Varun Khadayate A016

Page No.
Date 8 12 2020

DBMS

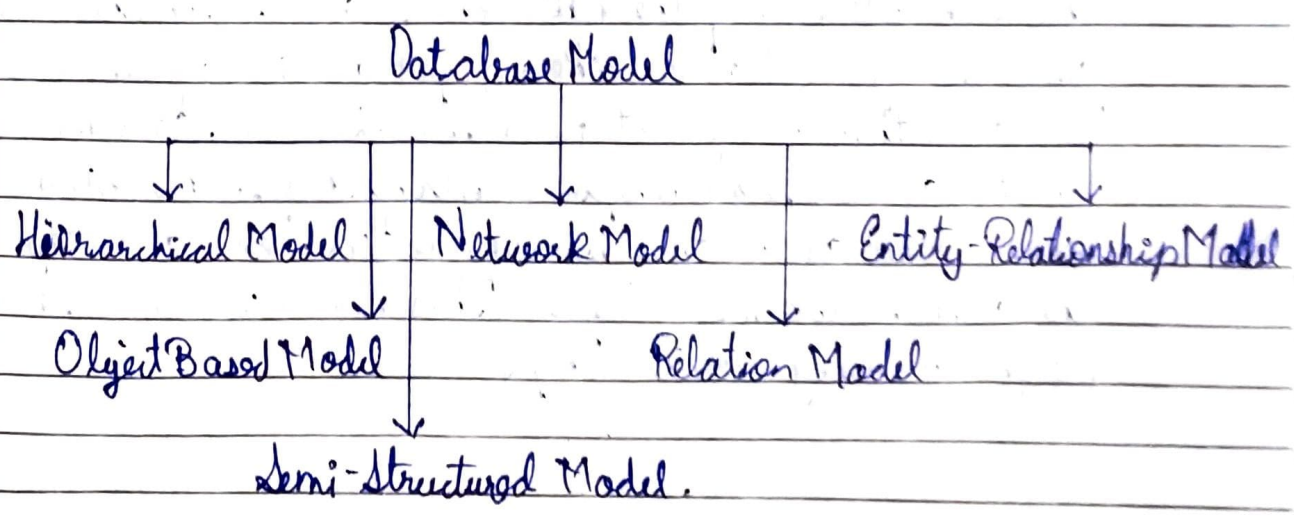
Assignment - I

Q1. Database Management System (DBMS) is a software for storing and retrieving users data while considering appropriate security measures. It consists of a group of programs which manipulate the database. The DBMS accepts the request for data from an application and instructs the operating System to provide the specific data. In a large systems, a DBMS helps users & other 3rd party software to store & retrieve data.

Advantages

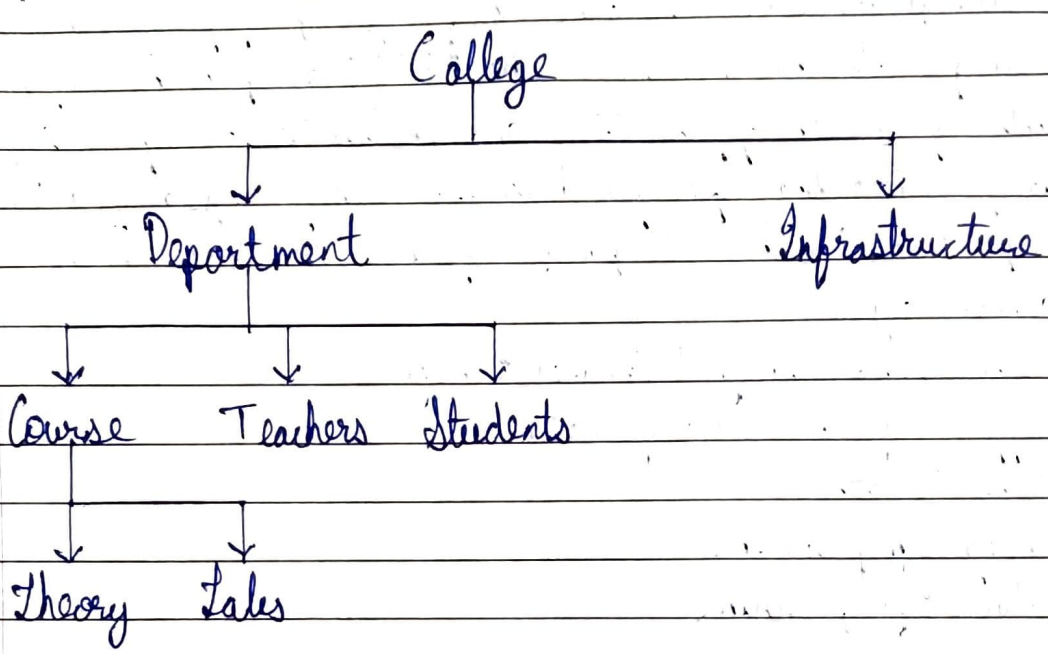
1. Data redundancy & inconsistency
2. Difficulty in accessing data
3. Data isolation
4. Integrity problems
5. Atomicity problems
6. Concurrent - access anomalies
7. Security problems.

