Database Systems

- *Instructor*

  Dr. Sanam Shahla Rizvi

  PhD in Information and Communication from Ajou University, Korea

Course Details

- Textbook
  - Database System by Catherine Ricardo
- Reference
  - Data Management Systems
    by Raghu Ramakrishnan and Johannes Gehrke
  - Wikipedia
  - Internet
- Lecture
  - Time: Mon (07:45-09:35), Wed (07:45-08:35)
  - Venue: Lecture Hall

Course Details

- Grading Scheme (Theory 100%)
  - Mid-Term Exam 30%
  - Final Exam 40%
  - Quizzes 15%
  - Class Participation 5%
  - Final Report + Assignments 10%

Quizzes

- Prepare previous lectures and class discussions in every class because there might be surprise quiz every time.

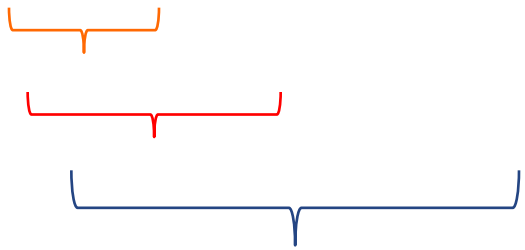
Final Report

- Search, read and evaluate database management systems like MS Access, MS SQL Server and Oracle.
- Determine how they are organizing and handling data in databases.
- Provide the functionality relevancies and differences between databases of **any two** DBMSs.
- Such technical report should not exceed 3 pages.

Database Systems

Database Systems

1. .
2. .
3. .



What is/are Data?

What is Data?

- Data is often viewed as the lowest level of abstraction from which information and knowledge are derived.
- Data can exist in a variety of forms -- as numbers or text on pieces of paper, as bits and bytes stored in electronic memory, or as facts stored in a person's mind.
- Raw data refers to a collection of numbers, characters, images or other outputs from devices that collect information to convert physical quantities into symbols, that are unprocessed.

WHAT IS DATA? AND WHERE IS IT NOW?

- Data is factual information about objects and concepts, such as:
  - measurements
  - statistics
- You can find data in:
  - filing cabinets
  - spreadsheets
  - folders
  - ledgers
  - lists
  - colleagues' memories
  - piles of papers on your desk

What is Database?

What is a Database?

- A database is a storage space for content / information (data).
- A database consists of an organized collection of data typically describing the activities of one or more related organizations.

What is a Database?

- A database can be of **any size** and of **varying complexity**.
  - For example, the list of names and addresses of friends.
  - The book catalog of a large library may contain half a million records.
  - A database of much greater size and complexity is maintained by the government to keep track of the tax information filed by taxpayers.

What does “managing information” mean?

- Making information *work* for us
- Making information useful
- Avoiding "accidental disorganisation”
- Making information easily accessible and integrated with the rest of our work

Managing as *re-organising*

- We often need to access and re-sort data for various uses. These may include:
  - Creating mailing lists
  - Writing management reports
  - Generating lists of selected news stories
  - Identifying various client needs
- Can you add to the list?

Managing as *re-processing*

- The processing power of a database allows it to:
  - Sort
  - Match
  - Link
  - Aggregate
  - Skip fields
  - Calculate
  - Arrange

Why study databases?

- Databases used to be *specialized applications*, now they are a *central component* in computing environments.
- Knowledge of database concepts is *essential* for computer scientists.

Why study databases?

- Databases are everywhere, even when you don't see them: most activities involve data
  - Banking + credit cards: all transactions
  - Airlines: reservations, schedules
  - Universities: registration, grades
  - Telecommunications/networks
  - Sales: customers, products, purchases
  - Manufacturing: production, inventory, orders, supply chain
  - Human resources: employee records, salaries, tax deductions
  - **Web sites**: front end to information stored in databases; e.g., Google, YouTube, Flickr, AddAll…
  - Scientific research, e.g., studying the environment
- Data needs to be *managed*

Why study databases?

