CSCB20
Introduction to Databases and Web Application

Week 0 - Introduction

Dr. Purva R Gawde

Thanks to Dr. Anna Bretscher for the material in this set of slides

About CSCB20

- Databases:
    - terminology and applications
    - creating, querying and updating databases
    - the entity-relationship model for database design
- Web App Development
    - Web documents and applications: static and interactive documents
    - Web servers and dynamic server-generated content
    - Web application development and interface with databases
- Required Background
    - Some experience with programming in an imperative language such as Python, Java, or C.
    - You may not take this course after - or concurrently with - any C- or D-level CSC course.
- **Exclusion** This course may not be taken after - or concurrently with - any C- or D-level CSC course.

Course Layout

- Database Design: 5-6 weeks
- Web Application Design: 6-7 weeks
- Lectures: 2 hours per week
  - In Lecture: we will cover worksheets and practise the course content posted previous week
- Tutorials: 1 hour per week – start in next week
- Each Week Tasks:
  - Before class: Carefully read the course content posted previous week on Quercus
  - In class: Recap of the concepts and solve worksheets and ask questions
  - In tutorial: More practise with TAs

Course Work

- Term Work
  - 3 Assignments (33%)
    - Assignment 1 - 8%
    - Assignment 2 - 10%
    - Assignment 3 - 15%
- Tutorial attendance
  - Attend at least 7 tutorials for 2%
- Exams
  - Midterm 25%
  - Final Exam 40%

How will you be evaluated?

- Final needs to receive at least 35% for you to pass the course
- Missed term test weight gets shifted to the final exam
    - Try your best not to miss exam!!!
- Instructors reserve the right to define the format of make up exam. The format includes but not limited to closed-book written exam or oral exam with live coding.

How Do I Stay Informed?

- Come to class!
- Join Piazza and check often.
- Check the calendar for due dates of term work.
- Check your utoronto email
  - this is where I will send out emails to the class.

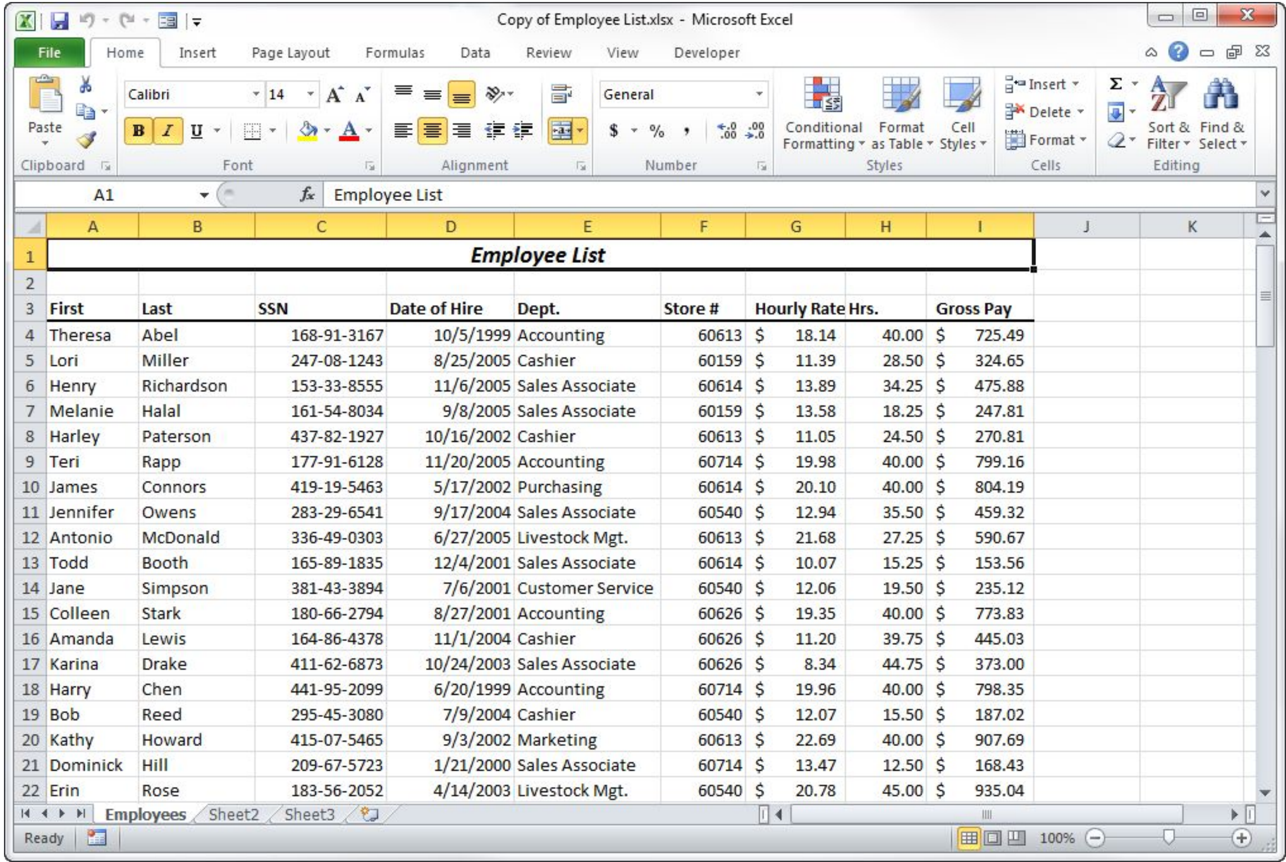
Today

- Databases:
  - What?
  - Where?
  - Why?
  - When?
  - How?
- Web application Introduction
- Back to databases
  - Introduction to Relational Algebra
    - Procedural query language
  - Terminology

Earliest Database

Why Do We Need A Database?

