

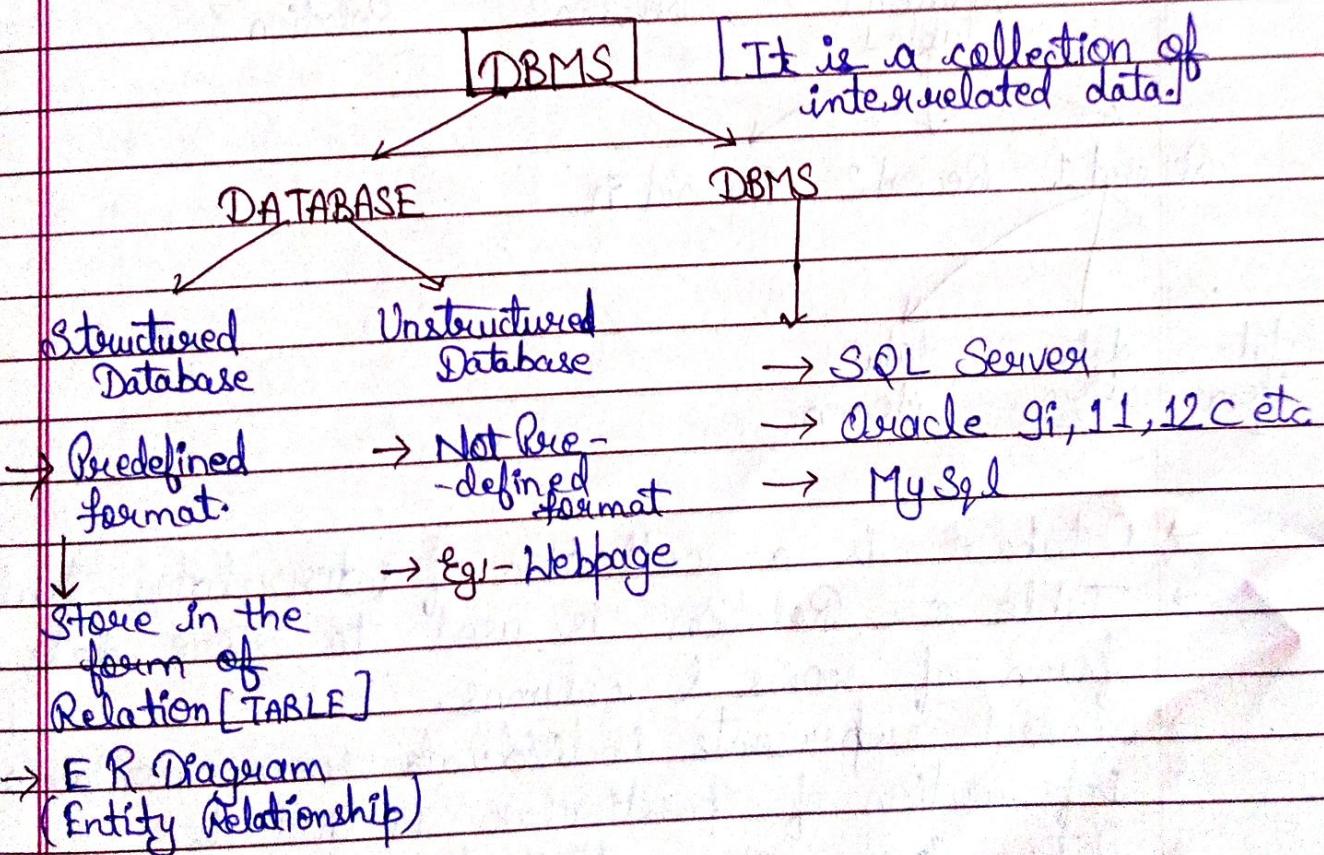
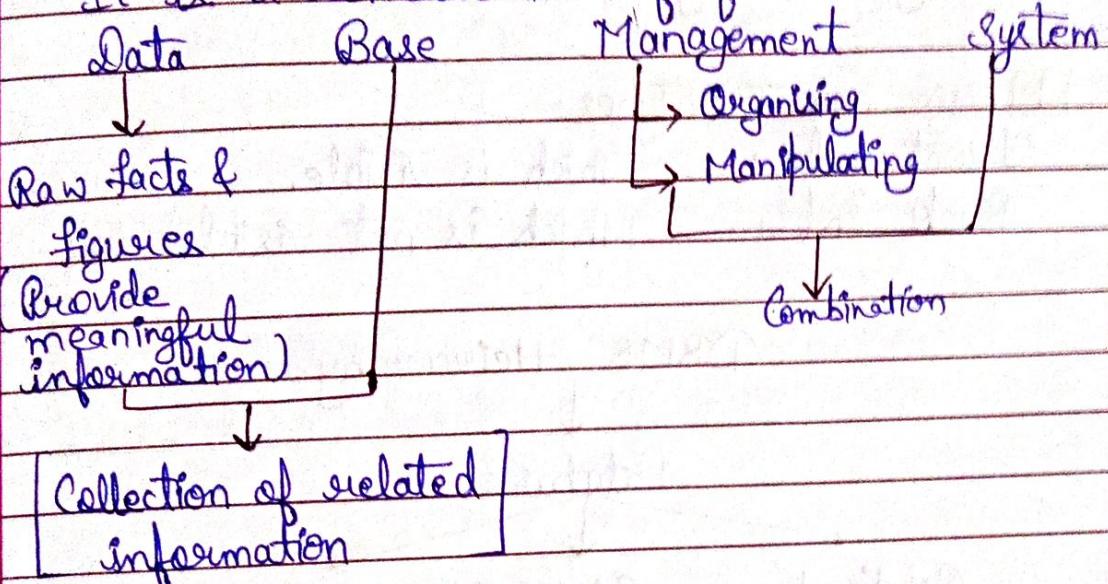
13/3/23

Page :
Date :

Database Management System

* DBMS

It is a combination of four words.



* Operations on Database

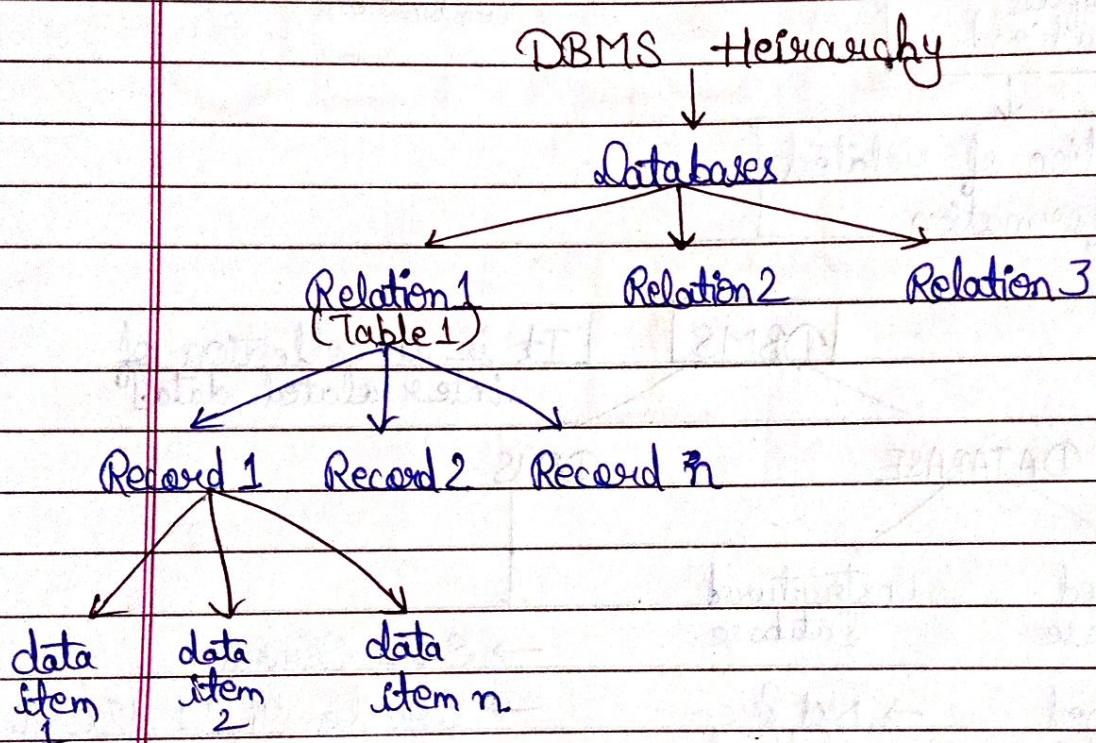
[DBMS will help to do that all operations]

- Read / Retrieving
- Writing
- Modification / Updation
- Erasing / Deleting.

