Course Introduction

COSC 404 – Database System Implementation

Introductions – Dr. Ramon Lawrence

Professor, Computer Science

Research area: database systems, Internet of Things, software development

Teaching experience:

- 2020 Killam Teaching Prize (top teaching award at UBCO – one per year)
- 2017 UBCO Teaching Excellence Award Winner (two per year)
- 9-time member of teaching honour roll (top 10% instructors)

Industry experience: GE Big Data, UnityJDBC company/consulting

Note: May address me as "Dr. Lawrence", "Professor", or "Ramon" (pronounced RAY-MUN).

The Essence of the Course

The overall goal of this course is for you to:

Understand database algorithms and techniques in order to:

1) Be a better, "expert" user of database systems.
2) Be able to use and compare different database systems.
3) Adapt the techniques when developing your own software.

This course opens the database system "**black box**".

My Course Goals

1) Provide the information in an effective way for learning.
2) Inspire and motivate students to learn and appreciate the course.
3) Strive for ***all*** students to understand the material and excel.
4) Be available for questions during scheduled times, office hours, and at other times as needed.
5) Provide a background on the fundamental concepts of database systems including transactions and concurrency.
6) Create opportunities to learn concepts by experimenting and programming with different database systems.
7) Encourage students to continue studying databases including further projects and graduate level research!

Your Course Goals

1) Sufficiently learn the material to pass the course.
2) Develop experience in using a variety of database systems.
3) Understand how a database system works in order to better understand how to use them properly.
4) Learn algorithms and techniques that constitute the ***foundations*** of database theory and implementation.
5) Realize that database technology is present in many areas including operating systems, networks, and programming.
6) Determine if you want to continue with database related research.

Course Objectives

1) Experience using and developing on many different SQL and NoSQL databases.
2) Proficiency in manipulating data in memory and storage and using index structures for improved performance.
3) Understand query processing including parsing, translation, optimization, and execution.
4) Apply principles of transactions, concurrency, recovery, and distribution for databases.
5) Use knowledge of database techniques to be better users with the ability to use different database systems, compare their properties, and adapt database techniques when developing software.

Academic Dishonesty

Cheating is strictly prohibited and is taken very seriously by UBC.

A guideline to what constitutes cheating:

- Labs
  - Submitting code produced by others.
  - Working in groups to solve questions and/or comparing answers to questions once they have been solved (except for group assignments).
  - Discussing HOW to solve a particular question instead of WHAT the question involves.
- Exams
  - Only materials permitted by instructor should be used in an exam.

Academic dishonesty may result in a "F" for the course and further actions by the Dean's office.

How to Excel in This Course

Attend ***every*** class:

- Read notes ***before*** class as preparation and complete the questions.
- Participate in class exercises and questions.

Attend and complete all lab assignments:

- Labs practice the fundamental employable skills as well as being for marks.

Practice on your own. Practice makes perfect.

- Do more questions than in the labs.
- Read the additional reference material and perform practice questions.
- Spend additional time programming
  - Programming assignments may take longer than a lab time. Extra time invested will payoff significantly in grades and future jobs.

Lab Assignments

There are weekly programming assignments (25% of overall grade):

- Experience applying concepts to a variety of database systems.
- You have until the week after the lab is assigned to complete it.
- No late assignments will be accepted.
- An assignment may be handed in any time before the due date and may be marked immediately by the TA.
- Lab assignments may take between 2 and 10 hours depending on the lab.

There is no scheduled lab time. TA hours will be posted.

Lab assignments are done in ***pairs***.

The programming assignments practice valuable skills for future employers.

Clicker Questions and Exercises

To promote understanding, ***10%*** of your overall grade is allocated to answering questions online and during class.

Questions may be multiple choice, short answer, or programming.

- All questions will be able to be answered both asynchronously (outside of class time) and synchronously (during class time).
- **No make-ups for forgetting to answer questions. Questions will have posted deadline for when they must be completed.**
- Canvas quizzes will be used as well as real-time polling questions (iClicker).

A student must only get 80% of questions right to get full 10%.

- That is, if there was a total of 100 marks of online questions, 80 out of 100 will give you 10%. 40 out of 100 would give you 5%.

These exercises practice questions that will be tested on exams.

Database Implementation Project

For graduate students only:

15% of your mark is for a major database development project.

Goal of the project is to experiment with new database systems or experiment with novel techniques expanding on class material.

This is ***not*** implementing a web site with a relational database like COSC 304.

