Unit-2 : Database System Concepts & Architecture

- Ankit Pangeni

DATE

Data models, Schemas, and Instances

* Data Abstraction:

It refers to the hiding mechanism of the details of the data organization and storage and highlighting only the essential features for an improved understanding of data. To provide data abstraction is one of the fundamental characteristic of database approach. There are three levels of abstraction.

i) Physical Level / Lowest Level
   ⇒ describes how data is actually stored in database.

ii) Logical level / second lowest level
   ⇒ describes type of data stored & relation between them.

iii) View level / highest level
   ⇒ describes interaction between the users and System.

¥ Data Models

It is the collection of concepts that can be used to describe the structure of a database. Structure of a data base includes the data types, relationships, constraints & operational behaviour on the data in an organization.

PAGE

DATE

It shows how data is stored, connected, accessed and updated in the DBMS.

Categories/Types:

1. High Level data (conceptual data model) — Entity relationship Data model
2. Low Level (Physical data model)
3. Logical (Representation or Implementation Data model) — relational, Hierarchical, Network

Extra: i) object oriented, ii) object based

1. Conceptual data model (High Level) (object-based)

This data model defines what the system contains. It is highly abstract as it does not include too much detail. It is easily understood by technical as well as non technical person. Only entities are visible and they have abstract relationship. No software tool is required to define a conceptual model as it can be drawn from a piece of paper. Ex: ER data model.

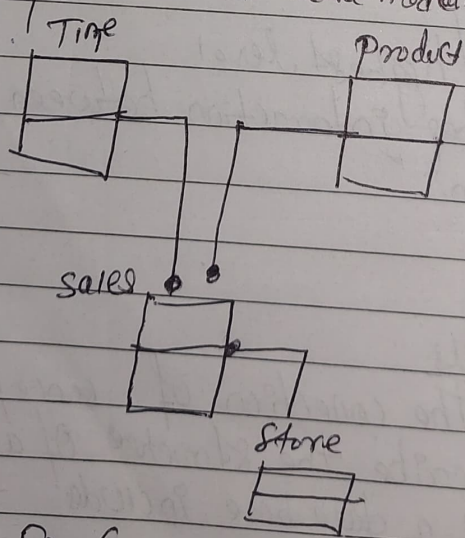
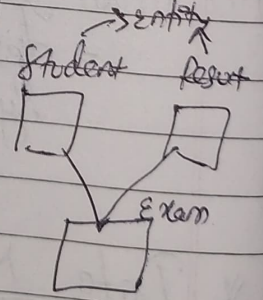


Fig: Conceptual model e.g. (object based model)



Exam in relationship betn std & result

Fig: Conceptual model eg:-

classmate

PAGE

(self-describing data models)
~~(record based)~~

DATE

2. Logical (Representation / Implementation) Data Model

Describes the data in as much detail as possible without regard to how they will be physically implemented in database. There is presence of attribute for each Entity. Contains both key and non-key attributes. Key attributes can also be used as a foreign key. This type of data model has user friendly ~~da~~ attribute names. It is more detailed than conceptual model.
Ex: Relational data model, Hierarchical data model, Network data model.

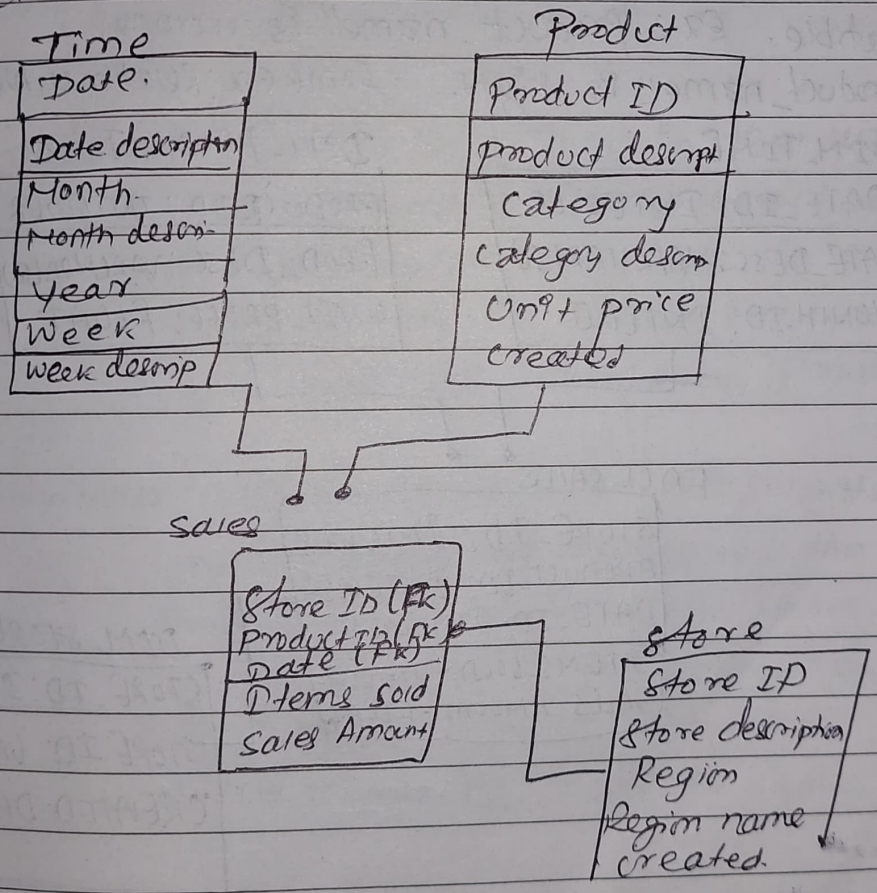


fig: Logical data model (self describing)

DATE

3. Physical data model (Low Level) (Record based model)

This data model describes the details of how data are stored in database. Concepts provided by low level data model are generally for Computer specialists not for typical users. It represents how the data model will be built in data base. Here, Entities are referred to as Table, attributes are referred to as columns. It is difficult for normal users to understand The table and column names should be database compatible. Ex: "product name" is wrong, "product_name" is right. · Some Ex: Relational, hierarchical.

