SOFTENG 351
Fundamentals of Database Systems

Introduction

Topics

- Focus on the fundamental concepts necessary for designing, using and implementing database systems and database applications.
  - Relational models, Entity-Relationship (ER) diagrams
  - Structured Query Language (SQL)
  - Functional Dependencies and Normalization
  - SQL and Web Database programming
  - File Storage and Indexing structures
  - Query processing and optimization
  - Transaction processing
  - Concurrency control
  - Database recovery

Teaching Team

- Lecturers
  - Assoc Prof Jing Sun
    - Email: jing.sun@auckland.ac.nz
    - Extension: 82286
    - Room: 303.522
    - Office hour: Friday 1-2pm (or arranged by email)
  - Dr Miao Qiao (Course Coordinator)
    - Email: miao.qiao@auckland.ac.nz
    - Extension: 88941
    - Room: 303. 524
    - Office hour: Wednesday 3-4pm (or arranged by email)
- Tutors
  - Zhenyun Deng
    - Email: zden658@aucklanduni.ac.nz
  - Elliott Wen
    - Email: jwen929@aucklanduni.ac.nz
  - Xiaojie Liu
    - Email: xliu785@aucklanduni.ac.nz

Assessment

- **Note: Students must obtain a pass in both the Practical (labs + assignments) and Theory (test + exam) work in order to pass the course as a whole.**
- Laboratories 10 weeks (starting week 2) ……..……..…….. 10%
  - Lab session is 1 hour duration – exercises to be submitted before the weekly due dates (usually by Sundays)
  - Note: First lab is on Monday 9 March, and NO lab during week 9 of the test
- Four Assignments ……..……..…….. 20%
  - Assignment 1 due Week 3, Thursday 19 March, 5pm (5%)
  - Assignment 2 due Week 6, Thursday 9 April, 5pm (5%)
  - Assignment 3 due Week 8, Thursday 7 May, 5pm (5%)
  - Assignment 4 due Week 11, Thursday 28 May, 5pm (5%)
- Mid-semester Test …….. 10%
  - Week 9, Wednesday 13 May, 6:25pm – 7:45 pm (tentative)
- Final Exam ……... 60%
  - Date to be announced (TBA)

Laboratories

- All Laboratories will be started from Monday 9 March 2020.
- One laboratory per week on three drop-in sessions (except for week 9)
  - Mondays
    - Mo 5:00PM - 6:00PM, 405-326
  - Tuesdays
    - Tu 3:00PM - 4:00PM, 405-326
  - Wednesdays
    - We 4:00PM - 5:00PM, 405-326

Resources

- Lectures
  - Note: All marks, lecture slides, recordings and announcements can be found on the Canvas system. https://canvas.auckland.ac.nz
- Recommended Textbook
  - Fundamentals of Database Systems, 7th Edition, by Ramez Elmasri and Shamkant B. Navathe, Pearson Publisher.
- Piazza Forum
  - Questions and answers – peers, tutors and lecturers
  - Linked from Canvas access
- Additional resources
  - PeerWise – exercise/question bank and answer recommendations
  - https://peerwise.cs.auckland.ac.nz/at/?uoa_nz
  - Use the Course ID '20981' and your UPI to register.

SE Specialization Representative

- ECSE has 2 representatives for each specialization and year
  - i.e., two Part-III SOFTENG students for the role in 2020
  - Please contact Andrew Austin a.austin@auckland.ac.nz for the nomination procedure.
- Attends staff student meetings
- Pass on student concerns to lecturers

SOFTENG 351
Fundamentals of Database Systems

Database and Database Users

OUTLINE

- Types of Databases and Database Applications
- Why Databases?
- Basic Definitions
- Typical DBMS Functionality
- Example of a Database (UNIVERSITY)
- Main Characteristics of the Database Approach
- Types of Database Users
- Advantages of Using the Database Approach
- When Not to Use Databases

