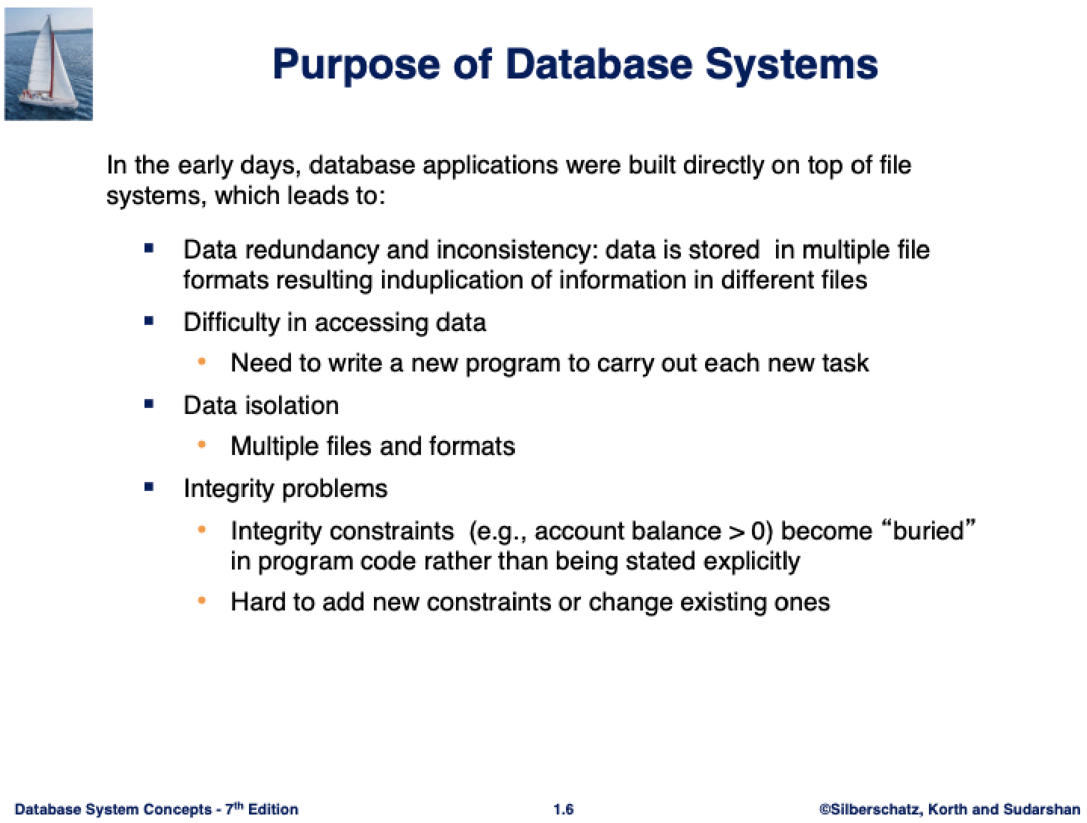
Answers: F24-W4111: Homework 1A

Overview

- Check CourseWorks and Ed for submission instructions, deadlines, clarifications, etc.
- This homework is a set of written questions testing knowledge and review of concepts covered in:
  - The lectures, both the presentations and any information communicated during the lecture.
  - The slides that accompany the recommended textbook.
    - URL: https://db-book.com/slides-dir/index.html
    - You only need to reference the slides for chapters 1, 2, 3, 6, and 7.

Concepts

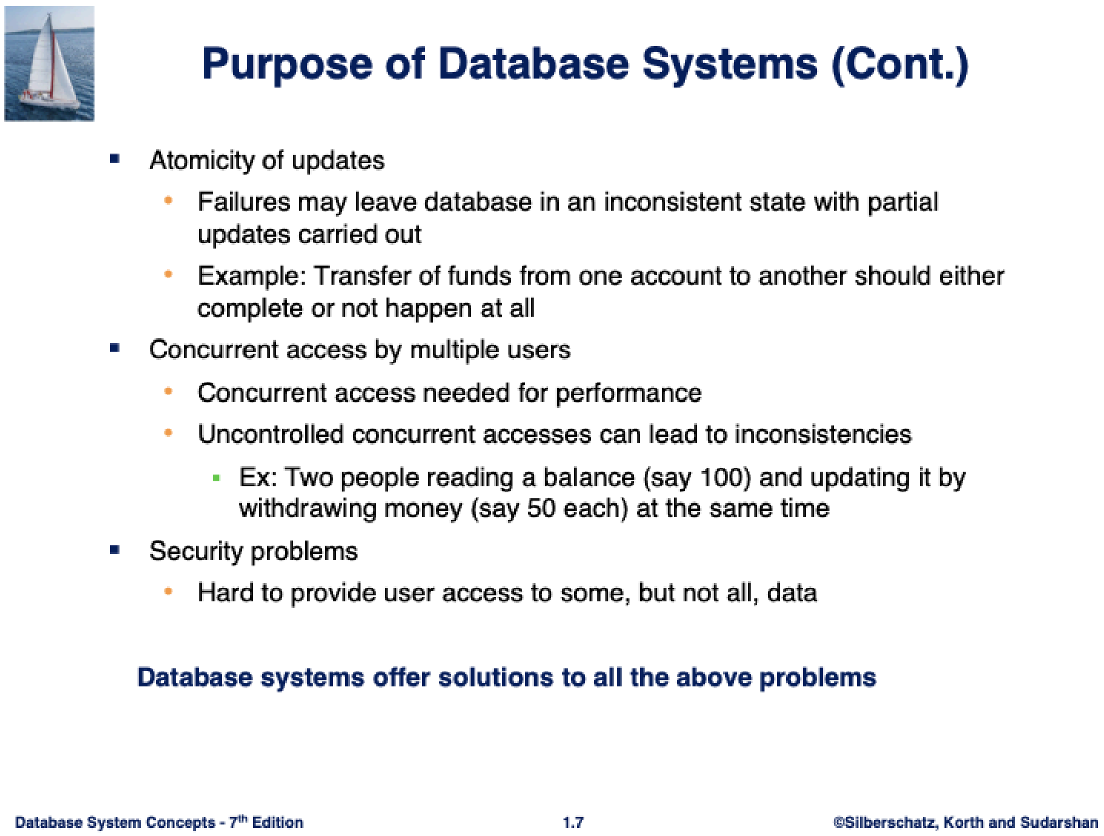
1. List purposes for/motivations for database systems compared to managing data by writing applications that process files.