* Webpage are Two Types :-

Front End :- Which is visible.

Back End :- Which is not visible.



- Database is a collection of interrelated tables.
- Table or Relation is used to store data in the form of rows & columns.
- Record represents individual rows containing information of particular individual.
- Data item or attributes is a single elementary unit of information.

Table: Student *data items*

	Enrollment	Name	Course	Age
Record 1	123	XYZ	BCA	
Record 2	124	ABC	BCA	
Record 3	125	PQR	BCA	

→ Select * from student;

→ Create database < database name >;

Eg:- Create database student;

→ Show database;

→ Create Table < Table name >

Create Table student_1
 (Enrollment_no Integer(5),
 Name varchar(30),
 Course varchar(10),
 Age integer(20));

Describle student; or Desc student_1;

→ Insert into student_1

values (12, 'XYZ', 'BCA', 19); 1 row created

→ Insert into student_2

values (13, 'ABC', 'BCA', 21); 2 rows created

Select * from student;

(You will get all your records in a table.)

Q. Difference between DBMS and RDBMS

DBMS / File System	RDBMS
→ It stores data in the form of text or paragraph.	→ It stores data in the form of relation i.e., Table.
→ It does not support client server architecture.	→ It supports client server architecture.
→ Normalization process is unavailable.	→ Normalization can be done with relation.
→ No ACID (Atomicity, consistency, Isolation and Durability) property is present.	→ A CTD is present.
→ Low software and hardware is required to implement.	→ High software and hardware is required to implement.

Q.

Q. What is Database?

⇒ Database is a collection of inter-related data which is used to retrieve, insert and delete the data efficiently. It is also used to organize the data in the form of table, schema, view and report etc.

⇒ Database Management System

* It is a software which is used to manage the database. For eg:- MySQL, Oracle etc are a very popular commercial database which is used in different application.

- * It provides an interface to perform various operation like database creation, storing data in it, updating data, creating a table etc.
- * It provides protection and security to the database in case of multiple users it also maintains data consistency.

20/3/23

Data Models

Way to represent or Model your data

→ Relational Data Model :- It will represent the data is in the form of Relations / Tables.

(v.i) → Entity Relationship Model :- It will represent the data is in the form of Entities and its relation.

→ Object Base Model :- It will represent the data is in the form of object like class, Encapsulations, etc
(Data hiding)

- * RDBMS [Relational Database Management System]
 - Relations / Tables
 - Tuples / Records
 - ↳ No duplicate Tuples
 - ↳ Follows same structure or format
 - Attributes
 - ↳ All attributes must have the Name

- Date item / Cell Data

↳ Intersection of Tuples and Attributes.

- Degree & Cardinality

Degree :- No. of attributes (columns) in a Table.

Cardinality :- No. of Tuples (Rows) in a Table.

Attributes



Student Table

	Stu. Enrollment no.	Name	Percentage
Tuples → 	101	XYZ	97
	102	ABC (datatype)	98
	103	PQR	99
Tuples →	104	DEF	100

Attribute 1

Attribute 2

Attribute 3

① Define Cardinality & Degree of a Table Student.

⇒ Cardinality → 4.

Degree → 3.

* Domain :- Set of possible values for an attribute.

Eg:- Percentage (0 - 100)

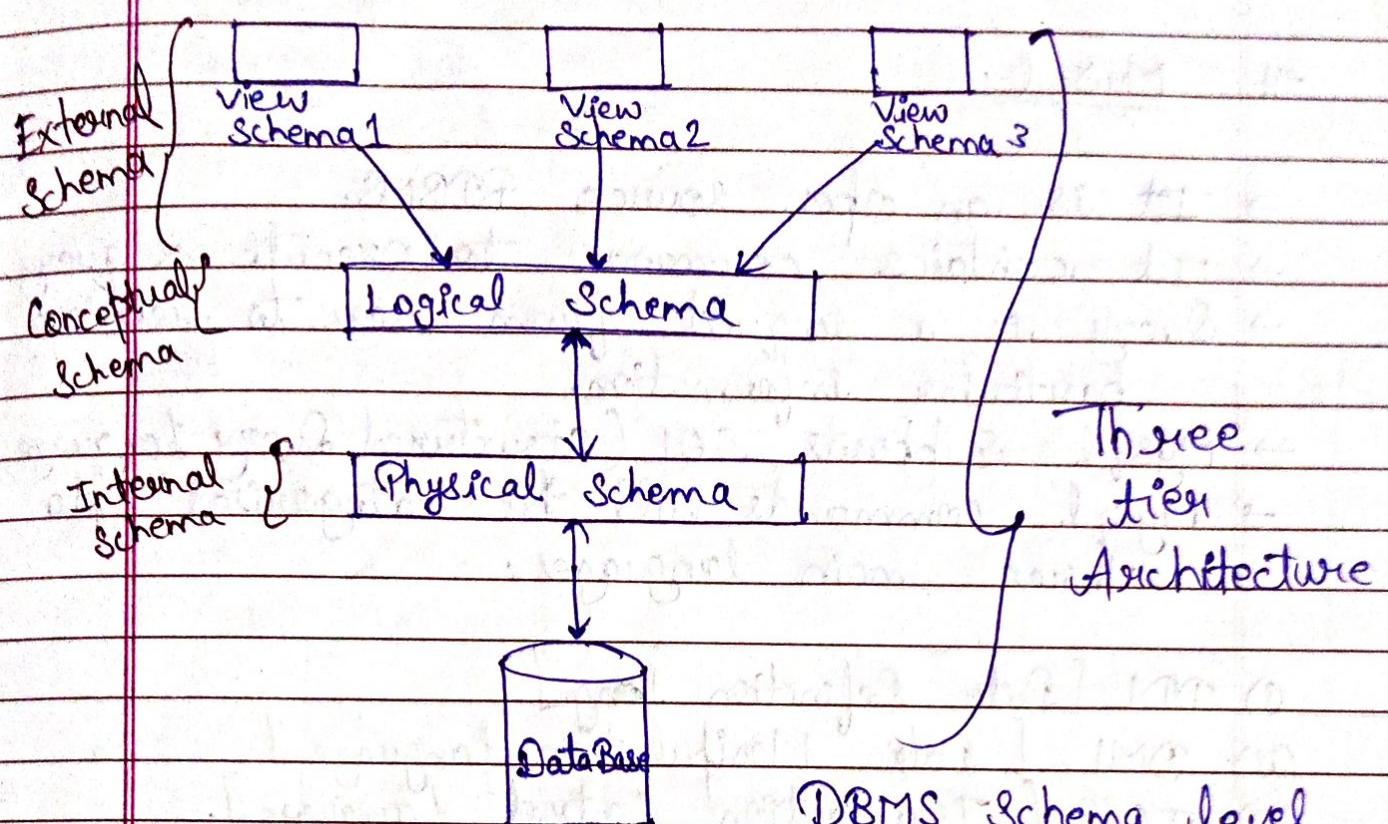
↓
Domain

* NULL Value :- When we kept any column / attribute left during insertion.

→ It will store NULL.

→ NULL is neither zero or not space (blank).

* Schema & Instances in DBMS / DBMS Architecture



- Schema is basically a structure of database.
- It is divided into three level.
 - a) Physical schema
 - b) Logical schema
 - c) View schema.
- How the data is going to be store in storage, e.g:- files/folders.
- It is the actual processing of data in terms of query generated by end user.
- It is an user interface to interact with the system.

24/3/23

* Instances :-

MySQL :-

- It is an open source RDBMS.
- It contains commands to execute a query.
- Query is a request from user to retrieve particular information.
- MySQL supports SQL (Structured Query Language).
- MySQL commands will be categorised into three main languages:

- (i) DDL [Data Definition Lang.]
- (ii) DML [Data Manipulation Language]
- (iii) TCL [Transaction Control Language].

