Basics of Sound Structure

CSC 4480
(from Data Modeling Essentials, 3rd ed.)

Why is the Data Model Important?

• Leverage: a small change in the data model may have a major impact on the whole system.
• Usually, the programs that interact with the db are much more complex than the db itself.
• Their structure is heavily influenced by the db design.

Why is the Data Model Important?

• It can be expensive to change the data model.
• Let’s say our repair db system only allows one phone number per customer.
• Now, the shop owner says he needs to record multiple numbers per customer.
• The db change is easy.
• However, significant changes to application code.
  – report format; screen display; loops added; etc.

What makes a good data model?

• Completeness
• Nonredundancy
• Enforcement of Business Rules
• Data Reusability
• Stability and Flexibility
• Elegance
• Communication

Relation

• Relations can be represented in the form a 2D table.
• Synonyms:
  – relation, table
  – attribute, field, column
  – tuple, record, row

Student table

<table>
<thead>
<tr>
<th>Name</th>
<th>ID</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>123</td>
<td>Bryn Mawr</td>
</tr>
<tr>
<td>Jones</td>
<td>456</td>
<td>Pittsburgh</td>
</tr>
</tbody>
</table>

Relation - definition

• Each cell contains a single value.
• All entries in a column are the same type.
• Each column has a unique name.
• The order of the columns doesn’t matter.
• The order of the rows doesn’t matter.
• No two rows may be identical.
Functional Dependency

• relationships between or among attributes.
• SSN → name
• dob → age
• (unit price, quantity) → cost
• classroom → course capacity

Keys

• A key is a group of one or more attributes that uniquely defines a tuple in a relation.
• The primary key is a minimal set of columns that contains a different combination of values for each row of the table.
• Each table must have a primary key!
• Purpose of pk: to refer unambiguously to a single row.
• Example question to business person:
  – If I nominate account number as pk, can you guarantee that there will never be more than one account with that number?

Normal Forms

Informal Example of Normalization

• Normalization is essentially a two-step process.
  1. Put the data into tabular form (by removing repeating groups)
  2. Remove duplicated data to separate tables.

Repeating Groups

• For the auto-repair example:
  REPAIR(customer_name, date, customer_phone, make, model, technician, tech_rate, part1, q1, cost1, part2, q2, cost2, ..., part10, q10, cost10)
• What might be a good pk?
  REPAIR(customer_name, date, customer_phone, make, model, technician, tech_rate, part1, q1, cost1, part2, q2, cost2, ..., part10, q10, cost10)

Repeating Groups

• Can you see any problems so far?
• We are in trouble if we ever need more than ten parts.
• And, we are wasting space if we only need two parts.
• Solution: move the repeating groups to a separate table.
Repeating Groups

REPAIR(customer_name, date, customer_phone, make, model, technician, tech_rate)
PART(customer_name, date, part, quantity, cost)

- This is now in First Normal Form (1NF).
- 1NF = definition of relation

Problems with 1NF

REPAIR(customer_name, date, customer_phone, make, model, technician, tech_rate)
PART(customer_name, date, part, quantity, cost)

- Notice that:
  - every row with customer "Jones" also contains his phone number.
  - every row with technician "Albert" also contains his labor rate.

Problems with 1NF

- What if we wanted to change Albert’s labor rate?
  We would need to change all rows containing Albert. Same thing with customer phone number. Change Anomaly.
- What if we deleted the last repair for customer "Smith"? We lose the fact that his phone number is xxx-xxxx. Deletion Anomaly.
- What if we hire a new technician? We can’t add his labor rate until he completes a repair. Insertion Anomaly.

Eliminate Redundancy

- Move technician info and customer info to separate tables.
- Use tech name and customer name as keys.

REPAIR(customer_name, date, make, model, technician)
PART(customer_name, date, part, quantity, cost)
CUSTOMER(customer_name, customer_phone)
TECHNICIAN(technician_name, tech_rate)

Back to FDs (or determinants)

- customer name → customer phone number
- technician name → tech labor rate
- Procedure:
  1. Identify all determinants, other than PK, and the columns they determine.
  2. Create a separate table for each determinant, and the columns it determines. The determinant becomes the key of the new table.
  3. Name the new tables.
  4. Remove the determined columns from the original table. Leave the determinants to provide links between the tables.

3rd normal form

- We are now in 3NF.
- What happened to 2NF?
- Is 3NF the same as “fully normalized”?
  - No, but most of the time tables in 3NF are also in 5NF.
- Performance…will all those tables slow things down?
Definitions

• Primary Key: column or combination of columns that has a different value for every row in the table.
• Candidate Key: occasionally, more than one choice for PK. The set of possible PKs are candidate keys.
• Requires slight change to step 1:
  1. Identify all determinants, other than CKs, ...

Definition of 3NF

• Nonkey column: a column that is not part of the primary key.
• A table is in 3NF if the only determinants of nonkey columns are candidate keys.

Foreign Key

• When we moved repeating groups to a new table, we included the primary key of the original table.
• This is the only way to link the two tables together.
• These columns are called foreign keys.
• For example, customer name is a PK in the customer table, but a FK in the repair table.
• Referential Integrity. Every customer name in the repair table must have a corresponding entry in the customer table. Most modern DBMSs provide referential integrity features to prevent bogus FKs from being used.