Introduction to course

- What is this course about?
- Let me know if you figure it out 😊

- ROOM CHANGE FOR WED LAB
  - Lab will meet Wednesday 6:10pm in Tompkins 410
**CS 2441: What is it about?**

1. Introduction to database design and implementation
2. Social impact and professional ethics analysis
3. Working in teams and team software development
4. Improving technical communication skills:
   1. Writing in the disciplines (WID)
   2. Presentation and discussion skills

*Course is not just about Database design – you have to learn and participate in the other three course objectives.*

---

**What Is a Database?**

- A very large, **integrated** collection of data.
  - Not arbitrary, unrelated data
- Models real-world **enterprise.**
  - Entities (e.g., students, courses)
  - Relationships (e.g., Cam Tucker is registered in cs2441)
**Why Databases?**

Most CS courses concentrate on code – our interest is managing, manipulating and representing data.

*Warning: this course doesn’t focus on teaching SQL or how to be an Oracle DBA (though it will get you started)*

---

**Why Databases??**

- Information gathering is first step to analysis
  - Huge amount of data can be collected easily
- To effectively analyze data:
  - collect relevant data
  - store in manner amenable to efficient access
  - provide **programming** interface
- Data analysis methods are current emphasis in the market
  - Introduces new problems in privacy
Data Analysis: Examples

- Data Warehousing
- Web search engines
- Data mining (personalization)
- Analytics on social networks
  - Twitter “sentiment analysis”
- Scientific data analysis
- …many more…

Data Analysis: Data Mining

- Data mining: finding ‘hidden patterns’ in data; i.e., patterns and relationships that are not ‘obvious’
- Example: purchasing patterns of supermarket customers
  - Pattern: 40% of Customers who buy beer also buy diapers.
  - How do you use the above pattern/knowledge to improve your marketing strategy??
    - Leave it to the Business Majors to worry about!!
- Data mining is “engine” behind Personalization software
**Databases…?**

- Why the discussion on Data mining etc.?
  - Analysis is important to make informed decisions
  - Efficient analysis requires efficient storage & design
  - Efficient storage & design requires study of DBMS!
    - Data analysis of no use if your data is useless/incomplete?
- Data Mining and other data analysis tools
  - Require database at the backend!!
- And ….- DBMS is basic backbone in Transaction Processing systems!
  - Airline reservations, banking, e-commerce

**Database & Application Development Process**

- Organize and Store relevant data
- Operate on the data
  - Search, aggregate,…
- Present results to user
  - Provide interface
- Analyze data
  - Extract patterns …
Example: PDA

- Information on Lucas’s PDA/Cell-phone

<table>
<thead>
<tr>
<th>Calendar</th>
<th>Event</th>
<th>Day</th>
<th>When</th>
<th>Who</th>
<th>Where</th>
</tr>
</thead>
<tbody>
<tr>
<td>lunch</td>
<td></td>
<td>1/15</td>
<td>1pm</td>
<td>John</td>
<td>TGIF</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Contacts</th>
<th>Who</th>
<th>Phone</th>
<th>Email</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>John</td>
<td>123-4567</td>
<td>john@..</td>
<td>1 main st.</td>
</tr>
</tbody>
</table>

Example:

- What if we want to include contact info on our calendar?
  - Do we also have to keep tel. numbers, email, etc.
  - Should we expand the number of “fields”?
  - Should we re-enter the data?
Solution

- Can we link calendar with contacts?
  - “link” calendar entries with contact information and show the results of the two
    - Link should be based on something – simple solution is to link on person’s name
    - What if name is not unique?
    - How to follow links?

Organizing information

- Person has attributes
  - SSN
  - GW ID
  - Name
  - ..
- Student IS a person who:
  - Takes courses at GW
  - Is given grades
  - registers.
  - ..
- This is yet another kind of information…..
  - Where have you seen this?
**How to organize the information?**

- **What is the data needed?**
  - Eg: What do we need to store to uniquely identify a student entity?

- **How to store & organize the data?**
  - How many attributes are really needed
  - What is an efficient way to organize the data?
    - This is why we need to study schema design and Normal forms

- **How to query the data and generate reports for the end users?**
  - Need a database query language, such as SQL

**Data Models and data representation**

- All of the data have an implicit *data model*
  - Basic assumption on what is an item of data, how to interpret it, etc.

