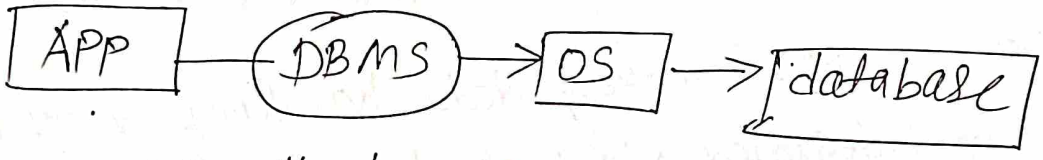
DBMS

* What is DBMS and Needs of DBMS.

ans ⇒ DBMS stands for Data Base Management System, It is a collection of interelated data and it has set of programs to access the data or DBMS is a interface b/w OS and Application, The primary goal of DBMS is

(1) To provide a Base to store data and
(2) Retrieve a data or information from database efficiently.



* Need of DBMS

(i) To organize the data efficiently.
(ii) Managing the data at large scale efficiently.
(iii) It provides Data security & privacy.
(iii) DBMS is scalability and flexibility.
(iv) Easy to insert, Delete, Update, retrieve data from database.

* Application of DBMS

(i) Railway Reservation system.
(ii) Library management system.
(iii) Banking.
(iv) Social media sites.
(v) online shoping
(vi) Healthcare.

* Advantages of DBMS

(i) Data security: It improves data security from unauthorized access

(ii) Reduction in data redundancy: DBMS helps to reduce redundancy of data, it does not maintain seprate copy of the same data, all data is kept at one place and it's helps to reduce duplicacy.

(iii) Data tranferring: DBMS provides a advantages to the user to transfer the data in a fast execution.

(iv) Data Integrity: DBMS makes data handling very easy and gives an integrated view of data in database.

(v) faster data access: DBMS provides faster and more accurate access of data from database.

(vi) Efficient data access and retrieval: DBMS allows for efficient data access and retrieval the data by providing indexing and query optimization techniques that speedup data retrival.

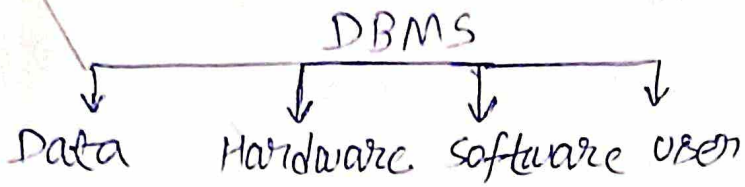
* Disadvantages of DBMS

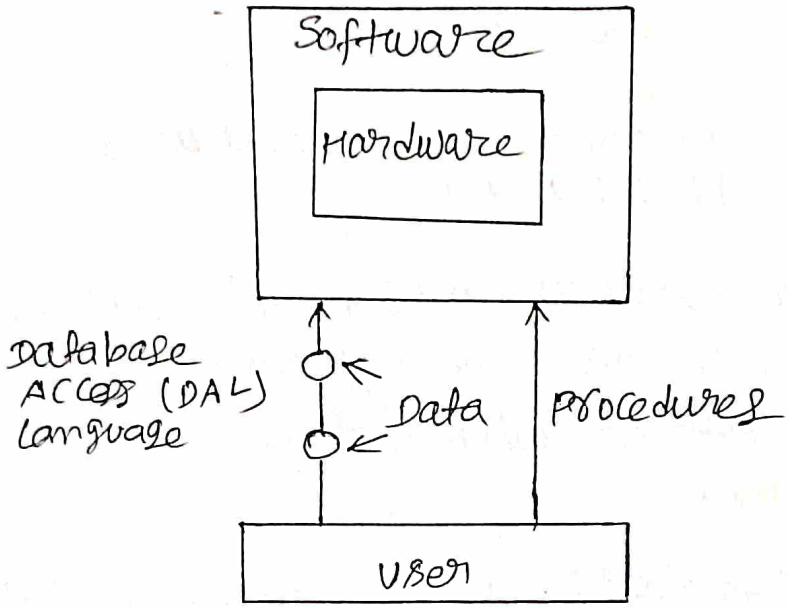
(i) Cost of Hardware and Software: Cost of the HW & SW is very high.

(ii) Complexity in backup and recovery.

(iii) Sometime centralization creates the problem.

* Explain the components of DBMS.





(i) Data: Data is the very important component of the database system. Data is that resource, for which DBMS was designed. Data can be any information, text, audio, video, Jpeg, PNG, Pdf, ppt, file. DBMS work on data for storing, retrieving, update, delete data.

(ii) Hardware: Hardware is a collection of physical components of the computer, that mean computer, hard disks, I/O device, storage device, This components are used for keeping and storing the data in the database.

(iii) Software: The main components of a DBMS is the software. It is the set of programs which is used to manage the database and to control the overall computerized database.
Ex ⇒ My SQL,
DBMS Software provides an interface to store, retrieve, and update, delete data in database.

(iv) Users: The users are the people who control and manage the databases and perform different types of operations on the databases in the DBMS.

There are three types of User

- Application programmers
- Database Administrator
- End users

(1) Application programmers: The person who write the application programes in programming lenguages such as Java, C++, MYSQL to interact with databases are called application programmer.

(ii) Database Administrators: A person who manages the overall DBMs is called DBA.
(DBA)

(iii) End-users: The end users are those person who interact with the database management system to perform various operations by using various data base commands such as insert, delete, updata, retrieve and delete on the data.

---

* Explain Data Redendancy: when multiple copies of the same information are stored in more than one place at a time this is called as data redundancy.

* define schema: A schema is the blueprint or structure that defines how data is organized and stored within a database. Types of schema

(1) Physical schema (2) Logical schema (3) External schema

(1) physical Schema: physical describe how data is to be represented and stored in secondary storage device using a particular ~~database~~ (DBMS).

* What is difference b/w database system and file system.

Database system	File system
(i) database is a software for managing the data	File system is a way of managing the data in a storage device with computer.
(ii) No data redundancy.	(ii) data redundancy is present.
(iii) Backup and recovery is possible in database system	(iii) Backup and recovery is not possible in ~~database~~ file system.
(iv) More complexity compared to the file system	(iv) less complexity compared to the DBMS.
(v) More security mechanisms	(v) less security mechanisms.
(vi) Higher cost than file system	(vi) Less cost than database system
(vii) Data abstraction is available	(vii) No data abstraction.
(viii) Ex ⇒ C++, cobol	(viii) Ex ⇒ SQL server, oracle.
(ix) Crash recovery	(ix) No crash recovery
(x) protection is high	(x) NO protection

(2) Logical schema: Logical schema describes structure of the data and relationship between various attributes, tables and entity.

(3) External schema: External schema defines/describes that how a specific user or application sees and interacts with data in dbms in database. It provides customized abstraction according to their requirements. view of the data for a specific user or application.

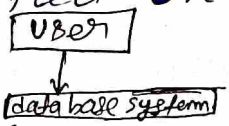
* Architecture of DBMS

ans=> DBMS architecture depends upon how users are connected to the database to get their request done.

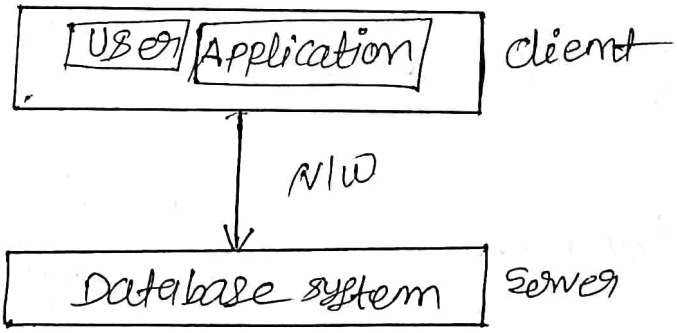
Types of DBMS Architecture

(1) 1-tier Arc	(2) 2-tier Arc	(3) 3-tier Arc

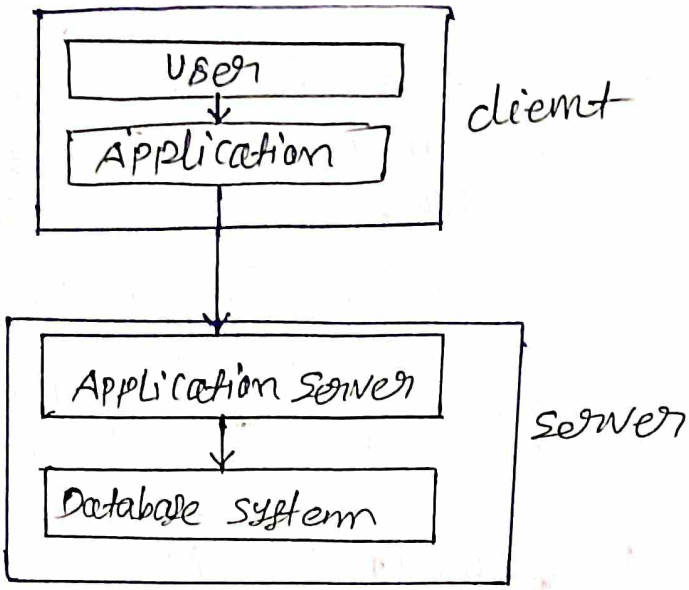
* 1-tier Architecture: In - 1 tier Architecture, the database is directly available to the user. It means the user can directly access the data from database. Any changes made by user will directly reflect on the database itself.



(2) 2-tier Architecture: In 2-tier architecture, Applications on the client end can directly communicate with the database at the server side for this interaction APIs are used.



(3) 3-tier Architecture: In 3-tier Architecture a another layer is between the client and server. client cannot directly communicate with the server. The Application on the client-end interacts with an application server and application server will further communicate with the database system.



* Data Model or Database model : Data models describe A database model in database management system is a framework that determines the logical structure of a database, including the relationships and constraints that will define how data is stored, organized and manipulated.

There are many kinds of database model

(i) Relational database model
(ii) Hierarchical database model
(iii) Network model
(iv) E-R model
(v) Object oriented database model.

Heron

(i) Relational database model : Relational database model organize data in the form of tables with rows and columns, where each row represents a record and each column represent a attribute. And tables are known as relations. Relationships between tables are established using keys.

