Database Design

Dr. Evren Eryilmaz
National University
12/20/2012

Presentation Outline

- Database Fundamentals
- Relational Model
- Modification Anomalies
- Normal Forms
- Forming Relationships
- Types of Databases
- Denormalization

Database Fundamentals

- Data: Meaningful facts
- Database: An organized collection of logically related data. Databases arise from:
  1. Existing data
  2. Development of new information systems projects
  3. The need to redesign an existing database to adopt to changing requirements

Database Fundamentals

- Database Management System (DBMS): Software package for defining and managing a database.
  - Microsoft Access
  - Oracle
  - Microsoft SQL Server
  - MySQL

Relational Model

- Was introduced in 1970 by Dr. E. F. Codd in his paper "A Relational Model of Data for Large Shared Data Banks" in 1970.”
- Why study the relational model?
- A relational data model is based on a collection of tables, and the model has three components
  - Data Structures: Data are organised in the form of tables
  - Data Manipulation: SQL language
  - Data Integrity: Specify business rules

Relational Model Terminology

- A relation is a table with the following characteristics:
  - Rows contain data about an entity
  - Columns contain data about attributes of the entities
  - All entries in a column are of the same kind
  - Each column has a unique name
  - No two rows may be identical
  - Each cell holds a single value
  - The order of the columns is irrelevant
  - The order of the rows is irrelevant.

Tables That Are Not Relations

a) Table with Multiple Entries per Cell

EMPLOYEENUMBER	FIRSTNAME	LASTNAME	DEPARTMENT	PHONE
100	Jerry	Johnson	Accounting	444-8898
200	Mary	Abernathy	Finance	236-0991 555-7171 444-9980

b) Entries in a column are not of the same kind

EMPLOYEENUMBER	FIRSTNAME	LASTNAME	DEPARTMENT	PHONE
300	Tom	Caruthers	tc@somewhere.com	767-0900
400	Liz	Smathers	Home: Fax:	236-9987 236-0900

Relational Keys

- Relational modeling uses primary keys and foreign keys to maintain relationships
- Primary Key: A column (or combination of columns) that uniquely identifies each row in a table
- Foreign Key: A column (or combination of columns) that is the primary key of another table

Modification Anomalies

- Insertion Anomaly: Occurs when extra data beyond the desired data must be added to a table.
- Example of an Insert Anomaly

Student_ID	CLASS_ID	CLASS_TITLE	INSTRUCTOR
562321124	IST-189	BASIC COMPUTER PROCESSING	SAMUAL MOORE
562321124	IST-189	BASIC COMPUTER PROCESSING	ADRIAN JONES
769017832	IST-250	ADVANCED COMPUTER PROCESSING	JAMES WILSON
989832423	OST-223	DATA PROCESSING	SAMUEL MOORE
908769987	IST-289	PROGRAMMING METHODS	NORMAN YOUNG
179077643	IST-220	BASIC PROGRAMMING	TAMMY GREEN

Modification Anomalies

- Deletion Anomaly: Occurs when deleting a row causes other data to be deleted
- Example of a Deletion Anomaly

Student_ID	CLASS_ID	CLASS_TITLE	INSTRUCTOR
562321124	IST-189	BASIC COMPUTER PROCESSING	SAMUAL MOORE
562321124	IST-189	BASIC COMPUTER PROCESSING	ADRIAN JONES
769017832	IST-250	ADVANCED COMPUTER PROCESSING	JAMES WILSON
989832423	OST-223	DATA PROCESSING	SAMUEL MOORE
908769987	IST-289	PROGRAMMING METHODS	NORMAN YOUNG
179077643	IST-220	BASIC PROGRAMMING	TAMMY GREEN

Modification Anomalies

- Update Anomaly: Occurs when it is necessary to change multiple rows to modify only a single fact
- Example of an update anomaly

Student_ID	CLASS_ID	CLASS_TITLE	INSTRUCTOR
562321124	IST-189	BASIC COMPUTER PROCESSING	SAMUAL MORE
562321124	IST-189	BASIC COMPUTER PROCESSING	ADRIAN JONES
769017832	IST-250	ADVANCED COMPUTER PROCESSING	JAMES WILSON
989832423	OST-223	DATA PROCESSING	SAMUEL MORE
908769987	IST-289	PROGRAMMING METHODS	NORMAN YOUNG
179077643	IST-220	BASIC PROGRAMMING	TAMMY GREEN

Normal Forms

- The requirements to satisfy the 1st NF:
  - Each table has a primary key: minimal set of attributes which can uniquely identify a record
  - The values in each column of a table are atomic (no multi-value attributes are allowed)
  - There are no repeating groups

1st Normal Form Example

Un-normalized table:

MANAGER	**EMPLOYEES**
James	Jerry, Mary
Smith	Tom, Richard

Is it 1NF?

