CSCD 327: Relational Database Systems

Introduction

Instructor: Dr. Dan Li

General Information

- Meeting Time:
- Instructor:
- Office:
- Phone:
- Email:
- Office Hours:
- Course Website:
- Pre-requisite:

Textbook

- SQL The Complete Reference, 3rd Edition (ISBN-10: 0071592555, ISBN-13: 978-0071592550)

- Other recommended books
  - Database System Concepts, 6th Edition (ISBN 978-0-07-352332-3)
  - Database Management Systems, 3rd Edition (ISBN 978-0-07-246563-1)
  - Database Systems The Complete Book, 2nd Edition (ISBN 978-0-13-187325-4)
  - SQL Bible, 2nd Edition (ISBN 978-0-470-22906-4)
  - SQL Cookbook, 1st Edition (ISBN 978-0-59-600976-2)

Course Structure/Approach

- What do you want to get out of your college education and this course?

- Lecture-centered

- Lab integrated

- Reading Outside of Class

Assessment & Grading System

	Total Weight
Homework including labs	40%
Midterm Exam	25%
Final Exam	30%
In-class participation including quizzes	5%

95% or above 4.0
70% or above (% - 70)/12.5 + 2.0
Below 70 0.0
Note: No grade will be given in the range 0.1 - 1.9

Course Policies

- Academic Integrity
- Disability Accommodation
- Makeup Tests
- Attendance
- Homework Late Policy
- Withdrawal/Incomplete Policy
- Electronic Device Usage
- Other University Policies

Course Outline

- Curriculum change in CS
- DBMS
- Two perspectives
  - External ("How to use a DBMS")
  - Internal ("How to build a DBMS")

Ready to Start?

- What is a DB?
- What is a DBMS?
- Why use a DBMS?
- Database Languages
- Data Models
- Database Users
- Database Architecture
- History of SQL

What Is a Database?

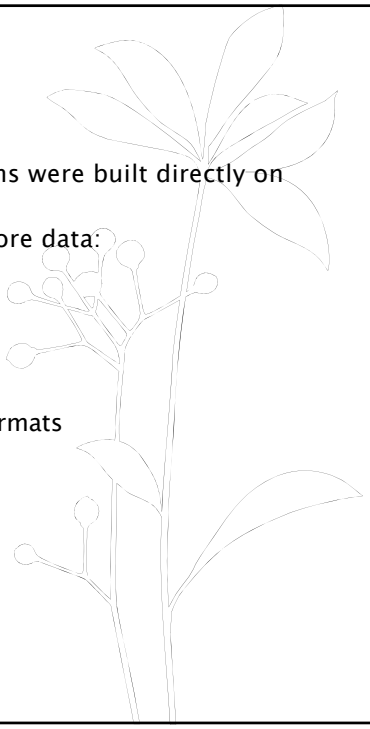
- A large collection of data
- Logically related
- Models some real-world enterprise
- Usually shared among many people

What Is a DBMS?

- A software package designed to store and manage databases.
- Database Applications:
- Databases touch all aspects of our lives

Why Use a DBMS?

- In the early days, database applications were built directly on top of file systems
- Drawbacks of using file systems to store data:
  - Data redundancy and inconsistency
  - Difficulty in accessing data
  - Data isolation — multiple files and formats
  - Integrity problems



Why Use a DBMS? (cont')

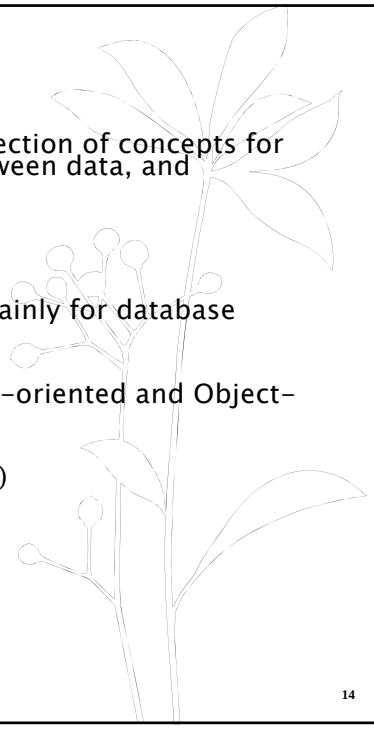
- Drawbacks of using file systems (cont.)
  - Atomicity of updates
  - Concurrent access by multiple users
  - Security problems
- Database management systems offer solutions to all the above problems