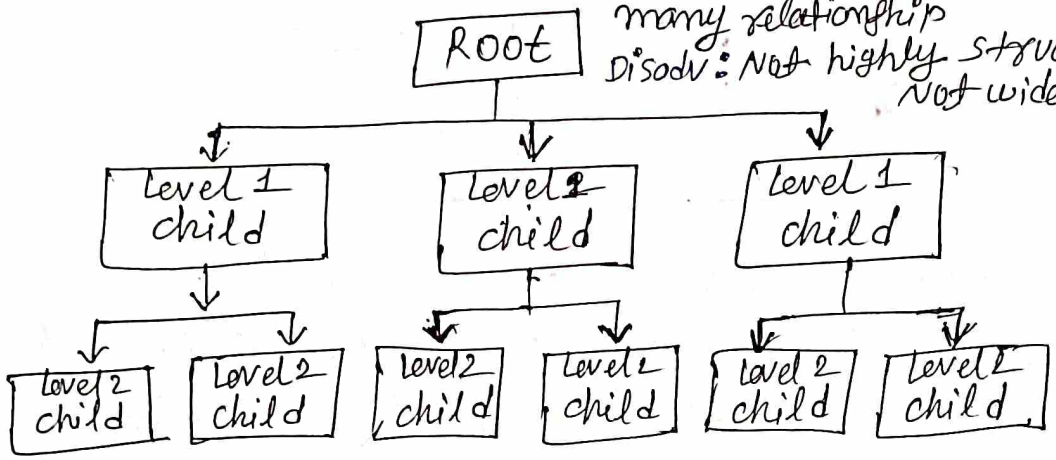
Ex → Table Student

Roll_No	Name	Age	SUB	Marks
1	Roushan	18	Math 1	95
2	Jay	36	Math 2	96
3	Aarumi	48	Math 3	97

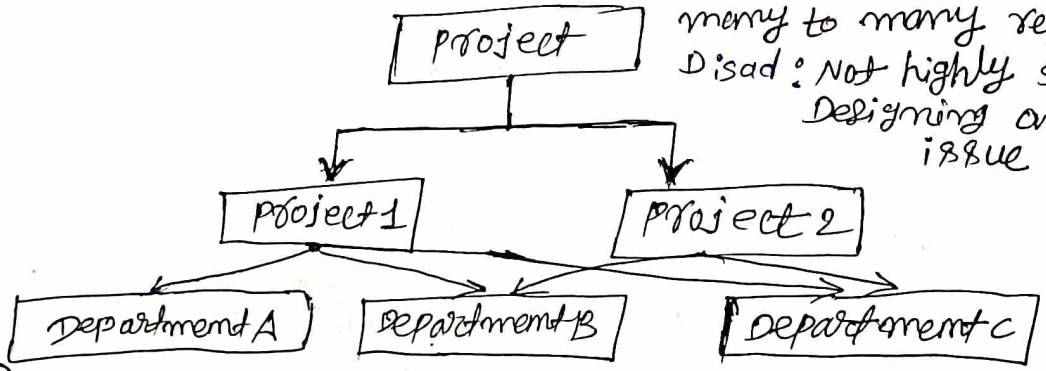
Advantage: Highly structured Data retrival using SQL query. data integrity

Disadvantage: Less efficient for non-tabular data, huge amount data

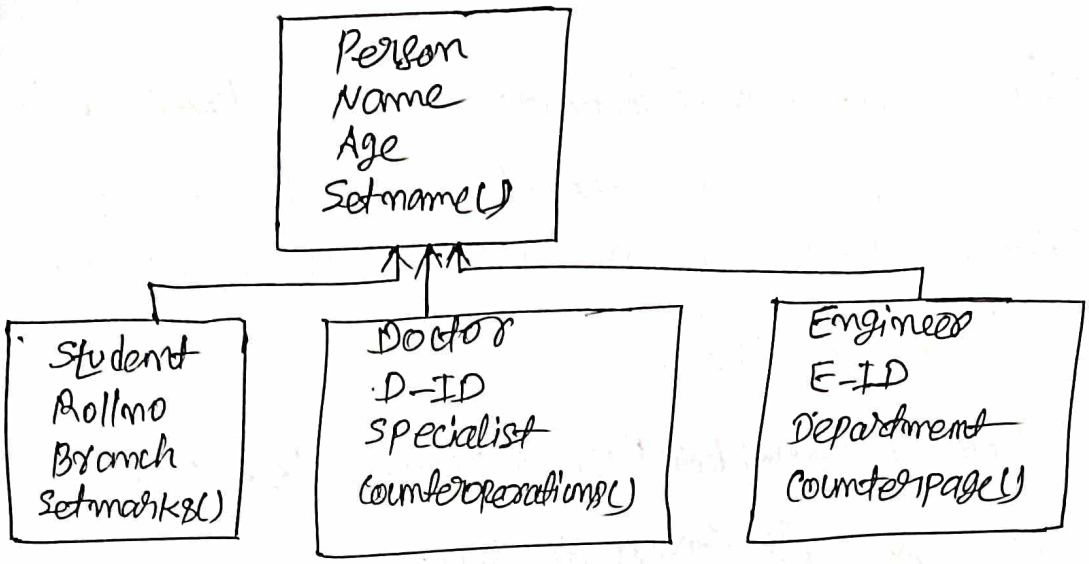
(ii) Hierarchical database model: Hierarchical database model organizes data in a tree-like structure, with parent-child relationship, Each parent can have multiple children, but each child has only one parent. This model is now not widely in use. Adv: efficient for representing one to many relationship
Disadv: Not highly structured, less flexible
Not widely in use.



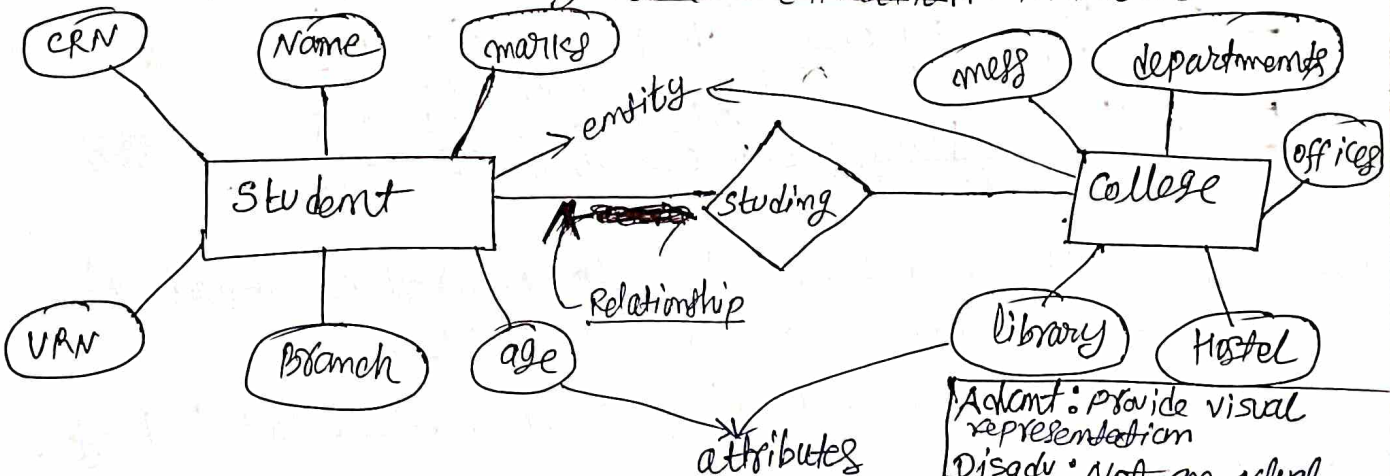
(iii) Network database model: Network database model organizes data in a graph-like structure, Network database model is the extended version of hierarchical model. In graph, nodes representing records and edges represents relationships between them. Advantage: efficient for representing many to many relationship
Disad: Not highly structured
Designing and implementation issue



(iv) Object oriented database model: Object oriented database model is the combination of object oriented programming and relational database model, In object oriented database model data is organized in object that represents real-world problem with different attributes.
Advant: efficient for large scale data.
Disadvant: Design and implementation issue.



(v) E-R database Model: Entity-Relationship (ER) model is a conceptual database model that represents relationships between entities. Entities are represented as rectangles, and Attributes of entities are shown inside ovals. Relationship b/w entities is represented by ~~line connection~~ Diamond.



Advant: Provide visual representation
Disadv: Not an actual database implementation model

Symbols used in ER Models

• Rectangles:		Entities in ER model
Ellipse		Attribute " " "
Diamond		Relationship among Entities
Line		Connection A to E & Es to others
Double Ellipse		Multi-valued Attributes
Double Rectangle		Weak Entity

* Explain keys and different types of keys with examples.

Definition: A key is an attribute or set of attributes that uniquely identifies any record (or tuple) from table.

Types of keys

① Super key ② Candidate key ③ primary key ④ Alternate key ⑤ foreign key ⑥ Composite key.

① Super key: A super key is a combination of all possible attributes that can uniquely identify the rows (or tuple) in the given table. super key is a superset of a candidate key.

Ex⇒

Emp_id	Name	Aadhar_No	Email_id	Dept_id
1	Rauthan	7766554433	abc@gmail.com	IT 1
2	Jay	1122334455	def@gmail.com	ECE 2
3	Rauthan	2233445511	ghi@gmail.com	II 2
4	Aarumi	9988665544	jkl@gmail.com	CE 3

fig: 1:1

Super keys:

① Emp_id
② Aadhar_No
③ Email_id
④ (Emp_id, Aadhar_NO)
⑤ (Aadhar_NO, Email_id)
⑥ (Emp_id, Email_id)
⑦ (Emp_id, Aadhar_NO, Email_id)
⑧ (Emp_id, Name)
⑨ (Emp_id, Name, Dep_id) etc.

② candidate key: A candidate key is derived from super key. A candidate key is an attribute or set of an attribute which can uniquely identify a record from table. candidate key is a minimal super key with no redundant attributes. in this Null values are not allowed. Ex⇒

Emp_id	Name	Aadhar_No	Email_id	Dept_id
1	Rauthan	7766554433	abc@gmail.com	IT 1
2	Jay	1122334455	def@gmail.com	EEE 2
3	Rauthan	2233445511	ghi@gmail.com	IT 2
4	aarumi	9988665544	jkl@gmail.com	CE 3

fig-1:1

Candidate keys
① (Emp_id)
② (Aadhar_No)
③ (Email_id)

③ primary key: primary key is one of the candidate key, chosen by the database designer, which can uniquely identify the record from the table. primary key cannot be NULL, cannot be changed, no updation and must be unique.

Calling figure 1.1

So primary is = Emp_id

④ Alternate keys: Out of all candidate keys, only one key gets selected as primary key and remaining keys are known as alternate keys.

Again calling figure 1.1

Candidate keys

① Emp_id (✓) as primary key
② Aadhar_NO
③ Email_id
→ alternate key

alternate keys: Aadhar_NO, Email_id

⑤ foreign key: A foreign key used to link two tables together, An attribute in one table that refers to the primary key in another table. foreign key may be NULL and redundant.

Employee table

Emp_id	Name	Aadhar_NO	Email_id	Dept_id
1	Rayhan	7766554433	abc@gmail.com	1
2	Jay	6655443322	def@gmail.com	2
3	Rayhan	5544332211	ghl@gmail.com	2
4	Aarumi	4433221100	jkl@gmail.com	3

→ foreign key

Department table

primary key →

Dept_id	Dept_Name
1	IT
2	ECE
3	CE

Foreign key : Dept_id

⑥ Composite key or Compound key

: A key that has more than one attributes is known as Composite key. It is also known as compound key.

Customer_id	order_id	product_code	product_count
C01	001	P111	5
C02	002	P111	8
C02	002	P222	6
C01	001	P333	9

Composite key: (cus_id, Product_code)

* Database languages

ans⇒ There are several types of database language

① DDL ② DML ③ DCL ④ TCL ⑤ DQL

① Data Definition Language (DDL): Data definition language language consists of the SQL commands that is used to define database schema. DDL is a set of SQL commands used to create, modify and delete database structures but not data.

DDL commands.

- CREATE: This command is used to create database
- DROP: This command is used to delete object from database
- ALTER: This command is used to alter the structure of the database.
- TRUNCATE: This command is used to remove all the records from database.
- RENAME: This command is used to ~~remove all the records from~~ an existing object of the database.
- COMMENT: This command is used add comments to the data dictionary.