Q. Diff. b/w DBMS / RDBMS.
Q. Define Schema with an example.
Q. DBMS Architecture / Schema level / 3 tier architecture.

17/3/23

* Keys in DBMS :-

It is an attribute or set of attributes that uniquely identifies a record (or tuple) within a relation or a table.

Type of Keys:-

- Super Key
- Candidate Key

- Primary key
- Alternate key
- Foreign key
- Composite key.

Consider the following Table : Employee

Ep-ID	Ename	E-Adhar	E-Phone	E-Dept-code
101	Ajay			
102	Neha			
103	Neha			
104	Abhay			

- Super key

- It is all possible set of attributes that can be uniquely identifies a record / tuple from a relation.
- It is a superset of candidate keys.
- It is a combination of extra attributes to identify a record / Table
- It may be allowed to attributes can be NULL.

Superkey → Candidate key → Primary key → Alternate key.

- Candidate key

- They are subset of superkey.
- It is also used to uniquely identify a record / tuple from the Table.
- It is a minimal set of super keys i.e, no redundant attributes will be allowed.
- Null values are not acceptable to candidate key.

- Primary key
 - It is an unique key.
 - It is used to uniquely identify a record/tuple from the relation.
 - It can't be NULL.
 - We can assign constraint while inputting a value in a table.
 - Duplicate values are not allowed.
 - Selected from the set of candidate key.

- Alternate key

- Out of all candidate key, rest keys are alternate keys except that who is designed as Primary key.
- It can be assigned as Primary key.

- Foreign key

- It is a primary key of another table.
- It is used in referential Integrity.

- Composite Keys

- Combination of one or more attributes that will be uniquely identifies a record/tuple in a relation.
- It is also called compound keys.
- It can be a primary key.
- NULL value is not acceptable while insertion.

Data Models

ER- Model

ER diagrams (ERDs)

(Rectangle)
Entity

(Ellipse)
Attributes

(Diamond)
Relationship

Strong

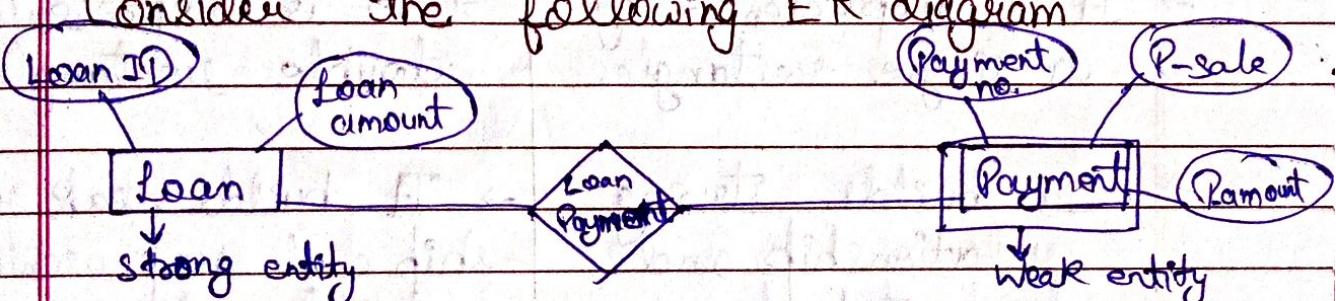
Weak

- Simple and Composite
- Single and Multivalued
- Shared and Derived.

→ It is an association among entities

→ Types of relationship.

Consider the following ER diagram



Loan ID	Loan Amount
L1	1,00,000
L2	2,00,000
L3	1,50,000
L4	5,00,000

Payment	P date	P amount
1	17-08-2023	1,50,000
1	29-02-2023	10,000
1	05-04-2023	20,000
2		
2		

Partial
key

NOTE If we write the ~~any~~ data in a double rectangle so, it will be called a weak entity.

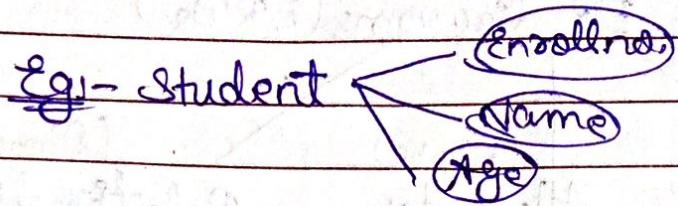
Name

Marks

Student

Subject

Entity - Entities are thing or object in real working environment that has certain properties.



* Strong Entities

- It has a primary key.
- It is not dependent on any other entities.
- It is represented by a single rectangle.

Weak

- It has a partial / discriminator key
- It is dependent on strong entities.
- It is represented by double rectangle.

- It builds strong relationship and represented by a single diamond.

- It builds weak relationship and represented by double diamond.

Atomic - The .

- * Simple Attribute - They are atomic it means it is not divide further.
Eg:- age, enrollment no.

* Composite Attributes :- These are further divided

Eg:- name, address.

* Multi-value attributes :- It means the table stores the multi data.

It is represented by two ovals i.e.,

Eg:- E-mail, Phone number.

* Single Attribute :- It contains a single value.

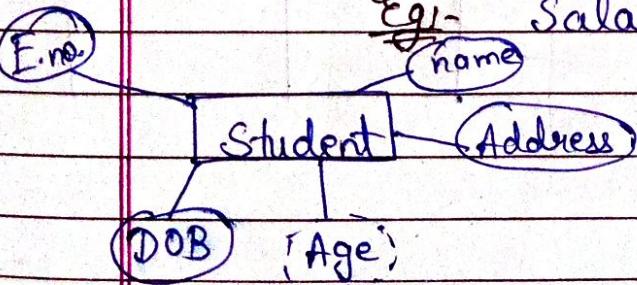
Eg:- enrollment no., Aadhar no.

* Stored Attributes :- These are physically stored in a database. All attributes are almost comes under this category.

For eg.:- Roll no., age

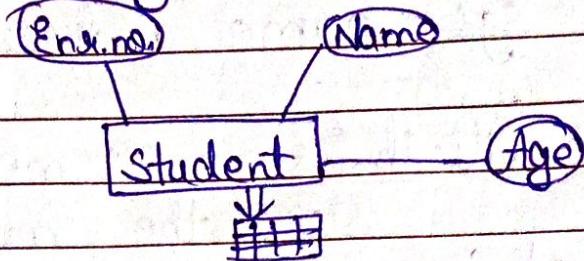
* Derived Attributes :- It is derive from any other stored attributes.

Eg:- Salary.

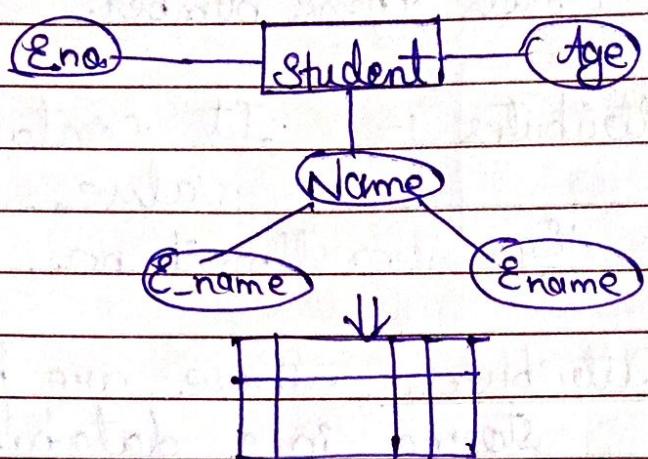


* Entities :