Database Languages

- DDL (Data Definition Language)
- DML (Data Manipulation Language)
  - Two classes of DML languages
    - **Procedural**
    - **Declarative (nonprocedural)**
  - SQL is the most widely used query language

Data Models

- A *data model* is an integrated collection of concepts for describing data, relationships between data, and constraints on the data.
- Relational model
- Entity-Relationship data model (mainly for database design)
- Object-based data data models (Object-oriented and Object-relational)
- Semi-structured data model (XML)
- Other older models



Relational Data Model

- *Entities*:
- *Relationship*:
- A *table* is used to represent an entity or a relationship.
- A *schema* is a description of data in terms of a given data model.
- An example of schema and table

Relational Data Model

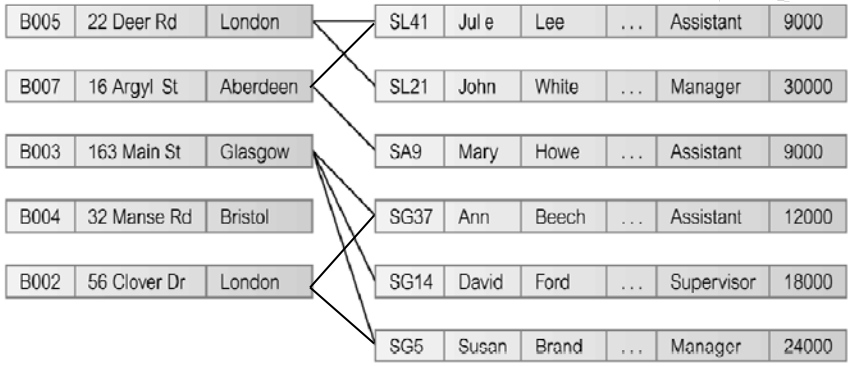
Branch

branchNo	street	city	postCode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

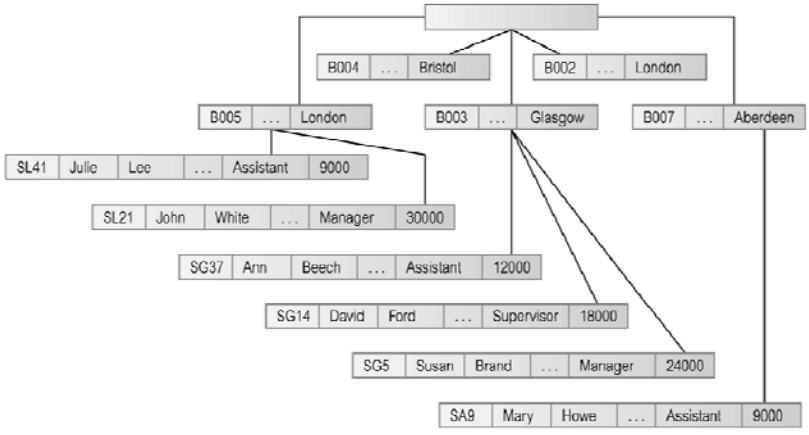
Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

Network Data Model

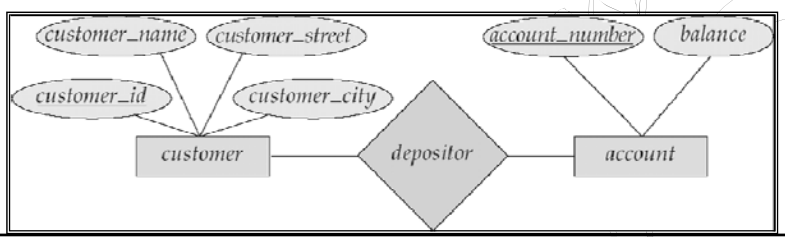


Hierarchical Data Model



The Entity-Relationship Model

- Models an enterprise as a collection of *entities* and *relationships*
- Represented diagrammatically by an *entity-relationship diagram:*