(ii) Data Manipulation Language: Data manipulation language consists of the SQL Commands that is used manipulate the data in database. DML is a set of SQL commands used to insert, update, delete data from in database but not schema or structure.

- INSERT: This command is used to insert data into a table.
- DELETE: This command is used to delete data from table.
- UPDATE: This Command is Update data in table.
- Lock: Table control concurrency.

(iii) (DCL) Data Control Language: Data Control language consists of the SQL commands that is used to give rights, permissions and other controls of the database system. DCL is a set of SQL commands used to grant and Revoke of the database system.

List of DCL Commands

- GRANT: This command is used to give users access privileges to the database.
- REVOKE: This command is used to withdraws the users access.

(iv) Transaction control language: Transaction control language consists of the SQL commands that is used to control the execution of a transaction. TCL is a set of SQL commands used to commit, rollback, savepoint a transaction. List of TCL commands

(i) COMMIT: This command is used to commit a Transaction.

(ii) ROLLBACK: This Command is used to rollback a transaction in case of any error occurs.

(iii) SAVEPOINT: This command is used to set a save point whin a transaction.

(v) (DQL) Data Query Language: Data query language consists of the SQL commands that is used to retrieve information from database. DQL has only one SELECT command used to ~~retrive~~ retrieve data from table.

DQL command

- SELECT: This command is used to ~~retrive~~ retrieve data from the database.

* Database interfaces: There are 6 types of interfaces in DBMS.

① Menu-Based interfaces ② Form-based interfaces
③ Graphical User interfaces ④ Natural language interfaces
⑤ Interfaces for parametric users ⑥ Interfaces for DBA

① Menu-Based interfaces: Menu-Based interfaces present the user with lists of options called menus. The advantage of using menu-based interface is that they remove the tension of remembering specific commands and syntax of any query language.

② Form-Based Interfaces: Form-based interface present a form to the ~~user~~ users, user can fill out the form for database operations. It is widely used in web applications and desktop application.

③ Graphical User interface: Graphical User interface present a schema to the user in the form of diagram. most of the GUIs use a pointing device such as mouse to pick a certain part of the presented schema diagram

④ Natural language interfaces: Natural language interfaces accepts requests written in English or some other languages and attempt to understand them. this interface is still under development.

⑤ Interface for DBA: Most database system contain privileged tools that can be used only by the DBA's staff. this tools include creating accounts, setting system parameters, modify schemas.

⑥ Interface for parametric user: Interface for parametric users contain some commands that can be handled with a minimum of keystrokes. It is generally used in bank transactions for transferring money. These operations are performed repeatedly.

* ER- diagrams (6.1)

(Q5) What do you mean by Entity Relationship diagram and why it is useful? Draw E-R diagram for hospital with the set of patient and medical doctors?

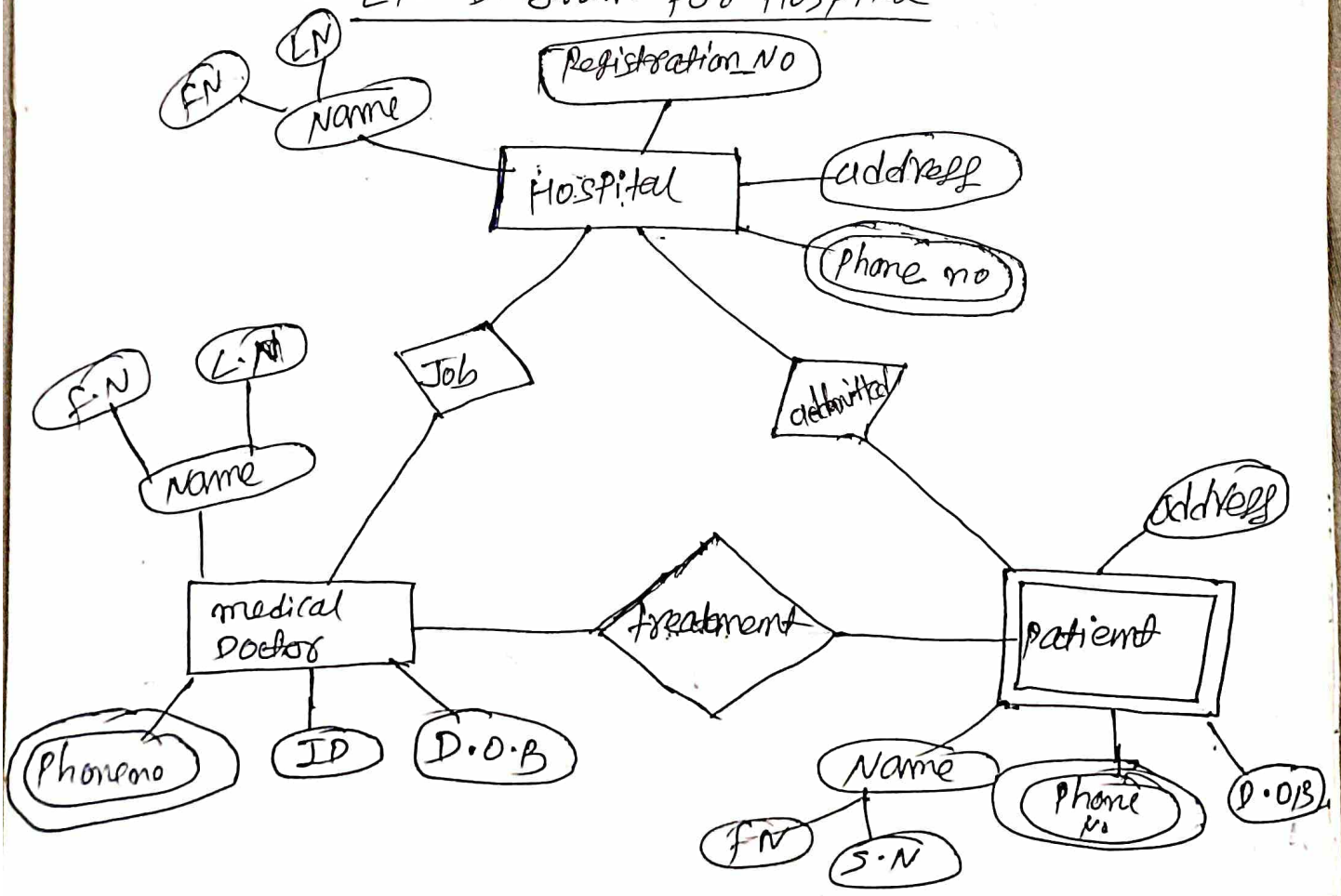
ans: The Entity Relationship Diagram explains the relationship among the entities present in the database

Useful:

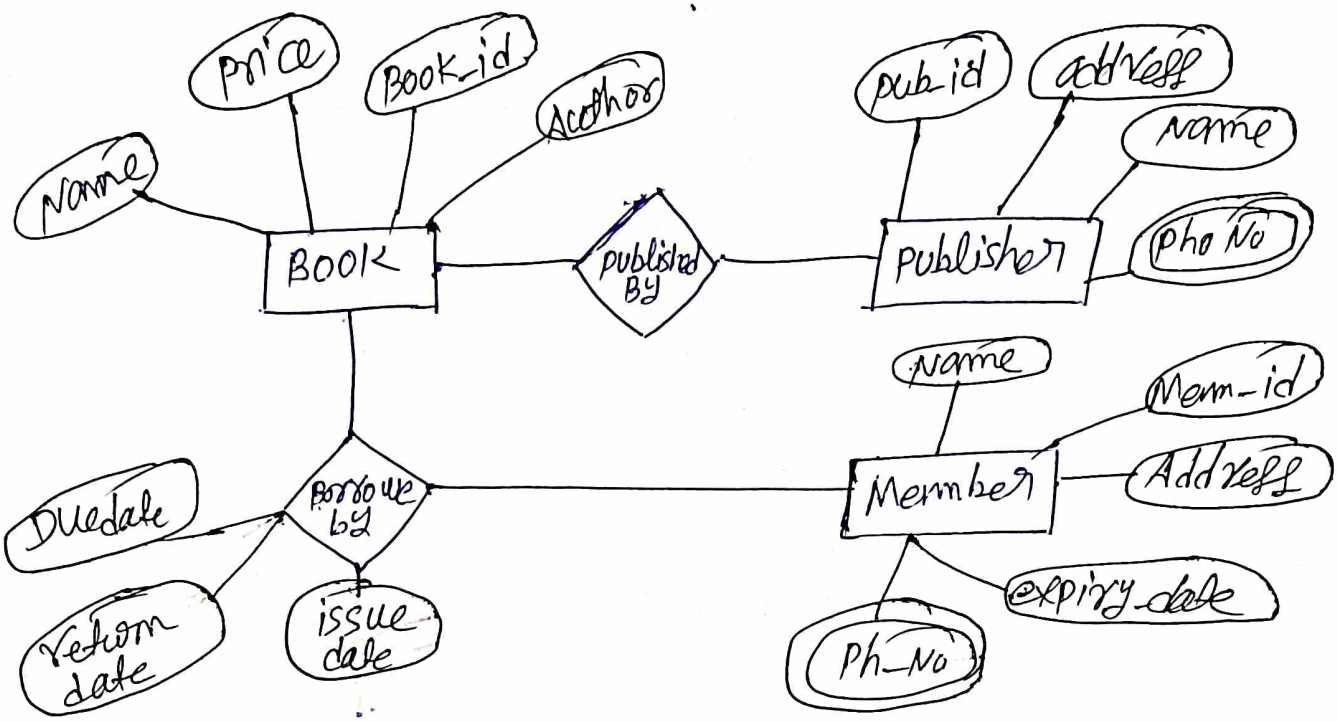
ER diagram is useful for represent the E-R-model in a database, which makes them easy to convert into tables (Relations).

ER diagram → Hospital
→ Student Enrollement system
→ Library management system
→ Banking system
→ College, school

ER - Diagram for Hospital

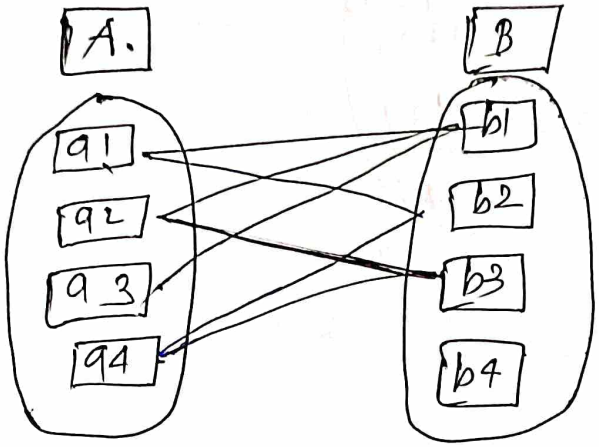


Q) Design an ER diagram for Library management system, Take "Books", "publisher", "Member" and "borrowed by" as entity.