- Data is valuable:
  - E.g., bank account records, tax records, student records, your videos and photos…
  - It must be protected - no matter what happens whether we have machine crashes, disk crashes, hurricanes/floods…
  - It also needs to be protected from **people.**

Why study databases?

- Data is often structured:
  - Bank account records all follow the same structure.
  - We can exploit this regular structure
    - To retrieve data in useful ways (that is, we can use a *query* language)
    - To store data efficiently

Why study Databases?

- Because the database field has made a number of contributions to basic computer science
  - Databases are behind many of important contributions and impact that CS has had
  - Find, gather, analyze and understand data, e.g.,
    Banks, human genome, ecommerce, Web:
- *Understand concepts and apply to different problems and different areas*
- Because DBMS software is highly successful as a commercial technology (Oracle, DB2, MS SQL Server…)
- Because DB research is highly active and **very** interesting!
  - Lots of opportunities to have practical impact

Big Names in Database Systems

Company	Product
Oracle	Oracle 8i, 9i, 10g
IBM	DB2, Universal Server
Microsoft	Access, SQL Server
Sybase	Adaptive Server
Informix	Dynamic Server

Who Needs Database Systems

Corporate databases

Typical Applications:

Personnel management
Inventory and purchase order
Insurance policies and customer data
… …

Web data management

Typical Applications:

Web page management
Personalize web pages
ECommerce
… …

What is in a Database?

- A ***database*** contains information about a particular *enterprise* or a particular *application.*
  - E.g., a database for an enterprise may contain everything needed for the planning and operation of the enterprise: customer information, employee information, product information, sales and expenses, etc.
  - You don't have to be a company to use a database: you can store your personal information, expenses, phone numbers in a database (e.g., using Access on a PC).
  - As a matter of fact, you could store all data *significant to a particular purpose* in a database.
  - This usually means that a database stores data that are *related* to each other.

Different parts of a database

- Fields
- Records
- Queries
- Reports

Fields

- Database storage units
- Generic elements of content

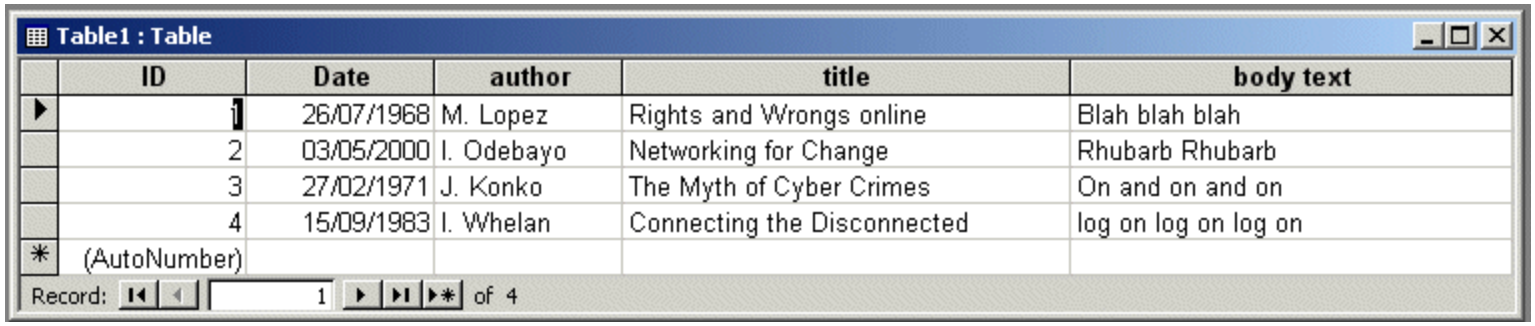
Field Name	Data
First name	Sana
Family name	Yousuf
Nationality	Pakistani
CNIC	123456
Date of birth	15 Sept 1983
Marital status	Single
Occupation	Journalist
Place of issue	Islamabad
Valid untill	17 Dec 2012

Records

A simple table showing fields (columns) and records (rows):

