Ch I Introduction

(1.1) Concept and applications (1.2) objectives and Evolution (1.3) Needs of DBMS

↳ Data can be defined as the collection of raw facts. It is the collection of statistics, information, facts that are often in raw or unprocessed form.

↳ Database can be defined as collection of logically interrelated data and description of those data, design to meet the information needed for an organization, which are typically stored on Larger disk and accessible to many users.

↳ Database Management System is a technology (S/W system) for storing, retrieving, editing, manipulating, deleting data with utmost (maximum) efficiency maintaining proper security measures.

In other word, its a collection of programs which allows the users to specify structure of database like: no. of attribute it will have, data types of attribute, etc., to create, query and modify the data in the database and control access to it.

Examples of DBMS software : MYSQL, MSSQL, Oracle, PostgreSQL, MongoDB,

⇒ Database System (DBMS + DB)

↳ The combination of Database and Data base Management System give the Database System, which is set of data along with their description, different process that DBMS follow to query, fetch, store, manipulate data on DBMS at utmost efficiency maintaing proper security measures.

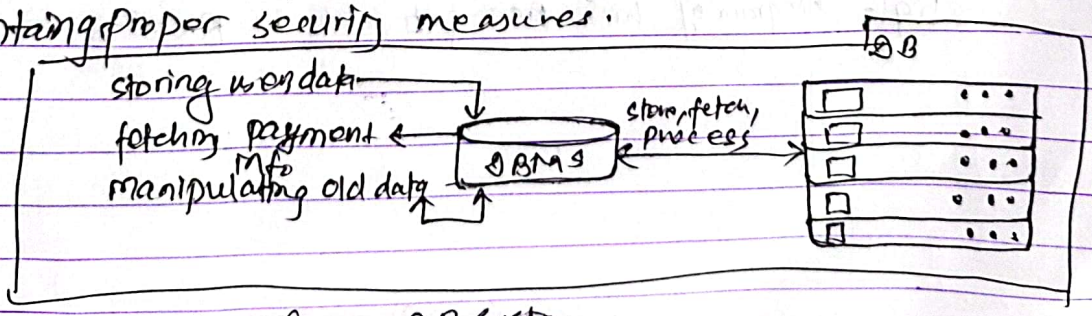


Fig :- DB System

This whole system is known to the DB system, which is the high level representation of database technology. (2)

Relational database Mgmt System (RDBMS)

1) It is one of the type of DBMS, that organizes and stores the data in the structured table (relation). It is most popular and widely used DBMS, because of easier understanding and implementation. Examples are: MYSQL, MSSQL, oracle, sqLite, postgreSQL etc. One of the features of RDBMS is that, it imposes integrity constraints, so the mutiple kind of integrity (rules) have to be maintained for successful completion of different task/process.

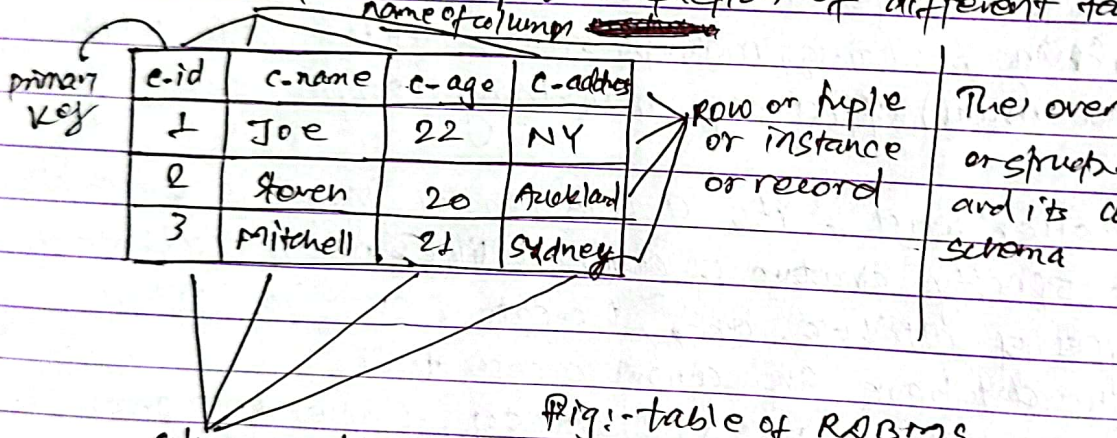


Fig:- table of RDBMS

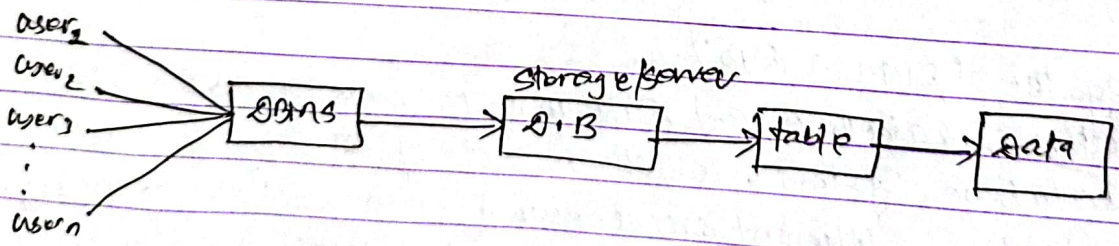


Fig:- Diagram of how user gets data in RDBMS.

Applications of DBMS

(a) Banking
(b) Uni
(c) Hospitals,
(d) Org tech companies
(e) Government system
(f) Research and development

Property of DBMS (सीखेंगे हम)

(a) **Abstraction**: Abstraction is maintained by maintaining 3 level as:

level 1: View label (user label)
↓
level 2: logical level
↓
level 3: physical level

(b) **Persistence of data** ⇒ Data stored in the DBMS must not be volatile, it must be permanently stored.

(c) **Data Independence**

↳ The schema (blueprint/structure) of the database must be independent of data manipulation. Which means the data can be independently manipulated without having effects on Database schema.

(d) **Consistency** ⇒ Database must be consistant such that it can capture each and every change happened and update the info as fast as possible

(e) **Controlled Redundancy**

↳ DBMS must be capable of controlling the redundancy, such that there is no any chance of data repeatation and stores those data which are only in need which leads to make efficient built of DBMS, and retreive data quickly

(f) Security

(g) sharing and access control.

FPS (Early Information System)

↳ FPS which stands for file processing system (aka early information system) is a traditional approach which was used in the early days to store information data in the file of papers.

↳ After computer was introduced, the paper based file system shifted to computer based file system where data were stored in the form of digital text file. eg:- By use of notepad, or text editor, we can create, read, write, open, close a file. Now problem was when we wanna see, search for specific information, we need to use different prog lang like C, C++. It was tougher to learn those language just to work with files and lengthy program were needed to be written for smaller task. It is huge problem before the invention of DBMS.

↳ Below diagram shows the problem of FPS where each application program generates its own data file. due to which the common data files or data will be repeated (redundant) because of non sharing of data.

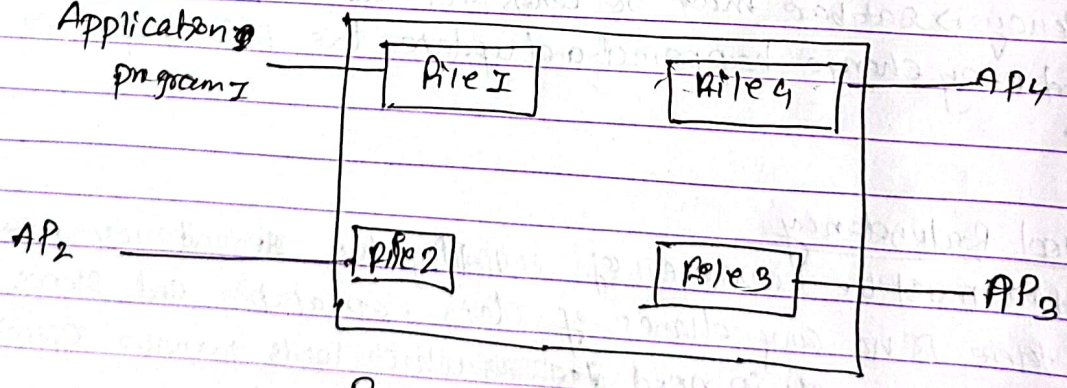


Fig:- Diagramatic representation of FPS

Pros of FPS	Cons
(i) No need of external storage	(i) Strong knowledge of prog lang was needed.
(ii) No need highly skilled person, to handle file	(ii) Hard to update, handle.
	(iii) No sharing of data was done due to which high chance of data redundancy.

(iv) No any access control, due to which not secure.
(v) storage issue may arise.

1
Joe
NY
2
Steven
Auckland
3
Kunindergill
Punjab

SN	Name	city
1	Joe	NY
2	Steven	Auckland
3	Kunindergill	Punjab

Fig:- FPS vs DBMS table

So, this problem need a proper way of managing users data in a structured way which leads to invention of RDBMS.

- Component of DBMS
  - ↳ Hardware
  - ↳ SW
  - ↳ Data
  - ↳ users
  - ↳ Procedure / method.

P.T.O.

Data Abstraction

↳ Here in the D.B, Abstraction is maintained by building three different level:

ⓐ physical level ⓑ Logical level ⓒ view Label.

In the term of D.B, Data Abstraction is the process of hiding the unwanted, complex processing, core working functionality of the system from the end users or outsider such that the user can interact with the system easily and mantain a proper structure of data based on its depth.

It plays a vital role in the security of the D.B, where the outside can't know how the data are processed internally and can't track the data flow in D.B. For the three different level developers uses different data structure based on the functionality they need to have and work they perform.

ⓐ physical level

↳ It is the lowest level of data abstraction in D.B, which describes how data are stored and managed in D.B system. Here the database specialist work on this level controlling how data flows in different path. In this level data storage in Disk, its mantanance, its update are handled, creation etc.

ⓑ Logical level

↳ It describes what data are stored in D.B and contains the logic of how and what happens after firing some query. It describes the relationship that exists between data. Data base administrator works on this level

ⓒ user/view Label

↳ It is the highest level of Data abstraction where the end user interact with system. All the data which are allowed by the system, can be accessed by them.
to outsider

Here it is needed to seperate the core level and logic level ⑦ of system from the user interaction level and mantain a simple user interactive system. Also users only want that much of information which are needed for them, which will be provided by this level.

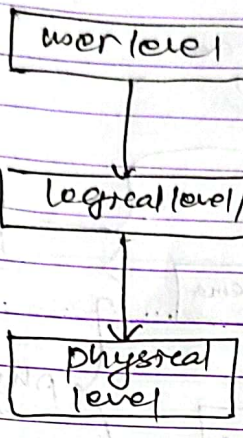


Fig:- Abstraction of data in D.B.