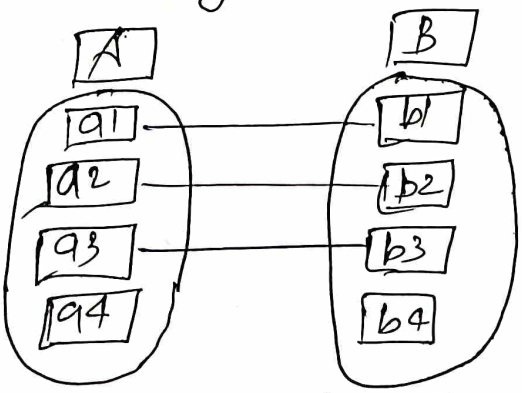


Q3) Give an example of following relationships: (a) many-to-many (b) one-to-one (c) one-to-many (d) many-to-one

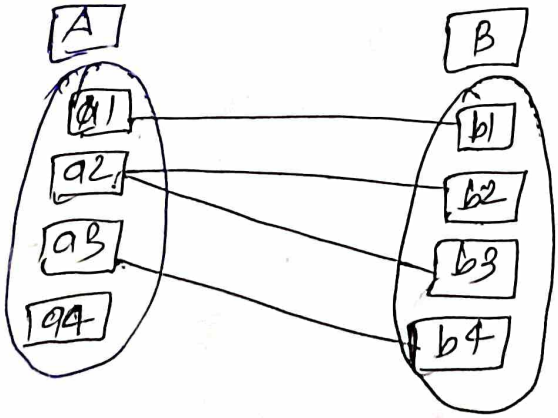
ans → (a) many-to-many Example: Let us assume that a student can take more than one course and one course can be taken by many students so the relationship will be many to many.



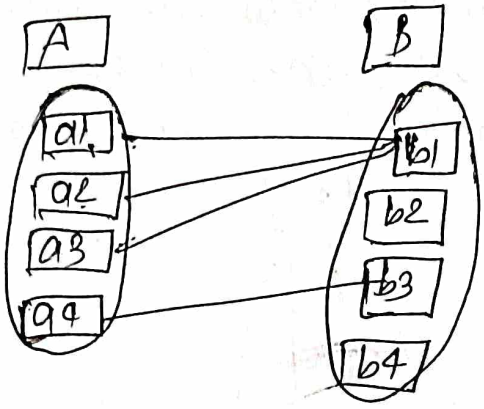
(b) one to one: Let us assume that a male can marry one female and a female can marry one male. So the relationship will be one to one.



(c) one to many: Let us assume a college have many-many different professors. So the relationship will be one to many.



(d) Many to one : Let us assume that a student can take only one course but one course can be taken by many students. So the relationship will be many to one.



Unit-2

* difference b/w DBMS and RDBMS

DBMS	RDBMS
(i) DBMS stands for Database management system.	(i) RDBMS stands for Relational Database management system.
(ii) It stores data in a hierarchical form or navigational form as files.	It stores data in tables and table consist row and column.
(iii) Normalization cannot be performed.	Normalization can be performed.
(iv) There is high data redundancy	There is low data redundancy.
(v) provides less security for data	provides more security for data
(vi) Difficult to modify data	Easy to modify data.
(vii) No keys and indexes.	Has key and indexes.
(viii) suitable to store a small amount of data	suitable to store a large amount of data.
(ix) Transaction is less secure.	Transaction is more secure.
(x) Ex: Microsoft Access, Libra office.	Ex: MYSQL, oracle

* Properties of Relational table

(1) Values are Atomic: The columns in a relational table are not repeating.

(2) Column values are of the same kind: Means All values in a column come from the same domain.

(3) Each Row is unique: There is no two rows in a relational table

(4) The sequence of columns is insignificant: Means the order of columns in relational table has no meaning.

(5) The sequence of Rows is insignificant: Means the order of rows in relational table has no meaning.

⑥ Each column has unique name: Inside a table column name should be unique. *

* Advantage & disadvantages of RDBMS.

Advantage:

① Data structure: They organize data into tables with rows and columns.

② Data Integrity: They enforce data integrity through constraints.

③ Data security: They offer security mechanisms to protect data.

④ Data Relationships: They enable relationship b/w tables using primary and foreign key.

⑤ Data indexing: They use indexing techniques to optimize data retrieval.

⑥ Data independence: This allows developers to modify the database schema without affecting application built.

⑦ Data accessibility: They provide concurrent access to multiple users and applications.

⑧ Data Backup and Relovery: They offer various backup and recovery options.

⑨ Scalability: They can handle large amount of data and scale up as growth of the dataset.

⑩ Querying and Reporting: They provide powerful query languages that allows user to retrieve specific data from the database quickly.

* Disadvantage: (i) scalability limitations
(ii) complex design
(iii) fixed schema
(iv) Data Modeling challenges
(v) concurrent issue
(vi) Machine learning process.

* Define Domain constraints

Domain constraints are the set of rule which defines what kind of attributes can be stored in an entity in database table.

* Define Trigger: Trigger is a set of procedural code that is automatically executed ("triggered") in response to certain events on a particular table.

* Key constraints: Key constraints are regulations that a DBMS uses to ensure data accuracy and consistency in a database.
other words → They defines how the value in a table's one or more columns are related to the values in other table.

* 12 Codd's rule in DBMS: Codd's rules are proposed by a computer scientist name Dr. Edgar F. Codd.

Rule 0: The Information rule: All information is to be represented as stored in cells of tables.

Rule 2: The Guaranteed Access Rule: Each data element is guaranteed to be accessible logically with a combination of the table name, primary key and attribute name.

Rule 3: Systematic Treatment of NULL values: Every a NULL value in a database must be given a systematic and uniform treatment.

Rule 4: Active online Catelog Rule: The database catelog, which contains metadata about database, must be stored and accessed using the same relational database management system.

Rule 5: The comprehensive data sublanguage rule : One well defined language must be there to provide all facilities related to data like defining querying, modifying information within the database. Ex ⇒ SQL

Rule 6: The view updating Rule: All views that are theoretically updateable must be updatable by the system.

Rule 7: High-level Insert, Delete and Update: There must be support high-level insert, delete and update operations at each level of relations.

Rule 8: physical data independence: Application programs and activities should remain unaffected when changes are made to the physical storage.

Rule 9: Logical data Independence: Application programs and activities should remain unaffected when changes are made to the logical structure of the database such as adding or modifying tables.

Rule 10: Integrity independence: Database should be able to emforce integrity constraints its own rather than using other programs.

Rule 11: Distribution independence: The Distribution of data across multiple locations should be individual to users and the database system should handle the distribution transparently.

Rule 12: Non-Subversion Rule: ~~If low level access is allowed to a system it should not be able to bypass integrity rule to change data~~. The system should not allow users to bypass the security and integrity constraints using low-level access methods.

* Describe relational algebra

ans ⇒ Relational Algebra is a procedural query language, it provides a theoretical foundation for relational database. The main purpose of using Relational Algebra is to define operators that transform one or more input relations into an output relation.

Types of Operators

* Basic
1. Selection (σ).
2. projection (π).
3. union (∪).
4. Set difference (−).
5. cartesian product (×).
6. Rename (ρ)

* Derived
7. Natural join (⋈).
8. set intersection (∩).
9. division (÷)

1. Selection : It is used to select required row of the table it is denoted by σ.

Ex→

roll_No	name	age
1	A	20
2	B	17
3	C	16
4	D	19
5	E	18

Query → select students whose age is greater than 17.

σage>17 (student)

roll_No	name	age
1	A	20
4	D	19
5	E	18

2. projection (π) : It is used to select required column data of the table, it is denoted by π, By default projection removes duplicate data.

Ex→ Query → Display the roll_No and name of the student

Student

roll_No	name	age
1	A	20
2	B	17
3	C	16
4	D	19
5	E	18
6	F	18

$\Pi_{rollno, name}$ (student)

roll_No	name
1	A
2	B
3	C
4	D
5	E
6	F

③ union (U): union operation in relational algebra is the same as union operation in set theory. it is denoted by 'U'. Query → Π_{Name}(student) U Π_{Name}(Employee)

Student

RollNo	Name
1	A
2	B
3	C
4	D

Employee

Emp_No	Name
2	B
8	G
9	H

→

Name
A
B
C
D
G
H

Note: union is commutative $A \cup B = B \cup A$

④ Set intersection ($\cap$): Set intersection in relational algebra is the same as set intersection operation in set theory. It is denoted by $\cap$, union is commutative $A \cap B = B \cap A$

Student

Roll_No	Name
1	A
2	B
3	C
4	D

Employee

Emp_No	Name
2	B
8	G
9	H

Query →

Π_{Name}(student) $\cap$ Π_{Name}(Employee)

→

Name
B

(5) Set difference (−): Set difference in relational Algebra is the same as set difference in set theory. It is denoted by (−). It is not commutative $A - B \neq B - A$

Student

Roll-No	Name
1	A
2	B
3	C
4	D

Employee

Emp-No	Name
2	B
8	G
9	H

Query → $\Pi_{Name}(Student) - \Pi_{Name}(Employee)$

→

Name
A
C
D

(6) Cartesian/cross product: Cartesian product combines information of two different table into one. it is also called cross product, It is denoted by ×

Query → $\sigma_{A=C}(R1 \times R2)$

R1

A	B
α	1
β	2

R2

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

→

R1 × R2

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

(7) Natural join (⋈): Natural join, join between two or more tables and result will be a set of all combinations of row where they have an equal common attribute denoted by (⋈).

EMP

Name	ID	Dept-Name
A	120	IT
B	125	HR
C	110	Sales
D	111	IT

DEPT

Dept_Name	Manager
Sales	Y
Production	Z
IT	A

→

EMP.Dept_Name = Dept.Dept_Name

Name	ID	Dept_Name	Manager
A	120	IT	A
C	110	Sales	Y
D	111	IT	A

② Conditional join : Conditional join work similarly to natural join but in conditional join we specify any condition such as greater than, less than or not equal.

R

ID	Sex	Marks
1	F	45
2	F	55
3	F	60

S

ID	Sex	Marks
10	M	20
11	M	22
12	M	59

join R and S with condition R.marks >= S.marks

R.ID	R.Sex	R.Marks	S.ID	S.Sex	S.Marks
1	F	45	10	M	20
1	F	45	11	M	22
2	F	55	10	M	20
2	F	55	11	M	22
3	F	60	10	M	20
3	F	60	11	M	22
3	F	60	12	M	59

Relational Database Design

(PYQ) Discuss the concept of functional dependencies and their role in database management & use.

ans⇒ functional dependency is a concept ~~that~~ specifies the relationship between ~~the value of~~ two sets of attributes. it is denoted as $x \rightarrow y$, where the attribute set on the left side of the arrow is called Determinant and y is called the Dependent.

Ex⇒

roll_NO	name	dept_Name	building
1	ABC	IT	A4
2	DEF	CSE	A2

roll_NO → {name, dept_name, building}

here roll no can determine name, dept_Name and dept_building hence a valid functional dependency.

use|role → it helps in organizing data efficiently, minimizing redundancy, a process that reduce data redundancy by organizing tables.

uses → Database Design, Normalization, Data integrity, Query optimization

Q. what do you mean by multivalued dependency

ans⇒ when one attribute in a database depends on another attribute and has many independent values, it is said to be multivalued dependency.

