CSC 355 Database Systems
Lecture 1

Eric J. Schwabe

School of Computing, DePaul University

Spring 2020

Topics:

- Course Organization
- Introduction to Databases
- SQLDeveloper

Contact Information

- Course web site: http://d2l.depaul.edu/
  - Weekly discussion forum for questions/comments
- Office hours: Monday and Wednesday 1:00pm-2:00pm, and Wednesday 3:00pm-4:00pm
  - Via Zoom at https://depaul.zoom.us/my/eschwabe
- Email: eschwabe@depaul.edu
  - Please begin subject line with “CSC 355” and sign your name
  - Expect reply within one business day

Course Texts

- Required text:
  A First Course in Database Systems (third edition),
  Ullman and Widom
  (ISBN 978-0136006374)
- Additional reference (optional):
  Murach's Oracle SQL and PL/SQL for Developers (second edition), Murach (ISBN 978-1890774806)

Course Policies

- Grading: 30% homework, 30% midterm exam, 30% final exam, 10% quizzes
  - HWs accepted late (up to 24 hours only) with penalty
  - Lowest HW score will be dropped
  - Submit HWs through d2l, no emailed submissions
  - Exam will be given through d2l, details TBA
- Course prerequisite: CSC 301 or CSC 393
- University policies: See posted syllabus
- Weekly schedule: See posted document

Academic Integrity Policy

- http://academicintegrity.depaul.edu/
  - Cheating: Any action that violates university norms or instructor guidelines for course work
  - Plagiarism: Any use of another's work without proper citation where original work is expected
  - Complicity: Any action that facilitates an academic integrity violation

Databases are Everywhere

- Amazon (or any online store…)
- Southwest (or any airline…)
- Chase (or any bank…)
- Campusconnect (or any university system…)
- …and those are just a few…you are interacting with databases every day…

What is a database?

- *Data* is information that can be recorded and has a known meaning
- A *database* is an organized collection of logically related data that are typically…
  - Persistent: are stored on a stable medium
  - Shared: have multiple uses and interested users
  - Interrelated: form a bigger picture

Why Use a Database System?

- Early data processing systems used files of data in plain text form
- Problem: program-data dependence led to
  - limited data sharing
  - duplication of data
  - increased time for development and maintenance

Why Use a Database System?

- A database uses a single repository of data accessed by multiple users
  - Contains information on the structure of the data
  - Allows sharing of and concurrent access to data
  - Supports different views of the data
- The costs are higher overhead for the design, implementation, and maintenance of the data
- What are the benefits?

Benefits of Database Systems

- Program-data independence
- Controlled data redundancy
- Controlled access to data
- Support for multiple user interfaces
- More efficient query processing
- Faster application development

Database Management Systems

- A *database management system (DBMS)* is a collection of software components that lets you
  - create (e.g., define, construct)
  - maintain (e.g., modify, keep available)
  - control access to (e.g., secure, allow queries to)

  a database

Database Management Systems

- DBMS Examples: Oracle, IBM DB2, MS Access/SQL Server, MySQL
- We can work with a DBMS directly or through an application that supplies a particular interface (e.g., SQLDeveloper)
- The database and DBMS together make up a *database system*

Database Models

- Older Models:
  - File Systems, Hierarchical, Network
  - All had drawbacks…
- The Relational Model
- Newer Models:
  - Semi-structured, Object-relational, NoSQL
  - …not as popular as Relational Model

File Systems

- Data stored in simple text files, each one possibly having a different fixed organization of its data
- High level of program-data dependence
- Difficult to share data
- Not practical to optimize queries

Hierarchical/Network Models

- Hierarchical Model: Data arranged in "parent-child" relationships
- Network Model: Can represent more general relationships among types of data
- Both models have similar weaknesses:
  - Applications must navigate relationships explicitly
  - DBMS can not rearrange data to optimize queries