MANAGER	**EMPLOYEE1**	**EMPLOYEE2**
James	Jerry	Mary
Smith	Tom	Richard

1st Normal Form Example

- Normalized table

EMPLOYEE	**MANAGER**
Jerry	James
Mary	James
Tom	Smith
Richard	Smith

Moving to 2nd Normal Form

- The requirements to satisfy the 2nd NF:
  - All requirements for 1st NF must be met.
  - Redundant data across multiple rows of a table must be moved to a separate table.
    - The resulting tables must be related to each other by use of foreign key.

2nd Normal Form Example

CUSTOMERID	FIRSTNAME	LASTNAME	ADDRESS	CITY	STATE	ZIPCODE
1	Bob	Smith	123 Main St.	Tucson	AZ	12345
2	John	Brown	555 2nd Ave.	St. Paul	MN	54355
3	Sandy	Jessop	4256 James St.	Chicago	IL	43555
4	Gameil	Hintz	569 Summit St.	St. Paul	MN	54355
5	Shiela	Green	12 Michigan Ave.	Chicago	IL	43555

This table is in 1NF: all fields are atomic and the CustomerID serves as the primary key

2nd Normal Form Example

Customer Table

CUSTOMERID	FIRSTNAME	LASTNAME	ADDRESS	ZIPCODE
1	Bob	Smith	123 Main St.	12345
2	John	Brown	555 2nd Ave.	54355
3	Sandy	Jessop	4256 James St.	43555
4	Gameil	Hintz	569 Summit St.	54355
5	Shiela	Green	12 Michigan Ave.	43555

ZipCode Table

ZIPCODE	CITY	STATE
12345	Tucson	AZ
54355	St. Paul	MN
43555	Chicago	IL

Summary of 1NF and 2NF

- 1NF:
  - Each table has a primary key
  - The values in each column of a table are atomic
  - There are no repeating groups
- 2NF:
  - All requirements for 1NF must be met
  - Redundant data across multiple rows of a table must be moved to a separate table

Moving to 3rd Normal Form

- The requirements to satisfy the 3rd NF:
  - All requirements for 2nd NF must be met.
  - All columns in a table must depend only on the primary key and not on any other field in the table
    - Any field which is dependent not only on the primary key but also on another field is moved out to a separate table.

3rd Normal Form Example

ORDERID	CUSTOMERID	PRODUCTID	PRICE	QUANTITY	TOTAL
1	1001	AB-111	50	1000	50000
2	1002	AB-111	60	500	30000
3	1001	ZA-245	35	100	3500
4	1003	MB-153	82	25	2050
5	1004	ZA-245	42	10	420
6	1002	ZA-245	40	50	2000
7	1001	AB-111	75	100	7500

3rd Normal Form Example

Order Table

ORDERID	CUSTOMERID	PRODUCTID	QUANTITY
1	1001	AB-111	1000
2	1002	AB-111	500
3	1001	ZA-245	100
4	1003	MB-153	25
5	1004	ZA-245	10
6	1002	ZA-245	50
7	1001	AB-111	100

