CS 4092 Database Design and Development (DDD)

00: Course Overview

Seokki Lee

Slides are adapted from:

Database System Concepts, 7th Ed. ©Silberschatz, Korth and Sudarshan

Who Am I?

Seokki Lee

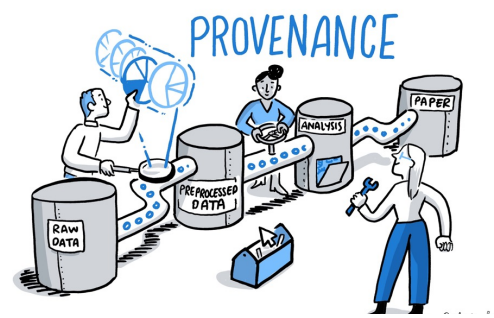
Assistant Professor

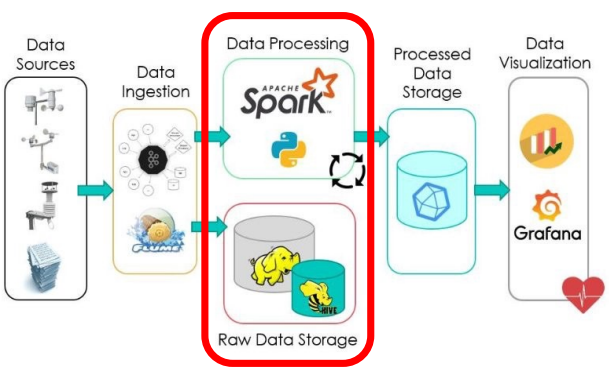
College of Engineering and Applied Science

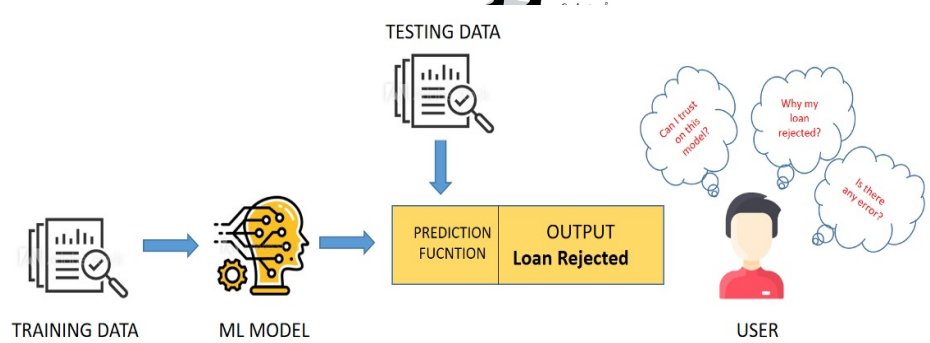
Department of Computer Science

Research Interest

- Efficient complex query processing and provenance capture over big data
- Generate concise and meaningful explanations
- Database and data provenance applications
  - Explainability of machine learning (ML)
  - Exploring provenance for explainable data sharing
  - Data visualization with enhanced explanations and recommendations







Interested?

Seokki Lee

Assistant Professor

College of Engineering and Applied Science

Department of Computer Science

[Contact Information]

Office: Rhodes 885

Email: seokki.lee@uc.edu

Phone: 513-556-5795

Homepage: https://researchdirectory.uc.edu/p/lee5sk

Group Webpage: https://shek21.github.io/

What is DBMS?