Record ID	Field a (date)	Field b (author)	Field c (title)	Field d (body text)
1	26/07/1968	M. Lopez	Rights and Wrongs online	Blah blah blah
2	03/05/2000	I. Odebayo	Networking for Change	Rhubarb Rhubarb
3	27/02/1971	J. Konko	The Myth of Cyber Crimes	on and on and on
4	15/09/1983	I. Whelan	Connecting the disconnected	Log on log on log

And as part of an MS Access database table:



Queries

- Queries are the information management (insert, retrieve, update, delete) requests you make to the database.
- Your queries are all about the information you are trying to gather or are already gathered.

Reports

- If the query is a question…

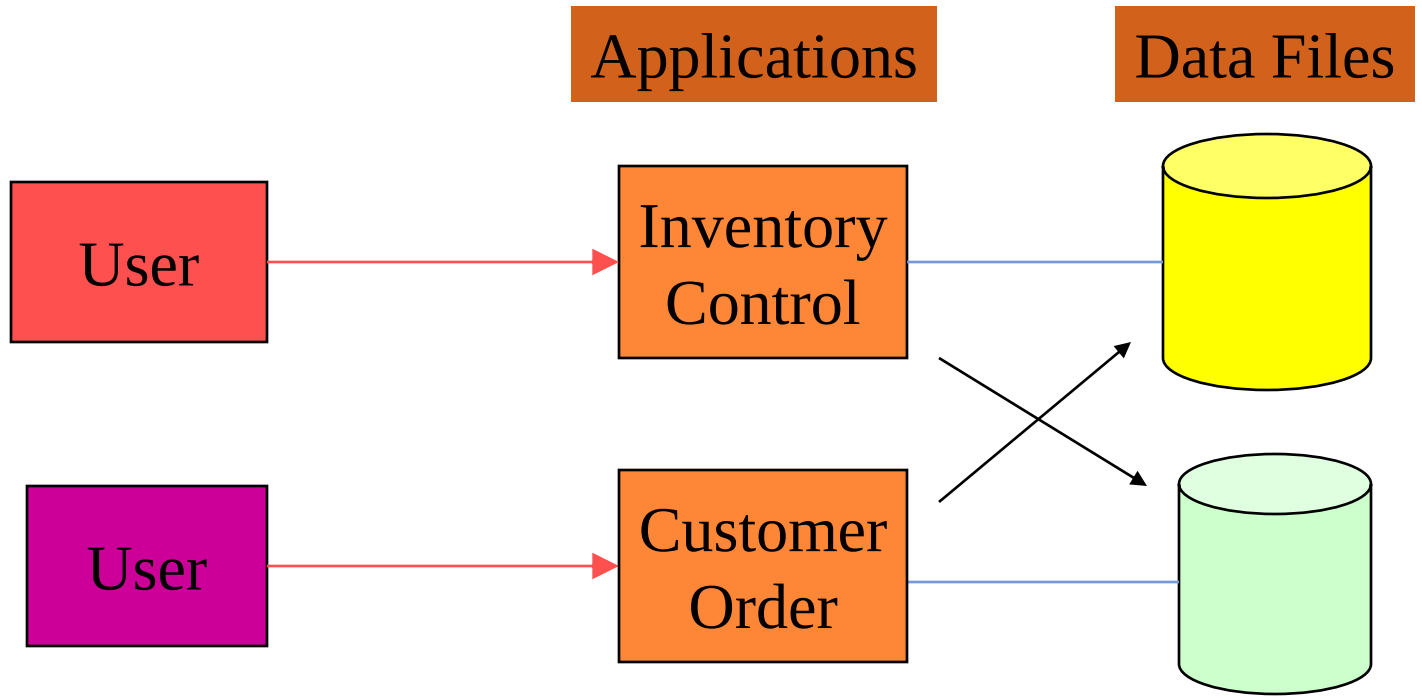
…then the report is its answer.

- Reports can be tailored to the needs of the data-user for making the information much more useful, they extract.

