Introduction to Database

Mona Taghavi

What is a database?

- A database is a collection of interrelated data, which represents some aspects of the real world.

- Databases are essential today to support commercial, engineering, scientific applications and machine learning.

Examples of Database Applications

- **Database Applications:**
    - Banking: all transactions
    - Airlines: reservations, schedules
    - Universities: Information about students, courses, grades, professors, registration,..
    - Companies: Information about employees, departments, salaries, managers, ..
    - Online shopping websites: productions, personal information of the users, shopping histories, ….
- And many other applications

The first step in learning database

To better understand database, we need to first understand **some basic concepts**.

- A **database** is an electronic store of data. It is a repository that stores information about different “objects” or “things” called **Entities** and also contains **relationships** among them.

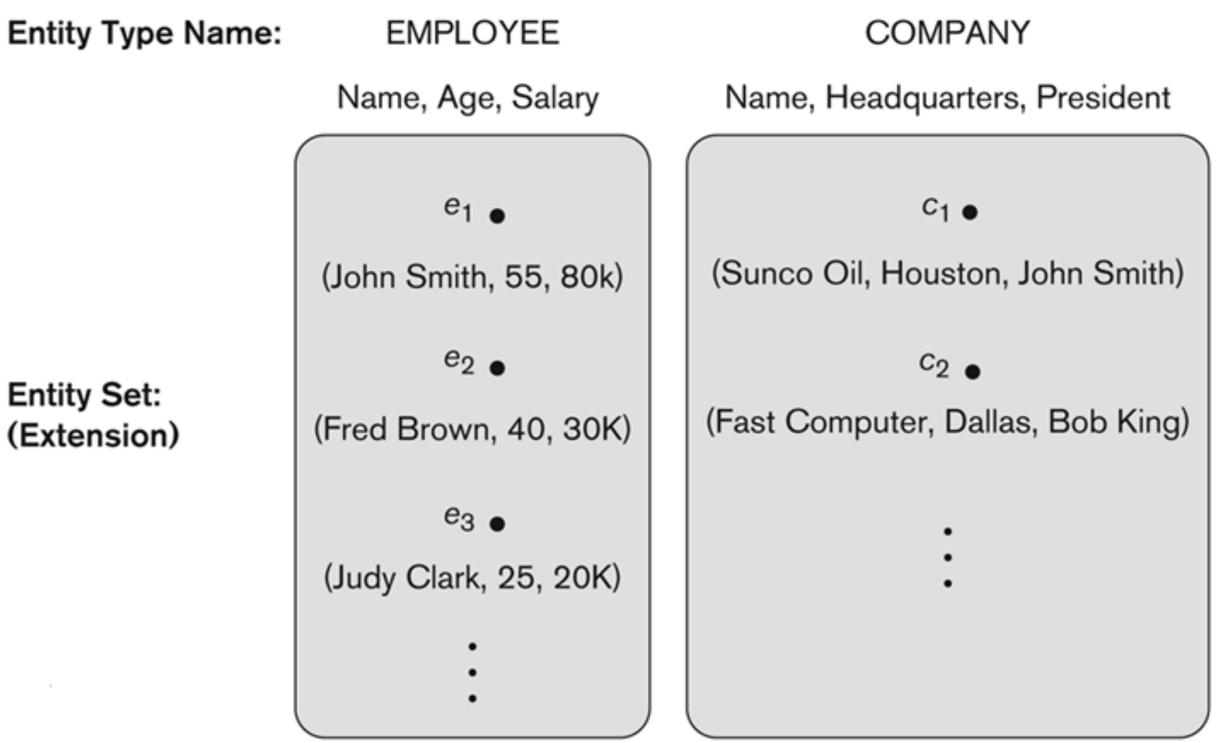
  - A person, place, event, or item are examples of an **entity**.
  - The facts describing an entity are known as **data**.

- **Each entity can be described by its characteristics, which are known as Attributes.** For example an EMPLOYEE entity may have a Name, SSN, Address, gender, BirthDate, email, etc.

