Cal Poly Spring 2013	**CPE/CSC 365** Introduction to Database Systems	Alexander Dekhtyar Eriq Augustine

Overview of the Course
Databases and DBMS in a nutshell

Data

Views of data:

- conceptual: captures relationships in the data. (CPE 366)
- logical: captures the format of the data as understood by DBMS. (CPE 365)
- physical: represents the exact way the data is stored and accessed by DBMS. (CPE 468)

Logical View of data

Data models

- historical
  - *network*
  - *hierarchical*
- modern
  - ***relational***
  - *object-oriented*
  - *object-relational*
- emerging
  - *key-value/non-relational* ("NoSQL")
  - *semistructured* (XML, JSON)

Steps of Database Design

Step 1: Requirements analysis. Collect information from customer about

- data;
- desired features of the database;
- information needs.

Step 2: Conceptual Database Design. Develop high-level description of data, describe constraints.

- High-level design: often done using Entity-Relationship diagrams (E-R diagrams).

Step 3: Logical Database Design. Select a DBMS, convert high-level design into (relational) database design (*database schema*) in Data Definition Language (DDL) of the DBMS.

- DDL for relational databases is a part of SQL.

Steps 1 − 3 are main steps in database design. Thee more steps, *enhance* the Logical design.

Step 4: Schema Refinement. Logical database design is analyzed and (potentially) improved.

- Goal of schema refinement: have database schema in one of **normal forms**.

Step 5: Physical Database Design. Tailor the database schema to expected workloads (queries, information needs).

- Choose *indexes*.
- *Tune* database design.

Step 6: Security Design. Identify user groups, information (parts of the database) to be made available to different user groups. Represent security information in DDL.

- SQL has some mechanisms to maintain security of the data.

Querying Databases

For **relational databases**, the main query language is

SQL = **S**tructured **Q**uery **L**anguage.

SQL consists of the following:

- SQL DDL - data definition language: define/manipulate *relational schemas*.
- SQL DML - data management language: add/update/delete data in the database.
- SQL QL - query language: request information from the database.
- SQL/PL - programming language: create complex sequences of instructions for DBMS to execute.

History of SQL:

- SQL'77 — first standard, many DBMS used their version of SQL even after it.
- SQL-92 — first **real** standard. We will mostly study SQL-92.
- SQL:1999 — new standard. Includes new features for different types of databases, expands to cover object-relational databases.

Popular SQL Server Implementations:

- Oracle SQL — proprietary and very expensive.
- MySQL (Oracle) — open source (for now), acquired by Oracle in 2010.
  - MariaDB — open source fork of MySQL.
  - Percona — open source fork of MySQL.
- PostgreSQL — open source. Typically considered slower than MySQL, but more "correct".
- SQLite — open source. Uses flat files and no server.

DBMS

The purpose of DBMS:

- Allow users to **describe** data format (*database schema*).
- **Store** very large amounts of data.
- **Answer** to user information needs (*queries*).
- **Control** access to data from multiple users.

DBMS organization in a nutshell

- Query Parser : parses the incoming SQL statement.
- Query Compiler : recognizes the query, finds the best way to execute it.
  - Query Translator
  - Logical Plan Generator
  - Physical Plan Generator
- Query operations: used in query plans, execute queries.
- Buffer Manager : efficiently handles disk I/O.
- Transaction Manager: schedules data accesses for different users.
- Recovery Manager: ensures that database state can be recovered after severe crashes.

In This Course

- We study **relational data model**.
- We learn **SQL** (including SQL/PL).
- We use **MySQL (MariaDB)** DBMS.
- We use MySQL's default client to access databases interactively or in a batch mode.
- We use **JDBC** (Java Database Connectivity) API to build database applications programs in Java.