Person (P)	Mobile (M)	Food (F)
P1	M1	f1
P1	M2	f2
P1	M1	f2
P1	M2	f1
P2	M3	F2

(P →→ M)

(P →→ F)

P	M	F
P1	M1, M2	f1, f2
P2	M3	f2

Q Explain join dependency: If we join each individual relation projection (π) and at end/last if we get original relation then it will called as join dependency.

$$\pi_{R1}(R) \bowtie \pi_{R2}(R) \bowtie \text{-------} \bowtie \pi_{Rn}(R) = R$$

R

pName	Skill	Job
Aman	DBA	J1
Mohan	Tester	J2
Rohan	Programmer	J3
Sohan	Analyst	J1

R1 (Pname, Skill)

Pname	Skills
Aman	DBA
mohan	Tester
Rohan	Programmer
sohan	Analyst

R2 (Pname, Job)

pname	Job
Aman	J1
mohan	J2
Rohan	J3
sohan	J1

Join (R1 ⋈ R2)

Pname	Skills	Job
Aman	DBA	J1
mohan	Tester	J2
Rohan	Programmer	J3
sohan	Analyst	J1

R3 (Skills, Job)

Skills	Job
DBA	J1
Tester	J2
Programmer	J3
Analyst	J1

Again join ((R1 ⋈ R2) ⋈ R3)

pname	Skills	Job
Aman	DBA	J1
mohan	Tester	J2
Rohan	Programmer	J3
sohan	Analyst	J1

We get original relation

$$(R1 \bowtie R2 \bowtie R3) = R$$

join dependency

(PYQ) Explain All forms of Normalization with relevant Example.

Normalization: It is a process of organizing data in the database.

① 1NF (First Normal form): Table should not contain any multivalued Attribute.

Roll	Name	Course
1	Raufhan	C/C++
2	Aarumi	Java
3	Pawan	C/DBMS

→

Roll	Name	Course
1	Raufhan	C
1	Raufhan	C++
2	Aarumi	Java
3	Pawan	C
3	Pawan	DBMS

Primary key → (Roll & Course)

* 2NF: (i) Table must be in 1st Normal form

(ii) All the non-prime attributes should be fully functional dependent on candidate key.

or should Not partial functional dependency

Example:

R

A	B	C
a	1	x
b	2	y
[redacted]		
c	3	z
d	3	z
e	3	z

→ R1

A	B
a	1
b	2
[redacted]	
c	3
d	3
e	3

→ R2

B	C
1	x
2	y
3	z
[redacted]	

Candidate key (A, B)

Prime Attribute → (A, B)

Non prime Attribute (C)

R(A, B, C, D)

3NF : (i) Table must be in second Normal form
(ii) There should not transitive dependency

Transitive dependency means → when a non prime attribute determine another non prime attribute

(A B C) $A \rightarrow B$ fine(✓)

where

A is Prime Attribute $B \rightarrow C$ Not fine(✗)

B → non prime attribute; C → non prime Attribute

Prime Attribute
A

R

A	B	C
a	1	x
b	1	x
c	1	x
d	2	y
e	2	y
f	3	z
g	3	z

→

R1

A	B
a	1
b	1
c	1
d	2
e	2
f	3
g	3

R2

B	C
1	x
2	y
3	z

BCNF : (i) The table should be in the 3rd normal form

Rule 2 : $x \rightarrow y$

-x should be a superkey for every functional dependency

R(A B C)

AB
AC

R1(C, B)

R2(A, C)

A	C
a	x
b	y
c	z
c	w
d	w
e	w

C	B
x	1
y	2
z	2
w	3

R

A	B	C
a	1	x
b	2	y
c	2	z
c	3	w
d	3	w
e	3	w

Q. How BCNF is different from 3NF or difference b/w BcNf and 3NF

BCNF	3NF
(i) BCNF stands for Boycee-Codd normal form.	3NF stands for third Normal form.
(ii) In BCNF for any relation A→B, A should be a super key	(ii) In 3NF for any relation A→B, A should be a prime Attribute.
(iii) It is comparatively more stronger than 3NF.	(iii) It is less stronger than BCNF.
(iv) In BCNF two main condition (a) table should be in 3NF (b) x→y x should be a super key for every functional dependency.	(iv) In 3NF two main condition (a) table must be in second NF. (b) There should not transitive dependency.
(v) The redundancy is comparatively low in BCNF.	(v) The redundancy is high in 3NF.
(vi) Difficult to achieve	(vi) Easy to Achieve.
(vii) Lossless decomposition is hard to achieve in BCNF.	(vii) Lossless decomposition can be achieved by 3NF.

* 4NF :- (i) 4NF → table should be in BCNF.
(ii) No multivalued dependency $x \to\to y$

* Multivalued dependency: when one attribute in a database depends on another attribute and has many independent value, it is said to be multivalued dependency.

Person(P)	Mobile(M)	Food(F)
P1	M1	F1
P1	M2	F2
P1	M1	F2
P1	M2	F1
P2	M3	F2

P	M	F
P1	M1, M2	F1, F2
P2	M3	F2

(P →→ M)
(P →→ F)

(Project Join normal form)

* 5NF : 5NF is also known as project - join Normal form.
  (i) It must be in 4NF
  (ii) Won't have lossless decomposition into smaller tables

Join dependency : If we join each individual relation projection (σ) and at end/last if we get original relation then it will called as join dependency.

$$\pi R_1(R) \bowtie \pi R_2(R) \bowtie \cdots \bowtie \pi R_n(R) = R$$

R

PName	Skills	Job
Aman	DBA	J1
Mohan	Tester	J2
Rohan	programmer	J3
sohan	Analyst	J1

R1(Pname, skill)

PName	Skills
Aman	DBA
mohan	Tester
Rohan	programmer
Sohan	Analyst

R2(Pname, job)

Pname	Job
Aman	J1
Mohan	J2
Rohan	J3
sohan	J1

Join (R1 ⋈ R2)

Pname	Skills	Job
Aman	DBA	J1
mohan	Tester	J2
Rohan	programmer	J3
sohan	Analyst	J1

R3(Skills, job)

Skills	job
DBA	J1
Tester	J2
programmer	J3
Analyst	J1

Again join((R1 ⋈ R2) ⋈ R3)

Pname	Skills	job
Aman	DBA	J1
mohan	Tester	J2
Rohan	Programmer	J3
sohan	Analyst	J1

$R1 \bowtie R2 \bowtie R3 = R$

join dependency

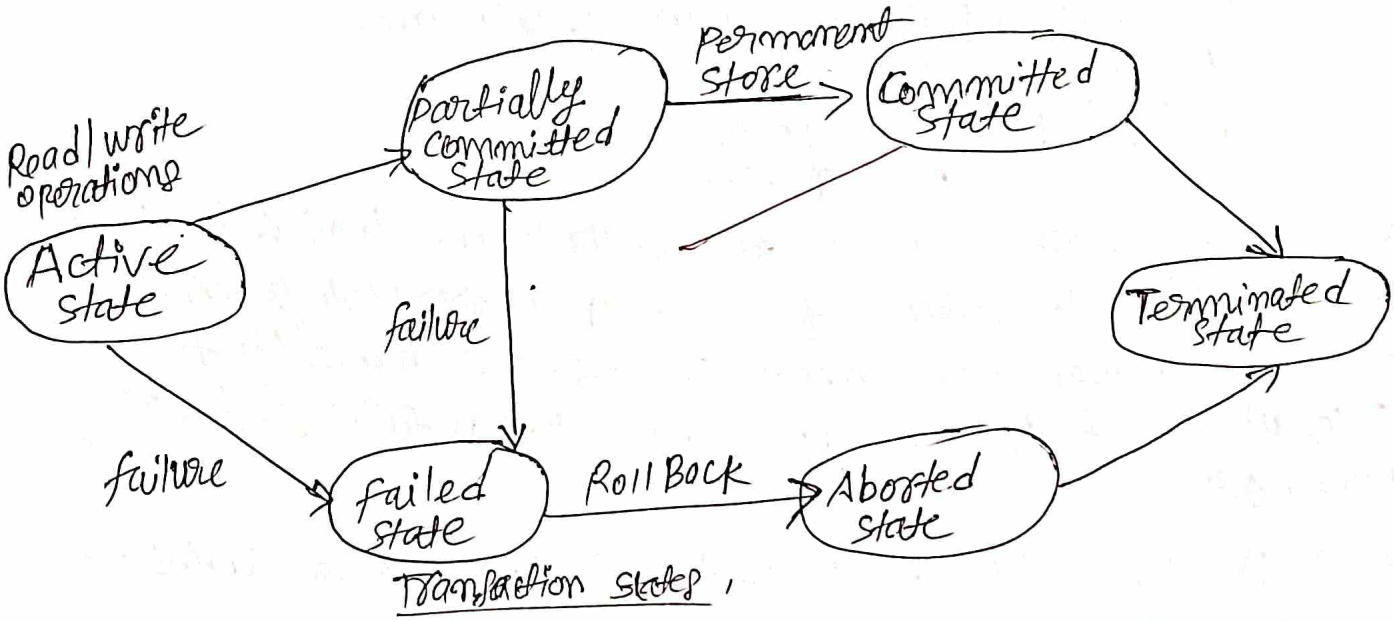
We get original relation

(10) Mtz

Unit → Transaction Management and Concurrency Control

(PYQ) Explain transaction and draw a state diagram of transaction showing its state. Explain ACID properties of a transaction.

Ans ⇒ Transaction: Transaction is a set of logical operations used to perform ~~a logical unit of work~~ on database. These operations typically involve modifying the database's data such as inserting new data, updating existing data or deleting data.



Transaction states.

ACID Properties:

(i) Atomicity: This properties ensures that either all the operations within a transaction are successfully completed or none of them are completed. If any part of the transaction fails, the entire transaction is rolled back.

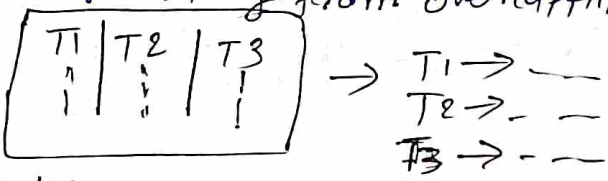
(ii) Consistency: Consistency ensures that Before transaction start and after transaction completed sum of money should be same.

T1
R(A) 2000
A = A - 100
W(A) 1000
R(B) 3000
B = B + 1000
W(B) 4000
commit

Before Tran Database
A = 2000
B = 3000
Sum of A&B = 5000

After Tran
A = 1000
B = 4000
sum of A&B = 5000

(iii) Isolation: Concurrently transactions execution at the same time must be isolated from each other, preventing any inconsistencies arising from overlapping data access.



(iv) Durability: This property ensures that the changes made to the database after a transaction is completely executed are durable. means permanent changes are made by the succesful exe of transaction. even if system failures occur such as power off.

(PYQ) Analyze various recovery techniques used in database management system.