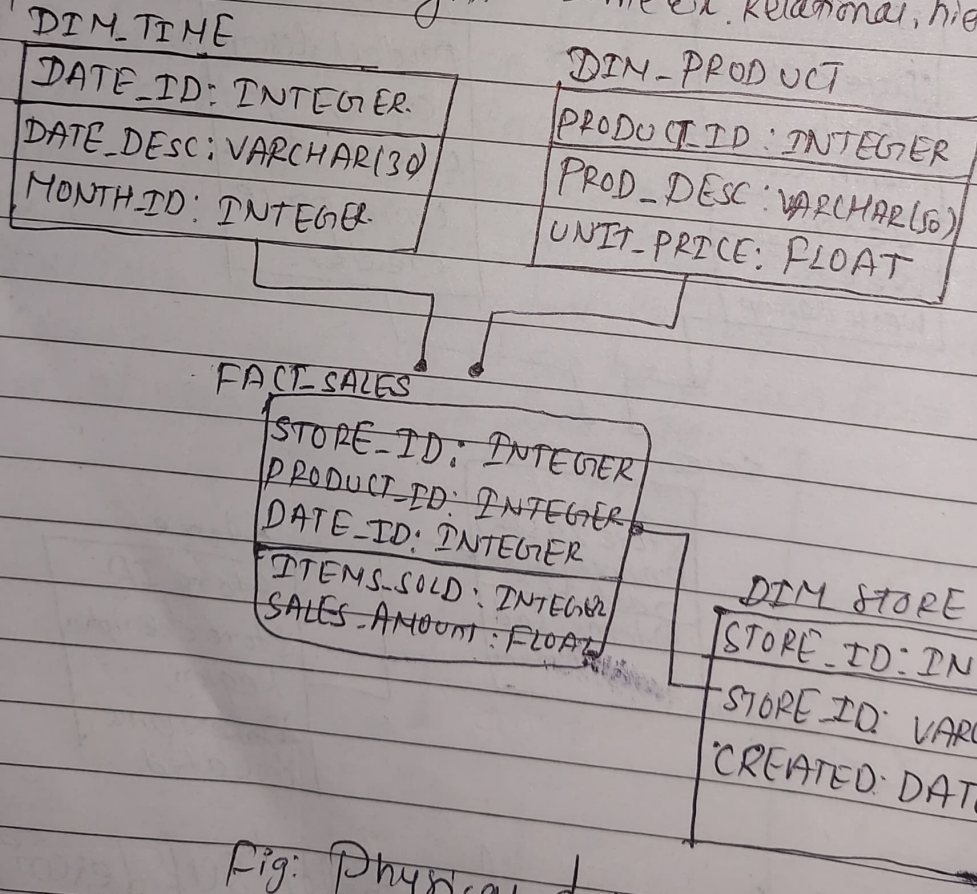


Fig: Physical data model

classmate

Note: If asked in Exam, explain types of data model, do these below.

DATE

(Some extra data models (IMP for exam view))

1. Object based data models (conceptual)

* Entity Relationship (ER) data model

ER data model is a widely used data model for database design. It describes the design of database in terms of entities and relationship among them. It ~~model~~ employs three basic notations:

1. Entity sets
2. Relationship sets
3. Attributes

Entity set: Entity is a thing in real world that has physical existence. Collection of entities of a particular entity type in a db at any point of time is called Entity set.
Ex: Employee, Company, Product etc.

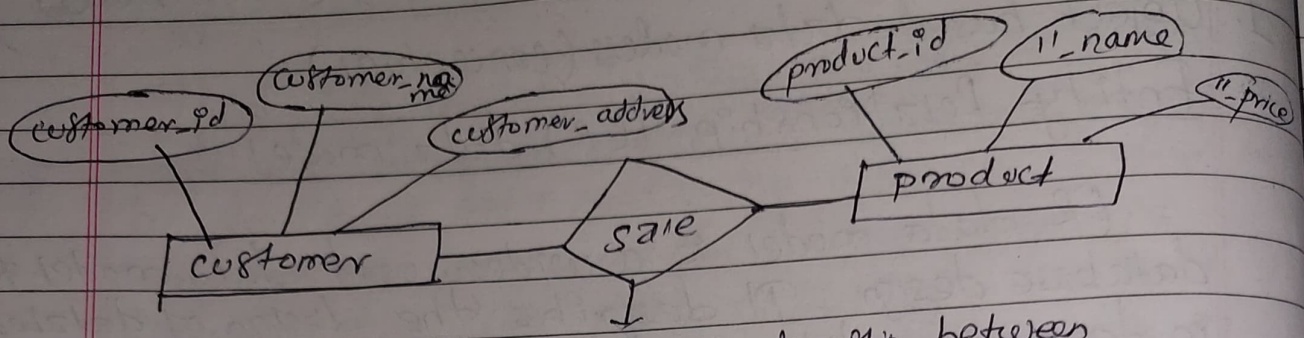
Attributes: Attributes are the components of Entities mostly table. It represents the descriptive properties possessed by each component/member of an entity set. Each entity may have its own value for each attribute.
Ex: Employee_id, company_name, product_id.

Relationship set: Relation is an association among several entities. Relationship set is a collection of relationships of same type. There are four types of relationship.

one-to-one => person $\frac{1}{}$ passport
one to many => customer - id, name
Many to one => students $\frac{M}{}$ study $\frac{1}{}$ college
Many to Many => student $\frac{M}{}$ Assigned $\frac{M}{}$ project

CLASSMATE

PAGE

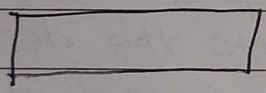
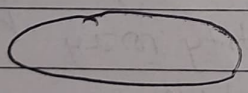
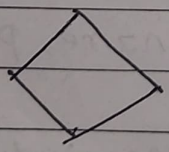
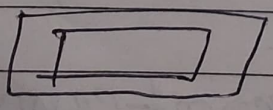
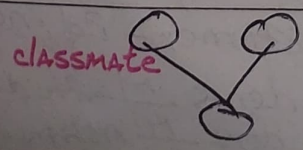


Sale in relationship between customer & product.

fig: ER Model

Entity →	Customer	Product	← Entity
attributes	customer_id	product_id	attributes
	customer_name	product_name	
	customer_address	product_price	

Components of ER diagram.