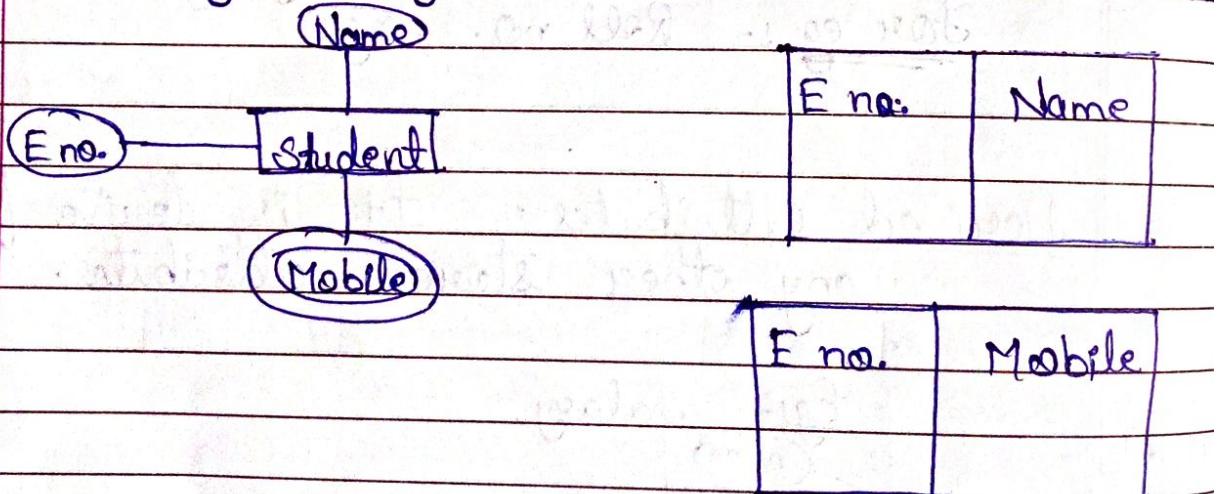
Rule 1: Strong entity set with single attribute



Rule 2: Strong entity with composite attribute.



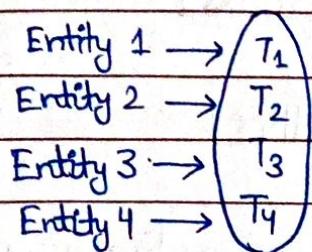
Rule 3: Strong Entity set with multivalued attribute



* Relationship :- An association among entities.

* Entity Set :- It represents set of distinct entities that have common or same properties.

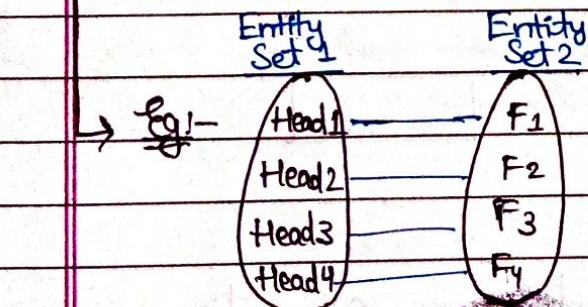
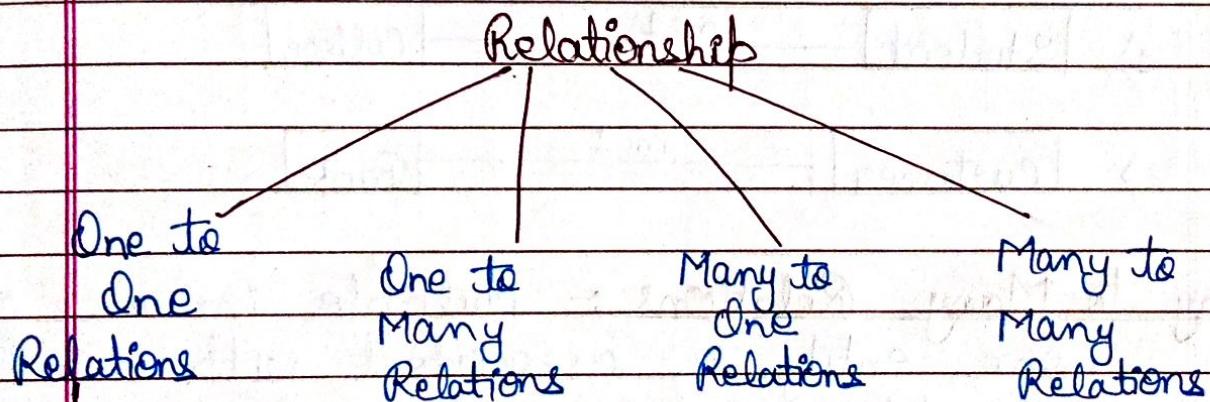
Eg:-



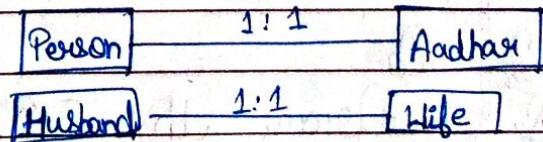
Teachers

T-ID	Teacher_name	T_specification	DOB	Location
------	--------------	-----------------	-----	----------

Teacher Entities
Set

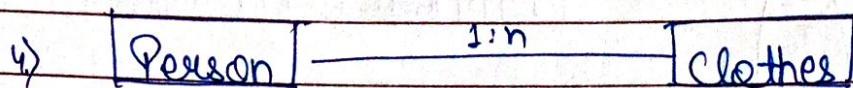
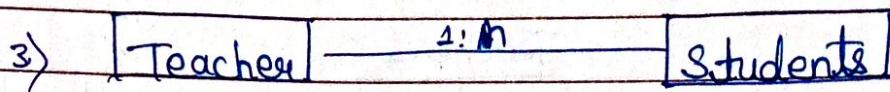
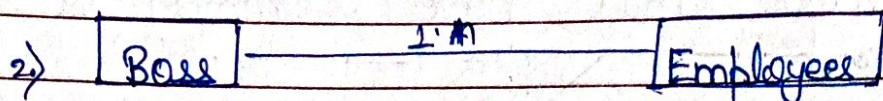
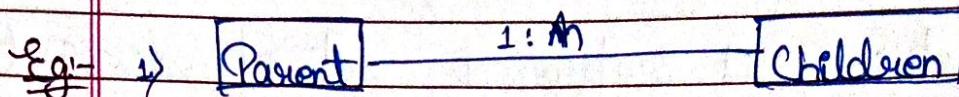


Eg:- Pictorial Representation

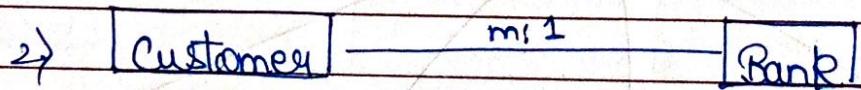
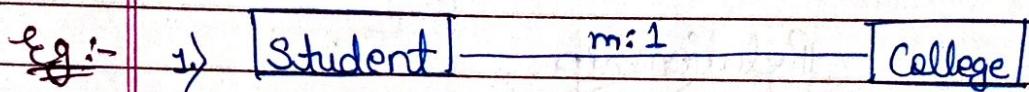


* One to One Relations :- A one to one relation means single instance of one entity is associated with another entity.

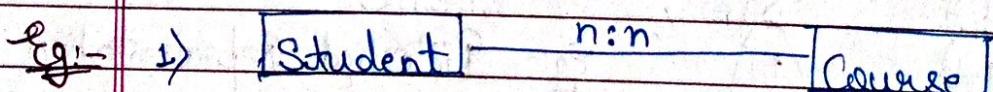
* One to Many Relations :- Single instance of one entity is associated with multiple entities.



* Many to One Relations :- Multiple instances of one entity is associated with ~~one~~ single entity.



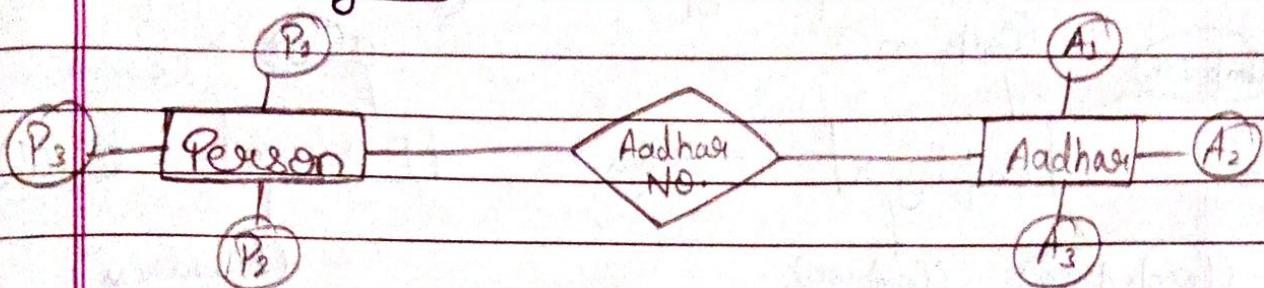
* Many to Many Relations :- Multiple instances of one entity is associated with multiple entities.