Types of Databases and Database Applications

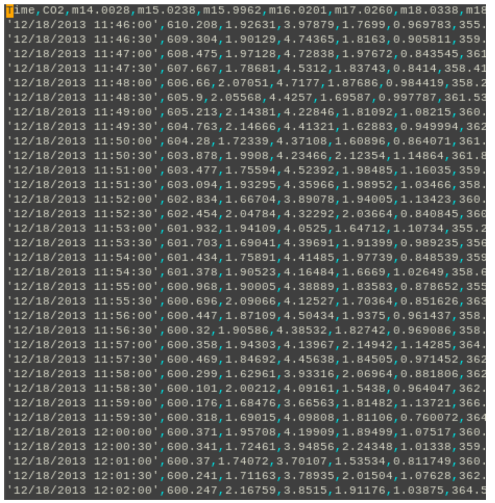
- Traditional Applications:
  - Numeric and Textual Databases
- More Recent Applications:
  - Multimedia Databases
  - Geographic Information Systems (GIS)
  - Social Networks started capturing a lot of information about people and about communications among people-posts, tweets, photos, videos in systems such as:
    - - Facebook
    - - Twitter
    - - Linked-In
  - Search Engines- Google, Bing, Yahoo : collect their own repository of web pages for searching purposes

Impact of Databases and Database Technology

- Businesses:
  - Banking, Insurance, Retail, Transportation, Healthcare, Manufacturing
- Service Industries:
  - Financial, Real-estate, Legal, Electronic Commerce, Small businesses
- Education :
  - Resources for content and Delivery
- More recently:
  - Social Networks, Environmental and Scientific Applications, Medicine and Genetics
- Personalized Applications:
  - based on smart mobile devices

Why Databases?

- So far: everything is stored in files
- What are the problem with this?



Why Databases?

- So far: everything is stored in files
- What are the problem with this?
  - Redundancies and inconsistencies
  - Data access
  - Data isolation
  - Integrity problems
  - Atomicity of updates
  - Concurrent access by multiple users
  - Security

Basic Definitions

- **Database:**
  - A collection of related data.
- **Data:**
  - Known facts that can be recorded and have an implicit meaning.
- **Mini-world:**
  - Some part of the real world about which data is stored in a database. For example, student grades and transcripts at a university.
- **Database Management System (DBMS):**
  - A software package/ system to facilitate the creation and maintenance of a computerized database.
- **Database System:**
  - The DBMS software together with the data itself. Sometimes, the applications are also included.

Simplified database system environment

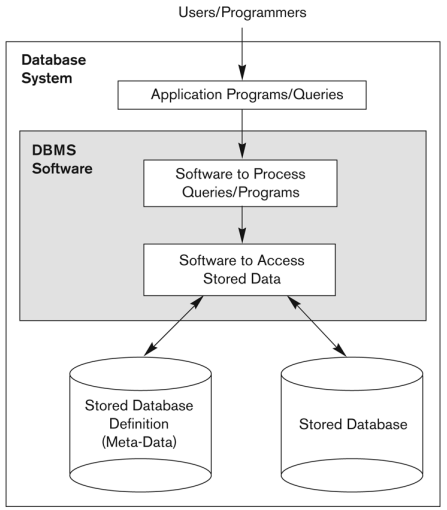


Figure 1.1
A simplified database system environment.

Typical DBMS Functionality

- *Define* a particular database in terms of its data types, structures, and constraints
- *Construct* or Load the initial database contents on a secondary storage medium
- *Manipulating* the database:
  - Retrieval: Querying, generating reports
  - Modification: Insertions, deletions and updates to its content
  - Accessing the database through Web applications
- *Processing* and *Sharing* by a set of concurrent users and application programs – yet, keeping all data valid and consistent

Application Activities Against a Database

- Applications interact with a database by generating
- Queries:
  - access different parts of data and formulate the result of a request
- Transactions:
  - may read some data and "update" certain values or generate new data and store that in the database
- Applications must not allow unauthorized users to access data
- Applications must keep up with changing user requirements against the database

Example of a Database (with a Conceptual Data Model)

- **Mini-world for the example:**
  - Part of a UNIVERSITY environment.
- **Some mini-world *entities*:**
  - STUDENTs
  - COURSEs
  - SECTIONs (of COURSEs)
  - (academic) DEPARTMENTs
  - INSTRUCTORs

Example of a Database (with a Conceptual Data Model)

- **Some mini-world *relationships*:**
  - SECTIONs *are of specific* COURSEs
  - STUDENTs *take* SECTIONs
  - COURSEs *have prerequisite* COURSEs
  - INSTRUCTORs *teach* SECTIONs
  - COURSEs *are offered by* DEPARTMENTs
  - STUDENTs *major in* DEPARTMENTs
- Note: The above entities and relationships are typically expressed in a conceptual data model, such as the ENTITY-RELATIONSHIP (ER) data model

Example of a simple database

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	04	King
92	CS1310	Fall	04	Anderson
102	CS3320	Spring	05	Knuth
112	MATH2410	Fall	05	Chang
119	CS1310	Fall	05	Anderson
135	CS3380	Fall	05	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Figure 1.2
A database that stores student and course information.

