Database Normalization

Introduction to normalization

Motivating questions

- What is database normalization?
- Why should we normalize our data models?
- How do we normalize a relational data model?
- What are the potential drawbacks of normalization?

Introduction to normalization

- Raw data may be in stored in a single table, containing redundant information or information about several different concepts

- These data can be separated into tables and normalized

- Updating the data model is done collaboratively during many meetings and discussions; it sets the business rules

Why normalize relational models?

- Normalization helps prevent:
  - Redundancy
  - Confusion
  - Improper keys
  - Wasted storage
  - Incorrect or outdated data

- During updates, normalization prevents mistakes and data inconsistencies

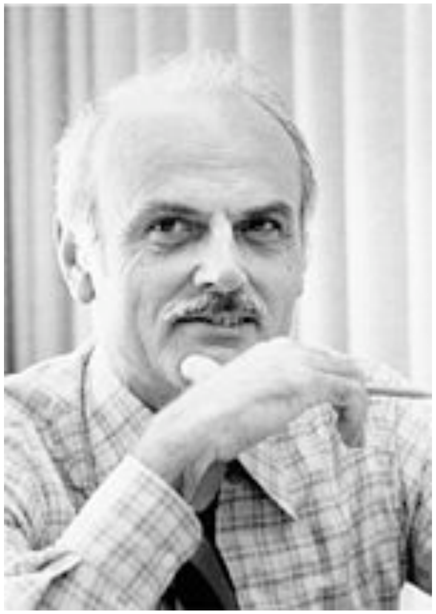
Objectives of normalization

1. To free the collection of [tables] from undesirable insertion, update and deletion dependencies;

2. To reduce the need for restructuring the collection of [tables], as new types of data are introduced, and thus increase the life span of application programs;

3. To make the relational model more informative to users;

4. To make the collection of [tables] neutral to the query statistics, where these statistics are liable to change as time goes by.



Edgar F. Codd: Inventor of the relational model

Review of relational model definitions

Entity: Departments

Table: A collection of records about the entity (departments)

Record: Information about department 372

Entry: Value of DeptNbr for the construction department

Attribute: DeptName – the names of the departments

Departments

DeptNbr	DeptName	DeptType	DeptStatus
930	Receiving	Mfg	Active
378	Assembly	Mfg	Active
372	Finance	Adm	Active
923	Planning	Adm	Active
483	Construction	Plant	Inactive

Primary key: Attribute which uniquely identifies each record in a table

Database: CompanyDatabase, includes tables such as: Departments, Employees, Sales

Summary of the five normal forms

1. All rows in a table must contain the same number of attributes; no sub-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.

Key points from lesson

- Normalization helps improve consistency and reliability of a database by avoiding redundant data
- Many real-life datasets are not normalized, which could lead to trouble if used directly in a database
- Large organizations may have many different users in a single database, making consistency essential

First normal form

First normal form

- All rows in a table must contain the same number of attributes; no sub-lists, no repeated attributes

- Customer table supports a single telephone number per individual, but what if we want to store another number?

Customer ID	First Name	Surname	Telephone Number
123	Robert	Ingram	555-861-2025
456	Jane	Wright	555-403-1659
789	Maria	Fernandez	555-808-9633

First normal form
Adding a second phone number

- Stuff it into the same field and just make that field longer?

Customer ID	First Name	Surname	Telephone Number
123	Robert	Ingram	555-861-2025
456	Jane	Wright	555-403-1659 555-776-4100
789	Maria	Fernandez	555-808-9633

- Add a second field for a telephone number?

Customer ID	First Name	Surname	Telephone1	Telephone2
123	Robert	Ingram	555-861-2025	
456	Jane	Wright	555-403-1659	555-776-4100
789	Maria	Fernandez	555-808-9633	