Q2.



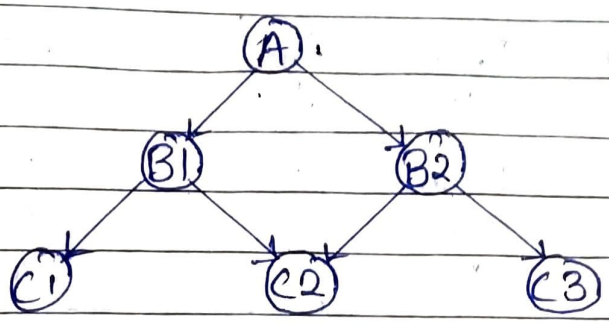
1. Hierarchical Model

This database model organises data into tree-like-structure, with a single root, to which all the other data is linked. The hierarchy starts from the Root data, and expands like a tree, adding child nodes to the parent nodes. In this model, a child will only have one parent node.



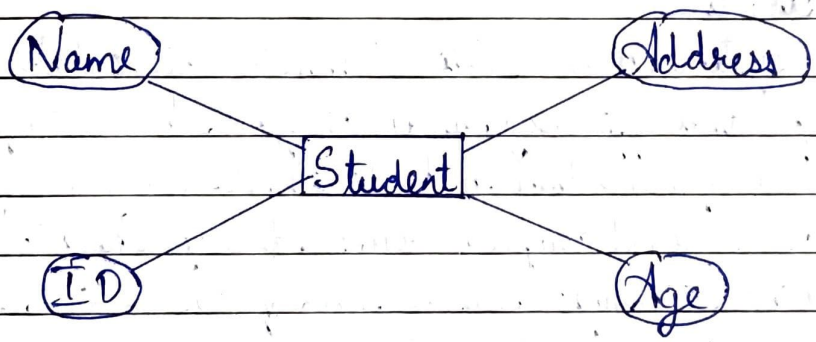
2. Network Model

This is an extension of the Hierarchical model. In this model data is organised more like a graph, and are allowed to have more than one parent node. In this model, data is more related as more relationships are established in this database model. Also, as the data is more related, hence accessing the data is also easier & fast. This model was used to many-to-many relationships.



3. Entity-Relationship Models

In this model, relationships are created by dividing object of interest into entity & its characteristics into attributes. Different entities are related using relationships. This models are defined to represent the relationship into pictorial form to make it easier to understand. This model is good to design a database, which can then be turned into tables in relation model.



4. Relational Model

In this model, data is organised in 2-D table & the relationship is maintained by storing a common field. The basic structure of data in relational model is tables. All the information related to a particular type is stored in rows of the table.

