Schema Refinement & Normal Forms

CSC 4480

Review

• 1NF = relation
• 3NF: a table is in 3NF if the only determinants of nonkey columns are candidate keys.
• Often, a table in 3NF is already in 5NF.
• But not always.
• We examine advanced forms: BCNF, 4NF, 5NF.

Redundant Info

• The basic problem is the redundant storage of information.
• Anomalies:
  – insertion
  – update
  – deletion
• Basic idea: replace “big” relation with several smaller ones. The “trick” is to guarantee that the new relations are equivalent to the original one.

Functional Dependencies

• the FD X→Y hold if, ∀ t1, t2 in r:
  If t1.X=t2.X then t1.Y=t2.Y
• Closure of a set of F is F*.
• Armstrong’s axioms:
  – Reflexivity
  – Augmentation
  – Transitivity
• If we take the original FDs and then apply the axioms, we obtain the closure. (Theorem 1).

Normal Forms

• 1NF: a relation
• 3NF
• BCNF
• 4NF
• 5NF

Boyce-Codd Normal Form

• For every FD X→A in F, one of following is true:
  – A ∈X (trivial)
  – X is a superkey
• Each attribute depends on the key, the whole key, and nothing but the key.
Third Normal Form

- For every FD $X \rightarrow A$ in $F$, one of following is true:
  - $A \in X$ (trivial)
  - $X$ is a superkey
  - $A$ is part of some key for $R$.

Violations of 3NF

- $X$ is a proper subset of some key $K$. Partial dependency.
- ex: registration: student_id, course_id, student_name
- $X$ is not a proper subset of any key. Transitive dependency.
- ex: tournament winners: tournament, year, winner, winner_country

3NF vs BCNF

- For a table to be in 3NF but not BCNF, we must have overlapping candidate keys.
- Example: branch_customer_relationship(customer_no, branch_no, visiting_frequency, salesperson_no)
- Rule that each branch will serve a customer through a single salesperson. Salesperson works for only one branch.
- Dependency between salesperson_no and branch_no
- Alternate candidate key: customer_no, salesperson_no
- Customer_salesperson_relationship(customer_no, salesperson_no, visiting_frequency)
  - Salesperson(salesperson_no, branch)
  - Trade: no longer enforce unique branch-customer-salesperson.
  - Must enforce in program logic.

(Desirable) Properties of Decompositions

- Lossless-Join: we can recover the original relation from the decomposed relations.
- Dependency-Preserving: we can enforce each original FD by examining a single decomposed table.

Decomposition into BCNF

- see algorithm top of p. 623.

- BCNF is more normalized than 3NF, but, we can’t guarantee that we can generate a BCNF decomposition that is also dependency-preserving. But we can for 3NF.

Decomposition into 3NF

- The previous algorithm would generate 3NF. But we already said it is not DP.
- We will need to modify the algorithm.
- This requires the concept of “minimal cover for a set of FDs”.
Minimal Cover

- Formal definition p. 625.
- A minimal cover for a set F of FDs is a set of dependencies that is minimal in two respects:
  - Every dependency is as small as possible; each attribute on the left side is necessary, and the right side is a single attribute.
  - Every dependency in it is required for the closure to equal $F^+$.
- An algorithm for obtaining a minimal cover is on p. 626, followed by algorithms for 3NF.

4NF, 5NF

- In most cases, a relation in BCNF is also in 5NF.
- 4NF and 5NF problems usually occur in tables where every column is part of the key. (Key-only tables)
- 5NF violations in non-all-key tables, but, according to Simsion & Witt,
  - examples are unrealistic.
  - have yet to encounter an example in practice.

Fourth Normal Form

- The definition of 4NF relies on the notion of a multivalued dependency. A table with a multivalued dependency is one where the existence of two or more independent many-to-many relationships in a table causes redundancy; and it is this redundancy which is removed by fourth normal form.