- A *data model* is a collection of concepts for describing data.
  - Starting point to design of DBMS
  - Provides us with the mathematical basis to prove/assert properties and show correctness of algorithms

- The *relational model* was the first model of data that is *independent* of its data structures and implementation
  - A theory of normalization guides you in designing relations
  - Other data models: network, hierarchical, Object Oriented…
**What Is a DBMS?**

- **Database** is a large, cohesive, collection of data.
  - Not arbitrary, unrelated data
  - Models real-world *enterprise*.

- A **Database Management System (DBMS)** is the software to store/retrieve and manage databases.
  - Provides an interface over the database
  - Examples: Oracle, MS SQL-server, MySQL,…

---

**Why use a DBMS?**

- Why do we need a DBMS, instead of coding in Java?
### DBMS Benefits

1. Generality: Programmer/user need not know implementation details; works with logical model
   - indices, sort orders, machine speeds, disk speeds, concurrency
2. Efficiency and Scale: DBMS takes care of optimizing for speed and scaling the system; user not aware
3. Concurrency and Reliability: DBMS handles these
   - What if system crashes – how to recover the data?
   - How to manage different users accessing data at the same time (eg. What if you have multiple threads)
     - Key concept of Transactions

### Layered Architecture of a DBMS

(Simplification!)

- **API/GUI**: SQL Query
- **Optimizer**: Physical plan
- **Catalog**: Schemas
- **Index/file/rec Mgr**: Data/etc Requests
- **Buffer Mgr**: Pages Requests
- **Storage Mgr**: Data Requests
- **Storage**: Pages
- **Logging, recovery**: Red = logical, Black = physical
The Layers of the DBMS

The Database Abstraction Provided by the DBMS

We think of databases at two levels:

- **Logical structure:**
  - What users/programmers see – program or query interface
- **Physical structure:**
  - Organization on disk, indices, etc.

The logical level is further split into:

- Overall database design (conceptual; seen by the DB designer)
- Views that various users get to see
Levels of Abstraction

- Many views, single conceptual (logical) schema and physical schema.
  - Views describe how users see the data.
  - Conceptual schema defines logical structure
  - Physical schema describes the files and indexes used.

* Schemas are defined using Data Definition Language (DDL);
* data is modified/queried using Data Manipulation Language (DML).

Data Independence

A user of a relational database system should be able to use the database without knowing about how the precisely how data is stored,

e.g.  
SELECT When, Where
FROM Calendar
WHERE Who = "Jane"

After all, you don't worry about IEEE floating-point when you do division in a Java program or with a calculator
More on Data Independence

Logical data independence
Protects the user from changes in the logical structure of the data:
could reorganize the calendar “schema” without changing how we query it

Physical data independence
Protects the user from changes in the physical structure of data:
could change how calendar is stored in memory without changing how the user would write the query

Presentation Layer (4th Tier):
Data-Driven Web Sites

- “Data driven web sites” also add an HTML “presentation” layer on top of what we’ve seen
- Or they use XML plus “style sheets” to get the same effect
How to define and use the database: Data Definition and Manipulation Languages

- data definition language (DDL) to specify database schema
- Data manipulation language (DML) allows users to access or manipulate data as organized by data model
  - procedural DMLs: require user to specify what data and how to get it
  - non-procedural DMLs: require user to specify what data is needed without specifying how to get it.
  - Commercial languages – SQL

Query Languages

- Formal query languages: Relational algebra, Relational Calculus, Domain calculus
  - Why study formal languages?
- Commercial query languages: SQL, QUEL
- SQL: “descendent” of SEQUEL; mostly relational algebra and some aspects of relational calculus
  - has procedural and non-procedural aspects
### Processing the Query

**Diagram:**
- **Web Server / UI / etc**
- **Execution Engine**
- **Storage Subsystem**

**Flow:**
1. **Hash**
2. **Merge**
3. **Optimizer**
   - **STUDENT**
     - Takes by cid
     - COURSE by cid

**SQL Query:**
```sql
SELECT *
FROM STUDENT, Takes, COURSE
WHERE STUDENT.sid = Takes.sID
AND Takes.cID = cid
```

### Architecture of Query Proc. Engine

**Diagram:**
- **SQL query**
- **Parse Query**
- **Select Logical Plan**
- **Select Physical Plan**
- **Query Execution**

