CTEC323 Lecture 1

Dan Zingaro
OISE/UT

September 4, 2008

Welcome!

- Welcome to CTEC323
- Previous experience: background to Access would be helpful
- Topics: Database Systems, Data Models, Relational Database Model, Entity Relationship (ER) Modeling, Normalization of Database Tables, Introduction to SQL, Advanced SQL
- Evaluation: assignments (25%), group project (15%), midterm exam (25%), final exam (35%)
- How to prepare for class: textbook chapter, slides

Group Project

- In four groups of up to four students each, you will complete a project
- Three steps:
  - Prepare database diagram based on a provided case study
  - Prepare SQL code based on this diagram
  - Prepare and give presentation

Learning Outcomes

- Explain the concepts of relational database management system (RDBMS), particularly:
  - What an RDBMS is, and how it differs from older flat file systems.
  - The importance of the data model, its building blocks, and how it relates to business rules.
  - How data is organized through the use of integrity rules and primary and foreign keys.
  - The importance of relational set operators, the data dictionary, and indexes.
  - Explain the fundamental differences between logical and physical database design.

Learning Outcomes...

- Explain higher normal form designs, denormalized designs, existence dependence, relationship strength, weak entities, relationship participation, relationship degree, recursive relationships, and composite entities.
- Do the following for a simple prescribed business problem:
  - Develop an Entity Relationship Model with the appropriate entities, attributes, relationships, connectivity, and cardinality using Crows Foot notation to represent 1-1, 1-M, and M-N relationships.
  - Produce a Third Normal Form database design to remove redundancies in tables.
  - Convert the Third Normal Form database design into a set of database tables using an appropriate tool such as Oracle or MS SQL Server, and SQL Data Definition Language (DDL).
  - Use SQL Data Manipulation Language to create and query sample data.

Course Decisions

- When are assignments due?
- What form should assignment submissions take?
- How should project groups be formed?
- What about breaks during class?

Data VS. Information

- Data: raw facts
- Information: processing data to reveal meaning; is the foundation for decision-making
- Data management: focuses on generation, storage and retrieval of data
- Database: shared, integrated structure that stores data and metadata (characteristics of data and their relationships)

Bluejays Wins per Year

Year	Wins
1977	54
1978	59
1979	53
1980	67
1981	37
1982	78
1983	89
1984	89

Is this data or information? Give other examples.

Database Management System (DBMS)

The DBMS manages structure and controls access to the data; it is the intermediary between user and database. Advantages:

- Allows data to be shared
- Better data integration
- Minimizes data inconsistency
- Improved data access with ad hoc queries (a query is a request for data manipulation)
- Improved decision making
- Increased end-user productivity

Database Classification

- Single vs. multi-user
- Centralized (one site) vs. distributed (multiple sites)
- Operational vs. data warehouse
- Operational supports day-to-day operations of business
- Data warehouse stores data used to make strategic decisions
- Question: why make these distinctions at all?

File Systems

- Managing data with file systems is obsolete, but we should be aware of where we came from
- Manual filing system makes complex reporting tasks difficult
- Initially, computer files in a file system were similar to manual files
- New programs were written to generate each new type of report
- Each file used its own programs to store, retrieve and modify data
- Files were owned by the department that had them created

Drawbacks of File Systems

- Even simplest data-retrieval task requires extensive programming in procedural 3GL
- No ad hoc query capabilities
- As the number of files expands, system administration becomes more difficult
- Difficult to make changes to existing structures
- Security features are difficult to program and so are often omitted

Drawbacks of File Systems...

- Structural dependence: access to a file is dependent on its structure; adding a field breaks all programs that work with the file
- Data dependence: all programs must be changed when the data type of a field is changed
- Storing a phone number or address in a single field can cause access problems

Drawbacks of File Systems...

Consider a customer file with fields CustomerName, CustomerPhone, InsuranceAmount, AgentName, AgentPhone. This structure is typical and leads to data redundancy and:

- Data inconsistency
- Data anomalies, including:
  - Update (when changes are made to existing records)
  - Insert (when entering new records)
  - Delete (when deleting records)

Databases

- A database is stored in a single logical data repository
- This helps eliminate data inconsistency, data anomalies, data dependency and structural dependency problems
- Database system refers to an organization of components that define and regulate the collection, storage, management, and use of data within a database environment
- Five components: hardware, software (OS, DBMS, applications), people (system admin, database admin, database designers, systems analysts and programmers, end users), procedures (govern design and use of system) and data

DBMS Functions

- Data dictionary management
  - Stores definitions of data elements and their relationships (metadata)
  - Removes structural and data dependency because all data access is through the DBMS
- Data storage management
- Data transformation and presentation
  - Transforms entered data to conform to expected logical format

DBMS Functions...

- Security management
  - Rules on data and operations permitted for users
- Multiuser access control (concurrency without compromising integrity)
- Backup and recovery management
- Data integrity management (through relationships)
- Database access languages and APIs (e.g. query languages)
- Database communication interfaces

Costs of Databases

- Hardware
- Software
- Personnel
- Management complexity
- Maintaining currency
- Vendor dependence

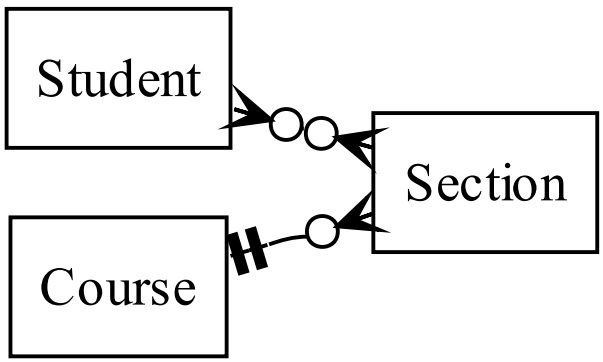
Database Diagraming Tools

- Often want to visualize what is stored in a database
- At least three approaches
- Drawing tools
  - Visio (Appendix A on the textbook's website has a tutorial)
  - Dia (free)
- Automatic diagram creation based on relationships among data
  - Graphviz
- Diagram creation from existing databases
  - Schemaspy (Java tool; integrates directly with database)
  - Sql::Translator (PERL tool; reads SQL dumped from database)

Database Diagraming Tools...

- Different approaches are valuable at different times in database development
- e.g. we cannot extract info from a nonexistent database!
- e.g. we should *not* create a database and *only then* extract the structure
- Sometimes our diagrams are at a higher level of abstraction than what ultimately gets implemented

Example: Students, Courses and Sections



A student can take zero or more sections. A section consists of zero or more students. A course contains zero or more sections. A section is associated with exactly one course.

Hands-on Problems

- Consider a file with fields for project code, project manager, manager phone, manager address, project bid price
- How many fields does the file have? How would we count records?
- What problem results when trying to make a listing by city. Solution? How about a listing by last name, area code, city, state, zip code?
- Where's the data redundancy?