Is a database the same as a file?

- You can store data in a file or a set of files, but …
  - How do you *input* data and to *get back* the data from the files?
- A database is *managed* by a DBMS.

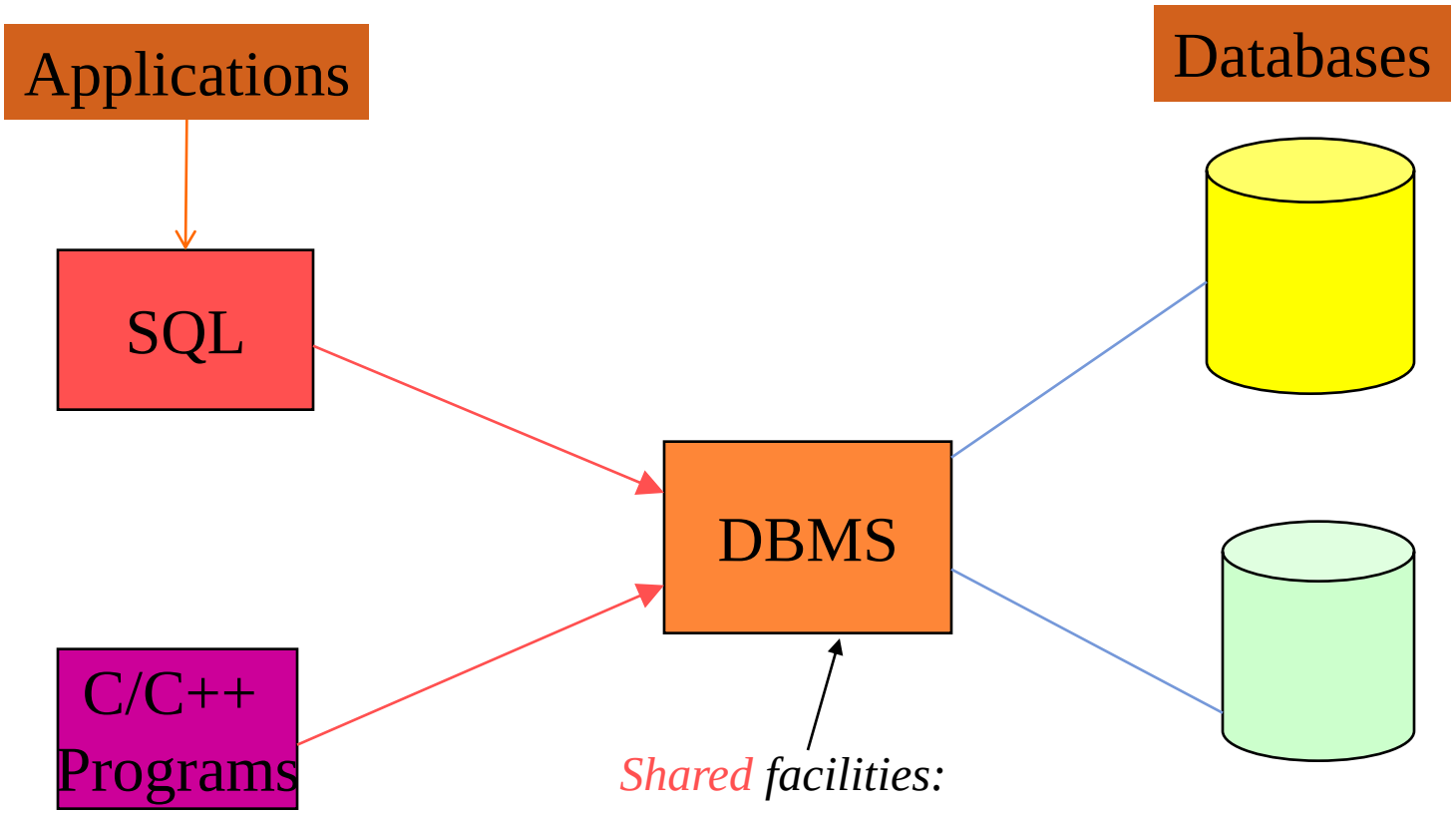
BEFORE WE HAVE DBMS



Question: When a customer ordered 10 PC monitors, how many files do you have to update?