Notations	Representation	Description
	Rectangle	Entity
	Ellipse	Attribute
	Diamond	Relationship
	Line	Link between attribute & entity set to relationship set
	Double rectangle	weak entity
	Composite Attribute.	Composite attributes that can be sub divided Ex: name → fname, lname

DATE

Hierarchical Model

→ It is a data model in which the data are organized into a tree-like structure where there is a single point for each record.
→ It is one of the oldest data models.
→ This model allows the one-to-one and one-to-many relationship between various types of data.
→ One can access the records by navigating down through the data structure.
→ Data are linked to each other in parent-child relationships.

Advantages
→ Simplicity, Data sharing, Data security, Integrity, Efficiency

Disadvantages
→ Implementation complexity, Inflexibility, Database management problem, implementation limitation
→ If parent node is deleted then the child note is auto deleted.

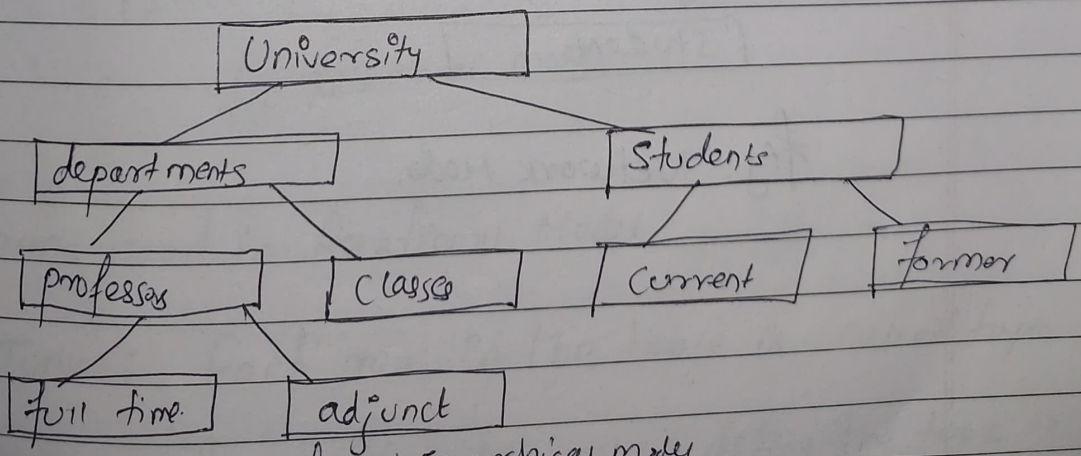


fig: Hierarchical model

PAGE

classmate Problem: Where do you put Professor who is also Student?

DATE

* Network Model

→ It was created to represent complex data relationships more effectively than the hierarchical model.

→ The main advantage of the network model is the ability to address the lack of flexibility of the hierarchical model.

→ Unlike the hierarchical model, the network model allows a record to have more than one parent to generate more complex, many-to-many relationships

Disadvantage:

→ System complexity, operational anomalies, not user friendly

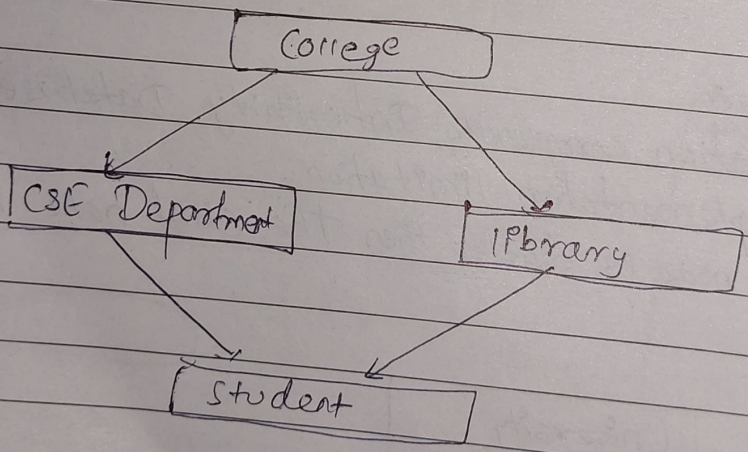


fig: Network Model.

DATE

* Relational Model.

→ It represents how data is stored in relational database.
→ The data is stored in the form of tables in a relational database.
→ The tables are also called relations in relational model.
→ The relationships between tables is maintained by using a primary key and foreign key.

Emp_id	Emp_name	Job_name	Salary	Mob_no	Dep_id	Proj_id
AfterA001	John	Engineer	100000	98469	2	99
AfterA002	Adam	Analyst	50000	98412	3	100
AfterA003	Kandee	Manager	890000	95614	2	65

Table: Relational Model.

Note:

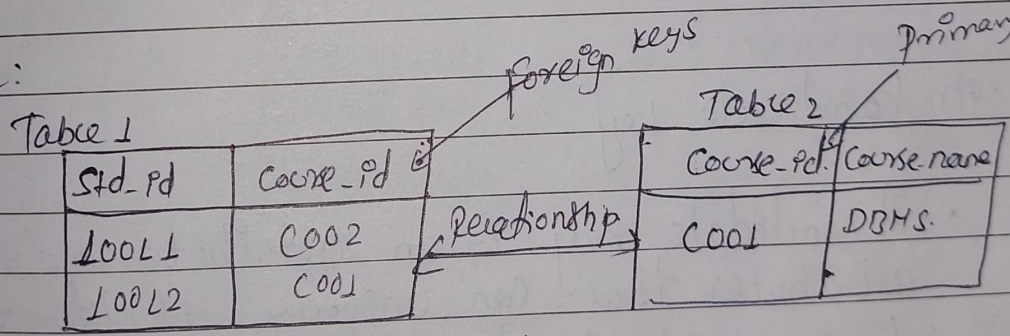


fig: Relational model.