- Store the Data
- Solution:
  - Spreadsheets



Employee List

First	Last	SSN	Date of Hire	Dept.	Store #	Hourly Rate	Hrs.	Gross Pay
Theresa	Abel	168-91-3167	10/5/1999	Accounting	60613	$ 18.14	40.00	$ 725.49
Lori	Miller	247-08-1243	8/25/2005	Cashier	60159	$ 11.39	28.50	$ 324.65
Henry	Richardson	153-33-8555	11/6/2005	Sales Associate	60614	$ 13.89	34.25	$ 475.88
Melanie	Halal	161-54-8034	9/8/2005	Sales Associate	60159	$ 13.58	18.25	$ 247.81
Harley	Paterson	437-82-1927	10/16/2002	Cashier	60613	$ 11.05	24.50	$ 270.81
Teri	Rapp	177-91-6128	11/20/2005	Accounting	60714	$ 19.98	40.00	$ 799.16
James	Connors	419-19-5463	5/17/2002	Purchasing	60614	$ 20.10	40.00	$ 804.19
Jennifer	Owens	283-29-6541	9/17/2004	Sales Associate	60540	$ 12.94	35.50	$ 459.32
Antonio	McDonald	336-49-0303	6/27/2005	Livestock Mgt.	60613	$ 21.68	27.25	$ 590.67
Todd	Booth	165-89-1835	12/4/2001	Sales Associate	60614	$ 10.07	15.25	$ 153.56
Jane	Simpson	381-43-3894	7/6/2001	Customer Service	60540	$ 12.06	19.50	$ 235.12
Colleen	Stark	180-66-2794	8/27/2001	Accounting	60626	$ 19.35	40.00	$ 773.83
Amanda	Lewis	164-86-4378	11/1/2004	Cashier	60626	$ 11.20	39.75	$ 445.03
Karina	Drake	411-62-6873	10/24/2003	Sales Associate	60626	$ 8.34	44.75	$ 373.00
Harry	Chen	441-95-2099	6/20/1999	Accounting	60714	$ 19.96	40.00	$ 798.35
Bob	Reed	295-45-3080	7/9/2004	Cashier	60540	$ 12.07	15.50	$ 187.02
Kathy	Howard	415-07-5465	9/3/2002	Marketing	60613	$ 22.69	40.00	$ 907.69
Dominick	Hill	209-67-5723	1/21/2000	Sales Associate	60714	$ 13.47	12.50	$ 168.43
Erin	Rose	183-56-2052	4/14/2003	Livestock Mgt.	60540	$ 20.78	45.00	$ 935.04

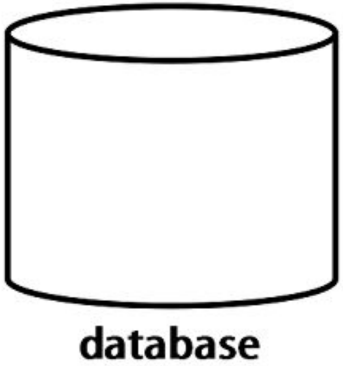
- Problems?
  - Scrolling 10,000 employees
  - Access to all
  - ....

When do we actually need a database?

- To fix potential *problems* such as:
    - Size
    - Accuracy
    - Security
    - Redundancy
    - Importance
    - Overwriting

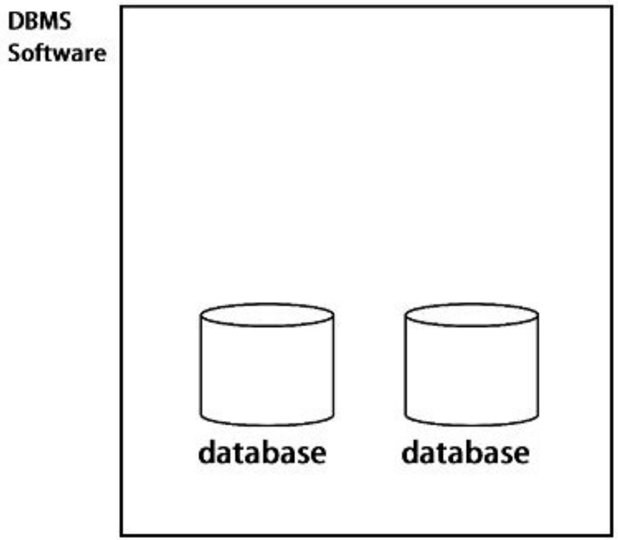
What...

- Is a *Database*?
  - A collection of interrelated data.
  - The data is relevant to an enterprise.



What...

- Is a *Database Management System (DBMS)*?
  - A database and A set of programs to access the database
  - Provides a way to store and retrieve database information.
  - Must be convenient and efficient.



Where..