Ques Draw the ER model of one to one relation, one to many relationship, many to one and many to many.

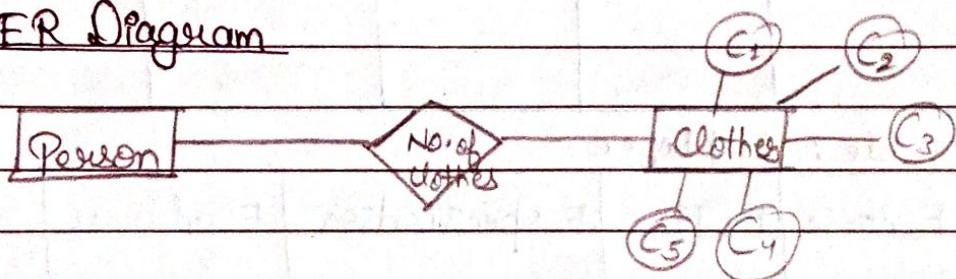
→ 1) One to One Relationship:-

ER Diagram



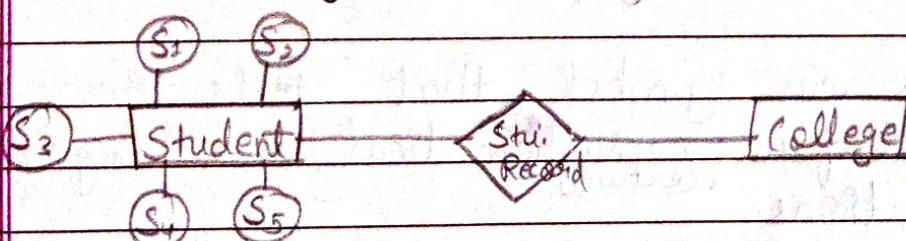
2) One to Many Relationship:-

ER Diagram



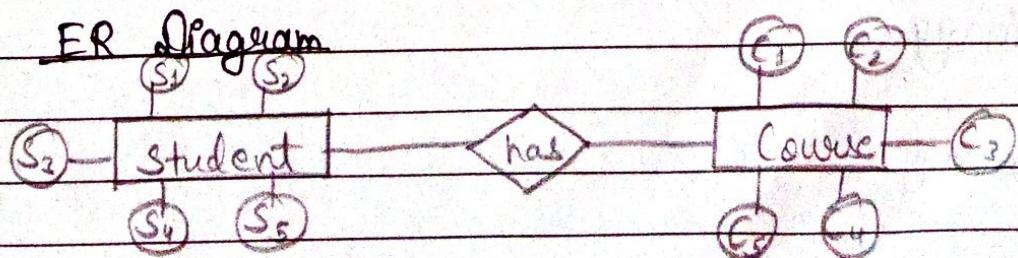
3) Many to One Relations :-

ER Diagram



4.) Many to Many Relations :-

ER Diagram



- Q. Consider the following ER Diagram and Map it in to Table and define Relationship type during the implementation.

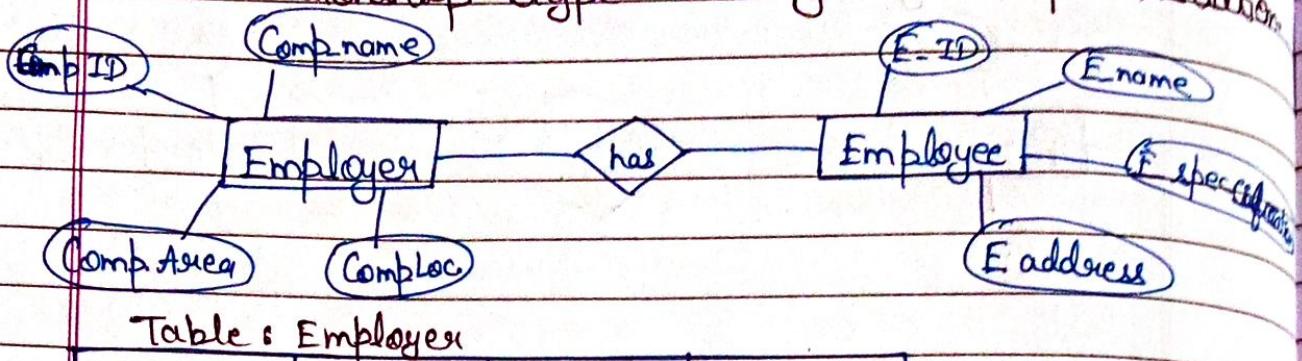


Table : Employer

Comp name	Comp loc.	Comp ID	Comp area

It is an
One-to-many
relations

Table : Employee

E name	E ID	F specification	F address

* Operators in MySql

→ It is a symbol that tells the computer to perform ^{certain} mathematical and logical operations.

- Unary

- Binary

- Ternary

Table: Employee

Eno	Ename	Job	Sal
118	RAJ	Manager	2000
196	ANKIT	Salesman	1250
182	RADHIKA	Analyst	3200
147	PAYAL	Manager	2450
156	MANOJ	President	5000
123	KRUPA	Saleman	1600
178	NEHA	Manager	2975

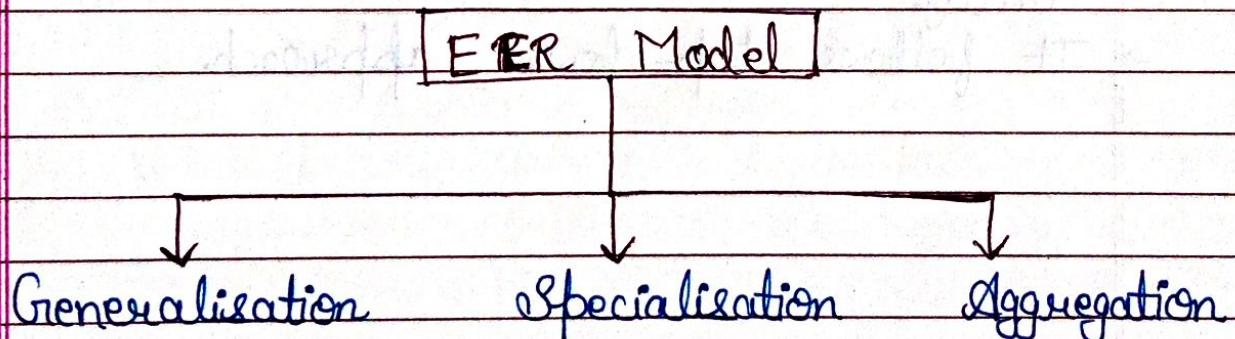
7 rows in a set (0.01 sec)

5/4/23

* Extended ER Model

It is an extended version of traditional ER model. Traditional ER models suffice with problem during the complex situation. So we add few new concepts within the existing ER model.

These concepts are :-

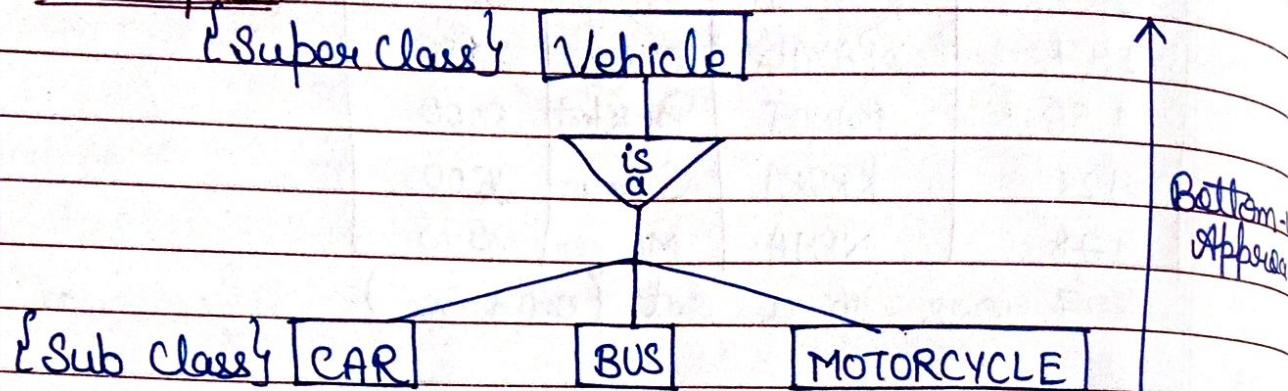


Generalisation :-

- It involves multiple entities and combine them into a single generalised entity.
- It reduces the size of the 'schema.'

