# Wk1_W: Introduction

CSCI3170 24T1

Introduction to Database Systems

# A Gentle Revision

- What do you notice?
- How many observations/ records are there?

| sid | sname | asg1 | asg2 | asg3 | exam |
|---|---|---|---|---|---|
| 1155100001 | Rick | 90 | 100 | 95 | 88 |
| 1155100002 | Ryan | 100 | 80 | 85 | 76 |
| 1155100003 | Bruno | 100 | 85 | 75 | 82 |
| 1155100004 | Alice | 70 | 55 | 80 | 64 |
| 1155100005 | Bob | 55 | 30 | 35 | 42 |

# Tabular Formats

- (data table/ data frame) - two dimensional structure
  - "Structured data is *when data is in a **standardized format**, has a well-defined structure"*


variable
(attribute/field/feature)


observation
(enitiy/record)


Value
(data)

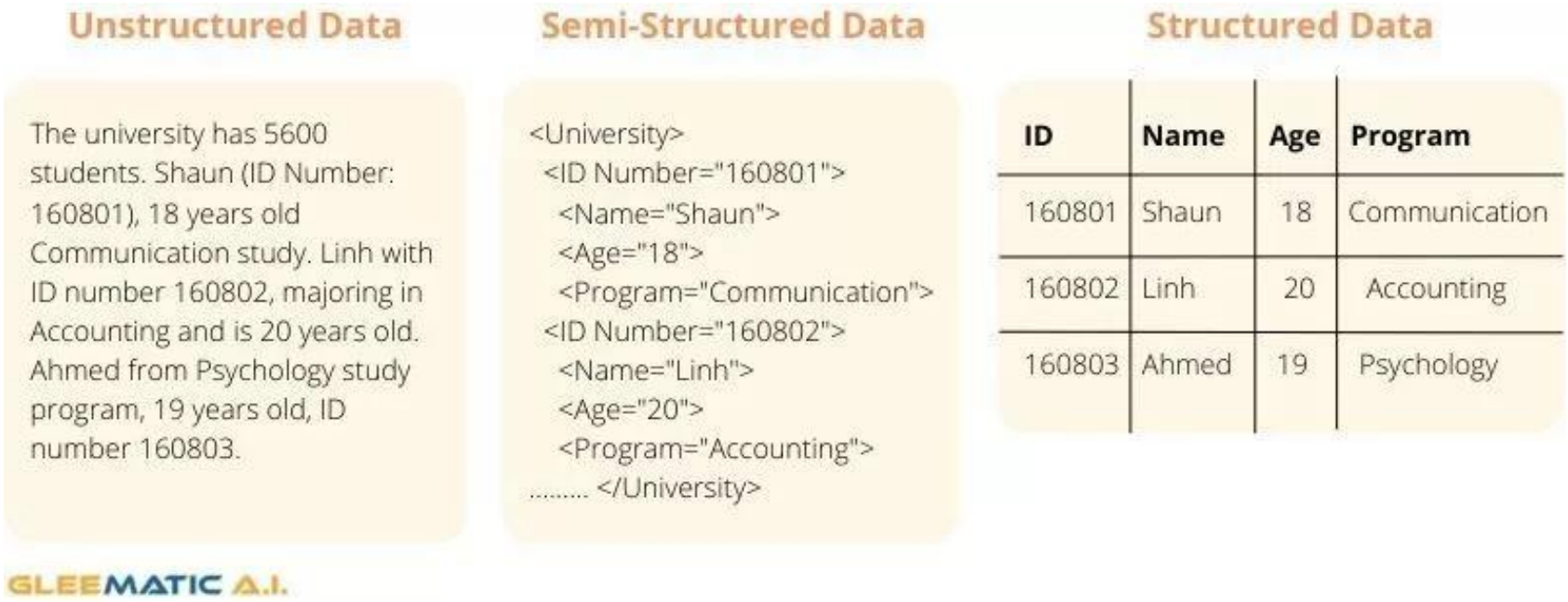
# A Family of Structured Data

- *"Structured data is highly specific and is stored in a predefined format"*
  - Very easily used by machine learning (ML) algorithms

https://www.ibm.com/cloud/blog/structured-vs-unstructured-data

# A Family of Structured Data

- There are three type of data




https://cdn-dlofh.nitrocdn.com/TASgNozwGfEfVHakzcSlOaFmdxhUvEBv/assets/static/optimized/rev-f0e30d7/wp-content/uploads/2021/12/Example-Structured-Unstructured-Semi-Structured-Data-e1638423518970.jpg