Example:

Update table student
Set marks = 50 where roll num > 50; } → database specialist work (physical work)

if the roll number of student is greater than 50, we will set makks equal to 50, } logical level

S.N	marks	roll num
1	20	3
2	45	89
3	44	55

Before update

↓ After updation

SN	marks	roll num
1	20	3
2	50	51
3	50	55

(what user level sees)

- Data independence
  - (a) physical data independence
  - (b) Logical Data independence

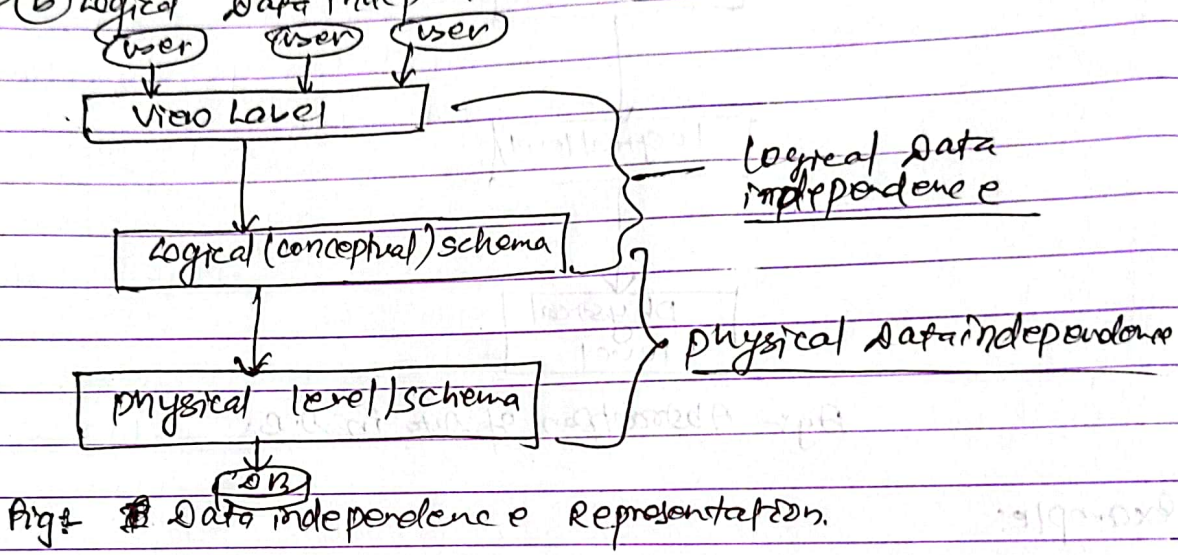


Fig: Data independence Representation.

physical Data independence

↳ The physical schema (level) changes done in the D.B wont affect the logical (concephual) level of the D.B. It can be defined as the ability to change the physical schema with changing the concephual schema in any D.B system is known to be physical Data independence.

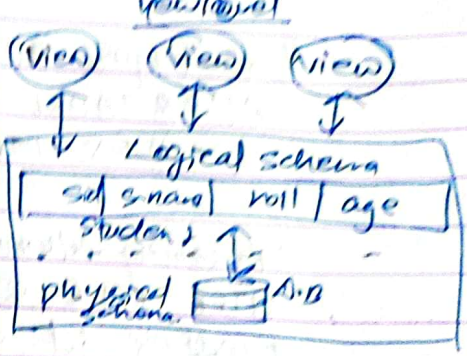
Logical Data independence

↳ Logical Data independence makes sure that the changes done in the concephual level of D.B wont affect the view level of the D.B. It is quite difficult to guarantee that no effect will be seen as retrieving of data mainly related with logical level. Due to this sometimes changes in Application program is needed, if new fields are added or old fields are deleted.

Schema

↳ It is like the blue print of the database that shows how the data are organized, different level of abstraction and the overall high level representation of D.B. systems.

- Database → Physical Schema (same as physical level)
- Database → Logical Schema (logical level same)



Database Instance

↳ The data in the database in the particular moment of time is called as Database Instance. It is also known to be database state or snapshot. It is also called current set of occurences.

⇒ The D.B instance tends to change with time.

Different type of SQL (Query Language)

- (a) DDL
- (b) DML
- (c) DCL
- (d) TCL

DDL
- CREATE: used to create the DB and table
- ALTER: It is used to make change in schema (Add, rename, drop, change DT)
- RENAME: It renames the database instance
- DROP: Deletes the (table) entity (same whole table)
- TRUNCATE: makes the entity (relation) Table empty

Data definition Language (DDL) can be defined as the schema or structure defining language. It consists of several keywords with will the frame (blue print) of database and table is created. It is the one that helps in modifying the structure of DB

eg: CREATE TABLE Student (id INT, age INT, name VARCHAR(40));

// Adding new column
ALTER TABLE Student
ADD MARKS DECIMAL(10,2);

// modifying column
ALTER TABLE Student
MODIFY MARKS INT;

// Renaming column id to serial_num
ALTER TABLE Student
MODIFY id serial_num INT;

// dropping column
ALTER TABLE Student
DROP age;

// Renaming relation
RENAME TABLE Student to Student_table

DROP TABLE Student_table

DML ↳ DML words stands for Data manipulation Language is used to manipulate the created schema where data are stored, fetched, update and deleted. In general here with the help of DML, CRUD operation is peyormed.

→ INSERT ⇒ used to insert data in the D.B table.

```
INSERT INTO Student (id, age, name)
VALUES (3, 22, 'Pritamanand'),
       (4, 22, 'Arneet');
```

→ SELECT,

→ UPDATE :- It is used to update the values (data) in table.
UPDATE student set age = 17 WHERE id = 1;

→ DELETE: used to delete tuple (row, instance, record) from table.
eg:- DELETE from Student ; // All row are deleted
DELETE from student where id = 2 // id 1 रहेगी

DCL (Data Control Language)
↳ It is used to control the access and privileges of the DB system
It used 2 command
→ GRANT: gives the user access privilege to DB.
→ REVOKE : takes back the mitted provided privileges from user.

TCL (Transaction control Lang)
↳ These language are used to validate, annul (रद्द), save the changes permanently or temporarily in DB. TCL deals with the data transaction done in DB
→ COMMIT : Save the changes made permanently
→ ROLLBACK : Annul the changes made and bringing the DB back to orginal state
→ SAVE POINT or SAVE TRANSACTION
↳ To save temporarily or to create a save point within a transaction.

3-tier architecture

- **Three layers:**

  Presentation Tier : *HTML/CSS*
  Application Tier: *PHP/JAVA*
  Database Tier: *MYSQL/SQL*

- **Why do we need it ?**

i. Separation of user applications and physical database which enables data independence.
ii. Proposed to support DBMS characteristics & its advantages
iii. Helps in enabling security features.
iv. Multiple views support of the data

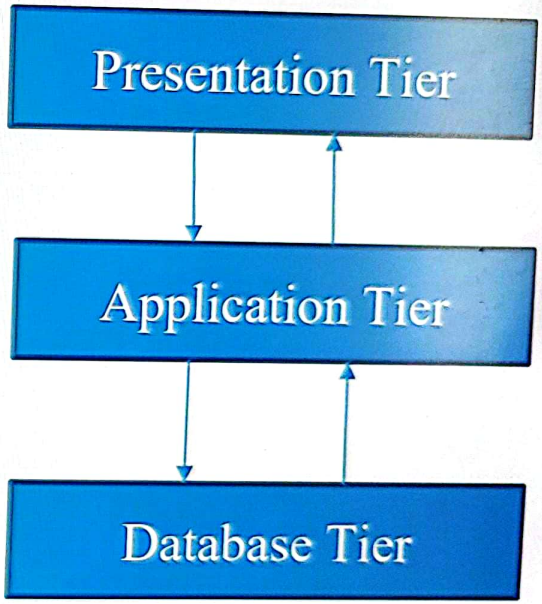


Fig. 3-tier architecture

Activate Win
Go to Settings to a

CamScanner

Data Dictionary

Data dictionary is the metadata about the data.
Data dictionary is a module or package which contains the information about database along with its description, data types used and relationship between data etc.

outlining student_db

Ex: A data dictionary on Student table relation

FieldName	data type	size
name	VARCHAR	20
age	INT	4
rollnum	int	4
address	VARCHAR	3

CamScanner

Management Information System (MIS):

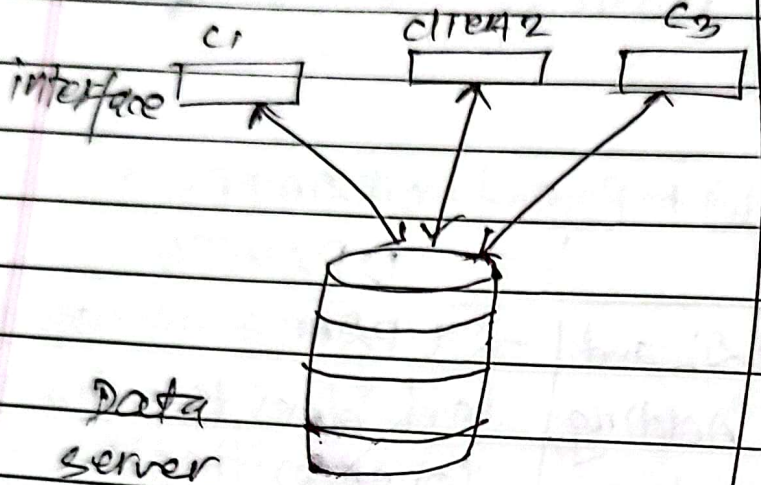
- MIS is a general term for software designed to facilitate the storage, organization and retrieval of information.
- MIS gives the business manager the information that they need to make decision.
- MIS provide a variety of information to managers.
- Periodic Scheduled Report, Exception report, Demand report and response report etc. are the part of MIS in business enterprise.

Activate Windows
Go to Settings to activate Windows.

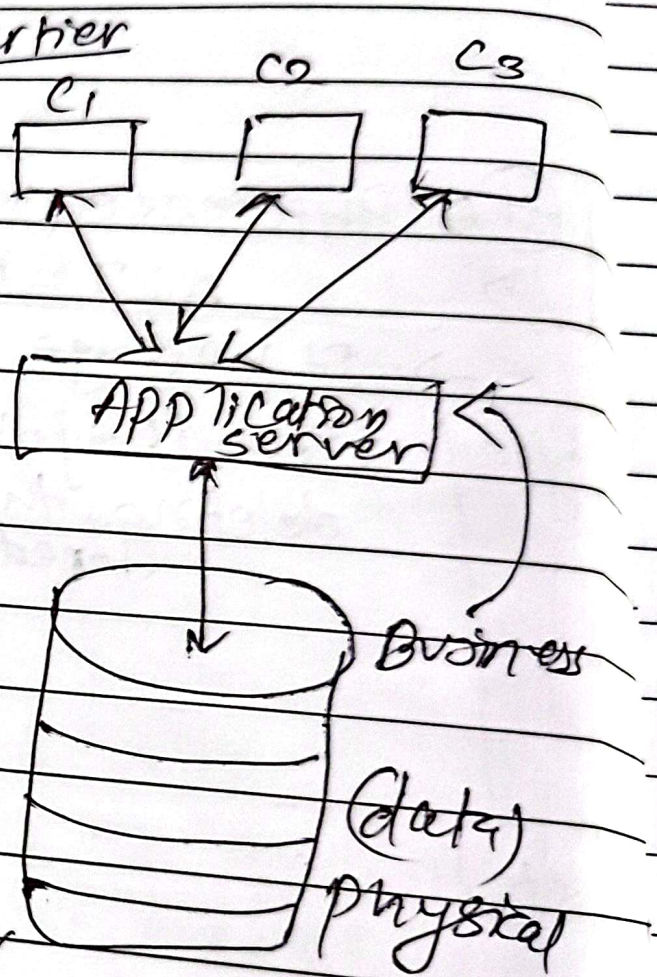
CamScanner

④ Government system.

2 tier Architecture



3 tier Architecture



Assignment :- explains the evalution of DBMS

Differentiate between UPDATE & ~~USER~~

Difference

Q7. explain the evolution of DBMS? DATE

→ DBMS which stands for database management system is a software program that helps to manage the whole data base such that those data and information stored on the server can be easily stored, fetched and used efficiently.