Purpose of Database Systems

In the early days, database applications were built directly on top of file systems, which leads to:

- Data redundancy and inconsistency: data is stored in multiple file formats resulting induplication of information in different files
- Difficulty in accessing data
  - Need to write a new program to carry out each new task
- Data isolation
  - Multiple files and formats
- Integrity problems
  - Integrity constraints (e.g., account balance > 0) become "buried" in program code rather than being stated explicitly
  - Hard to add new constraints or change existing ones



Purpose of Database Systems (Cont.)

- Atomicity of updates
  - Failures may leave database in an inconsistent state with partial updates carried out
  - Example: Transfer of funds from one account to another should either complete or not happen at all
- Concurrent access by multiple users
  - Concurrent access needed for performance
  - Uncontrolled concurrent accesses can lead to inconsistencies
    - Ex: Two people reading a balance (say 100) and updating it by withdrawing money (say 50 each) at the same time
- Security problems
  - Hard to provide user access to some, but not all, data

Database systems offer solutions to all the above problems

2. Database systems provide users with an abstraction view of the data.

a. Briefly explain the concept.
b. Why do writing applications that access files do not provide an abstraction?

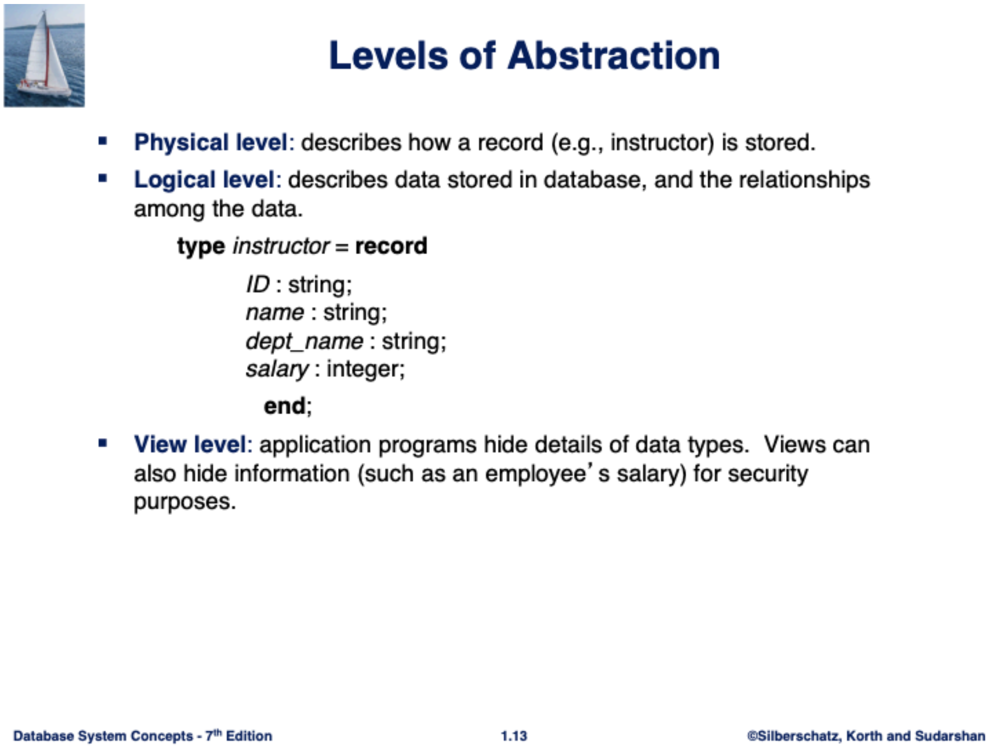
<answer>

- Ultimately, data resides in a physical format on storage devices/in memory. Additionally, the physical format is a block of bytes/stream of bytes that must map into programing language data structures. DBMS provide a common, consistent abstraction that isolates users and programmers for the physical details and layouts.
- At the lowest level, reading/writing a file involves reading/writing blocks of bytes. The application needs to understand how to interpret the bytes and map to data structures.