# Why Study Databases?

- Most significant modern computer application rely on huge quantities of data.
- Data will always have to be:
  - stored (typically on a disk device)
  - manipulated/accessed (efficiently, effectively)
  - shared (by many users, concurrently)
  - transmitted (all around the Internet)
- Red points are handled by databases; brown by networks.

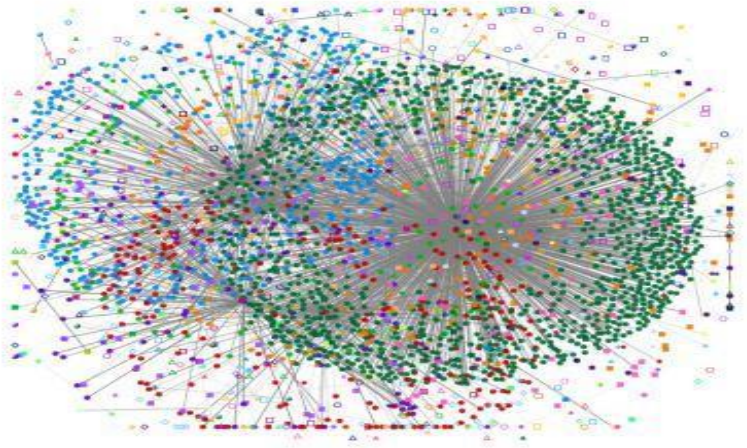
# Big Data

- There are many different types of data: text data, image data, audio data, video data, etc.
- The amount of data grows very fast.
  - Zettabyte 1,180,591,620,717,411,303,424 ($2^{70}$) byte
  - Exabyte 1,152,921,504,606,846,976 ($2^{60}$) byte
  - Petabyte 1,125,899,906,842,624 ($2^{50}$) byte
  - Terabyte 1,099,511,627,776 ($2^{40}$) byte
  - Gigabyte 1,073,741,824 ($2^{30}$) byte
  - Megabyte 1,048,576 ($2^{20}$) byte
  - Kilobyte 1,024 ($2^{10}$) byte
  - Byte 1 byte

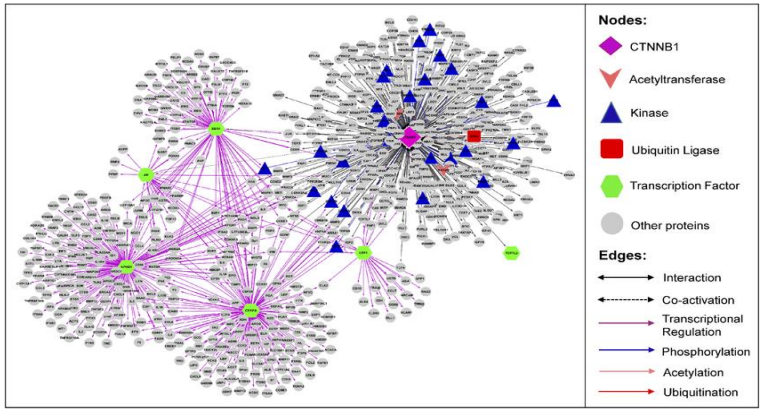
# The Vs of Big Data

- **Volume:**
  - The amount of data matters.
- **Variety:**
  - The many types of data that are available.
- **Velocity:**
  - Desire for data to be received and acted on quickly.
- **Veracity:**
  - Accuracy of your data, how well it conforms to facts.
- **Value:**
  - Data is of no use until that value is discovered

