CISC 3140 (CIS 20.2) Design & Implementation of Software Application II

Instructor : M. Meyer
Email Address: meyer@sci.brooklyn.cuny.edu
Course Page: http://www.sci.brooklyn.cuny.edu/~meyer/
Contents

• Database Management Systems
  • Early Database Models (Hierarchical Databases)
  • Codd's Relational Database Model & DBMS
  • DBMS's and RDBM's
  • Why DBMS's?
• Data Models
• Database Models
• Relational Database Systems
  • Entity-Relationship Model
  • Normalization
  • SQL
  • Help with MySQL
Beyond Data Structures

• Complex data storage and retrieval systems have been around since the time of the ancient Greeks (library catalogs, literature indexes and appendices).
• Automated data retrieval systems, that DID NOT depend on computers were in use as early as 1931 (referential links embedded within Microfiche documents).
• Within computers arrays, stacks, queues, heaps, simple structures and class derived objects are useful ways to store small collections of data.
• However, these options are limited in their scope and retrieval ability and inappropriate for very large collections of data.
Hierarchical Databases

• Prior to 1970, computer data storage and retrieval systems relied to a large extent on hierarchical and network *database models*.
• These models required specific knowledge of how data were stored and any modification to the data storage model required equivalent changes in the retrieval system.
• You are familiar with hierarchical storage systems if you have ever looked a phone book.
Hierarchical Databases Continued

- Hierarchical database systems can be fast and useful if the types of query desired are limited.
  - (Easy) Find Loyd, Harold K
  - (Hard) Find all 92004 Zipcodes
- In order to quickly and reliably answer the zipcode query, we would probably want to duplicate our entire phonebook, and order it by Zipcodes (redundant information storage).
- NOTE: XML Database Systems (we will discuss later in the semester) are fundamentally hierarchical in nature.
Codd & Relational Database Model

- In 1969, Dr. Edgar F. Codd published the first in a series of highly original papers describing what would become the Relational Model of *Database* systems.
- In the relational database model, data are typically accessed and viewed in a table format, but this format does not necessarily bear any resemblance to how the data are actually stored in a computer’s hardware.
- In order to access and retrieve information in a relational database, a user is dependant upon a Database Management System.
- We will describe the relational model of data storage in greater detail in a few slides.
Database Management Systems

- A Database Management System (DBMS) is a software program that enables the creation and management of databases.
- Most of today's database systems are referred to as a Relational Database Management System (RDBMS), because the use the Relational database model of data storage and retrieval.
- Some of the more popular RDBMS's in use today include:
  - Microsoft Access
  - Microsoft SQL Server
  - MySQL
  - Oracle
Why DBMS's

- Before DBMS there were file processing systems that could search for information stored in files:
- Problems with these file processing systems included:
  1. Data redundancy and inconsistency
     - Same information may be duplicated in several places.
     - All copies may not be updated properly.
  2. Difficulty in accessing data
     - May have to write a new application program to satisfy an unusual request.
  3. Data isolation
     - Different data might be stored in different files.
     - Different data might be stored in different formats.
     - Together these difference could make it very difficult to write new application programs to use a specific collection of data. Example: Find me all the people with last names that begin with M in the 90224 zipcode.
Why DBMS's (continued)

• Supporting multiple users
  ▫ Want concurrency for faster response time.
  ▫ Need protection for concurrent updates.
    • i.e. two customers withdrawing funds from the same account at the same time – account has $500 in it, and they withdraw $100 and $50. The result could be $350, $400 or $450 if no protection.