**Flow:**
1. **Query optimization**
2. **Logical plan**
3. **Physical plan**
Summary

- DBMS used to maintain, query large datasets.
- Levels of abstraction give data independence.
  - A DBMS typically has a layered architecture.
  - Design and use of a database can be done using a query language.
- DBAs hold responsible jobs and are well-paid!
- DBMS form essential piece in information processing applications
  - Data mining, search engines, Human Genome

Course Administrivia...
Course Objectives

- Relational database theory and design
  - Concepts of data storage and retrieval
- Fluency in SQL and database application dev.
  - Working with relational database systems
- Software integration experience and team experience
  - Design and deploy a large database application
- Social impact analysis skills
  - Social impact of computers, professional ethics
- Technical writing skills and oral communication skills

Very Important: You must come prepared to class (read BEFORE class) and will work on problems during class

Course Outline: Topics

- **Part 1:** Design of Relational Databases
  - Entity-Relationship Model (similar to UML)
  - Formal Query Languages: Rel. algebra
  - Query languages: SQL
  - Relational Schema Design and Normal Forms, Tuning
  - Overview of DBMS architecture
    - File manager, transaction processing, query processing
  - Team based term Project
- **Part 2:** Social Impact of Computing (class discussions, guest speakers)
  - Social impact analysis
  - Codes of professional ethics
  - Intellectual property and software copyrights
  - Privacy
- Writing requirements: discussion of papers, term project report
Course Information - URL

- All course material will be placed at the URL:
  - [www.seas.gwu.edu/~bhagiweb/cs2441/](http://www.seas.gwu.edu/~bhagiweb/cs2441/)
  - Also linked from my homepage
- All course announcements will be placed on web-- check once a week!
- Textbook:
    - You can use any other Database textbook
  - Gift of Fire, Sara Baase. (for Part 2)

Course Requirements: Grading

- Exams: 30-35*%
  - Exam 1 on Databases
  - Exam 2 focuses more on Social Impact
- In-class discussions, assignments, Quizzes: 10-15*%
- Homeworks, Lab assignments, and programming assignments: 20%
  - Most are Database related
- Writing assignments: 10%
  - Includes report of project
- Team term project: 20-25*%
  - Demos required...to pass you must have a working project
- * will depend on final enrollment and size of teams
- Grades scaled as percentage of highest score in class
**Project: An example**

- A large project requirement with a set of “applications” will be posted on the web site
  - Objective: Build a complete end-to-end working project
- Team based project with multiple “phases”
  - Phase 1: each student assigned to an application, and required to provide a design of the database schema.
  - Phase 2: One set of teams build application A, Second set of teams build application B
  - Phase 3: Create new teams, with members from group A and B. Integrate the applications to provide a complete solution.
  - Integration is not = redesign!
  - Integration reflects real world team SW development practice
- A clear set of “minimum” requirements will be specified – note that meeting minimum requirements only implies a C grade on the project.
- Clear deadlines for specific phases and steps in the project will be posted – no extensions!

**Lab Sections and TA**

- Lab sections conducted by TA: Roxana Leontie
- Lab sections will cover
  - Intro MySQL, Oracle: SQL, PL/SQL, JDBC
  - Short tutorials – including application development using PHP, JDBC, XML
  - Clarifications on Programming Assignments
  - Help with analysis of Project (but not in the design of the project)
  - In-class exercises in some weeks: have to implement the queries during the lab; no extensions!
- Lab Assignments will be posted “separate” from homeworks – you may have BOTH due concurrently
Academic Integrity Policy

- www.cs.gwu.edu/academics/integrity.html
  - details and FAQ
- No collaboration (of any sort) on homeworks and programming assignments
  - Including external resources
- No collaboration between teams
  - within team each team member must have clear role -- i.e., clearly partitioned tasks for each team member
- violation of integrity policy -- default is maximum punishment (at least F for course)

Next ..

- Read Chapter 1,2
- Intro to Entity-Relationship Model
  - Homework 1 assigned – due Jan.24th !!
  - Lab Homework 1 will be posted – due next week.
  - Notes posted