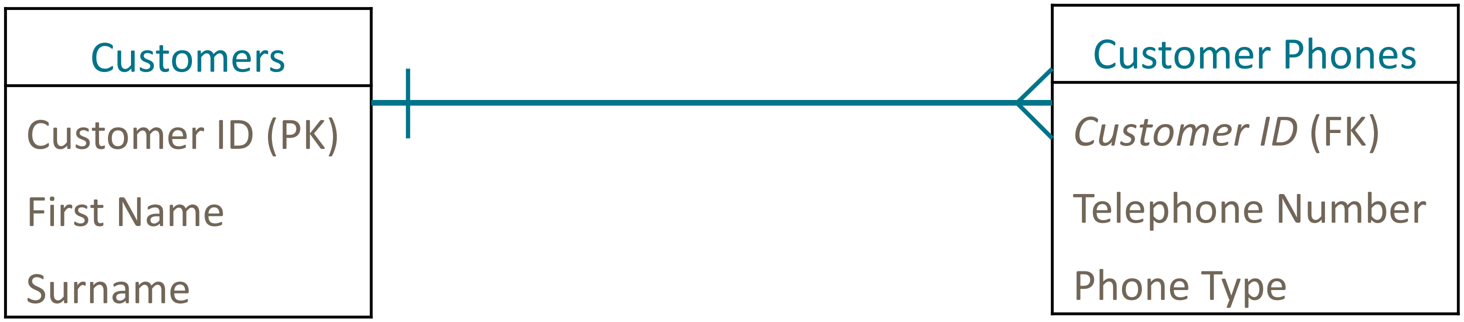
First normal form
Adding a second phone number

- List one telephone number per row?

Customer ID	First Name	Surname	Telephone Number
123	Robert	Ingram	555-861-2025
456	Jane	Wright	555-403-1659
456	Jane	Wright	555-776-4100
789	Maria	Fernandez	555-808-9633

First normal form
Solution

- How many entities are there?
  - (1) Customers; (2) Customer Phone Numbers
- Split these distinct entities into two tables:



Customers

Customer ID (PK)

First Name

Surname

Customer Phones

Customer ID (FK)

Telephone Number

Phone Type

Customer ID	First Name	Surname
123	Robert	Ingram
456	Jane	Wright
789	Maria	Fernandez

Customer ID	Telephone Number	Phone Type
123	555-861-2025	Mobile
456	555-403-1659	Mobile
456	555-776-4100	Home
789	555-808-9633	Mobile

Key points from lesson

- Put only one observation of data in any database entry
- Avoid creating numbered lists in tables
- A normalized solution may result in more tables
- The first normal form can make databases robust to change and easier to use in large organizations

Second and third normal forms

Second normal form

- Must first be in first normal form
- All non-key fields must be a function of the primary key; only store facts directly related to the primary key in each row

- A user found it convenient to add Warehouse Address to the Parts table, to make report creation easier

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

Second normal form
Adding Warehouse Address to Parts

- The primary key is Part, and Warehouse Address is an attribute unrelated to Parts

- Warehouse address is repeated in every row that refers to a part stored in a warehouse
  - What if warehouse address changes?
  - What if at some time there were no parts stored in the warehouse?

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

Second normal form
Solution

- How many entities are there?
  - (1) Parts; (2) Warehouses

- Advantage: satisfies second normal form; solves problems from last slide
- Disadvantage: if report needs address of each a warehouse stocking a part, it must access two tables instead of one

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

Warehouse	Warehouse Address
Boston	24 Main St
New York	99 Broad St

Third normal form

- Must first be in second normal form
- Non-key fields cannot be a function of other non-key fields

- A user found it convenient to add department location to the employees table, to make report creation easier

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

Third normal form
Adding Dept. Location to Employees

- Department location is a function of department, which is not the primary key of the employees table

- Department location is repeated in every employee record
  - What if the department location changes?
  - What if at some time a department had no employees?

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

Third normal form
Solution

- How many entities are there?
  - (1) Employees; (2) Departments

Employee	Department
234	Finance
223	Finance
399	Operations

Department	DepartmentLocation
Finance	Boston
Operations	Washington

Key points from lesson

- Be careful to only store attributes in an entity if they are directly related to that entity
- Even if would be useful to have certain information stored in the same table, attributes should only be directly related to the key and entity in the table where they are stored
- The duplication of data can result in erroneous or lost data

Fourth normal form

Fourth normal form

- Must first be in third normal form
- A row should not contain two or more independent, multi-valued facts about an entity

- A user found it convenient to add language to a table about employees and their skills

Employee	Skill	Language
Brown	cook	English
Smith	type	German

Fourth normal form
Storing languages with skills

- An employee may have several skills and languages

- There is uncertainty in how to maintain the rows
  - What happens when one or more attributes has one or more values? How should these be stored?