• Security problems
  ▫ Every user of the system should be able to access only the data they are permitted to see.
  ▫ File processing approach relies on the application to handle this

• Integrity problems
  ▫ Data may be required to satisfy constraints.
    • i.e. phone number must have 7 digits
  ▫ Difficult to enforce or to change constraints with the file-processing approach.
Data Models

• Don't confuse data models and database models!
• A data model is a methodology for describing data, data relationships, data semantics and data constraints. There are three different groups:
  1. **Physical Data Models** – used to describe data at the lowest level (physical level of the hardware)
  2. **Logical Data Models** – describes the semantics such as descriptions of tables and columns or object oriented classes or XML tags in use.
  3. **Conceptual Data Models** – describes data at very high conceptual and view levels (i.e. E-R Model)
Data Abstraction

• Data models and database systems hide (abstract) certain details of how data is stored and created and maintained. This is a good thing! Complexity should be hidden from database users.

• Our 3 data models provide 3 levels of abstraction:
  ▫ **Physical Level** – Lowest Level
    • How the data are stored. (i.e. index, B-tree, hashing)
  ▫ **Logical Level**
    • Describes what data are stored. (table descriptions etc, done using DDLs)
    • Describes the relationships among data.
    • Database administrator (DBA) level.
  ▫ **Conceptual and/or View Level** – Highest Level
    • Describes part of the database for a particular group of users.
    • There can be many different views of a database
    • User level
Entity-Relationship Model

- An entity-relationship model (ERM) is a data model (abstract-conceptual) used in software engineering to represent structured data.
- The entity-relationship model perceives a domain (world) as a collection of basic objects (entities) and relationships among these objects.
  - An **entity** is a distinguishable object that exists.
  - Entity **attributes** are used to describe entities.
  - A **relationship** is an association among entities.
    - i.e. A student_class relationship associates a student with his or her classes
  - An essential element of the E-R diagram is the mapping **cardinalities**, which express the number of entities to which another entity can be associated via a relationship set.
Basic ER Diagram Symbols

- Entity
- Attribute
- Relationship
- Links
Examples of ER Diagrams
Database Models

• Don't confuse data models and database models!
• A database model is a theory or specification describing how a database is structured and used. Several such models have been suggested. Common models include:
  1. **Hierarchical model**: Data is organized into a tree-like structure, with nesting, and a sort field to keep the records in a particular order in each same-level list.
  2. **Network model**: Organizes data into records and sets. Records contain fields, and sets define one-to-many relationships between records: one owner, many members. (OO Model)
  3. **Relational model**: Database model based on first-order predicate logic. Describes database as a collection of predicates over a finite set of predicate variables.
Database Models (cont)

Hierarchical Database Model

Network Database Model
Database Models (cont)

Relational Database Model

Relation variable (Table name)

Attribute (Column) \{unordered\}

Heading

Body

Relation (Table)

Tuple (Row) \{unordered\}
Relational Database Model

• The dominate model for storing and accessing data in database systems for the past forty years has been the relational database model.
• Relational database management systems (RDBMS) like Oracle and MS SQL Server, are based on Codd's work and RDBMS's are the most common type of database management system (DBMS) in use today.
• In the relational model, data are typically accessed and viewed in a table format, but this format does not necessarily bear any resemblance to how the data are actually stored in a computer’s hardware (abstraction at physical, logical, conceptual levels).
Microsoft Access
SQL Server
Relational Database Tables

Source Table (People)