There are mainly two types of recovery techniques used in DBMS. (i) Rollback/undo Recovery Technique (ii) Commit/Redo Recovery Technique (iii) Checkpoint Recovery (iv) Shadow paging

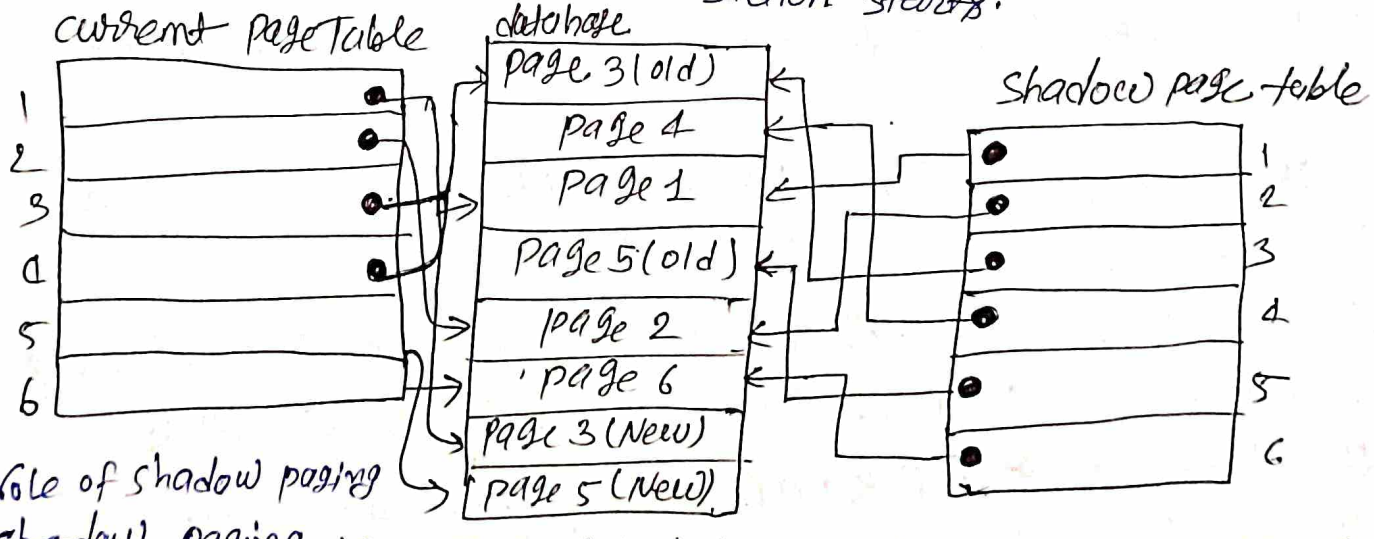
(i) Rollback: This technique is accomplished by undoing the changes made by the failed transaction using log records stored in the transaction log. The transaction log contains a record of all the transaction that have been performed on the database. system uses the log records to undo the changes made by the failed transaction and restore the database to its previous state.

(ii) Commit: This technique is accomplished by using the log records stored in the transaction log to redo the changes made by the transaction that was in progress at the time of the failure. The system uses the log records to reapply the changes made by the transaction and restore the database to its most recent consistent state.

(iii) Checkpoint Recovery: This technique is used to reduce the recovery time by saving the state of the database in a checkpoint file. In the event of a failure, the system can use the checkpoint file to restore the database to the most recent state before the failure occurred, rather than going through the entire log to recover the database.

(iv) Shadow paging: shadow paging uses two page tables
(i) current page table (ii) Shadow page table

(i) current page table: current page table points to the most recent versions of the pages on disk.
(ii) Shadow page table: It is created as a copy of the current page table when a transaction starts.



Role of shadow paging

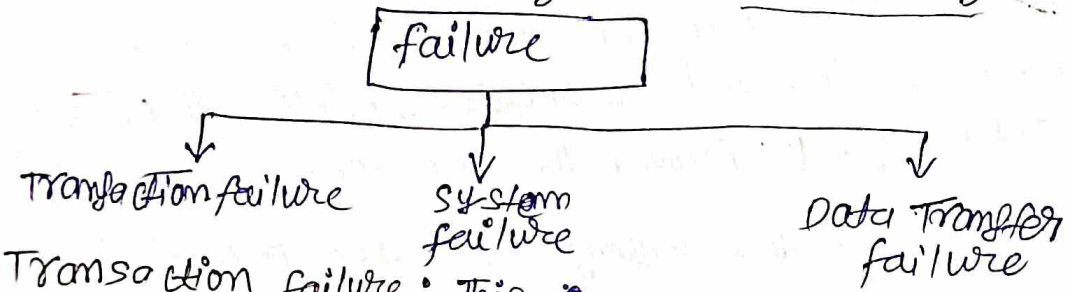
~~Shadow paging is a recovery technique that facilitate in case of~~ failure.

(1)

* role of shadow paging

Ans ⇒ The primary role of shadow paging is to provide a mechanism for recovery technique. It is achieved by maintaining a shadow copy of the page table, which allow for atomic updates and rollback of changes. shadow paging helps improve system performance by reducing the overhead associated with traditional disk-based recovery mechanism.

(Q15) Explain the different types of database failure ~~with advantages and disadvantages~~. Discuss the role of recovery and atomicity in ensuring data consistency.



(i) Transaction failure: This occurs when a transaction is unable to complete its operations due to reasons such as system crashes, power outages or network failures.

(ii) system failure: This occurs when there is some sort of hardware or software. some other problems which are external to the system which leads to system failure such as operating system errors, power cuts, main memory crash.

(iii) Data-Transfer failure: When a disk failure occurs during data-tranfer operation it result into loss of content from disk storage then such failure are categorized as data-transfer failure. some other reason for disk failure disk head crash, disk unreachability, read-write errors on the disk etc.

Role of recovery and atomicity

- Recovery: Recovery mechanisms in a database system are designed to restore the database to a consistent state after a failure. This involves techniques such as logging changes made by transactions, maintaining checkpoints and performing database backups.
- Atomicity: This property ensure that either all the operations, within a transaction are successfully completed or none of them are completed. If any part of the transaction fails, the entire transaction is rolled back.

PYQ why concurrency control is important in DBMS.

ans ⇒ Concurrency control is a crucial Database management system component. it manages simultaneous operations without conflicting them with each other. The primary aim is maintaining consistency, integrity, and isolation when multiple users access the database simultaneously.

PYQ what is database recovery: It is the process of restoring the database to a correct (consistent) state in the event of a failure. The primary goal of database recovery is to maintain data integrity and consistency.

PYQ what is Database recovery? Explain different types of recovery techniques with advantages and disadvantages.

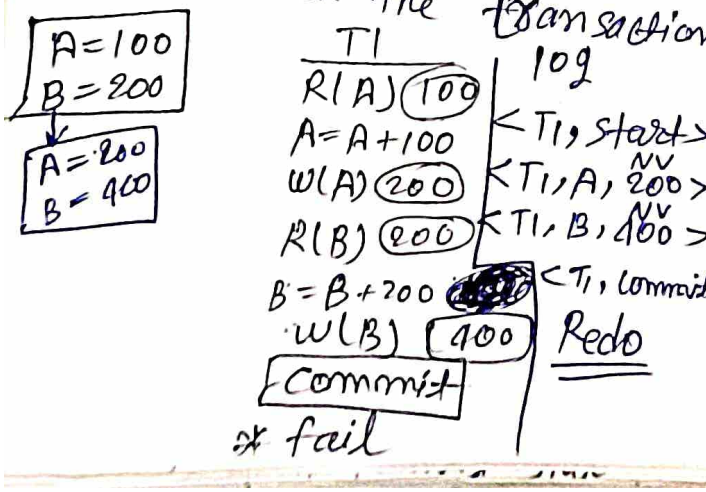
ans ⇒ (i) log based recovery (ii) shadow paging
(iii) checkpoints recovery (iv) Transaction rollback (v)
(v) Backup and restore (vi) Database replication

(i) Log based recovery: In log based recovery technique log is used, log contains sequence of records, each record refers to a write operations. We can say log file store history of all updates. log file contains transaction number, record number old value, new value, end of transaction etc.

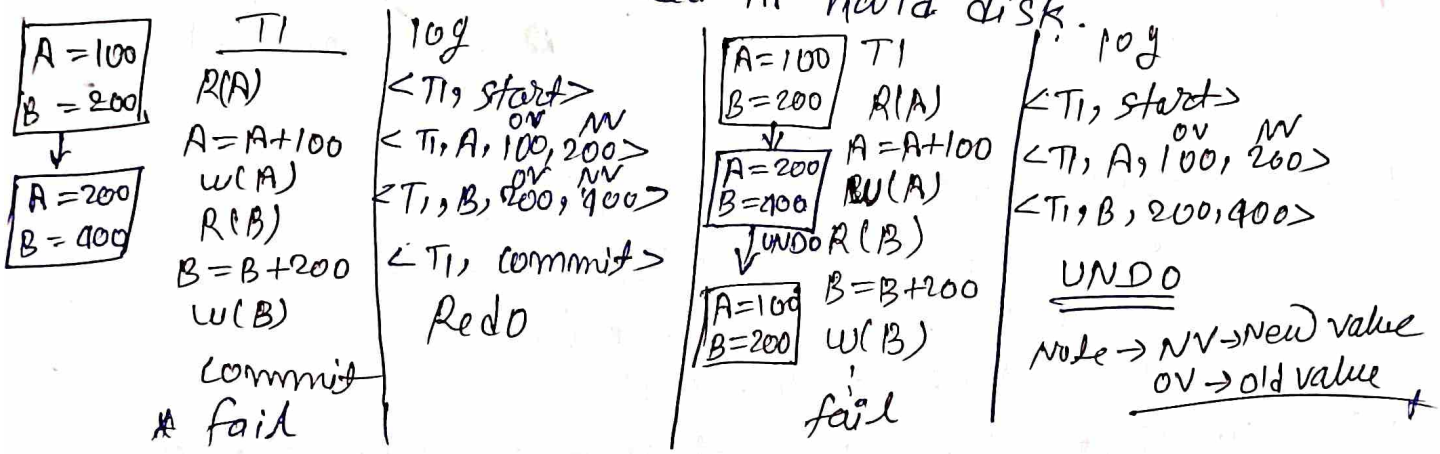
Ex ⇒ <Ti, Xi, V1, V2>

log based recovery use two techniques
(i) Deferred database modification Redo/No-undo technique
(ii) immediate database modification Redo/undo technique

(i) Deferred database modification: It is a recovery technique It is also called NO-UNDO/REDO technique. if Ti transaction is start and performed read and write operation them after commit all the changes that are performed in RAM that will be updated in hard disk. if after commit any failure gets occure then transaction manager will Redo the transaction means the new value is overrided in hard disk but before commit if any failure gets occure the transaction manager will rollback the transaction

