1.264 Lecture 10

Data normalization

Next class: Read Murach chapters 1-3. Exercises due after class
Make sure you've downloaded and run the .sql file to create the database we'll be using in the next two classes before the next class.
Remember to register SQL Server if you didn't when you installed it.

Normalization

- Normalization rules
  - Prevent update anomalies (mistakes) and data inconsistencies
  - Degrade performance, usually only slightly
    - More impact on reads, where several rows vs one are read
    - Little impact on writes, which tend to be the bottleneck anyway
  - Denormalization is common on read-only databases and in report generation or data warehouses.
    - You can't have update anomalies if you don't do updates!
  - Your homework 4 initial data is not normalized.
    - Homework 4 and 5 require you to normalize your data, for correctness
  - Building the data model is done collaboratively with many meetings and discussions; it sets the business rules
  - Normalizing the data model is a technical exercise, done in a back room; it does not change the business rules
    - Though it may raise questions that refine the rules

Five normal forms

- 1: All occurrences of an entity must contain the same number of attributes.
  - No lists, no repeated attributes.
- 2: All non-key fields must be a function of the key.
- 3: All non-key fields must not be a function of other non-key fields.
- 4: A row must not contain two or more independent multi-valued facts about an entity.
- 5: A record cannot be reconstructed from several smaller record types.

Examples based on William Kent, "A Simple Guide to Five Normal Forms in Relational Database Theory", Communications of the ACM 26(2), Feb. 1983

Definitions

- Row or record: a fixed tuple (set) of attributes (fields) that describes an instance of an entity
- Key: a unique identifier for a row in a table, used to select the row in queries. It can be composed of several fields. Primary key.
- Non-key: all the other fields in the row, including the foreign key fields
- Entity: object defined in system model about which data is stored in the database. A table in a relational database.

First normal form

- All rows must be fixed length
  - Restrictive assumption, not a design principle.
  - Does not allow variable length lists.
  - Also does not allow repeated fields, e.g., vehicle1, vehicle2, vehicle3…
    - However many columns you allow, you will always need one more…
    - Use a many-many relationship instead, always. See our vehicle-driver or vehicle-specialist examples from the previous lecture.

Second normal form

Part	Warehouse	Quantity	WarehouseAddress
42	Boston	2000	24 Main St
333	Boston	1000	24 Main St
390	New York	3000	99 Broad St

- All non-key fields must be a function of the full key
  - Example that violates second normal form:
    - Key is Part + Warehouse
    - Someone found it convenient to add Address, to make a report easier
    - WarehouseAddress is a fact about Warehouse, not about Part
  - Problems:
    - Warehouse address is repeated in every row that refers to a part stored in a warehouse
    - If warehouse address changes, every row referring to a part stored in that warehouse must be updated
    - Data might become inconsistent, with different records showing different addresses for the same warehouse
    - If at some time there were no parts stored in the warehouse, there may be no record in which to keep the warehouse's address.

Second normal form

- **Solution**
  - Two entity types: Inventory, and Warehouse
  - Advantage: solves problems from last slide
  - Disadvantage: If application needs address of each warehouse stocking a part, it must access two tables instead of one. This used to be a problem but rarely is now.

Part	Warehouse	Quantity
42	Boston	2000
333	Boston	1000
390	New York	3000

Warehouse	WarehouseAddress
Boston	24 Main St
New York	99 Broad St

Third normal form

Employee	Department	DepartmentLocation
234	Finance	Boston
223	Finance	Boston
399	Operations	Washington

- **Non-key fields cannot be a function of other non-key fields**
  - **Example that violates third normal form**
    - **Key is employee**
    - **Someone found it convenient to add department location for a report**
    - **Department location is a function of department, which is not a key**
  - **Problems:**
    - **Department location is repeated in every employee record**
    - **If department location changes, every record with it must be changed**
    - **Data might become inconsistent**
    - **If a department has no employees, there may be nowhere to store its location**

Third normal form

- **Solution**
  - **Two entity types: Employee and department**

Employee	Department
234	Finance
223	Finance
399	Operations

Department	DepartmentLocation
Finance	Boston
Operations	Washington

TV: "The truth, the whole truth, and nothing but the truth"
DB: "The key, the whole key, and nothing but the key"