</answer>

3. What are the 3 levels of data abstraction that a DBMS provides?

<answer>



Levels of Abstraction

- **Physical level**: describes how a record (e.g., instructor) is stored.
- **Logical level**: describes data stored in database, and the relationships among the data.

```
    type instructor = record
            ID : string;
            name : string;
            dept_name : string;
            salary : integer;
              end;
```

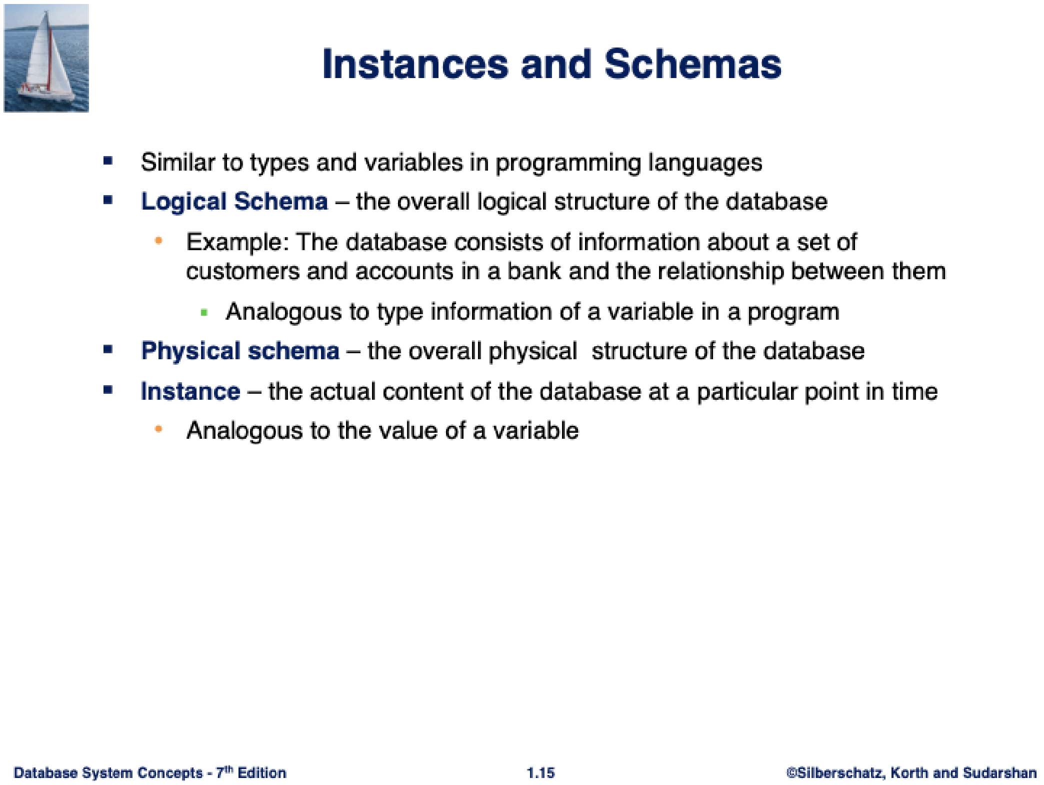
- **View level**: application programs hide details of data types. Views can also hide information (such as an employee's salary) for security purposes.

Database System Concepts - 7th Edition 1.13 ©Silberschatz, Korth and Sudarshan

</answer>

4. Explain the difference between database schema and database instance. What two concepts in an object-oriented language correspond to schema and instance?

<answer>



Instances and Schemas

- 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 System Concepts - 7th Edition 1.15 ©Silberschatz, Korth and Sudarshan

In an OO language,

- Classes define the structure of the data.
- Objects are the instance data.

</answer>

5. Briefly describe the concepts of data definition language and data manipulation language. What are the two types of data manipulation language?