- Enterprise Information:
  - Sales
  - Accounting
  - Human Resources
  - Manufacturing
  - Online Retailers
- Banking and Finance:
  - Banking
  - Credit Card Transactions
  - Finance
- Other Applications?
  - Universities
  - Airlines
  - Telecommunications

Types of DBMS

- Relational Database Management Systems (RDBMS)
- Hierarchical Database Systems
- Network Database Systems
- Object-Oriented Database Systems
- NoSQL Database Systems

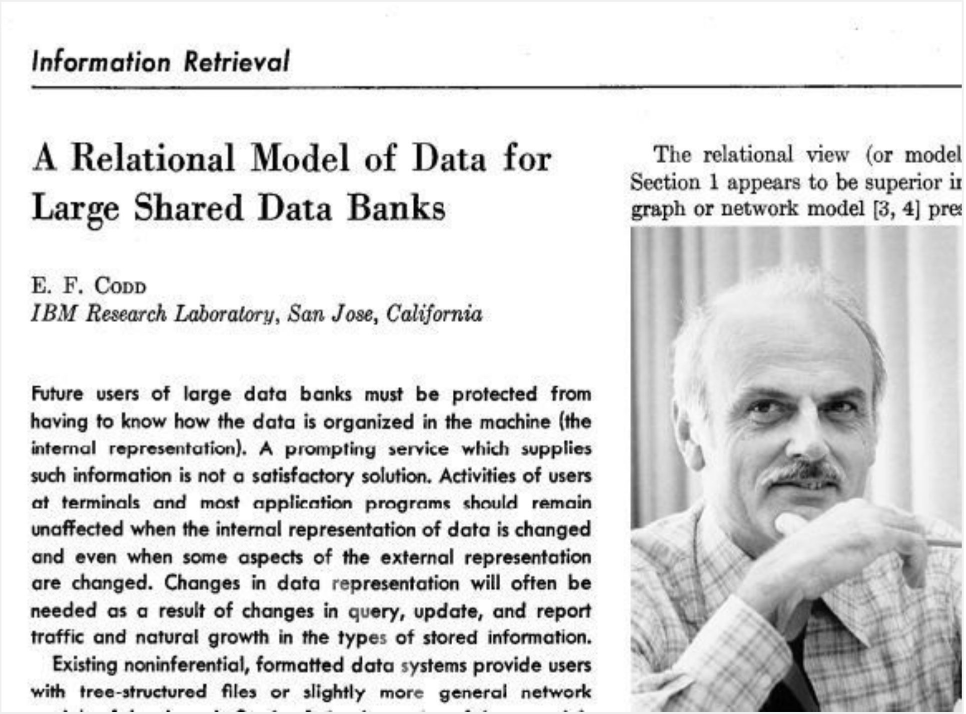
- Why RDBMS?
    - Commonly used
    - Similar principles across platforms
- Examples of RDBMS: PostgreSQL, Oracle, MySQL, SQL Server, SQLite, DB2, ...etc.

How did we get here?

A brief history of Databases

1960’s and 1970’s: The beginning

- Before 1960’s:
  - Punched cards to input data in magnetic tapes
- Late 1960’s and 1970’s:
  - Hard disks allowed direct access to data
  - Network and hierarchical data models in widespread use
  - Ted Codd defines the relational data model
  - Would win the ACM Turing Award for this work



Information Retrieval

A Relational Model of Data for Large Shared Data Banks

E. F. Codd
IBM Research Laboratory, San Jose, California

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Existing noninferential, formatted data systems provide users with tree-structured files or slightly more general network

The relational view (or model
Section 1 appears to be superior in
graph or network model [3, 4] pres

80’s and 90’s

- 1980s:
  - Research relational prototypes evolve into commercial systems
    - SQL becomes industrial standard
  - Parallel and distributed database systems
    - Wisconsin, IBM, Teradata
  - Object-oriented database systems
- 1990s:
  - Large decision support and data-mining applications
  - Large multi-terabyte data warehouses
  - Emergence of Web commerce

2000’s and 2010’s

- 2000s
  - Big data storage systems
    - Google BigTable, Yahoo PNuts, Amazon,
  - “NoSQL” systems.
  - Big data analysis: beyond SQL
    - Map reduce
- 2010s
  - SQL reloaded
    - SQL front end to Map-Reduce systems
    - Massively parallel database systems
    - Multi-core main-memory databases

How do we study Databases?

- Key Concepts
  - View of Data
  - Database Languages
  - Database Design
  - Database Engine
  - Database Application Architecture
- Key People
  - Database Users
  - Database Administrators