The first step in learning database

- **Entities** with the same basic attributes are grouped to form an **Entity Set**.
- An **Entity Set** is given a singular name. For example:
  - The STUDENT entity set contains data about students only. All related entities in the STUDENT entity set are students.
  - Similarly, a company keeps track of all its employees in an entity set called EMPLOYEE. The EMPLOYEE entity set does not contain information about the company's customers.

The first step in learning database



The first step in learning database

- **A database is a collection of entity sets**.
  - For example, a college's database may include information about entities such as student, faculty, course, score , building, and so on.
  - Information about an enterprise is broken up into parts, with each relation storing one part of the information.

    account: stores information about accounts

    customer: stores information about customers

Problems with Lists: Redundancy

- In a list, each row is intended to stand on its own. As a result, the same information may be entered several times
  - For example, a list of projects may include the project manager's name, ID, and phone extension.
    - If a particular person is currently managing 10 projects, his/her information would appear in the list 10 times

Problems with Lists: Multiple Themes

- In a list, each row may contain information on more than one theme or business concept
- As a result, certain information might appear in the list only if information about other themes or business concepts is also present
  - For example, a list of projects may include project manager information (Name, ID, and Phone Extension) and project information (ProjectName, ID, StartDate, Budget) in the same row

List Modification Issues

- Redundancy and multiple themes in lists create modification problem
  - Deletion problems
  - Update problems
  - Insertion problems

Problems with the lists. **Update anomaly**

emp_id	emp_name	emp_address	emp_dept
101	Rick	Delhi	D001
101	Rick	Delhi	D002
123	Maggie	Agra	D890
166	Glenn	Chennai	D900
166	Glenn	Chennai	D004

- If we want to update the address of Rick then we have to update the same in two rows or the data will become inconsistent.
- If somehow, the correct address gets updated in one department but not in other then as per the database, Rick would be having two different addresses

Problems with the lists. **Insert anomaly**

- Suppose a new employee joins the company, who is under training and currently not assigned to any department then we would not be able to insert the data into the table if emp_dept field doesn’t allow nulls.

Problems with the lists. **Update anomaly**

- Suppose, if at a point of time the company closes the department D890 then deleting the rows that are having emp_dept as D890 would also delete the information of employee Maggie since she is assigned only to this department.

List Modification Issues

If Adviser **Baker** is changed to **Taing**, we need to change *AdviserEmail* as well. If changed to **Valdez**, we need to change *AdviserEmail*, *Department*, and *AdminLastName*.

	A	B	C	D	E	F	G
1	**LastName**	**FirstName**	**Email**	**AdviserLastName**	**AdviserEmail**	**Department**	**AdminLastName**
2	Andrews	Matthew	Matthew.Andrews@ourcampus.edu	Baker	Linda.Baker@ourcampus.edu	Accounting	Smith
3	Brisbon	Lisa	Lisa.Brisbon@ourcampus.edu	Valdez	Richard.Valdez@ourcampus.edu	Chemistry	Chaplin
4	Fischer	Douglas	Douglas.Fischer@ourcampus.edu	Baker	Linda.Baker@ourcampus.edu	Accounting	Smith
5	Hwang	Terry	Terry.Hwang@ourcampus.edu	Taing	Susan.Taing@ourcampus.edu	Accounting	Smith
6	Lai	Tzu	Tzu.Lai@ourcampus.edu	Valdez	Richard.Valdez@ourcampus.edu	Chemistry	Chaplin
7	~~Marino~~	~~Chip~~	~~Chip.Marino@ourcampus.edu~~	~~Tran~~	~~Ken.Tran@ourcampus.edu~~	~~InfoSystems~~	~~Rogers~~
8	Thompson	James	James.Thompson@ourcampus.edu	Taing	Susan.Taing@ourcampus.edu	Accounting	Smith
9	???	???	???	???	???	Biology	Kelly

Deleted row—Student, Adviser, and Department data lost

Inserted row—both Student and Adviser data missing

Addressing Information Complexities

- Relational databases are designed to address many of the information complexity issues that arise business

Relational Databases

- A relational databases stores information in table. Each informational theme (business concept) is stored in its one table.
- In essence, a relational database will break-up a list into several parts.
  - One part for each theme in the list
  - For example, a Project List might be divided into a CUSTOMER Table, a PROJECT Table, and a PROJECT_MANAGER Table

