UNIT-I

Data	Information.
1. Data refers to raw facts that have no specific meaning	1. Information refers to processed data that has a meaning.
2. The word 'data' is derived from the latin word 'datum' which means 'something that is given'	2. the word 'information' is derived from the latin word 'informatio' which means 'formation' or 'conception'.
3. The data is independent of the information	3. Information is dependent on the data.
4. Data is not enough to make a decision	4. The information is sufficient to make a decision.

Database :-

A database is an organized collection of data stored and accessed electronically.

Database Management System :-

A database-management system (DBMS) is a collection of interrelated data and a set of programs to access those data.

Difference between File System and DBMS

File System	DBMS
1. It is used to manage and organize the files stored in the hard disk of the computer.	1. A software used to store and retrieve the user data.
2. Redundant data is present	2. No presence of redundant data
3. Data consistency is low	3. Data consistency is high.
4. Less security	4. More security.
5. Less expensive	5. More expensive.
6. Does not support crash recovery	6. Supports crash recovery.
7. Does not support complicated transactions	7. Supports complicated transactions.

Drawbacks of file system

* Data Redundancy
* Data Inconsistency.
* Data Isolation
* Dependency on Application programs
* Atomicity Issues
* Data Security.

Advantages of DBMS over file system.

* No redundant data.
* Data consistency and Integrity.
* Data Security
* privacy
* Easy access to data.
* Easy recovery
* Flexible.

Data Abstraction

Data Abstraction refers to the act of representing essential details and hiding the internal details.

Data abstraction can be achieved in three levels

1. physical level
2. Logical level
3. view level.

Physical level :-

The lowest level of abstraction describes how the data are actually stored.

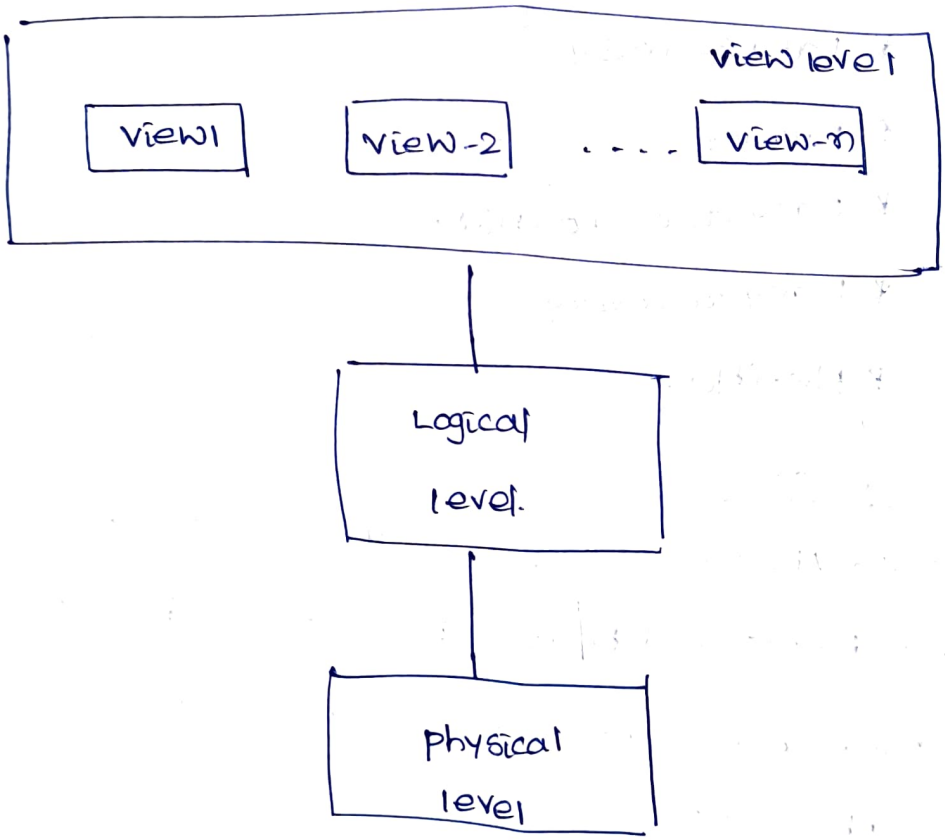
It describes complex low-level data structures in detail.

Logical level :-

It describes what data are stored in the database and what relationships exist among those data.

View level :-

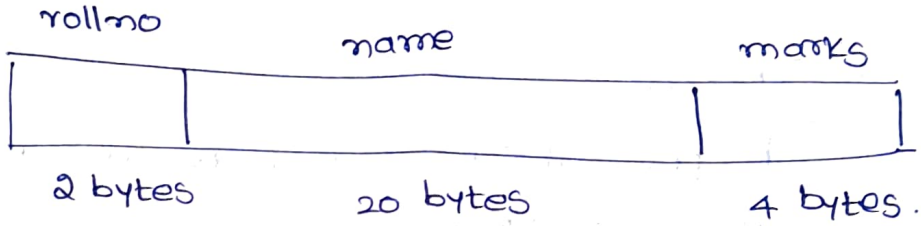
It describes only a part of the database.



The three levels of Abstraction.

```
Struct  student
{
    int rollno;
    char name[20];
    float marks;
};
```

// A sample code segment of c-language.



The physical level describes the allocation of memory for member variables of structure.

The logical level describes the creation of variables for the structure.

The view level can hide the internal details and projects only the required details.

Database Schema

The overall design of the database is called database schema.

Database systems have three types of schemas according to the levels of abstraction.

1. physical schema
2. Logical schema
3. View schema.

Physical Schema:-

The physical schema describes the database design at the physical level.

Logical Schema:-

The logical schema describes the database design at the logical level.

View Schema:-

The view schema describes the database design at the view level.

Instance of a Database

The collection of information stored in the database at a particular moment is called an instance of the database.

Applications of Database System

Databases are widely used. The major applications of database system include:

* **Banking:**

  For customer information, accounts, loans and banking transactions

* **Airlines:**

  For reservations and schedule information.

* **Universities:**

  For student information, course registrations and grades

* **Credit card transactions:**

  For purchases on credit cards and generation of monthly statements.