Key issues: data sharing, data redundancy

After we have DBMS



Shared facilities:

- *Backup and recovery*
- *Data storage and access modules*
- *Programming tools, etc.*

Purpose of Database Systems

- Database management systems were developed to handle the difficulties caused by different people writing different applications independently.

Purposes of Database Systems

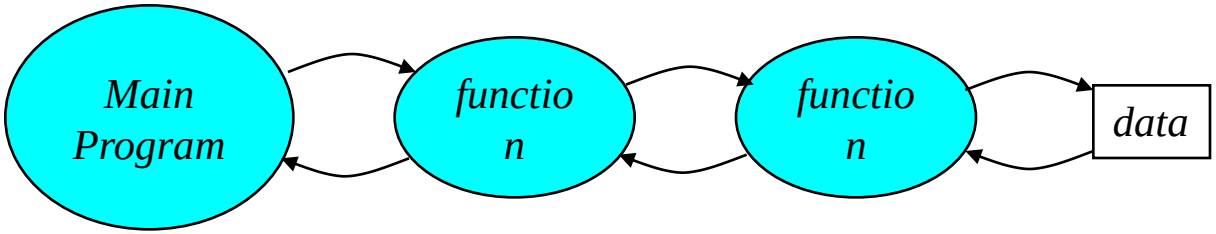
- A DBMS attempts to resolve the following problems:
  - Data redundancy and inconsistency by keeping one copy of a data item in the database
  - Difficulty in accessing data by provided query languages and shared libraries
  - Data isolation (multiple files and formats)
  - Integrity problems by enforcing constraints (age > 0)
  - Atomicity of updates
  - Concurrent access by multiple users
  - Security problems