Relational Databases

Employee ID	Name	Phone Number
101	Dan	714-555-7270
102	Fry	657-555-6561
103	Leela	303-555-3247
104	Bender	480-555-4439

Employee ID	Name	Phone Number	Department ID
101	Dan	714-555-7270	1
102	Fry	657-555-6561	2
103	Leela	303-555-3247	2
104	Bender	480-555-4439	2

Department ID	Department Name
1	Information Systems
2	Package Delivery

Putting the Pieces Back Together

- In our relational database example, we broke apart our list into several tables. Somehow the tables must be *joined* back together
- In relational database, tables are joined together using matched pairs of values
  - For example, if a PROJECT has CUSTOMER, the CUSTOMER_ID can be stored as column in the PROJECT table. Whenever we need information about a customer, we can use Customer_ID to look up the customer information in the CUSTOMER table

Putting the Pieces Back Together

Project Name	Project ID	Start Date	Budget	Customer ID
DB upgrade	110	19 Nov 2015	$8700	2
New email server	111	09 Jan 2016	$11900	2
3D printers	112	06 Jul 2016	$25000	2
Network upgrade	113	24 Aug 2016	$33200	1

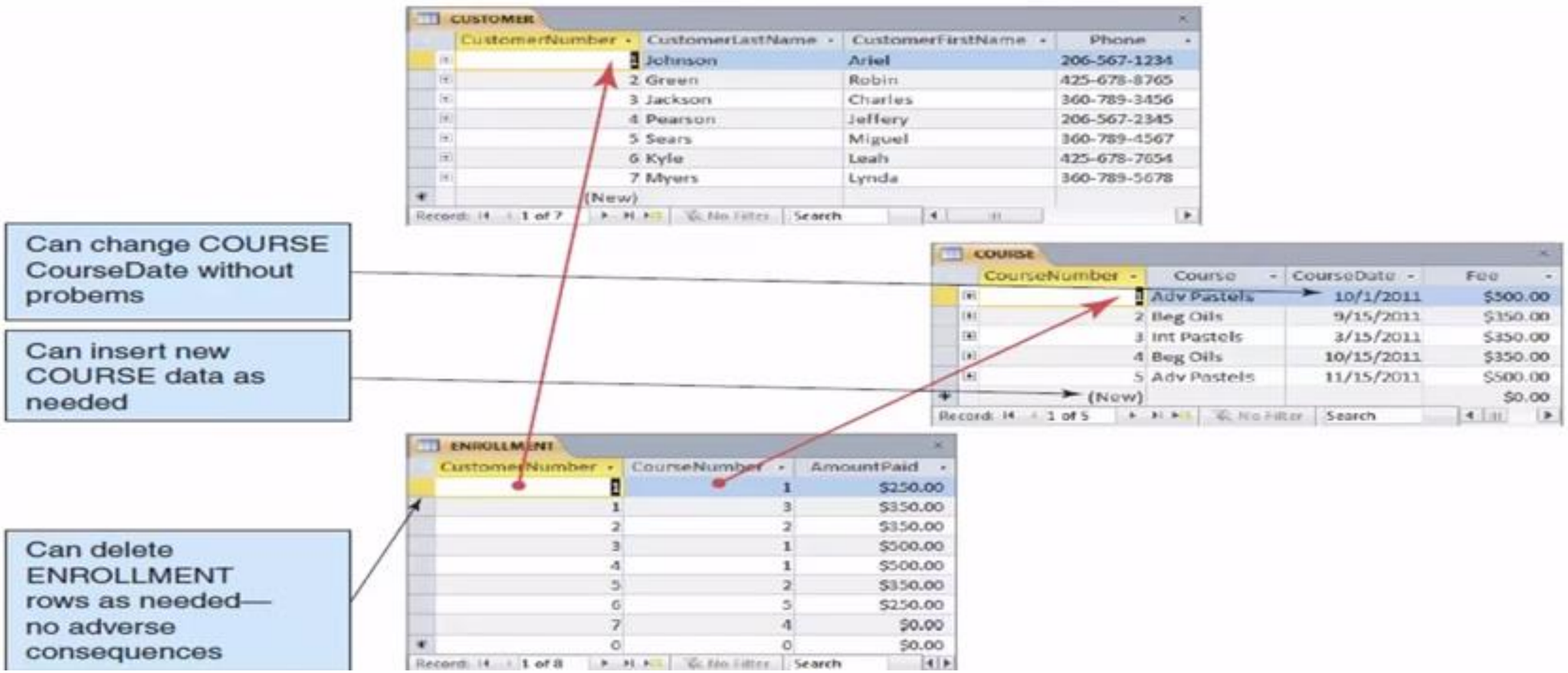
Customer ID	Customer Name
1	Dan
2	Priya

Sounds like More Work, Not Less

- A relational databases is more complicated than a list
- However, a relational database minimizes data redundancy, preserves complex relationships among topic, and allows for partial data (null values)
- Furthermore, a relational database provides a solid foundation for creating user interface forms and reports