In the ancient times, where there was no computer, data were used to be stored in books and ledger but due to the advancement in tech, the technique of storing data in book transfered into first real D.B. Database is the compilation of data in a structured way. So the evolution of DBMS started in 1960s. Initially data were used to be stored in comp file systems. But storing files in computer file system have certain limitations like redundancy, inconsistency and difficulty in accessing data.

logic

In the late 1960s, first gen of DBMS was developed where hierarchical data model system was introduced and used mainly in large coorporations and gov organization. Here data organization is done in tree like structure where each record used to have one parent and one or more child.

Similarly in late 1970s, 2nd gen DBMS was introduced where network data model was used. It used to have more flexibility than hierarchical data model where data organization is done in graph like structure which each data could have multiple parent and child.

In 1980s to 1990s, 3rd gen of DBMS was developed which used relation data model, which is widely used at present. Relational database stored data in tabular formed manner schema and instance where tables are linked using primary and foreign key.

CLASSMATE PAGE 05

DATE

object oriented DBMS and object relation DBMS was introduced in late 1990s and early 2000s where these model combine the property of RDMS with object oriented programming

similarly modern d.b like NOSQL are used to handle large volume of unstructured or semistructured data. NOSQL used non relational data model like: key-value, document oriented, graph

so it is the evolution of DBMS from 1980 - present but most popular DB are: MYSQL, ORACLE which are RDBMS and NOSQL DB like: MongoDB.

Q2. Explain the difference in TRUNCATE, DROP and DELETE

→ TRUNCATE	DROP	DELETE
① It deletes all the records but not schema.	→ It drops out complete table from the database.	→ It can delete minimum one or more rows from the table.
→ We can't restore all the deleted rows from the database	→ We can't restore complete table which was previously dropped	→ We can restore any deleted one or more row using ROLLBACK command until the COMMIT is done.
→ TRUNCATE commands doesn't free up the space reserved by table from memory.	→ DROP commands free up the space used by table in memory	→ It also doesn't free the space which is reserved or used.

classmate

DATE

performance speed of TRUNCATE is faster than DROP and DELETE as it delete all the data from the table without any condition.	performance speed of DROP is faster than DELETE but slower than TRUNCATE since it deletes all the data at first and delete the table at last.	performance speed is comparitively slower as it deletes one or more rows based on conditions.
Integrity constraints remains same in the TRUNCATE command.	Integrity constraints gets removed from the DROP command.	Integrity constraints remains same by using DELETE Command.
Syntax. TABLE TRUNCATE table_name;	DROP TABLE table_name;	DELETE FROM table_name WHERE condition;

UPDATE VS ALTER

UPDATE	ALTER
→ It belongs to Data Manipulation Language	→ ALTER belongs to Data Definition Language
→ UPDATE edits the existing data that means it overwrites the existing data	→ ALTER adds/delete the schema
→ changes are made to ~~data~~ table's ~~Structure~~ data level	→ changes are made to data table's structure level.
→ Syntax UPDATE table_name WHERE movie_name = '_ _'	→ ALTER TABLE ~~Movies~~ table_name ~~ADD CONSTRAINT~~ ADD COLOUMN movie_name int(5);

CLASSMATE

CH2 Data Models (4 hrs)

Data model can be defined as model that defines how logical structure of database is modeled. They are the conceptual tools for D.B design. Data Models are the set of concepts to describe the structure of D.B and constraints that D.B should obey. It shows the relation of how data are connected with each other and how they are processed and stored in system.

There are 3 stages of data model

1. Conceptual Data model
2. Logical Data model
3. Physical Data model

There are different stage which must be processed to design a data model.

① Conceptual Model

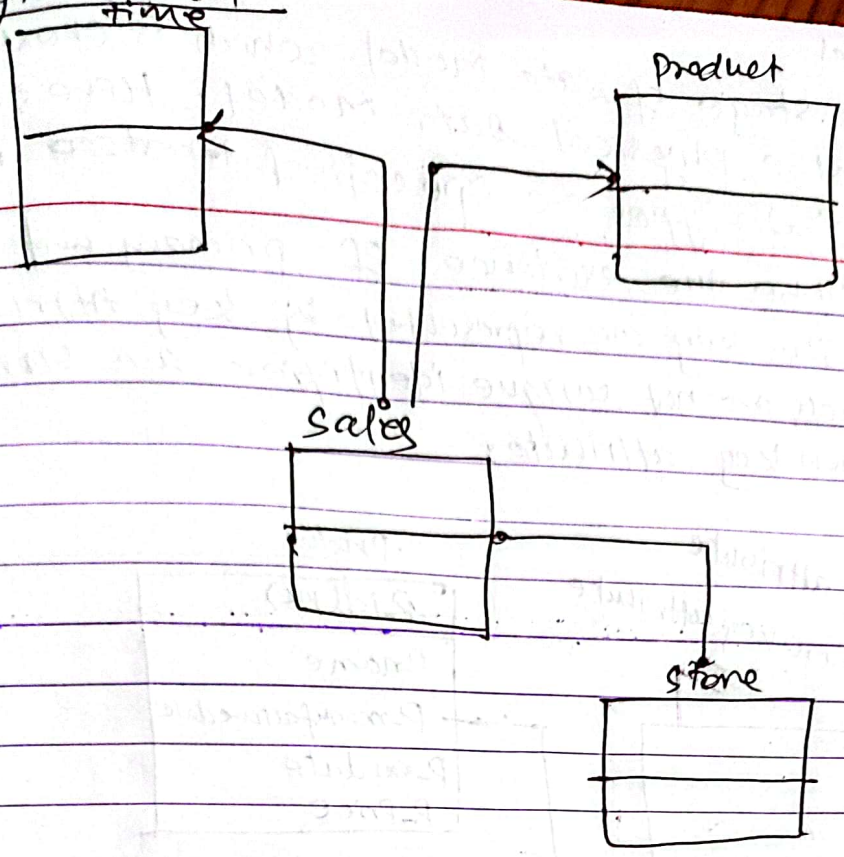


fig:- Representation of data model in conceptual data model.

↳ Here In the conceptual data model, data modelling is carried out by connecting the box, which represents the schema by lines that represents the relationship between schema: Here it is the preliminary stage of data modeling where no any attributes, it's data type are specified. It is a high level visualization of data model.

Some of the key points of conceptual data model

↳ It is usually draw on a paper or white board such that any further change can be done easily.

↳ It is highly abstract and represent the data model succesfully.

↳ It is easily understandable. such that whether a technical or non technical audience can understand it.

↳ No software tools are required to develop conceptual data model.

↳ The relationship between entity is also abstract

stands for
ed modeling

ER stands for entity relation Ship.

(14)

(2) Logical Data Model

↳ It is the second stage of Data Model which is created by enhancing the physical data model. Here the Attributes ~~and their types~~ are specified ~~earlier~~.

Here there will be the existence of primary and foreign key. The key are represented by key attribute and those which are not unique identifier are stored in group of non key attribute.

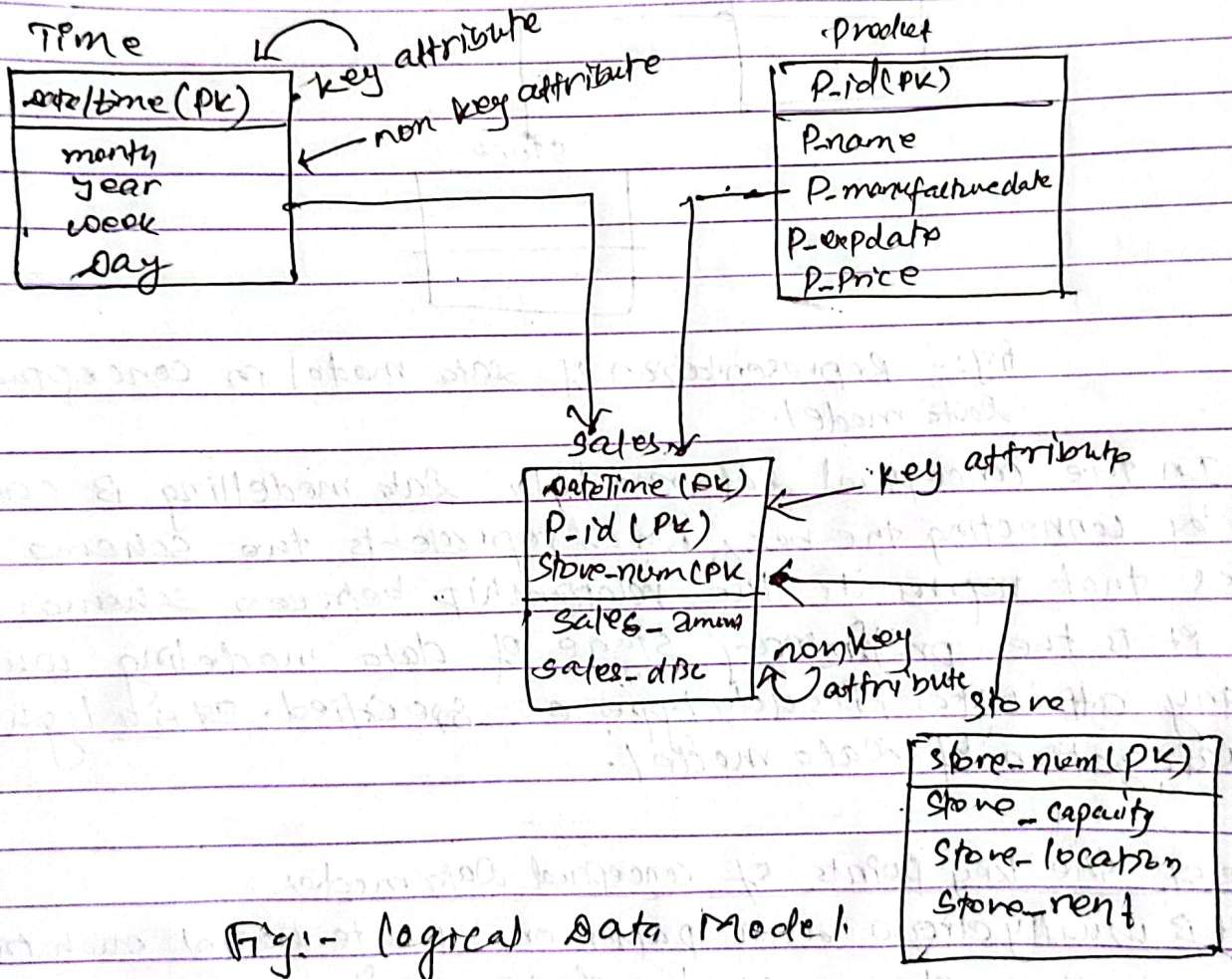
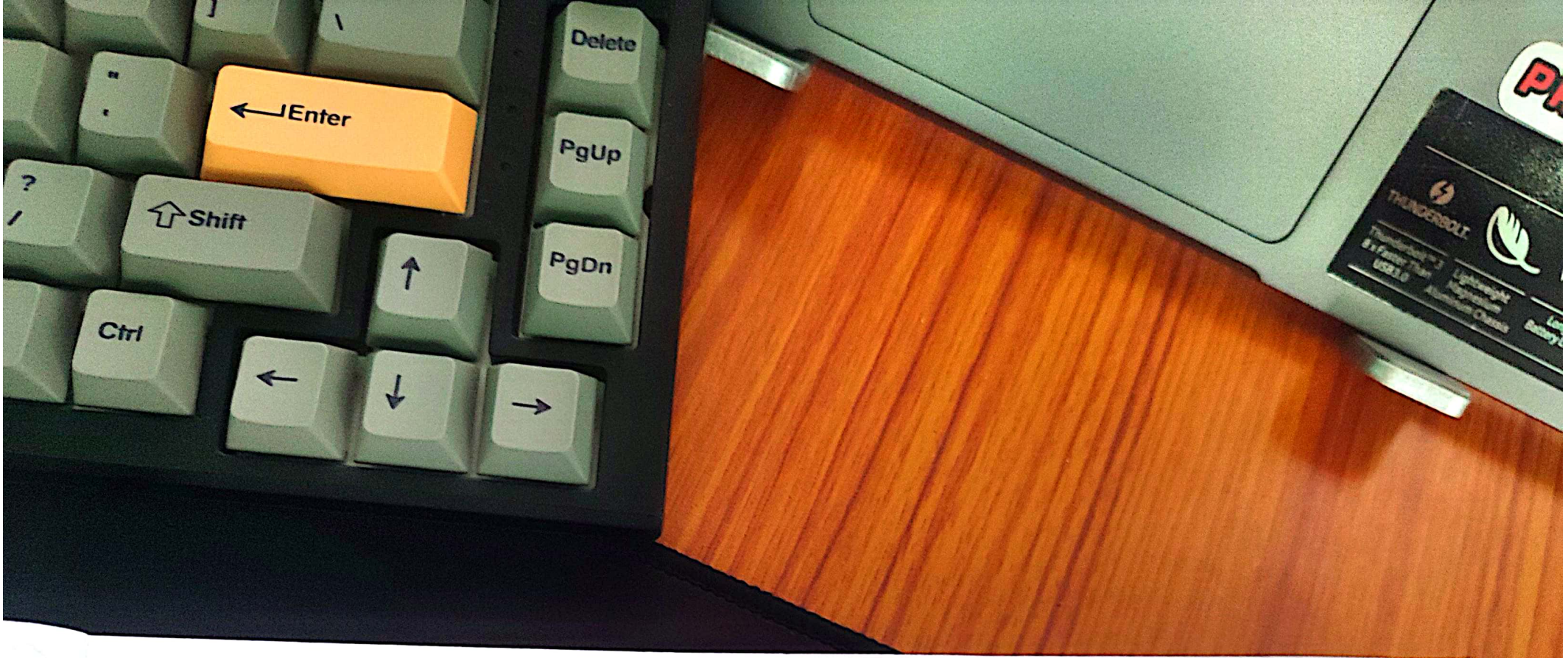