Terms used in Relational Model:-

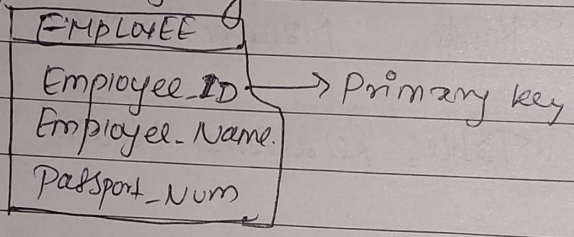
- Tuples: Each row in the table is called tuple.
- Attribute or field: Properties which define the table or relation.

DATE

- Key: It is used to uniquely identify any record or row of data from the table. It is also used to establish & identify relationship between tables.

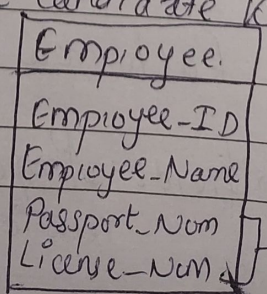
- Types of key:

i) Primary key
→ It is the first key used to identify one and only one instance of an entity quickly. An entity can contain multiple keys. The key which is most suitable from those lists become a primary key.



ii) Candidate key

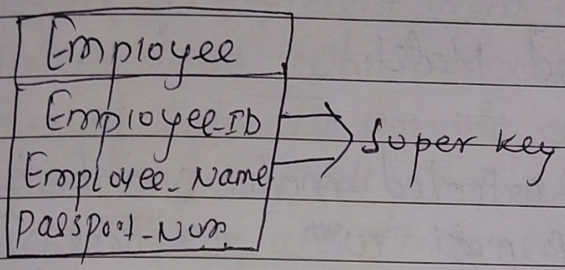
- A candidate key is an attribute or set of attributes that can uniquely identify a tuple. (uniquely identify 1 or more attributes).
- The primary key should be selected from the candidate keys.
- A table can have multiple candidate keys but only a single primary key.



DATE

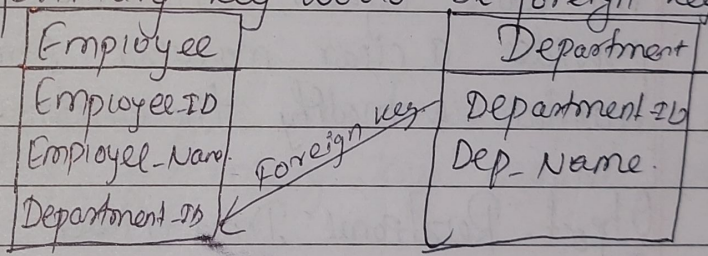
(ii) Super key:

- Super key is an attribute set that can uniquely identify a tuple. A super key is a superset of candidate key.
- In the EMPLOYEE table, the name of two employees can be same, but their EMPLOYEE_ID can't be. Hence, this combination can also be a key. (ie. Super key).



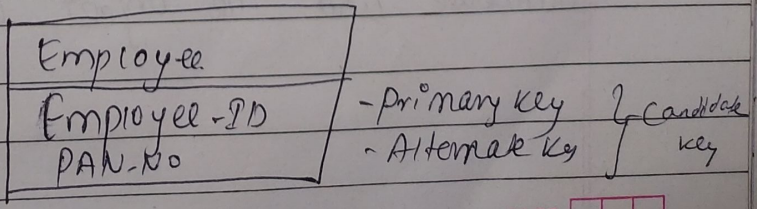
(iii) Foreign key

If the primary key of one table appears in another table, then the primary key would be foreign key for that table.



(iv) Alternate key

The total no of the alternate keys is the total number of candidate keys minus the primary key. The alternate key may or may not exist. If there is only one candidate key in a relation, it does not have an alternate key.



DATE

v) Composite key:

- Whenever a primary key consists of more than one attribute, it is known as a composite key

* Object Oriented Model.

The object-oriented model can be seen as extension of ER model with notations. It stores the data in the form of objects, classes and inheritance. This model handles more complex applications such as Geographic information system (GIS). It is used in file management system. Represents real word objects, attributes & behaviour. It provides a clear modular structure & easy to maintain & modify the existing code.

* Object-Relational Data Model.

It is the combination of an object oriented database model and a relational database model. So it supports classes, objects, inheritance etc. One of the major goals of object relational data model is to close the gap between relational database and the object oriented practices.

DATE

Write short notes on:

1) Data Abstraction.

Data abstraction is the process of hiding or removing the unnecessary or irrelevant information from the end user and providing a simplified interface. It provides a different view and helps in achieving data independence which is used to enhance the security of data. To provide data abstraction is one of the fundamental characteristics of database approach. There are three levels of abstraction:

i) Physical / Lowest level.
ii) Logical / second lowest level
iii) View level / highest level.

2) Foreign key

If the primary key of one table appears in another, then the primary key would be foreign key for that table.

It is a column or a group of columns in a relational database that provides a link between data in two tables. It acts as a cross-reference between the tables because it references the primary key of another table, thereby establishing a link between them.

PAGE

DATE

* Schemas

- In any data model, it is important to distinguish between the description of the data-base & the database itself.
- The description or the basic structure of a database is called the database schema.
- It is specified during database design & is not expected to change frequently.