The Structured Query Language (SQL)

- The Structured Query Language (SQL) is an international standard language for creating, processing, and querying databases and their tables
- The vast majority of data-driven applications and websites use SQL to retrieve, format, report, insert, delete. and/or modifv data for users



SQL Example

- We can use SQL to combine the data in the three tables in the Art Course Database to recreate the original list structure of the data
  - We do this using a SQL *SELECT* statement
  - SELECT Customer.customerLastname,
    Customer.customerFirstName,
    Customer.phone,
    Course.courseDate,
    Enrollment.amountPaid,
    Course.course,
    Course.fee
    FROM Customer, Enrollment, Course
    WHERE Customer.customerNumber=Enrollment.customerNumber
    AND Course.courseNumber = Enrollment.courseNumber

CustomerLastName	CustomerFirstName	Phone	CourseDate	AmountPaid	Course	Fee
Johnson	Ariel	206-567-1234	10/1/2011	$250.00	Adv Pastels	$500.00
Johnson	Ariel	206-567-1234	3/15/2011	$350.00	Int Pastels	$350.00
Green	Robin	425-678-8765	9/15/2011	$350.00	Beg Oils	$350.00
Jackson	Charles	360-789-3456	10/1/2011	$500.00	Adv Pastels	$500.00
Pearson	Jeffery	206-567-2345	10/1/2011	$500.00	Adv Pastels	$500.00
Sears	Miguel	360-789-4567	9/15/2011	$350.00	Beg Oils	$350.00
Kyle	Leah	425-678-7654	11/15/2011	$250.00	Adv Pastels	$500.00
Myers	Lynda	360-789-5678	10/15/2011	$0.00	Beg Oils	$350.00