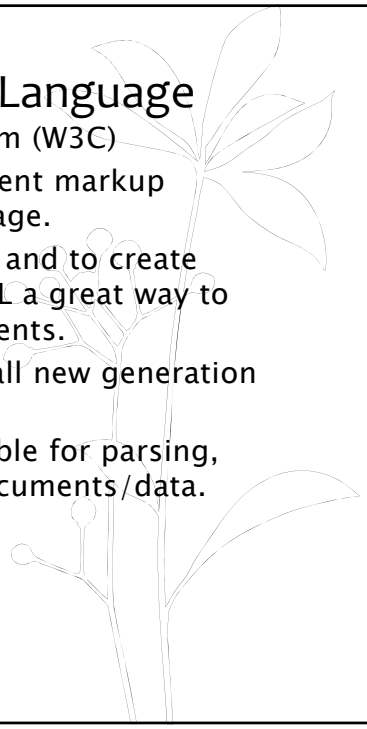


Object-Relational Data Models

- Extend the relational data model by including object orientation and constructs to deal with added data types.
- Allow attributes of tuples to have complex types, including non-atomic values such as nested relations.
- Preserve relational foundations, in particular the declarative access to data, while extending modeling power.
- Provide upward compatibility with existing relational languages.

XML: Extensible Markup Language

- Defined by the WWW Consortium (W3C)
- Originally intended as a document markup language not a database language.
- The ability to specify new tags, and to create nested tag structures made XML a great way to exchange **data**, not just documents.
- XML has become the basis for all new generation data interchange formats.
- A wide variety of tools is available for parsing, browsing and querying XML documents/data.



Database Users

- **Users** are differentiated by the way they expect to interact with the system.
  - **End users**
  - **Application programmers**
  - **Database Administrator (DBA)**

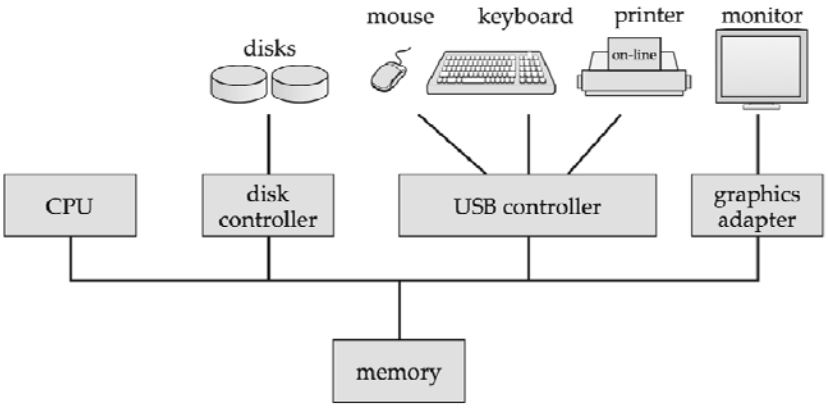
Database Architecture

The architecture of a database systems is greatly influenced by the underlying computer system on which the database is running:

- Centralized
- Client-server
- Parallel (multi-processor)
- Distributed

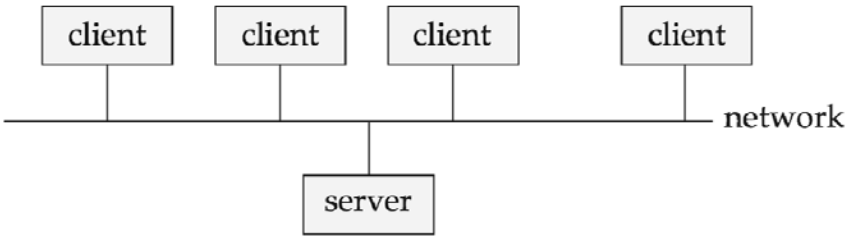
A Centralized Computer System

Run on a single computer system and do not interact with other computer systems.