# Internet of Things

Web Graphs




Biological Networks

Social Networks

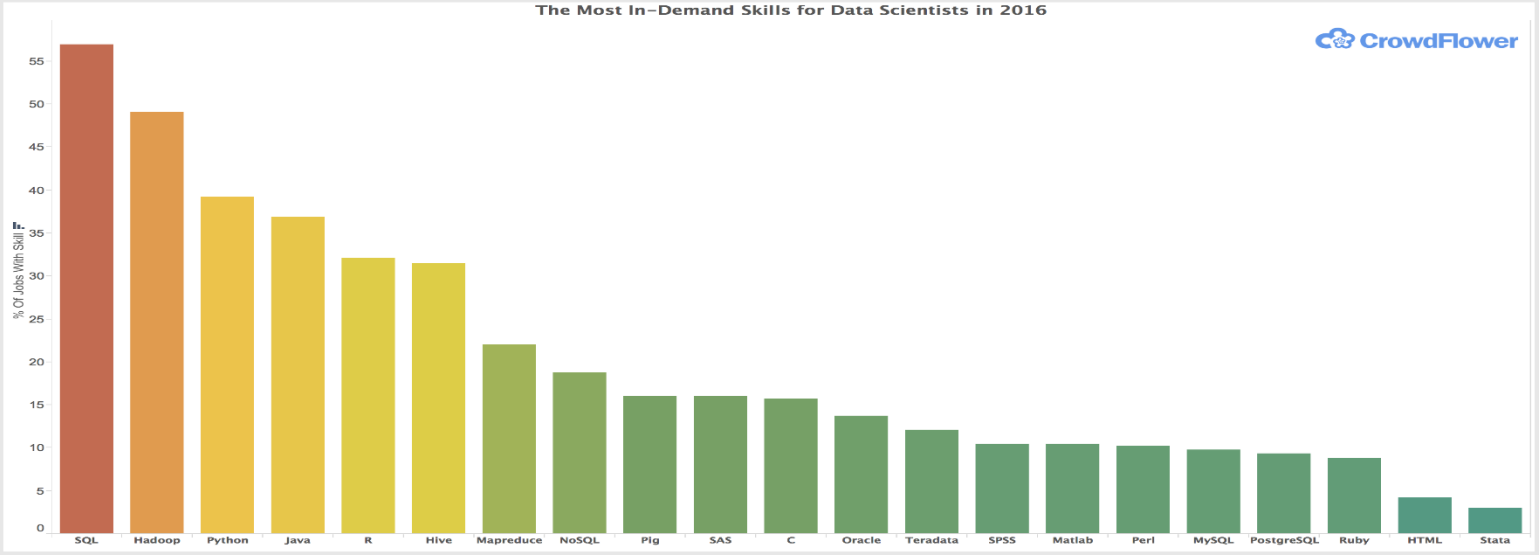
# Types of Data (Revisited)

- *Data that is* **Unstructured**
  - No need to pre-define the data
  - Requires expertise to prepare the data due to its non-formatted nature
  - Can be a combination of various data

- *Data that is* **Structured**
  - Stored with a rigid and strict schema
  - Can be organized into relational databases

# Data Science Skills Employers Want

- Writing SQL Queries & Building Data Pipelines (KDnuggets 2022)
  - "Learning how to write robust SQL queries and scheduling them on a workflow management platform like Airflow will make you extremely desirable as a data scientist, hence why it's point #1."






https://www.datanami.com/2016/01/07/what-data-science-skills-employers-want-now
https://www.kdnuggets.com/2021/10/11-most-practical-data-science-skills-2022.html

# Some Resources

1. https://www.coursera.org/articles/sql-skills
2. https://www.knowledgehut.com/blog/database/database-future
3. https://blogs.451research.com/information_management/?s=relational+database
4. https://db-engines.com/en/ranking

# Files vs. DBMS

- **File based system:**
  - Contains various information on a storage device
  - Files (such as txt files, object files, source files)

First issue:

- *Data redundancy and inconsistency*
  - Multiple file formats, duplication of information in different files
- Lacking support for expressive queries
  - we need additional programs to answer different queries.