<answer>

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))
  ```
- DDL compiler generates a set of table templates stored in a ***data dictionary***
- Data dictionary contains metadata (i.e., data about data)
  - Database schema
  - Integrity constraints
    - Primary key (ID uniquely identifies instructors)
  - Authorization
    - Who can access what

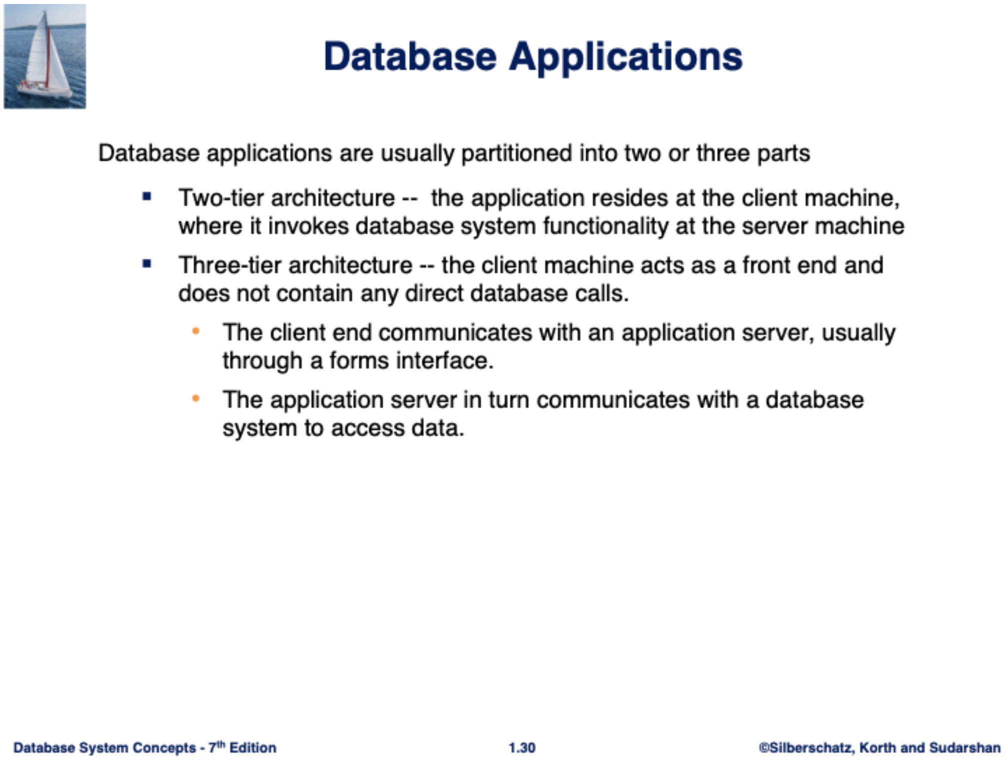
Data Manipulation Language (DML)

- Language for accessing and updating the data organized by the appropriate data model
  - DML also known as query language
- Two classes of languages
  - **Pure** – used for proving properties about computational power and for optimization
    - Relational Algebra
    - Tuple relational calculus
    - Domain relational calculus
  - **Commercial** – used in commercial systems
    - SQL is the most widely used commercial language

</answer>

6. Briefly explain two-tier and three-tier database application architectures. Is a full-stack web application a two-tier architecture or a three-tier architecture?

<answers>

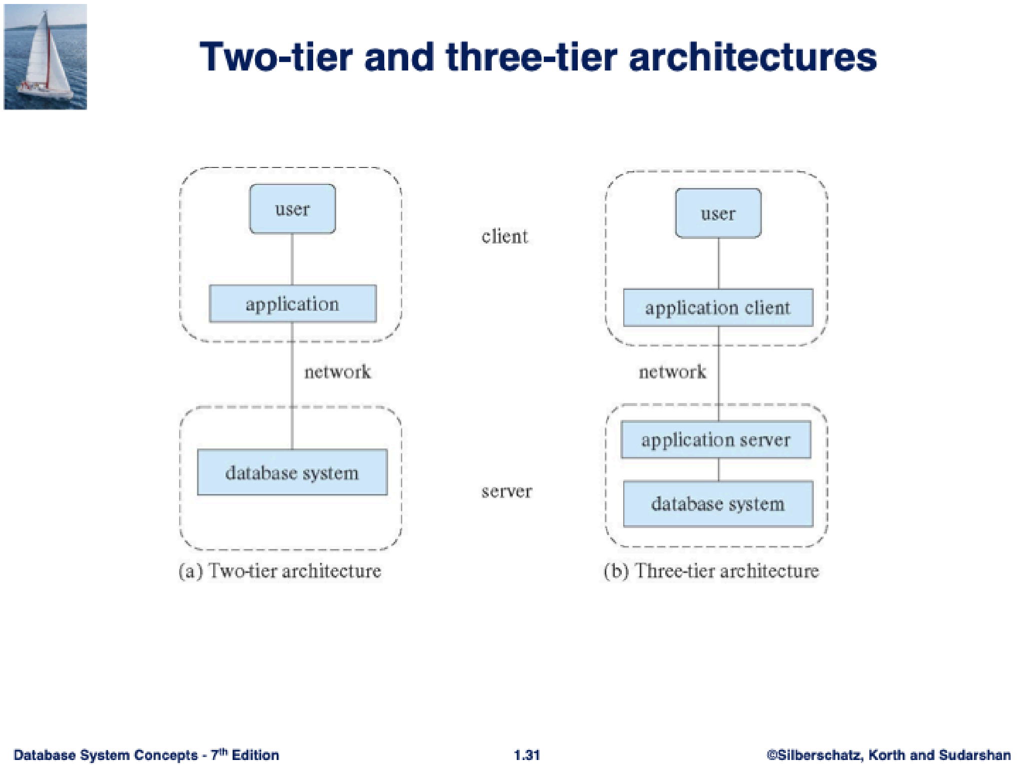


Database Applications

Database applications are usually partitioned into two or three parts

- Two-tier architecture -- the application resides at the client machine, where it invokes database system functionality at the server machine
- Three-tier architecture -- the client machine acts as a front end and does not contain any direct database calls.
  - The client end communicates with an application server, usually through a forms interface.
  - The application server in turn communicates with a database system to access data.

Database System Concepts - 7th Edition   1.30   ©Silberschatz, Korth and Sudarshan



Two-tier and three-tier architectures

(a) Two-tier architecture

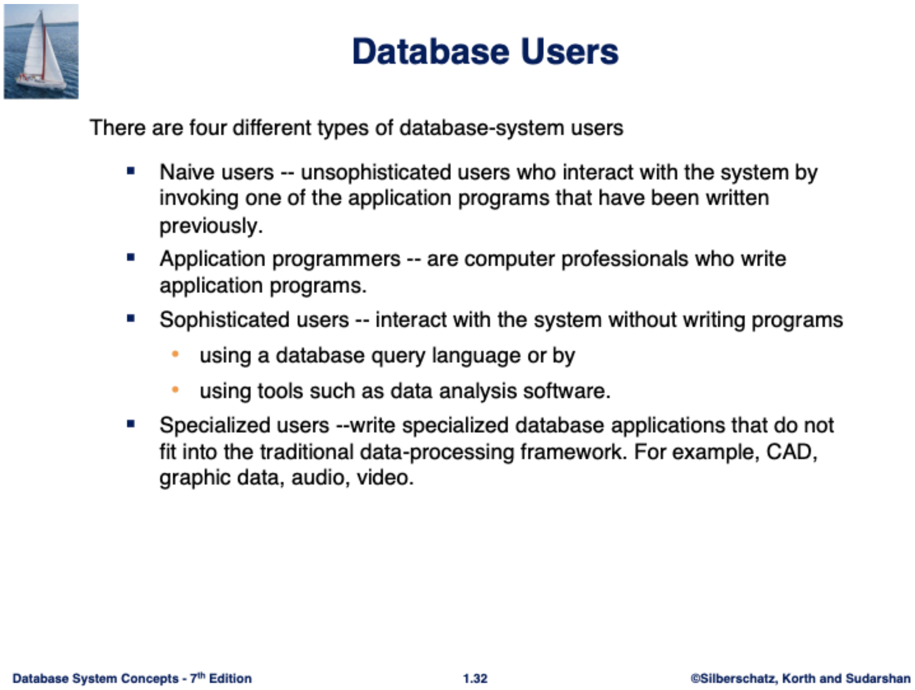
(b) Three-tier architecture

Database System Concepts - 7th Edition   1.31   ©Silberschatz, Korth and Sudarshan

</answers>

A full-stack application has three layers/elements: 1) UI, 2) Application, 3) Database. The “stack” is the runtime supporting the model, for example Browser – FastAPI/NodeJS – MySQL. Full-stack is inherently 3-tier.

7. What are the four types of database users based on skill level and for what they use the database. Which type of user defines schema and defines what information users can access?

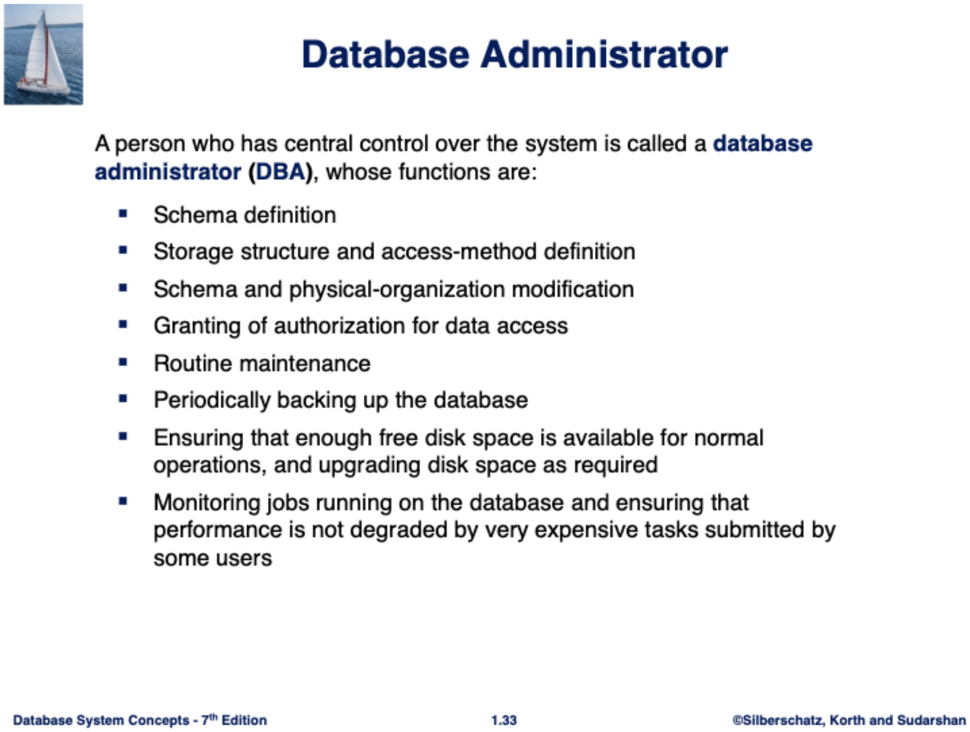


Database Users

There are four different types of database-system users

- Naive users -- unsophisticated users who interact with the system by invoking one of the application programs that have been written previously.
- Application programmers -- are computer professionals who write application programs.
- Sophisticated users -- interact with the system without writing programs
  - using a database query language or by
  - using tools such as data analysis software.
- Specialized users --write specialized database applications that do not fit into the traditional data-processing framework. For example, CAD, graphic data, audio, video.