Student_Id	Name	Age
1	Akon	17
2	Bkon	18
3	Ckon	17
4	Dkon	18

Subject_Id	Name	Teacher
1	Java	Mr J
2	C++	Miss C
3	C#	Mr C Hash
4	Php	Mr PHP

Student_Id	Subject_Id	Marks
1	1	98
1	2	78
2	1	76
3	2	88

Q3. Data Abstraction

For the system to be usable, it must retrieve data efficiently. The need for ~~data~~ efficiency has led designers to use complex data structures to represent data in the database. Since many database system users are not computer trained, developers hide the complexity from the users through several levels of abstraction, to simplify users interaction with the system:

1. Physical Level

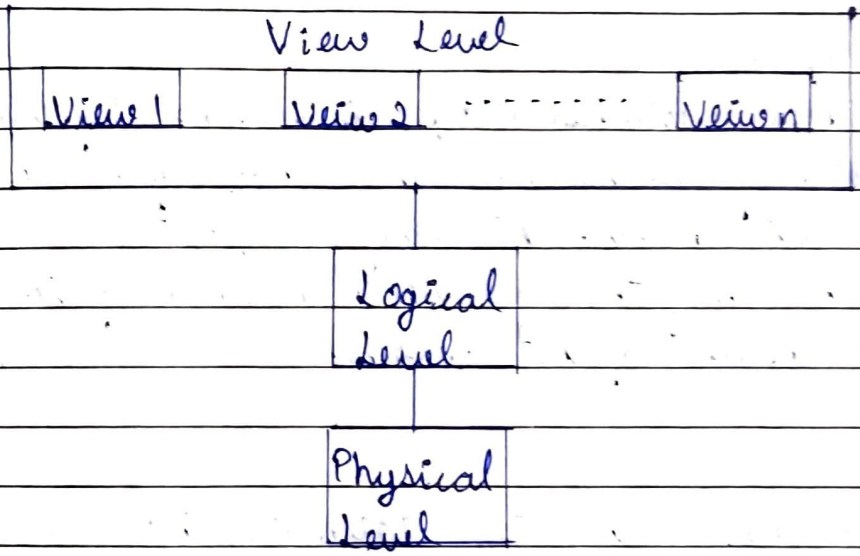
The lowest level of abstraction describes how the data are actually stored. The physical level describes complex low-level data structures in detail.

2. Logical Level

The next-higher level of abstraction describes what data are stored in the database, and what relationships exist among those data. The logical level thus describes the entire database in terms of a small number of relatively simple structures.

3. View Level

The highest level of abstraction describes only part of the entire database. Even though the logical level uses simpler structures, complexity remains because of the variety of information stored in a large database.



DBMS Schemas

Design of a database is called the Schema. Schema is of 3 types: Physical Schema, Logical Schema & View Schema.

The design of a database at physical level is called Physical Schema, in this level how the data to be stored in blocks of storage is described here.

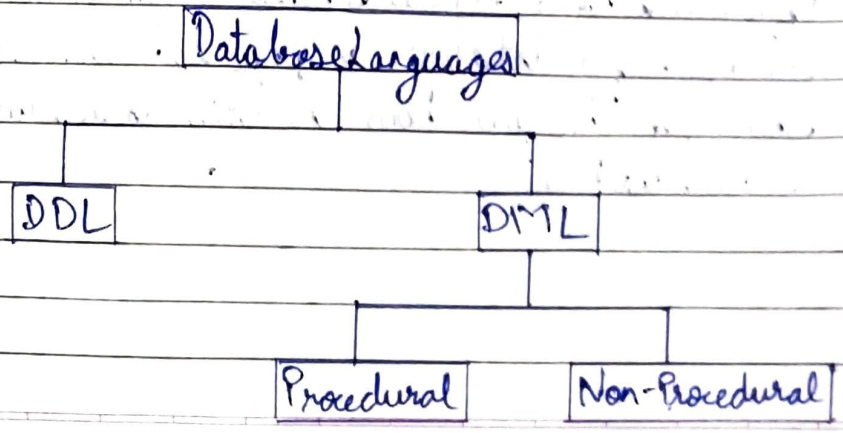
Design of database at logical level is called Logical Schema, programmers & database administrators work at this level, at this level data can be described as certain types of data records gets stored at this level in data structures, however the internal details such as implementation of data structure is hidden at this level.