4NF, 5NF example

- Financial market dealers
  - each instrument (stocks, bonds, etc.) can be traded only at a specified set of locations.
  - Each dealer is allowed to trade in a specified set of instruments.

example

<table>
<thead>
<tr>
<th>DealerID</th>
<th>InstrumentID</th>
<th>InstrumentID</th>
<th>LocationID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>Ordinary Stocks</td>
<td>Gov. Bonds</td>
<td>New York</td>
</tr>
<tr>
<td>Bruce</td>
<td>Futures</td>
<td>Gov. Bonds</td>
<td>Sydney</td>
</tr>
<tr>
<td>Bruce</td>
<td>Gov. Bonds</td>
<td>Futures</td>
<td>Singapore</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Futures</td>
<td>Tokyo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Options</td>
<td>Tokyo</td>
</tr>
</tbody>
</table>

note that both tables are all-key.

example

- We can derive a list of all authorized combinations by joining two tables:

<table>
<thead>
<tr>
<th>Dealer</th>
<th>Instrument</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>Gov. Bonds</td>
<td>New York</td>
</tr>
<tr>
<td>Smith</td>
<td>Gov. Bonds</td>
<td>London</td>
</tr>
<tr>
<td>Smith</td>
<td>Gov. Bonds</td>
<td>Sydney</td>
</tr>
<tr>
<td>Bruce</td>
<td>Futures</td>
<td>Singapore</td>
</tr>
<tr>
<td>Bruce</td>
<td>Futures</td>
<td>Tokyo</td>
</tr>
<tr>
<td>Bruce</td>
<td>Gov. Bonds</td>
<td>New York</td>
</tr>
<tr>
<td>Bruce</td>
<td>Gov. Bonds</td>
<td>London</td>
</tr>
<tr>
<td>Bruce</td>
<td>Gov. Bonds</td>
<td>Sydney</td>
</tr>
</tbody>
</table>
example

- But what if we tried to use the derived table as the solution?
- You should be able to see redundancy problems, even though the table is in BCNF.
- The problem is a result of a MVD.
- We tried to resolve two many-to-many relationships with a single table!
- Don’t do that.

Fifth Normal Form

- Stop splitting when:
  - any further splitting would violate lossless join
  - only splits left are trivial (i.e. same key in both tables)
- More complex version of example: additional rule: each dealer can only operate at a specified set of locations.
- Can we resolve all three relationships with one entity?

ER diagram – new example

Example – individual tables

<table>
<thead>
<tr>
<th>Dealer</th>
<th>Location</th>
<th>Instrument</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>Sydney</td>
<td>90-Day Bill</td>
</tr>
<tr>
<td>Smith</td>
<td>Tokyo</td>
<td>180-Day Bill</td>
</tr>
<tr>
<td>Philip</td>
<td>Sydney</td>
<td>180-Day Bill</td>
</tr>
<tr>
<td>Philip</td>
<td>Perth</td>
<td>180-Day Bill</td>
</tr>
</tbody>
</table>

Example – Combined Table

<table>
<thead>
<tr>
<th>Dealer</th>
<th>Location</th>
<th>Instrument</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>Sydney</td>
<td>90-Day Bill</td>
</tr>
<tr>
<td>Smith</td>
<td>Sydney</td>
<td>180-Day Bill</td>
</tr>
<tr>
<td>Smith</td>
<td>Tokyo</td>
<td>90-Day Bill</td>
</tr>
<tr>
<td>Smith</td>
<td>Tokyo</td>
<td>10-Year Bond</td>
</tr>
<tr>
<td>Philip</td>
<td>Sydney</td>
<td>180-Day Bill</td>
</tr>
<tr>
<td>Philip</td>
<td>Perth</td>
<td>180-Day Bill</td>
</tr>
</tbody>
</table>

Example - continued

- Some three-way relationships are valid. Problem only when they are derivable from simpler relationships.
- Suppose authorities decided on case-by-case basis, independent of any rules.
- Any set of two-column tables will either fail to cover some permitted combinations, or will generate combinations that are not permitted.
Example - continued

<table>
<thead>
<tr>
<th>Dealer</th>
<th>Location</th>
<th>Instrument</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>Sydney</td>
<td>90-Day Bill</td>
</tr>
<tr>
<td>Smith</td>
<td>Tokyo</td>
<td>90-Day Bill</td>
</tr>
<tr>
<td>Smith</td>
<td>Tokyo</td>
<td>180-Day Bill</td>
</tr>
<tr>
<td>Philip</td>
<td>Sydney</td>
<td>180-Day Bill</td>
</tr>
</tbody>
</table>

- Smith can deal in Sydney.
- Smith can deal in 180-day bills.
- 180-day bills can be traded in Sydney.
- But, Smith cannot deal in 180-day bills in Sydney.