Exercise

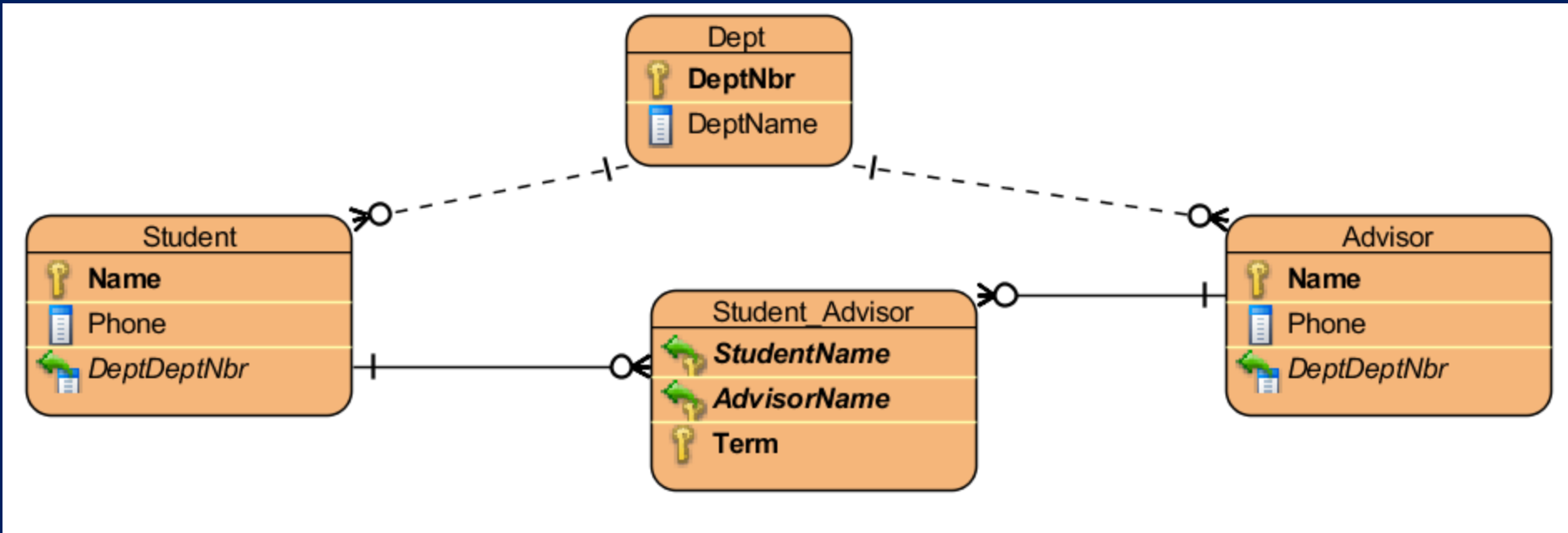
- **Put the following data into a model in third normal form. Ignore the 'term' field initially; handle it last.**

StudentName	StudentPhone	StudentDept	StudentDeptName	AdvisorName	AdvisorPhone	AdvisorDept	Term
Tolstoy	593-3824	21	English	Caplice	253-3233	ESD	Fall
Thoreau	644-2343	21	English	Caplice	253-3233	ESD	Fall
James	534-2534	21	English	Lapide	253-1111	ESD	Fall
Woolf	643-5436	18	Mathematics	Toomre	253-6322	Mathematics	Spring
Shakespeare	634-6344	8	Physics	Smith	253-8453	Physics	Spring
Pushkin	534-9832	7	Biology	Griffith	253-9833	Bio Engr	Spring

- **Use Visual Paradigm to draw the data model**
  - **Draw entities and relationships**
  - **Set primary, foreign keys**
  - **What ambiguities are there? How do you correct them?**
  - **How do you handle the term field? What does it mean?**

- **This is what you'll be doing in much of homework 4**

Solution



Ambiguities:

1. **Advisor department name/ID**
2. **Term could be a domain entity**
3. **Name not a good primary key, may have duplicates**
4. **Advisors, students could be in multiple departments**
5. **Student might have one (or more?) advisors in a term (?)**

Fourth normal form

Employee	Skill	Language
Brown	cook	English
Smith	type	German

- A row should not contain two or more independent multi-valued facts about an entity.
  - Example that violates fourth normal form:
    - An employee may have several skills and languages
  - Problems
    - Uncertainty in how to maintain the rows. Several approaches are possible and different consultants, analysts or programmers may (will) take different approaches, as shown on next slide

Fourth normal form problems

Brown	cook	
Brown	type	
Brown		French
Brown		German
Brown		Greek
Smith	cook	
etc		

- **Disjoint format. Effectively same as 2 entity types.**
  - **Blank fields ambiguous. Blank skill could mean:**
    - **Person has no skill**
    - **Attribute doesn't apply to this employee**
    - **Data is unknown**
    - **Data may be found in another record (as in this case)**
  - **Programmers will use all these assumptions over time, as will data entry staff, analysts, consultants and users**

Fourth normal form problems, cont.