Design of database at view level is called View Schema. This generally describes end user interaction with database systems.

DBMS Instances

The data stored in database at a particular moment of time is called instance of database. Database schema defines the variable declaration in table that belong to a particular database, the value of this variables at a moment of time is called the instance of that database.

Q4. Database Languages are the set of statements, that are used to define & manipulate a database. A Database Language has Data Definition Language [DDL] which is used to construct a database & it has Data Manipulation Language [DML] which is used to access a database. Eg: Oracle, SQL etc.



Page No.
Date

Data Definition Language [DDL]

DDL defines the statement to implement the database schema. If a clear seperation between logical & physical level is not there, then DDL defines both the logical & physical schemas & it also defines the mapping between logical & physical schema.

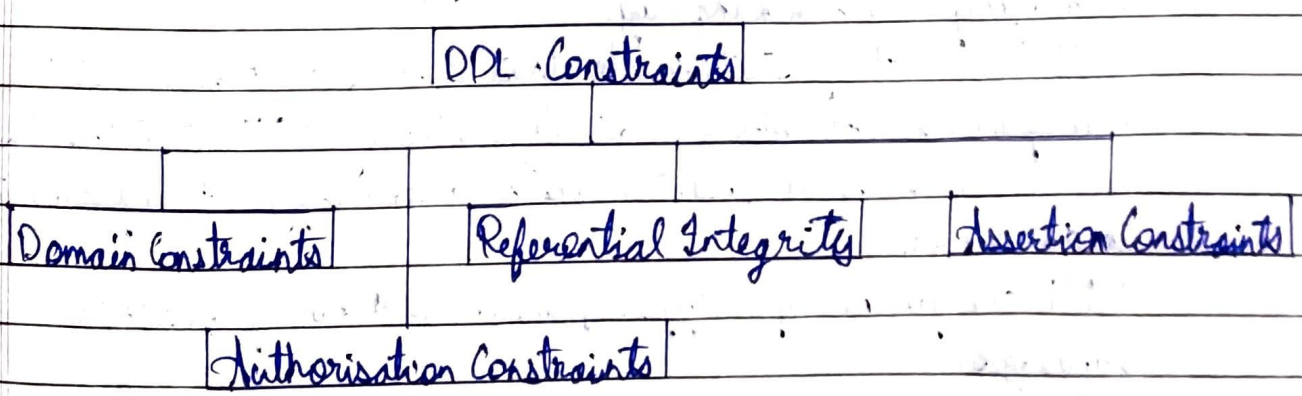
If there is a clear seperation between the logical & physical schema, then the Storage Defination Language [SDL] is used to define the physical schema.

After implementing logical & physical schema, its time to specify the view schema. For that View Defination Language [VDL] is used which also maps the view schema to the logical schema.

The set of statements in DDL used to implement database schema are as follows:

CREATE : Used to construct a relation in database.
ALTER : Used to reconstruct the data in database.
DROP : Used to delete a relation in the database.
TRUNCATE: Deletes all the entries from the relation but keeps the relation structure secured in the database.
RENAME : Renames the relation in a database.

Well, it DDL also defines some consistency constraints on data stored in database.



1. Domain Constraints: Whenever we define any attribute in the database, we must specify its domain. The domain of any attribute defines the constraints on the values that an attribute can take. Eg: Specifying an attribute it will only take that input.
2. Referential Integrity Constraints: A value of for a given set of attributes in one relation must also appear for the same set of attribute attributes in another relation. Eg: Record of address of name of a person. Address must be present in location relation.
3. Assertion Constraints: A constraint that must always be satisfied in a database is assertion constraint. Like, domain constraint & referential integrity constraint are also an assertion constraints. If the assertion constraint is violated the modification is rejected. Eg: Adding AADHAR NO. in form. Since it's mandatory it will not be NULL.
4. Authorization Constraints: We cannot allow every user to access & modify the database. So, certain authorization constraints are introduced those are READ, INSERT, UPDATE, DELETE which allows the user to read, add new data, modify the database & delete the data in database.