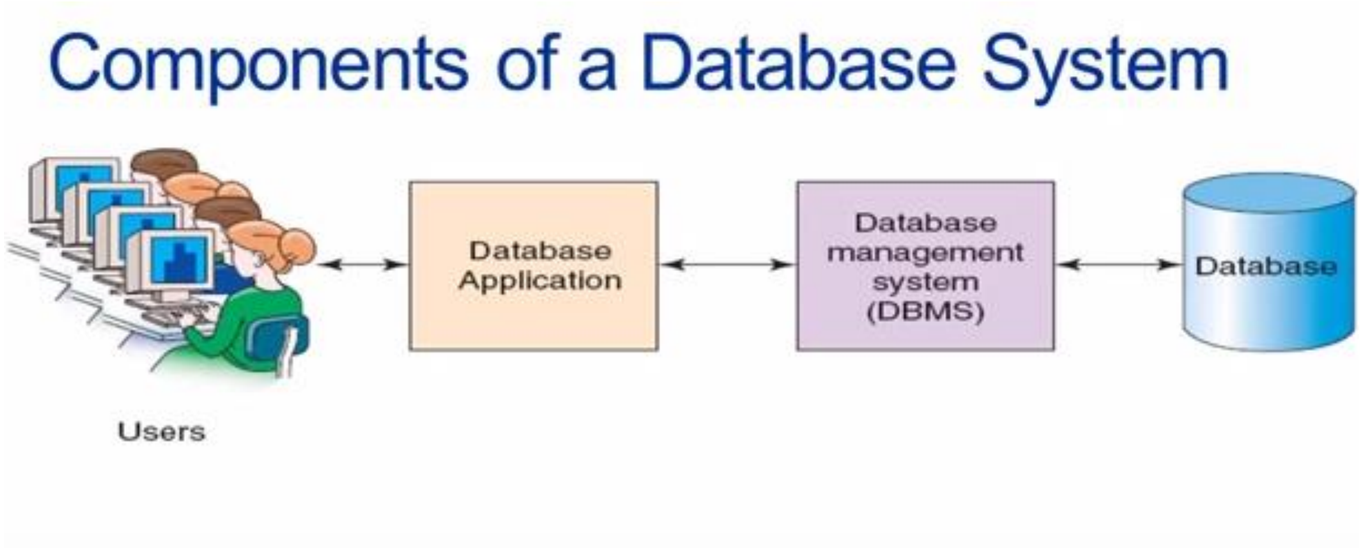
Exercise

- Design a database for a hotel
- Design a database for a college

Database Systems

The four components of a database system are:

- Users
- Database Application(s)
- Database Management System (DBMS)
- Database



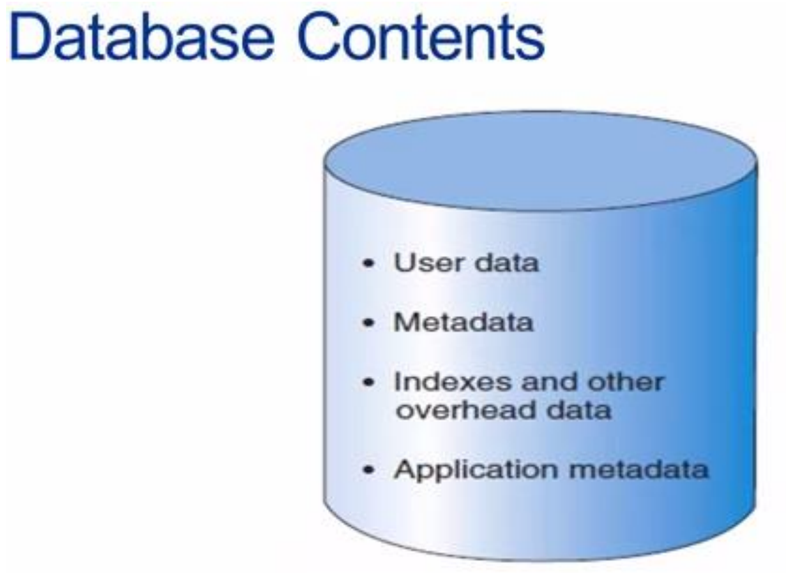
Users

A user of a database system will:

- Use a database application to keep track of information
- Use different user interface forms to enter, read, delete, and query data
- Produce report

The Database

- A database is a *self-describing* collection of *related* records
- Self-describing:
  - The database itself contains the definition of its structure
  - *Metadata* are data describing the structure of the data in the database
- Tables within a relational database are related to each other in some way

