Sub :- DBMS

* Unit :- 1 :-
Marks - 5 Max Marks - 08

PAGE NO.
DATE 17 /02 / 23

1) What is DBMS ? List application of DBMS ?
=> DBMS stand for Database Managment System.
- DBMS is a system software that youse to manage, create or organize the database.
- Database is a collection of inter-related data that can be used to insert, retrive or delete data efficiently.
- It also used to organised data in form of view; table, schema or reports.
- DBMS provide protection and security to database
- In case of multiple users, DBMS provide of maintain the consistency.
- Some popular DBMS example are MySQL, oracal, SQL serve, etc.

* Applications of DBMS :-
i) Railway Reservation System :-
- It is used to keep record of arrival of train, deprature of train, ticket booking and give information to user by using DBMS.
ii) Liabary Managment System :-
- There are lot's of book in Libary of it's difficult to keep their record in registers.
- So DBMS is neccery to keep record of issue date, authore, name of book, etc.
iii) Banking :-
- People are doing lot of transaction daily without directly going into bank.
- It is possible because of DBMS, DBMS manage all transaction data of a users over the database.
iv) Education Institute :-
- All the student's and examination data maintain over internet by help of DBMS.
- It contain student detail, results, greads, cources aviable.
v) Social Media application :-
- The application stored user data like profile, post of interaction.
- And It help to provide a personalized user experience.

Q. * Characteristics of DBMS *

PAGE NO.
DATE 11/02/23

⇒ What is DBMS ? What is Database Architecture ?

* DBMS :- check Q. 1

* Database Architecture :-

- Database Architecture is refers to overall design and structure of database system.
- Database Architecture help to design, develop, implement & ~~develop~~ maintain the database management system.
- By ~~this~~ help of DBMS architecture we can divide database system into indivisual components.
- These components can be independently modified, replaced, changed altered.
- It also help to understand components of database.

- Database architecture includes serval key components :-

i) Data Model :- It is a visual representation of data stored in database.

ii) DBMS :- It is system software that allow user to store, access, ~~modify~~ manage data in database.

iii) Data storage :- Data stored in various format such as table, file, document.

iv) Data processing :- It Inveles data manpulation and analysing by using various techniques.

v) Data Security :- It's a way to protect confidentility, intergrity, avilability of data stored in database.

* optional ! G - ~~DBMS~~ DBMS point & G - Architecture point ✓

- Well design database architecture improve performance, scalability & reability of database.

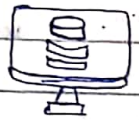
Q. Levels of Database Architecture ?

Q. What is database Architecture ? Explain their type ?

* Database Architecture :- Written in Q.2. 6-points.
* There are three type of architecture in DBMS :-

i) 1-tier Architecture :-

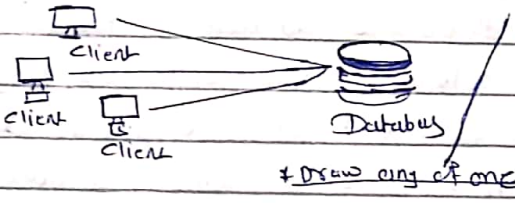
- In 1-tier Architecture client, server and database all situated on same system.
- These type of architecture used to build local application where programmer can directly communicate to database for quick response.
- Diagram :-



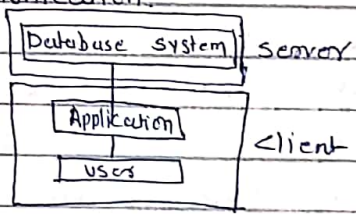
- These architecture rearly used in Production.

ii) 2-tier Architecture :-

- In 2-tier Architecture presentation layer runs on client side and data is stored in server.
- These architecture add security to dbms as it not exposed to end user directly.
- Also provide direct & faster communication.
- Dig. :-



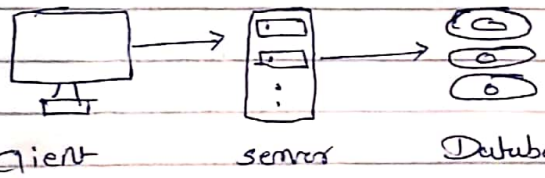
* Draw any of one



iii) 3-tier - Architecture :-

- These architecture is extension of 2-tier architecture.
- There are three layers in these architecture Presentation layer, Application layer, Database servers.
- In these architecture application layer lie between users & DBMS
- Application layer contain functional logic, constraint & rules. before passing data to users.
- These is most popular database Architecture

Dig :-



Q. What is Data mode? Explain their types?

PAGE NO.
DATE / /

26)	File system	DBMS
-	It is software that manage files & organize file in storage medium.	It is software for managing the database.
-	Redudent data can be present	No Redudent data present.
-	Doesn't provide backup & recovery if data lost.	Provide backup & recovery even if data lost.
-	No efficient query processing.	There is efficient query processing.
-	It less complex than DBMS.	It is more complex than file sy.
-	Less data consistency.	More data consistency.
-	Provide less security	Provide more security
-	one user access data at a time	Multiple user access data at a time.
-	ex. Cobol, C++	ex Oracle, SQL server.

27) Discuss Structure of relational database?

=> Relational Database :-

- It is type of database that store data in structural formate. using table that related to each others. using keys.
- Basic building block of Relational database is table.
- Table contains rows & columns. Row represent single record in database. & Each column represent specific data about record
- Column have specific datatypes. i.e string, number or date, etc.
- Structure of relational database define by its schema. Which is set of rules that ~~tabl~~ describe table & its relation to each other.
- Relation is estabilished by using keys. between tables.
- keys are column level's use to link tables.
- Primary key & foreign key two type.
- To retrive data we can use SQL, which is programming language to intract with relational database.

Q Who is DBA ? state & explain function performed by DBA ?

=> DBA :-

- A Database Administrator is a professional responsible for managing, maintaining & securing company's database.
- DBA consist of a team of people rather than a single individual.
- DBA role required high level of technical expertices skills.
- They are responsible for overall health, performances & security of database that they are managing.