- What about the disjoint format?

Employee	Skill	Language
Brown	cook	English
Smith	type	German

Employee	Skill	Language
Brown	cook	
Brown	type	
Brown		French
Brown		English
Brown		German
Smith	cook	
Etc…		

Fourth normal form
Storing languages with skills

- Disjoint format

- What do the empty entries mean?
  - Person has no skill/language
  - Attribute doesn't apply to the particular employee
  - Value is unknown
  - Data may be in another record

Employee	Skill	Language
Brown	cook	English
Smith	type	German

Employee	Skill	Language
Brown	cook	
Brown	type	
Brown		French
Brown		English
Brown		German
Smith	cook	

Fourth normal form
Storing languages with skills

- Cross product format

- What if the value of entry has to be updated?
  - Updates must be done to multiple records and there can be inconsistencies

Employee	Skill	Language
Brown	cook	English
Smith	type	German

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

Fourth normal form
Storing languages with skills

- What if a new skill has to be added?
  - Should it be inserted in records where skill is empty?
  - Should records be added with an empty language?
  - Or should a new record be added with some or all languages?

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

Employee	Skill	Language
Brown	cook	English
Smith	type	German

Fourth normal form
Storing languages with skills

- What if a skill has to be deleted?
  - Delete the skill from all relevant records
    - Are there multiple records with the same language and no skill?
    - Should the record with that skill be deleted?
    - What if the record with that skill is the last mention of a language?

Employee	Skill	Language
Brown	cook	English
Smith	type	German

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

Fourth normal form
Solution

- How many entities are there?
  - (1) Employee Skills; (2) Employee Languages

Employee	Skill
Brown	cook
Brown	type
Smith	type

Employee	Language
Smith	English
Smith	German
Smith	Greek
Brown	English

Fourth normal form
Additional notes

- Note that skills and languages could be related
  - If Smith can cook Greek food and can type in German, 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:
  - Employee on two projects, in two departments
  - Part from two vendors, used in four assemblies

Employee	Skill	Language
Brown	cook	English
Brown	type	English
Smith	cook	Greek
Smith	type	German

Key points from lesson

- Unrelated or independent facts about an entity should be stored in separate tables

- Typically, create a one-to-many relationship between the entity and the list of choices for each fact

- When two facts about an entity are related, store them in a single table

Fifth normal form

Fifth normal form

- Must first be in fourth normal form
- A record cannot be reconstructed from several smaller record types

- Agents represent companies, companies make products, agents sell products

- A user found it convenient to store information about the agents, the companies and the related products together

Agent	Company	Product
Smith	Ford	car
Smith	GM	truck

Fifth normal form
Storing company with product

- In the most general case, this table should allow for any combination of agent, company and product

- If the business rules are that Smith does not sell Ford trucks nor GM cars, then this single entity is OK

Agent	Company	Product
Smith	Ford	car
Smith	GM	truck

Fifth normal form
Storing company with product

- What if Smith stops selling cars?
  - More than one record needs to be updated for this fact

- What are the business rules?

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

Fifth normal form
Storing company with product

- 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

- 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

Fifth normal form
Additional considerations

- Size of this single table increases multiplicatively, while the normalized tables increase additively

- Much easier to write the business rules from the three tables in the fifth normal form, rules are more explicit

- Supply chains tend to have fifth normal form issues

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

Fifth normal form
More subtle business rules

- Can you deduce the business rules from this table?

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

Fifth normal form
What are the business rules now?

- 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

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

Fifth normal form
What are the business rules now?

- 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
- Fifth normal form:

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

Normalization
Implementation details

- Degrades performance, usually only slightly
  - Greater impact on reads, several records now required
  - Less of an impact on writes

- Large, read-only databases for report generation or data warehouses may not be normalized

- Normalizing the data model is a technical exercise
  - It does not change the business rules, but may help refine them through review