Employee	Skill	Language
Brown	cook	French
Brown	cook	German
Brown	cook	Greek
Brown	type	French
Brown	type	German
Brown	type	Greek

(Smith similar)

- **Cross product format. Problems:**
  - **Repetitions: updates must be done to multiple records and there can be inconsistencies**
  - **Insertion of a new skill may involve looking for a record with a blank skill, inserting a new record with possibly a blank language or skill, or inserting a new record pairing the skill with some or all of the languages.**
  - **Deletion is worse: It means blanking a skill in one or more records, and then checking you don't have 2 records with the same language and no skill, or it may mean deleting one or more records, making sure you don't delete the last mention of a language that should not be deleted**

Fourth normal form solution

- **Solution: Two entity types**
  - **Employee-skill and employee-language**

Employee	Skill
Brown	cook
Brown	type
Smith	cook

Employee	Language
Smith	French
Smith	German
Smith	Greek
Brown	French

- **Note that skills and languages may be related, in which case the starting example was ok:**
  - **If Smith can only cook French food, and can type in French and Greek, then skill and language are not multiple independent facts about the employee, and we have not violated fourth normal form.**
- **Examples you're likely to see:**
  - **Person on 2 projects, in 2 departments**
  - **Part from 2 vendors, used in 4 assemblies**

Fifth normal form

- A record cannot be reconstructed from several smaller record types.
- Example:
  - Agents represent companies
  - Companies make products
  - Agents sell products
- Most general case (allows any combination):

Agent	Company	Product
Smith	Ford	car
Smith	GM	truck

  - Smith does not sell Ford trucks nor GM cars
  - If these are the business rules, a single entity is fine
  - But…

Fifth normal form

- In most real cases a problem occurs
  - If an agent sells a certain product and she represents the company, then she sells that product for that company.

Agent	Company	Product
Smith	Ford	car
Smith	Ford	truck
Smith	GM	car
Smith	GM	truck
Jones	Ford	car

(Repetition of facts)

  - We can reconstruct all true facts from 3 tables instead of the single table:

Agent	Company
Smith	Ford
Smith	GM
Jones	Ford

Agent	Product
Smith	car
Smith	truck
Jones	car

Company	Product
Ford	car
Ford	truck
GM	car
GM	truck

(No repetition of facts)

Fifth normal form

- Problems with the single table form
  - Facts are recorded multiple times. E.g., the fact that Smith sells cars is recorded twice. If Smith stops selling cars, there are 2 rows to update and one will be missed.
  - Size of this table increases multiplicatively, while the normalized tables increase additively. With big operations, this is a big difference.
    - 100,000 x 100,000 is a lot bigger than 100,000 + 100,000
- It's much easier to write the business rules from 5th normal
  - Rules are more explicit
  - Supply chains usually have all sorts of 5th normal issues

Fifth normal form, concluded

- An example with a subtle set of conditions

Non-normal

Agent	Company	Product
Smith	Ford	car
Smith	Ford	truck
Smith	GM	car
Smith	GM	truck
Jones	Ford	car
Jones	Ford	truck
Brown	Ford	car
Brown	GM	car
Brown	Toyota	car
Brown	Toyota	bus

Can you quickly deduce the business rules from this table?

Fifth normal

Agent	Company
Smith	Ford
Smith	GM
Jones	Ford
Brown	Ford
Brown	GM
Brown	Toyota

Company	Product
Ford	car
Ford	truck
GM	car
GM	truck
Toyota	car
Toyota	bus

Agent	Product
Smith	car
Smith	truck
Jones	car
Jones	truck
Brown	car
Brown	bus

- Jones sells cars and GM makes cars, but Jones does not represent GM
- Brown represents Ford and Ford makes trucks, but Brown does not sell trucks
- Brown represents Ford and Brown sells buses, but Ford does not make buses

Morals of this story

- Systems are ephemeral. Data is permanent
- If you mess up a system, you rewrite it and it's fixed
- If you mess up the data, it's usually irretrievable
  - 90% error rate in a circuit design database I worked with
  - Allowed null foreign keys, so they were never filled in.
  - Design not normalized, so same fact stored in multiple places
  - Errors in data model, which didn't match business rules
- Real business have subtle business rules
  - Care in data modeling and business rules is needed to achieve good data quality
    - This is an interactive process, done with lots of people
  - Care in data normalization is needed to preserve data quality
    - Normalization ensures that each fact is stored in one and only one place (with rare exceptions). If a fact is stored in two or more places, they can and will become inconsistent, and then you won't know the fact at all.
    - This is a technical process, done with just a few technical people

1.264J / ESD.264J Database, Internet, and Systems Integration Technologies
Fall 2013