Page :
Date :

- In generalisation a higher level entity must have low level entity.
- It follows bottom up approach.
- Example:-

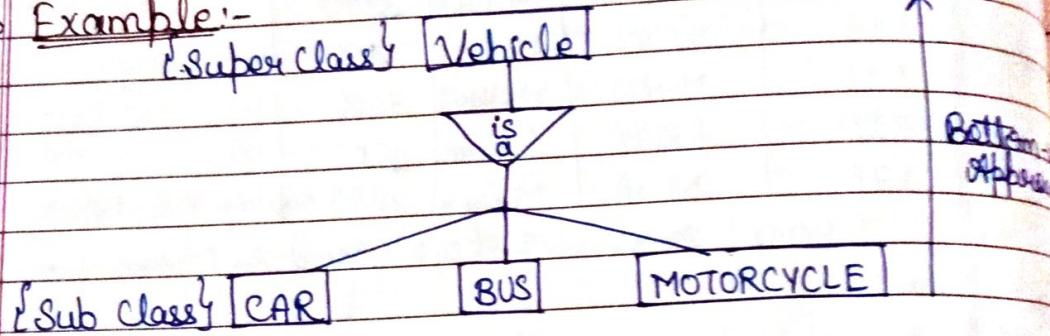


Specialisation :-

- It is the reverse process of generalisation.
- It involves single entity and combine them into multiple entities.
- It increases the size of 'schema'.
- Higher level entity may not have low level entity.
- It follows top-down approach.

- In generalisation a higher level entity must have low level entity.
- It follows bottom up approach.

→ Example:-



Specialisation :-

- It is the reverse process of generalisation.
- It involves single entity and combine them into multiple entities.
- It increases the size of 'schema'.
- Higher level entity may not have low level entity.
- It follows top down approach.

MySQL
↓
DATA TYPES

Language

* Lit

→ T
2

→ H

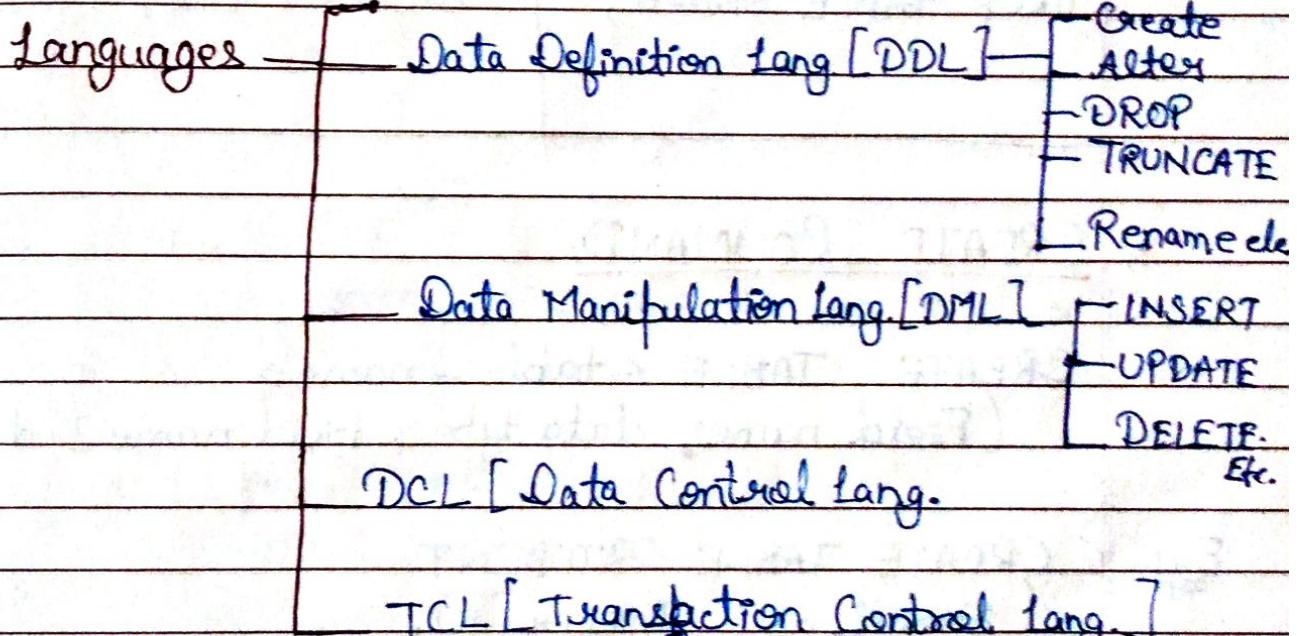
10/4/23

MySQL → SQL [Structured Query Language]

DATA TYPES — Numbers → integer, number, numeric, int, float, real, decimal

Char/String → char, varchar or varchars, text

Data → Date, [mm - dd - yyyy], Year,



* Difference b/w DDL & DML:-

DDL

→ It stands for Data Definition Lang. (DDL).

→ It is used to define the structure of the database.

DML

→ It stands for Data Manipulation Lang. (DML).

→ It is used to modify or access the existing table.

Page :
Date :

→ It works with entire table.	→ It works on one or more records of the table.
→ DDL commands are create, alter, drop etc.	→ DML Commands are insert, delete, update etc.
→ Eg:- DROP TABLE <table_name>; DROP TABLE Student;	→ Eg:- SELECT * FROM <table_name>; SELECT * FROM Student; SELECT Eno, Name FROM Student;

→ CREATE COMMAND

CREATE TABLE <table-name>
(Field name₁, datatype, Field name₂ datatype);

Eg:- CREATE TABLE STUDENT
(Enroll_no Integer(5),
Name varchar(25),
Course varchar(10));

⇒ ALTER COMMAND

- It is used to modify the existing table

Syntax :- ALTER TABLE <table-name>;

- It always work with 3 keywords

(i) ADD :- It is used to add a new column into existing table

(ii) MODIFY :-

(iii) DML :-

Syntax of - (i) ADD :-
ADD column_name datatype;

Eg:-

DE SCR

DES
(It

Syntax of
MODIFY :-

Syntax of
DROP :-

⇒ DRO

This
is
Com

DRC
DRC

(iii) DROP :-

Syntax of 1) ALTER TABLE Student

ADD :- ADD <column-name datatype>;

Eg:- ALTER TABLE Student
ADD Age integer(3);

DESCRIBE Student;

or

DESC Student;

(It shows table structure)

Syntax of 2) ALTER TABLE Student

MODIFY :- MODIFY COLUMN Name varchar(33);

Syntax of 3) ALTER TABLE Student

DROP :- DROP COLUMN AGE;

⇒ DROP COMMAND

- This command is used to drop a existing table completely.