Systems and Tools

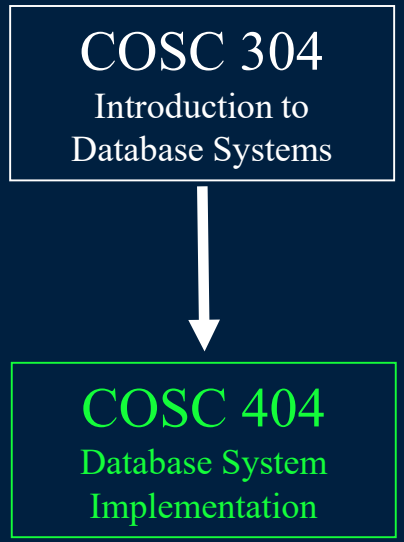
Course material, online quizzes, discussion forums and feedback, and marks are on Canvas.

Labs will use GitHub for distribution and Docker for execution.

All software used will be open source or free to install on your computer.

You must have a laptop available during class and for exams.

COSC 304 vs. COSC 404



Database Design and Programming

- Data models - ER, relational, XML, JSON
- Query languages - SQL, relational algebra
- Design project
- Database skills and techniques as a user
- How to use a DBMS ; how to build a database

Database System Implementation

- Storage and index structures
- Transaction management, concurrency control
- Query processing, recovery and reliability
- How to build a DBMS
- Non-relational systems and architectures
- How to select a DBMS

My Expectations

My goal is for you to learn the material and walk out of this course confident in your abilities:

- To understand how databases work and how to use them properly
- To write code to access a variety of database systems
- To make intelligent decisions on data allocation, indexing, and physical designs
- To use fundamental database concepts when selecting a database to use

I have high standards on the amount and difficulty of material that we cover. I expect a strong, continual effort in keeping up with readings, doing assignments, and participating in class activities.

The course is straightforward – if you do the work, you will do well.

Your mark is 70% perspiration and 30% inspiration.

Why are you here?

A) I need an upper-year Computer Science elective, and this course was all there was…

B) I liked COSC 304 (Intro. Databases) and thought this course may be okay too.

C) I am curious about what is in the database "black box".

D) I want to be a better developer and database user to improve my skills for future jobs.

E) I am interested in database research and advanced studies.

What Topic are You Most Interested In?

A) Improving my programming abilities while using databases

B) Learning how SQL queries get processed inside a database system

C) Learning how a database handles multiple users and recovers from failures

D) Experimenting with different databases like PostgreSQL, MongoDB, and MySQL

E) None of the above

What Grade are You Expecting to Get?