<table>
<thead>
<tr>
<th>First</th>
<th>Last</th>
<th>Address</th>
<th>City, State</th>
<th>Zip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffet</td>
<td>Jimmy</td>
<td>Somewhere on the Beach</td>
<td>Key West, FL</td>
<td>33040</td>
</tr>
<tr>
<td>Obama</td>
<td>Barrack</td>
<td>1000 Pennsylvania Ave</td>
<td>Washington, DC</td>
<td>20500</td>
</tr>
<tr>
<td>Cartman</td>
<td>Eric</td>
<td>84 Bigbowed Way</td>
<td>South Park, CO</td>
<td>84214</td>
</tr>
<tr>
<td>Crockett</td>
<td>Davey</td>
<td>The Alamo</td>
<td>San Antonio, TX</td>
<td>78210</td>
</tr>
<tr>
<td>Doe</td>
<td>Jane</td>
<td>821 Zimbabwe Ave</td>
<td>Washington, DC</td>
<td>20021</td>
</tr>
<tr>
<td>Gates</td>
<td>Bill</td>
<td>1 Microsoft Way</td>
<td>Redmond, WA</td>
<td>98052</td>
</tr>
<tr>
<td>Jefferson</td>
<td>George</td>
<td>194 Deelux Apartments</td>
<td>In the Sky, NY</td>
<td>10041</td>
</tr>
<tr>
<td>Kong</td>
<td>King</td>
<td>Empire State Building</td>
<td>New York, NY</td>
<td>10118</td>
</tr>
<tr>
<td>Munster</td>
<td>Herman</td>
<td>1313 Mockingbird Lane</td>
<td>Fargo, ND</td>
<td>58102</td>
</tr>
<tr>
<td>Rockne</td>
<td>Knute</td>
<td>146 Keenan Hall</td>
<td>Notre Dame, IN</td>
<td>46556</td>
</tr>
<tr>
<td>Simpson</td>
<td>Homer</td>
<td>742 Evergreen Terrace</td>
<td>Springfield, US</td>
<td>12345</td>
</tr>
<tr>
<td>Smith</td>
<td>Bob</td>
<td>12 Main Street</td>
<td>Anytown, IN</td>
<td>46001</td>
</tr>
<tr>
<td>Obama</td>
<td>Michelle</td>
<td>1600 Pennsylvania Ave</td>
<td>Washington, DC</td>
<td>20500</td>
</tr>
</tbody>
</table>

Query Result Table (First, Last, Zip = "20500")

<table>
<thead>
<tr>
<th>First</th>
<th>Last</th>
<th>Zip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obama</td>
<td>Barrack</td>
<td>20500</td>
</tr>
<tr>
<td>Obama</td>
<td>Michelle</td>
<td>20500</td>
</tr>
</tbody>
</table>
Accessing Multiple Tables

- Tables in RDBMS systems contain rows of related information.
- Each row is called a tuple or record.
- The columns denote individual fields or attributes in the records.
- Tables in an RDBMS can be linked together based on "key fields". This can create relationships between tables.
- In an RDBMS it is possible to construct queries that pull information from many different tables.
Table Relationships
Normalization

• *Normalization* is the process of simplifying the design of a database so that it achieves the optimum structure.

• In an RDBMS (for the most part) this means changing the number and content of or Tables.
  
  • Example: One of our goals in using a DBMS was to reduce redundancy.
  
  • In RDBMS's the fact that we can create and link multiple tables means that we can (in theory) eliminate redundant information.
Benefits of Normalization

1. Reduce redundancy
2. Creates a flexible model (easily add new attributes or relationships etc)
3. Allows for the continual enforcement of integrity constraints
4. Most Importantly: Prevent update, insertion, and deletion anomalies

There are several different types of normalization!
First Normal Form (1NF)

• A relation is in first normal form if, and only if, every attribute is single-valued (no sets, lists or groups)

<table>
<thead>
<tr>
<th>Student Table</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>StuId</strong></td>
</tr>
<tr>
<td>1001</td>
</tr>
<tr>
<td>1002</td>
</tr>
<tr>
<td>1003</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Second Normal Form (2NF)

- A relation is in second normal form if, and only if, it is in first normal form and all non-key attributes are fully functionally dependent on the key.
- In a table a key attribute must be unique!
- In this table the key is actually composed of 2 keys, ClasNo and StuID.
- Only Grade is completely dependent on that key!