Completely [Table structure as well as within it]

Eg:- DROP TABLE <Table-name>

DROP TABLE student;

Page : ← →
Date : ← →

⇒ TRUNCATE EA COMMAND

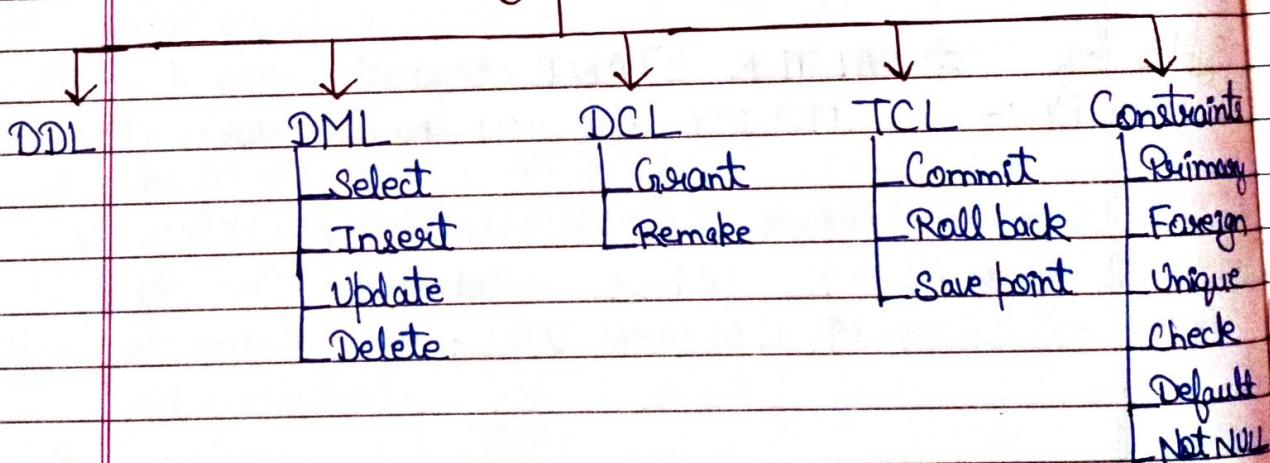
- It is also used to delete the records from the table.

Syntax - TRUNCATE TABLE < Table-name >
TRUNCATE TABLE Student;

- This command will lock the entire table and delete all the records associated with that table.

11/4/23

MySQL



Aggregate Function in MySQL

Min :- To find minimum salary from table.

Syntax :- Select min (salary) from Employee;

Max :- To find maximum from the table

Syntax :- Select max (age) from Employee;

Count - It is used to count the number of rows/records from the table.

Syntax → Select count (*) From Employee;

→ Select distinct (Count (E_Salary)) From Employee;
→ It will exclude null during the count of the specific field.

Avg - Calculate the average value from the table

Syntax → Select Avg (Salary) From Employee;

Sum - It is used to sum the value of all the values.

Syntax → Select sum (Salary) From Employee;

→ Update Employee

Set Salary = Salary + Salary * 0.02;

→ It is DML Command.

Q. Write a command to modify all employee.

→ Update Emp.

Set E_Age = 31;

12/4/23

Page :
Date :

Q.	EMP_ID	ENAME	DEPTT	SALARY
1	1	HARRY	HR	10000
2	2	PETER	MRKT	20000
3	3	SUNNY	HR	30000
4	4	MUNNY	MRKT	40000
5	5	FUNNY	IT	50000

a.) Write a SQL query to display a highest salary from the Employee table?

⇒ Where Salary = (Select max(salary) From Employee); 50000

b.) Display the name of the employee who is taking a highest salary?

⇒ Where Salary = (Select max(salary) From Employee);
Select Ename From Employee;

c.) Write a SQL query to display second highest salary from the Employee table?

⇒ Select max(salary) From Employee (where salary != (Select max(salary) From Employee));

d.) Write a SQL query to display the name of the employee who is taking second highest salary.

⇒ Select Ename From Employee where salary =
(Select max(salary) From Employee) = Salary !=
(Select max(salary) From Employee);

e.) Write a query to display all the deptt names along with no. of employees working in that?

Order By

- It is used to display the information either in ascending order or in descending order.
- It is not mandatory to use Aggregate functions in order to use order by command.
- Example:-

Select * From Student
order by MARKS;

Group By

- It is used to group the data on the basis of particular column.

- It is mandatory to use Aggregate function in order to use Group by command.

- Example:-

Select Stream, max(marks)
From Student Group by
Stream;

* WHERE

- The WHERE clause specifies the criteria which individual records must meet to be selected by a query.

- The WHERE clause selects rows before grouping.

- The WHERE clause cannot contain aggregate function.

- Ex:- Select * From student
Where MARKS > 40;

HAVING

- The HAVING clause cannot be used without the GROUP By Command.

- The HAVING clause selects rows after grouping.

- The HAVING clause can contain aggregate function.

- Ex:- Select Stream, MAX(Marks)
From Students
Group By Stream.

* Query

- e.) Write a query to display all the dept names along with no. of employees working in that?

Sol

Select dept count (*) From Emb.

GROUP BY Deptt.

- f.) Write a query to display all the dept names where no. of employees are less than 2;

Sol

Select distinct (dept) from employee GROUP BY deptt. having count(deptt) < 2;

Sol

- g.) Write a query to display the name of employee where no. of employees are less than 2;

SELECT Ename FROM Employee
Where deptt In (Select deptt From Employee
GROUP BY Deptt having count(*) < 2);

 * IN and NOT IN Operators in Sql :-

⇒ It is used to specify multiple values in WHERE clause.

Syntax :-

SELECT col1, col2 -

FROM <Table-name>

WHERE COLUMN-NAME IN (Value1, Value2);

WHERE COLUMN-NAME NOT IN (Value1, Value2);

Q1 Find the name of students whose age is 20 or 21.

Table : Student

S-ID	NAME	AGE	COURSE	CITY
1	Funny	20	BCA	New Delhi
2	Munny	20	BCA	Agra
3	Sunny	19	MCA	Noida
4	Gunny	21	BCA	New Delhi
5	Hunny	22	B.Sc.	Agra

Sol 1 SELECT Name From Students

WHERE AGE = 20 OR AGE = 21;

OR

SELECT Name From Student

WHERE AGE IN (20, 21);

Q2 Find the record whose city is New Delhi
or Noida?

Sol Select * From Students

where City IN ('New Delhi', 'Noida');

Q3 Find the student ID name of those students
whose course is BCA or MCA?

Sol Select ID, Name From Students

where Course IN ('BCA', 'MCA');

18/4/23

Page :
Date :

Q1 Write a query to find the name of those students whose first alphabet starts with S.

~~Ans~~

* Like Operator & Not like Operator

→ It is used with where clause to search a specific pattern from the table.

→ It will used two wildcard character to accomplish that.

• % (Percentage) - It will go with zero, single or multiple characters.

• _ (Underscore) - It will go with only a single character.

Sol1 Select name From student
where name like 'S%';

→ %a% :- 'a' place at any position.

→ %a_ :- 'a' will display at second last position.

→ a% :- It will display at starting position.

→ __a :- 'a' will display at third position.

→ %__a :- 'a' will display at fourth last position.

→ ___a% :- 'a' will display at second position.

Deeps

→ %a%
co

→ %a_ =

→ a%_

→ __a

→ %a_

→ _a%

Q. Consider
ope
stud

Roll

1

2

3

4

5

6

1) Ti

2) Ti

3) G

4) G

5) G

Deebanshu

→ `%a%` = It will display string that would contain 'a' in string.

→ `%a` = Deepan

→ `a%` = No output.

→ `--a` = No output.

→ `*%a--` = Deepans

→ `-a%` = No output.

Q. Consider the following table and perform operations:

Student:

Roll no	Name	Course	City
1	Ajay	BCA	New Delhi
2	Ramesh	BCA	Agra
3	Suresh	MCA	New Delhi
4	Funny	MCA	Mumbai
5	Munny	BCA	Bangalore
6	Tunny	BCA	Mumbai

- 1) Find the name of student whose starts with 'S' alphabet.
- 2) Find the name of those students containing 'n' alphabet at 4th position.
- 3) Display the name of city whose 1st alphabet start with 'N' and ends with 'B'.
- 4) Find the name of those students who is pursuing MCA.
- 5) Display the name of all students whose name ends with 'y'.

Sol 1) Select name From student
Where name like 'S%';

Sol 2) Select name From student
Where name like '_ % n %'

Sol 3) Select city From student
Where city like 'N%I'

Sol 4) Select name From student
Where course like 'MCA%'.

19/4/23

Sol 5) Select name From student
Where name like '(g)y%'.

My SQL Functions

→ String / char

→ Numeric

→ Date / time

→ Char() :- It returns the ASCII value equivalent char.

→ Lcase / Lower() :- It converts the string in ^{lower} lowercase.

→ Ucase / Upper() :- It converts the string in uppercase.