* Function Performed by DBA :-

- Database Design :- Design & implement new database. This involves creating a data model, defining data structure & how data will be stored.
- Performance tuning :- DBA optimize database performance by monitorise resources usage & tuning database parameters.
- Backup & recovery :- DBA develop & implements backup & recovery prodcedure to ensure that data is protected aginst data lost or corruption.
- Maintainces & troubleshooting :- DBA perform routine DBA tasks. They also troubleshoot & resolve issue that arise, such as system failure, data corruption.
- Security Managment :- DBA implement's security measures to ensure data confidentility, intergity & aviability.
  Set user access permission. set security protocol.

Q. What is Data Models & Explain type of Data models :-

Data models :-

i) Data model is the modeling of data description, data semantic, consistency constraint of data.

ii) Provide a set of conceptual tools that are used to represent the description of data.

iii) Data models are used to describe how is the data is stored, accessed & modified in DBMS.

There are four main data models :-

i) Relational model
ii) Entity - Relational data - Model.
iii) Object - based data - Model.
iv) Semi - Structured data Model.

1) Relational Data Model :-

i) It is a most popular data model among the all.
ii) Database is represented as collection of relation in the form of row & column of 2-D table.
iii) Row known as 'tupele' & column known as 'Attribute'.
iv) Data stored in table called 'Relation'.
v) Relation can be Normalized.

2) Entity - Relational Data - Models :-

i) ER model is based on the notion of real-world entities and relationship among them.
ii) While planing real-world scenario into database model ER model creates entity set, relation set, attribute.
iii) It is best used to conceptual design of database.
iv) ER Model based on :-

i) Entities :- An Entity is nothing but an real-world object.

Every entity have properties called Attributes.

ii) Relationship :- The logic between among the entities known as relationship.
There are 4 type of Relationship :-
(i) one to one (ii) one to many
(iii) many to one (iv) many to many.

3) Object - Relational Data model :-
i) It is combination of Object - Oriented data model and Relational data model. to deal with added datatypes.
ii) It allow attribute of tuples to have complex type. such as nested relation.
iii) Keep relational foundation, in particular declarative access of data, while ~~con~~ extending modeling power.
iv) Provide upward compatibility with existing relational language

4) Semi - Structured Data Model :-

Q. ~~What~~ Explain in detail Intances of schema.

1) Schema :-

i) It is the overall description of database.

ii) The basic structure of how data is stored in database is called schema.

Three type of schema :-

i) Physical Schema :- Describe database design at physical level.

Can be changed without affecting application program

ii) Logical Schema :-

i) Describe database design at logical level.

ii) Programmers construct application using this.

~~iii) V~~

iii) View Schema :-

i) Describe database design at view level.

ii) Highest level of Schema & describe view for end user

2) Instance :-

i) Instance is collection of information that stored in a database at a particular moment.

ii) Database instance is dynamic value, thus is keep changing.

iii) Instance is also known as extension or current state of database state.

Unit - 2

Database Design and ER-Model :-

Q. What is Database Design & Explain it's important.

* Database Design :-

i) Database Design define as collection of task & process that enhances the designing, implementation, development and maintances of enterprise data management system.

ii) By designing proper database it reduce the maintances.

iii) Also improve data consistency and cost-effective.

iv) The main objective is to produce physical and logical design of a database system.

* Importances of database Design :-

i) It give the blueprint of how is data is going to stored.

ii) By proper design of database it highly affect on overall application performances.

iii) It also give a clear idea of behaviour of any application and how request are processed.

iv) Meet ~~pr~~ all requirment of user.

v) If database design is proper it reduce the processing time of application.

*imp :- What is ER model ? explain concept of er?
imp :- What is strong entity & weak entity.

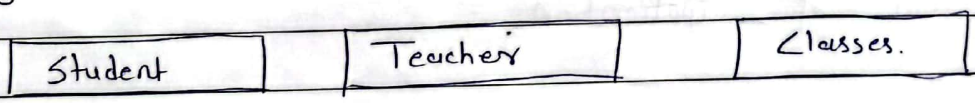
Q. What is ER-Model & Explain Entity and Attribute.

+ ER-Model :-

i) ER-Model is high level datamodel.
ii) ER-Model stand for Entity Relationship Model.
iii) ER-Model defines logical view of a database.
iv) ER-Model works on real-world entites.
v) ER-Model is build on three basic concepts :-
   i) Entity ii) Attribute iii) Relation

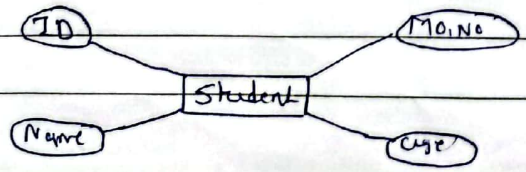
* Entity :-

- Entity is nothing but a real-world object, that c
  be identified.
- ex. In school database - student, teacher, classes are
  entity.
- Object that physically exists and logically constructed
  in real world called entity.
- All entity have some attribute that give them
  identity.
- Group of similar type of entities called entity set.
- Entity is represent by rectangle.
- ex



+ Attributes :-

- Attribute are the properties that defines entity
- All attribute have value.
- Attribute are represented by oval/ellipses.
- There are range of value assigne to attribute.
  ex. student name cannot be numeric / age cannot be ne
- ex. student have attribue like ID, Name, Mo.No, age.



Extra :-

* While when ask to explain type also :-

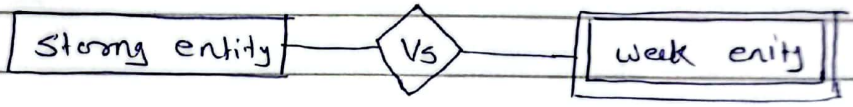
→ Types of Entity :-

i) Strong entity :-

- It is entity type that have key attribute.
- It has always has primary key.
- It represent in single rectangle.
- Ex :- roll-Number in class of student is unique no.

ii) Weak entity :-

- It is entity type that doesn't have key attribute.
- A forgin key must used in combination with its attribute to create primary key.
- They can't identified by them own so called weak entity.
- Represent by double outline rectangle.
- Ex. address of student is weak entity as many student can be from same locality.



