Chapter 1 : Introduction

1.1

Concepts and Applications:

Data:

Data is the raw fact and figures.

Information:

Information is the processed data.

Entity:

Real world object that are represented in database.

Attribute:

Properties used to describe entity.

Attributes (characteristics of card)

Name on card	Type	Number	ExpMonth	ExpYear
Ram	VISA	4675458822...	05	2019
Hasi	VISA	....	11	2020
Shyam	MC	....	04	2018

Entities (Individual credit cards)

Field:

A character or group of characters having specific meaning and used to define and store data.

Record:

A logically connected set of one or more fields that describes a person, place or thing.

File:

A collection of related records

For example:

A telephone book is analogous to file. It contains a list of records, each of which consists of three fields: name, address & telephone numbers.

Database Management system:

Database:

- A collection of related data
- can be of any size and varying complexity
- can be generated and maintained manually or can be computerized.

Database Management system:

- Collection of program that enables users to create, organize and manage the database.
- Provides an environment that is convenient and efficient to use.
- contains the information about the particular enterprise.

Database System Applications:

i. Entesprise Information.

Sales: customers, products, purchases

Accounting: payments, receipts, assets.

Human Resources: Information about employees, salasies, payroll taxes.

ii. Manufactusing.

Management of production, inventory orders, supply chain.

iii. Banking and finance.

↳ customer information, accounts, loans, and banking transactions.

↳ Credit casd transactions

↳ Finance: sales and purchases of financial instruments (e.g., stocks and bonds; storing real-time masked data)

iv. Universities:

Registration, grades

v. Airlines.

Resesrations, schedules.

vi. Telecommunication:

Records of calls, texts, and data usage, genesating monthly bills, maintaining balances on prepaid calling cards.

vii. web-based sesvices.

online retailers: orders tracking, customized recommendations.

online advestisements.

viii. Document databases

ix. Navigation systems.

For maintaining the locations of vasious places of intesesting along with the enact routes of roads, train systems, buses etc.

1.2 Objective and Evolution.

In early days, database applications wese built disectly on top of file systems. which leads to:

i. Data reduclancy and inconsistency

Data is stored in multipk files formats resulting in duplication of information in diffesent files.

ii. Difficulty in accessing data.

Need to write a new program to carry out each new task.

iii. Data Isolation

Multiple files and formats

iv. 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

v. 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.

vi. Concurrent access by multiple users.

↳ Concurrent access needed for performance

↳ Uncontrolled concurrent accesses can lead to inconsistencies

Example: Two people reading a balance (say 100) and updating it by withdrawing money (say 50)

each) at the same time

vii. Security problems:

Hard to provide uses access to some, but not all data

Database systems offes solutions to all the above problems.

~~Evolution of Database. (Assignment)~~

Components of Database System.

Data:

↳ Stored information in DBMS

↳ It can be integrated & shared.

i. Integrated data refers to the visualization of database unification of several files with no redundancies.

ii. Shared data can be used by multi-uses simultaneously for different purpose.

Software:

DBMS that act as medium to communicate database with users and application programs.

Hardware:

↳ Hardware components of system consists of secondary storage volumes.

↳ Hasdware processor and associated main memory to support the execution of database system softwase.

Users:

People who use or access data.

Type of User:

Application Programmess.

AP implement the specification genesated by system analyst in some high level programming language which access the database by issuing appropriate request to the DBMS.

Database Administrator.

↳ Central controller oves the system.

↳ Responsible for authorizing access to the database, for co-ordinating and monitoring its use, acquiring softwase, and hasdware resources, controlling its use and monitoring efficiency of opesations.

Functions:

schema definition, storage structuse and access method definition, schema & physical organization modification, authorization of data access, routine maintainenance.

End User.

Naive or Parametric end user.

↳ Naive users are typically unaware of the DBMS.
↳ They access the database through ~~typically~~ specially written application programs.
↳ Their main job function revolves around constantly quesying and updating the db. They use form based interface where uses can fill in appropriate fields of forms.

Sophisticated end users.
↳ Intesact with system without writing programs and may use high level query such as SQL to perform any requised opesations.
↳ Engineers, scientist, business analyst & others who throughly familiarize themselves with the facilities of DBMS so as to implement theis application to meet theis complex requisement.

1.3

Data Abstraction and Data Independence:

Data Abstraction.

↳ A major puspose of a database system is to provide users with an abstract view of the data. That is, the system hides certain details of how data are stored and maintained.

↳ Since many database-system users asenot computes trained, developers hide the complexity from users through seveal levels of abstraction, to simplify user's interactions with the system.

Physical level:

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

↳ The physical level descsibes complex low-level data structures in detail.

↳ Descsibes the organization of files, access path etc.

Logical level.

↳ logical level descsibes what data are stored in the database, and what relationship exist among those 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 employee's salary) for security purposes.