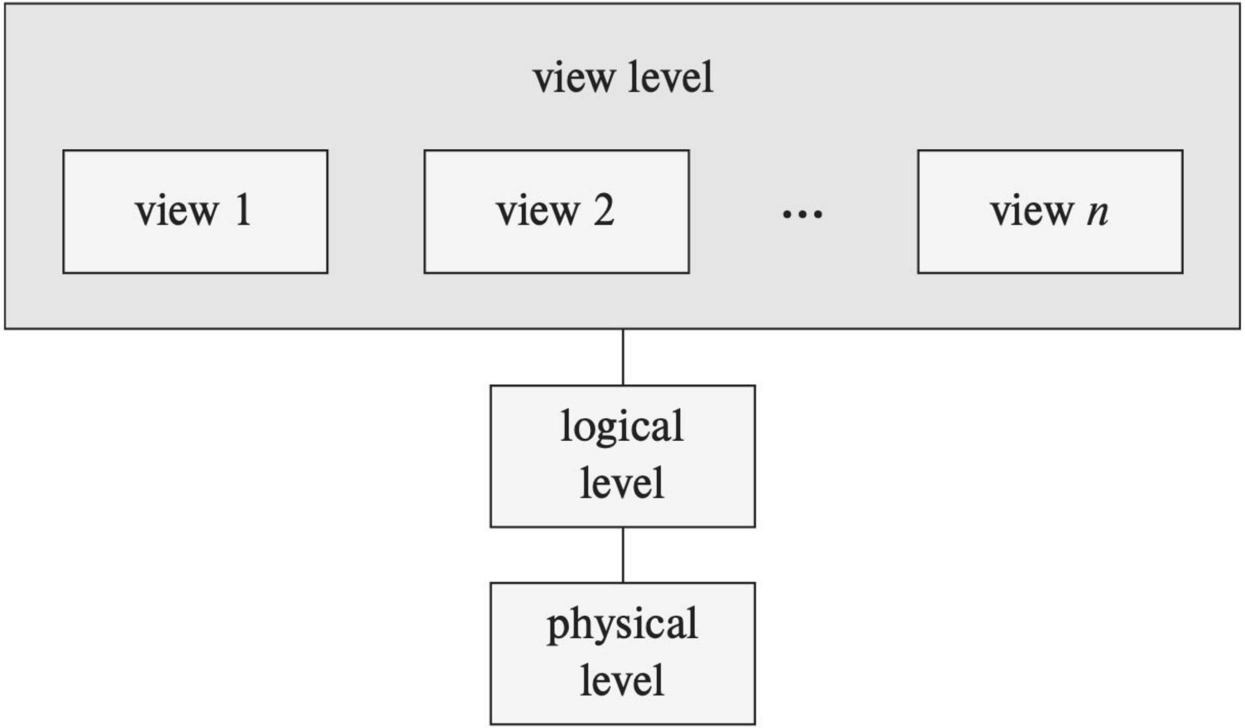
View of Data

Data Abstraction and Data Model

Data Abstraction - How?

- Physical Level
  - Lowest level, how the data are actually stored.
  - Usually in complex low-level data structures.
- Logical Level
  - What data are stored in the database and what relationships exist between the data.
  - Implementing the simple structure of the logical level may require complex physical low level structures.
  - Users of the logical level don’t need to know about this.
  - We refer to this as the physical data independence.
- View Level:
  - Highest level of abstraction - describes only a small portion of the database
  - Allows user to simplify their interaction with the database system.
  - Can have many views. WHY is this good?

Data Abstraction



*taken from: Database System Concepts 7th Ed.,Korth,Silberschatz, Sudharshan

Data Model

- Data Model: a collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints.
- Types:
  - Relational model
  - Entity-Relationship data model (mainly for database design)
  - Object-based data models (Object-oriented and Object-relational)
  - Semi-structured data model (XML)
  - Other older models:
    - Network model
    - Hierarchical model

Relational Model

- All the data is stored in various tables.
- Example of tabular data in the relational model

Columns

Rows

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

Ted Codd
Turing Award 1981

dept_name	*building*	*budget*
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table

Database Instance and Schema

- Similar to types and variables in programming languages
- **Logical Schema** – the overall logical structure of the database
  - Example: The database consists of information about a set of customers and accounts in a bank and the relationship between them
  - Analogous to type information of a variable in a program
- **Physical schema** – the overall physical structure of the database
- **Instance** – the actual content of the database at a particular point in time
  - Analogous to the value of a variable

Database Languages

Database Language

- **Data Definition Language (DDL)**
- Specification notation for defining the database schema
- Example:

```
create table instructor (
    ID          char(5),
    name        varchar(20),
    dept_name   varchar(20),
    salary      numeric(8,2))
```

- **Data Manipulation Language (DML)**
- Language for accessing and updating the data organized by the appropriate data model
- DML also known as query language
- Types:
  - Procedural DML
  - Declarative DML