Q. Write Short note on Attribute :-

* Attribute :-

→ - There are five type of attribute :-

i) Simple Attribute
ii) Composite Attribute
iii) Derived Attribute
iv) Single-Value Attribute
v) Multi-Value Attribute.

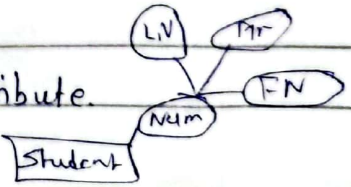
i) Simple Attribute :-

- Are have atomic value.
- Which can not be divided further.
- ex. student phone-no. atomic value of 10 digit.

ii) Composite Attribute :-

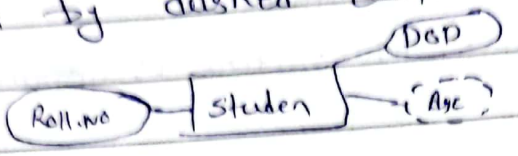
Made up of more than one simple attribute.

ex. student name have last name, from m-



iii) Derived Data - Attribute :-

+ Based on other attribute.
- Are not store directly in database.
- Ex From date-of-Birth we can get age
- shown by dashed ellipse.



iv) Single-value - Attribute :-

- It have only one value.
- Ex social-security-NO.

v) Multi-value attribute :-

It have more than one value.

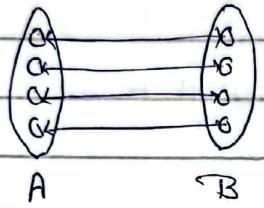
Ex. Mo.No / Email.

Q. Explain in detail Relationship.

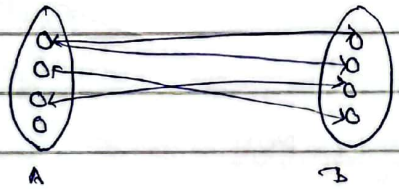
* Relationship :-
- The association between entities known as relationship.
- It represented in diamond shape.
- Relationship written inside the box.
- Connected to entities by a line.
- Group of similar type of relationship called as relationship set.

* There are four type of relationship :

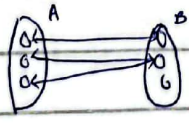
i) One - to - One :-
- One enity from set A can associate only with one entity from set B.



ii) One - to - Many :-
- One entity from entity set A can be associate with more than one entitiy from entity set B. However entity from set B can associate with at ~~least~~ most one entity.



iii) Many - to - One :-
More than one entity from entity set A can associate with at most one entity from entity set B. however entity from entity set B can associate with more than one entity from set A.



iv) Many - to - Many :-
~~Until~~ one entity from entity set A can associate with more than one entity from B & vice versa

* Unit 2 :-

Marks :- 10 Max Marks :- 12

PAGE NO.
DATE 17 / 02 / 25

Q. ~~Explain ER-Model~~

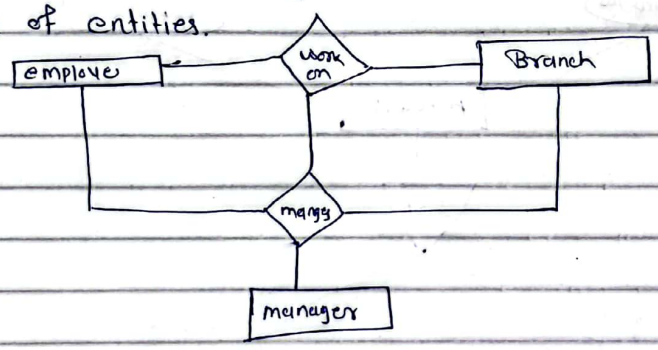
Q Extended feature of ER Model.

- The extended ER model is an extension of ER model that includes additional feature to better representation complex data relationship.
- There are three type of extended er model :-

i) Aggregation :-

- It is an abstraction in which relationship set are treated as higher level entity set and can participate in relationship.
- Aggregation allow us to indicate that relationship set participates another relationship set.
- Aggregation used to simplify the details of given database where ternary relationship will be changed into binary relationship.
- Ternary relationship is only relationship which working with three type of entities.

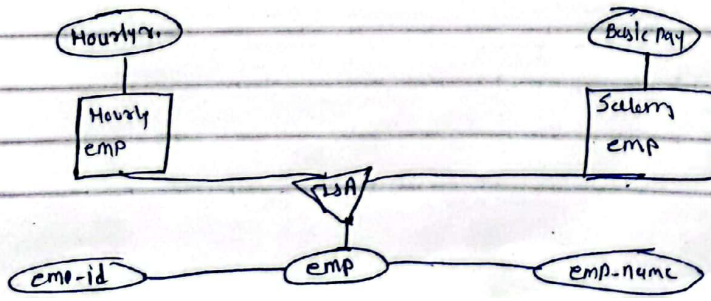
→ Dig.



ii) Generalization :-

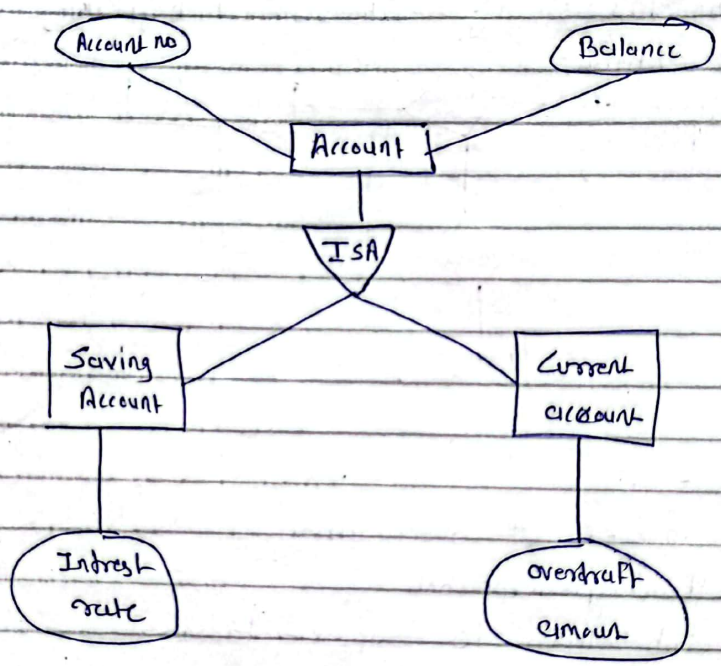
- It is reverse process of specilization.
- It is a bottom up approach.
- It convert sub-class to super-classes.
- It combines a number of entity sets that shares the same features into high-level-entity set.
- If the sub-class information is given for entity set then ISA relationship type will be used.
- It represent the connectivity between the subclasses and superclasses.

