COSC344
Database Theory and Applications

Lecture 1: Introduction
Welcome to

COSC 344

Database Theory and Applications
Course Goals

• Introduce the fundamental concepts, principles, and problems in database.
• Introduce the principles of database design and database management system
• Learn how to design, implement, optimize the underlying database management system

What will you get from this course?

Necessary knowledge and fundamental skills you’ll need in your future career

Theory + Practice
Teaching Team

• Lecturers

  Haibo Zhang (Course Coordinator)  Yawen Chen
  2.47, Owheo  2.46, Owheo
  479-8534  479-5740
  haibo@cs.otago.ac.nz  yawen@cs.otago.ac.nz

• Lab Demonstrator

  James Walmsley
  Room 1.28, Owheo
  walja829@cs.otago.ac.nz

• Oracle Database Administrator

  Cathy Chandra
  1.21, Owheo
  479-58580
  cathy@cs.otago.ac.nz
Course Details

• Recommended Textbook
    http://www.pearsoned.co.nz/9781292097619
• Course webpage
  – http://www.cs.otago.ac.nz/cosc344
• Consultation
  – Send email to book a time slot
• Almost one lab per week (no lab in week 1, 10, and 13)
  – Two streams, only need to attend one
  – Students enrolled in COSC242: attend the 2nd hour of both labs
Outline of Lectures

Database fundamentals and design

- Introduction to databases, models
  - Database design, ER modeling
  - Relational model
  - Relational algebra & operator
  - Functional dependencies
  - Normalization
  - View & NULL

- SQL, Java & SQL, C & SQL, PHP & SQL
  - PL/SQL, Triggers

- DBMS Architecture & System catalog
  - Database file and storage
  - Database indexing (1 & 2)
  - Database security & auditing
  - Transactions
  - Concurrency control
  - Query optimisation
  - NoSQL and big data

Database programming

- DBMS architecture, design and optimization

- Non-relational models
Assessment

- Two assignments - 25%
  - Friday, 02/8/19 - 10%
  - Friday, 23/8/19 - 15%

Late submissions incur a 10% penalty (of marks) per working day late.

- Lab assessments - 15%
  - Not every lab has assessment – check labnotes

- Exam - 60%
  - 3 hours
  - A mark of at least 50% is required to pass the course
A Useful Study Tip

Do homework first → Attend lectures → Lab practice

Remember to check your university email account regularly!
Academic Integrity and Academic Misconduct

**Academic integrity** means being honest in your studying and assessments. It is the basis for ethical decision-making and behaviour in an academic context. Academic integrity is informed by the values of honesty, trust, responsibility, fairness, respect and courage. Students are expected to be aware of, and act in accordance with, the University’s Academic Integrity Policy.

**Academic Misconduct**, such as plagiarism or cheating, is a breach of Academic Integrity and is taken very seriously by the University. Types of misconduct include plagiarism, copying, unauthorised collaboration, taking unauthorised material into a test or exam, impersonation, and assisting someone else’s misconduct. A more extensive list of the types of academic misconduct and associated processes and penalties is available in the University’s Student Academic Misconduct Procedures.
Academic Integrity Continued

It is **your responsibility** to be aware of and use acceptable academic practices when completing your assessments. To access the information in the Academic Integrity Policy and learn more, please visit the University's Academic Integrity website at [www.otago.ac.nz/study/academicintegrity](http://www.otago.ac.nz/study/academicintegrity) or ask at the Student Learning Centre or Library.

- Academic Integrity Policy
  [www.otago.ac.nz/administration/policies/otago116838.html](http://www.otago.ac.nz/administration/policies/otago116838.html)

- Student Academic Misconduct Procedures
  [http://www.otago.ac.nz/administration/policies/otago116850.html](http://www.otago.ac.nz/administration/policies/otago116850.html)
Learning Objectives of Lecture 1

• You should
  – Understand **database** and **database management system**
  – Be able to explain the advantage of managing data using the database approach
  – Understand **data modelling** fundamentals such as data model, database schema, and three-schema architecture
  – Understand **data independence** and its advantage

• Source
  – Textbook: Chapters 1, Chapter 2.1 – 2.4
Try It

• List some ways you encounter databases in everyday activities.
A Motivating Example - eVision

- To store the information about:
  - students
  - courses
  - staff
  - who takes what, who teaches what

- Allow users to query/update:
  - who teaches “COSC344”, enroll “Mary” in “COSC344”

- Allow >1000 users to access data simultaneously

- Store the data for a long period of time
  - Protect against crashes
  - Protect against unauthorized use
Introduction to Databases (1)

• What is a database?
  – A database is a collection of related data.