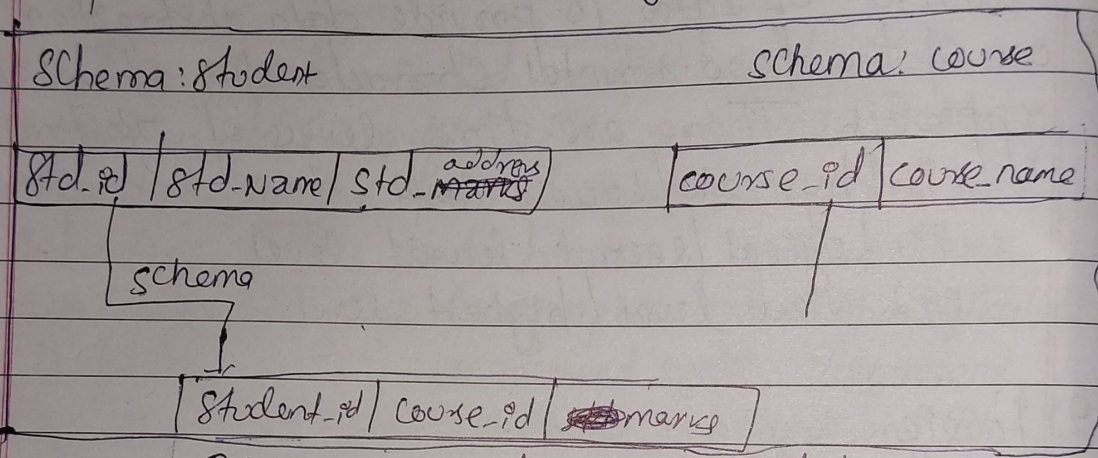


Fig: Schema diagram of a database.

* Instances

- When we define a new database, we specify its database schema only to the DBMS
- Instance is the actual content of the database at a particular point of time.
- As the value of variable varies, the databases also change over time when we insert, delete or modify the information
- Eg. in the above schema, the database changes whenever we add a new course or a student.

DATE

Three-Schema Architecture

It is one of the architecture of DBMS. The DBMS architecture describes how data in database is viewed by users. It is not concerned with how data is handled and processed by the DBMS. The database users are provided with an abstract view of data by hiding certain details of how data is physically stored. This enables users to manipulate the data without worrying about where it is located & how it is actually stored.

The three-schema approach offers three types of schema

1. (External Schema) / (View level)
   - → Also known as view level
   - → This is the highest level of database abstraction.
   - → This level describes the actual view of data that is relevant to the particular user.
   - → Ex: when users access google photos they are not known where the actual data is stored. Also different users can view different data based on their login information. So, we have made different views in external schema

2. (Conceptual Schema) / (logical level)
   - → Also known as Logical Level.
   - → It describes the design of a database at the conceptual level
   - → It describes what data are to be stored in the database & also describes what relationship exists among data.
   - → Programmers & database admin work at this level.
   - → DB users don't need to have knowledge of this level
   - → ex: ER Model

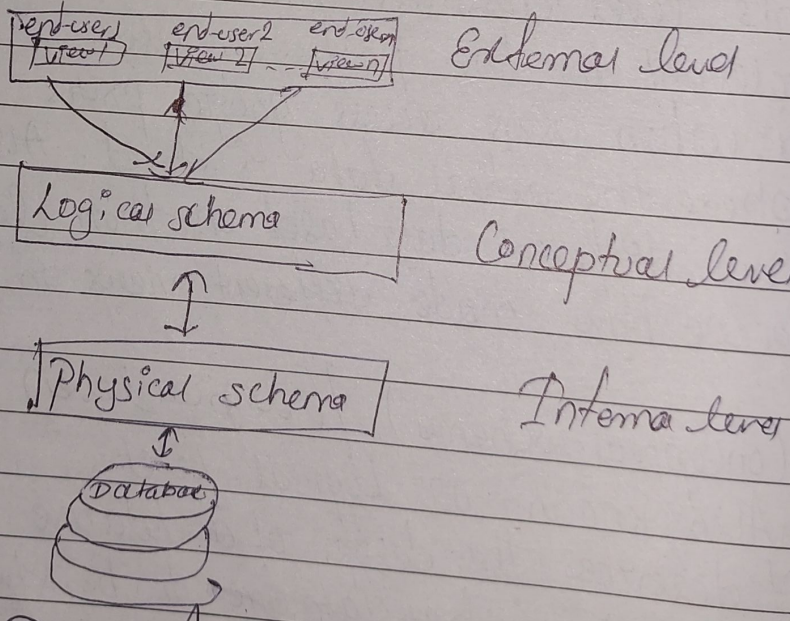
PAGE

classmate

DATE

3. (Physical schema): / (Internal level)

→ Also known as internal level.
→ It is the lowest level of abstraction
→ Describes where the actual data is physically present.
→ Uses the physical data model. Used to define that how the data will be stored in a block.
→ Complex low-level data structures are described in detail in this schema.
→ Data admin may be aware of certain details of physical organization of data



DBMS Architecture

fig. Three level schema architecture.

DATE

* Data independence.

The three schema architecture further explains about the concept of data independence. The capacity to change the schema at one level of database system without having to change the schema at the next higher level is called data independence.

Two types of data independence:

1) Logical data independence.

The capacity to change the conceptual (logical level) schema without having to change associated application programs is called logical data independence. If the underlying conceptual schema is changed, the definition of a view relation can be modified so that the same relation is computed as before. Database administrator is responsible for redefining view level schema

2) Physical data independence.

The capacity to change the internal schema without affecting application programs is called physical data independence. This means we can change physical level storage details such as file structure, indexes as long as conceptual schema remains same without altering associated application programs. But performance may be affected due to changes in physical level. DBA is responsible to manage such changes

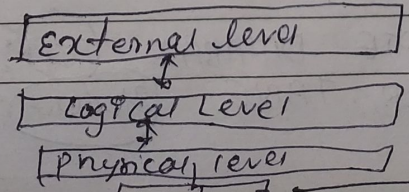


fig: data independence

Logical Level indep.

Physical level indep

DATE

* Database Languages

→ Database languages are the classification of programming languages that programmers or developers use in order to define and access the database.
→ It can be used to read, store and update the data in the database.
→ They are used basically to communicate with the database.

* Types of database languages

1) DDL (Data Definition Language)

→ It is used to define database structure & patterns
→ It is used to create schema, tables and indexes in the database
→ DDL is used to store the information of metadata like the number of tables & schemas, their indexes, columns in the table, etc.

Some commands of DDL are:
- CREATE : To create database instance
- RENAME: Use to rename database instance
- DROP : Used to delete database instance
- ALTER: To alter the database instance.