* **Telecommunication:**

  For keeping records of calls made, generating monthly bills, maintaining balances on prepaid calling cards and storing information about the communication networks.

* **Finance:**

  For storing information about holdings, sales and purchases of financial instruments such as stocks and bonds; also for storing real-time marketing data.

* Sales :

For customer, product and purchase information.

* On-line retailers :

On-line order tracking, generation of recommendation lists and maintenance of on-line product evaluations.

* Manufacturing :

For management of the supply chain and for tracking production of items in factories, inventories of items in warehouses and stores and orders for items.

* Human resources :

For information about employees, salaries, payroll taxes, benefits and for generation of paychecks.

Database Users

The primary goal of a database system is to store information and retriere information from the database.

The people who work with database are known as Database users.

Database users can be divided into four types.

1. Naive users
2. Application programmers
3. Sophisticated Users
4. Specialized Users.

1. Naive Users :-

Naive users are unsophisticated users who interact with the system by invoking one of the application programs that have been written previously.

E.g. A bank teller who needs to transfer Rs. 50 from account A to account B invokes a program called transfer.

2. Application Programmers :-

Application programmers are computer professionals who write application programs.

An application programmer can use Rapid Application Development (RAD) tools to develop user interfaces.

3. Sophisticated Users :-

Sophisticated users interact with the system without writing programs.

They form their requests in a database query language.

4. Specialized Users :-

Specialized users are sophisticated users who write specialized database applications that do not fit into the traditional data-processing framework.

E.g. Computer-aided design systems

Knowledge-based systems

Expert systems.

Database Administrator (DBA)

A person who has central control over the database management system is called a Database Administrator.

Functions of DBA

The functions of a DataBase Administrator include:

* Schema definition
* Storage structure and access-method definition
* Schema and physical-organization modification
* Granting of authorization for data access.
* Routine Maintenance.

* Schema definition

The DBA creates the original database schema by executing a set of data definition statements in the Data Definition Language (DDL).

* Storage Structure and Access-method definition.

The DBA is responsible for choosing the Storage structure and defining the access methods for retrieval of data.

* Schema and physical-organization modification.

The DBA carries out changes to the schema and physical organization to reflect the changing needs of the organization or to alter the physical organization to improve performance.

* Granting of authorization for data access:

By granting different types of authorization, the database administrator can regulate which parts of the database various users can access.

* Routine Maintenance

The routine maintenance activities of DBA are:

* periodically backing up the database, either on to tapes or onto remote servers, to prevent loss of data in case of disasters such as flooding.

* 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 Architecture

A database system is partitioned into modules that deal with each of the responsibilities of the overall system.

The functional components of a database system can be divided into

1. Storage Manager components.
2. Query processor components.

The Storage Manager components include:

1. Authorization and integrity Manager
2. Transaction Manager
3. File Manager
4. Buffer Manager.

The query processor components include:

1. DDL interpreter
2. DML compiler
3. Query evaluation engine.

The data structures implemented are:

1. Data files
2. Data dictionary
3. Indices.

Storage Manager :-

It is a program module that provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system.

* Authorization and integrity Manager :-

It tests for the satisfaction of integrity constraints and checks the authority of users to access data.

* Transaction Manager

It ensures that the database remains in a consistent state despite system failures and that concurrent transactions executions proceed without conflicting.

* File Manager

It manages the allocation of space on disk storage and the data structures used to represent information stored on disk.

* Buffer Manager

It is responsible for fetching data from disk storage into main memory and deciding what data to cache in main memory.

Query processor :-

It is responsible for query processing and optimization.

* DDL Interpreter :-

It interprets DDL statements and records the definitions in the data dictionary.

* DML compiler :-

It translates DML statements in a query language into an evaluation plan that the query evaluation engine understands.

* Query evaluation engine :-

It executes low-level instructions generated by the DML compilor.

Data structures

* Data files

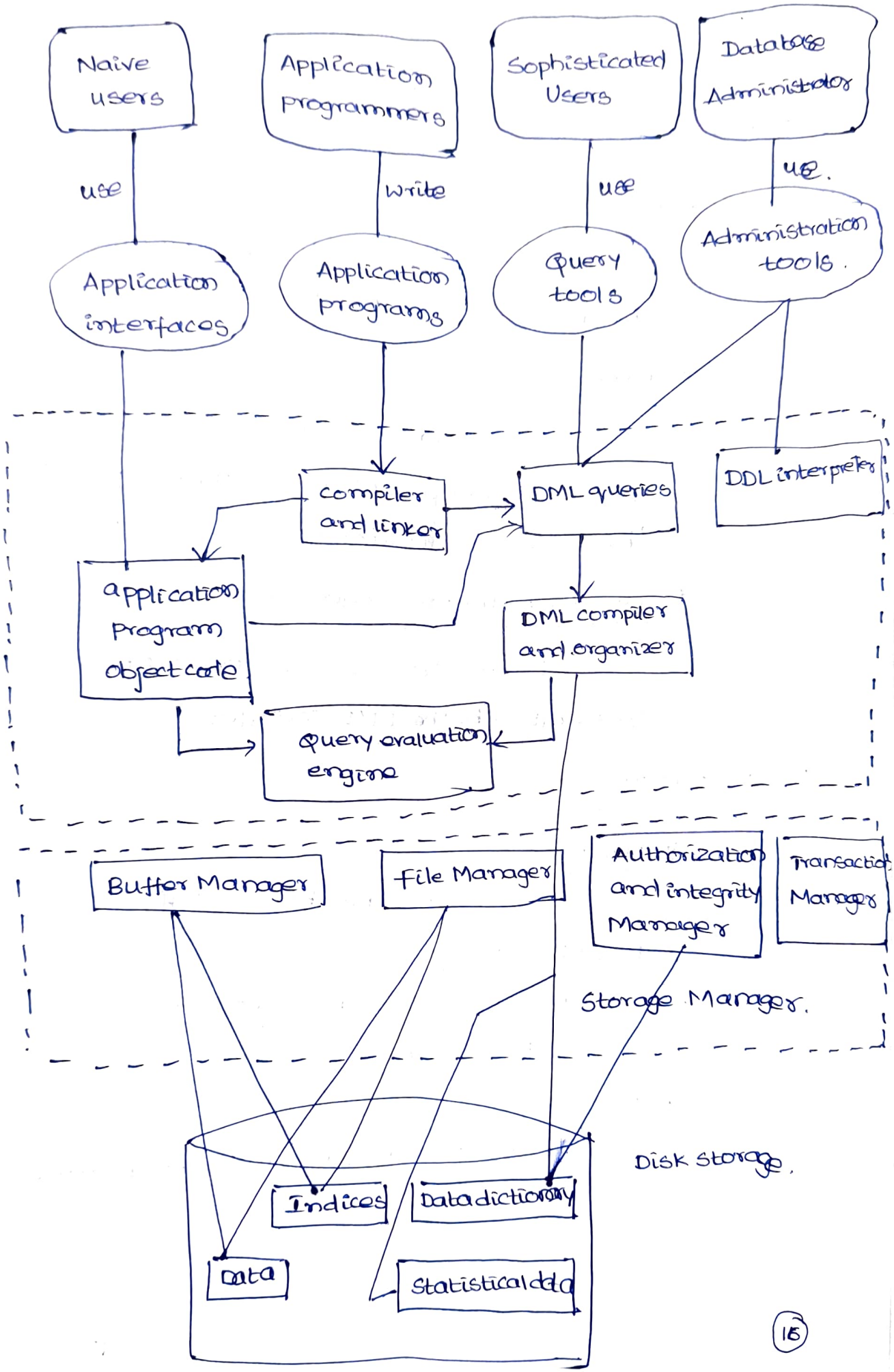
It store the database itself.