# Why Database Systems (1)

*Drawbacks of using file systems to store data:*

– *Integrity problems*

- Integrity constraints become “buried” in program code rather than being clearly kept and stated
- Hard to add new constraints or change existing ones

# Why Database Systems (2)

*Drawbacks of using file systems to store data (cont.):*

- *Atomicity of updates*
  - Failures may leave the data in an inconsistent state
- *Hard to allow concurrent access by multiple users*
  - Uncontrolled concurrent accesses can lead to inconsistencies

# Why Database Systems (3)

*Strength of DBMS to store data (cont.):*

- Data Independence *(to be covered later)*
  - Application program should not be exposed to details of data representation and storage.
  - DBMS provides an abstract view of the data that hides such details.

- **Moreover, database systems offer solutions to all the aforementioned problems**

# Other Adv. of a DBMS (1)

- Data Integrity and Security
  - DBMS can enforce access controls that govern what data is visible to different classes of users.

- Concurrent Access and Crash Recovery
  - DBMS schedules concurrent access such that users can think of the data as being accessed by only one user at a time.
  - Protects users from the effects of system failures.

# What is a DBMS?

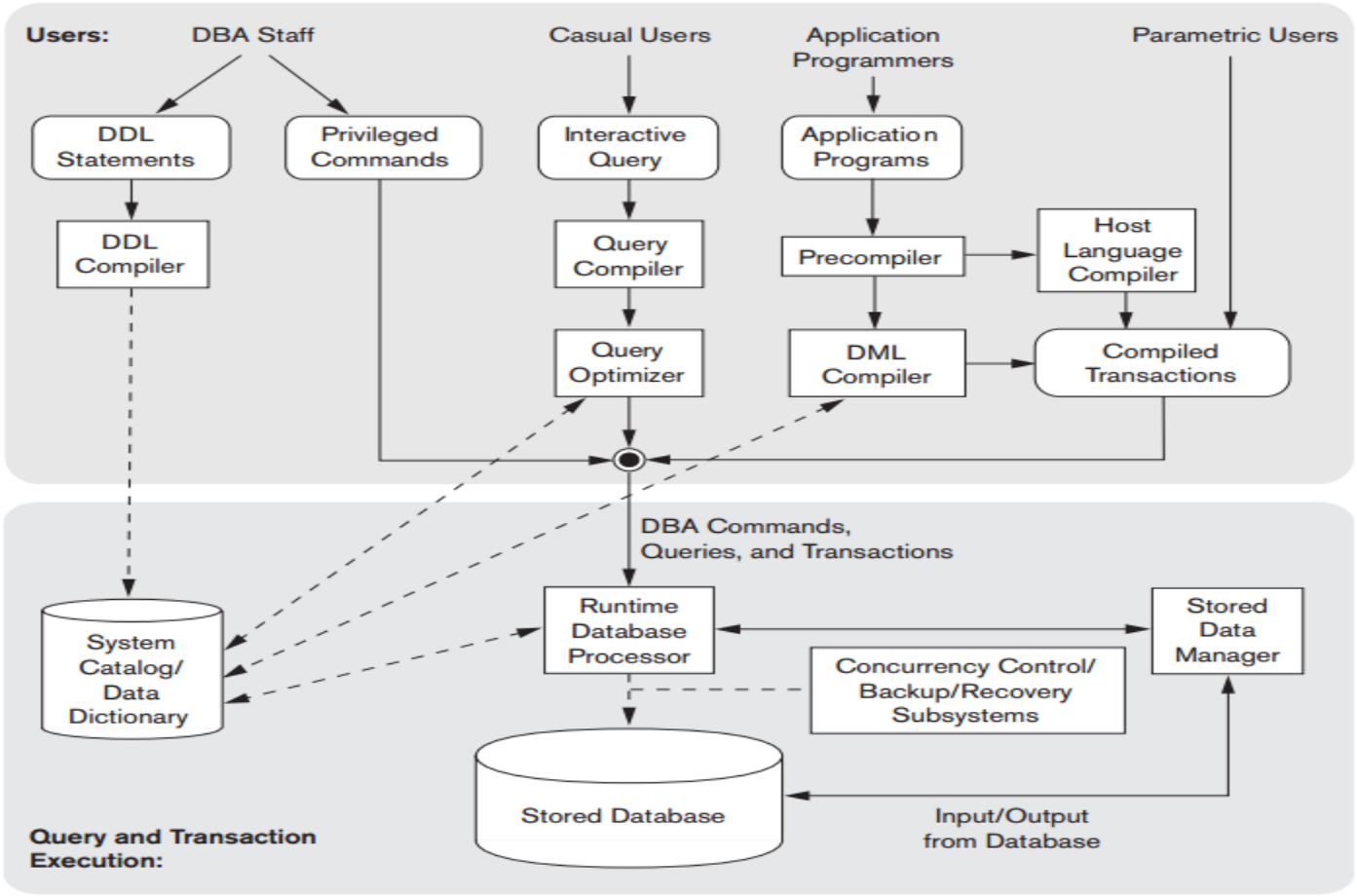
- A database management system (DBMS) is a **software package** designed to assist in maintaining and utilizing large collections of data.