→ Dia



* Specialization :-

- The process of designing sub grouping within an entity set is called specialization.
- It is top-down process.
- If an entity set given with all attributes in which instances of entity set are differentiated according to the given attribute value, then that sub-clases or sub-entity set can formed from given attribute.
- ex. Specilization of account creates two entity sets saving account of current account
- In following ER-model specilization represented by traiangle labled as ISA.
- ISA relationship refered as super-class.



*imp :- Explain specilization with example.
Explain generalization with example.

imp:- What is functional depency ! explain 2NF & 3NF with example.
imp:- explain 3NF & BCNF with example.
imp:- diff bet 3NF & BCNF.

Q. Functional Depedency :-
- Functional Depedency is relationship between attribute of a table that depend on each others.
- Functional Dependency introduce by E.F. Codd.
- Function dependency represented by 'arrow sign (->).
- It help in preventing data redundency & to get known about bad design.
- Ex :- A → B
  B functionally dependent on A
  A determinant set
  B dependent attribute.

Q. We have <Deparment> table with DeptID & DeptName attribute.
- DeptID is primary key.
- DeptID uniquely identifies the DeptName.
- If we want to know DeptName we first have to know DeptID

Dept ID	DeptName
001	Finance
002	Marketing
003	HR

- Therefor, DeptID is functionally dependent on deptName.
  DeptId → DeptName

* Type of Functional dependency :-

1) Trivial Functional dependency :-
- It occure when B is subset of A.
- ex. {DeptId, DeptName} → DeptId

2) Non-Trivial Functional dependency :-
- It occure when B is not subset of A
- ex DeptID → DeptName

3) Complete Non-Trivial Functional dependency :-
- It occure when A intersection B is null in
  A → B

Q. Normalization

-> Normalization is process of organizing data in database.

- Normalization is used to minimize redudency from set of relation.
- Normalization divide the lorgers table into smaller table and link them using relationship.
- Normal form is used to reduce redundancy from database table.
- Type of normal form :-

1) 1NF 2) 2NF 3) 3NF 4) BCNF

1) First Normal form (1NF) :-

- A relation will be 1NF if it have atomic value.
- An attribute can not hold multiple value in a table, It must hold single value attribute.
- 1NF disallowed the multi-value attribute.
- Ex :- Relation Employee is not 1NF because multi-value attribute emp-phone.

employee table :-

emp-id	emp-name	emp-phone	emp-state
14	Sam	9601584714 1204132147	UP
20	Ram	8601020304 0506070809	Bihar
22	Sham	1234567890 2345678900	Punjab

- Decomposition of ~~1NF~~ employee table into 1NF :-

emp-id	emp-name	emp-phone	~~UP~~ emp-state
14	Sam	9601584714	UP
20	Ram	8601020304	Bihar
22	Sham	1234567890	Punjab

2) 2NF - second Normal Form :-

- In 2NF, relational must be 1NF.
- In 2NF all non-key attributes fully functional dependent on the primary key.
- Ex. Teacher table

Teacher-id	Subject	Teacher-age
25	Chemistry	36
25	Biology	30
47	English	35
83	Math	38

- In above example, non-prime attribute Teacher-age is dependent on Teacher-id.
- Which is proper sub-set of candidate key.
- So it didn't follow rules of 2NF.
- To convert table in 2NF, decompose it in 2 table

• Teather-detail table :-

Teacher-id	Teacher-age
25	30
47	35
83	38

• Teachers-subject table :-

Teacher-id	Subject
25	Chemistry
25	Biology
47	English
83	Math
83	computer.

* Third Normal Form (3NF) :-
- A relation will be 3NF if is in 2NF of not contain any transitive partial dependency.
- It is used to reduce data duplication.
- Also used to get data integrity.
- If there is no transitive dependency for non-prime attributes, then relation is 3NF.
- A relation is 3NF if it have atleast one of following condition for every non-trivial functional dependency :-
- $X \rightarrow Y$
- X is super key.
- Y is prime attribute.

Ex. In Customers table :-

Customers (customer-id, Name, email, phone, address);
- In this customer-id → Name, email, phone, address.
- To bring table in 3NF, we need to eliminate any transitive dependency.
- So we will create seprate customer-info table for email, phone, address.
- Customers (customer-id, name);
- Customer-info (customer-id, email, phone, address);
- In this tables all attribute depend on only one primary key & there is no transitive dependencies.

4) Boycee Codd normal form (BCNF):-

- BCNF is advance version of 3NF.
- It is stricter than 3NF
- A table is BCNF if every functional depedency is $X \rightarrow Y$
- X is super key of table.
- For BCNF, table must be 3NF
- Every Functional dependency, LHS is super key.