Data Manipulation Language [DML]

DML has a set of statements that allows user the access & manipulate the data in database. Using DML statements users can retrieve, insert, delete or modify the information in database.

DML are further of 2 types procedural & non-procedural languages:

1. Procedural DML's : Procedural DML are considered to be low-level languages, and they define what data needed & how to obtain that data. Also called as one-at-a-time DML's as it retrieves & process each record seperately.

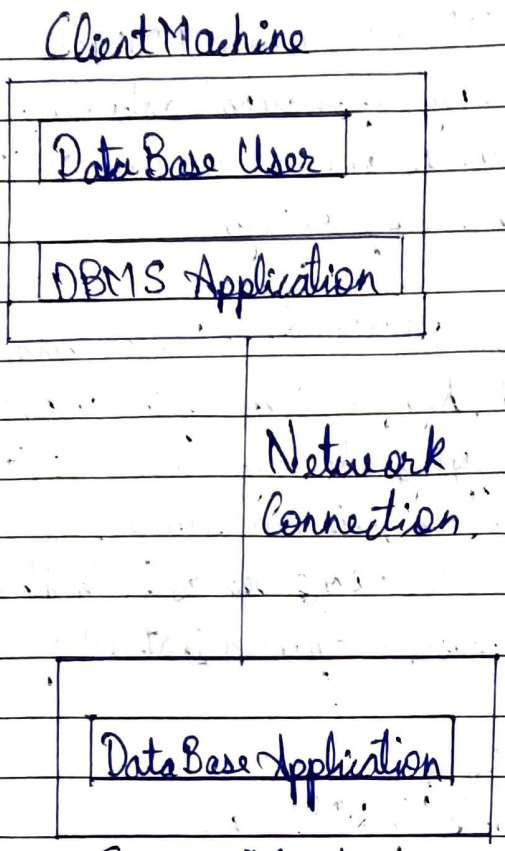
2. Non-Procedural DML's : Non-Procedural DML are high-level languages, and they precisely define what data is required without specifying the way to access it. Also called as set-a-time DML's as they can retrieve several records using a single DML command. They are also called as Declarative Languages as it only declares what data is required instead of showing how it should be obtained.

Some Statements of DML's are :

SELECT : Reads & pulls out the record from the database.
INSERT : Adds new record to the database.
UPDATE : Modifies the data in database.
DELETE : Deletes the record in database.

