



CENTER FOR TEXTUAL STUDIES AND DIGITAL HUMANITIES

402 - Introduction to Digital Humanities Design and Programming

Spring Semester 2016

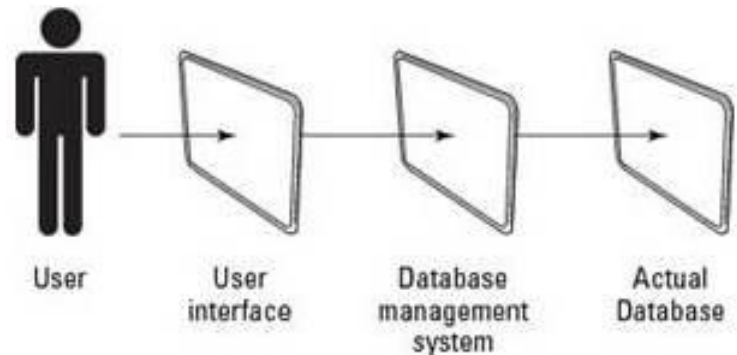
Week 2

Databases - Data mining

- large chunks of data often referred to as 'data warehouses'
- data mining simply looks at separate databases to find information that's not obvious otherwise
- link databases and mine the data for combined patterns, inferences...
- data mining is useful for helping us to find hidden data in seemingly innocuous databases
- criminal profiling, political activists, genetic disposition...

Databases - Data programming

- if we want to store large amounts of data and manipulate it in different ways
- three parts of a database program to consider
 - user interface
 - database management system
 - actual information stored in a database



- UI should allow a user to use and manipulate the data without having to know how the data is stored or how to write commands to manipulate the data
- commands for manipulating data may include printing, searching, sorting...
- customised management layer with chosen programming language

Databases - SQL

- structured query language
- a way of accessing and manipulating a database

options and tools

- create a new database, rename...
- create tables in a database
- query a specified database and retrieve data
- insert, update, delete records within a database
- set privileges and permissions
- create groupings, views, orders...

there's a lot we can do with an SQL database...

Databases - SQL

generically we can consider SQL in two parts

1. Data Manipulation

- select
- insert into
- update
- delete

2. Data Definition

- create database
- drop database
- create table
- drop table
- create index
- drop index

and so on...

Databases - MySQL

creating DBs and tables:

- names (formally known as *identifiers*)
 - should be clear, meaningful, and 'easy to type' in queries etc
 - should only contain letters, numbers, and underscore (no spaces)
 - should not be the same as an existing keyword (eg: SQL term etc)
 - column names are case-insensitive (often best to use lowercase throughout)
 - unique with DB scope (DB realm)
 - eg: a table cannot have two columns with the same name
 - a DB cannot have two tables with the same name

to name DB data:

- determine the DB name
- determine the table name
- determine the column name in the table

Databases - MySQL

column types:

- determine data type for a column
 - needs to be explicitly specified
 - specifies type of data that can be stored in the column
- three primary types
 - Text (strings..)
 - Numbers
 - Dates and Times
- each primary type has a number of variants
 - Text (CHAR, VARCHAR, TEXT...)
 - Numbers (INT, FLOAT, DOUBLE...)
 - Dates and Times (DATE, DATETIME, TIMESTAMP...)
- many types can take an optional 'length'
 - eg: INT(10), VARCHAR(150)
 - excessive length will be truncated
 - lengths for numeric types determines output & not input length
- TIMESTAMP automatically set to current date and time
- other variants are available such as BLOB...

Databases - MySQL

Choosing a column type

- decide whether column should store text, number, or date/time
 - often an obvious decision
 - date/time can be stored either specifically as a date or as text or as a number
 - further manipulation and use dependent on type
- select subtype for data per column
 - userid = int (10)
 - username = varchar(200)
 - usercreated = timestamp
- CHAR or VARCHAR?
 - CHAR always uses specified full string length
 - VARCHAR only uses space required for stored data
 - speed of fixed size CHAR vs flexibility of VARCHAR

Databases - MySQL

Choosing any other column properties

- every column can be set to NOT NULL
 - NULL = no value
- default value for any column can also be set
 - if no default value, and no value set, NULL will be set for a new record
 - NOT NULL set, and no value set, error will occur for a new record
- UNSIGNED limits the stored data to positive numbers and zero
 - used with number types
- ZEROFILL pads extra space with zeros
 - used with number types
 - automatically UNSIGNED

Databases - MySQL

Finishing your columns

- define and identify your PRIMARY KEY
 - nearly always a numerical value & must always have a value
 - unique way to refer to a given record
 - eg: userid
 - can be defined arbitrarily
 - critical for referencing and finding a record
- identify those columns that cannot store a NULL value
- always specify a numeric type as UNSIGNED if a negative is either not necessary or illegal
- if applicable establish the default value for a column
- check and double check that your table can store the required data

Databases - MySQL

Finishing your columns

- INDEX basically maintains a record of the values in a specified column of a given table
 - allows FULLTEXT search etc (MyISAM)
 - improves performance when retrieving records
 - slightly reduces performance when inserting or updating records
- KEY (Primary and Foreign)
 - each table should have one primary key
 - primary key in one table often linked as a foreign key in another
 - foreign key constraints, cascading updates and deletes... (InnoDB)
- AUTO_INCREMENT
 - added to a given column
 - automatically increments the column upon new insert by next highest number
 - eg: 1, 2, 3, 4 etc

etc, etc...

Databases - SQL

Example Queries

```
SELECT FirstName from Employees
```

```
SELECT FirstName from Employees WHERE FirstName = 'Emma'
```

```
INSERT INTO Employees VALUES ('Emma', 'Smith', '773-749-9246')
```

```
DELETE FROM Employees WHERE LastName = 'Smith'
```

```
UPDATE Employees SET PhoneNumber = '779-751-9248' WHERE LastName = 'Smith'
```

```
UPDATE Employees SET PhoneNumber = '779-751-9248' WHERE LastName = 'Smith'  
AND FirstName = 'Emma'
```

```
UPDATE Employees SET PhoneNumber = '779-751-9248' WHERE id = 223
```

- DISTINCT

```
SELECT DISTINCT FirstName from Employees
```

- SQL is not case sensitive

- NB: this is not an exhaustive list!

Databases - SQL & PHPMyAdmin

Development processes...

- PHP front-end to manage and control installed MySQL
 - create new database
 - populate DB from scratch or import existing DB
 - create tables
 - add data per table and field
 - perform batch operations on a selected DB
 - manage users and privileges per DB
 - export DB or table for backup...

and on and on...

- effectively provides a front-end, graphical solution for MySQL management
 - terminal/command line is standard interface for MySQL management
- without a 3rd party tool such as PHPMyAdmin

Databases - PHPMYADMIN

Setup

- LAMP, XAMPP, WAMP - [Download](#)
 - 5.6.15 / PHP 5.6.15 version for XAMPP
- Raspberry Pi Setup
 - [further details](#)
- add project directory, add HTML, CSS, JS, PHP etc files
- load phpmyadmin and setup MySQL database