Ex. Emp table :-

emp-id	emp-country	emp-dept	dept-type	emp-dept-no
264	India	Design	D 394	283
264	India	Testing	D 394	300
364	UK	stores	D 283	232
364	UK	Development	D 283	549

→ In above table

- emp-id → emp-country
- emp-dept → dept-type, emp-dept-no
- This is NOT BCNF decuse of two key there
- To convert in BCNF, decompose in 3 table

1) Emp-country table -

emp-id	emp-country
264	India
364	UK

2) emp-dept table

emp-dept	dept-type	emp-dept-no
Designing	D 294	283
Testing	D 294	300
Storeg	D 394	232
Development	D 394	549

3) Emp-dept-mapping table :-

emp-id	emp-dept
264	283
264	300
364	232
364	549

* Functional dependency

emp-id → emp country

emp-dept → dept-type, emp-dept-no

* Keys :-

1st table :- emp-id

2nd table :- emp-dept

3rd table :- emp-id, emp-dept

∴ Now this is 3NF.

Unit - 3

Mark - 12 Max Mark - 20

PAGE NO.
DATE / /

Q. Explain relational algebric oprations?

Relational algebra have six oprators.

i) Select :- σ (condition)

- Select oprations select tuples that stisfy specific condition.
- It is denoted by sigma (σ)

Notation :- σ p (r)

- Where :-

σ is used for prediction.

- R is used for relation.
- P is used as proposational logic formula. also use as connectors like AND, OR
- Ex

Stud-Name	marks	roll-no
Ram	70	12
sham	80	24
Ramesh	90	36

o/p :-

Stud-Name	marks	roll-no
Ram	70	12

σ stud-name = "Ram" (marks)

ii) Project oprations :- ∏(Display) :-

- Project oprations show list of attributes those we want to display result.
- It element other attribute
- It denoted by ∏
- Notation :- ∏ A1, A2, An (r)

where A1, A2, An are attribute of relation (r).

- Ex student relation.

Stud-name	marks	roll-no
Ram	70	12
sham	80	24
Ramesh	90	36

→

∏ stud-name, marks (student)

O/P

Stud name	Marks
Ram	70
sham	80
Ramesh	90

iii) Union Operator = ∪

- If there are 2 tupels R & S. This oprations contain all the tuples that are either in R or S or both.
- It eliemente duplicate tuples.
- Denoted by ∪
- Notation :-

R ∪ S

- Union operation must hold following condition
- R & S must have attribute of same number
- Duplicate tuples eliemented automatically.

eg → Depositor

Customer-name	Acc-no
Ram	101
Sham	102
Ramesh	103
Suresh	104

Borrow

Customer-name	Loan-no
Virat	11
Sham	12
Viraj	13
Raj	14

I/O :- Π customer-name (Borrow) $\cup$ Π customer-name (Depositor)

O/P :-

Customer-name
Ram
Sham
Ramesh
Suresh
Virat
Viraj
Raj

iv) set intersection :-

- Suppose two tuples R & S. Set interception operation contain all tuples that are in both R & S.
- Denoted by $\cap$.
- Notation :- $R \cap S$

eg Using above Depositor & Borrower table

I/P :- 'Depositor $\cap$ Borrower

Π Customer-name (Depositor) $\cap$ Π customer-name (Borrower)

Sham

v) set difference :-

- Suppose two tuples R & S. The set intersection operation contain all tuples that are in R but not in S.
- Denoted by intersection minus (-)
- Notation :- $R - S$

eg above

Π customer-name (Depositor) $-$ Π customer-name (Borrower)

Customer-name
Ram
Ramesh
Suresh

vi) Cartesian product :-

- It is used to combine each row in one table with each row in other table.
- Also known as cross product.
- Denoted by x
- Notation :- R x S
- Emp :-

Emp_ID	Emp_name	Emp_dept
1	Ram	A
2	shyam	C
3	vinay	B

Dept :-

Dept_no	Dept_name
A	Marketing
B	Sales
C	Legal

I/P :- Emp x Dept

o/p :-

Emp_ID	Emp_name	emp_dept	dept_no	dept_name
1	Ram	A	A	M
1	Ram	A	B	S
1	Ram	A	C	L
2	shyam	B	A	M
2	shyam	B	B	S
2	shyam	B	C	L
3	vinay	B	A	M
3	vinay	B	B	S
3	vinay	B	C	L

DATE / /

Q. What SQL ? Which are build in datatype in SQL ?

* SQL

- SQL stand for structured Query Language.
- It is design for managing of manipulating data in relational database managment system.
- SQL used to perform task like insert, retriving, updating, creating of managing ~~data~~ database schema.
- All RDBMS like MySQL, MS Aless, oracle, SQL server use SQL as their standered language.
- Also they use diffrent dialect. i.e
- MS SQL use T-SQL
- oracle usel PL/SQL.

* Datatypes :-

- It define as type of data that can store in column of table.
- buit in datatypes :-

i) Number (P, S) :-

- Used to store numeric value such as integer, float, smallint, BigInt.
- P is Precision & S is scale.
- Max size is 38 digits.
- eg emp-id Nomber (2);

ii) Char (size) :-

- It is fixed length datatype.
- Accept alphabet & alpha-numeric value
- Max size 2000 bytes / char
- eg emp-name char (10); = Ram';

iii) varchar 2 (size) :-

- It support dynamic memory allocation :-
- Accept alphabet of alphanumeric value.
- Maxi size 4000 / bytes
- eg ename varchar 2 (20); 'ram'

* Read DDL / DML / DCL comands onces

iv) Date :-
- It is fixed length datatype
- Accept only date value
- It occure 7 byte memory.
- Default format is 'DD/MON/YY'
  DD - Day of. month MON - First 3 char of Month YY - last 2 digit of Year
- DOB date ; 05-JUL-21

v) Long :-
- Accept alphabet, numeric of alpha-numeric value.
- max size 2gb.
- Using Long we can upload Document.
- Table can accept 2 long only.
- ex std-resume Long ;

Q. Create table student (
  std-id number(2);
  std-name char(20);
  std-add varchar2(20);
  std-dob date);

o/p

std-id	std nam	std add	DOB
01	Ram	Shivpur	01-JAN-21

6. What is Constraint ? Explain various intergity constrain with example?

* Constraint :-
- Constraint are use to specify rule for data in table.
- Constraint are pre-define rule that applied on database columns.
- Constraint can be specify at time of table creation with CREATE TABLE command.
- Also after table created with ALTER TABLE command.
- Constraint can be column level or table level.
- Constrain get active when DML command perform
- There are 6 type of constraint :-
  i) NOT NULL
  ii) UNIQUE
  iii) PRIMARY KEY
  iv) REFRENCES KEY
  v) CHECK
  vi) DEFAULT

i) NOT NULL :-
- In dutubase table by defaut all column accept null value.
- To avoid null value in column we use NOT NULL constraint.
- It force to have a value in each row.