<table>
<thead>
<tr>
<th>ClassNo</th>
<th>StuId</th>
<th>LastName</th>
<th>Schedule</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>ART50</td>
<td>1001</td>
<td>Simpson</td>
<td>MWF9</td>
<td>A</td>
</tr>
<tr>
<td>CSC11</td>
<td>1002</td>
<td>Simpson</td>
<td>TTH2</td>
<td>B+</td>
</tr>
<tr>
<td>HIS20</td>
<td>1003</td>
<td>Bailey</td>
<td>MWF11</td>
<td>B</td>
</tr>
</tbody>
</table>
Third Normal Form (3NF)

- A relation is in third normal form if, whenever a non trivial functional dependency $X$ implies $A$ exists, then either $X$ is a superkey or $A$ is a member of some candidate key.
- There should be no transitive dependencies
- Another way to look at it is that there should be no SUMs or calculated results in a table. Only generate those fields when you need them (in a view or as a result of a query).

<table>
<thead>
<tr>
<th>StuId</th>
<th>LastName</th>
<th>Credits</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td>Simpson</td>
<td>90</td>
<td>Senior</td>
</tr>
<tr>
<td>1002</td>
<td>Simpson</td>
<td>15</td>
<td>Freshman</td>
</tr>
<tr>
<td>1003</td>
<td>Bailey</td>
<td>63</td>
<td>Junior</td>
</tr>
</tbody>
</table>
Higher Normal Forms

• After Codd defined the original set of normal forms it was discovered that Third Normal Form, as originally defined, had certain inadequacies. This led to several higher normal forms, including the Boyce/Codd, Fourth and Fifth Normal Forms.
• We will not be covering these higher normal forms, instead, several points are worth noting here:
  1. Every higher normal form is a superset of all lower forms. Thus, if your design is in Third Normal Form, it is also in 1NF and 2NF.
  2. If you've normalized your database to 3NF, you've likely also achieved Boyce/Codd Normal Form (maybe even 4NF or 5NF).
  3. To quote C.J. Date, the principles of database design are "nothing more than formalized common sense."
  4. Database design is more art than science.
SQL = Structured Query Language

- SQL is used to communicate with a database, it is the standard language for RDBMS's.
- SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database.
- Common RDBMS that use SQL:
  - Oracle
  - Sybase
  - Microsoft SQL Server
  - Access
  - Ingres
- **NOTE:** Even though most database systems use SQL, the syntax used in SQL varies from system to system and many systems also have their own additional proprietary extensions that are usually only used on their system.
A Data Definition Language (DDL) is used to specify a database schema (how the database is defined and constructed).

DDL statements are compiled, resulting in a set of tables stored in a special file called a data dictionary or data directory.

The data directory contains metadata (data about data).

Most important SQL DDL commands:
- CREATE TABLE
- CREATE INDEX
- ALTER TABLE
- RENAME TABLE
- DROP TABLE
- DROP INDEX

NOTE: An index is a copy of a part of a database table (linked back to the source table) used to help speed up queries. (ZIP example).
SQL -> Data Manipulation Language (DML)

- A Data Manipulation Language is used for:
  - retrieval of information from the database
  - insertion of new information into the database
  - deletion of information in the database
  - modification of information in the database

- SQL DML commands
  - SELECT
  - INSERT
  - DELETE
  - UPDATE
Don’t Panic

1. We will have a lab on this and you are NOT expected to become SQL and/or RDMS experts in this class.

2. There are many, many wonderful places to get help online (for SQL #1, and MySQL #2):
   1. [http://www.w3schools.com/sql/](http://www.w3schools.com/sql/)
      - Fantastic resource with tutorials, & Tests.
   2. [http://dev.mysql.com/](http://dev.mysql.com/)
      1. Click Documentation tab
      2. Click MySQL 5.5 Reference Manual (on left hand side)
      3. Click 12. SQL statement syntax (on right hand side)
Take a deep breath.
You got this!
Getting Started on the Lab

- Your instructor should provide you with the following pieces of information:
  - IP Address of Server you will use
  - Username
  - Password

- You can then use any secure shell client to access the LAMP server we will be working on for the rest of the semester.