• A database generally has the following implicit properties:
  – Represents some aspect of the real-world, called miniworld
  – Is a coherent collection of data with inherent meaning
  – Is designed, built, and populated with data for a specific purpose.

• A database can be of any size and complexity
  – Small database: Address book, student database in CS
  – Large database: Inland Revenue Department database, Amazon.com,
Large Databases
Introduction to Databases (2)

• What is a database management system (DBMS)?
  – A collection of programs that enable to define, construct, manipulate, and share databases among various users and applications.
  – Commercial DBMSs: DB2, Oracle, SQL Server …

• Application program
  – Accesses the database by sending queries and requests for data to the DBMS
A Simplified Database System Environment

Database System

Users/Programmers

Application Programs/Queries

DBMS Software

Software to Process Queries/Programs

Software to Access Stored Data

Database definition (Meta-data)

Stored Database
An Example Database

<table>
<thead>
<tr>
<th>STUDENT</th>
<th>Name</th>
<th>StudentNumber</th>
<th>Class</th>
<th>Major</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>17</td>
<td></td>
<td>1</td>
<td>CS</td>
</tr>
<tr>
<td>Brown</td>
<td>8</td>
<td></td>
<td>2</td>
<td>CS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COURSE</th>
<th>CourseName</th>
<th>CourseNumber</th>
<th>Credits</th>
<th>Dept</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intro to Comp Sci</td>
<td>COSC1310</td>
<td>4</td>
<td>CS</td>
<td></td>
</tr>
<tr>
<td>Data Structures</td>
<td>COSC3320</td>
<td>4</td>
<td>CS</td>
<td></td>
</tr>
<tr>
<td>Discrete Maths</td>
<td>MATH2410</td>
<td>3</td>
<td>MATH</td>
<td></td>
</tr>
<tr>
<td>Database</td>
<td>COSC3380</td>
<td>3</td>
<td>CS</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SECTION</th>
<th>SectionID</th>
<th>CourseNumber</th>
<th>Semester</th>
<th>Year</th>
<th>Instructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>85</td>
<td>MATH2410</td>
<td>Fall</td>
<td>98</td>
<td>King</td>
<td></td>
</tr>
<tr>
<td>92</td>
<td>COSC1310</td>
<td>Fall</td>
<td>98</td>
<td>Anders</td>
<td></td>
</tr>
<tr>
<td>102</td>
<td>COSC3320</td>
<td>Spring</td>
<td>99</td>
<td>Knuth</td>
<td></td>
</tr>
<tr>
<td>112</td>
<td>MATH2410</td>
<td>Fall</td>
<td>99</td>
<td>Chang</td>
<td></td>
</tr>
<tr>
<td>119</td>
<td>COSC1310</td>
<td>Fall</td>
<td>99</td>
<td>Anders</td>
<td></td>
</tr>
<tr>
<td>135</td>
<td>COSC3380</td>
<td>Fall</td>
<td>99</td>
<td>Turing</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GRADE_REPORT</th>
<th>StudentNumber</th>
<th>SectionID</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>112</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>119</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>85</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>92</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>102</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>135</td>
<td>A</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PREREQUISITE</th>
<th>CourseNumber</th>
<th>PrerequisiteNumber</th>
</tr>
</thead>
<tbody>
<tr>
<td>COSC3380</td>
<td>COSC3320</td>
<td></td>
</tr>
<tr>
<td>COSC3380</td>
<td>MATH2410</td>
<td></td>
</tr>
<tr>
<td>COSC3320</td>
<td>COSC1310</td>
<td></td>
</tr>
</tbody>
</table>

Example database (Adapted from Elmasri & Navathe Fig 1.2)
Characteristics of the Database Approach

• Self-describing nature of a database system
  – Database + Meta-data (a complete definition of data structure and constraints stored in DBMS catalog)

• Insulation between programs and data
  – Program-data independence
  – Program-operation independence

• Support of multiple views of data
  – A subset of the database
  – Virtual data derived from the database files

• Sharing of data
  – Concurrency control
  – Online transactions processing
Actors on the Scene

• Database Administrators (DBA)
  – Authorizing access to the database
  – Coordinating and monitoring its use
  – Maintain software and hardware resources

• Database Designers
  – Identify the data to be stored and choose the appropriate structure to represent and store the data

• End Users

• System Analysts and Application Programmers.
User name and password are stored in an encrypted file.