Data Independence

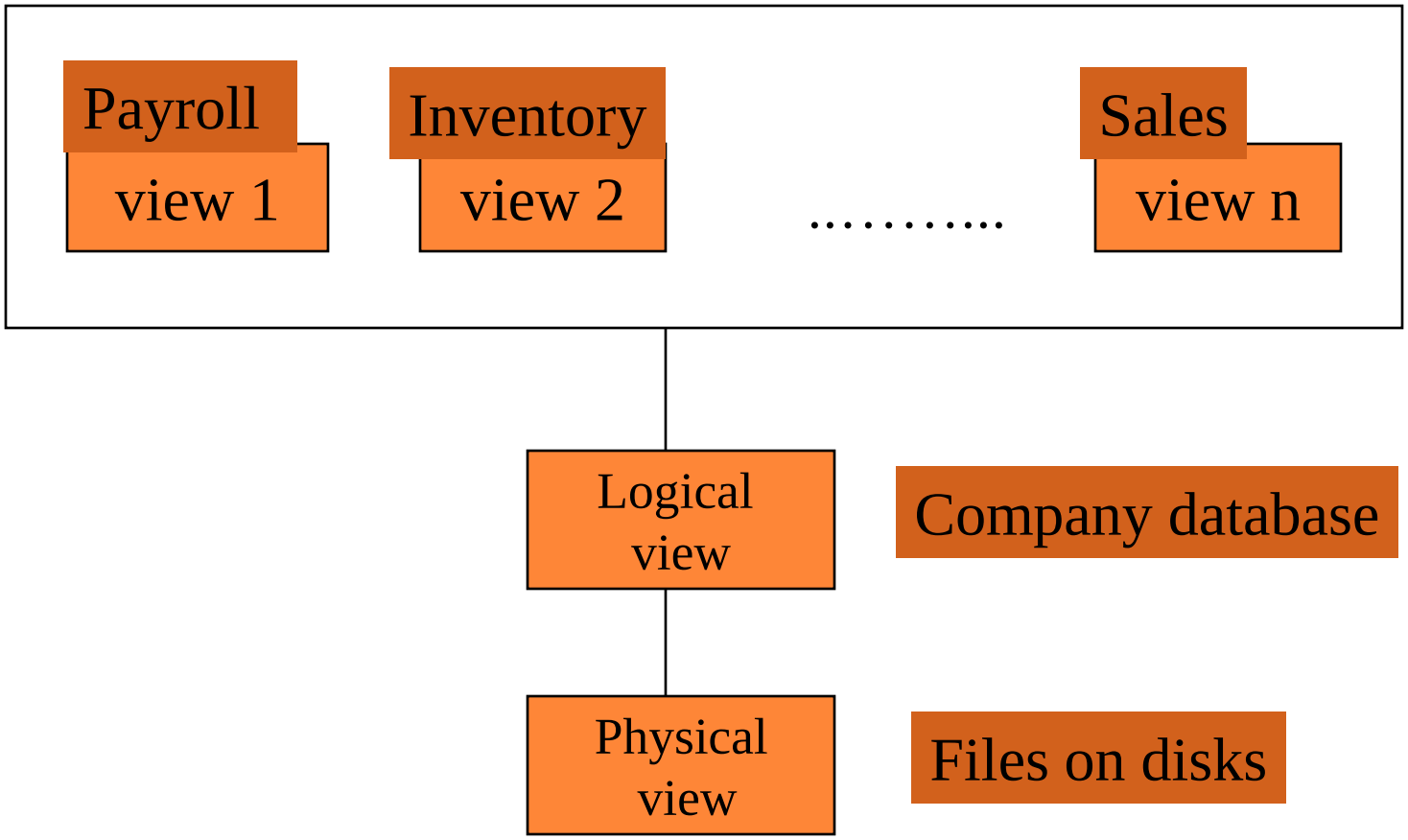
- One big problem in application development is the *separation* of applications from data
- Do I have change my program when I …
  - replace my hard drive?
  - store the data in a b-tree instead of a hash file?
  - partition the data into two physical files (or merge two physical files into one)?
  - store salary as floating point number instead of integer?
  - develop other applications that use the same set of data?
  - add more data fields to support other applications?
  - … …

Data Abstraction

- The answer to the previous questions is to introduce levels of *abstraction* of *indirection*.
- Consider how do *function calls* allow you to change a part of your program without affecting other parts?



Three Levels of Abstraction



Three Levels of Abstraction (cont.)

- *Physical level:* describe how a record is stored on disks.
  - e.g., "Divide the customer records into 3 partitions and store them on disks 1, 2 and 3."
- *Logical level:* describes data stored in database, and the relationships among the data.
  - e.g., *Students(sid: string, name: string, login: string, age: integer, gpa: real)*
- *View level:* Define a subset of the database for a particular application. Views can also hide information (e.g. salary) for security purposes.

Data Independence

- Ability to modify a schema definition in one level without affecting a schema definition in the next higher level.
- The interfaces between the various levels and components should be well defined so that changes in some parts do not seriously influence others.
- Two levels of data independence:
  - Physical data independence
  - Logical data independence

Instances and Schemas

- Each level is defined by a *schema,* which *defines* the data at the corresponding level
  - A logical schema defines the logical structure of the database (e.g., set of customers and accounts and the relationship between them)
  - A physical schema defines the file formats and locations
- A database *instance* refers to the actual content of the database at a particular point in time. A database instance must conform to the corresponding schema

Database Languages
Data Definition Language (DDL)

- Specification notation for defining the database schema
  - Express what were in the previous two slides to the DBMS in a formal language
- *Data storage and definition language* - special type of DDL in which the storage structure and access methods used by the database system are specified
- *SQL*
  - *CREATE (CREATE TABLE customer (Name char(50), Address char(50), Birth_Date date))*
  - *DROP (DROP TABLE customer)*
  - *ALTER*

Database Languages
Data Manipulation Language (DML)

- Language for accessing and manipulation the data organized by the appropriate data model.

