but deal with problems from multiple primary keys
5. Fourth Normal Form (4NF)
   • like BCNF but for multi-valued dependencies
6. others: 5NF, 6NF (obscure)
first normal form

• all relational schemas are in 1NF by assumption
• the idea is that all entries contain one piece of information
• .... it cannot be broken up
• no sets, subfields in an attribute

• “one place holds one fact” 😊
• this rule is frequently broken !!
A → B
is taken to mean that the value of B depends on the value of A, in other words, if any two rows in a table have the same A entry, they must always have the same B entry

• for example, zipcode → state
• related to the concept of keys
partial dependency

• a *full* functional dependency $X \rightarrow Y$ is the case when removing anything from $X$ will cause the dependency to not hold
• also called *irreducibly dependent*
• here $X$ and $Y$ can be sets of attributes
2NF: second normal form

- R is in 2NF if every attribute not belonging to the primary key is fully functionally dependent on the primary key.
- "no partial dependencies"

(Example) STUDENT_ENROLL: ssn, crn, name, bdate
with partial dependency ssn → name, bdate

- should be decomposed
Given a relation $R$ having a violation of $X \rightarrow Y$ we can decompose $R$ as follows:

$R = (R1 \bowtie R2) = X \cup Y$

where $R1 = R - Y$ and $R2 = X \cup Y$.

Note that $R = R1 \bowtie R2$.
example decomposition

• let R = ssn, crn, name, bdate
• ... and ssn → name, bdate be a violating (of 2NF) FD

• according to the decomposition strategy
  • R1 = R-Y = ssn crn
  • R2 = X ∪ Y = ssn name bdate
3NF: third normal form

• consider a chain of FDs of the form
  • \(<key> \rightarrow <non-key> \rightarrow <non-key>\)
  • the second FD \(<non-key> \rightarrow <non-key>\) is called a \textit{transitive} dependency

• \(R\) is in 3NF if it is in 2NF and has no transitive dependencies

• \textit{(example)} \(SUPPLIER: \ s\#, \ city, \ status\)
  • FDs \(s\# \rightarrow city, \ status\) and \(city \rightarrow status\)
  • in other words, the status of the supplier depends only on the city they are in
  • \(city \rightarrow status\) is a transitive dependency and thus \(SUPPLIER\) is not in 3NF
key terminology

• a **superkey** is a unique identifier (note that it may not be minimal)
  • CAR: license, make, model, year, vin
  • license is a superkey, so is (license, make) and (license, vin, year)

• a **candidate key** is a minimal superkey
  • CAR has two candidate keys
    1. license
    2. vin

• the **primary key** is a designated candidate key

• any candidate key which is not the primary key is an **alternate key**

• alternate keys can be identified/enforced using the UNIQUE key word in SQL
more on functional dependencies

• let F be a set of functional dependencies
• $F^+$ is the set of fds that can be derived from F (in obvious ways)
• formally use Armstrong’s Axioms
  let X, Y, and Z be sets of attributes
  • (reflexivity) if $Y \subseteq X$, then $X \rightarrow Y$
  • (transitivity) if $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$
  • (augmentation) if $X \rightarrow Y$, then $XZ \rightarrow YZ$

• (informal) a minimal cover for a set of fds F is a "smaller" set G such that $F^+ = G^+$
formal definition of 3NF

A relation R is in 3NF if for all fds $\alpha \rightarrow \beta \in F^+$ that hold on R at least one of the following conditions is true

- $\alpha \rightarrow \beta$ is trivial,
- $\alpha$ is a superkey for R, or
- each attribute $A \in \beta$ belongs to a candidate key of R
a previous example

• SUPPLIER: s#, city, status
  • FDs $s# \rightarrow city, status$ and $city \rightarrow status$

• CK (candidate keys): s#

• 3NF violation: $city \rightarrow status$
  • it is not trivial
  • $city$ is not a superkey
  • $status$ does not belong to any candidate key

• So SUPPLIER is not in 3NF according to the formal definition either
example with multiple CKs

- **R**: student, prof, dept
- **F**: $p \rightarrow d$, $sd \rightarrow p$
- the idea is that a prof (p) is in only one department (d) and a student (s) can be in several departments and in each department they have one advisor
- candidate keys: sd, sp
- **R** is in 3NF, consider
  - $sd \rightarrow p$: sd is a superkey
  - $p \rightarrow d$: d belongs to a CK
  - (this is not really a full proof since not looking at ALL fds)

but R still has some redundancy!
BCNF: Boyce-Codd Normal Form

A relation R is in BCNF if for all fds $\alpha \rightarrow \beta \in F^+$ that hold on R at least one of the following conditions is true

- $\alpha \rightarrow \beta$ is trivial or
- $\alpha$ is a superkey for R

notes
- $\alpha$ holds on R if the attributes in $\alpha$ are also in R
- $\alpha \rightarrow \beta$ is trivial if $\beta \subseteq \alpha$
last example is not BCNF