Database System Concepts - 7th Edition 1.32 ©Silberschatz, Korth and Sudarshan



Database Administrator

A person who has central control over the system is called a **database administrator (DBA)**, whose functions are:

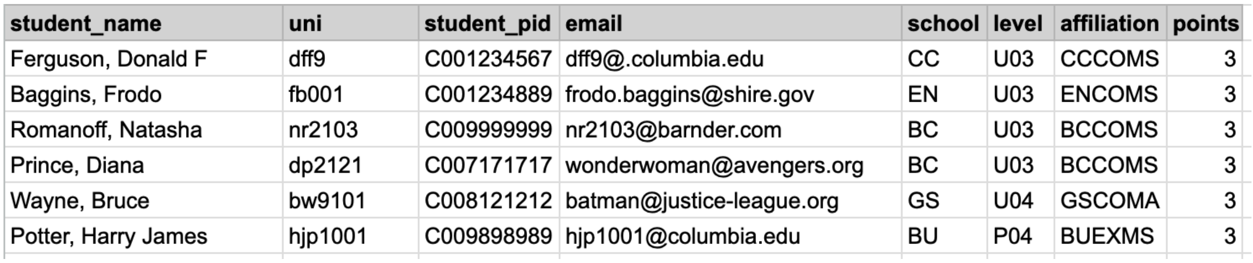
- Schema definition
- Storage structure and access-method definition
- Schema and physical-organization modification
- Granting of authorization for data access
- Routine maintenance
- Periodically backing up the database
- Ensuring that enough free disk space is available for normal operations, and upgrading disk space as required
- Monitoring jobs running on the database and ensuring that performance is not degraded by very expensive tasks submitted by some users

Database System Concepts - 7th Edition 1.33 ©Silberschatz, Korth and Sudarshan

Relational Model

Understanding Data

The following is data in the SSOL format for our course COMS W4111 but with made up values for Student Name, Student UNI, Student PID, Email, School, Level, Affiliation, Points.



student_name	uni	student_pid	email	school	level	affiliation	points
Ferguson, Donald F	dff9	C001234567	dff9@.columbia.edu	CC	U03	CCCOMS	3
Baggins, Frodo	fb001	C001234889	frodo.baggins@shire.gov	EN	U03	ENCOMS	3
Romanoff, Natasha	nr2103	C009999999	nr2103@barnder.com	BC	U03	BCCOMS	3
Prince, Diana	dp2121	C007171717	wonderwoman@avengers.org	BC	U03	BCCOMS	3
Wayne, Bruce	bw9101	C008121212	batman@justice-league.org	GS	U04	GSCOMA	3
Potter, Harry James	hjp1001	C009898989	hjp1001@columbia.edu	BU	P04	BUEXMS	3

Answer the following questions based on your general knowledge of Columbia University and examples given in class.

1. Which attributes are not from an atomic domain?
   - ○ student_name is clearly non-atomic and is 3 domains: last_name, first_name, middle_name/middle initial.
   - ○ affiliation is also clearly non atomic and has two domains: school_code, department.
   - ○ email could be two domains: email_id and internet domain. But, people do not think this way. This question/comment arose in class and I explained it.
   - ○ level is possibly non-atomic, i.e. U=undergrad and 03 is year. P is probably “Professional.” I would not split apart, however.
2. Which attributes are likely to be candidate keys?
   - ○ uni, student_pid and possibly email.
3. Which attributes are likely to be foreign keys into another relation?
   - ○ The most obvious one is school. I explicitly showed this example in lectures.
4. Which attribute is clearly a surrogate key?
   - ○ student_pid

An Algebra

“The result of a relational-algebra operation is a relation and therefore relational-algebra operations can be composed together into a relational-algebra expression.”

Assume that the name of the relation above is SSOL. Give two relational algebra expressions that are examples of composition that when applied to the relation above produce the following relation.

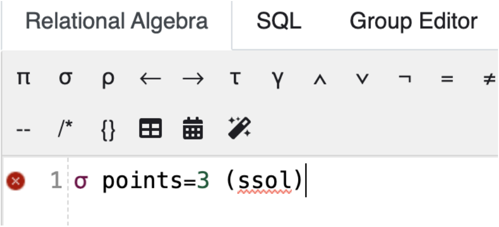
student_name	uni	school
Romanoff, Natasha	nr2103	BC
Prince, Diana	dp2121	BC