A) A

B) B

C) C

D) D

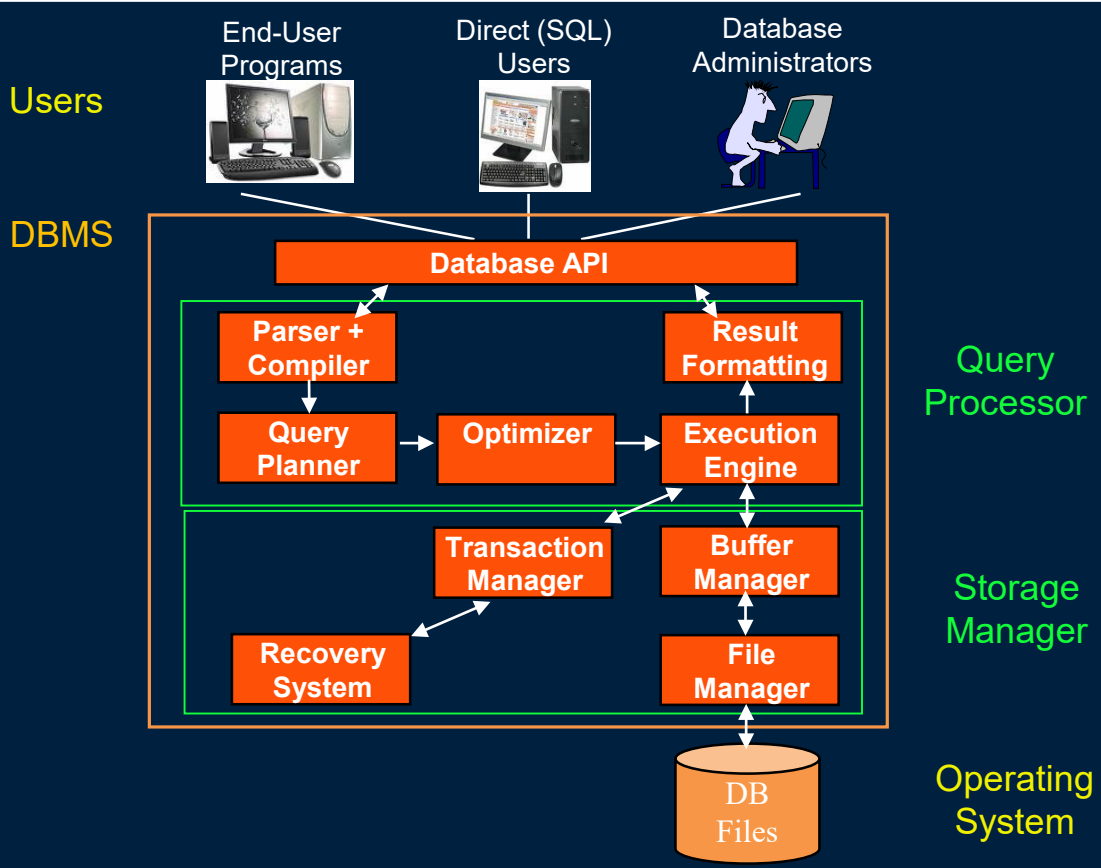
E) F

Database System Implementation Motivation

Key requirements of a database system:

- 1) Data Storage and Persistence:
  - How is data organized? Where is it located?
- 2) Query Processing:
  - How does the user query the data? How efficient is it?
- 3) Transactions, Consistency, and Reliability:
  - What happens if the computer crashes while the user is updating data?
- 4) Concurrency:
  - Can multiple users access the data at the same time? What happens if multiple users update the same data item?
- 5) Security:
  - How do you verify the user has access to the data?
- 6) Scalability:
  - How do you handle Big Data and lots of users?

Traditional Database Architecture



Database Architectures
Not "One Size Fits All"

Relational databases (RDBMS) are still the dominant database architecture and apply to many data management problems.

- $35 billion annual industry (NoSQL about $1 billion)

However, recent research and commercial systems have demonstrated that "one size fits all" is not true. There are better architectures for classes of data management problems:

- Transactional systems: In-memory architectures
- Data warehousing: Column stores, parallel query processing
- Big Data: Massive scale-out with fault tolerance
- "NoSQL": simplified query languages/structures for high performance, consistency relaxation

COSC 304 Review Question

Question: What was the acronym used to describe transactional processing systems?

A) TP

B) OLAP

C) OLTP

D) DBMS

Research Question

Question: What company is the largest database software vendor by ***sales volume***?

A) Microsoft

B) Oracle

C) IBM

D) Google

Database Architectures: NoSQL vs Relational

"NoSQL" databases are useful for several problems not well-suited for relational databases with some typical features:

- **Variable data:** semi-structured, evolving, or has no schema
- **Massive data:** terabytes or petabytes of data from new applications (web analysis, sensors, social graphs)
- **Parallelism:** large data requires architectures to handle massive parallelism, scalability, and reliability
- **Simpler queries:** may not need full SQL expressiveness
- **Relaxed consistency:** more tolerant of errors, delays, or inconsistent results ("eventual consistency")
- **Easier/cheaper:** less initial cost to get started

NoSQL is not really about SQL but instead developing data management architectures designed for scale.

- NoSQL – "Not Only SQL"

Example NoSQL Systems

MapReduce – useful for large scale, fault-tolerant analysis

- Hadoop, Hive, Spark

Key-value stores – ideal for retrieving specific items from a large set of data (architecture like a distributed hash table)

- high-scalability, availability, and performance but weaker consistency and simpler query interfaces
- Cassandra, Amazon Dynamo, Google BigTable, HBase

Document stores – similar to key-value stores except value is a document in some form (e.g. JSON)

- MongoDB, CouchDB

Graph databases – represent data as graphs

- Neo4J

Survey Question

Question: Have you used any database system besides MySQL and Microsoft SQL Server used in COSC 304?

A) Oracle

B) MongoDB

C) PostgreSQL

D) More than two different databases used

E) No other databases used

Why this Course is Important

DBMS technology has applications to any system that must store data persistently and has multiple users.

- Even if you will not be building your own DBMS, some of your programs may need to perform similar functions.
- The core theories expand on topics covered in operating systems related to concurrency and transactions.

A DBMS is one of the most sophisticated software systems.

- Understanding how it works internally helps you be a better user of the system.
- Understanding of database internals is valuable if you will perform database administration duties or be responsible for deciding on a database architecture for an application.

Database technology is a key component of our IT infrastructure that will continue to require innovation in the future.


UBC
THE UNIVERSITY OF BRITISH COLUMBIA