- *SQL*
  - *SELECT*
  - *INSERT*
  - *UPDATE*
  - *DELETE*

Database Languages
Data Control Language (DCL)

- DCL used to control access to data in a database.

- *SQL*
  - GRANT to allow specified users to perform specified tasks.
  - REVOKE to cancel previously granted or denied permissions.

People who deal with databases.
Database Administrator (DBA)

- Coordinates all the activities of the database system; the database administrator has good understanding of the enterprise's information resources and needs.
- Database administrator's duties include:
  - Schema definition/modification
  - Storage structure definition/modification
  - Authorization of data access
  - Integrity constraint specification
  - Monitoring performance
  - Responding to changes in requirements

People who deal with databases.
Database Users

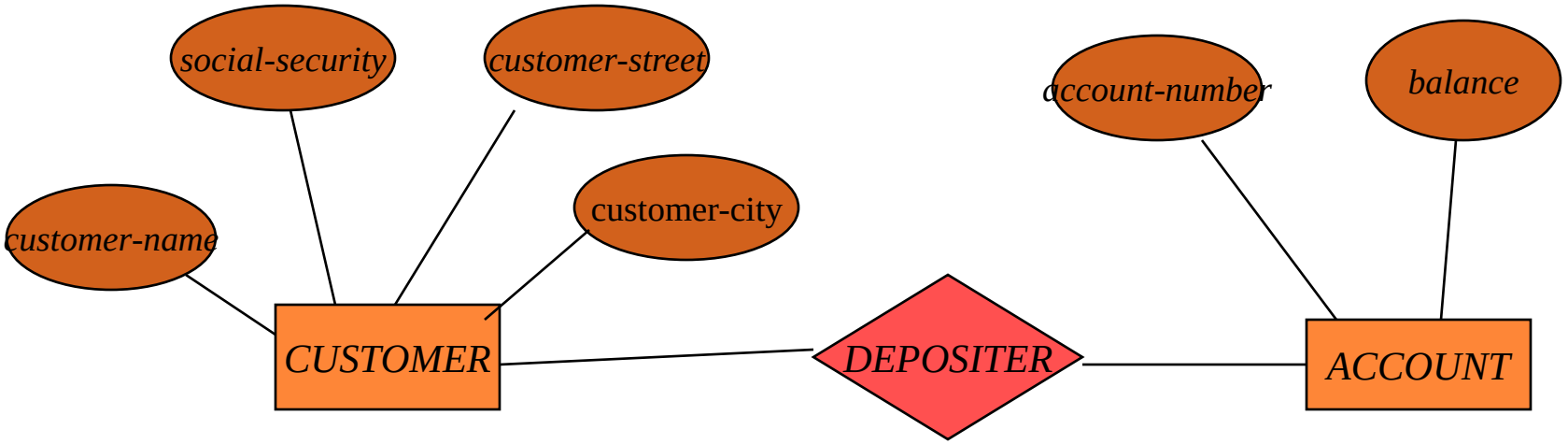
- Users are differentiated by the way they expected to interact with the system
- Application programmers
  - Develop applications that interact with DBMS through DML calls
- Sophisticated users
  - Post request in database query language (SQL interface)
- End users
  - Invoke one of the existing application programs (e.g., print monthly sales report)
  - Interact with applications through GUI

Data Models

- A data model is a collection of tools or concepts for describing data, the meaning of data, data relationships and data constraints.

Entity-Relationship (ER) Model

- Allows to pictorially denote entities and the relationships among them.