```
<answer>
        π student_name, uni, school(
                σ school='BC' (SSOL)
        )

        σ school='BC'
        (
                π student_name, uni, school(SSOL)
        )
</answers>
```

You can use the relax calculator to type your expressions, and then copy/paste the expressions into your answers. The syntax check will indicate errors but you are not executing the expressions.

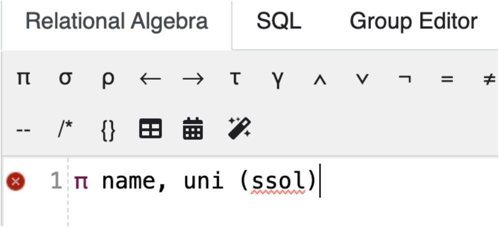
For example,



```
σ points=3 (ssol)
```

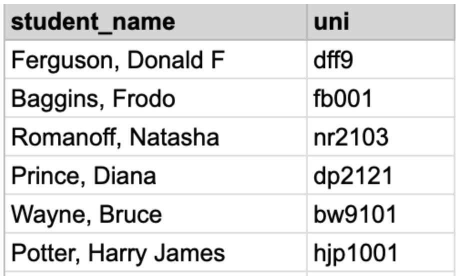
shows how to type an expression that is a select which matches all rows in the original relation.

And,