Fig:- Logical Data Model

Key points of Logical Data Model

↳ The relation between attribute will be visible. Also the primary key and foreign key will be shown

↳ user friendly attribute name will be given such that both technical and non technical audience can understand



↳ Little bit complex to accomodate any kind of change since lot of component will already be added.

↳ Proper CASE tools like: ERWin or power designer will be used above to which this model will later on can be converted into physical data model.

↳ This model is also free and independent of any D.B.

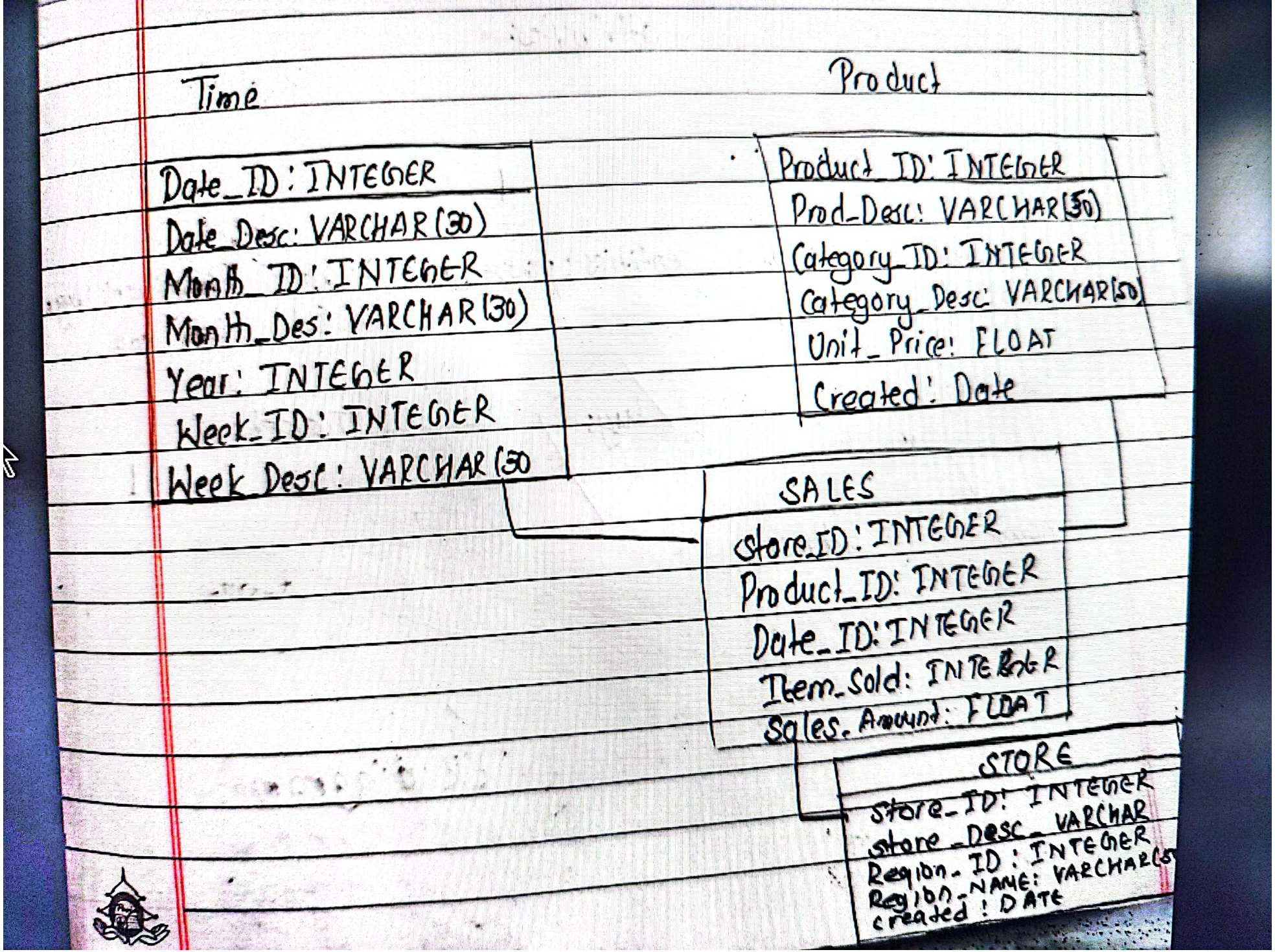
Physical Data Model

(c) Physical Data Model

↳ It is the enhanced and complex version of Logical data model where the entities are referred to as Tables, and the attribute are named as columns.

↳ The tablename will no longer be user friendly instead they will be database compatible and oriented towards the name of related DB. similarly the colmn name will be short and describe the content of one look.

↳ Data type will also be provided based on the data base developer will be working upon.
Thus the physical Data model is specific to a database.


Time
Date_ID: INTEGER
Date_Desc: VARCHAR(30)
Month_ID: INTEGER
Month_Des: VARCHAR(30)
Year: INTEGER
Week_ID: INTEGER
Week_Desc: VARCHAR(30
Product
Product_ID: INTEGER
Prod_Desc: VARCHAR(30)
Category_ID: INTEGER
Category_Desc: VARCHAR(30)
Unit_Price: FLOAT
Created: Date
SALES
store_ID: INTEGER
Product_ID: INTEGER
Date_ID: INTEGER
Item_Sold: INTEGER
Sales_Amount: FLOAT
STORE
store_ID: INTEGER
store_Desc: VARCHAR
Region_ID: INTEGER
Region_NAME: VARCHAR(
created: DATE

be storing.

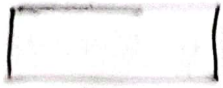
ER Model (Entity Relationship) {ERM}

↳ ER Model which stands for Entity Relationship model is a diagram containing the schema (entity / table), relationship among them and attributes of those schema.

It is one of the primilinary steps of Database design that helps designers to understands and specify the desired components of D.B schema and relation between those those schemas and components.

↳ The logical structure of the D.B is expressed by a graphical diagram known to be ER diagram which consists of following components

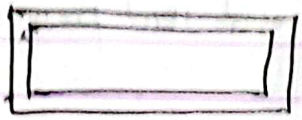
:

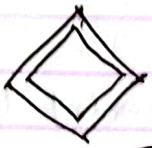
(a) Rectangle :  Represent Entity (16)

(b) Diamonds :  Represent relationship between entity sets

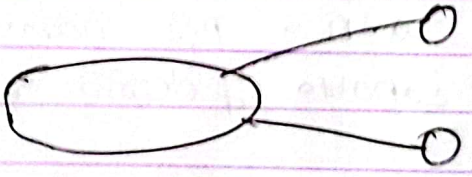
(c) Ellipse : Represent the attribute 

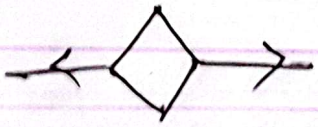
(d) Line : ~~Represents~~ Links the attribute with Rectangle (entity) via Relation.

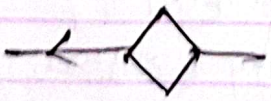
(e) Double rectangle :  Represents weak entity (those who don't have their own pk)

(f) Double Diamond :  weak entity relationship

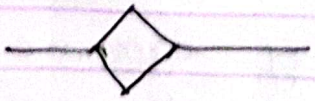
(g) ~~Multi~~ Double ellipse :  Represent the multi-valued attribute.

(h) Composite attribute : 

(i) Relationship :  : one to one relationship

 : one to many relationship

 : many to one "

 : many to many relationship.

Some of the Basic Concepts of ER model is:

→ ⓐ Entity & Entity set

Entity are the real world object which exists and distinguishable from other objects. Entity may be concrete such as customer, book or abstract like: Concept, customer etc.
Eg:- person, ABCcompany, Dog etc

→ Entity set are the collection of entities with common properties. (Table) → E.S attribute → Entity
Eg:- Market is an entity, then collection of market is entity set.

ⓑ Relationship and Relationship set

↳ Relationship is the relation between entities (attribute). whereas the Relationship set is the collection of relation of same type

Eg:-

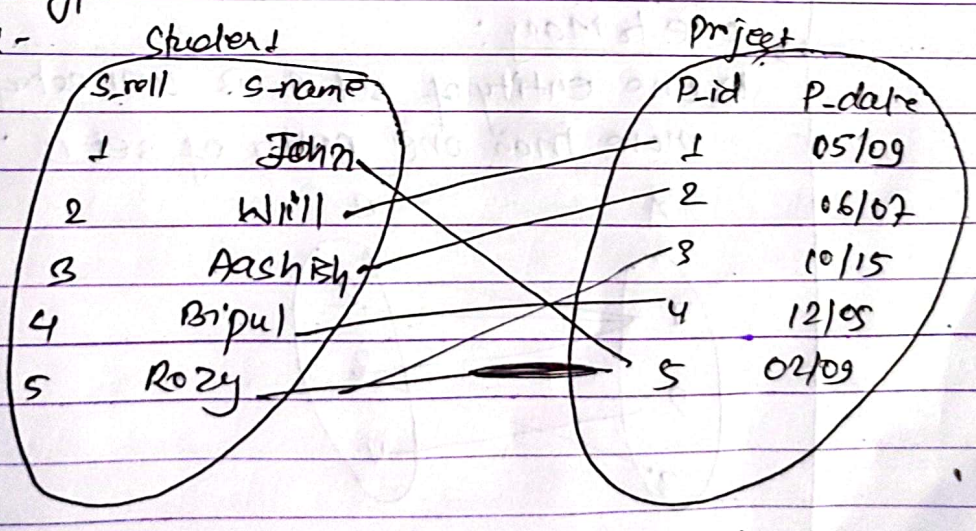


Fig:- Relationship set, convey each individual mapping is relation.