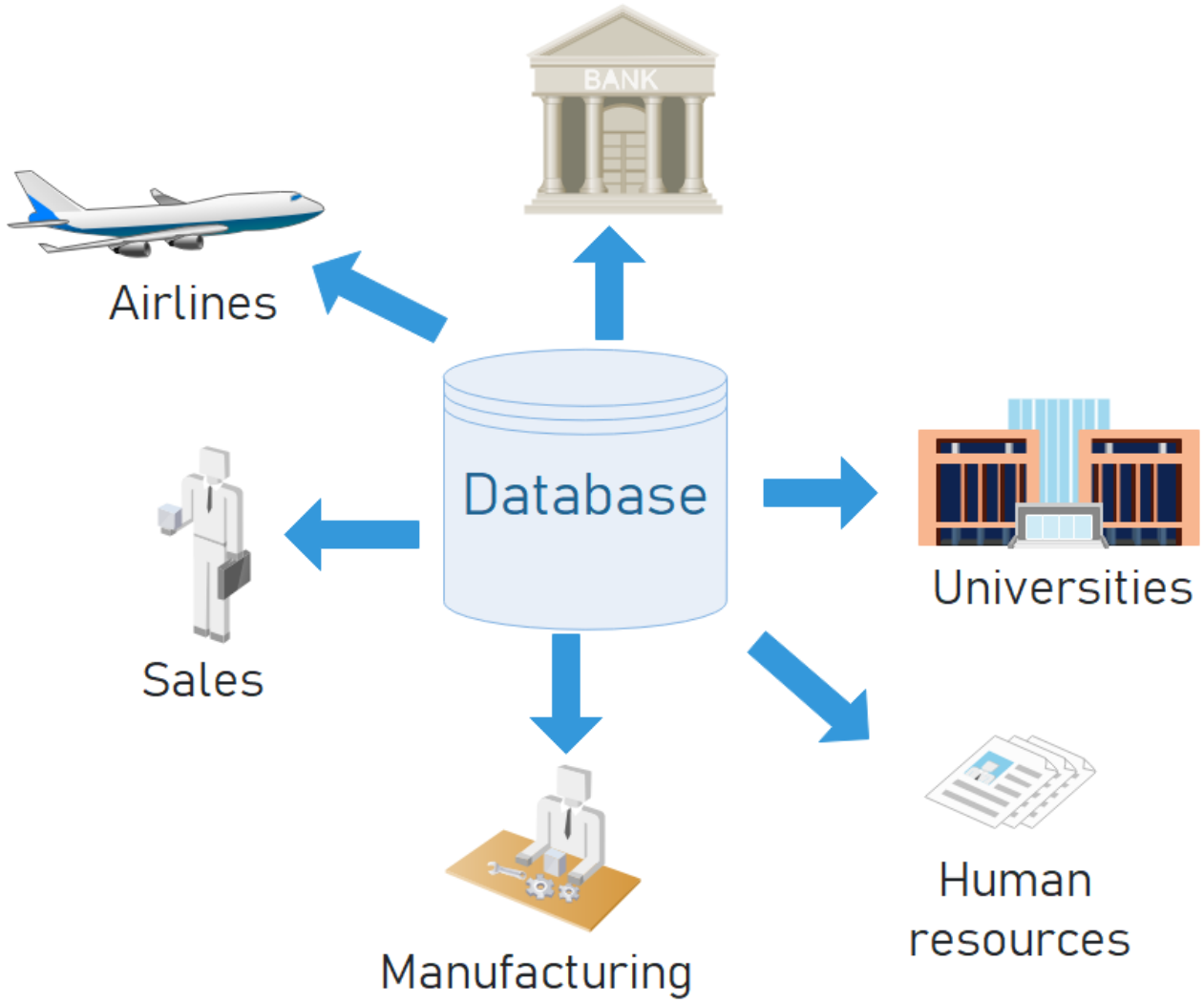
- Collection of interrelated data and a set of programs to access those data
- Providing a way to store and retrieve database information

Why are Databases important?

- **What do Databases do?**
  - Provide persistent storage
  - Efficient declarative access to data → Querying
  - Protection from hardware/software failures
  - Safe concurrent access to data

Database Applications Examples

Database Applications Examples



https://www.cseworldonline.com/dbms-tutorial/dbms-application.php

Database Applications Examples

- **Enterprise** Information
  - Sales: customers, products, purchases
  - Accounting: payments, receipts, assets
  - Human Resources: Information about employees, salaries, payroll taxes.
- **Manufacturing**
  - Management of production, inventory, orders, supply chain.
- **Banking** and **finance**
  - Customer information, accounts, loans, and banking transactions.
  - Credit card transactions
  - Finance: sales and purchases of financial instruments (e.g., stocks and bonds; storing real-time market data
- **Universities**
  - Registration, grades

Database Applications Examples

- **Airlines**
  - Reservations, schedules
- **Telecommunication**
  - Records of calls, texts, and data usage
  - Generating monthly bills, maintaining balances on prepaid calling cards
- **Web-based** services
  - Online retailers: order tracking, customized recommendations
  - Online advertisements
- **Document** databases
- **Navigation** systems
  - For maintaining the locations of varies places of interest along with the exact routes of roads, train systems, buses, etc.

Popular Database Systems

Relational Database

NoSQL / Distributed Database

Popular Database Systems

Relational Database

ORACLE

PostgreSQL

Microsoft SQL Server

MySQL

TERADATA

SQLite

IBM DB2

NoSQL / Distributed Database

APACHE HBASE

mongoDB

cassandra

GOOGLE FILE SYSTEM

hadoop HDFS

HIVE

snowflake

Why are Databases Interesting?

- **Pragmatic perspective**
  - Background in databases makes you competitive in the job market ☺
- Database research has a **strong systems aspect**
  - Hacking complex and large systems
  - Low-level optimization
    - Cache-conscious algorithms
    - Exploit modern hardware
- Databases have a strong **theoretical foundation**
  - Complexity of query answering
  - Expressiveness of query languages
  - …

Why are Databases Interesting?

- **Connection** to many CS fields
  - Distributed systems
    - Getting more and more important
  - AI and machine learning
    - Data mining
  - Operating and file systems
  - Hardware
    - Hardware-software co-design

What We Cover

- **Topics**
  - Formal relational languages
  - Structured query language (SQL)
  - Database design
    - ER model
    - Normal forms
  - Transaction management

Course Objectives

- Studying query processing techniques by writing SQL queries
- Learning about relational language
- Understanding and solving abstract relational algebra (RA) problems and learning the relationship between SQL and RA.
- Constructing Entity-Relationship (ER) diagrams and performing normalization and normal forms
- Learning fundamental concepts in DBMS for transactions, concurrency control, and recovery from failure including additional topics such as indexing and functional dependencies
- Implementing an application using a commercial or open-source relational database management system (DBMS), e.g., Oracle, Postgresql, etc.