• R: student, prof, dept
• F: $p \rightarrow d$, $sd \rightarrow p$
• CK: sd, sp

• $p \rightarrow d$ is a violation of BCNF
  • not trivial
  • p not a superkey of R

• BCNF decomposition
  • R1: student, prof
  • R2: prof, dept

might be ugly
need a join to enforce $sd \rightarrow p$
3NF versus BCNF

- if R is in BCNF, then R is in 3NF
- if R is in 3NF and has only one CK, then R is in BCNF

- a decomposition is *dependency preserving* if all FDs can be verified without computing a join
- for every R,F, there is a decomposition into 3NF that is dependency preserving
- this is not always true for BCNF
exercise 1

• relation R: A B C D E F G H I J
• dependencies F:
  • $AB \rightarrow C$
  • $A \rightarrow DE$
  • $B \rightarrow F$
  • $F \rightarrow GH$
  • $D \rightarrow IJ$

• decompose this
exercise 2

• relation R: A B C D E
• dependencies F:
  • $A \rightarrow B$
  • $C \rightarrow D$

• identify highest normal form
• decompose to 3NF/BCNF
exercise 3

• relation R:  A  B  C  D

• dependencies F:
  • C → A
  • C → D
  • B → C

• identify highest normal form
• decompose to 3NF/BCNF
exercise 4

• relation R: A B C D

• dependencies F:
  • \( D \rightarrow A \)
  • \( B \rightarrow C \)

• identify highest normal form

• decompose to 3NF/BCNF
exercise 5

• relation R: A B C D
• dependencies F:
  • $D \rightarrow A$
  • $ABC \rightarrow D$

• identify highest normal form
• decompose to 3NF/BCNF
exercise 6

• relation R:  A  B  C  D
• dependencies F:
  • $A \rightarrow B$
  • $BC \rightarrow D$
  • $A \rightarrow C$

• identify highest normal form
• decompose to 3NF/BCNF
exercise 7

• relation R: A B C D

• dependencies F:
  • $AB \rightarrow C$
  • $AB \rightarrow D$
  • $C \rightarrow A$
  • $D \rightarrow B$

• identify highest normal form
• decompose to 3NF/BCNF
exercise 8

• find a minimal cover, then decompose
• R: A B C D E F G H
• F:
  • $A \rightarrow B$
  • $ABCD \rightarrow E$
  • $EF \rightarrow G$
  • $EF \rightarrow H$
  • $ABCDF \rightarrow EG$
exercise 9 (trick)

Consider the CANADA_POST relation, illustrated here. The only fd is \textit{postOfficeCity}→\textit{country}.

There appears to be no normal form violation but there is clearly redundancy (since country is always ‘Canada’). What’s wrong?

<table>
<thead>
<tr>
<th>postOfficeCity</th>
<th>country</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vancouver</td>
<td>Canada</td>
</tr>
<tr>
<td>Edmonton</td>
<td>Canada</td>
</tr>
<tr>
<td>Tuktoyaktuk</td>
<td>Canada</td>
</tr>
<tr>
<td>Toronto</td>
<td>Canada</td>
</tr>
<tr>
<td>Corner Brook</td>
<td>Canada</td>
</tr>
<tr>
<td>Saskatoon</td>
<td>Canada</td>
</tr>
</tbody>
</table>
multi-valued dependencies (MVD)

- an mvd such as \textit{class} \rightarrow \textit{text} indicates that each instance of a particular class has the same group of texts
- similar to an fd but refers to several rows
- can express as an fd if we break 1NF
  - \textit{class} \rightarrow \text{set_of(text)}
- describes redundancy problems not expressible with fds
if each instance of a class (such as CIS451) uses the same group of texts, regardless of the instructor, we’d have redundancy in this table. Problem can arise by poor modeling – using a 3-way relationship instead of two 2-way ones.

Here the rule $\text{class} \rightarrow \text{text}$ says that each time we add a new instructor for CIS451, we have to enter two rows for the two texts (which are the same as for the other instructor).
4NF: Fourth Normal Form

A relation R is in 4NF if for all mvds $\alpha \rightarrow \beta$ that hold on R at least one of the following conditions is true
- $\alpha \rightarrow \beta$ is trivial or
- $\alpha$ is a superkey for R

notes
- a fd is also a mvd, that is $\alpha \rightarrow \beta$ implies $\alpha \rightarrow \beta$
- therefore if R is in 4NF then R is in BCNF
decompose class-instr-text into 4NF

split based on $class \rightarrow text$ in the usual way

<table>
<thead>
<tr>
<th>class</th>
<th>instructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIS451</td>
<td>CW</td>
</tr>
<tr>
<td>MTH231</td>
<td>AB</td>
</tr>
<tr>
<td>CIS451</td>
<td>DD</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>class</th>
<th>text</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIS451</td>
<td>DB</td>
</tr>
<tr>
<td>CIS451</td>
<td>PHP</td>
</tr>
<tr>
<td>MTH231</td>
<td>Disc</td>
</tr>
</tbody>
</table>