# Part 1: What is a Database?

- A collection of data.
  - Typically describing the activities of one or more related organizations.
- Models real-world enterprise
  - Entities
    - e.g. students, professors, courses, classroom
  - Relationships between entities
    - e.g. student's enrollment in courses, professor teaching courses, and use of room for courses.

# Database System

- Component modules of a DBMS and their interactions.



# People who deal with databases

- Database Administrator (DBA)
  - The ones with central control over the database(s)
  - Responsible for the following tasks:
    - Schema definition/modification
    - Storage structure definition/modification
    - Authorization of data access
    - Integrity constraints specification
    - Database Recovery

# People who deal with databases

- Sophisticated users
  - Those who form request in database query languages.

- Naive users
  - Those who invoke application programs that have been written previously e.g., transfer fund between accounts.

# People who deal with databases

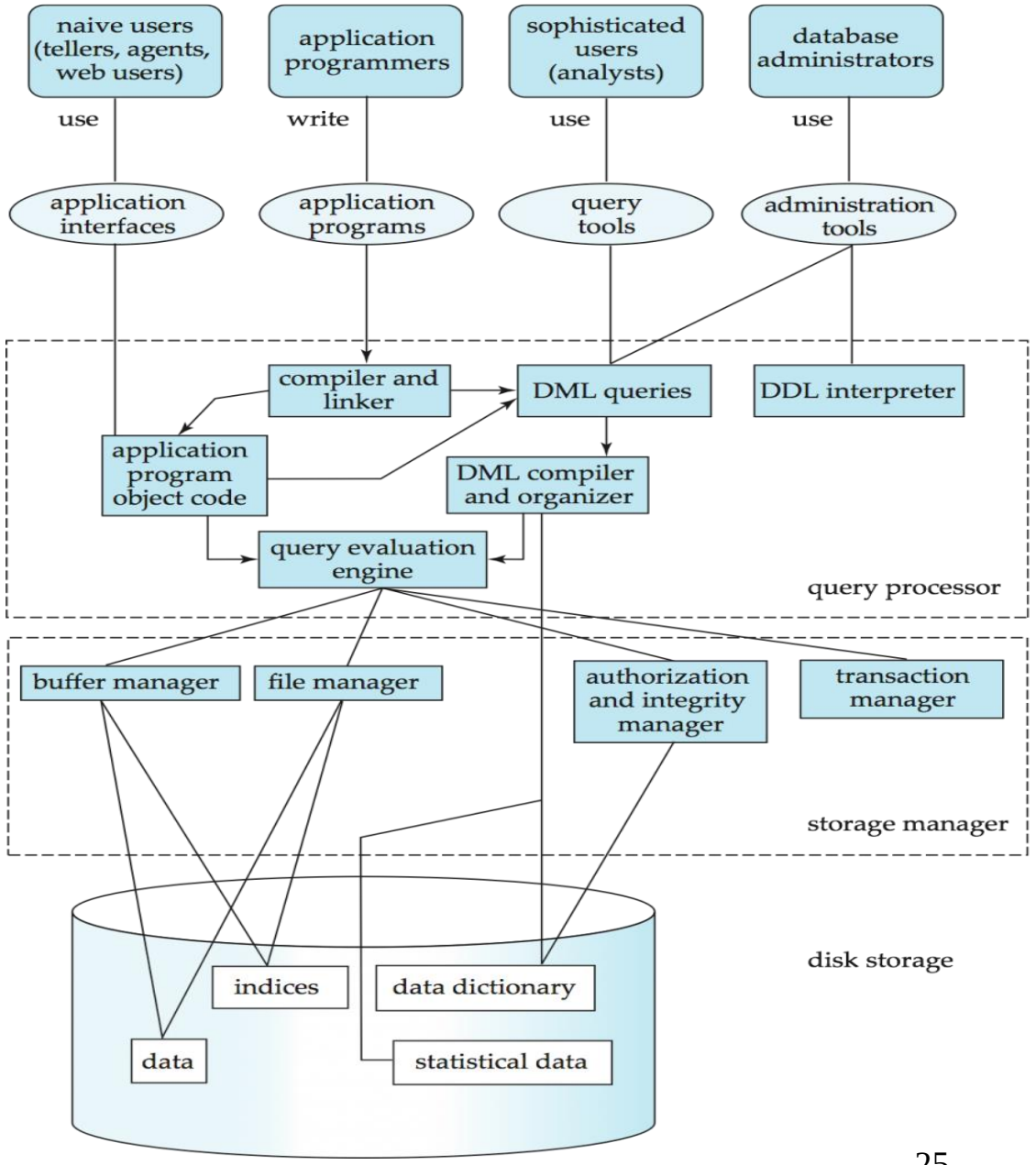
- Application Programmers
  - Those who write programs (Cobol, C, Java) with embed DML calls or develop packages with software tools provided by the DBMS vendor.
  - Example: programs that generates payroll checks, transfer funds between accounts etc.

- Systems Analyst
  - Determine end users requirements
  - Develop specs. for canned transactions and reports
  - May also take part in database design

# Revisit Adv. of a DBMS

- Data Administration
  - DBMS Provides facilities for
    - Organizing data representation to minimize redundancy
    - fine-tuning the storage of data to make retrieval efficient
- Reduced Application Development Time
  - DBMS supports important functions that are common to many applications
    - High-level interface