CS5151/6051 Database Theory

- **Topics**
  - Intermediate SQL in-depth
  - Advanced SQL
    - Window functions
    - OLAP queries
  - Datalog (logic-based declarative query language)
  - Database architecture
    - Parallel and distributed databases

CS7071 Advanced Database Management

- **Topics**
  - Storage and buffer management
  - Indexing
  - Query optimization and query execution
  - Concurrency control and recovery

- **Functional Components** of Database Engine
  - The storage manager
  - The query processor component
  - The transaction management component

Course, TA, and Faculty Information

- **Course Webpage**: Canvas
  - Syllabus and course materials
  - Announcements and discussions
  - Assignments and project
  - Grades

Course, TA, and Faculty Information

- **Course Webpage**: Canvas

2241-1_20CS4092001

2241 Courses

Home
Discussions
Grades
People
Pages
Files
Syllabus
Collaborations

(24SS-Full) DBASE DESIGN/DEVEL (001)

View Course Stream
View Course Calendar
View Course Notifications

To Do

Nothing for now

Instructor

Seokki Lee (https://researchdirectory.uc.edu/p/lee5sk)

Email: seokki.lee@uc.edu

Office: Rhodes 885

Office hours: Monday 2:30 - 4 pm (or by appointment)

TA

Shemon Rawat

Email: rawatsn@mail.uc.edu

Location: Rhodes 800B

Office hours: Tuesdays 3 - 4:30 pm (or by appointment)

Workload

- **Exams (45%)**
  - Midterm (20%)
  - Final exam (25%)
- **Course Project (25%)**
  - ER model (10%)
  - Database design (5%)
  - Implement an application (10%)
- **Take-Home Assignments (20%)**
  - Four homework (5% each)
- **In-class activities (10%)**
  - Four activities to participate

Grading

Letter Grade	Score
A	[89,100]
A-	[84,89)
B+	[79,84)
B	[74,79)
B-	[69,74)
C+	[65,69)
C	[61,65)
C-	[57,61)
D+	[53,57)
D	[49,53)
D-	[45,49)
F	[0,45)

Course Project

- **Forming groups**
  - In group of 3
  - Finding your team members **ASAP**
    - If you do not, contact me or TA
  - Record your team in the **discussion board!**
    - Under the discussion namely “Forming Groups”
    - Once your team is confirmed, add the group information in the excel sheet provided through the link

- **Submission through GitLab repositories**
  - Create an account of GitLab using your UC email
  - Once all groups are formed, you will receive an invitation to join the repository for your group.
  - All required portions of the project must be submitted to the repository.
  - Self-study: https://docs.gitlab.com/ee/tutorials/

Course Project

- **Important dates for the project**
  - Project release (Feb 1)
  - ER model due (Mar 21)
  - Design database and DDL scripts due (Apr 2)
  - Implementation and recorded demo due (Apr 18)

- **Contribution**
  - Every student **MUST** contribute to **every component** of the project! The level of contributions should be clearly stated in each submission.
  - **Don't let others freeload on your hard work!**
    - Inform me or TA immediately

Fraud and Late Assignments

- **All work MUST be original!**
  - **Cheating = 0 points**
  - Any behavior defined in Student Code of Conduct is considered academic misconduct
    - All HW, Project, and Exams are included.
    - Your responsibility to have a look at the document

- **Late policy**
  - -10% for every 3 days
  - Only exception: **Health issue with an official proof**

PostgreSQL

- **PostgreSQL** is an open-source object-relational database system that adheres to the client-server model.

- **Cluster**
  - A collection of databases that is managed by a single instance of a running database server

- **Server**
  - A server process managing the database files, accepting connections to the database from the client, and performing database actions sent from the client

- **Client**: used to interact with the server
  - GUI: pgAdmin (https://www.pgadmin.org/)
  - CLI: built-in command line interface using **psql**

In-class Activities and Examples

- **Install a working version of PostgreSQL client and server (local)**
  - Using a Postgres package
    - Client: https://www.pgadmin.org/download/
    - Server: https://www.postgresql.org/download/
  - Using a source code
    - https://www.postgresql.org/ftp/source/

- **Get used to it**
  - Create a cluster
  - Start/stop the server
  - CLI: run **psql** to check whether you can connect to the server running

Last But Not Least

- Please contact the instructor or TA **immediately**
  - Concerning your performance
  - Need any additional support
  - …

Reading Materials

- Silberschatz, Korth, and Sudarshan, **Database System Concepts**, 7th Edition, McGraw Hill, 2019
- Garcia-Molina, Ullman, and Widom, **Database Systems: The Complete Book**, 2nd Edition, Prentice Hall, 2008