Main Characteristics of the Database Approach

- **Self-describing nature of a database system:**
  - A DBMS **catalog** stores the description of a particular database (e.g. data structures, types, and constraints)
  - The description is called **meta-data***.
  - This allows the DBMS software to work with different database applications.
- **Insulation between programs and data:**
  - Called **program-data independence**.
  - Allows changing data structures and storage organization without having to change the DBMS access programs.

Example of a simplified database catalog

RELATIONS

Relation_name	No_of_columns
STUDENT	4
COURSE	4
SECTION	5
GRADE_REPORT	3
PREREQUISITE	2

COLUMNS

Column_name	Data_type	Belongs_to_relation
Name	Character (30)	STUDENT
Student_number	Character (4)	STUDENT
Class	Integer (1)	STUDENT
Major	Major_type	STUDENT
Course_name	Character (10)	COURSE
Course_number	XXXXNNNN	COURSE
….	….	…..
….	….	…..
….	….	…..
Prerequisite_number	XXXXNNNN	PREREQUISITE

Note: Major_type is defined as an enumerated type with all known majors. XXXXNNNN is used to define a type with four alpha characters followed by four digits

Figure 1.3
An example of a database catalog for the database in Figure 1.2.

Main Characteristics of the Database Approach (continued)

- **Data Abstraction:**
  - A **data model** is used to hide storage details and present the users with a conceptual view of the database.
  - Programs refer to the data model constructs rather than data storage details
- **Support of multiple views of the data:**
  - Each user may see a different view of the database, which describes **only** the data of interest to that user.
    - A view that groups all the students who took each section and gives each student's grade – useful for grade reporting purpose.
    - A view that gives the number of courses taken and the GPA for each student – useful for determine honors students.

Main Characteristics of the Database Approach (continued)

- **Sharing of data and multi-user transaction processing:**
  - Allowing a set of **concurrent users** to retrieve from and to update the database.
  - *Concurrency control* within the DBMS guarantees that each **transaction** is correctly executed or aborted
  - *Recovery* subsystem ensures each completed transaction has its effect permanently recorded in the database
  - **OLTP** (Online Transaction Processing) is a major part of database applications. This allows hundreds of concurrent transactions to execute per second.

Database Users

- Users may be divided into
  - Those who actually use and control the database content, and those who design, develop and maintain database applications (called "Actors on the Scene")
    - Database administrators:
    - Database Designers
    - End-users
    - System Analysts, Application Programmers and Business Analysts
  - Those who design and develop the DBMS software and related tools, and the computer systems operators (called "Workers Behind the Scene").
    - DBMS System Designers and Implementors, Database Tool Developers
    - Operators and Maintenance Personnel - running and maintenance of the database system hardware and software environment.

Advantages of Using the Database Approach

- Controlling redundancy in data storage and in development and maintenance efforts.
  - Sharing of data among multiple users.
- Restricting unauthorized access to data. Only the DBA staff uses privileged commands and facilities.
- Providing Storage Structures (e.g. indexes) for efficient Query Processing
- Providing backup and recovery services.
- Providing multiple interfaces to different classes of users.
- Representing complex relationships among data.
- Enforcing integrity constraints on the database.

When not to use a DBMS

- Main inhibitors (costs) of using a DBMS:
  - High initial investment and possible need for additional hardware.
  - Overhead for providing generality, security, concurrency control, recovery, and integrity functions.
- When a DBMS may be unnecessary:
  - If the database and applications are simple, well defined, and not expected to change.
  - If access to data by multiple users is not required.
- When a DBMS may be infeasible:
  - In embedded systems where a general purpose DBMS may not fit in available storage.

Summary

- A database is a collection of related data (recorded facts).
- A DBMS is a generalized software package for implementing and maintaining a computerized database.
- The database and the software together form a database system.
- Advantages of databases approaches verses traditional file-processing applications.
- Different database users, such as administrators, designers, end-users, system analysts, software engineers, personals associated to DBMS software/environment, etc.