Relational Model

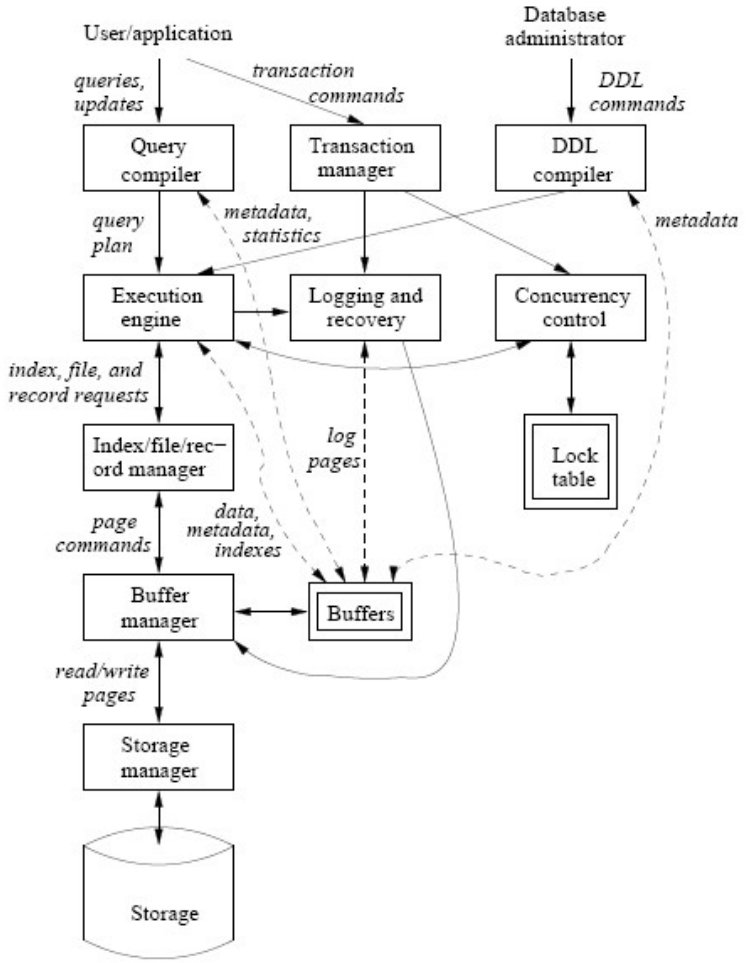
- First model to separate the logical structure of the database from its physical implementation
- Data are divided into two-dimensional tables called relations
- Tables are linked by shared columns of data
- Rules exist for dividing data among tables
- A standardized query language exists (SQL)

Newer Models

- Semi-structured databases: Store collections of data in XML files
- Object-relational databases: Add support for structured data types to relational databases
- Document databases: Have a less restrictive structure, typically without a fixed schema
- Data warehouses: Integrate multiple sources of data, possibly from different models

Components of a DBMS

(From Ullman/Widom)



User Interactions with DBMS

- Database Definition: Create database schema, links between tables, constraints
- Query Processing: Request retrieval or modification of data (“queries”/“actions”)
- Transaction Processing: Execute sets of operations that must be executed as a unit (“transactions”)

Approximate Course Schedule

- Week 1: Introduction and Relational Model
- Weeks 2-5: SQL DDL, Queries, Transactions
- Weeks 6-7: Relational Database Design
- Weeks 8-9: Constraints and Triggers, Database Programming, Views
- Week 10: Slack Time / Course Review

SQLDeveloper

- SQLDeveloper is an application that works as a “front-end” connection to a server running an Oracle DBMS (e.g., Oracle 12c)
- SQL commands can be run individually, or collected in script files.
- Can be downloaded free from Oracle

Setting Up a Connection

- To set up a new connection to acadoradbprd01:
  - Connection Name: *YOURNAME355*
  - Username: your campusconnect username
  - Password: cdm######## (initially uses your 7-digit Student ID)
  - Hostname: acadoradbprd01.dpu.depaul.edu
  - Port: 1521
  - SID: ACADPRD0
  - Test, then Connect…
- Double-click to Open an existing connection
- Disconnect (and commit) when you're done!

Running SQL Commands

- Single SQL command:
  - Type command, then Execute (Ctrl-Enter)
    - e.g, to change password, ALTER USER *username* IDENTIFIED BY *newpassword* ;
- Script (SQL commands stored in a file):
  - Type @ followed by full path to script file, then Run Script (F5)
- Output will appear in bottom window under Query Result or Script Output

Browsing Database Tables

- Left window shows current Tables, click on + to expand list
- Right-click on Tables and choose Refresh to see changes (can also Commit changes)
- Click on a table to view it in the center window (may need to Refresh view also)
  - COLUMNS shows schema
  - DATA shows contents

Saving SQLDeveloper Output

- Three ways:
  - Click on Save icon to save contents of Script Output window to a file
  - Highlight and then copy and paste contents of Script Output window to a file
  - Take and save screenshot of SQLDeveloper display

Next:

- The Relational Model