

Concepts

1. List purposes for/motivations for database systems compared to managing data by writing applications that process files.
 - Database systems allow for domain and integrity constraints to be placed on data which ensure its consistency.
 - Database systems use standardized query languages like SQL that are optimized for efficient and fast storage, manipulation, and retrieval of data, and can represent complex relationships between data.
 - Database systems are scalable and secure and can handle concurrency control between multiple users accessing data at the same time.
 - Database systems provide a layer of abstraction between data and application code, allowing changes to be made to either the database schema or the application independently of the other.
2. Database systems provide users with an abstraction view of the data.
 - a. Briefly explain the concept.

Database systems allow users to interact with the data without needing to fully understand how the data is stored. Users can use high-level query languages like SQL to view and update data in specific ways without needing to implement a complex program or understand the physical storage of the data. Furthermore, changes can be made to the database schema without affecting any applications that access the data.

- b. Why do writing applications that access files do not provide an abstraction?

Applications that access files do not provide an abstraction because they require developers to write custom programs to manually parse data and understand the physical structures of the files they are accessing. This results in tight coupling between the files containing the data and the application code, meaning that changes made to either would result in significant changes

needing to be made to the other. Finally, accessing files directly means that users lose out on the advantages of database systems, including a standard query language, representation of complex relationships between data, data integrity, scalability, security, and concurrency.

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

The 3 levels of data abstraction that a DBMS provides are the conceptual, logical, and physical levels.

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

A database schema represents the logical structure of the data being stored, including tables, fields, constraints, and relationships. A database instance is a specific state of the database schema which contains the actual data stored in the database. In terms of object-oriented language concepts, a schema is analogous to a class, while an instance is analogous to an object of a class.

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

Data definition language (DDL) includes commands for specifying integrity constraints, defining views, controlling transactions, and establishing data relationships. Data manipulation language (DML) provides the ability to perform create, read, update, and delete operations on tuples in the database. The two types of DML are procedural DML and declarative DML. Procedural DML requires a user to specify what data are needed and how to get it, and declarative DML requires a user to specify what data are needed without specifying how to get it.

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?

Two-tier and three-tier database application architectures describe how many parts a database application is partitioned into. In two-tier architecture, the application resides at the client machine and invokes database system functionality at the server machine. In three-tier

architecture, the client machine acts as a front-end and does not contain any direct database calls. Rather, the client communicates with an application server, which in turn communicates with a database system to access data. A full-stack web application is a three-tier architecture.

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?

The four types of database users are:

- Naive users: unsophisticated users who invoke a previously written application program.
- Application programmers: professionals who write application programs.
- Sophisticated users: users who interact with the system without writing programs either by using a database query language or by using tools such as data analysis software.
- Specialized users: users who write specialized database applications that don't fit into the traditional data-processing framework.

Database administrators define the schema and define what information users can access.

Relational Model

Understanding Data

1. Which attributes are not from an atomic domain?

- student_name
- student_pid
- email
- level
- affiliation

2. Which attributes are likely to be candidate keys?

- uni
- student_pid
- email

3. Which attributes are likely to be foreign keys into another relation?

- school
- affiliation

4. Which attribute is clearly a surrogate key?

- student_pid

An Algebra

- π student_name, uni, school (σ school='BC' (ssol))
- π student_name, uni, school (σ affiliation='BCCOMS' (ssol))

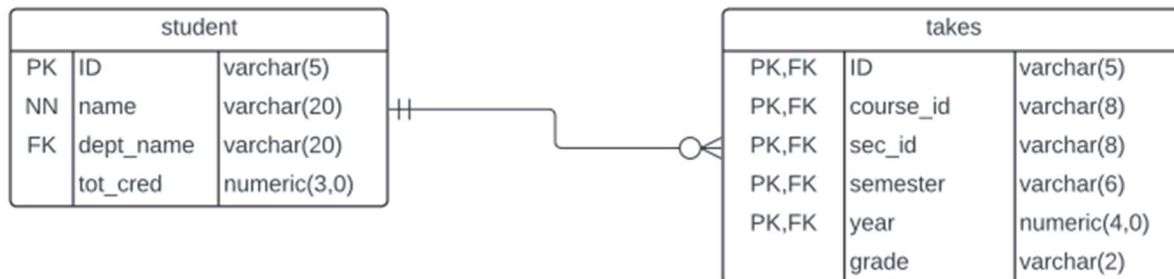
Schema

CREATE TABLE instructor

```
(  
  uni          VARCHAR(10),  
  last_name    VARCHAR(64),  
  first_name   VARCHAR(64),  
)
```

Entity Relationship Modeling

Reverse Engineering



Pros and Cons

Advantages of ER Modeling:

- ER models are conceptually simple and can be easily drawn if the relationships between entities and attributes are known.
- ER models are a good visual diagrammatic representation of the logical structure of a database and allow for easy understanding of relationships among entities.
- ER models are an effective communication tool for database designers with other technical and non-technical stakeholders.
- ER models can easily be converted into relational data models using tables, as well as many other data models including hierarchical and network data models.

Disadvantages of ER Modeling:

- ER models have limited constraints and specifications, which can result in a loss of some information content.
- ER models have a limited ability to represent complex relationships in comparison to other data models such as relational data models.
- ER models have a limited ability to represent data manipulation.
- ER models lack an industry standard for notation and can require specialized software tools to create.