Ex: CREATE DATABASE (Test); - name
DROP TABLE (Student); - table name

classmate

PAGE

DATE

DML (Data Manipulation Language)

→ It is used for accessing and manipulating the data in the database
→ It handles the user requests (insert, update, delete)
→ Express database queries and updates

Some commands of DML are:
- SELECT: To read records from the table
- INSERT: To insert records in the table.
- UPDATE: To update data in the table
- DELETE: To delete all the records from the table.

Ex: INSERT INTO table_name VALUES(data1, data2, ...)
ex: consider a table student with following fields:

S_id	name	age.

INSERT INTO student VALUES (101, 'Adam', 15);

DCL (Data Control Language)

→ It is used for granting and revoking access in the db.
→ To perform any operations in the database like creating tables, we need privileges: Such privileges are controlled by DCL.

Some Commands:
- GRANT: To grant access to the user
- REVOKE: To revoke(take back) the access from the user.

Ex: GRANT CREATE TABLE TO username;
REVOKE CREATE TABLE FROM username;

DATE

iv) TCL (Transaction Control Language)

→ It provides commands that are used to manage transactions in the database
Some important commands:

- COMMIT: To save transactions in the database
- ROLLBACK: To restore the database to the original since the last commit
- SAVEPOINT: Used to temporarily save the transaction so you can rollback to previous point

v) Storage Definition Language (SDL):

→ It is used to define internal schema. It defines that what will be the physical data structure of database like how many bites per field will be used, what will be the order of fields and how records will be accessed.

vi) View Definition Language (VDL):

→ This language is used to specify user views & their mapping to conceptual schema. It defines the subset of records available to classes of users. It creates virtual tables & the view appears to appears to users like conceptual level. It specifies user interface

Implementation of commands (for practical)

Source: www.studypoint.com

* Creating a Database:-

Syntax: `CREATE DATABASE <DB-NAME>;`

Ex: `CREATE DATABASE Test;`

The above command will create a database named Test, which will be an empty schema without any table.

* Creating a Table:-

Syntax:

```
CREATE TABLE <TABLE_NAME>
(
column_name1 datatype1,
column_name2 datatype2
);
```

Ex:

```
CREATE TABLE student (
student_id INT,
name VARCHAR(100),
age INT );
```

(commas.)

The above command will create a new table with name student in current database with 3 columns namely student_id, name, age.

Ex. If we have a database with name Test & we want to create a table student in it, then

```
CREATE TABLE Test.student (
student_id INT,
name VARCHAR(100),
age INT );
```

DATE

* ALTER Command: Add a new column

Using ALTER command we can add a column to any existing table.

Syntax:
```
ALTER TABLE table_name ADD (
    column_name datatype);
```

Ex:
```
ALTER TABLE Student ADD (
    address VARCHAR(200)
    );
```

The above command will add a new column address to the table student, which will hold data of type varchar which is nothing but string of length 200.

* ALTER command: Drop a column

Syntax:
```
ALTER TABLE table_name DROP (column_name);
```

Ex:
```
ALTER TABLE Student DROP (address);
```

* TRUNCATE Command: Removes all records from table but not destroy the table's structure

Syntax:
```
TRUNCATE TABLE table_name
```

Ex:
```
TRUNCATE TABLE Student;
```

DATE

* DROP Command: Completely removes a table from DB. It will also destroy table structure & data stored in it.

Syntax: DROP TABLE table_name;
Ex:- DROP TABLE student;

To delete a database:
DROP DATABASE Test;

* RENAME query:- Used to set new name to existing table.

Syntax: RENAME TABLE old_name to new_name
Ex: RENAME TABLE student to students_info;

* INSERT query:- Insert a data into a table.

Syntax: INSERT INTO table_name VALUES (data1, data2...
Ex: INSERT INTO student VALUES (101, 'Ankit', 15);

Output:

S_id	name	age
101	Ankit	15

Ex: INSERT INTO student VALUES(null, 'Ankit', 15);
=> id will be empty.

* INSERT into specific columns:
Ex: INSERT INTO student(id, name) values (102, 'Alex');

PAGE

DATE

Interfaces in DBMS.

A DBMS interface is a user interface that allows the users to query the database without using a query language itself.

UI provided by DBMS are

1. Menu-Based Interface.

→ These interfaces present the users with a list of options (menus)
→ Here, users don't need to memorize specific commands or syntax for query language
→ Pull-down menus is most popular
→ other example is browsing on online store

2. Forms-Based Interface.

→ Displays a form to each user where the users can fill the necessary data.
→ Designed & programmers for naive users
→ Ex: when taking online attendance, or checking result online by filling online forms

3. Graphical user Interfaces

→ It is a type of interface through which users interact with schema in diagrammatic form.

DATE

Users can specify the query by manipulating the diagram.
→ In most of the cases, it uses both menu & form based interface (like searching youtube videos and choosing accordingly).

4. Natural Language Interface.

→ These are the simple interfaces where the system and user communicate via natural language ie. human language.
→ It has its own schema and dictionary.
- Ex: Google search.

5. Speech Input and output

→ It is the most common type of interface used in today's world where user queries the interface in the form of speech & gets response the same way.

→ Here also the natural language is used by the interpreter & finds related keyword so that it can retrive data from database.

→ Ex: Google Assistant, Alexa, Siri, etc.