* Data Dictionary

It stores metadata about the structure of the database.

* Indices

It can provide fast access to data items.


Naive users
Application programmers
Sophisticated Users
Database Administrator
use
write
use
use.
Application interfaces
Application programs
Query tools
Administration tools.
compiler and linker
DML queries
DDL interpreter
application Program object code
DML compiler and organizer
Query evaluation engine
Buffer Manager
File Manager
Authorization and integrity Manager
Transaction Manager
Storage Manager.
Indices
Data dictionary
Data
Statistical data
Disk storage.

Relational Algebra

Relational algebra is a procedural query language. The different operations involved in relational algebra are:

1. Selection (σ)
2. projection (Π)
3. Rename (ρ)
4. Union ($\cup$)
5. Intersection ($\cap$)
6. Difference ($-$)
7. cartesian product ($\times$)
8. Natural Join ($\bowtie$)
9. Left Outer Join ($\bowtie$)
10. Right outer Join ($\bowtie$)
11. Full outer join ($\bowtie$)

1. Selection (σ)

It is used for selecting a subset of the tuples according to a given selection condition.

2. Projection (Π)

It is used to display specific attributes from a given relation.

Selection operator

1. Select records from employee table where employee salary is greater than 500

$$\sigma_{empsal > 5000}(employee)$$

2. Select records from employee table where employee salary is greater than 5000 and employee designation is team member.

$$\sigma_{empsal > 5000 \wedge empdesig = 'team\ member'}(employee)$$

Projection operator

Display employee numbers from employee table.

$$\Pi_{empno}(employee)$$

Display employee numbers, employee names from employee table.

$$\Pi_{empno, empname}(employee)$$

Display employee names whose salary is greater than 5000

$$\Pi_{empname}\ \sigma_{empsal > 5000}(employee)$$

Rename (ρ) :-

It is used for renaming attributes of a relation.

Union (∪) :-

It is used to find the union of two relations.

Intersection (∩) :-

It is used to find the intersection of two relations

Difference (-) :-

It is used to find the difference of two relations

cartesian product (×) :-

It is used to find the cartesian product of two relations

Natural Join (⋈) :-

It is used to form the natural join of two relations.

Left outer Join (⟕) :-

It is used to find the left outer join of two relations.

Right outer Join (⟖) :-

It is used to find the right outer join of two relations.

Full outer Join (⟗) :-

It is used to find the full outer join of two relations.

Relational Calculus

It is a Non-procedural query language that tells the System what data to be retrieved but does not specify how to retrieve it.

Relational calculus exists in two forms:

1. Tuple Relational calculus
2. Domain Relational calculus.

Tuple Relational Calculus

A tuple relational calculus is a non-procedural query language that specifies to select the tuples in a relation.

A query in tuple relational calculus is of the form

$$\{ t \mid P(t) \}$$

* It is the set of all tuples t such that predicate P is true for t.
* t is a tuple variable
* t[A] denotes the value of tuple t on attribute A
* $t \in r$ denotes that tuple t is in relation r
* P is a formula similar to that of the predicate calculus.

Domain Relational calculus

A domain relational calculus uses the list of attributes to be selected from the relation based on the condition.

A query in domain relational calculus is of the form

$$\{ \langle x_1, x_2 \dots x_n \rangle \mid P(x_1, x_2 \dots x_n) \}$$

Where

$x_1, x_2 \dots x_n$ represent domain variables

P represents a formula similar to that of the predicate calculus.

Difference between Tuple Relational Calculus and Domain Relational Calculus.

Tuple Relational Calculus (TRC)	Domain Relational Calculus (DRC)
1. It is used to select tuples from a relation	1. It is used to select attributes from which to choose based on the condition.
2. In this, the variables represent the tuples from a relation	2. In this, the variables represent the values drawn from a specified domain.

3. A tuple represents a single row in a relation	3. A domain refers to the set of values taken by an attribute.
4. The query cannot be expressed using membership condition	4. The query can be expressed using membership condition.
5. QUEL is an example of Tuple relational calculus	5. QBE is an example of Domain Relational calculus.
6. The syntax of query is $\{ t \mid P(t) \}$	6. The syntax of query is $\{ \langle x_1, x_2 \ldots x_n \rangle \mid P(x_1, x_2 \ldots x_n) \}$
7. Tuple relational calculus is used to test every row using a tuple variable and return those tuples that met the condition.	7. Domain relational calculus makes use of domain variables and based on the condition, it returns the required attribute.