Pricing List Table

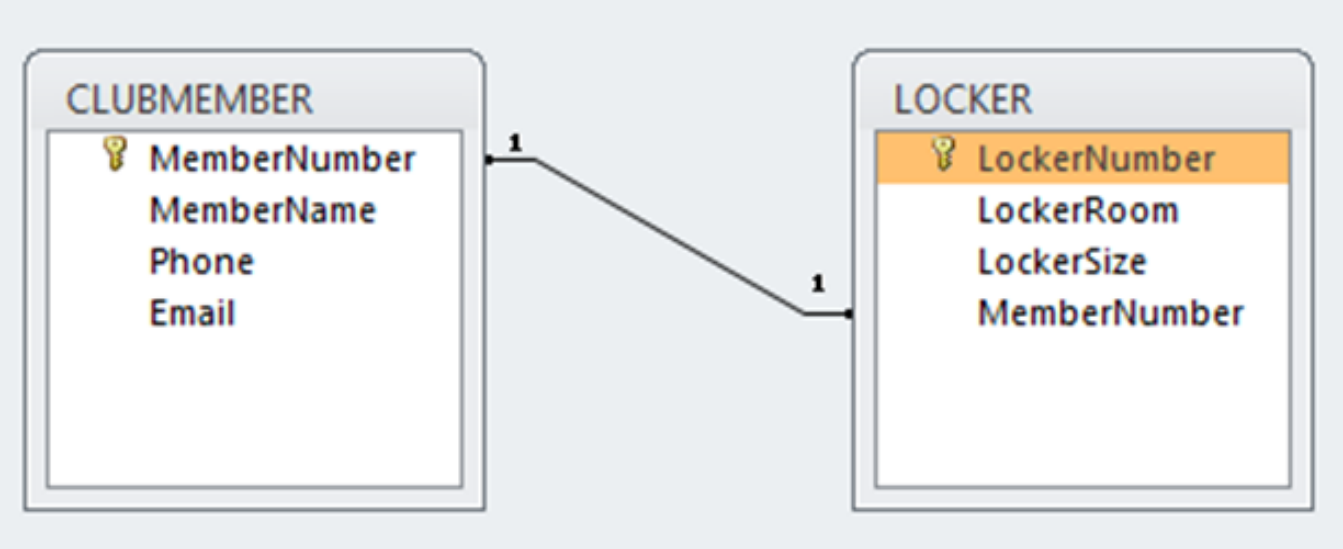
PRODUCTID	QUANTITY	PRICE
AB-111	1	75
AB-111	101	60
AB-111	501	50
ZA-245	1	42
ZA-245	11	40
ZA-245	51	35
MB-153	1	82

Summary of 1NF, 2NF, and 3NF

- A database is in 3NF if:
  - It is in 2NF
  - It has no transitive dependencies
- A database is in 2NF if:
  - It is in 1NF
  - There is no repeating data in its tables
- A database is in 1NF if:
  - All rows have a unique primary key
  - The values in each column of a table are atomic
  - There are no repeating groups (e.g., EMPLOYEE1, EMPLOYEE2…)

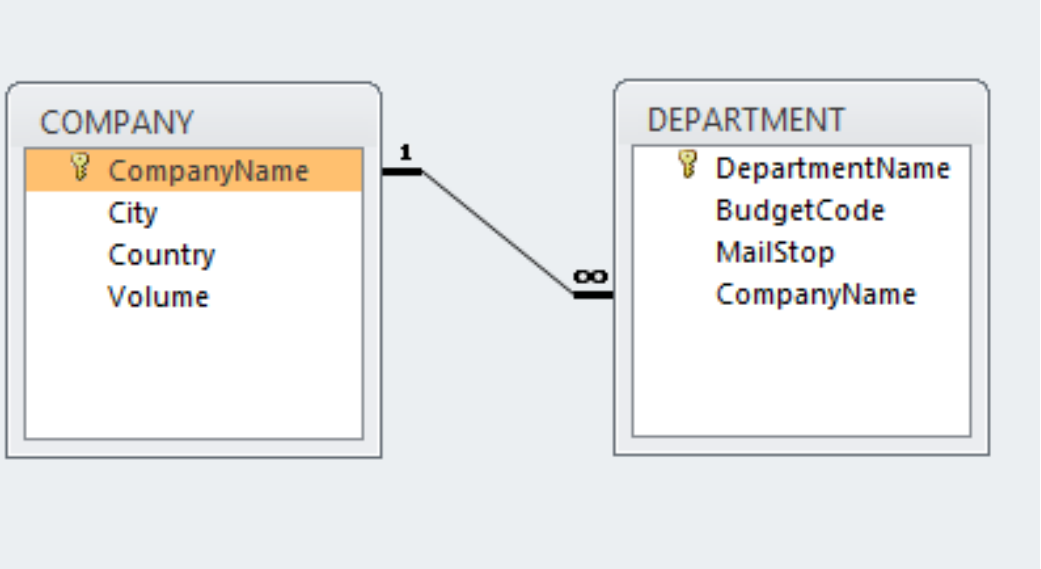
Forming Relationships

- One to One Relationship: Occur when each entry in the first table has one, and only one, counterpart in the second table.