Constraints

Constraints are the rules or condition that the schema sets up, with which the data has to be sincere and conform (follow).

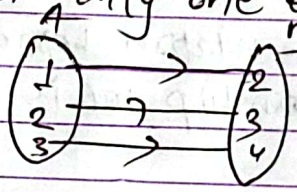
→ ⓐ Cardinality constraints
↳ Cardinality constraints are the types of constraints
ⓑ that specifies number of relationship between entities of the entity set.

It specifies how many entity one of one entity set can be associated with how many entity of another entity set.

↳ Cardinality constraints are:

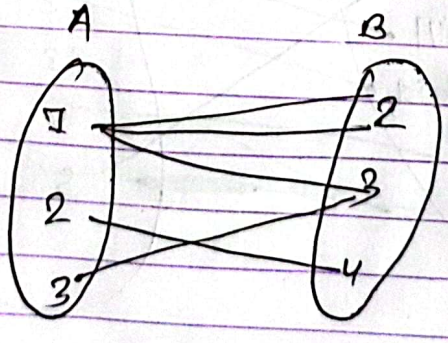
→ One to one:

One entity of entity set (say A) B connected with only one entity set of B.



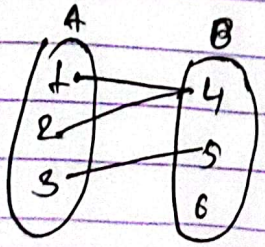
→ one to many:

↳ One entity of set A B connected with more than one entity of set B.



→ many to one

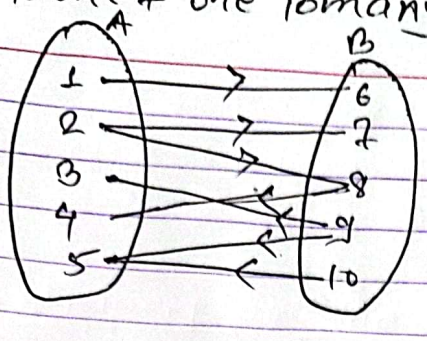
↳ many entity of set A B connected with one entity of set B



many to many

↳ one to one + one to many + many to one

↳



⑤ Participation constraints

↳ It is the kind of constraints that specifies the participation of entity of one set to another set.

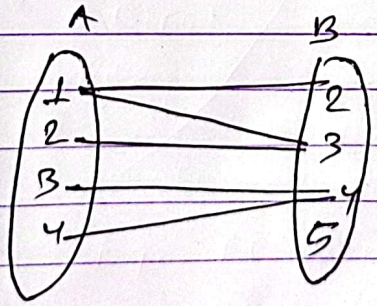
→ ⓐ Total participation

↳ Those type of participation where all the entity of one set is connected with entity of another set.

→ ⓑ Partial participation:

↳ Those type of participation when only limited entity of one set is connected to entity of another set.

Eg:-



A denotes total participation with entity of set B.

B " partial " " " " A.

cardinality of set A.

Attributes

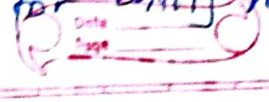
↳ Attributes are the components (part) that describe ~~abo~~ the entity.

→ Composite Attribute : Those made up of multiple sub attribute. eg: Name is composite since it further contains first name, Last name, middle name.

→ Derived Attribute: Those which are derived from other Attribute. eg:- Age is derived from DOB. Also, from attribute named "currentdeposit", the "totalamount" is increased.

→ Multi-valued Attribute : Those attribute which have multiple value that define the same property. eg:- Phone number is multivalued attribute.

key

UML class diagram	ER diagram
↳ UML in UML class diagram stands for Unified modeling Language	⇒ ER stands for entity relation ship. 
⇒ It is the blue print of programming language.	⇒ It is the logical structure of database scheme. (diagram that shows)
⇒ classes and objects are the building block of UML class diagram.	⇒ Entities are the building block of er diagram
⇒ Software engineer does the job of designing and handling UML diagram	⇒ Database administrator does the job of designing and using ER diagram
⇒ Here the flow of control is depicted	Here the relationships are depicted.
⇒ It is mainly used in case of large scale software development.	⇒ It is used in case of DB development of all scales
⇒ It uses swim lanes	It doesnot uses swimlanes

Ch3 Relational data Model

Player_info

Pid	Pname	Page	SSN	city
P1	Joe	32	2212	Manchester
P2	Will	31	2202	Lords
P3	Henry	33	2222	Auckland
P4	Amla	35	2111	Joburg
P5	Starc	32	2000	NSW

(a) **Relation** : It is the mathematical term of a table in Database. Relation is also known to be schema / table / entity etc.
eg:- Player_info

Tuple : It is the row of the table. Eg:
P1, Joe, 32, 2212, Manchester

(c) **Cardinality** : It is the number of tuples in a relationship. eg: player_info have cardinality = 5.

(d) **Attributes**
↳ They are the column name / column itself of a relation.
eg:- Pid, SSN etc

(e) **Degree** : No. of Attributes of a relation is called Degree of that Relation.
Eg: Player_info is of Degree 5

(f) **Domain** (P2)
↳ They are the pools of values in a relation.

Constraints in Relation Model
→ Domain constraints
→ key constraints
→ Referential Integrity constraints.

Domain constraints are the type of constraints of Relational model which defines the domain of the data. Value of the attribute must lie inside the specified domain.

Eg:- If the domain constraints of any attribute is NOT NULL, while inserting data value must be necessarily provided.

What is keyconstraints in relational data model with example

In the context of a relational data model, key constraints are rules or conditions that enforce the uniqueness and integrity of keys within a relational database table. Keys are attributes (or combinations of attributes) that uniquely identify each row or record in a table. There are two primary types of key constraints:

1. **Primary Key Constraint:**
   - A primary key is a set of one or more attributes that uniquely identifies each row in a table.
   - Each table can have only one primary key.
   - Primary keys enforce uniqueness, meaning that no two rows can have the same values for the primary key attributes.
   - Primary keys also ensure that the key attributes are not NULL (i.e., they must have values).
   - A primary key is typically used for identifying and linking records in a table.
   - Example: In a "Students" table, the "StudentID" column could be the primary key because it uniquely identifies each student.
2. **Unique Key Constraint:**
   - A unique key constraint enforces uniqueness for a set of attributes, similar to a primary key.
   - However, a table can have multiple unique keys.
   - Unlike primary keys, unique key attributes can allow NULL values.
   - Unique keys are used when you want to enforce uniqueness for a set of attributes, but they are not necessarily used for identifying records.
   - Example: In an "Employees" table, you might have a unique key constraint on the "EmployeeEmail" column to ensure that no two employees have the same email address, but the primary key could be the "EmployeeID" column.

Here's a simple example of a table with a primary key constraint:

Table: Students

diff Copy code

```
+-----------+------------+----------+-----------+
| StudentID | FirstName  | LastName | Birthdate |
+-----------+------------+----------+-----------+
| 1         | John       | Smith    | 1998-05-10|
| 2         | Jane       | Doe      | 1999-02-15|
| 3         | Robert     | Johnson  | 1997-08-22|
+-----------+------------+----------+-----------+
```

In this example, the "StudentID" column is the primary key because it uniquely identifies each student. The primary key constraint ensures that each StudentID is unique, and it cannot be NULL.

Key constraints are essential in maintaining data integrity and ensuring that the data in a relational database remains accurate and consistent. They help prevent duplicate records and maintain the relationships between tables.

CamScanner

Key constraints are the constraints that specifies attribute to be unique and hence serves as unique identifier.

Eg:-

Id	Name	Age	Roll
1	AB	12	22
2	BC	10	23

Here, the id is primary key that uniquely identifies the row. So while inserting data, it must be always provided.

(3) Integrity / Referential Integrity constraints

↳ Integrity constraints ensures that the changes done by authorized user in a D.B should result in loss of data consistency.

That means: Say table1 consists a primary key Rollnum which is acting as a foreign key in table2. If any tuple of table2 is deleted, tuple of table1 with that deleted foreign key must also be deleted.

Roll num	Name	age
1	Joe	22
2	Steven	21
3	Harpreet	22

must also delete

Rollnum (Fk)	marks
2	55
3	52
1	51

If deleted

Relational Model

↳ Relational Model is a database model where the data are represented in a tabular form establishing relations between different attribute. It is one of the mostly used data model. It is the model for Relational Data Base.

Schema Diagram

↳ Schem means the blue print of a table. It is a structure for an entity (Relation).

The schema diagram is the graphical representation of structure and relationship of database tables. It plays a vital role in database design since it gives the blue print of the whole system specifying the tables, attribute, their types that is need in D.B.

Schema diagram of a Bank:

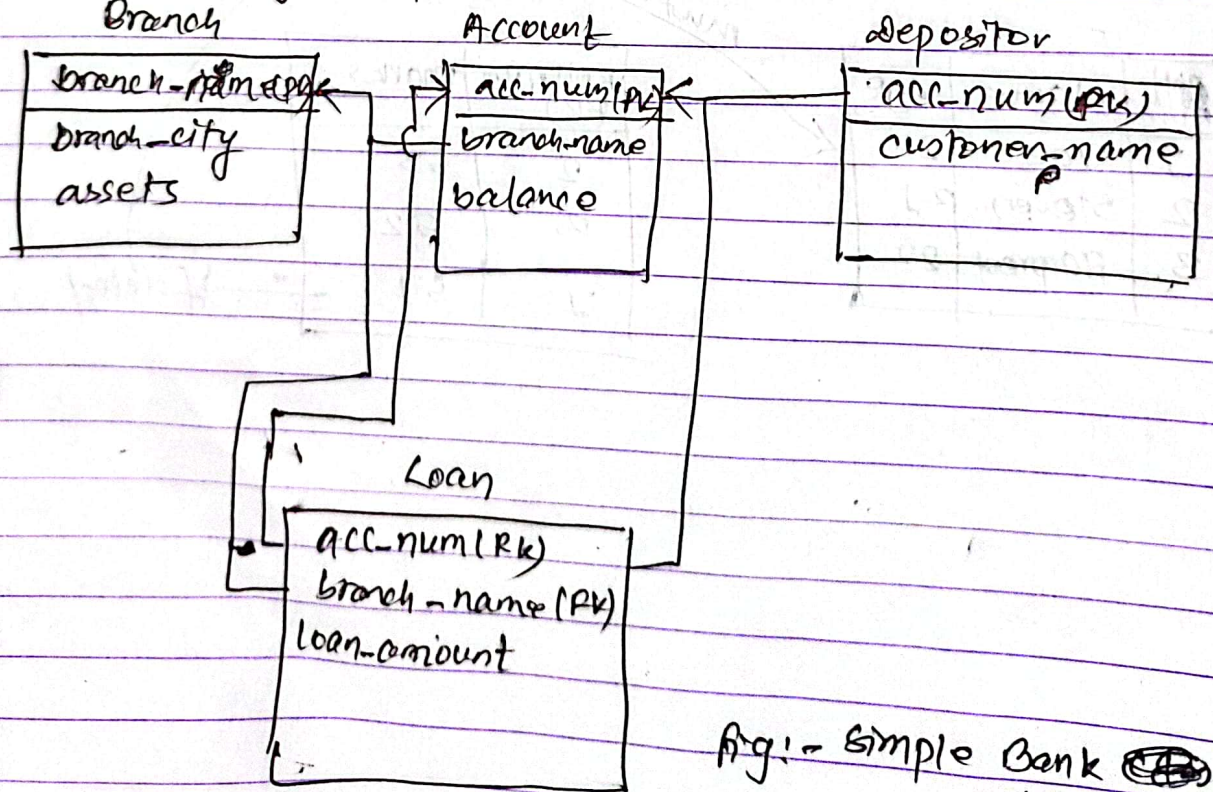


Fig:- Simple Bank Schema diagram

The Relational Algebra

Relational Algebra are the mathematical representation of query executed in SQL. It is a procedural query language, There are some operation in Relational Algebra :

① Select (σ) : It output those tuples from the relation, which satisfies the given predicate (condition). The syntax :

$\sigma_p(r)$ — relation
↑ predicate

Student

id	Name	age
1	A	22
2	B	23
3	C	24

Eg:-

$\sigma((age<23) OR (age>23))($ Student $)$

$\sigma(age>23)$ (Student)

id	Name	Age
3	C	24

② Project (Π) ⇒ selects a set of attribute from a given relation.
eg. Name, age selection from Student-table.