View of data:

An architecture for a database system:

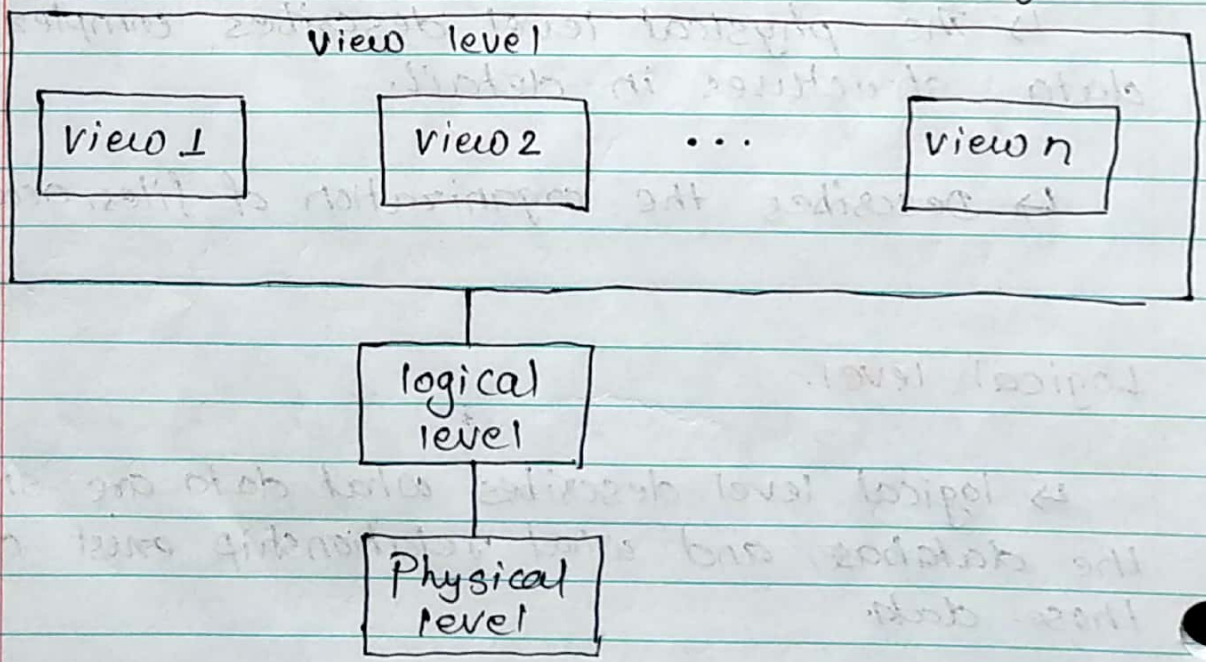

Figure: The three levels of data abstraction.

1.4

Instances and schemas

Instance:

The collection of information stored in the database at a particular moment is called an instance of the database. (similar to variable declaration.) → schema

values of variables in a program at a point in time → instance

Schema.

The overall design of the database is called the database schema. Schemas are changed frequently.

Logical schema.

Logical schema describes the ~~logical~~ database design at the logical level.

Example: The database consists of information about a set of customers and accounts in a bank & the relationship between them.

Analogous to type information of a variable

Physical schema.

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

Physical Data Independence.

The ability to modify the physical schema without changing logical schema.

↳ Applications depend on the logical schema

↳ In general, the interface between the various

levels and components should be well defined so that changes in some parts of do not seriously influence others. Logical पहुचाई दु।

1.5 Concepts of DDL, DML and DCL

Data Definition Language.

↳ DDL deals with the description of the database schema where it creates and modifies the structure of the database objects on the database.

↳ ~~DDL~~

Example: create table instructor (
ID char(5),
name varchar(20),
dept-name vaschas(20),
salary numesic(8,2))

↳ DDL compiler generates a set of table templates stored in a data dictionary.

↳ Data dictionary contains metadata (i.e. data about data)

i. Database schema

ii. Integrity constraints
primary key ID (ID uniquely identifies instructors)

iii. Authorization
Who can access what.

[Create, Altes, Truncate
comment & Rename]

Data Manipulation Language (DML)

↳ A data-manipulation language (DML) is a language that enables users to access or manipulate data as organized by the appropriate data model.
↳ DML is also known as query language.
The types of access are:

i. Retrieval of information stored in the database
ii. Insestion of new information into the database.
iii. Deletion of information from the database.
iv. Modification of information stored in the database

These are basically two types of data-manipulation language.

Procedusal DMLs

Requise a uses to specify what data are needed and how to get those data.

Declasative DMLs.

Requise a user to specify what data are needed without specifying how to get those data.

↳ Declasative DMLs are usually easier to learn and use than procedusal DMLs.

```
Select instructor.ID, department.dept-name
from instructor, department
where instructor.dept-name = department.dept-name and
  department.budget > 95000;
```

↳ Declarative DMLs are also refered to as non-procedural DMLs.

↳ The portion of DML that involves information retrieval is called query language.

Example: Select, Insert, Delete, Update.

DCL (Data Control Language)

DCL deals with permissions, rights and other controls of the database.

Example: Revoke, Grant

Logical Data independence.

↳ It is the capacity to change the conceptual schema without having to change external schemas or application programs.

↳ We may change the conceptual schema to expand the database (by adding a record type or data item), or to reduce the database (by removing a record type or data item).

Assignments:

Why data independence is important in data modeling? Differentiate between physical & logical data independence.