Implement a class to read user name and password from the file.

What about database design?

```java
public class UserPass {
    private String password;
    private String username;

    // Constructor - Also reads the username and password from the file.
    public UserPass () {
        String line = null;
        String passwordFile = "pass.dat";
        try {
            String line = null;
            String passwordFile = "pass.dat";
            try {
            ...
        } catch (IOException e) {
            // Handle exception
        }
    }

    public String getPassword() {
        return password;
    }

    public String getUsername() {
        return username;
    }
}
```
Data Modeling and Data Models

• Data modeling: the first step in designing a database
  – the process of creating a specific data model for a
determined problem domain
• A data model is a collection of concepts that can be used to describe the structure of a database such as
data types, relationships, and constraints that should hold for the data.
Categories of Data Models

• High-level or **conceptual** data models
  – close to the way most users perceive data
  – entity
  – attribute
  – relationship

• Low-level or **physical** data models
  – storage details

• **Representational** or **implementation** data models
  – bridge the gap
  – relational
  – network
  – hierarchical
Database Schema

- **Database Schema**: the description of a database

  - **STUDENT** | Name | StudentNumber | Class | Major
  - **COURSE** | CourseName | CourseNumber | Credits | Dept
  - **SECTION** | SectionID | CourseNumber | Semester | Year | Instructor
  - **GRADE_REPORT** | StudentNumber | SectionID | Grade
  - **PREREQUISITE** | CourseNumber | PrerequisiteNumber

Example Database Schema (Adapted from Elmasri & Navathe Fig 2.1)
Three-Schema Architecture (1)

- **Goal:** to separate the user applications from the physical database

- External Level
- Conceptual Level
- Internal Level

- External View
- Conceptual Schema
- Internal Schema
- Stored Database

Also known as the ANSI/SPARC architecture
Three-Schema Architecture (2)

- **Internal Schema**
  - Uses a physical data model
  - Describes the physical storage structure of the database

- **Conceptual Schema**
  - Hides the details of the physical storage structures
  - Concentrates on describing entities, data types, relationships, user operations, and constraints

- **External Schema**
  - Describes the part of the database that a particular user group is interested in
  - Hides the rest of the database from that user group

- **Mapping**
  - Transform requests and results between adjacent levels
Data Independence

Defined as the ability to change the schema at one level of a database system without having to change the schema at the next higher level

- **Logical** data independence
  - The capacity to change the conceptual schema without having to change external schemas
  - e.g. add/remove record type, change constraints

- **Physical** data independence
  - The capacity to change the internal schema without having to change the conceptual schemas
  - e.g. creating additional access structures
DBMS Languages

• Data definition language (DDL)
  – Used to define the conceptual schema
  – A DDL compiler is used to process DDL statements

• Storage definition language (SDL)
  – Used to define the internal schema
  – Most relational DBMS do not have specific SDL languages

• View definition language (VDL)
  – To specify user views and their mapping to conceptual schema
  – Most DBMSs use DDL for both conceptual and external schema

• Data manipulation language (DML)
  – High-level or nonprocedural DML (set-at-a-time)
  – Low-level or procedural DML (record-at-a-time)
Question to Ponder

• Given some mini-world, how do we design a database?
  – What do you put into it?
  – How are the pieces interrelated?
Are you

✓ Proactive, friendly and keen to contribute to your learning environment?
✓ A great communicator who can represent your peers?

What’s in it for you?

✓ Great friendships
✓ Access to FREE professional development opportunities and support
✓ A reference letter from OUSA for your CV
✓ FREE food

Don’t wait any longer... sign up now!
go to www.ousa.org.nz/support/class-reps