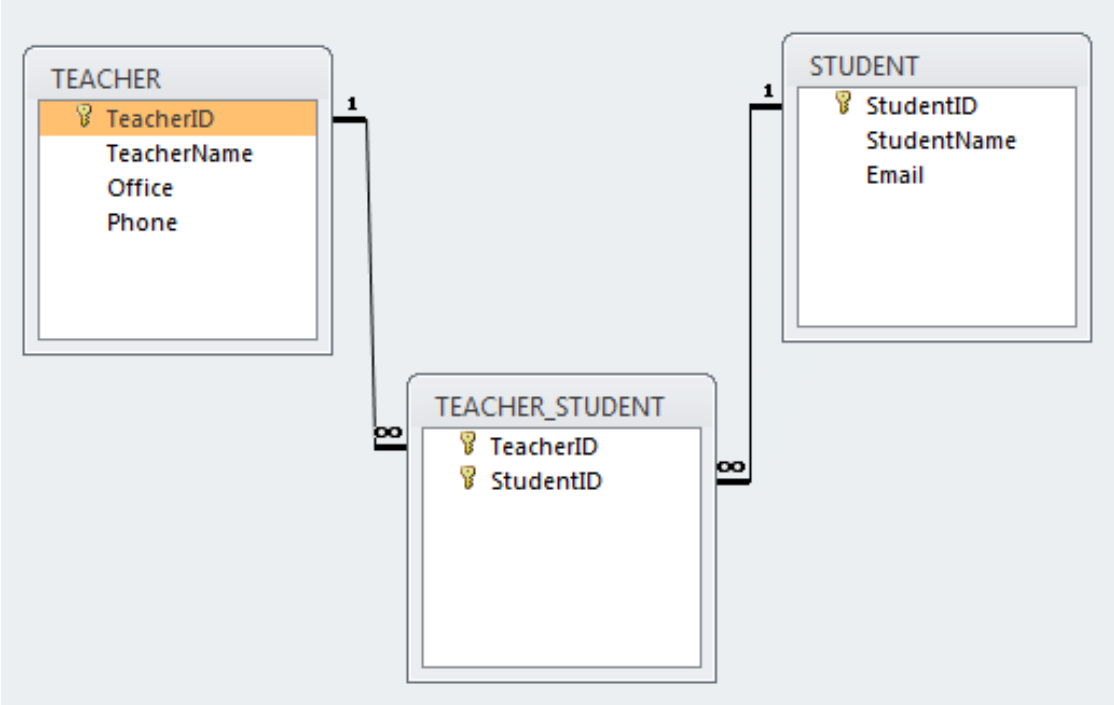
Forming Relationships

- One to Many Relationship: Occurs when one row in the first table corresponds to many rows in the second table



Forming Relationships

- Many to Many Relationship: Occurs when multiple rows from table A correspond to multiple roes in table B



Types of Databases

- **On-line Transaction Processing** (OLTP)
  - Day-to-day handling of transactions that result from enterprise operation
  - Database design of OLTP is highly normalized with many tables
  - Example applications: ERP, CRM
  - Backing up operational data is crucial to run the business

Types of Databases

- **On-line Analytic Processing** (OLAP)
  - Analysis of information in a database for the purpose of making management decisions
  - Database design of OLAP is typically denormalized
  - Example applications: Management information system, decision support system
  - Instead of regular backups, some environments may consider simply reloading the OLTP data as a recovery method

Denormalization

STUDENT

StudentID	Name
100	Jones
200	Chau
300	Garrett
400	Jones

CLUB

Club	Cost
Climbing	150
Scuba	400
Skiing	550

PAYMENT

StudentID	Club	AmtPaid
100	Scuba	0
100	Skiing	550
200	Scuba	400
300	Climbing	150
400	Skiing	550

Denormalization

Insert into PAYMENT_DATA

SELECT STUDENT.StudentID, STUDENT.Name, CLUB.Club, CLUB.Cost, PAYMENT.AmtPaid

FROM STUDENT, CLUB, PAYMENT

WHERE STUDENT.StudentID = PAYMENT.StudentID

AND PAYMENT.Club = CLUB.Club;

Denormalization

StudentID	Name	Club	Cost	AmtPaid
100	Jones	Scuba	400	0
200	Chau	Scuba	400	400
200	Chau	Skiing	550	550
300	Garrett	Climbing	150	150
400	Jones	Skiing	550	550