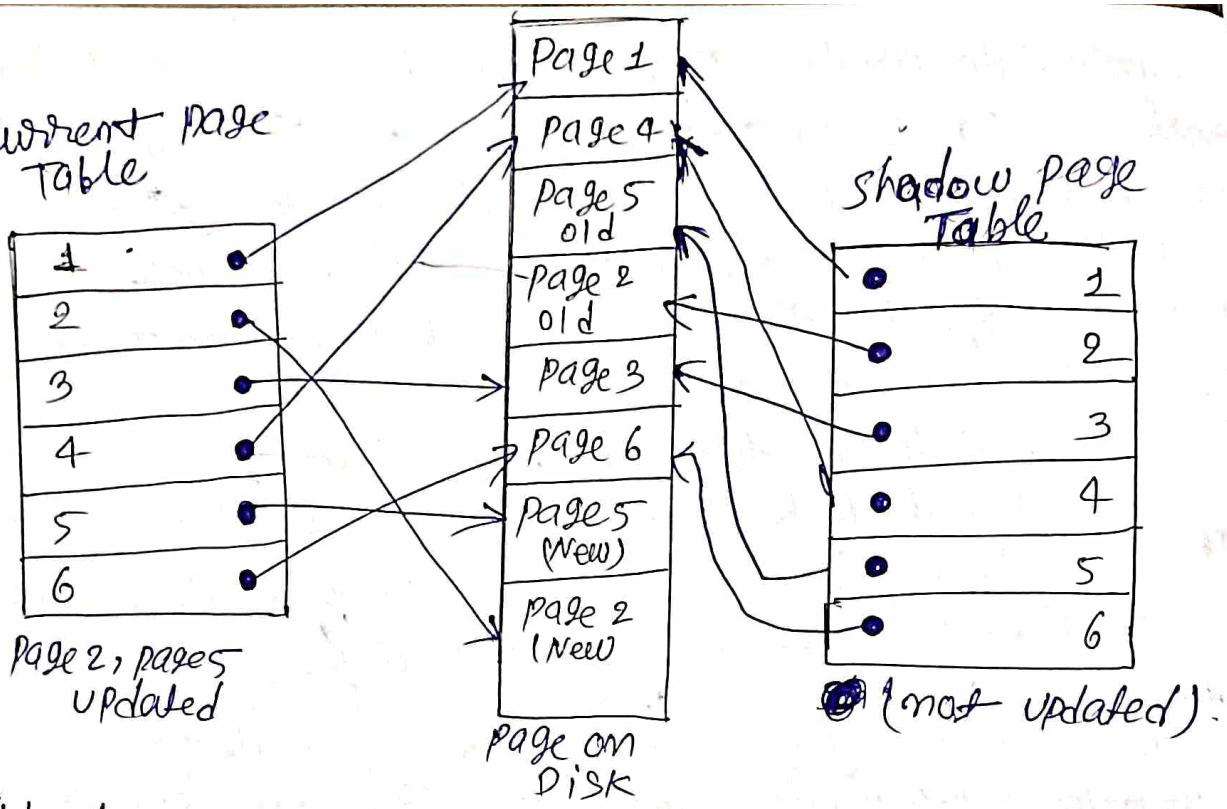


(ii) Immediate Database Modification: It is a recovery technique It is also called UNDO/REDO technique. If T1 Transaction is start and performed read and write operation the moment write operation is completed them all the changes will be reflected directly in the hard disk before commit and if any failure occure after commit the transaction manager will redo the tranfaction means new value is overrided in hard disk but if any failure occure before commit then transaction manager will UNDO the tranfaction mean old value is overrided in hard disk.



Advantages • faster recovery • Durability • low risk of data corruption
disadvantages → • complexity • Time consuming • reduce performance of system

(2) shadow paging: Shadow paging is a recovery technique. shadow paging uses two page tables
(i) current page table (ii) shadow page table
(i) current page table: This page table points to the most recent versions of the data pages on the disk.
(ii) shadow page table: This is a copy of current page table created at the start of a transaction.



Advantages: • It requires fewer disk access
• recovery is fast • less expensive
• No need of undo and redo
• No need of log file

Disadvantages: • Data fragmentation • Garbage collection • Performance overhead • Limited controll

③ checkpoint recovery: This technique is used to reduce the recovery time by saving the state of the database in a checkpoint file. In the event of a failure, the system can use the checkpoint file to restore the database to the most recent state before the failure occured, rather than going through the entire log file to recover the database.

~~④ Rollback~~ Advantage → • reduce recovery time • enhance the consistency of the database

disadvantages: It is used only in logical error | It cannot restore of media failure

④ Rollback: This technique is accomplished by undoing the changes made by the failed transaction using log records. log file Transaction Containing a record of all the transaction that have been performed in database.

system uses the log records to undo the database to its previous state.

Advantages: • Reliability • Data consistency • cost effective

Disadvantages: • Complexity • Time consuming • reduce performance

(5) Backup and restore: This is the most basic recovery technique. it involve creating a copy of the database at regular intervals. in case of a failure, the database can be restored from the latest backup.

Advantages: • simple to implement • low cost

Disadvantages: data loss if the backup is not recent.

(6) Database Replication: This technique involves maintaining copies of the database on multiple servers. if one server fails, the database can continue to operate using remaining copies.

Advantages: • High availability • faster recovery

Disadvantages: • Increased cost • Complexity

(Pyq) Contrast log based recovery and cascading rollback

log-Based recovery	cascading Rollback
(i) It purpose is to ensure database consistency, durability by maintaining log file.	It purpose is to maintain data integrity by rolling back changes in related tables.
(ii) It uses the transaction log to redo or undo. transactions during system failure.	It identifies and rolls back changes made in related tables to maintain consistency when a transaction fails. It uses
(iii) It uses Transaction log which records all modifications made to the database.	(iii) Foreign key constraints defined in the database schema.
(iv) It operates at the level of individual transactions.	(iv) It operates at the level of all transactions.
(v) Less Complex	(v) More complex
(vi) low risk of data loss	(vi) High risk of data loss.
(vi) More consistent	(vii) less consistent.

(Q18) Define TPS : TPS stands for Transaction Processing System for business transactions involving the collection, modification and retrieval of all transaction data.

(Q19) What are the fields used in log based recovery.

(i) Transaction ID (TID)
(ii) Operation TYPE (insert, update, delete)
(iii) Data item ID
(iv) Old value
(v) New value
(vi) Timestamp
(vii) Commit
(viii) Redo/Undo
(ix) check-point information

(Q20) suppose that there is a database system that never fails, Analyze a recovery manager required for this system

Ans → Even if a database system that never fails, a recovery manager is still necessary for several reasons.

(1) Human error (2) Software bugs (3) Data corruption
(4) Data validation (5) Disaster recovery.

(pyq) How many timestamps are associated in validation based protocols.

ans → Three timestamps

(i) start (Ti): This timestamp records the time when transaction Ti begins its execution.

(ii) ~~Val~~ validation: This timestamps captures the time when Ti finishes reading data and enters the validation phase.

(iii) Finish: This timestamps marks the completion of Ti's write phase.

(pyq) Define TPS: TPS stands for transaction processing system. A transaction processing system is a software or application that is designed to handle large amount of transaction operations like business transaction that involve database management concurrently ~~and~~ in a reliable and secure manner.

NOSQL Database

M
(PYQ) How you Evaluate NOSQL, explain NOSQL database along with cafe study of Metlife, facebook, google.

or

Define the term NOSQL with example Analyze why a NOSQL database is used by facebook and google applications.

ans ⇒ Evaluation of NOSQL

① Data Model: NoSQL databases offers various data models such as document, key-value, column-family and graph.

② Scalability: NOSQL database are designed to scale Horizontally allowing them to handle large volumes of data.

③ Data integrity and Durability: NO-SQL databases offer mechanisms for ensuring data integrity and durability through features like replication, backups and automatic recovery

④ Consistency Model: NOSQL offer different consistency models such as eventual consistency, strong consistency.

NOSQL: The term NoSQL referred to "non-SQL", "non-relational", ("Not only SQL") database. NOSQL is a type of database management system that is designed to handle and store large volumes of unstructured and semi-structured data unlike relational database that use tables with pre-defined schema to store data, NOSQL database use flexible data models that can adapt to changes in data structures and capable of scaling horizontally to handle growing amount of data.

Examples of NoSQL Databases:

(i) Key-value Stores: Simple data storage with key-value (eg → Memcached, Redies)

(ii) Document Stores: Store data in JSON formate (ex: MongoDB, CouchBase)

(iii) Column family stores: Group related data columns together (ex: Cassandra, HBase).

(iv) Graph Database: Represent relationships between data entities ex → (infiniteGraph)

why facebook and google use NOSQL: Both facebook and google handle massive datasets and require high scalability and performance, Relational database might struggle under their workloads.

Case study of MetLife, facebook and google using NOSQL

① Metlife: Metlife a leading insurance company, deals with vast amounts of customer data. NOSQL database allow them to handle data flexibly and scale horizontally. They likely use NOSQL databases for customer profiles, policy management and claims processing.

② facebook: facebook manages an enormous social network with billions of users. NOSQL database enable efficient storage and retrieval of user profiles, posts and connections. They use NOSQL database for user profiles, news, feeds and reels.

③ Google: Google's services spam search, advertising and cloud computing. NOSQL database provide the scalability and able to adapt and change rapidly to meet new challenges. for these diverse applications. google use NOSQL database for Bigtable (for large scale data), firestore (for real time data sync) and Memorystore (for caching).

Q(5) List the four types of NOSQL Database

(i) key-value stores: This is the simplest type of NOSQL database. It stores data as key-value pairs similar to hash table.

Ex⇒

Key	value
Raufhan	7631867534
Anitabh	abcdefghijklmn

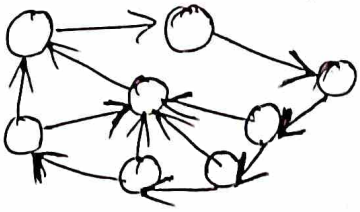
(ii) Document database: It stores data in JSON format
ex⇒ (mongoDb)

```
{ "id": "raufhan-123",
  "firstname": "Raufhan",
  "lastname": "Kumar",
  "password": "Mai kyo batau"
}
```

(iii) Column-oriented database: Column oriented database store data by column instead of by row.
Ex⇒

Grade	ID
4.00	001
6.00	002
8.00	003

(iv) Graph database: These store data in the form of node and edges form a graph. Ex⇒



(v) where NOSQL database is preferable over a relational database
Sol^n: When you need scalability and flexibility, Performance, Big data and real time Analytics, fast access and retrieval of data then NOSQL database is preferable over a relational database.

(p9) Design a database of any case study using NOSQL database terms and terminology. Explain with comments

ans ⇒ I am designing creating a database for an e-commerce website using NoSQL to store information about user and product details.

Collections: (1) Users: (Document store)

- user_id: unique identifier for the user
- name: user's name
- Email: unique identifier for login
- purchase_history: Array of ~~reference~~ purchase collection
- browsing_history: Array of product_id and time spend)

Comments: user_id is a unique identifier, while email serves as a login credential.

- purchase_history: It is an array of purchase collection.
- ~~Doc~~ browsing_history: It is an array of product_id and time_spend)

(2) Products: * product_id → unique identifier for the product
* name → product name
* description → detailed product description
* category → product category
* brand → Product brand
* reviews → array of user_id, rating, review ~~text~~

(3) Purchases: purchase_id: unique identifier for the purchase
user_id: Reference to document ID in users collection
products: Arry of product_id and quality
timestamp: Date and time of the purchase

Benefits of using NOSQL for this case study:

- flexible schema: NOSQL database is flexible fast retrieval and allows of data
- scalability: Document database can easily scale horizontally as users scale.
- Performance: NOSQL databases are optimized for fast reads and write, very good performance on large amount of data.

SQL

(Pyq) Discuss any ten SQL queries in DBMS with syntax.

ans ⇒ (1) SELECT statement: Retrieve data from a table

Ex⇒ SELECT col1, col2 FROM table ; Select * from tabl_na)

(2) INSERT statement: Insert new records into a table

Ex⇒ INSERT INTO table_name (col1, col2) VALUES (val1, val2);

(3) UPDATE statement: (update existing records in a table)
UPDATE table_name SET col1 = val1 WHERE con;

(4) DELETE statement: Delete records from a table

DELETE FROM table WHERE condition;

(5) CREATE TABLE statement: Create a new table in the database

```
CREATE TABLE table_name (
    col1 datatype,
    col2 datatype,
    ...
);
```

(6) ALTER TABLE statement: Modify an existing table structure

ALTER TABLE table_name ADD col_name datatype ;

(7) DROP TABLE statement: Delete a table from the database.