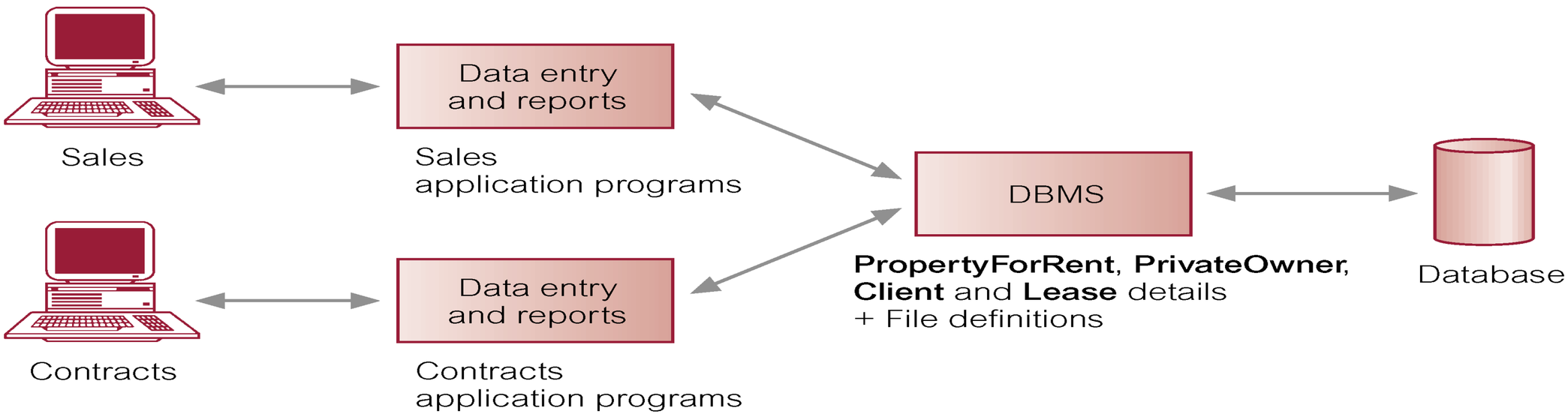


Database Management System (DBMS)

- A Database management system (DBMS) serves as an intermediary between database applications and database
- The DBMS manages and controls database activities

- *Dr. E. F. Codd published the paper, "A Relational Model of Data for Large Shared Data Banks", in June 1970 in the Association of Computer Machinery (ACM) journal, Communications of the ACM. Codd's model is now accepted as the definitive model for relational database management systems (RDBMS). The language, Structured English Query Language (SEQUEL) was developed by IBM Corporation, Inc., to use Codd's model. SEQUEL later became SQL.*

Database Management System (DBMS)



PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)
PrivateOwner (ownerNo, fName, lName, address, telNo)
Client (clientNo, fName, lName, address, telNo, prefType, maxRent)
Lease (leaseNo, propertyNo, clientNo, paymentMethod, deposit, paid, rentStart, rentFinish)

What does a DBMS provide?

- Supports convenient, efficient, and secure access and manipulation of large amounts of data
- Gives users the ability to create, query, and modify the data
- Supports the storage of data over a long period of time
- Controls access to shared data from multiple, simultaneous users
- Supports durability, the ability to recover from failures or errors of many types. And if by accident the database is damaged and some data are lost, DBMS also have the ability to restore the data.

Database design

- The database design requires you to create entity sets, each describing a set of related entities.

- The design also requires you to establish all the relationships between the entity sets within the database.

Database relationships

- The entities in a database are likely to interact with other entities. **The interactions between the entity sets are called Relationships**.

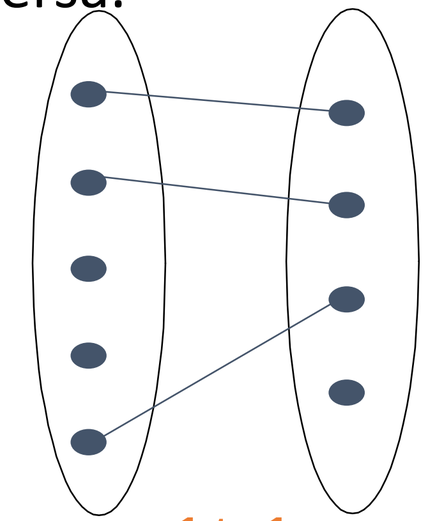
- A relationship relates two or more distinct entities with a specific meaning. For example, EMPLOYEE John Smith **works on** the ProductX PROJECT or EMPLOYEE Franklin Wong **manages** the Research DEPARTMENT

- Depending on the type of interaction, the relationships are classified into three categories:
  One-to-one (1:1)
  One-to-many (1:N) or Many-to-one (N:1)
  Many-to-many (N:M)

Relationships

One-to-One relationship:

- A one-to-one relationship exists between two entity sets, X and Y, if an entity in entity set X has only one matching entity in entity set Y, and vice versa.
- **For example,**
  - A department in a college has one chairperson and
    a chairperson chairs one department in a college.
  - An employee manages one department in a company, and
    only one employee manages a department.