PAGE

DATE

Database System Environment

A database system environment is a collective system of components that compromise and regulates the groups of data, management, and use of data, which consist of hardware, software, people, techniques of handling database, and the data also

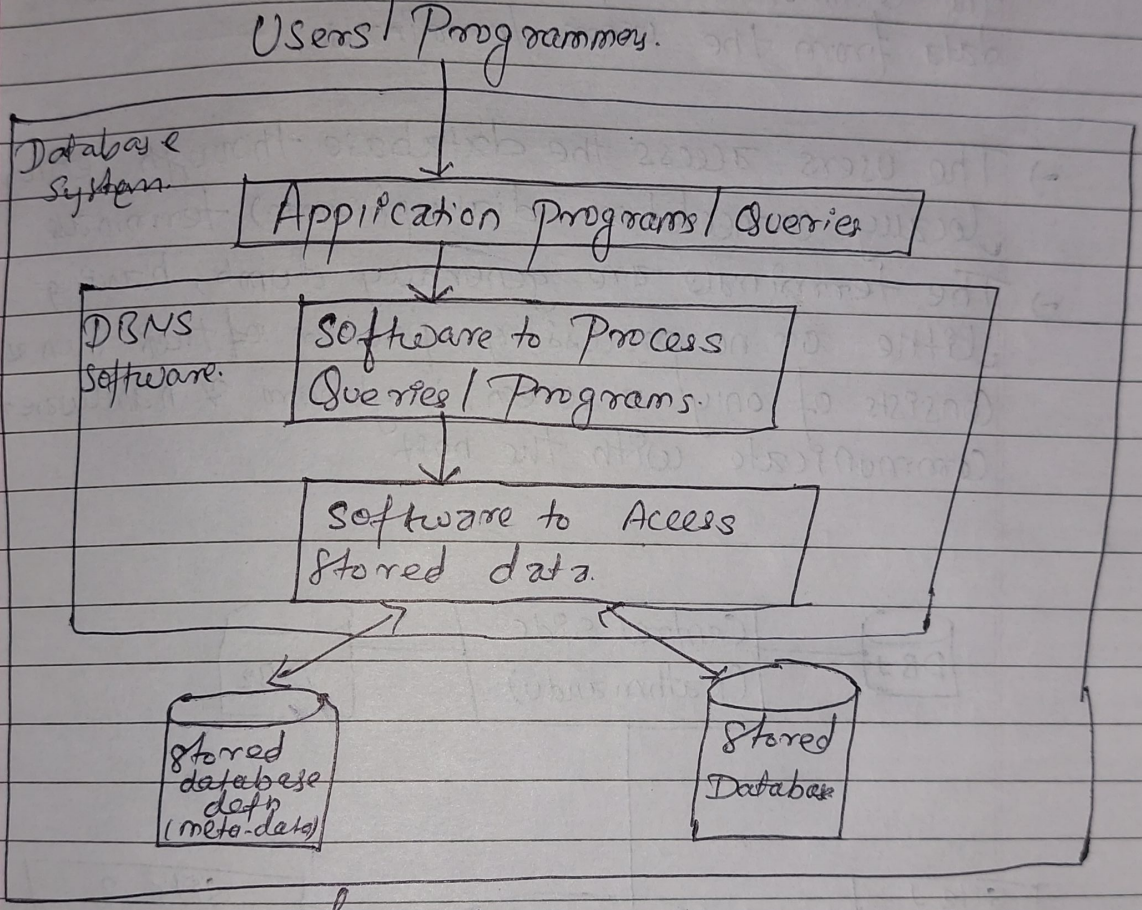


Fig: Database System environment.

→ Hardware in database environment means the computers & computer pheripherals used to manage database.

→ Software means the whole thing right from os to the application programs that include database management software like MS. Access or SQL server.

→ People in database environment include include those people who administrate use the system

→ Techniques are the rules, concepts & instructions to manage db.

PAGE

CLASSMATE

DATE

* Types of database Systems

1) Centralized Database System

→ In centralized system, all programs run on the main host computer, including the DBMS, the application that access the database and the communication facilities that send or receive data from the users terminals.

→ The users access the database through either locally connected or dial-up(remote) terminals
→ The terminals are generally dumb, having little or no processing power of their own & consists of only a screen, keyboard & hardware to communicate with the host.

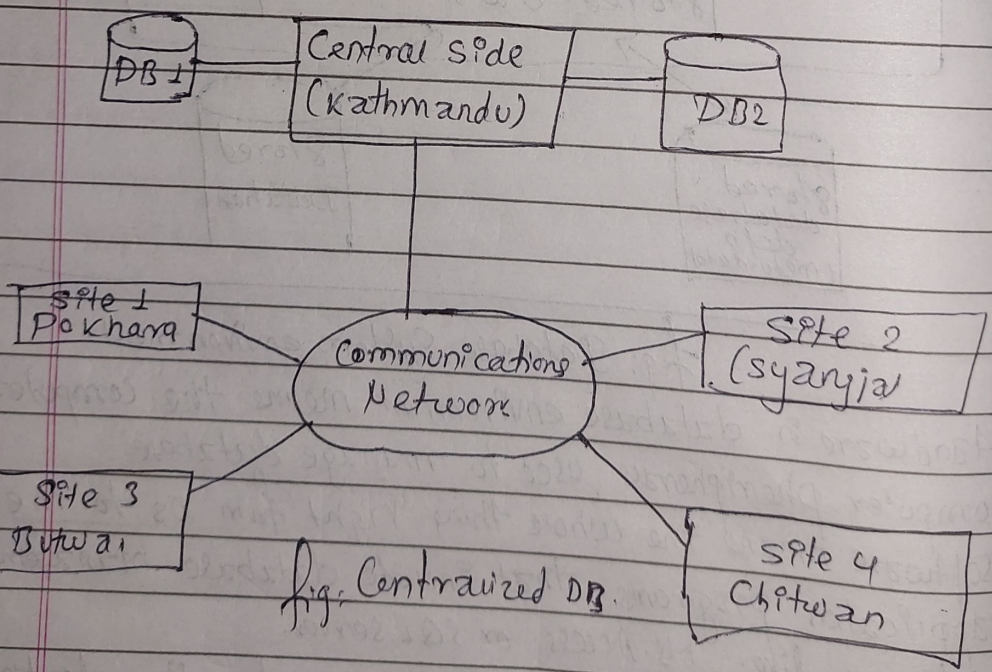


fig: Centralized DB.

DATE

2) Client-Server Database System

→ In a generalized concept, client PC is the computer from where the user requests for data and information and the server provides the requested information.

→ The database application on the client pc referred to as the 'front end system' that handles all the screen and user input/output processing.

→ The 'back end system' on the database server handles data processing and disk access.