DROP TABLE table_name;

(8) SELECT DISTINCT statement: Retrieve unique values from a column.

SELECT DISTINCT col_name FROM table_name;

(9) COUNT function: Count the number of rows in a table

SELECT COUNT(*) FROM table_name;

(10) MAX, MIN, SUM function: Calculate the max, min, sum of values in column.

→ SELECT SUM (col1) FROM table_name;

(PYQ) Write a syntax of table creation and insertion command in sql.

Sol:

```
CREATE TABLE table_name(
    col1 datatype,
    col2 datatype,
    ....
);
```

Table creation

```
CREATE TABLE Employee(
EmployeeID INT PRIMARY KEY,
FirstName VARCHAR(50),
LastName VARCHAR(50),
DepartmentID INT,
);
```

Insertion command

```
INSERT INTO Employee (EmployeeID, FirstName, LastName, DepartmentID)
VALUES (1, 'John', 'Mom', 101);
```

employee . department = department . dep...

(Pyq) what is data dictionary used for?

A data dictionary is used to provide detailed information about the structure, organization and usage of data within a database or information system. It includes data definitions, data types, relationships b/w data elements, constraints and other metadata.

(Pyq) Define different types of Relational calculus.

① Tuple Relational calculus ② Domain relational calculus

(i) Tuple Relational calculus: TRC focuses on retrieving entire tuples (rows) from relations (tables). It uses tuple variables to represent these rows and your specify conditions.

(ii) Domain Relational calculus: DRC goes a step further and retrieves individual values from a relation. It uses domain variables to represent values within specific attribute (column) It uses quantifiers (like exists and for all) to express conditions on these value.

(Pg9) Elaborate the concept of logical database design.

Logical database design is the process of defining how data will be organized into tables and their relationships, focusing on what data is stored not how, it ensures efficient data storage and retrieval.

In other words →

Logical database design involves designing the schema including tables, relationships and constraints, based on the requirements of the system. This design focuses on organizing data in a logical and efficient manner to ensure data integrity, minimize redundancy and support the required functionalities of the system.

(Pg9) Explain the use of triggers in database management system

ans → Triggers in database management systems are like automated actions that are executed in response to certain events. They are used to enforce data integrity and automate tasks like logging changes or updating related records. Triggers can fire before or after events like INSERT, UPDATE or DELETE operations on tables.

(Pg9) write a query to get the highest, lowest, sum, and average salary of all employees from employee table having employee ID, Salary, firstname columns.

Syntax:

```
SELECT
MAX (Salary) AS highest_salary,
MIN (Salary) AS lowest_salary,
SUM (Salary) AS total_salary,
AVG (Salary) AS average_salary

from
    employee;
```

(Pg 8) Elucidate timestamp and validation protocol.

Timestamp: Each transaction is assigned a unique timestamp when it starts. This timestamp acts as a reference point for ordering transactions. Common methods include using the system clock.

Validation protocols: Validation protocols are mechanisms used to ensure that transaction in a database system maintain consistency and integrity. one common validation protocol is the timestamp-based on concurrency control protocols.

(Pg 9) interpret the role of concurrency and seriallizability in DBMS.

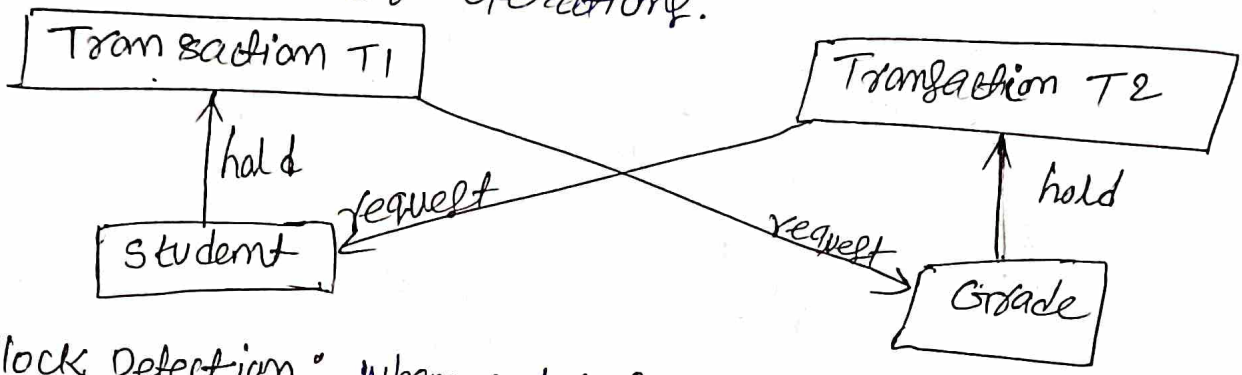
ans → (i) concurrency: In a database management system concurrency refers to the ability of multiple transactions to execute simultaneously without interfering with each other.

~~Seriallizability: Seriallizability is a property of a schedule of transactions in DBMS that ensures the final outcome of the schedule is~~

Serializability: This ensures that even though transactions are running concurrently, the final outcome is the same as if they were executed one after another in a specific order. This maintains data integrity and consistency

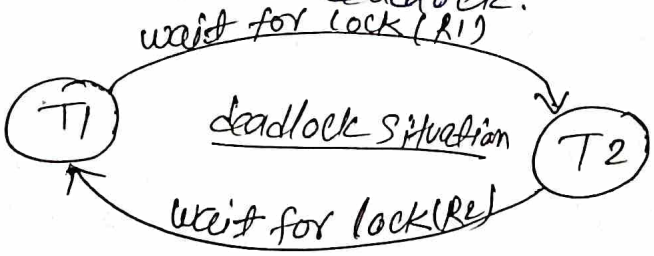
(Try) Why do deadlock occurs? create the complete process of deadlock detection and resolution with significant example:

In DBMS, a deadlock occurs when two or more transactions are ~~wait~~ waiting for each-other to release resources. such as locks on database objects, that they need to complete their operations.



Deadlock Detection: When a transaction waits indefinitely to obtain a lock, then DBMS detects whether the transaction is involved in a deadlock or not.

wait for graph: is one of the method for ~~de~~ detecting the deadlock situation. This method is suitable for smaller database. in this method, a graph is drawn based on the transaction and its lock. if the graph has a closed loop or a cycle then there is a deadlock.



Resolution: once a deadlock is detected, there are several approaches to resolve it.

- Deadlock prevention
- Deadlock Avoidance
- Deadlock detection and recovery

(i) Deadlock prevention: This involves ensuring that the conditions leading to deadlock cannot occur.

... data structure

Disadvantages: Increased complexity in design and implementation. less popular due to the rise of relational model.

(Q5) Define normalization. why we need to normalize a database in SQL? Briefly discuss the insert, delete, and update anomalies. if relations are not in 2NF.

ans ⇒ Normalization is the process of organizing data in a database to reduce redundancy and dependency in SQL. normalization involves breaking down a database into smaller, more manageable tables and defining relationships b/w them.

we need normalization for several reasons.

① Eliminating redundancy ② Maintaining data integrity ③ facilitating flexibility and scalability.

Insert, update, delete anomalies occur when a database is not normalized in second normal form.

(i) Insert Anomaly: This occurs when adding new data requires inserting information into multiple places potentially leading to inconsistencies or data duplication

(ii) Delete Anomaly: If we delete some rows from the table and if any other information or data which is required is also deleted from the database this is called the deletion anomaly.

(iii) Update Anomaly: Updating data in one place but not in others can lead to inconsistencies or errors, affecting the overall accuracy of the database.

(PYQ) Write a syntax for outer join and inner join with example:

ans ⇒

Inner join: It returns only the tuples that have matching value in both tables based on the specified condition It excludes tuples that do not have matching values.

Example:

Employees table

emp_id	name	dep_id
1	A	10
2	B	20
3	C	10

Departments

dept_id	dept_name
10	HR
20	Engineering
30	Marketing

Inner join on dept_id:

emp_id	name	dep_id	dept_name
1	A	10	HR
2	B	20	Engineering
3	C	10	HR

Relational algebra syntax: $R1 \bowtie_{R1.dept\_id = R2.dept\_id} R2$

~~$R1 \bowtie_{R1.dept\_id = R2}$~~

SQL syntax:
```
SELECT *
FROM Employees
INNER JOIN Departments
ON Employees.dept_id=Departments.dept_id;
```

* Outer join: It returns tuples that have matching values in both tables as well as the tuples that do not have matches in one of the tables. There are three types of outer join (i) Left outer join (ii) Right outer join (iii) full outer join

(i) Left outer join: It returns all tuples from the left table and matched tuples from the right table. Non-matching tuples from the left relation table are padded with NULL values

Example:

Employee

emp_id	name	dept_id
1	A	10
2	B	20
3	C	40

Departments

dept_id	dept_name
10	HR
20	Engineering
30	Marketing

emp_id	name	dept_id	dept_name
1	A	10	HR
2	B	20	Engineering
3	C	40	NULL

* Relational Algebra syntax:

$R1 \leftouterjoin_{R1.dept\_id = R2.dept\_id} R2$

SQL Syntax:
```
SELECT *
FROM Employees
LEFT OUTER JOIN Departments
ON Employee.dept_id = Departments.dept_id;
```

(iii) Right outer join: It returns all tuples from the right table and the matched tuple from the left table. Non-matching tuples from the right table is padded with "NULL" value.

Example:

Employees

emp_id	name	dept_id
1	A	10
2	B	20
3	C	40

Departments

dept_id	dept_name
10	HR
20	Engineering
30	Marketing

emp_id	name	dept_id	dept_name
1	A	10	HR
2	B	20	Engineering
NULL	NULL	30	Marketing

Relational Algebra Syntax:

R1 \rightouterjoin R1.dept_id = R2.dept_id R2

SQL Syntax:

```
SELECT *
FROM Employees
RIGHT OUTER JOIN Departments
ON Employees.dept_id = Departments.dept_id;
```

(iv) Full outer join: It returns all the tuples that have matched in both tables as well as it returns all tuple from left table and right table as well. Non-matching tuples are padded with NULL values.

Example:

Employees

emp_id	name	dept_id
1	A	10
2	B	20
3	C	40

Departments

dept_id	dept_name
10	HR
20	Engineering
30	Marketing

HW-1
22/9/25

emp_id	name	dept_id	dept_name
1	A	10	HR
2	B	20	Engineering
3	C	40	NULL
NULL	NULL	30	Marketing

Relational algebra syntax:

R1 ⟗ fullouter join R1.dept_id = R2.dept_id R2

SQL syntax:

```
SELECT *
FROM Employees
FULL OUTER JOIN Departments
ON Employees.dept_id = Departments.dept_id;
```

FYQ) Diff b/w inner join and outer join

Inner join	outer join
(i) It returns all the tuples that have matched in both table	It return all the tuples that have matched in both table as well as tuples that donot have matched in one of the tables.
(ii) Relational Algebra syntax: R1 ⋈ R1.dept_id = R2.dept_id R2	Relational algebra syntax: R1 ⟗ fullouterjoin R1.dept_id = R2.dept_id R2
(iii) SQL syntax: SELECT * FROM Employees INNER JOIN Departments ON Employees.dept_id = Departments.dept_id;	SQL syntax: SELECT * FROM Employees FULL OUTER JOIN Departments ON Employee.dept_id = Departments.dept_id;
(iv) Inner join is more faster than outer join	(iv) outer join is slower than INNER join