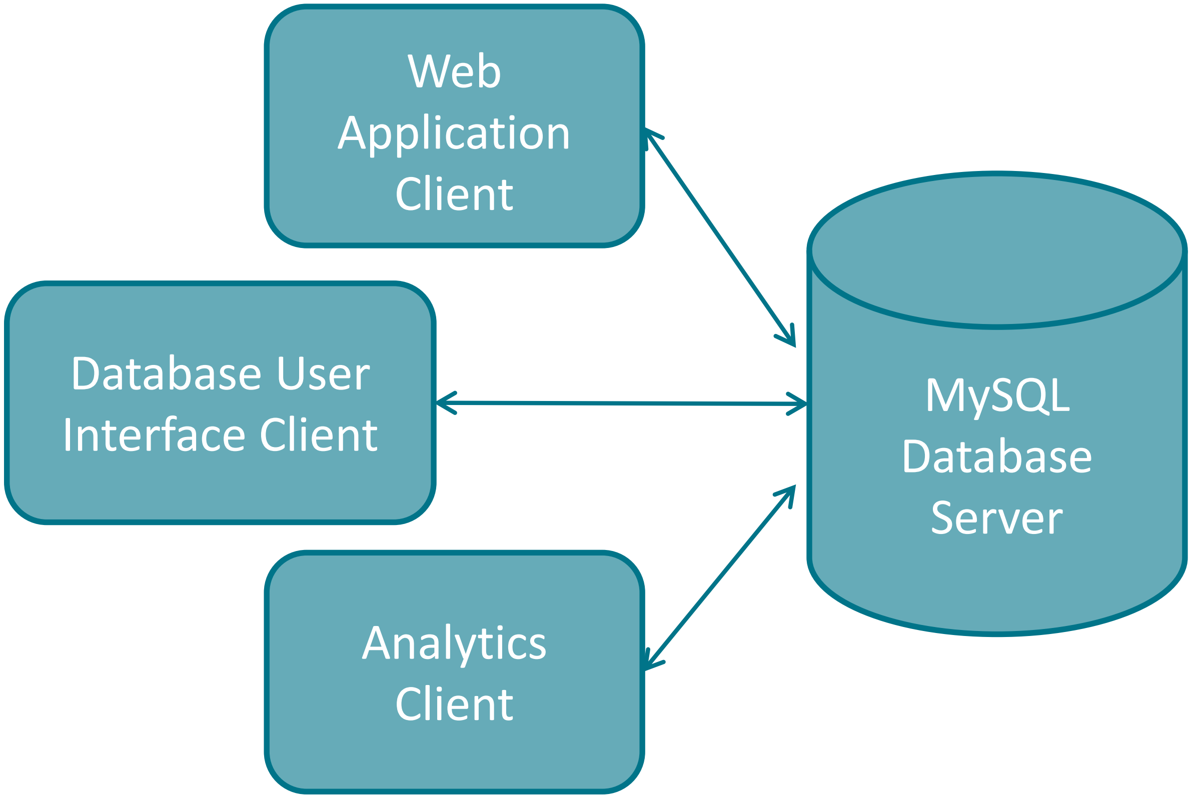
Key points from lesson

- Systems are ephemeral, data is permanent
- Real businesses may have subtle business rules
- Care in data modeling and business rules is needed to achieve good data quality
- Care in data normalization is needed to preserve data quality
- Normalization ensures that each fact is stored in one and only one place, to ensure data remain consistent

Client-server architecture

Client-server model

- Clients can connect to servers to access a specific service using a standardized protocol



Database servers

- Databases are hosted on a server

- Database servers are not usually accessible through a file system or directory structure

- Companies typically host servers centrally or on the cloud

Database clients

- The client has software which allows it to connect and communicate with the database server using a standardized protocol
- Client software is usually not very complicated or bloated
- There are client user interfaces for many databases

Remote hosting

- Databases may be accessed remotely, over the Internet

- They may be hosted:
  - On a single server
  - In a database cluster (a set of database servers that distribute tasks over multiple physical machines)
  - As a cloud service (virtualized database query service)

- These systems are designed to abstract the implementation details

Key points from lesson

- Databases are hosted on servers, which may be stored locally, over the internet, or on the cloud
- Many users can access a database, each with different goals and client software
- The database client-server model allows different clients to communicate with the database server using a set of standard protocols

Sources and Image Information

- Normalization examples
  - Examples based on William Kent, "A Simple Guide to Five Normal Forms in Relational Database Theory", Communications of the ACM 26(2), Feb. 1983
  - First normal form example – wikipedia
  - Picture of Edgar Codd: https://en.wikipedia.org/wiki/Edgar_F._Codd