Eg. Create table student (
std-id number(2) ~~NOT NULL~~ constraint student-std-id-nn NOT NULL,
std-name varchar(2) NOT NULL);

ii) UNIQUE constraint :-
- UNIQUE constraint ensure that all value in colume are diffrent.
- It avoid dublicate value.
- It accept null.

Eg Create table student (
std-id number(2) ~~UNIQUE~~ constraint student-std-id-un - UNIQUE
std-name varchar(2) ~~NOT NULL~~
);

iii) PRIMARY KEY :-
- This constraint uniquly identify each record in table
- It must contain unique value
- It cannot be null value
- A table have only one primary key.

Create table student (
std-id number(4) constrain student-~~std~~-id-PK PRIMARY KEY
std-name varchar2(20));

iv) FOREIGN KEY - (REFERENCES KEY)
- References key used to prevent action that can destroy link between tables.
- References key is value in one table appears in second table.
- A table with primary key called parent table
- A table with reference key called child table
- A forgeign key is filed that references to primary key of another table.

Eg create table class (
std-id number(2),
std-name varchar2(20),
constraint class-std-id-fk references student (std-id.));

v) Defaut :- Set A value for a column when no value specified.
Eg create table student (std-id number(2), std-country varchar2(20) DEFAULT 'India');

v) DEFAULT constraint :-

- It is used when to set a default value for a column when no value specified.

eg Creat table Student (
    ~~Stid~~ stud id number(2),
    std name varchar2(20),
    std county varchar 2(20) DEFAULT 'India');

vi) CHECK Constraint :-

- It is used to check given condition.
- If condition is true record inserted otherwise it return error massage.

eg Create table Student (
    Std_id numbers(2),
    std-name Varchar2(20),
    std-dob date constraint student-std-dob-chk check(dob > 01/JAN/05));

Q. What is Join? Explain Type of join?

* Join :-

- Join is used to combine rows from two or more table, on a related column between them.
- To retriving data from more than one table in a single query known as join.
- Its best example for relational database.
- Syntax :- Select table1.col1, table1.col2, table2.col1, table2.col2 From <table 1>, <table 2> where <table1.col = table2.col>;
- Join condition is placed inside where clause of select statment.

* Type's of join :-

i) Equi - join :-

- To retriving data from more than one table in a single query using equality oprator is called equi-join.
- For use equi join, Joining tables must contain at lest one similar column.
- Corresponding column datatype must be same.
- Syntax :- Select columns-name From table1, table2
    where colome = colume;

PAGE NO.
DATE / /

Ex. emp table

emp_ID	F.name	dept_ID
1	Ram	10
2	Sham	20
3	Viry	30
4	Shem	40

② dept. table

dept_id	dept_name
10	Sales
20	testing
30	coding
40	server

I/P :- Select emp_ID, F.name, emp.dept_ID, dept.dept_ID, dept_name from emp, dept where emp.dept_id = dept.dept_id;

O/P :-

emp_ID	Fname	dept_id	dept_name
1	Ram	10	Sales
2	sham	20	testing
3	Viry	30	coding
4	Shem	40	server

ii) NON-EQUI JOIN :-

- To retriving data from more than one table in a single query without using equality operator known as ~~equi~~ non-equi-join.
- In non-equi join, joining condition is compared with >, <, >=, <=, between, not between operators.

Ex. Refer above table :-

Sol:- Select a.emp_id, a.F.name, b.dept_id, b.dept_name from emp a, dept b where dept.dept_id >= 20;

emp_id	F.name	dept_id	dept_name
2	Ram	20	testing
3	Viry	30	coding
4	Shem	40	server

iii) Cross join - (Cartesian join) :-

- Cross join also known as cartesian join.
- In cross join output display with no. of record in table 1 * no. of record in table 2.
- Syn :- Select table1.colum1, table2.column2
  from table1
  Cross join table2;

of hisul;

iv) Self Join :-

- To join table itself called selfjoin.
- In from clause both table are same.

Ex. select M.ename as Boss-Name, E.Name as sub

Q8.

* Outer Join :-
- In outer join, first it display matching record from table 1, table 2
- and unmatched data from ither table 1/ table / both based on join condn.

i) Right outer join :-
- In this join, first display matching records from table 1, table 2.
- And unmatched records from either t1, t2 right side of table on join condition

Ex. Select Ename, Empno, job, ~~from~~ Emp.Deptno, Dept.Deptno, Dname from Emp, Dept where Emp.Deptno (+) = Dept.Deptno;

ii) Left outer join :-
- In this join, first display matching records from table 1 & table 2.
- And unmatched record from ~~right~~ left side of the join condition.

Ex. Select empno, empname, job, Emp.Deptno, Dept.Deptno, dname, loc from emp, dept where Emp.Deptno = Dept.Deptno (+);

iii) Full outer join :-
- In this join, it display matching records from table 1, table 2
- And unmatched record from both table.

Ex Select empno, empname, job, emp.deptno, dept.deptno, dename, loc
from emp, dept where emp.dept = dept.deptno (+)
UNION
Select empno, empname, job, emp.deptno, dept.deptno, deptname, loc
where emp.dept (+) = dept.deptno;

Q.G. What is view explain with example.