```
π name, uni (ssol)
```

shows how to type an expression that produces



student_name	uni
Ferguson, Donald F	dff9
Baggins, Frodo	fb001
Romanoff, Natasha	nr2103
Prince, Diana	dp2121
Wayne, Bruce	bw9101
Potter, Harry James	hjp1001

Schema

Translate this relation schema definition into an SQL create table statement. This question uses the notation from class slides.

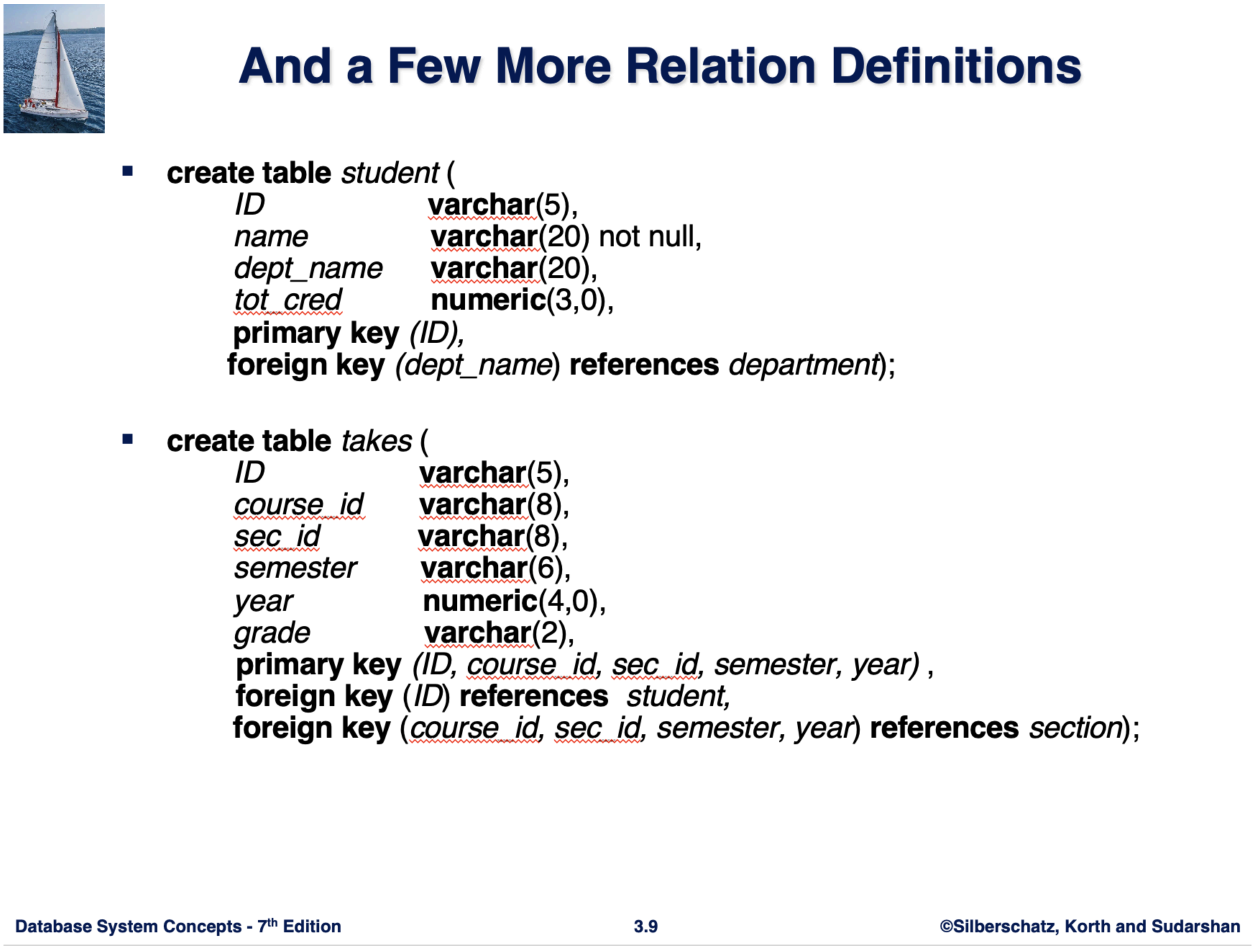
instructor = (UNI, last_name, first_name)

```
/*
        The types are not important for this answer. Not concerned about NULL/NOT NULL.
        Neither is relevant at the algebra/schema level.
*/
create table if not exists F24_examples.hw1a_instructor
(
   uni        varchar(32) not null
      primary key,
   last_name  varchar(64) not null,
   first_name varchar(64) not null
);
```

Entity Relationship Modeling

Reverse Engineering

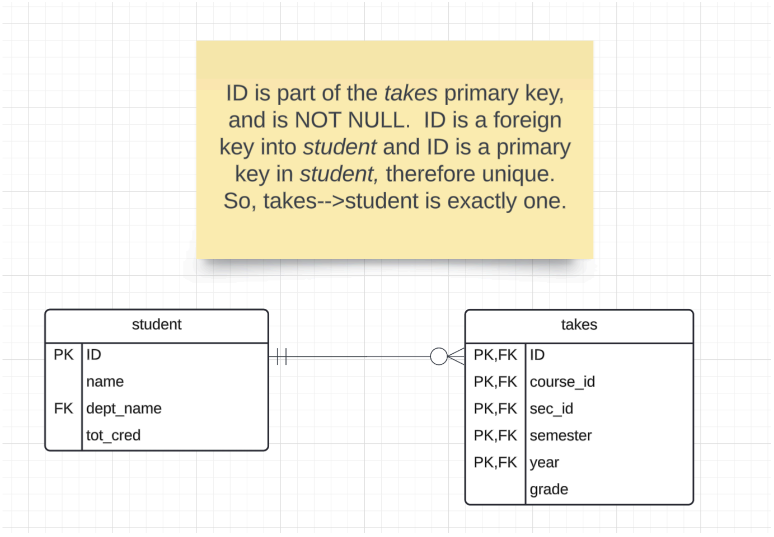
The following slide is from the slides associated with the recommended textbook.



And a Few More Relation Definitions

-
  ```
  create table student (
          ID              varchar(5),
          name            varchar(20) not null,
          dept_name       varchar(20),
          tot_cred        numeric(3,0),
          primary key (ID),
         foreign key (dept_name) references department);
  ```