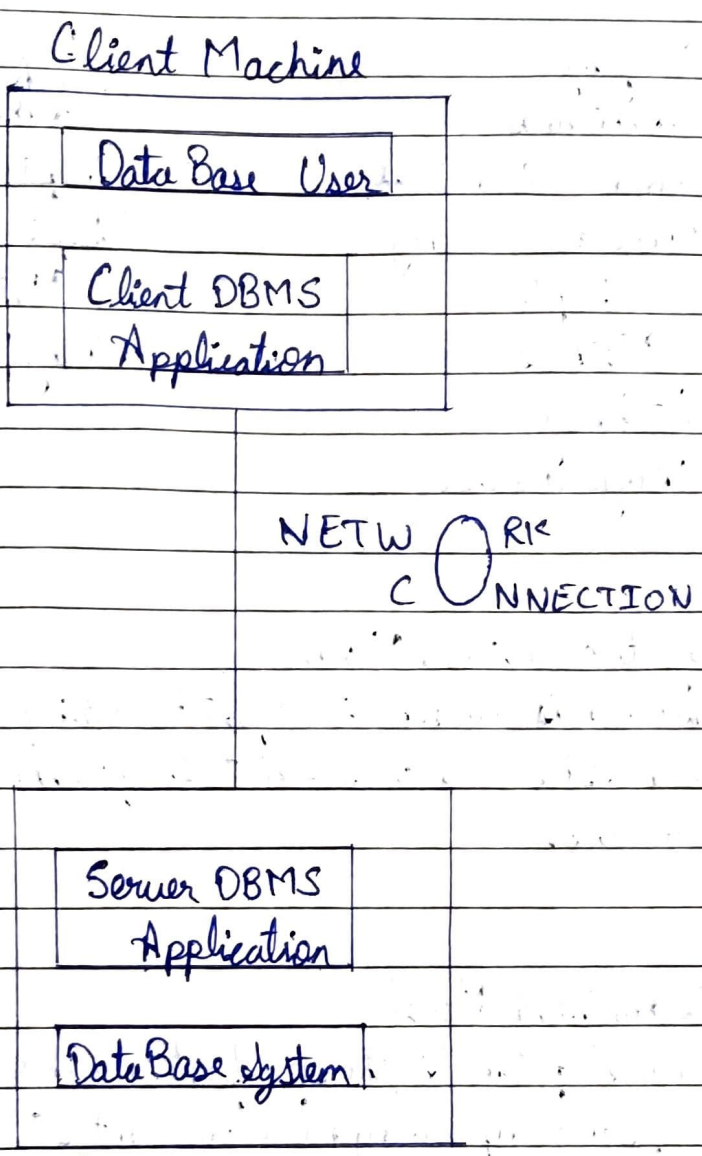
Q5. TWO-TIER

In 2 tier architecture, the Database System is present at Server Machine and the DBMS-Application is present at client machine. The given 2 machines are connected to each other via a secured network. So, whenever the client machine, makes a request to access the database present at server using a query like SQL, the server performs the request on the database & returns the result back to the client. And, for the interaction between server & client the application connection interface used is JDBC [Joint DataBase Connectivity] or ODBC [Open DataBase Connectivity].



THREE-TIER

In this architecture, another layer is present between the client machine & server machine. Here, the client application doesn't communicate the database system present at server machine directly, rather the client application communicates with server application and the server application internally communicates with the database system present at the server.

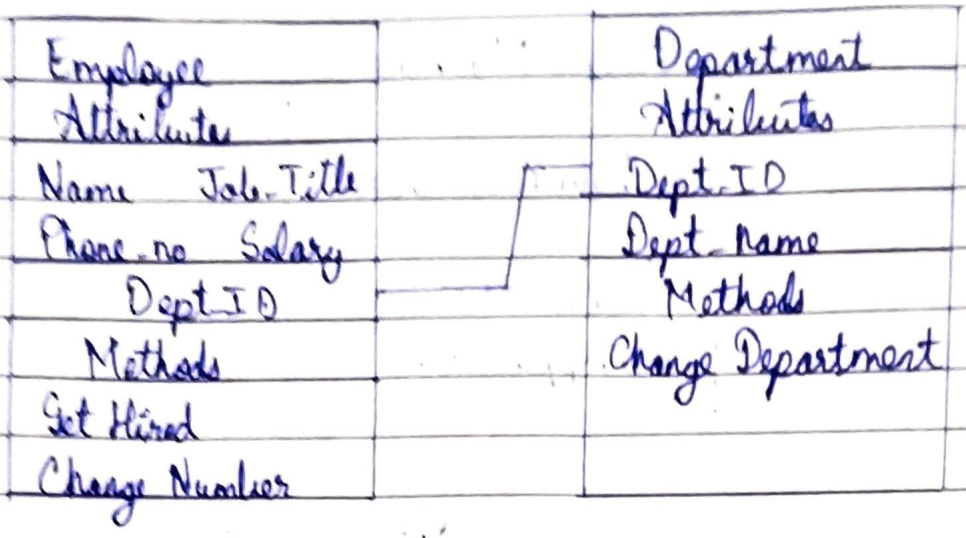


Q2. CONTD.

5. Object-Based Model

- Object-Oriented Data Model

This model can be more closely related with real-world problems. Here, both data & relationship are present in a single structure known as an object. Here, 2 or more objects are connected through links. We use this link to create one object to another.



- Object Relational Model
This model is combination of both relational & object-oriented model. This model was built to fill the gap between this 2 models.

6. Semistructured Model
This model permits the specification where individual data items of same type may have different set of attributes. The XML is widely used to represent the same model. Here there is no seperation between data & schema.

NOTE: Network & Hierarchical Models are Traditional models.