* view :-
- view is one of the logical Database object.
- view are create by BASED on SELECT statment output.
- view is based on BASE TABLE.
- view doesn't hold any data of it re-present only structure
- view provid security
- We can hold multiple table information into single view by using view
- All created view names stored in USER-VIEWS

* Type of view :-

▷ Simple view :-
- These view created by SELECT statment.
- It doesn't contain Join cond^n / Group by / Having clause.
- Simple view support DML, SELECT & DESC comnd.
- Syntax :-

Create or Replace view <view-name> As <select statment> ;

Eg. CREATE VIEW std-view AS SELECT std-id, std-Name FROM student WHERE class = 'MCA';

Example :-

Create view dept10-view as select empno, ename, sal, deptno from emp where deptno = 10;

Select * from dept10-view;

empno	ename	sal	deptno
11	Ram	4000	10
22	Sham	5000	10
33	Viru	6000	10
44	Shery	7000	

2) Complex view

- These view created by SELECT statment / JOIN condⁿ / Group by Having clause.
- In complex view DML comnd not supported.
- only select of ~~DESE~~ DESC comnd supported.
- By complex view we can hold multiple table info into single view

Ex. CREATE VIEW mark-view AS SELECT std-id , AVG (marks), std-name FROM student GROUP BY std-id ;

Select * from mark-view;

std-id	Marks	std-name
11	80	Ram
22	90	Sham
33	70	viny
44	60	sheny

Q.3. What is operators ? Explain special operators in details.

* Operators :-

- The reserve words and character are called operators, which are used with WHERE clause in sql query.
- To query database using operater we use WHERE clause
- Operator are neccesary to define condition in sql.
- Operator are act as connector between condition.
- Syntax :-

Operator SQL-Operand.

* Special Operators :-

- Special operator used to decrease condition in where clause.
- By using this it improve performance while retriving.

1) In :-

- In used to compare given fixed list of value.
- Supports all datatypes.

Ex. Select * from student where std-id IN (11, 13, 15);

2) Not in :-

- Not in used to compare otherthan given fixed list of value.
- support all datatype.

Ex. SELECT * FROM student WHERE std-age NOT IN (20, 18, 17);

3) Between :-

- Between used to compare the value within range.
- Support no. of Date datatype

Ex. SELECT * FROM student WHERE std-age BETWEEN 18 AND 20;

4) Not Between :-

- Not between used to compare out of given range of value.
-

Ex. SELECT * FROM student WHERE std-age NOT BETWEEN 18 AND 20;

* Cursors :-
- Cursor is one of the private context area.
- It create a temporary buffer of hold transactional data.
- Cursor are created by using select statment output.
- Cursor ~~block~~ program are anonymus block program.

* Explicit Cursor :-
- Cursor define by user explicitly.
- To write explicit cursor program user must know cursor operation of cursor attribute.
- Cursor operation.

① Decleare cursor
We decleare cursor
syntax.
Cursor <cursor_name> is <select_statment>;

② open cursor
It open cursor
open <cursor_name>;

③ fetch cursor
- It fetch ~~cursor~~ data from cursor into PL/SQL variable
- fetch record one at a time.
<Fetch <curs_name> into <PL/SQL var>;

④ close.
It close cursor.
- Close <curs_name>

* Attribute :-
It return status of cursor.
① %. open - return true if cursor open sucefully.
② %. Found - retur true if cursor contain data
③ %. Not found - retu fa cur desn't contain any data
④ %. rowcount - no. of. fecth statem found / Execute

Q.

* DDL -
- DDL stand for Data Definition Language
- By use of this language we can create a table or modify structure of table.
- It has 5 comand :-

i) Create
- Used to create new DB object
- Create table <table-Name> (col1 dt(size), col2 dt(size, ...);

Ex Create table studend (std-id int(2), std-Name varchar2(20));

ii) alter :-
- Used to modify structure of table.
- work on columns structure only.

Alter table emp add (Job varchar2(10));

iii) drop
- Used to delete all records and structure from DB permenatly.
- Rollback not supported
- Table is not ~~avicit~~ avilable
- Drop Table <table-Name>;
- drop table employ123;

iv) Truncate :-
- Used to delete all record from a table permenantly
- Table structure is available
- Rollback not supported

turncat table <table-name>;
tumcat table student.

v) Rename :-
Used to rename table
rename <old-table-name> To <new-Table-name>;
rename emp To emp1;

Q.

* DML :-
- Stand for Data Manipulation language.
- By using this we can manipulate existing data.

i) insert
- this command used to insert new record in existing table.
- insert into <table_name> Values (Data1, Data2);
- insert into emp values (emp_id int(2), emp_name varchar2(20));
- To insert multiple records continously
- insert into emp values (&empno, &empname);

ii) update
- To update table data,
- we can update single or group column at onces.

Update <table_name> set &col2 = <val2> [<col2> = <val2> ...
where <condition>];

- update emp set ename = "raj", sal = 3000 where empno = 1002;

iii) Delete
- used to delete a record or particular record temp.
- Table avilable
- Rollback supported.
- delete from <table name> [where <condition>];
- delete from emp;

PAGE NO.
DATE / /

Q What is transaction ? Explain Acid Property of Transaction.

* Transaction :-
- A transaction is unit of program execution that access and possibly updates various data items.
- Transaction access data by read and write operation.
- In order to maintain consistency in database, before & after transaction, certain properties followed.
- This properties known as 'ACID property'.

Eg Transaction of 50Rs transfer From Account A to Account B.

read (A)	read (B)
A := A - 50	B := B + 50
write (A)	write (B)

* ACID properties :-

i) Atomicity :- (All or nothing rule) :-
- Either all operations of the transaction are properly reflected in the database or none of.
- Each transaction is consider as one unit and either runs to complition or not executed at all.
- Involes two operation :- *Extra
  i) Abbort :-
  If transaction abbort, change made to database not visiable.
  ii) Commit :-
  If transaction commit, change made to database visiable.

ii) Consistency :-
- This mean that intergity constraint must remained / maintened.
- So data is consistant before of after transaction.

iii) Isolation :-
- This ensure that multiple ~~process~~ transaction occure concurrently.
- Each transaction must be unaware of other concurrently transaction
- One concurrently transaction result must be hidden From other concurrently occuring transaction

iv) Durability :-
- This ensure that onces transaction done his execution, update & modification to database, stored or written in disk as if system failure occure they will safe.
- Upade become perment in non-volatile memory.