(3) Consider the following schema.

branch (branch_name, branch_city, assets)

customer (customer_name, customer_street, customer_city)

loan (loan_number, branch_name, amount)

borrower (customer_name, loan_number)

account (account_number, branch_name, balance)

depositor (customer_name, account_number)

Write the following queries in tuple relational calculus.

(1) Find the branch_name, loan_number and amount for loans of over $1200;

$$\{ t \mid t \in loan \wedge t[amount] > 1200 \}$$

(2) Find the names of all customers who have a loan from the perryridge branch.

$$\{ t \mid \exists s \in borrower (t[customer\_name] = s[customer\_name] \wedge \exists u \in loan (u[loan\_number] = s[loan\_number] \wedge u[branch\_name] = \text{"perryridge"})) \}$$

3. Find the names of customers who have a loan, an account or both at the bank.

$$\{ t \mid \exists s \in borrower (t[customer\_name] = s[customer\_name] \vee \exists u \in depositor (t[customer\_name] = u[customer\_name]) \}$$

4. Find the customers who have account at the bank but do not have a loan from the bank.

$$\{ t \mid \exists u \in depositor (t[customer\_name] = u[customer\_name] \wedge \neg \exists s \in borrower (t[customer\_name] = s[customer\_name]) \}$$

5. Find all customers who have an account at all branches located in Brooklyn.

$$\{ t \mid \exists r \in customer (r[customer\_name] = t[customer\_name] \wedge (\forall u \in branch (u[branch\_city] = \text{"Brooklyn"} \Rightarrow \exists s \in depositor (t[customer\_name] = s[customer\_name] \wedge \exists w \in account (w[account\_number] = s[account\_number] \wedge w[branch\_name] = u[branch\_name])))) \}$$

Domain Relational Calculus

1. Find the loan number, branch name and amount for loans of over $1200

$$\{ \langle l, b, a \rangle \mid \langle l, b, a \rangle \in loan \wedge a > 1200 \}$$

2. Find all loan numbers for loans with an amount greater than $1200.

$$\{ \langle l \rangle \mid \exists b, a (\langle l, b, a \rangle \in loan \wedge a > 1200) \}$$

3. Find the names of all customers who have a loan from the perryridge branch.

$$\{ \langle c, a \rangle \mid \exists l (\langle c, l \rangle \in borrower \wedge \exists b (\langle l, b, a \rangle \in loan \wedge b = \text{"perryridge"})) \}$$

4. Find the names of all customers who have a loan, an account or both at the perryridge branch.

$$\{ \langle cl \mid \exists l (\langle c, l \rangle \in borrower \wedge \exists b, a (\langle l, b, a \rangle \in loan \wedge b = \text{"perryridge"})) \vee \exists a (\langle c, a \rangle \in depositor \wedge \exists b, n (\langle a, b, n \rangle \in account \wedge b = \text{"perryridge"})) \}$$

5. Find the names of all customers who have an account at all the branches located in brooklyn

$$\{ \langle c \rangle \mid \exists s,t (\langle c,s,t \rangle \in \text{customer}) \wedge$$

$$\forall x,y,z (\langle x,y,z \rangle \in \text{branch} \wedge y = \text{"Brooklyn"}$$

$$\Rightarrow \exists a,b (\langle a,x,b \rangle \in \text{account} \wedge \langle c,a \rangle \in \text{depositor})) \}.$$

1. Define Data Mining. Explain about the functionalities of Data Mining.

Data Mining

Data Mining refers to the extraction of nontrivial implicit, previously unknown and potentially useful information from vast amounts of data in terms of management's decisions.

Functionalities of Data Mining

* Class/concept Description
* Association Analysis
* Classification.
* Clustering
* Outlier Analysis.
* Evolution Analysis.

Class/concept Description.

Data can be associated with classes or concepts.

It can be useful to describe individual classes and concepts in summarized, concise and yet precise terms. Such descriptions of a class or concept are called class/concept description.

* Data characterization.
* Data Discrimination.

Data characterization.

It is a summarization of the general characteristics or features of a target class of data.

Data Discrimination.

It is a comparison of the general features of target class data objects with the general features of objects from one or a set of contrasting classes.

Association Analysis

- Association Analysis is the discovery of association rules in a given set of data.

Association rules are of two types.

1. Single Dimensional Association rule.
2. Multidimensional Association rule

Single Dimensional Association Rule.

$$contains(T, \text{"computer"}) \Rightarrow contains(T, \text{"software"})$$

Multi Dimensional Association rule.

$$age(X, \text{"20...29"}) \wedge income(X, \text{"20K...29K"}) \Rightarrow buys(X, \text{"DVD-player"})$$

Classification

Classification is the process of finding a set of models that describe and distinguish data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose classlabel is unknown.

Clustering

Clustering is the process of grouping objects based on similarity

Outlier Analysis

The process of finding outliers is known as Outlier Analysis.

An outlier is an object whose behavior is different from the behavior of other objects in a group.

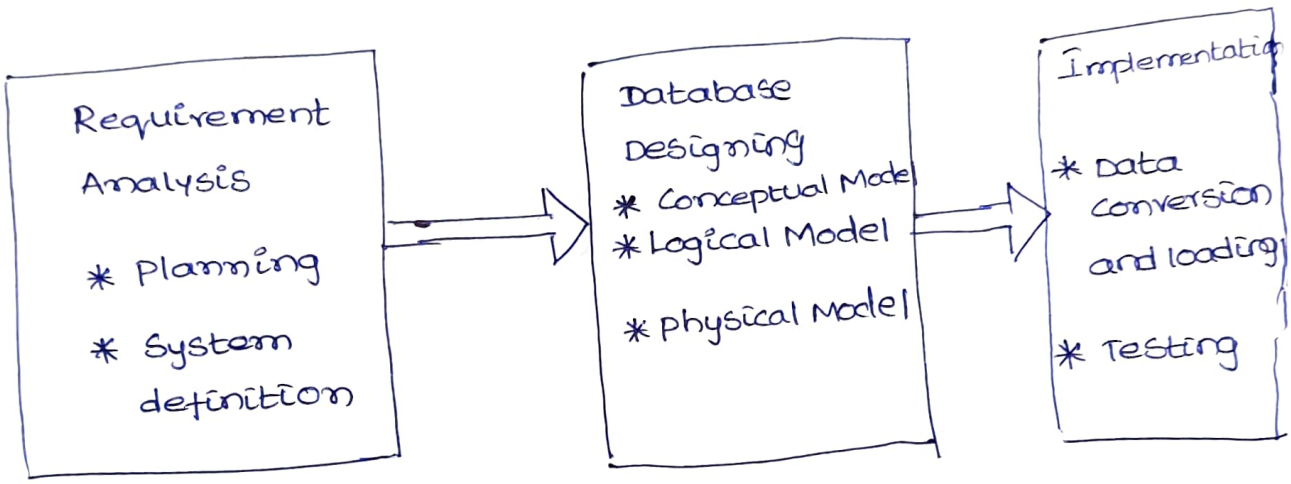
Evolution Analysis

Evolution analysis describes and models regularities or trends for objects whose behavior changes over time.