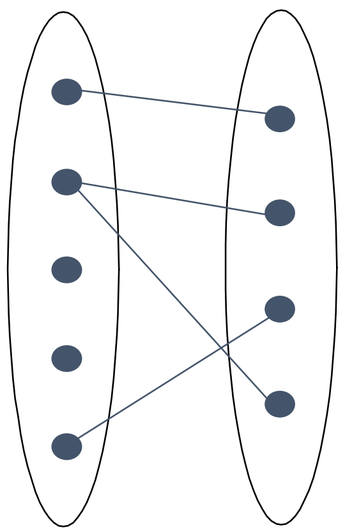
1-to-1

Relationships

One-to-Many relationship:

- A one-to-many relationship **exists between two entity sets**, X and Y, **if an entity in entity set X has many matching entities in entity set Y but an entity in entity set Y has only one matching entity in entity set X**.

- For example
  - An employee works in a department, but a department has many employees.



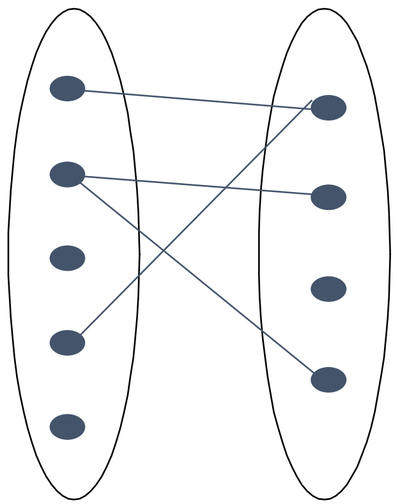
1-to Many

Relationships

Many-to-Many relationship:

- A many-to-many relationship **exists between two entity sets**, X and Y, **if an entity in entity set X has many matching entities in entity set Y and an entity in entity set Y has many matching entities in entity set X**.

- For example
  - A student takes many courses, and many students take a course. An employee works on many projects, and a project has many employees.



Many-to-Many

Relationships - quiz

- A customer's order may contain one or more products; and a product can appear in many orders.
- A teacher may teach zero or more classes, while a class is taught by one teacher.
- In a "company" database, a manager manages zero or more employees, while an employee is managed by one manager.
- In a "product sales" database, a customer may place many orders; while an order is placed by one particular customer.
- In a "bookstore" database, a book is written by one or more authors; while an author may write zero or more books.

- Everyone give me an example of one of these three relationships.

Relational Data Model

- The most common data models.
- Based on mathematical concept of a relation.

➢It uses a relation as the building block of the database.

Consists of three components:

- Relational data structure
  - Where data is organized
- Data manipulation
  - Operations used to manipulate data stored in the data structure
- Relational data integrity
  - Rules that maintain the integrity of data when manipulated

Relational Data Model

- Data is organized in tables

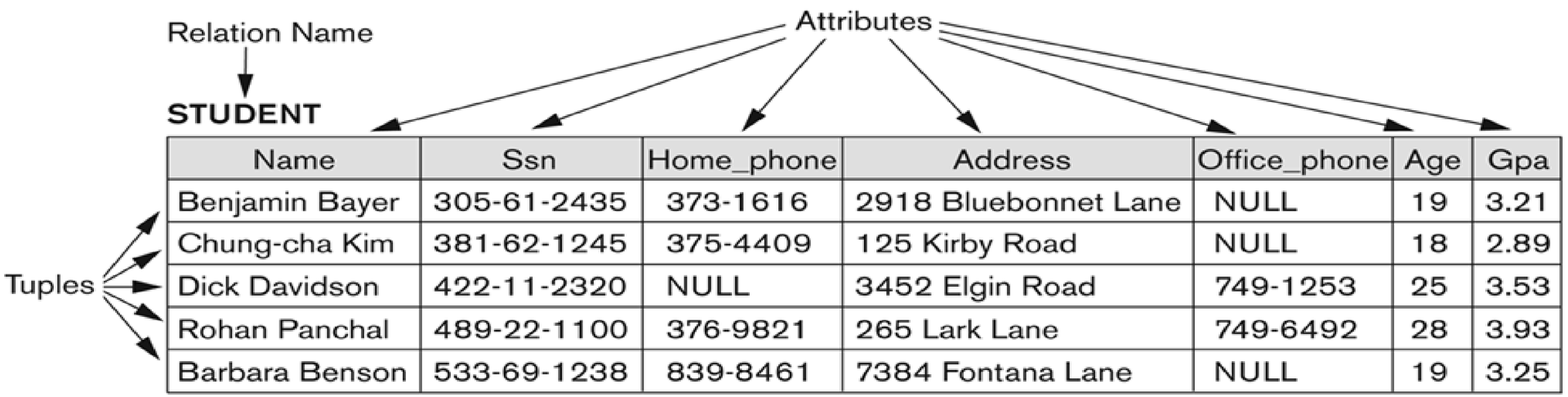
- Table also called a relation because the relational model's creator, Codd, used the term relation as a synonym for table

- A **relation** is the main data structure that stores and organizes data in the relational data model
- A table is a matrix of rows and columns in which each row represents an entity and each column represents an attribute.