-
  ```
  create table takes (
          ID              varchar(5),
          course_id       varchar(8),
          sec_id          varchar(8),
          semester        varchar(6),
          year            numeric(4,0),
          grade           varchar(2),
          primary key (ID, course_id, sec_id, semester, year) ,
          foreign key (ID) references  student,
         foreign key (course_id, sec_id, semester, year) references section);
  ```



ID is part of the *takes* primary key, and is NOT NULL. ID is a foreign key into *student* and ID is a primary key in *student,* therefore unique. So, takes-->student is exactly one.

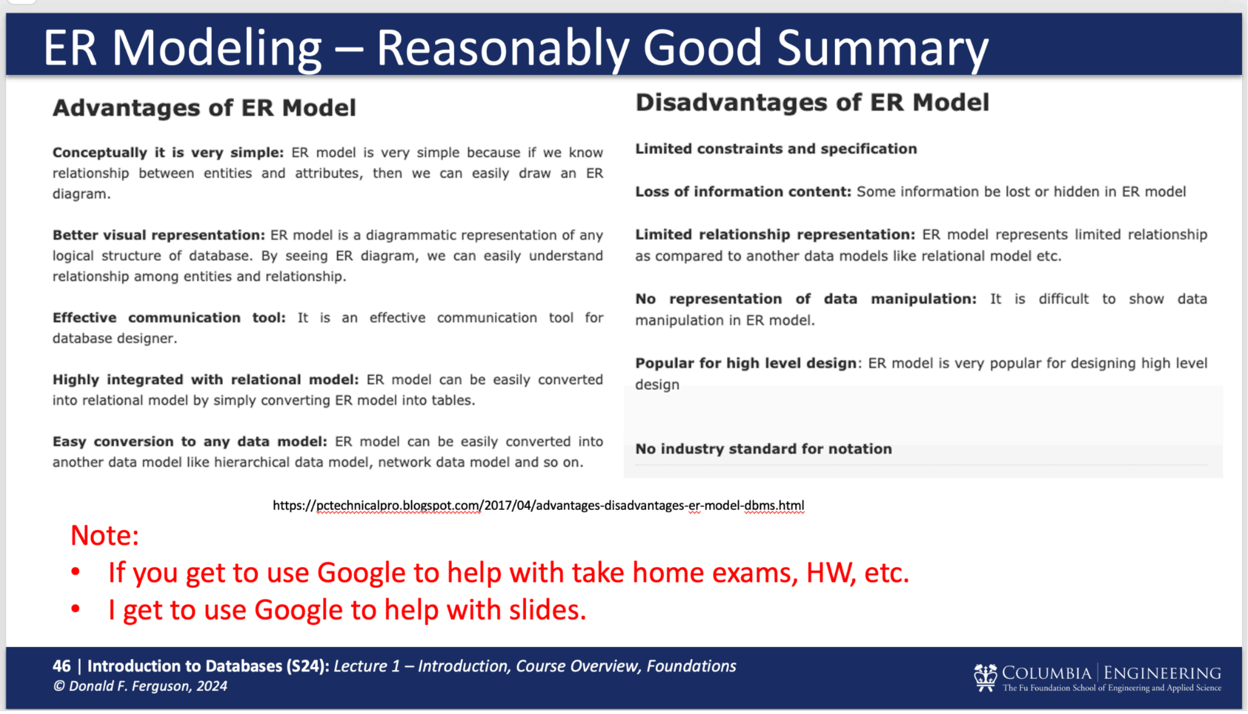
student	
PK	ID
	name
FK	dept_name
	tot_cred

takes	
PK,FK	ID
PK,FK	course_id
PK,FK	sec_id
PK,FK	semester
PK,FK	year
	grade

Use LucidChart to draw an ER diagram in Crow's Foot Notation that is a reverse-engineered, logical model of the schema. You only need to do the entities *student* and *takes.* Do not worry about other referenced entities.

Pros and Cons

List some advantages and disadvantages of ER Modeling.



ER Modeling – Reasonably Good Summary

Advantages of ER Model

Conceptually it is very simple: ER model is very simple because if we know relationship between entities and attributes, then we can easily draw an ER diagram.

Better visual representation: ER model is a diagrammatic representation of any logical structure of database. By seeing ER diagram, we can easily understand relationship among entities and relationship.

Effective communication tool: It is an effective communication tool for database designer.

Highly integrated with relational model: ER model can be easily converted into relational model by simply converting ER model into tables.

Easy conversion to any data model: ER model can be easily converted into another data model like hierarchical data model, network data model and so on.

Disadvantages of ER Model

Limited constraints and specification

Loss of information content: Some information be lost or hidden in ER model

Limited relationship representation: ER model represents limited relationship as compared to another data models like relational model etc.

No representation of data manipulation: It is difficult to show data manipulation in ER model.

Popular for high level design: ER model is very popular for designing high level design

No industry standard for notation

https://pctechnicalpro.blogspot.com/2017/04/advantages-disadvantages-er-model-dbms.html

Note:

- If you get to use Google to help with take home exams, HW, etc.
- I get to use Google to help with slides.

46 | Introduction to Databases (S24): *Lecture 1 – Introduction, Course Overview, Foundations*
© Donald F. Ferguson, 2024

COLUMBIA | ENGINEERING
The Fu Foundation School of Engineering and Applied Science