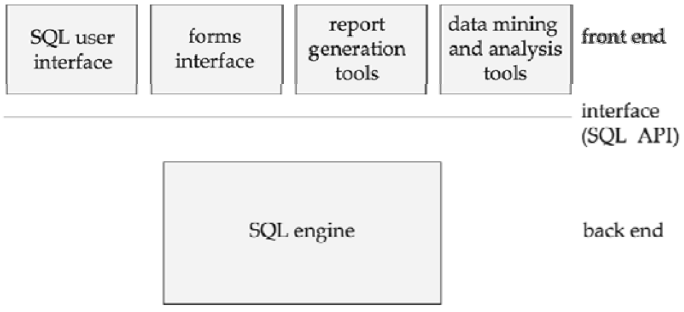
Client-Server Systems

Server systems satisfy requests generated at *m* client systems, whose general structure is shown below:



Client-Server Systems (Cont.)

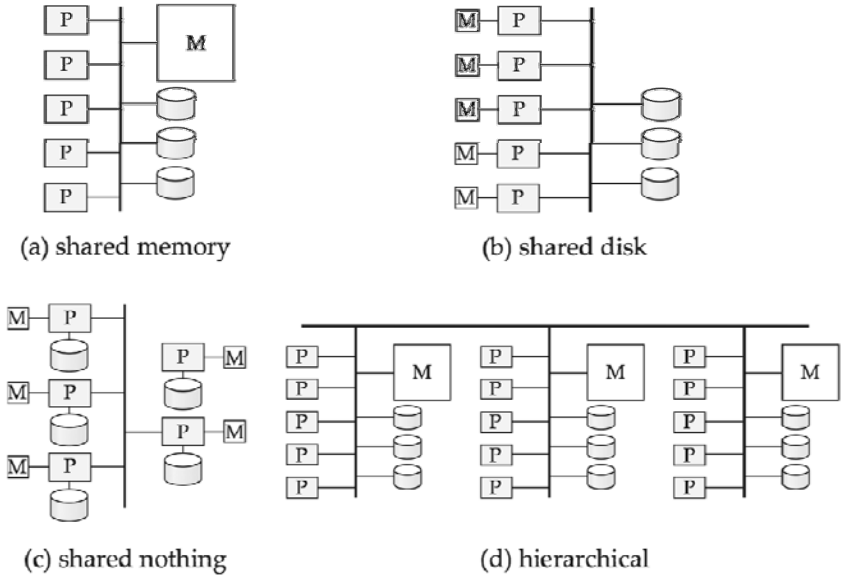
- Database functionality can be divided into:
  - **Back-end**:
  - **Front-end**:
- The interface between the front-end and the back-end is through SQL or through an application program interface.



Parallel Database Architectures

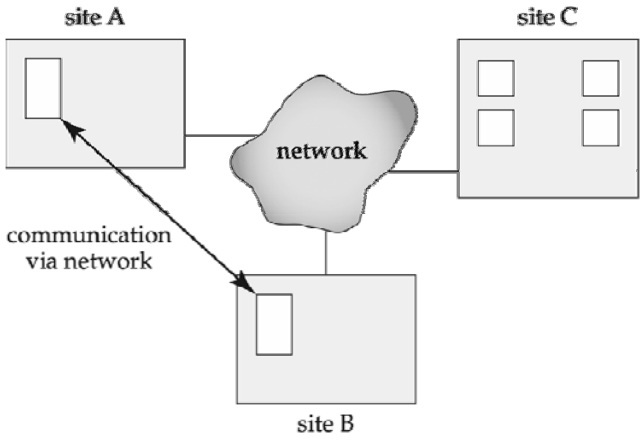
- **Shared memory**
- **Shared disk**
- **Shared nothing**
- **Hierarchical**

Parallel Database Architectures



Distributed Systems

- Data spread over multiple machines (also referred to as **sites** or **nodes**).
- Network interconnects the machines
- Data shared by users on multiple machines



History of SQL

- **SQL** (**Structured Query Language**) is a programming language designed for managing data in relational database management systems (RDBMS).
- SQL was one of the first commercial languages for Edgar F. Codd's relational model, as described in his influential 1970 paper, "A Relational Model of Data for Large Shared Data Banks".
- SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standards (ISO) in 1987. (SQL-86/87 or SQL1)
- SQL-89 (SQL1.1)
- SQL-92 (SQL2
- SQL-99 (SQL3)
- SQL-2003
- SQL-2008