Π(Name, age) (Student).

Π(Name) (σ(age > 22) (Student));

③ Set union ($\cup$) : This operation combines the data of two tables where both table must have same number of colum of same type eg:

id	Name
1	Joe
2	Stacy

id	Name
3	ABC
4	AFG

A ∪ B =

id	Name
1	Joe
2	Steven
3	ABC
4	DPG

Select name from the both table whose id > 2,

$\Pi(name)(\sigma(id>2)(table_1 \cup table_2))$;

(iv) Set Intersection (∩): This operation is used to get the common tuple from two tables.

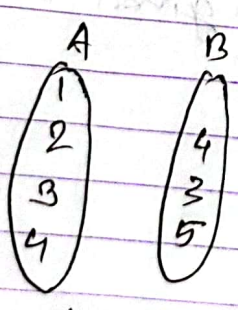
id	Name	age
1	Joe	22
2	XYZ	23

id	Name	age
2	XYZ	23
1	Joe	22
5	GHI	25

$\Pi(name)(\sigma(age>20)(table_1 \cap table_2))$;

(v) Set difference (−)

↳ This Returns the items of one table that is not part of another table.



A − B = {1,2,3,4} − {4,3,5}

= {1,2}

table1

id	Name	gender
1	AB	M
2	BC	M
3	DE	F

table2

id	Name	gender
3	DE	F
4	PG	M
5	XZ	M

$\Pi_{name}(table1) - \Pi_{name}(table2)$

$\Pi(name)(table_1 - table_2)$

SELECT E-name
comp. E-no = Dept-

(vi) Cartesian product (*)

↳ It is also known to be cross product or extra join that provides all possible of each tuple of table1 with each tuples of table2. combination

eg: table1

id	Name
1	AB
2	BC
3	CE

table2

Aege	Address
22	KTM
23	BRT
24	Chitwan

$\Pi_{Name}(\sigma_{(age>22)}(table1 * table2))$;

op:

Name
AB
BC
CE

id	Name	Aege	Address
1	AB	22	KTM
2	AB	23	BRT
3	AB	24	CHT
:			
:			
:			
3	CE	24	Chitwa

(vii) Rename : This operation is used to rename attribute, Table

(viii) Join (⋈): This as a database operation that is used to combine rows from two or more table based on a related column between them. The main purpose of join is to combine data from two or more table and create a (table2) resultset that combines the info in meaningful way.

- Natural join
- cross join (cartesian join) cross product
- self join
- inner join
- outer join (LO, RO, FO)

@ Natural join is the type of join that combines the rows from two or more table based on the common attribute of both table, with same name and same D.T.

Eg:

t1

id	Name	department_id
1	AB	100
2	CD	102
3	EF	104
4	GH	106

t2

department_id	Job
100	coder
104	musician
106	designer
102	QA tester

SELECT id, Name FROM t1 NATURAL JOIN t2

id	Name	department_id	Job
1	AB	100	coder
2	CD	102	QA
3	EF	104	musi
4	GH	106	des

@ cross join

↳ It is also known to be cartesian product or cross product and it combine every tuples of table1 with the tuples of table2. resulting in large result set

(c) Self join

Self join is a type of join where a table is joined with itself. It is often used when a table contains hierarchical or recursive data.

eg:-

Sid	Cid	Since
S1	C1	2016
S2	C2	2015
S1	C2	2017

So, find the sid of those student who are involved in more than one course

↳

SELECT a.sid FROM T1 as a, T2 as b where a.sid = b.sid AND a.cid <> b.cid

	t1		t2	
Sid	cid	Since	Sid	cid	Since
S1	C1	2016	S2	C2	2015
S1	C1	2016	S1	C2	2017
S2	C2	2015	S1	C1	2016
S2	C2	2015	S2	C2	2015
S2	C2	2015	S1	C2	2017
S1	C2	2017	S1	C1	2016
S1	C2	2017	S2	C2	2015
S1	C2	2017	S1	C2	2017

(d) Inner Join

↳ It is another kind of join which is also known to be equi join where a result set is found out by comparing the value of same column or different column as well.

⑥ inner join (Equijoin)

DATE

⇒ Find the Emp name who worked in a department having location same as their address.

emp

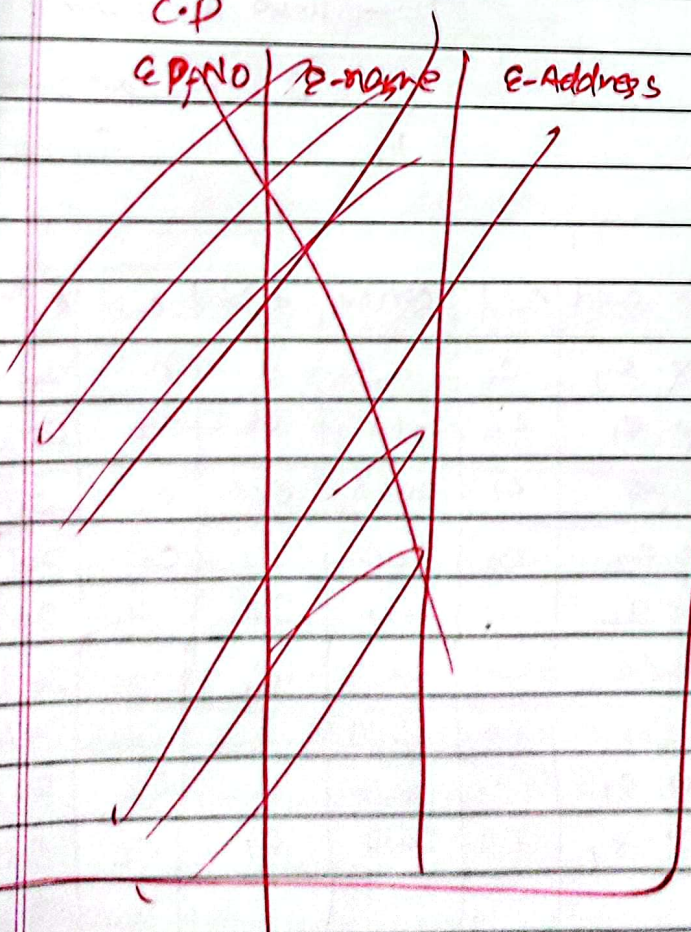
E-NO	E-name	E-Address
1	Ram	Delhi
2	Varun	chd
3	Ravi	chd
4	Amrit	Delhi

Dept

DepNo	Location	Eno
D_1	Delhi	1
D_2	Pune	2
D_3	patna	4

C.P

EDeptNO	E-name	E-Address



It is the type of join where we compare the value of the attribute that is common to the relation. but also those value of attribute which are not common.

SEECT-E-name from Emp, Dept where Emp.E-no = Dept.E-no and Emp.Address = Dept.city

SELECT E-name FROM Emp JOIN = Dept WHERE Emp.E-NO = Dept.E-NO and Emp.Address = Dept.city

PAGE

can be determined

nction:

(5) Outer Join

↳ It is that type of join where the matched rows are returned and also those row of either one or another table or both table will be returned.

→ Left outer join

↳ Type of outer join where the matching row and the remained row of Left table will be returned.

→ Right outer join

↳ Type of join where the matching row of both table and remained row of right table is returned

→ Full outer join

↳ matched + left table remaining + Right table remaining

CH 1 + CH 2 + CH 3 (SBank)

remaining Right table

CH 1 + CH 2 + CH 3 (Bank)

22 fall 19 Define DBMS ? What is the advantage of DBMS over traditional file system?

Soln DBMS which stands for Database Management System is the technology for software system by means of which the structure of Database can be made, different query like: ALTER, UPDATE, DROP etc can be executed with the utmost efficiency maintaining a proper security measures.

As we know that Data is the facts, information, statistics etc, which will be in raw and unorder form. Database are the logical and structured collection of those data and their description in a way that it fullfills the data based demand of organization in a large sized memory or drive.

...ed attribute as s_id, book_id which is also a foreign key and here in this table it acts as a composite primary key since same student can't borrow same book more than 1 time.

CamScanner

Hence the Database Mgmt System is the collection of computer programs by which those data can be used in most efficient way to perform different operations. (23)

Traditional file System (TFS) is a method of storing data and information in the form of file in a physical paper or a computer device. Here in this file system, the data and information will be noted in random order without maintaining standard pattern and have high chance of storing redundant data.

There are lots of advantage of DBMS over Traditional file system:

(a) The data and information can be easily retreived, update, altered with the help of DBMS, which was quite tough and time consuming in TFS.

(b) Abstraction is maintained by DBMS where the user(view label) is topmost level, logical level is the second level where the pure logic is stored and last is physical level, which leads to maintain high security, which is not possible in TFS.

(c) DBMS provide features like COMMIT, ROLLBACK, SAVEPOINT to make sure that change should be persistent and can be step back unless COMMIT happens but TFS don't have it.

(d) Data Backup is possible in DBMS where most of the data will be stored in remote database securely but in TFS, Device memory were used to store data, that leads to high chance of data loss in case of system failure.

(e) DBMS controls the redundancy not allowing the duplicate data to be entered in DB, which is not possible in TFS.

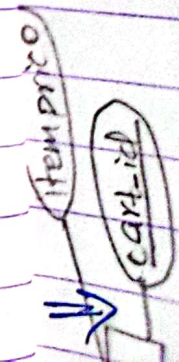
So, these were some Advantage of DBMS over TFS

22fall IY Why do you need ER diagram ?. Draw an ER diagram for online shop Mgmt system, with assuming relevant entities and attributes for the given system. (20)

↳ ER diagram which stands for Entity relationship is a diagramatic representation of logical structure of Database that helps developers and Database designers to understand and visualise the whole system and know about what are the attributes, entities that is needed.

ER diagram is one of the most important part of Database design and development where the whole data flow of the system can be depicted in pictorial form that leads to development of most efficient database.