SQL Query Language

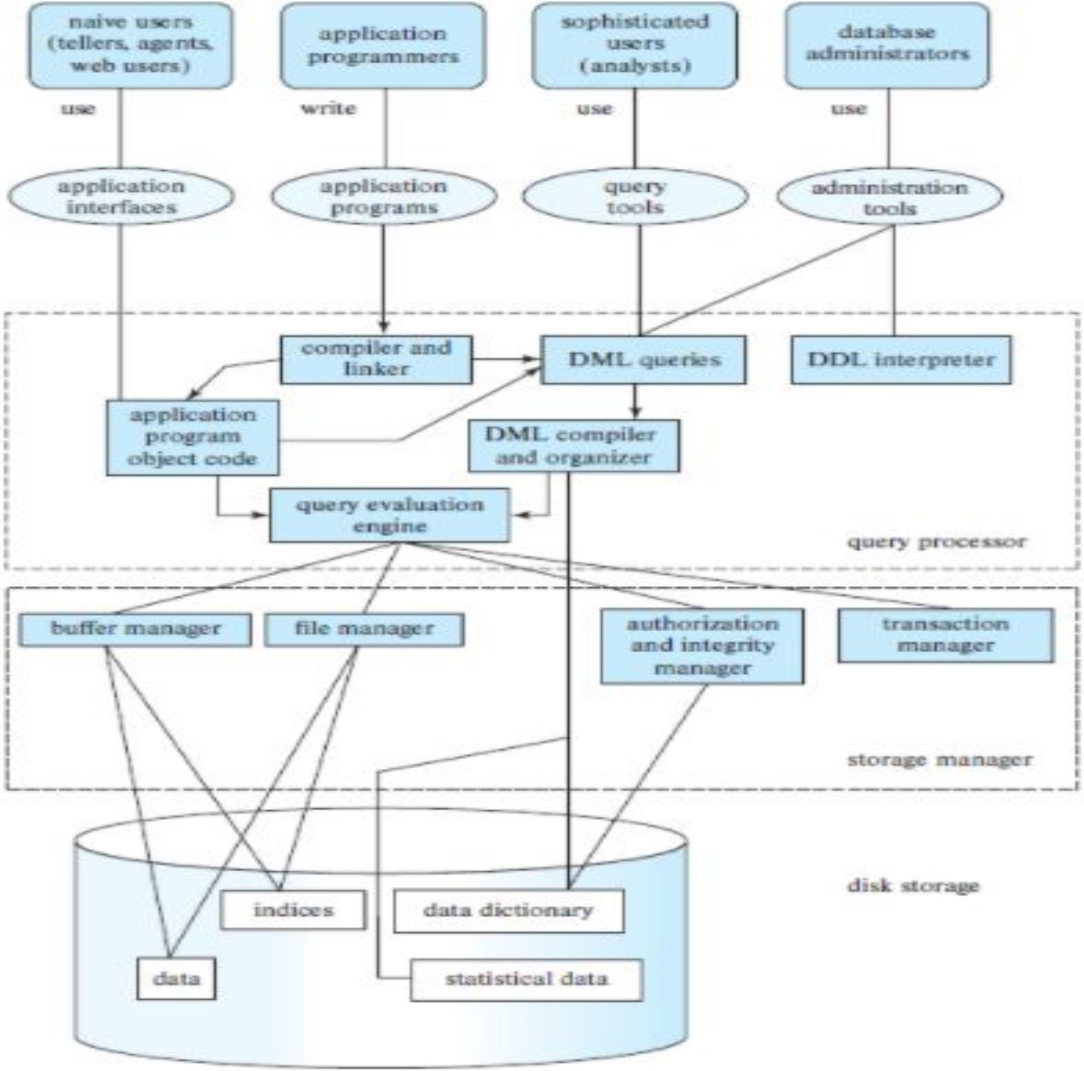
- SQL query language is nonprocedural. A query takes as input several tables (possibly only one) and always returns a single table.
- Example to find all instructors in Comp. Sci. dept

```
SELECT name
FROM instructor
WHERE dept_name = 'Comp. Sci.'
```

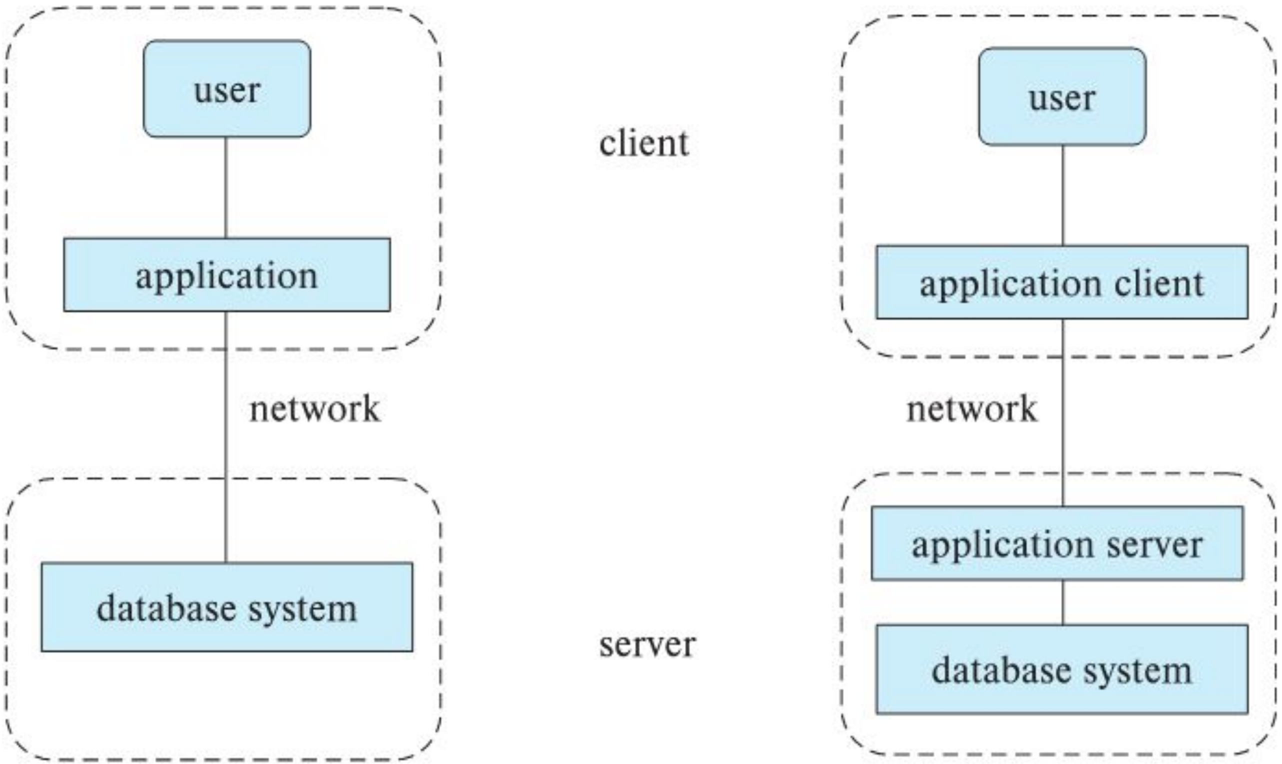
- SQL is **NOT** a Turing machine equivalent language
- SQL is usually embedded in some higher-level language
- Application programs generally access databases through one of
  - Language extensions to allow embedded SQL
  - Application program interface (e.g., ODBC/JDBC) which allow SQL queries to be sent to a database