→ LTRIM() :- It removes blank spaces from the begining of the string. sentences

→ RTRIM()

→ TRIM() or ALLTRIM()

→ INSTR()

→ CONCAT()

→ LENGTH()

→ SUBSTR()

* Syntax of char():- CHAR(65) → A
CHAR(57) → a

* Syntax of Lcase / Lower():- LCase('BCIIT') → bciit
Lower('BCIIT') → bciit

* Syntax of Ucase / Upper():- Ucase('Bciit') → BCIIIT
Upper('Bciit') → BCIIIT

* RTRIM()

⇒ It removes the blank spaces from the ending or last of the string.
Syntax :- \rightarrow RTRIM('BCIIT ') → BCIIIT

* TRIM() or ALLTRIM()

⇒ It removes spaces from the both end of the string.

⇒ Syntax :- TRIM(' BCIIT -- ') → BCIIIT
 \rightarrow leading space

* Syntax of LTRIM():- LTRIM(' _ BCIIT') → BCIIT
By default it will remove Blank Space.

* INSTR()

⇒ It will search the second string into first string.

⇒ Syntax :- INSTR(first string, Second string)

* CONCAT()

⇒ It will join or combine two strings.

⇒ Syntax :- CONCAT (First string, Second string).

* LENGTH () :-

⇒ It will find the length of a string i.e., no. of characters within in a string.

⇒ Syntax :- length (string)

* SUBSTR () :-

⇒ It will find / search substring from the name string.

Syntax :- Substr (str , pos , no. of char)

NOTE :- $S_1 = ('xxx BCIIT')$

LTRIM (leading x from S_1);

NOTE :- $S_2 = ('BCIIT xxx')$

RTRIM (Trailing x from S_2);

Eg:- INSTR ('BCIIT' , 'IT');

⇒ It will return the starting position of second string within the main string.

Eg:- substr ('BCIIT NEWS' , 3,3)

$$O/P = IT$$

('BCIIT NEWS' , 4,5)

IT_NE

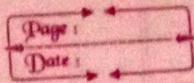
('RAVAN KUMAR' , 6,2)

$$O/P = K$$

('BCIIT NEWS' , -4,3)

O/P NEW

9/12/23



Numeric Function

mod()
pow()
round()
sign()
sqrt()

Date

curdate()
Date()
month()
month name()
day()
year()

Mod() :- This function is used to find the remainder value after division process.

Eg:- Select mod (5,2) $\rightarrow 1$

Pow() :- This function is used to raise the power of first number.

Eg:- Select pow (4, 2) $\rightarrow 4^2$
power (2, 3) $\rightarrow 2^3$

Round() :- This function is used to round off the in the integer value near to decimal places.

Eg:- Select round (15.692, 1);
 $\rightarrow 15.7$

Select round (14.7834, 3);
 $\rightarrow 14.783$.

Sign() :- This function is used to find the sign of a given number. It will return '1' for 'tve' no., '-1' for 've' no. and for '0' for '0'.

- Eg:-
 1) Select sign (-4) → 1
 2) Select sign (4) → 1

Sqrt():- This function is used to find the square root of given number.

- Eg:-
 1) Select sqrt (196) → 14
 2) Select sqrt (25) → 5
 3) Select sqrt (26) → 5
 4) Select sqrt (16) → 4

* Date Functions

Curdate():- This function is used to display the current date.

Eg:-
Select curdate();

or

Select current_date();

2023-04-21

Q Display the date after 10 days from today.
or,

What date will be after 10 days.

Ans
Select current_date() + 10;

Date():- This function is used to find the date.

Eg:-
Select date ('2023-04-21');

Select date ('2023-04-21 1:20:50').

Month

⇒ Select

Month

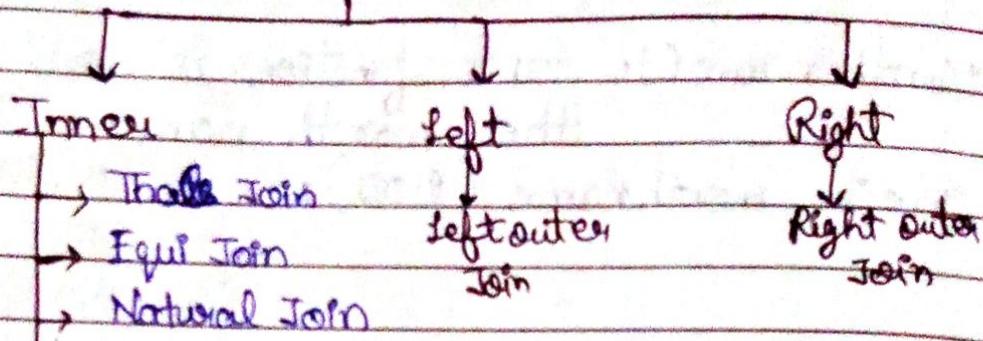
⇒ Select

Month () :- This function is used to display the month number.

⇒ Select mon (DOJ);

Month name () :- This function is used to display the month name.

⇒ Select monthname (DOJ);

JOINS

Select customer ID , Products From

INNER JOIN ORDER
ON Customer ID = ORDERID;

CUSTOMER

cust-ID	cust-name	cust-Address
1	Funny	New Delhi
2	Sunny	Jaipur
3	Tunny	Bhopal
4	Gunny	Mumbai

ORDER

Order-ID	Order-quantity	Cust-ID
11	5	1
12	6	2
13	2	3

SUPPLIER

Supp_ID	Name	Cust_ID
1	XYZ	1
2	BXY	2
3	RSG	2

>Select order quantity customer name, ID
 Using INNER JOIN , LEFT JOIN , RIGHT JOIN . &
 OUTER JOIN .

INNER JOIN Syntax :-

⇒ Select customer. cust_ID , cust_name , order -
 quantity FROM CUSTOMER
 INNER JOIN ORDER
 ON Customer. cust_ID = ORDER.cust_ID ;

② LEFT JOIN SYNTAX :-

⇒ Select .customer. cust_ID , cut_name , order_quantity
 FROM CUSTOMER
 LEFT JOIN ORDER
 ON Customer. cust_ID = ORDER.cust_ID ;

RIGHT JOIN SYNTAX :-

⇒ Select customer. cust_ID , cust

8/5/83

Page: ← →
Date: ← →

9/5/83

Name	Address	Dept	Course
a.	New Delhi	Computer	C ₁
b	Haryana	IT	C ₂
a	Punjab	Computer	C ₁
b	Haryana	Computer	C ₂
c	Mumbai	IT	C ₃

Super Keys

- 1) {Name, Address}
- 2) {Name, Address, Deptt}
- 3) {Name, Address, Deptt, Course}
- 4) {Name, Deptt}
- 5) {Name, Course}
- 6)

Referential Integrity

Entity Integrity

Foreign Key

Constraint [Conditions]

Rule of

* CONSTRAINTS:-

- Every relation has some condition that must hold for it to be a valid relation or Relational Key constraints.

* Key Constraints :-

- It is also known as Entity Constraints.
- Duplicate Values do not exist.
- Every attribute must have a specific value.

* Domain Constraints :-

- Domain refers to value of an attribute
- Every attribute must have specific range of ~~non~~ values.
- For eg:-

If we have age attribute in a relation
the specific range of values are:-

- (i) Age cannot be a '-ve.'
- (ii) Age always greater than or equal to '0.'

* Referential Integrity Constraints :-

- It works on the concept of foreign key
- A foreign key is a key that act as a reference key for another table that will be a primary key.

Integrity [Values]
Consistency [Row]

* Rule :

- 1) We can't update the information of child table which is not present in master table.
- 2) We can't delete record of master table which is not refer in child table.

Q. Write SQL Query for the queries of following table:-

a) Increase income of all employee by 5% in table employee.

⇒ Update Employee Income Set Salary = Salary +
Salary * 0.05;

b) Find the name of the employee starting with 'A'

⇒ Select name From Employee
Where name like 'A%' ;

c) Find number of employees working in department A, B, C...

⇒ Select deptt count(*) From Emp.
GROUP BY deptt.

d) Find details of employees whose first name ends with 'A' and contains 6 alphabets.

⇒ Select * From Employee
Where fname like '%A%' .