Relational Data Model

- A column of a relation is referred to as an **attribute**
  - The number of attributes in a relation is called the **degree of the relation**

- A row is referred to as a **record** or a **tuple**
  - The number of records in a relation is defined as the **cardinality of the relation**



Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	749-1253	25	3.53
Rohan Panchal	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	NULL	19	3.25

In daily practice, the terms ***table***, ***relation***, and ***entity set*** are used interchangeably.

Properties of a Relation

- Each **relation** is **uniquely identified by its name**
- Each **cell** of a relation contains exactly **one (atomic) value**
- Each **record** of a relation is **unique**
- Each **attribute** in a relation has a **distinct name**
- The values of an attribute are from the **same domain**
- The order of attributes is irrelevant
- The order of records is also irrelevant

Primary Key

- We need to specify one or more attributes that uniquely identify each **Entity in an Entity Set.**
- **Primary key (PK)** is an attribute (or a combination of attributes) that uniquely identifies any given entity (row)
    - It must be unique at all times
    - The candidate key can never change
    - It cannot hold a NULL value
- Nulls:
    - No data entry - Should be avoided in other attributes - Can represent:
        - An unknown attribute value
        - A known, but missing, attribute value
        - A “not applicable” condition

Foreign Key

- In a relational database, tables are related to each other through a common column.
- **A column in a table that references a column in another table is known as a foreign key**.
- A **foreign key** is an attribute or a set of attributes in a relation that serves as a primary key of some other relation

STUDENT

SSN	Name	Email	*DeptName*
234-23-2535	John Doe	John.Doe@mail.com	ISE
545-28-2324	Samuel Ed	Samuel.Ed@mail.com	CISE
368-98-4234	Jay Leno	Jay.Leno@mail.com	ISE
975-34-3558	Ricky Grey	Ricky.Grey@mail.com	PHY

Foreign Key

DEPARTMENT

Primary Key

DeptName	Address	Phone	*College ID*
ISE	Weil Hall, GNV, FL	352-222-1111	COE
CISE	Marston Science, GNV, FL	352-333-9999	COE
PHY	Physics Bldg, GNV, FL	352-444-8888	CLAS
CHEM	Chemistry Bldg, GNV, FL	352-555-3333	CLAS

Relational Data Integrity

- Two integrity of the relational database:
    - Entity Integrity
    - Referential Integrity

Relational Data Integrity

Entity Integrity

- Primary key attributes may not be null. The primary key uniquely identifies an entity, so it cannot be null and it must have a unique value.
- Guarantees that each entity will have a unique identity and ensures that foreign key values can properly reference primary key values.

- Requirement
  - No component of the primary key is allowed to accept nulls.
  - By "null" here, we mean that information is missing for some reason.

Relational Data Integrity

Referential Integrity –

- A foreign key may have a null value or it may have a value that exists as a primary key value in another table it references.
- The database must not contain any unmatched foreign key values.
- The referential integrity rule simply says that if B references A, then A must exist.

Alternative Terminology for Relational Model

Formal terms	Alternative 1	Alternative 2
Relation	Table	File
Tuple	Row	Record
Attribute	Column	Field

Summary so far

Entity – A person, place, item, event or a ‘thing’.

Entity Set – A collection of related entities. For example, entity INVOICE contains information about invoices.

Attribute – A characteristic that describes an entity. For example, invoice number and invoice date are attributes of entity INVOICE.

Tuple – Another name for a row or a record. It contains attributes of an entity.

Domain – A set of all possible values for an attribute.

Key – A minimal set of attributes, which uniquely defines an entity in an entity set.

Null – Unknown, unspecified, not known or not applicable value.