Relational Model

- Main concept: relation is basically a table with rows and columns

customer-name	*social-security*	*customer-street*	*customer-city*	*account-number*
Johnson	192-83-7465	Alma	Palo Alto	A-101
Smith	019-28-3746	North	Rye	A-215
Johnson	192-83-7465	Alma	Palo Alto	A-201
Jones	321-12-3123	Main	Harrison	A-217
Smith	019-28-3746	North	Rye	A-201

account-number	*balance*
A-101	500
A-201	900
A-215	700
A-217	750

Flat Files

Characteristics:

- Data is stored as records in regular files
- Records usually have a simple structure and fixed number of fields
- For fast access may support indexing of fields in the records
- No mechanisms for relating data between files
- One needs special programs in order to access and manipulate the data

Relational Database

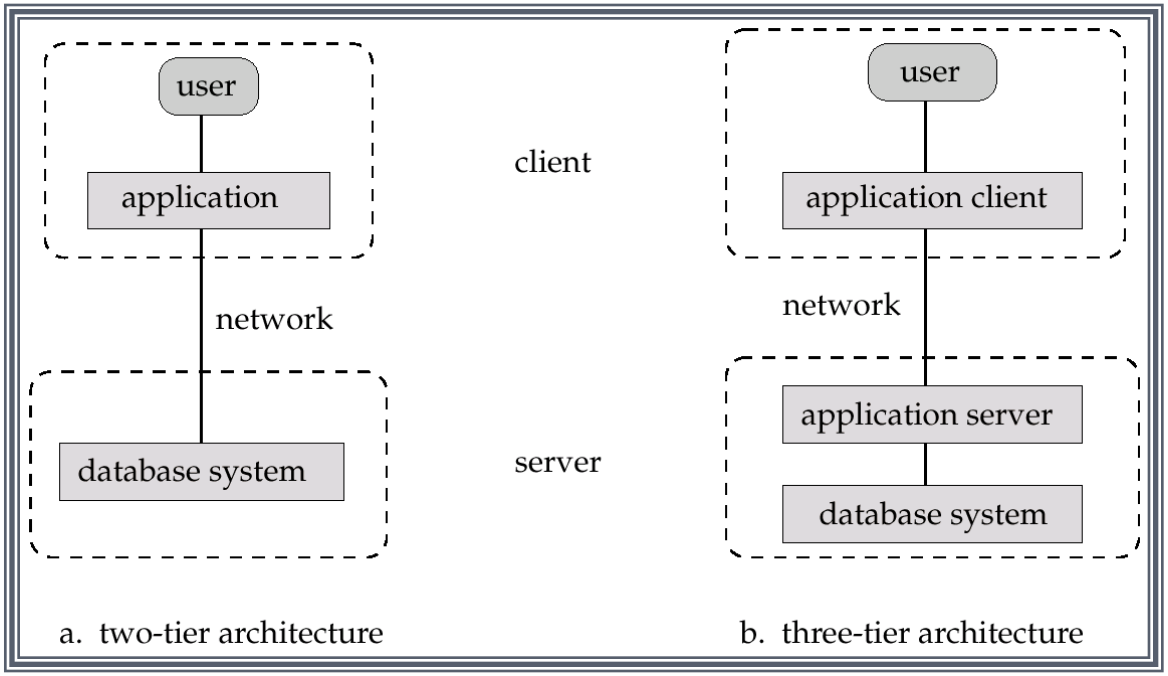
Characteristics:

- Data is organized into tables: rows & columns
- Each row represents an instance of an entity
- Each column represents an attribute of an entity
- Metadata describes each table column
- Relationships between entities are represented by values stored in the columns of the corresponding tables (keys)
- Accessible through Standard Query Language (SQL)

Metadata

- Data that describes the properties or characteristics of other data
- Allows database designers and users to understand the meaning of the data

Application Architectures



- **Two-tier architecture**: E.g. client programs using ODBC/JDBC to communicate with a database
- **Three-tier architecture**: E.g. web-based applications, and applications built using "middleware"

Database Types

Type	Typical number of users	Typical architecture	Typical size
Personal	1	Desktop/Laptop/ PDA	MB
Workgroup	5-25	Client/server:2 tier	MB-GB
Department	25-100	Client/server:3 tier	GB
Enterprise	>100	Client/server: distributed	GB-TB
Internet	>1000	Web sever & application servers	MB-GB