Lecture 2 – Database Concepts

INLS 523
Database Systems I
Spring 2013
Rob Capra
Databases and Users

EN6 Figure 1.1
Data Abstraction and Models

• Databases support data abstraction

• What is data abstraction?
  – Hide unnecessary details
  – Focus on important aspects
  – Different users can see different views

• There is a common underlying data model
Data Model

• Specifies the **structure** of the database

• Also may provide pre-defined **operations**
  
  Compute_hours_worked()
  
  Calculate_sales_tax()

• Levels of data models
  
  – High-level (conceptual)
  
  – Low-level (physical)
Schemas

• The **database** is separate and different from the **description** of the database

• The **database** is the data

• The **description** of the database is the schema

• Schemas
  – Don’t change often
  – Often have diagrams to represent them (which may be abstractions)
  – May include data types and constraints
  – Can be used to **define** a database
Example Schema – MediaWiki

• http://www.mediawiki.org/wiki/Manual:Database_layout
Databases

• The database itself
  – Contains data which may change very frequently
  – Has state
    • Meaning that at any given time, it contains specific data
    • The initial state of a new DB is probably empty
    • Creating, updating, and deleting records changes the state
    • Each CRUD operation moves the DB to a new state
    • DMBS often tries to keep DB in a valid state
Three-Schema Architecture

EN6 Figure 2.2
Data Independence

• “the capacity to change the schema at one level of a database system without having to change the schema at the next higher level” (EN6, p??)

• Logical data independence
  – Can change conceptual schema without changing external
    • e.g. extend the DB, remove unneeded parts

• Physical data independence
  – Can change physical schema without changing conceptual or external
    • Change how DB files are stored, access paths, etc.
Database Languages

• EN6 describes DDL, SDL, VDL, DML
• Two main to focus on are
  o DDL – Data Definition Language
    • Statements used to define the schema
      create table employee (eid int, name varchar(40));
  o DML – Data Manipulation Language
    • Statements to search, insert, update, delete data in the database
      insert into employee values (1234, ‘Jane Smith’);
• In many modern DBMS, DDL and DML are part of one integrated language
• The SQL we will use supports DDL and DML
  o SQL = structured query language
Database Interfaces

• Menu-based (web clients/browsing)
• Forms-based, forms languages
• Graphical User Interfaces
• Natural Language Interfaces
• Spoken Input and Output
• Parametric Users
  – Optimized UI for small set of repeated queries/actions
• DB Administrator (DBA) Interfaces
DBMS Components

EN6, Fig 2.3
DB System Utilities

• Bulk data loader
• Backup utilities
  – Often called “dumping” the database
  – Despite the name, dumping does not delete the data
  – Dump often outputs SQL commands that can be used to re-create and re-populate the data.
• Storage re-organization tools
• Performance monitoring
Development Environments

• Tools to aid in the development of databases
• Kind of like going from a text editor to a word processor
MySQL command line interface
Example of an Integrated Development Environment
DB Architectures

EN6, Fig 2.5

EN6, Fig 2.7
Classification of DMBS

• Data model
  – Relational, object, hierarchical, network, XML

• Number of users
  – Single-user systems vs. multi-user systems

• Number of sites
  – Centralized: DBMS and DB at a single computer site
  – Distributed: DBMS and DB distributed across many sites
  – Homogeneous & heterogeneous DDBMSs
    • Do all sites use the same DMBS software?
    • Federated: loosely coupled, local autonomy

• Cost
  – Open source, licensed, components

• General purpose vs. special purpose DBs
“NoSQL”

• Document Store
  – CouchDB
  – SimpleDB
  – MongoDB

• Graph
  – Neo4j

• Key-Value
  – Cassandra
  – BigTable

http://en.wikipedia.org/wiki/NoSQL
Embedded DBMS

http://developer.android.com/about/versions/index.html
Exercise: Airline Flight Database

• What types of users are there?
• What types of interfaces does each set of users need?
• Outline the main components of a schema
• What architecture should we use? Why?
• What are some constraints on our data?
• How does data independence matter in this DB?