Database Design

Database Design is a collection of processes that facilitate the designing, development, implementation and maintenance of enterprise data management systems.

Database Development Life Cycle



Requirement Analysis

* Planning :- This stage is concerned with planning of entire Database Development Life Cycle.

* System definition :- This stage defines the scope and boundaries of the proposed database system.

Database Designing

* Logical Model :- This stage is concerned with developing a database model based on requirements.

* Conceptual Model :- It describes the database at very high level and is useful to understand the needs or requirements of the database.

physical model :- This Stage implements the logical model of the database taking into account the DBMS and physical implementation factors.

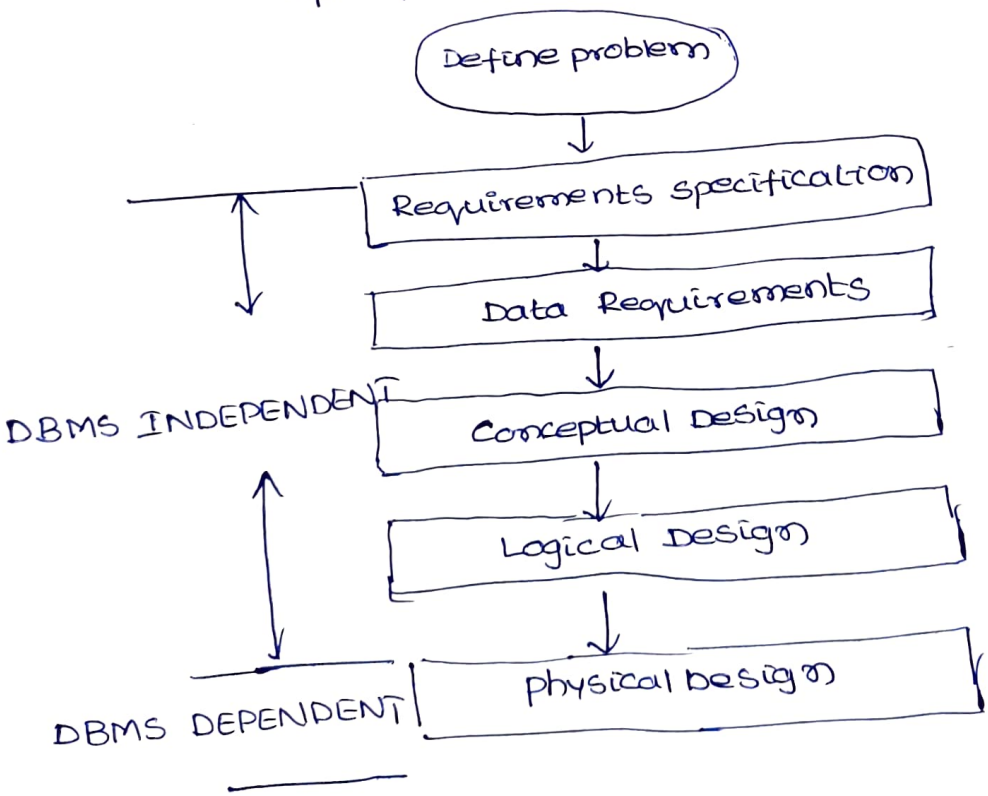
Implementation

* Data conversion and loading :-

It is concerned with importing and converting data from the old system into the new database.

* Testing :-

This stage is concerned with the identification of errors in the newly implemented system. It checks the database against requirement specifications.



UNIT-I (Questions from previous papers)

1. Describe about Database Architecture.
2. List various users of DBMS and specify the roles of each
3. Explain the difference between two-tier and three-tier architectures. Which is better suited for web applications? Why?
4. Discuss the relative merits of procedural and nonprocedural languages.
5. What are the responsibilities of a DBA? If we assume that the DBA is never interested in running his or her own queries, does the DBA still need to understand query optimization? Why?
6. Explain about Database Languages.
7. What is a Data Model? Discuss about various Data Models.
8. What are the advantages of DBMS over file systems?
9. Explain about Data Abstraction.
10. Explain about integrity constraints over relations with examples.
11. Discuss about various set operations in relational algebra with examples.
12. Explain about Tuple Relational Calculus and Domain Relational calculus.
13. Write the following queries in relational algebra using university schema

Data Model :-

A data model is used to represent the logical structure of a database.

