**CSCI 4333.1 Classroom Notes and Demonstrations**

8/20/2025

**4. Introduction**

* *Persistent* data is the backbone of many applications.

1. [SEE-I](https://en.everybodywiki.com/SEE-I): State, Elaborate, Exemplify
   1. Persistent data: data does not change.
   2. Data that is stored.
   3. Data stored in the same type/format.
   4. E.g., persistent device: hd, ssd, magnetic tape.
   5. Transient (non-permanent): RAM, register, etc.
   6. Files can be used to store persistent data.

* Three main choices of storing persistent data:
  1. Files
  2. Databases: focus of this course.
  3. Cloud-based storage and database.
* Some advantages of DBMS (according to Ricardo, the optional textbook of this class):
  1. Sharing of data
  2. Control of redundancy
  3. Data consistency: data should satisfy the constraints, requirements of the problems. DBMS provides many features for specifying and enforcing data constraints and requirements for consistency.

John Young’s dob: 1/2/2000

32/31/2025

a/b9/25

1/2/1800

* 1. Improved data standards
  2. Better data security
  3. Improved data integrity
  4. Balance of conflicting requirements
  5. Faster development of new applications
  6. Better data accessibility
  7. Economy of scale: efficiency achieved because of the large size of the resources.
  8. Better scaling
  9. More control over concurrency
  10. Better backup and recovery procedures
* How do we *make sense* of these 12 different advantages?
  1. Different textbooks may have different collections of the advantages of DBMS because of different classifications.
  2. No need to memorize them.
  3. Better to assimilate them and construct your own list.
  4. Make your own notes. Use [SEE-I](https://en.everybodywiki.com/SEE-I) (In your own words, state, elaborate, and exemplify with examples, and illustrate the concept.)
     + However, do not overuse metaphors.
  5. Learning through documentation, communications, and teaching.
* What are some disadvantages of DBMS?
  1. Complexity
  2. Cost
  3. Learning curve
  4. Possible single points of failure and bottleneck

**3.1 MySQL Server Setup**

We will use MariaDB that is a part of XAMPP. Do not recommend installing standalone MySQL.

[1] Install XAMPP, which contains many server software configured to work together for development purposes. For XAMPP, we will use MySQL/Maria DB and Apache (Web server).

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Recommended to install XAMPP in the*top* level: c:\xampp (likely the default).

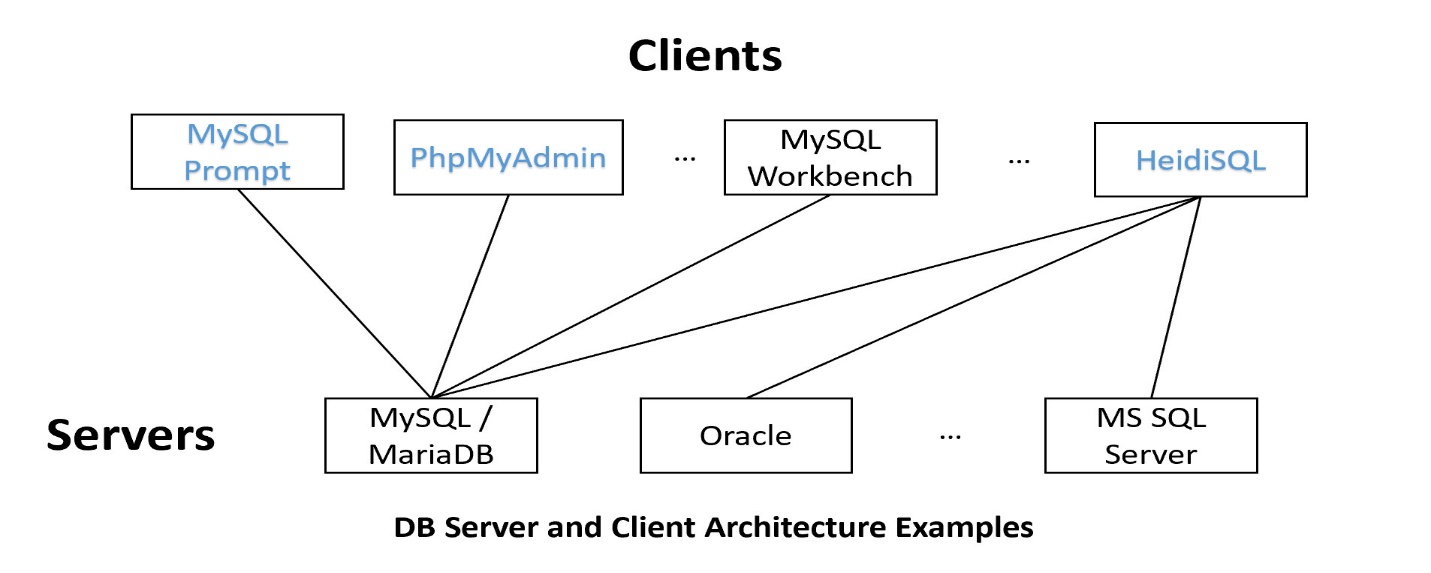
A screenshot of a computer

AI-generated content may be incorrect.

1. Set up development accounts immediately using *phpMyAdmin* after installation.
2. Change the root password (optional but recommended): a secure step that requires tinkering.
3. To ensure that PHPMyAdmin will work on a new admin account (optional):
   1. Use PHPMyAdmin to create a new admin account, e.g., "frog\_ad", with the password "a\_new\_prince" for both hostname '%' and 'localhost'
   2. PhpMyAdmin uses the default root account (with no initial password) via localhost.
   3. Thus, you will need to supply the new username and password to start up PhpMyAdmin by editing the file c:\xampp\phpMyAdmin\config.inc.php, search change the line to, for example:
      1. $cfg['Servers'][$i]['user'] = 'frog\_ad';
      2. $cfg['Servers'][$i]['password'] = 'a\_new\_prince';

**3.2 MySQL Clients Setup:**

It is common to use multiple clients to connect to a backend database server. In this course, we will use three clients in our classes. You may use your own favorite clients (e.g., MySQL Workbench). However, I may not be as helpful in these clients.



**5. A Simple Introduction to the Relational Model**

* Relational databases are the most popular databases: <https://db-engines.com/en/ranking>. It is based on the relational model.
* There are many other data models.
* In layman's term: A *table* (relation) is the basic unit of a relational database.
* A table is composed of many *rows* (tuples).
* Each row has many *column* (attribute) values.

A screenshot of a computer

AI-generated content may be incorrect.

* A primary key is roughly a *minimal* set of columns in a table that*uniquely identify* a row.
* Two tables can be related to each other by *foreign keys*. A foreign key is roughly a column in a table in which its value must be equal to the referenced value of the primary key in another table (called the paren or referenced table).
* Relational DBMS is the most popular DBMS. Examples:
  + DB-engine ranking: <https://db-engines.com/en/ranking>
  + Top 10 DBMS in Data Science: <https://towardsdatascience.com/top-10-databases-to-use-in-2021-d7e6a85402ba>
* SQL is the 'glue' in many DB systems.