Database Application Architecture

Database Architecture (Centralized/ Shared-Memory)



Two-Tier and Three-Tier Architecture

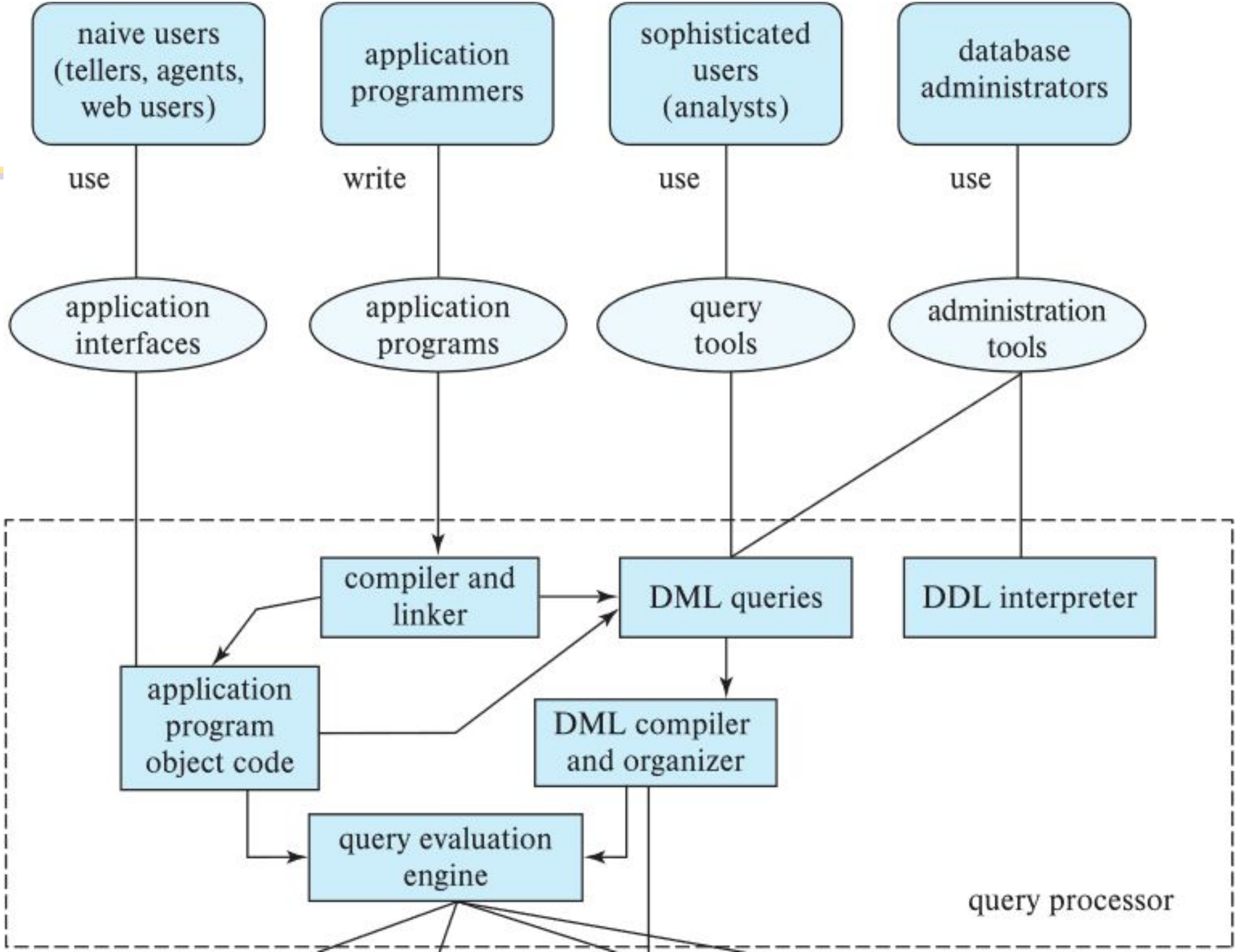


(a) Two-tier architecture

(b) Three-tier architecture

Database Users and Administrators

Database Users



Database Administrators

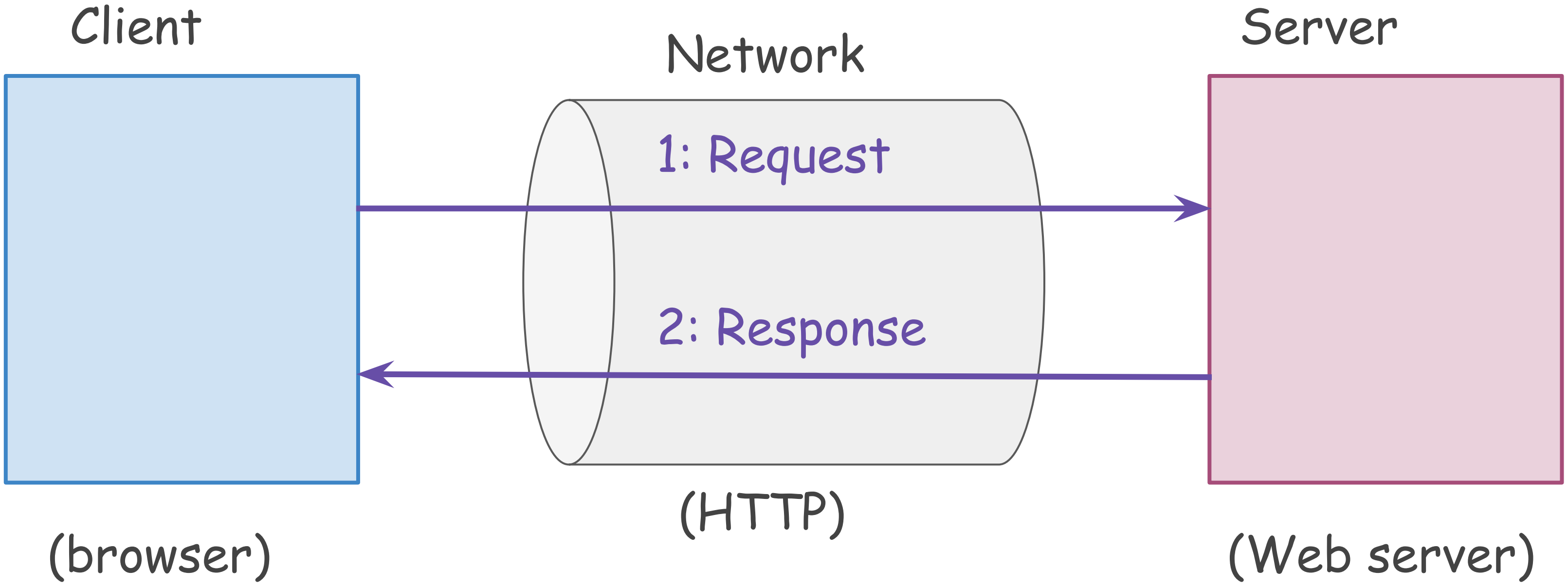
- A person who has central control over the system is called a database administrator (DBA). Functions of a DBA include:
  - Schema definition
  - Storage structure and access-method definition
  - Schema and physical-organization modification
  - Granting of authorization for data access