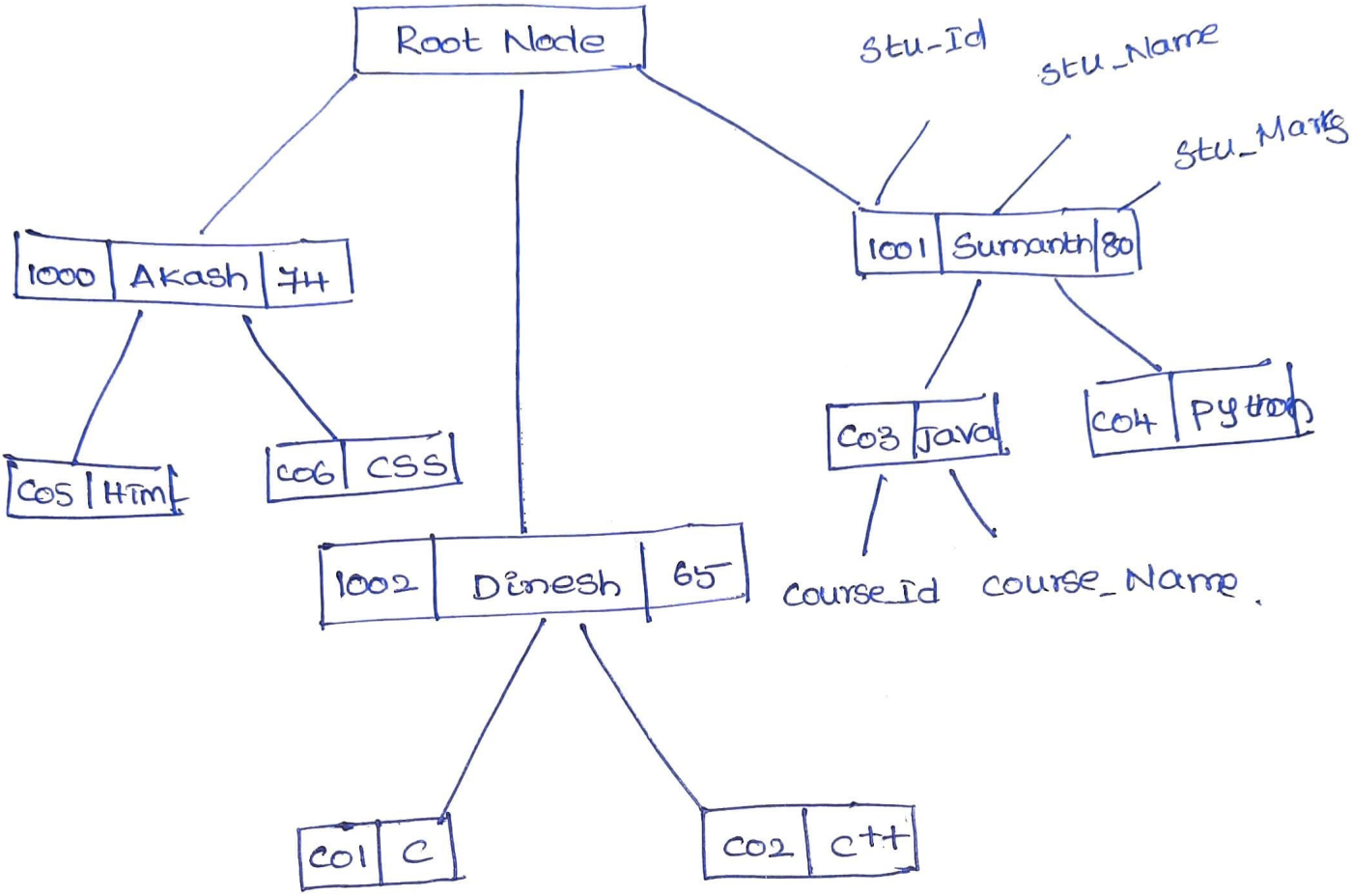
A data model is used to show how data is stored, connected, accessed and updated in the database management system.

Data models are of different types - they are

1. Hierarchical Model
2. Network Model
3. Entity-Relationship Model
4. Relational Model
5. Object-oriented Model.
6. Object-Relational Data Model
7. Semi-structured Data Model
8. Flat Data Model
9. Associative Data Model
10. Context Data Model.

Hierarchical Data Model :-

It is a data model in which the data is organized in tree-like structure.

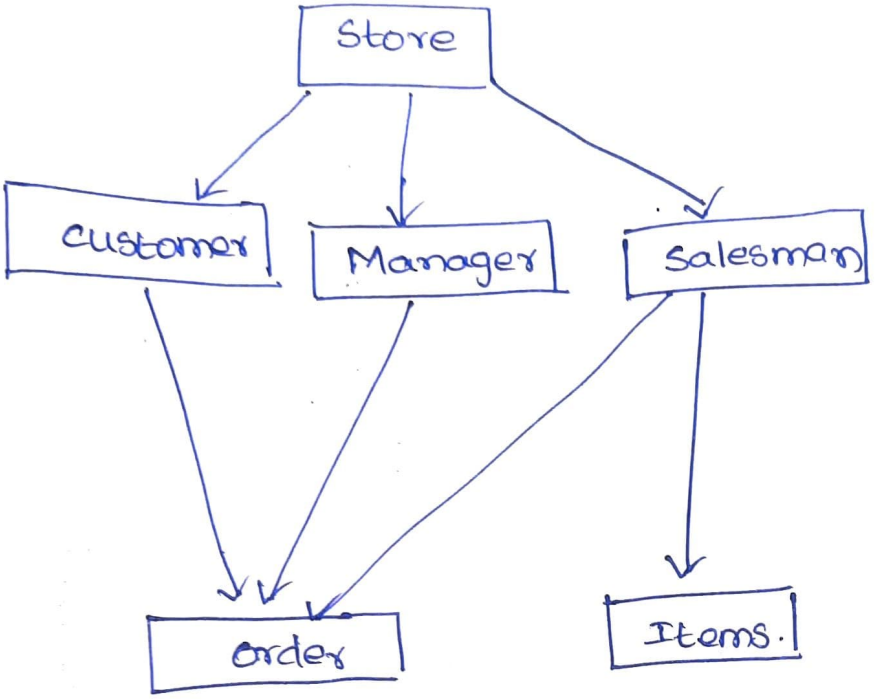


Each record has one parent node but many child nodes. The records are connected through links.

Network Model

It is a data model in which the data is organized in the form of a graph.

Each record is represented as a node and each node can have multiple parent nodes.