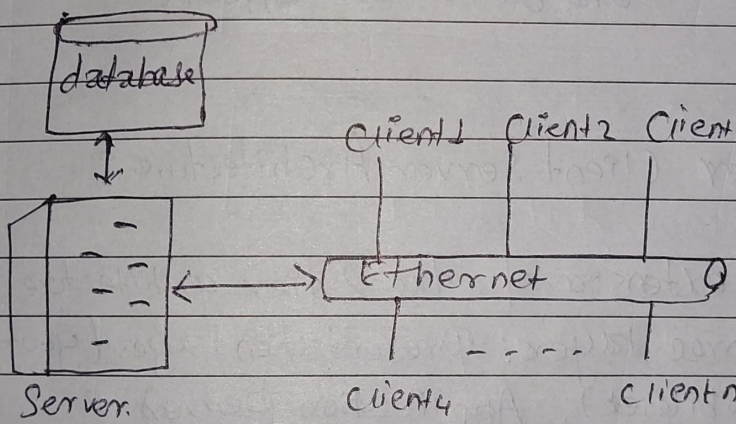


fig: Client server database system

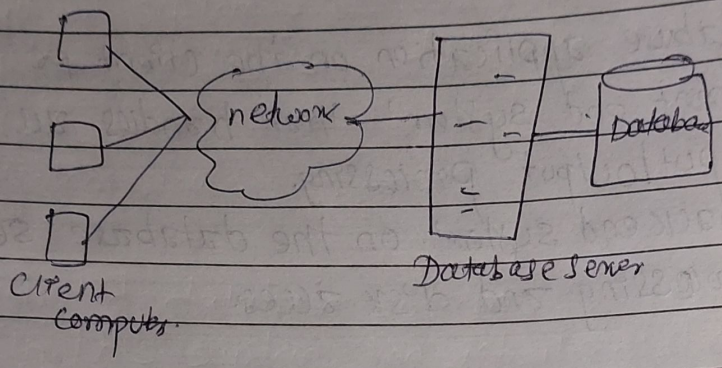
→ Client server approach is implemented by two approaches two-tier architecture & three-tier architecture.

- Two-tier architecture:
  - → The user interface & application programs are placed on the client side & database system on server side.
  - → The application programs that reside at client side ~~revoke~~ invoke the DBMS at server side.

DATE

→ Open Database connectivity (ODBC) and JAVA Database connectivity (JDBC) are used for interaction between client & server.

→ Because of tight coupling 2 tier application will run faster.



→ Advantages: Easy to understand, modify & maintain

→ Disadvantage: Time consuming when large users and little cost effective.

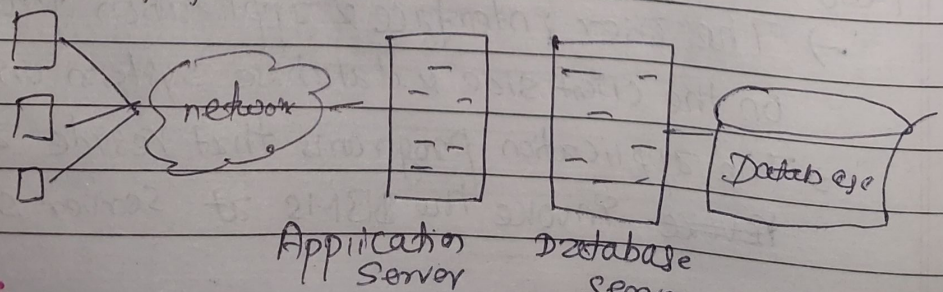
• Three tier Client-Server Architecture:

→ It is an extension of the 2-tier architecture.

→ It has three layers: Presentation layer (your PC, Tablet, Mobile, etc), Application (Server) layer & Database server.

→ The application layer is responsible for communicating the user's request to DBMS system & send response from DBMS system to user.

→ It is more popular DBMS architecture.



classmate

PAGE

DATE

→ Advantages: Easy to maintain & modify, Improved security and good performance.

→ Disadvantage: It is little more complex & little more effort is required in terms of hitting the db.

* Classification of Database Management system.

i) Classification based on data model.

→ The most popular data model used today is the relational data model.

→ Well known DBMS like Oracle, MS SQL Server, support this model.

→ Other traditional models like hierarchical, network data models are still used in industry mainly on mainframe platforms. However, they are not commonly used due to their complexity.

→ In recent years, newer object-oriented were introduced where information is represented in the form of objects as used in object-oriented programming.

ii) Classification based on user numbers.

→ Based on the number of users supported by DBMS: It can be classified into single-user database system which supports one user at a time, or a multiuser database system, which supports multiple users concurrently.

→ In multiuser DBMS, the data is both integrated & shared.

CLASSMATE PAGE

DATE

(iii) Classification based on number of sites:

Based on the number of sites of DBMS, it has been classified into two types

* Centralized systems → With a centralized database system, the DBMS & database are stored at a single site that is used by several other systems too

* Distributed database system:
In this system, the actual database & the DBMS software are distributed from various sites that are connected by a computer network.

(iv) Classification based on type of access path.

* Homogeneous distributed database systems:
They use the same DBMS software for multiple sites. Data exchange between these various sites can be handled easily.

* Heterogeneous distributed database systems
In this system different sites might use different DBMS software, but there is additional common software to support the data exchange between these sites.

DATE

Old questions asked from this Chapter

Q. What is data abstraction? What are the three levels of abstraction? Explain (5 marks - 2076

Q. Explain different data models with example. (7 marks-2078)

Q. What is the difference between Logical & physical data independence? (2078 - 5 marks)

Q. What are the components of ER diagram? Explain the function of various symbols used in ER diagram. (6 marks) 2078

Q. Explain the data independence with example. (2078-5 marks)

Q. Explain: a) Three-schema architecture
b) Data abstraction.
c) Two-tier & three-tier client/server arch.

Q. Difference between distributed & client-server DBMS. (2073 - 5 marks)

Q. What is data definition language? How is it different from data manipulation language? (2075 - 5 marks)

Q. What is database system architecture? Describe three levels & benefits of this architecture. (5 marks)

PAGE