Q. What is Transaction ? Explain Transaction State ?

* Transaction :-
  // Written in privious question

* Transaction State :-
- State of transaction defined by its current performing activity.
- states :-

i) Active
- This is initial state.
- Transaction is executing in this state.

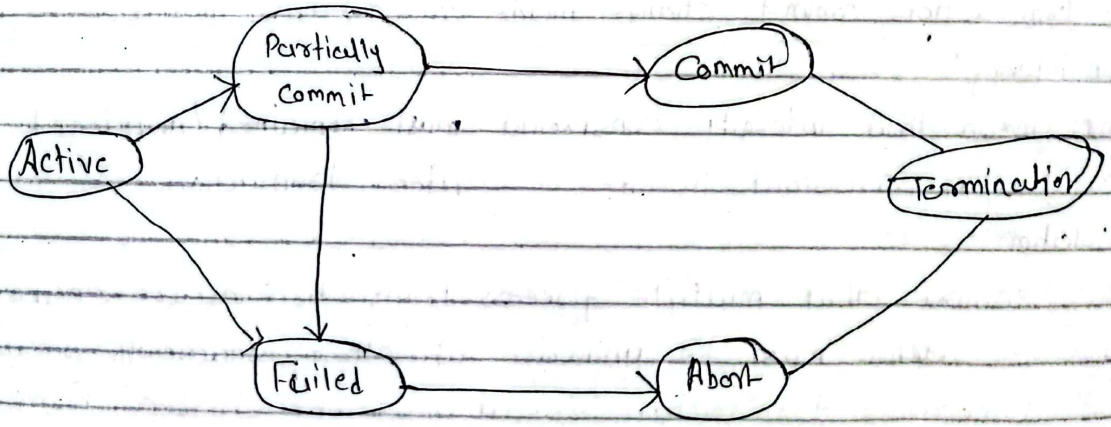
ii) Aborted :-
- Changed made by transaction cancelled. (RollBack).
- After transaction rollback and database restored to it state before transaction start.

iii) Commited :-
- Transaction completed the execution.

iv) Failed :-
- Transaction failed to complete successfully.

v) Partialy Commit :-
- Final statment of transaction completed./executed.

vi) Terminated :-
- Transaction is finished.



Q. Discuss log based recovery in brief?

* A log kept on stable storage.
- A log based recovery is technique used in DBMS to ensure data consistency and integrity at event of failure.
- It involves use of transaction log to restore a database to consiste state after failure.
- In database system, All transaction recorded in ~~file~~ log file that stored in non-volatile memory.
- In this file all change made in database while transaction is stored.
- Also old & new value of data affect by transaction is stored.
- When failure occure, the recovery process involves restoring of database to it consist state by roll back.
- This done by analyzing transaction log & apply appropiate change to database.
- There are two phase :-

i) Undo phase :-
- In this phase, any incompleted transaction is rollebacked by reversing change they made in database.

ii) Redo phase :-
- All committed transaction are reapplied to database to ensure that it is up to date.
- It involves analyzing transaction log & applying any change that were not yet written to database before the failure occured.
- log-based recovery is fundamental feature of mordern database system.

Q Explain two phase locking :-

* ~~The two phase locking protocol divides the execution of process into three parts.~~ two

* This is a protocol which ensure conflict-serialization schedules.

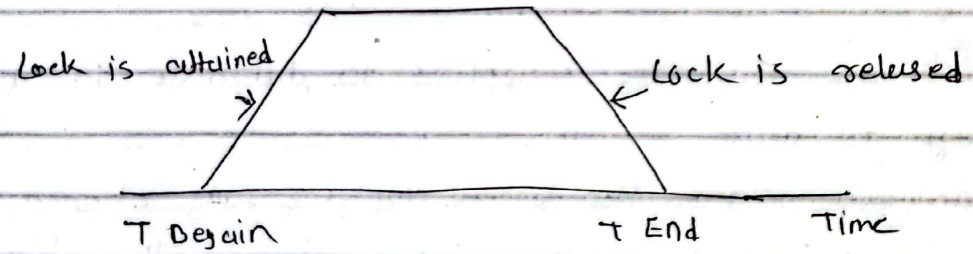
- There are 2 phase of 2PL :-

i) Growing phase :-

- Transaction may obtain lock.
- Transaction may not release lock.

ii) Shrinking phase

- Transaction may release lock.
- Transaction may not obtain lock
- Two phase-locking protocol ensure serializability of transaction.
- It proved that the transaction can be serialized in order to their lock-point.
- It gurentee that no transaction interfears with another. So data consistency is maintained.
- There ~~is~~ can be deadlock in two-phase locking.



Q What is paging ? Explain shadow pagging.

* Pagging :-

- Pagging is a technique of dividing database into fixed size block or page.
- This pages loaded into memory as need by DBMS.
- When query or transaction ~~occure~~ access page, it loaded into memory ~~from disk~~.

* Shadow pagging.

- Shadow pagging is alternative to log-based recovery.
- It maintain two page table during lifetime of transaction current page & shadow page.
- It stored shadow page in non-volatile memory. As if it can recover to it prepior transaction state.
- Shadow page never modified during execution.
- Only current page table is used to data access during transaction execution

Q. List of explain various database users?

→ There are several database user with different levels of privilegs and access to database.

- Some common type of database :-

i) Superuser :- The superusers also know as DBA, is
- It is responsible for managing all database of maintaning.
- They have full access of database of perform administrative task.

ii) Developers :- They create of maintain application that intract with Database.
- They have read/write permission of table that they need in applicati

iii) Enduser :- They use database to retrive of analize data.
- Typically have limited access. of can only read data.

iv) Application user :- They access data the database through an application
- They have limited access of specific part of database. depending on permission set by system administrator.

v) External user :- They access database by outside of organisation.
- They may be patners, suppliers, who need specific database part. access for certain task

vi) Anonymos Users :- They access database without providing any logi
- limited access, can perform simple quergs.