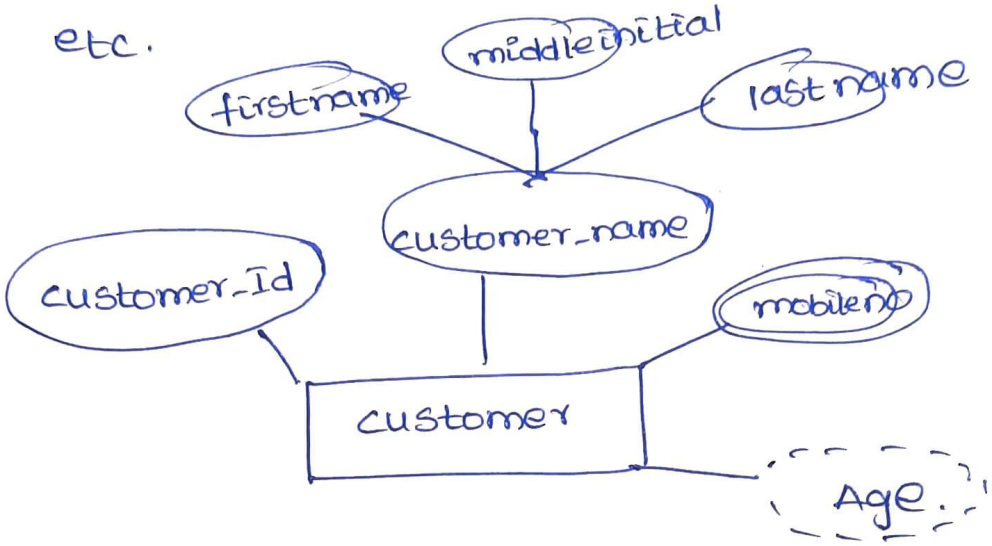
This model is used to represent complex data relationships more effectively.

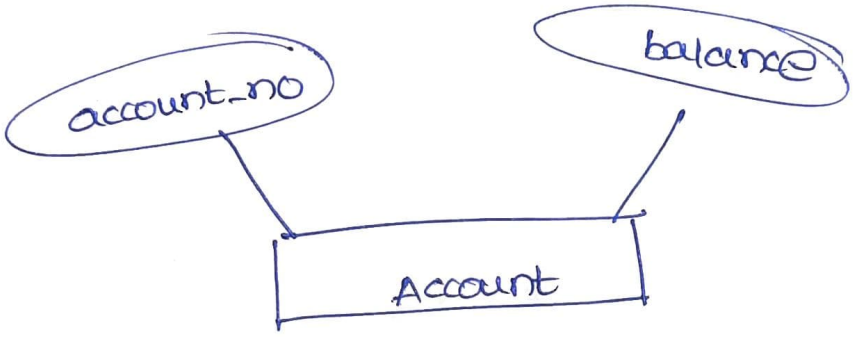
E-R model

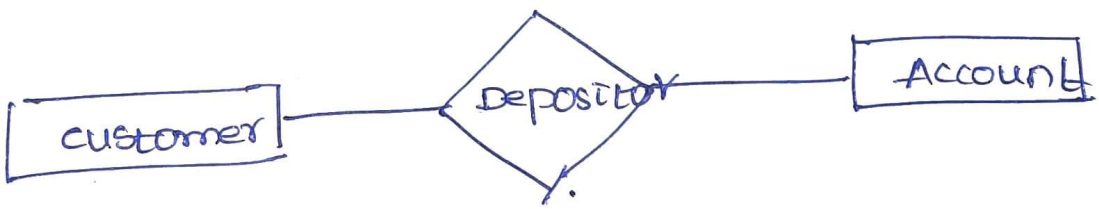
E-R stands Entity-Relationship.

This model describes the structure of the database with an E-R diagram.

An E-R diagram represents the entity sets, relationship between entity sets, attributes of each entity set etc.







Relational Model :-

In this model, the data is represented in the form of tables and the relationship exists between tables in the form of primary key and foreign key.

Student

Primary key

Student_Id	Student_Name	Student_Marks
1000	Akash	74
1001	Sumanth	80
1002	Dinesh	65

course_Id	course_Name	Student_Id
c01	C	1002
c02	c++	1002
c03	Java	1001
c04	python	1001
c05	HTML	1000
c06	CSS	1000