# Architecture of a DBMS



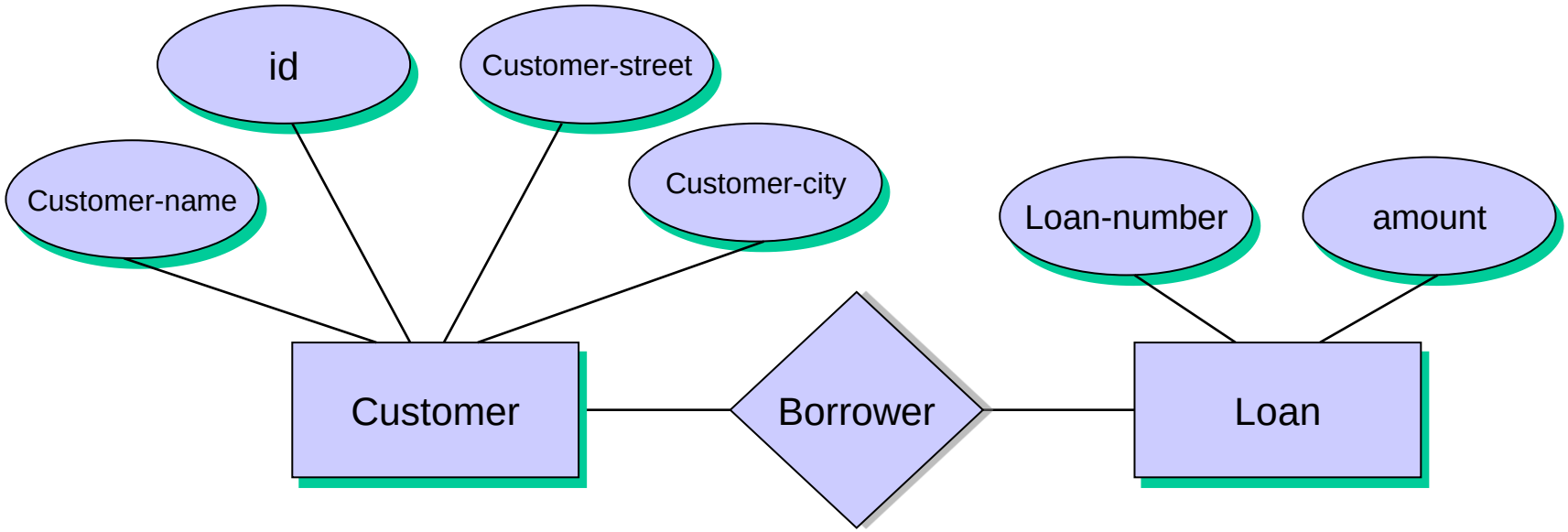

Source: Database System Concept, 6th edition

# In Conclusion

- Hopefully, you now know…
  - course structure
  - who to contact (where to seek help before emailing me)
  - how you're assessed and scored
  - the database applications around you
  - what goes on in databases (and is interested)

- Next Week: ER Diagram, (some) Data Modelling

# Next Week




An example of an ER data model

# Data Model

- **Data model:**
  - the concepts used to describe the allowed structure of a database. *i.e., the structure of the data.*

- **3 Levels** of Data Models:
  1. High-level or conceptual *e.g., ER model – concerns entities, attributes and relationships* (Next Week)
  2. Implementation or record-based *e.g., Relational, Network, Hierarchical*
  3. Low-level or physical

# Data Model (cont) Concepts

- **Database Schema**:
  - *a* formalism of the data model, *the* structural description of what information will database holds
- **Database Instance** *(or State):*
  - *any combination of actual information populated* in the database at a particular time.

- Checking understanding:
  1. We define a database by specifying its schema.
  2. The state is then an empty instance of the schema.
  3. After this, each change in state is an update to the instance.

# Semantic Data Model

- A semantic data model is a very abstract, *high-level* data model.
  - For a user to come up with a good initial description of the data in an enterprise.
- Entity-relationship (ER) model
  - A widely used semantic data model.
  - Allows us to pictorially denote entities and the relationships among them.

# Relational Data Model

- Most DBMS today are based on the relational data model.
- Relation
  - the central data description construct in this model.
  - It can be thought of as a set of records.
  - A table with rows and columns.
    - Row – a record
    - Column – field, attribute.

| sid | name | login | age | gpa |
|---|---|---|---|---|
| 53666 | Jones | Jones@cs | 18 | 3.4 |
| 53688 | Smith | Smith@ee | 18 | 3.2 |
| 53650 | Smith | Smith@math | 19 | 3.8 |
| 53831 | Madayan | Madayan@music | 11 | 1.8 |
| 53832 | Guldu | guldu@music | 12 | 2.0 |

An example of a student relation

- A description of data in terms of a data model is called a schema.
- The schema of the above table is

  *Students(sid: string, name: string, login: string, age: integer, gpa: real)*

# End

- Feedback is welcomed