The simple ER diagram of online shop management system is:



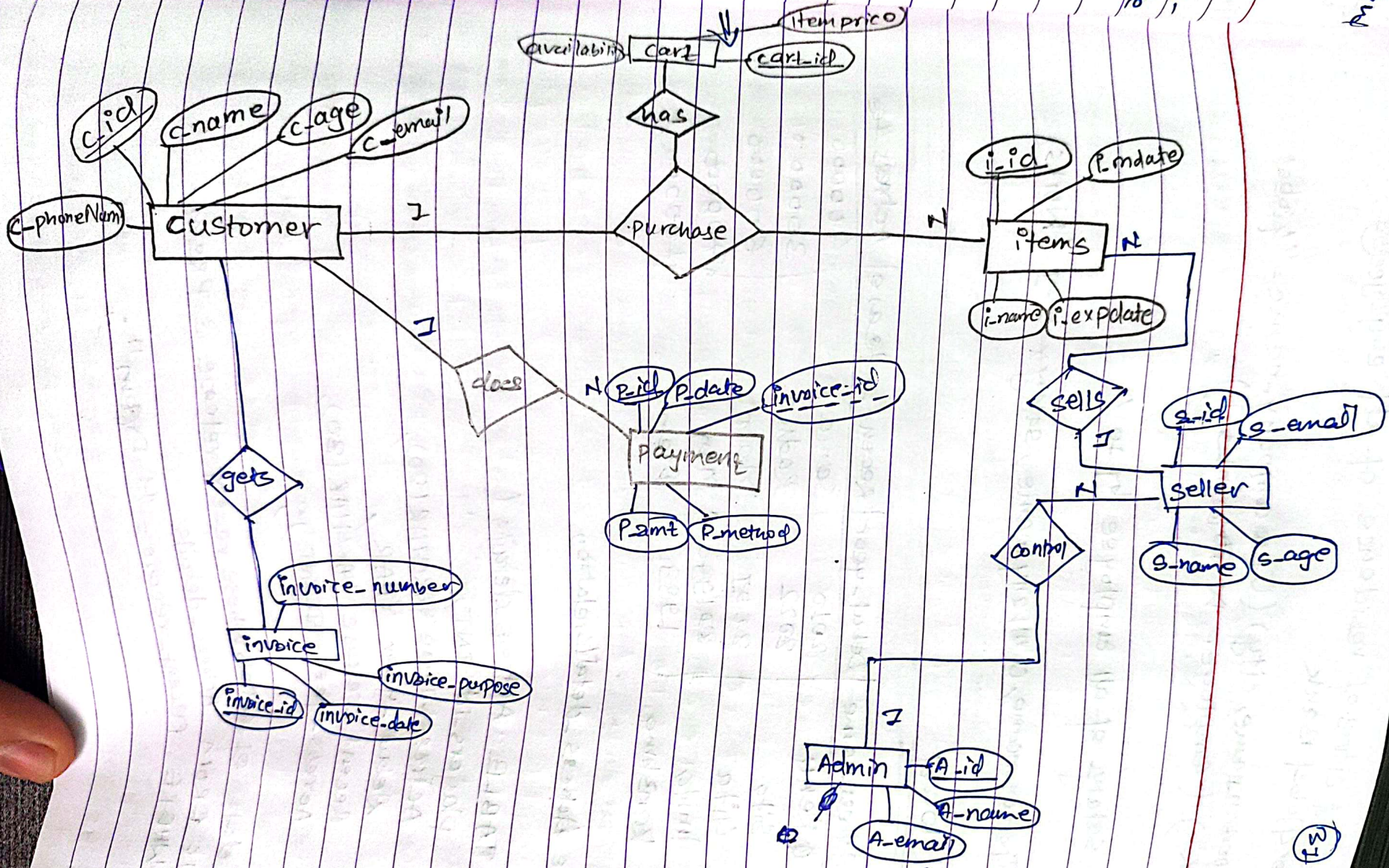
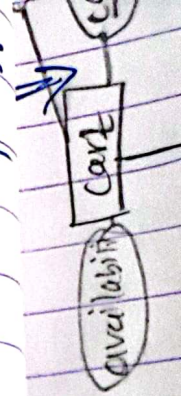


fig:- Online shop Mgmt system.

The simple ER diagram of online shop management system is:



22fall29 The given relations are:

Employee (person_name, street, city)
Works (person_name, company_name, salary)
Company (company_name, city)

Write Relational algebraic exp:

(i) name of employee who lives in "Butwal" and have salary < 50,000

↳ ~~Π(name)~~

$$\Pi(\text{person\_name})(\sigma(\text{city} = \text{"Butwal"} \wedge \text{salary} < 50{,}000)(\text{Employee} \bowtie \text{Works}));$$

(ii) Name of all employees who works for "Nabil Bank Limited"

Soln

$$\Pi(\text{person\_name})(\sigma(\text{company\_name} = \text{"Nabil Bank Ltd"})(\text{employee} \bowtie \text{Works}));$$

is also a foreign key and have in this table it acts as a composite primary key since same student can't borrow same book more than 1 time.

CamScanner

(3) Find the name and cities of residence of all employee (32) who work for Global Bank

↳ ~~Fname~~ Π (person_name, city) (σ (company_name = "Global Bank") (employee $\bowtie$ works));

(4) Update the salary of all employee by 10%.

↳

works ← Π person_name, company_name, salary * 1.1 (Works)

22 fall SQL wala

players_id	Actress_name	Debut_year	Recent_release	Actress_fee
1	Renu	2010	Samay	400000
2	Sita	2022	Radha	300000
3	Gita	2001	Mato	600000
4	Amita	1990	Man	700000
5	Kanchma	1989	Prem	100000

(i) create table Actress_details relation;

↳

```
CREATE TABLE Actress_details (
        players_id INT,
        Actress_name VARCHAR(20),
        Debut_year YEAR,
        Recent_release VARCHAR(30),
        Actress_fee BIGINT);
```

(ii) Delete data of actress whose recent release is Prem.

↳

```
DELETE FROM Actress_details
    WHERE Recent_release = "Prem";
```

(iii)

is also a foreign key and here in this

(iii) Modify the database so that RENU's new release is "Win the RACE";

Sol^n ⇒ UPDATE TABLE Actress-details
SET Recent-release = "Win the Race"
WHERE Actress-name = "Renu";

(iv) Insert a new record
↳ INSERT INTO Actress-details VALUES (
6, "Babita", 2022, "Hello world!", 2500000);

21SP79 | S.Notes | 21 SPRING

Data Dictionary

⇒ Data dictionary is the metadata about data. It can be defined as the module, package where the detailed information of data, their types, Database, detailed documentation, size of data, degree of database, guidelines etc sort of thing will be defined in a standard pattern.

Some of the advantage of D.D

↳ Assists in providing data consistency across the whole project
↳ Defines the semantic of data element and for what purpose they are there.
(semantic = meaning)
↳ Makes easier to analyze the data.
↳ It forces to follow the standards and guidline which have been in practise for long time.

Thus, the Data Dictionary describes the semantic and use case of data elements with in the domain of project and provide roadmap / guidance on Design & Development of Strong, efficient Data base.

(34)

2 ISPya What is data independency? Why is it needed in DBMS? Explain detailly.

Sol^n Data Independency can be defined as the independent relation of Database schema with the data of the D.B. As we know that Database schema is the blue print (structure) of the Database table, that is required for the construction of successfull, strong, robust relation without any mistake. So, the term data independency states that, the change done in the data stored in the relation should not affect the structure of the table.

Data independency is needed in the DBMS, to ensure the Data persistency and reduce the chances of data loss. As we know Data is the new oil. Those companies which have the user Data are in top of the stock with high profit. Assume a condition, say we a team of 5 good top software Engineers built a social media platform like Facebook. All the thing are well developed and top notch and deployed. But where will the audience comes from. Those audience would have been using FB since long time, why will the join our platform. So, without audience i.e. Data our platforms will be useless and we won't be getting any profit. So, for this purpose, Facebook makes sure the user Data won't get loss in case of any kind of changes. That why D.I is needed.

The data independence is divided into logical data independence and physical Data Independence where the physical makes sure that any changes done in the physical level of D.B won't shows its effect in logical level of DB. Whears the Logical Data independence ensures that any changes done in the logical schema of DB won't effect the top view level, which will be difficult since top level is application program directly contacts with logi

is also a foreign key & and here in this table it acts as a composite primary key since same student can't burrow same book more than 1 time.

(35)

cal level
Some of th
(a) Fle
inte
(b) Pa
(c) A
(d) C

cal level for fetching & storing data.

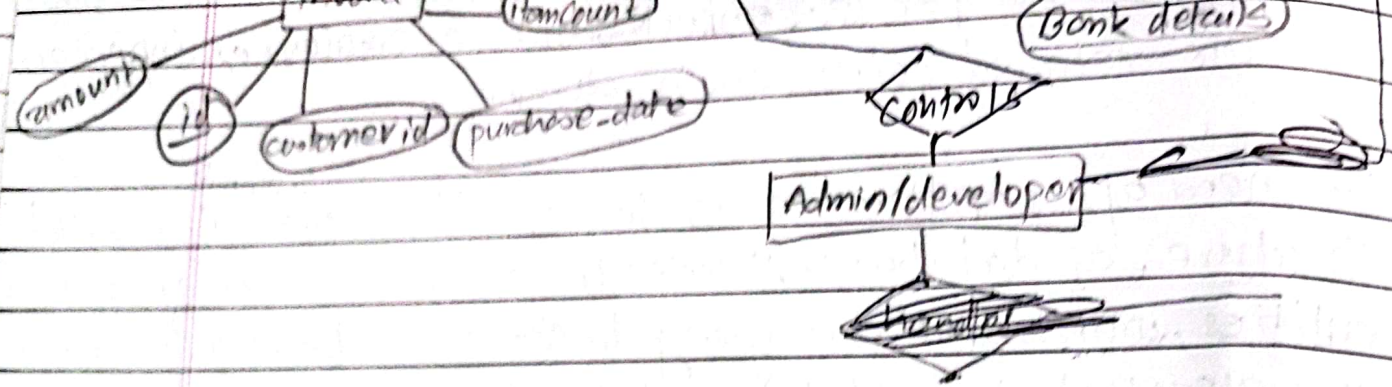
Some of the reason to use Data Independency DBMS

↳ (a) Flexibility ⇒ Allows to change the technology and logic without interrupting existing program.

(b) Maintainability ↳ It makes easy to maintain

(c) Abstraction ↳ 3 level of Abstract

(d) Compatability ↳ It ensure the D.B will be as compatible as it was initially, when certain changes in any level happens.