  - Routine maintenance
    - Periodically backing up the database
    - Ensuring that enough free disk space is available for normal operations, and upgrading disk space as required
    - Monitoring jobs running on the database

Food for thought

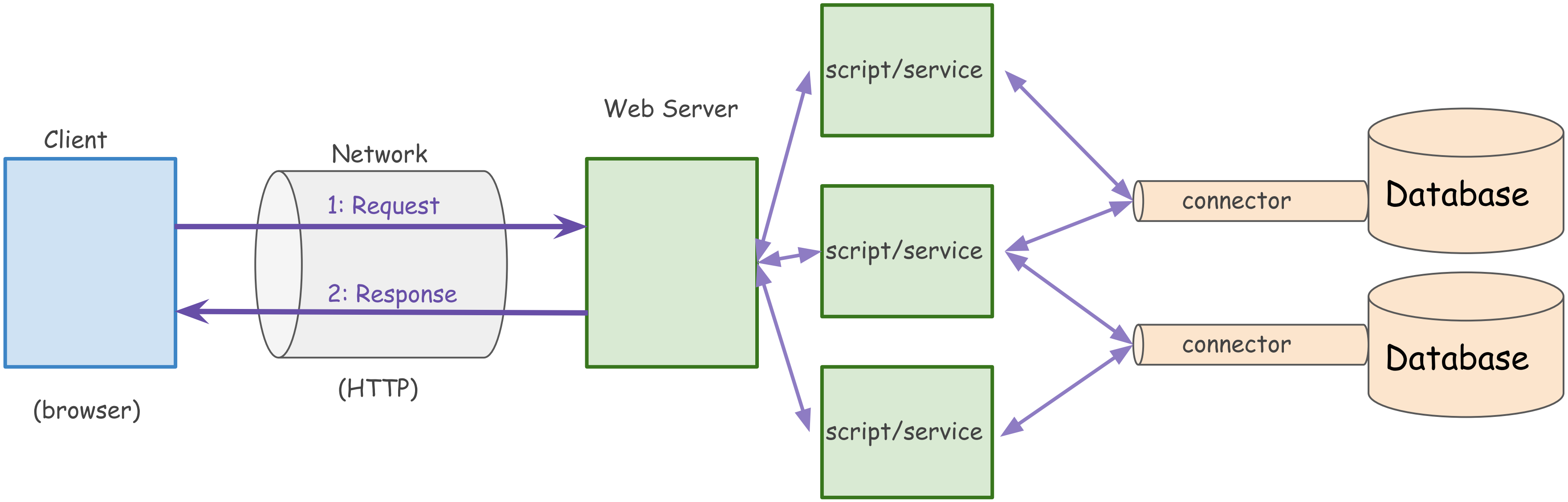
- Is www a DBMS?
  - Fairly sophisticated search available
  - But:
    - Data is unstructured and untyped
    - Search only
    - Few guarantees of accuracy, durability, consistency
  - The picture is changing
    - New standards like XML can help data modeling
    - The WWW/DB boundary is blurry!