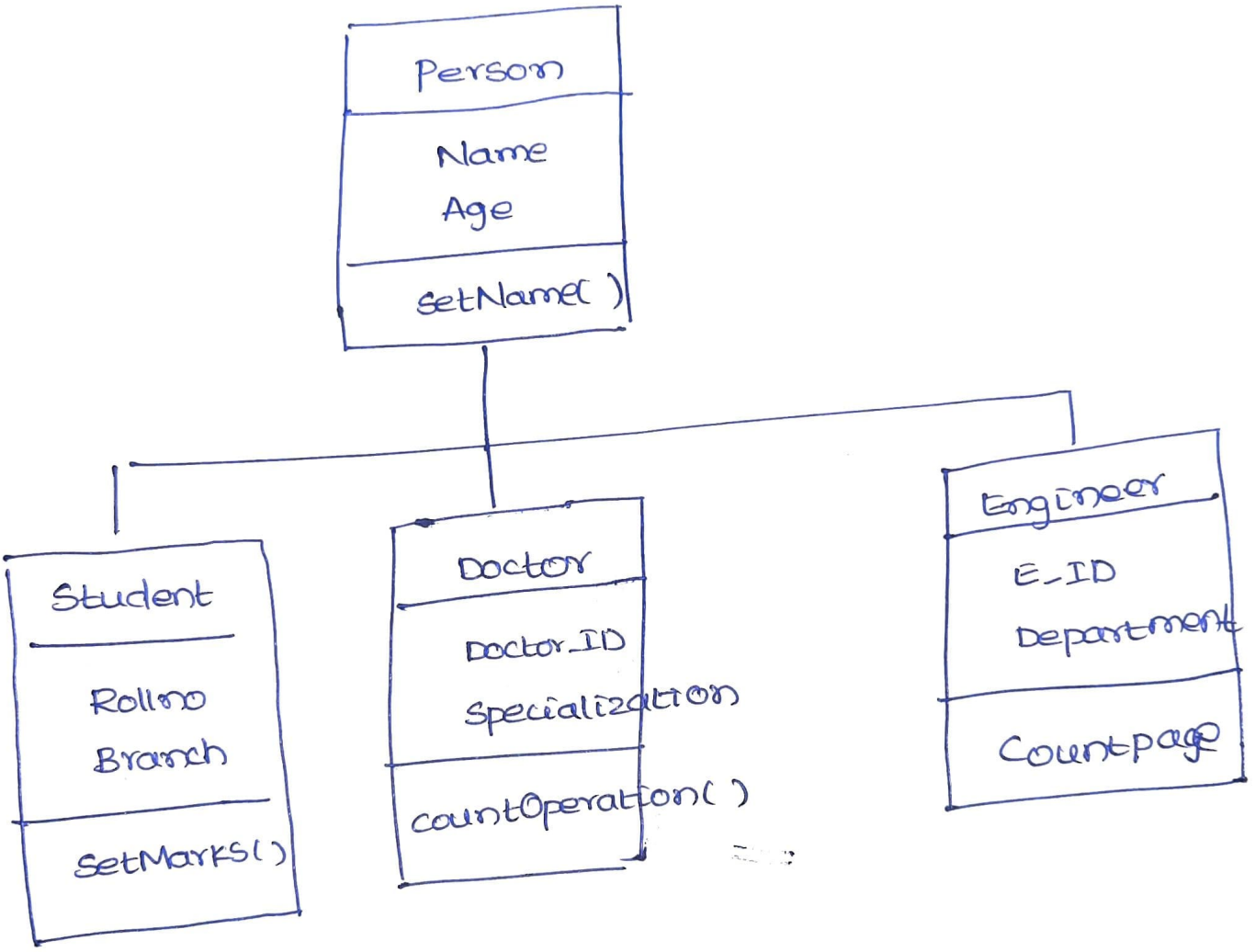
Object-oriented Model :-

In this model, the real-world entities are represented as objects.

object oriented Data Model = object oriented Programming

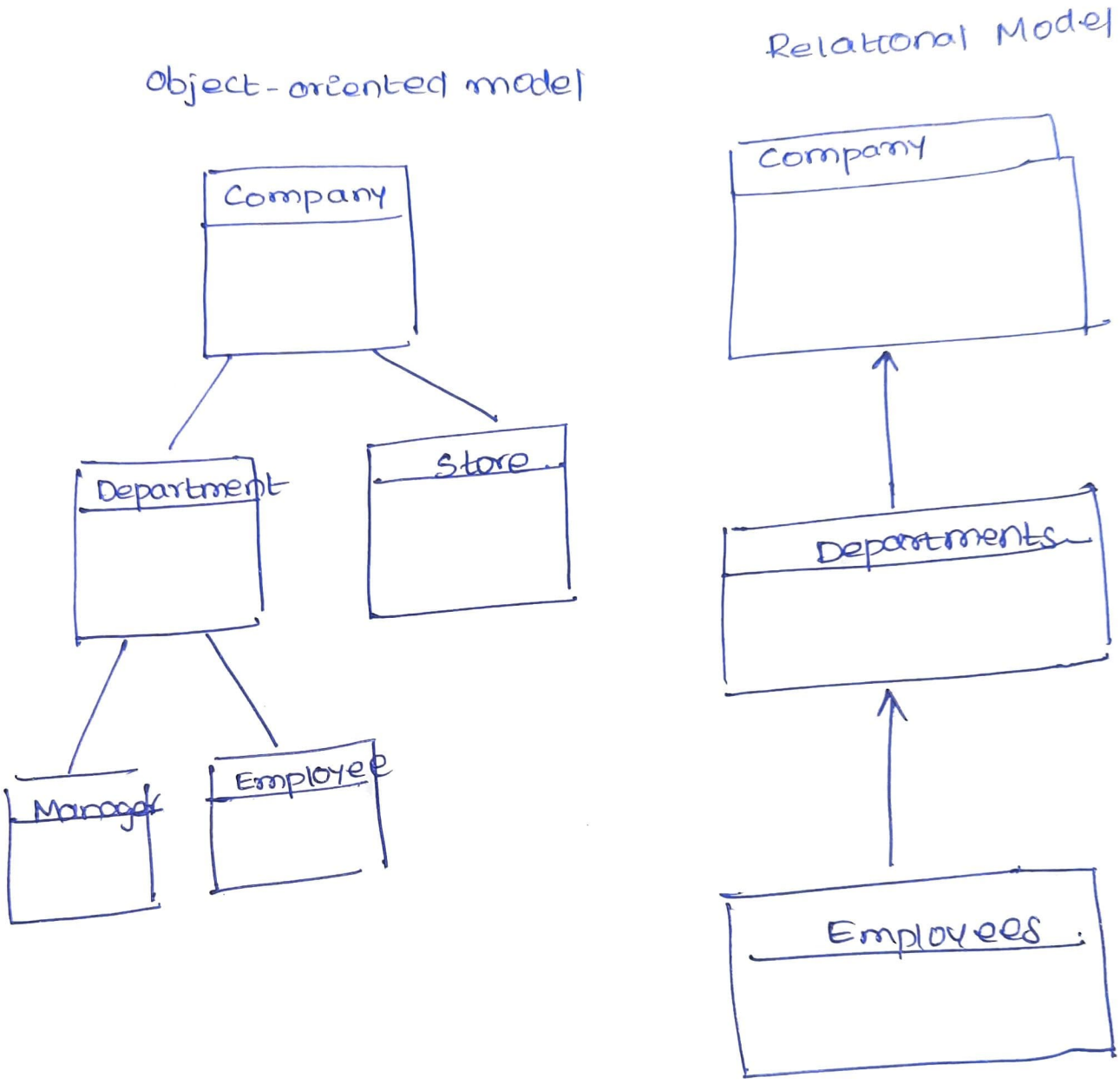
+

Relational database Model.



Object Relational Model

It is a combination of object oriented database model and Relational database model.



Semistructured model

In this model, there is no separation between the data and the schema.

Semistructured data refers to data that is not captured in conventional way.

Examples of Semi-structured data

* E-mails
* XML
* Binary executables
* Zipped files.

The model for expressing Semi-structure data is Object Exchange Model (OEM) and the other is XML.

E.g.
```
<Letter>
    <From> Akash </From>
    <To> Sumanth </To>
    <Heading> Reminder </Heading>
    <Body> Don't forget me this Weekend
    </Body>
</Letter>
```

Flat Data Model

The entire database is represented as a table consisting of rows and columns.

Associative Data Model :-

In this model, the data is represented as items and links.

Context Data Model :-

It is a combination of two or more models.