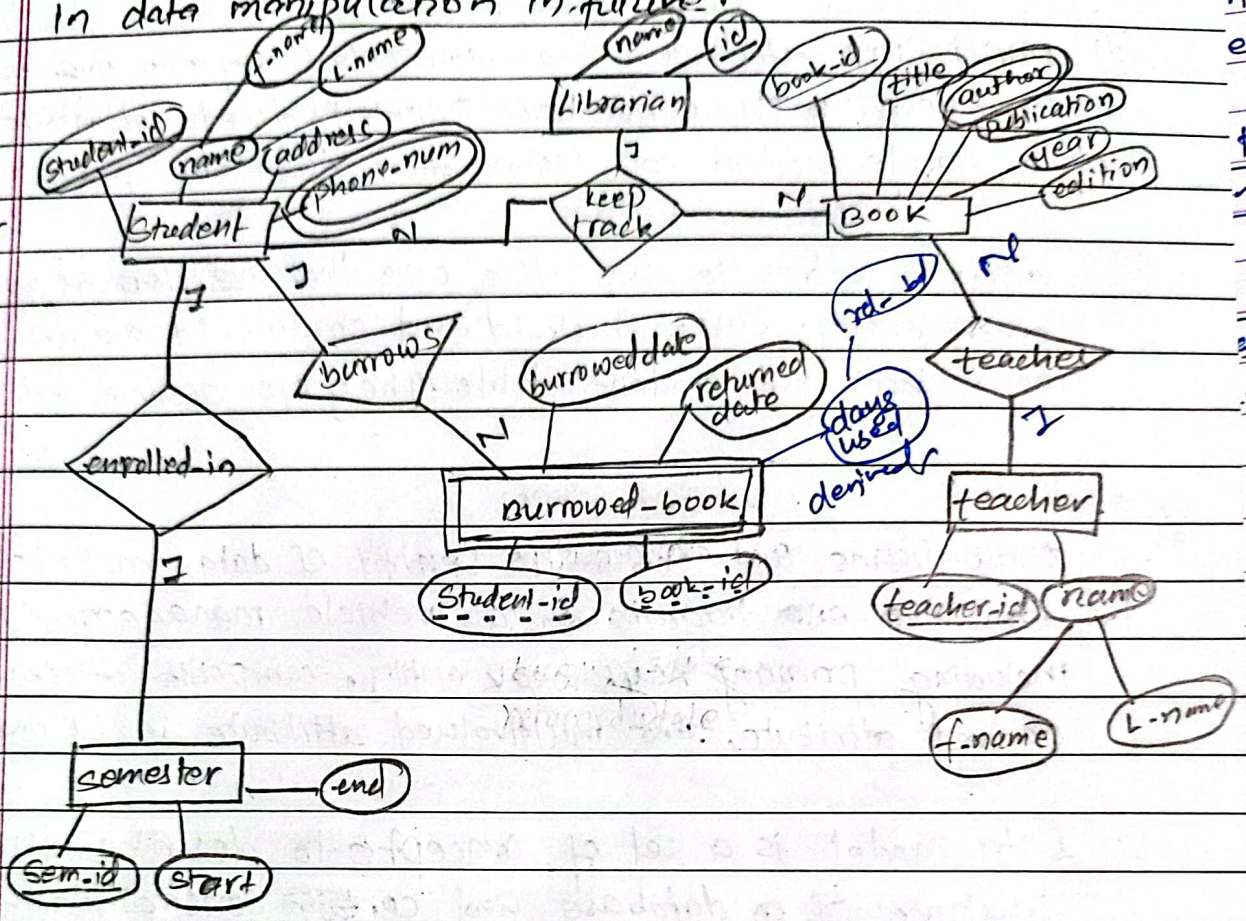
for 2nd

(b) What is ER diagram? Draw an ER diagram for a library system assuming the entities student, teacher, book, and semester. Also illustrate the concept of strong entity, weak entity, composite attribute, multivalued attribute and derived attribute.

↳ ER which stands for Entity Relationship is a diagram that diagramatically represents the relationship between entity group. It is a modelling technique that is used to design and represent the logical structure of DB.

classmate

It creates a roadmap of the data and attribute that we need to or have uploaded in database that helps in data manipulation in future.



Here, the explanation is done as:

(a) Here student and teachers are strong entity set since they have their own unique identifier as student_id and teacher_id. Also, semester, book are also strong entity since they have their own sem_id and book_id.

(b) Here the weak entity is borrowed book since borrowed book doesn't have its own unique identifier since we need the students id and book id to identify which student has burrowed which book. Also, here burrowed book has derived attribute as s_id, book_id which is also a foreign key and here in this table it acts as a composite primary key since same student can't burrow same book more than 1 time.

CamScanner

DATE

(c) Here name is composite attribute that contains 2 sub attribute fname and Lname.

(d) Multivalued attribute are represented by two oval boundary. Here phone number is multivalued attribute since a single student can have more than one phone number.

(e) Derived attribute are the one that are derived from other entity. Since book_id and student_id are derived from book and student table. They are derived attribute

2078 fall

7c ~~Draw~~ Define and explain the benifit of data model. Draw the ERD diagram for a vehicle manage

level of Abstraction

(d) **Compatability** ⇒ It ensure the D.B will be as compatible as it was initially, when certain changes in any level happens.

2JSP29 ⇒ Schema can be defined as the blueprint for the structures of the relation which is going to be built that represents the high level representation of the relation and relation of attributes.

View on the other hand can be defined as the virtual table which contains some portion of tables data. It provides an abstraction layer over the underlying data, allowing user to query and manipulate data without the need of accessing underlying table.

Syntax: CREATE VIEW view_name AS
SELECT col1, col2, ...
FROM table_name
WHERE condition;

→ 3 level of Abstraction

(d) **Compatability** → It ensure the D.B will be as compatible as it was initially, when certain changes in any level happens.

2JSP29 ⇒ Schema can be defined as the blueprint for the structures of the relation which is going to be built that represents the high level representation of the relation and relation of attributes.

View on the other hand can be defined as the virtual table which contains some portion of tables data. It provides an abstraction layer over the underlying data, allowing user to query and manipulate data without the need of accessing underlying table.

Syntax: CREATE VIEW view_name AS
SELECT col1, col2, ...
FROM table_name
WHERE condition;

Sailor(sid, sname, rating, age)
Boats(bid, bname, color)
Reserves(sid, bid, day)

Sailor

sid	sname	rating	age
1	A	1	22
2	B	1.5	23
3	C	2	24
4	D	5.6	25

Boats

bid	bname	color
103	Z	green
104	X	blue
126	Y	purple
133	W	white

Reserves

sid	bid	day
4	103	5
2	104	1
3	126	6
1	133	2

(i) Find the record of sailor who have reserved boat num 103 (bid=103)

↳ $\Pi(sid, sname, rating, age)(\sigma((sailor.sid = Reserves.sid) \wedge (Reserves.bid = 103))(sailor \bowtie Reserves))$;

(ii) Update the color of boat where bid = 104 into green

↳ Boats ← $\Pi(bid, bname, color = \text{"Green"})(\sigma(bid = 104)(Boats))$;

(iii) Find the name of sailor who have reserved green or red boats.

↳ $\Pi(sname)(\sigma(color = \text{"Green"} \wedge color = \text{"Red"})(sailor \bowtie Boats \bowtie Reserves))$;

(iv) Find the name of sailors who have reserved boat num = 103 on day 5.

↳ $\Pi(sname)(\sigma(bid = 103 \wedge day = 5)(sailor \bowtie Boats \bowtie Reserves))$;

(v) Name of sailor whose name is not Ram

↳ $\Pi(sname)(\sigma(sname != \text{"Ram"})(sailor))$;

(58)

2J sp 25 DDL which stands for data definition language is a structure of schema defining Language and contains:

- → CREATE
- → ALTER
- → DROP
- → TRUNCATE
- → RENAME

DML ⇒ It stands for Data manipulation Language and is used to manipulate data on the existing data base.

It consists:

- → INSERT
- → SELECT
- → UPDATE
- → DELETE

(i) SELECT Sname FROM Sailors NATURAL JOIN Reserves WHERE bid = 103;

(ii) UPDATE TABLE Boats
SET color = "green"
WHERE bid = 104;

(iii) SELECT Sname FROM sailor NATURAL JOIN (SELECT * FROM Boats NATURAL JOIN Reserves)
WHERE color = "green" OR color = "RED";

(iv) SELECT Sname FROM Sailor NATURALJOIN Reserves WHERE bid = 103 AND day = 5;

DATE

(38)

(v) ⇒ SELECT bname FROM Boats;

2]Sp19 Explain the types of integrity constraints: Explain with eg:

→ (a) Domain c.s

→ (b) Key constraints → must be unique enough to differentiate each row (by CS Eng gyan) (✓) pre)
Primary key constraints & unique key cons (by GPT)

→ (c) Referential integrity constraints:
PK & FK ko rein

→ (d) Entity integrity constraints - where the PK must not be null. as it unique identifier.

2] fall

2fall

(1a)

Schema	Instance
↳ schema is the blue print of the relation that represent the structure of table to be designed.	Instance is the snapshot of data at a particular instant of time.
↳ It is categorised into 2 types: (a) physical schema (b) Logical schema	↳ No any categorisation is needed, so not done yet
↳ No frequent update is done in schema.	↳ Here the data Instance changes over time and
↳ When the change needs to be done in D.B schema, carefull planning is done where D.B administrator and schema designer	↳ The changes are extremely fast process. Hence when performing CRUD

(39) gets

Any changes shows table stru

If we try to c
, it indicate
of programm

Eg: In a
might
ge
of enti

2]fall1b

It is the ty
has the lo
dabase

↳ Data Mod
(a) Concep
(b) Logic
(c) Physi

↳ The design
complex th
in multi
specificati
and mo

(39) gets involve

	operation, it will be changed by the user itself.
Any changes done in the schema shows its effect in overall table structure.	The change happens in the database's data, so no structural effect will be seen.
If we try to correspond it with programming, it indicates the variable declaration of programming lang.	⇒ It is like the value of the variable of programming, which will change overtime.
Eg: In a D.B Library, schema might contains, Book, Author, Genre, publication etc kind of entity (table).	Eg:- In the same D.B Library, the instance is like: DBMS-for-Beginners, Andreo Riemann, Database development, AR publishers etc.

21/11/16

Data Model	ER Model
It is the type of model that defines how the logical structure of the dabase is designed.	ER Model is the graphical representation which contains the entity, attributes and their relationship.
↳ Data model consists of 3 stages (a) Conceptual Data Model (b) Logical Data Model (c) Physical Data Model	↳ No any stages are there but is consists of set of components like: Rectange, oval, diamond which represent entity, attribute and relation between entity and attribute resp.
↳ The design of data model are quite complex than ER diagram since in multiple phase the level of specification will be increased and move toward special d.B	↳ It is way more simpler to design.

(4) Data model often deals with normalization & denormalization to control the redundancy and optimize data storage	(4) It don't deal with normalization but provide a base for designs of Normalized D.B. (40)
Ex: Relational Data Model, Hierarchical " 4, Network " "	eg:- Actor entities contains customer_id, customer_age as attribute

(42)

↳ Π(bname)(Boats)

2. Aliza Give an expression in relational algebra to express each of following queries.

Doctor(SSN, FirstName, LastName, Speciality, Years of experience, phonenum)
patient(SSN, FirstName, LastName, Address, DOB, PrimaryDoctor-SSN)
Medicine(TradeName, Unitprice, GenericFlag)
Prescription(Id, Date, Doctor-SSN, patient-SSN)
prescription-Medicine(PrescriptionId, TradeName, NumOfUnits)

(i) List the trade name of generic medicine with unit price less than $50
↳ Π(TradeName)(σ(unitprice < 50)(Medicine))

(ii) List the first and Last name of patients whose primary doctor named 'John smith'
↳ Π(FirstName, Lastname)(σ(Primarydoctor = 'John smith')(patient))

as attribute

2 If all 2a (iii) List the first & lname of doctor who are not primary doctor to any patient

↳ π(FirstName, LastName)(σ(SSN="π(SSN)(Doctor)" − π(primaryDoctor-SSN)(patient))(Doctor)).

This is the only way to retrieve data since the join used in relational algebra & natural join which gives the tuples based on same name and type of attribute. But we want those doctor who weren't involved.

(iv) List the SSN of distinct patient who have "Aspirin" prescribed to them by doctor named "John Smith".

Soln

π(SSN)(σ(tradeName="Aspirin" ∧ FirstName="John" ∧ LastName="Smith")(patient ⋈ prescription ⋈ prescription_medicine ⋈ Doctor));