Web Application

Web Applications (The big picture)



Web Applications (The more granular picture)



Front End

- Front end is what we see when we open a web page or app.
- The front end is built out of three languages: HTML, CSS, and JavaScript.
- HTML:
    - allows us to put *content* on our page
- CSS:
    - used to *style* our page
- JavaScript:
    - makes our page *dynamic*

Back end

- This term usually refers to what happens ‘behind the scenes’:
  - servers, databases, etc.
- Data storage (databases) and servers running to provide data for the front end.
  - JavaScript, Ruby, Java, or Python
- The database logic required in back end development often utilize a database language,
  - such as SQL.

We will start with Databases

Relational Model

- Database is a collection of *tables* each having a unique name.
- Each table also known as a *relation*.
- Rows are referred to as *tuples*.
- Columns are referred to as *attributes*.
- An *instance* of a database is the information stored at a particular moment in time.
- A database *schema* is the overall design of the database.
- Which changes frequently? The *instance* or *schema* of a database?

University Example

Instructor Relation

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Course Relation

course_id	*title*	*dept_name*	*credits*
BIO-101	Intro. to Biology	Biology	4
BIO-301	Genetics	Biology	4
BIO-399	Computational Biology	Biology	3
CS-101	Intro. to Computer Science	Comp. Sci.	4
CS-190	Game Design	Comp. Sci.	4
CS-315	Robotics	Comp. Sci.	3
CS-319	Image Processing	Comp. Sci.	3
CS-347	Database System Concepts	Comp. Sci.	3
EE-181	Intro. to Digital Systems	Elec. Eng.	3
FIN-201	Investment Banking	Finance	3
HIS-351	World History	History	3
MU-199	Music Video Production	Music	3
PHY-101	Physical Principles	Physics	4

- Give an example of an attribute, tuple.
- What is the *domain* of the column *salary*?

Terminology

- Database Schema: The logical design of the database.
- Database Instance: A snapshot of the data in the database.
- Relation Schema: A list of attributes and their corresponding domains.
- Relation Instance: A snapshot of data in the relation

The *department relation* has the schema:

- `department(dept_name, building, budget)`

The *instructor relation* has the schema:

- `instructor(ID, name, dept_name, salary)`

Why is it useful to have `dept_name` in both schemas?

dept_name	*building*	*budget*
Biology	Watson	90000
Comp. Sci.	Taylor	100000
Elec. Eng.	Taylor	85000
Finance	Painter	120000
History	Painter	50000
Music	Packard	80000
Physics	Watson	70000

University Example: Relations

- So far we have the following schemas:

```
department(dept_name, building, budget),
instructor(ID, name, dept_name, salary),
course(course_id, title, dept_name, credits),
```

- What other schemas might we want?

```
teaches(ID, course_id,sec_id, semester, year),
section(course_id, sec_id, semester, year, building, room_number, time_slot_id),
student(ID, name, dept_name, tot_cred),
takes(ID, course_id, sec_id, semester, year, grade),
time_slot(time_slot_id, day, start_time, end_time),
...,
```

- How do we uniquely refer to a tuple or row in a schema?

Keys

- Superkey
  - a set of one or more attributes that taken together uniquely identify a tuple in the relation.
- What are possible superkeys for the `instructor` relation?
  - `instructor(ID, name, dept_name, salary)`

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Keys

- Superkey
  - a set of one or more attributes that taken together uniquely identify a tuple in the relation.
- What about the `teaches` relation?

ID	*course_id*	*sec_id*	*semester*	*year*
10101	CS-101	1	Fall	2017
10101	CS-315	1	Spring	2018
10101	CS-347	1	Fall	2017
12121	FIN-201	1	Spring	2018
15151	MU-199	1	Spring	2018
22222	PHY-101	1	Fall	2017
32343	HIS-351	1	Spring	2018
45565	CS-101	1	Spring	2018
45565	CS-319	1	Spring	2018
76766	BIO-101	1	Summer	2017
76766	BIO-301	1	Summer	2018
83821	CS-190	1	Spring	2017
83821	CS-190	2	Spring	2017
83821	CS-319	2	Spring	2018
98345	EE-181	1	Spring	2017

We are interested in superkey sets that are minimal.

Candidate Keys (Minimal Superkeys)

`instructor(ID, name, dept_name, salary)`

- Superkeys for relation *`instructor`*:
  - `{ID}, {name, dept_name}, {ID, name}`
- Candidate Key
  - A minimal superkey.
- Q: Which of the above superkeys are candidate keys?
  - A: `{ID}, {name, dept_name}`
- Primary Key
  - A candidate key chosen by the database designer to distinguish between tuples.

Things to do this week

- Go through the syllabus uploaded on Quercus. Familiarize yourself with the rules we follow in class.
- Read the lecture notes for first week
- Check out Piazza.
- If you are not enrolled in a tutorial for this course, please do so via Acorn

Next week

- Tutorials begin

- Relational Model Continued
- Relational diagrams
- Relational operations
- Relational algebra

- Intro to SQL and SQLite (tentative)

Image Credits

A big thank you to the sources of all images and content used in this presentation:

1. **Instructor Relation and Course Relation Table**:
    - Adapted from educational examples in database textbooks.
2. **Relational Model Examples**:
    - Inspired by material commonly used in introductory database courses.
3. **Web Application Architecture Diagrams**:
    - Created based on standard client-server models.
4. **Historical Database Materials**:
    - Sourced from public domain resources and educational archives.
5. **Database Abstraction Diagrams**:
    - Adapted from "Database System Concepts" by Korth, Silberschatz, and Sudarshan (7th Edition).
6. **Other Images**:
    - All remaining graphics and visuals were designed specifically for this course or sourced from royalty-free or academic sources.