DATABASE MANAGEMENT SYSTEM

DATE: __/__/____
PAGE _____

Record: It is a collection of interrelated data.

File: It is a collection of number of records.

Database: It is a collection of inter related files.

Relationship among data, information and knowledge

→ Data is raw facts and figures.
→ Information is meaningful data
→ Knowledge is application of information towards different fields.
→ Wisdom is using information in various ways.

Data
- ① Raw Data
- ② Meta Data (data about data or details of data)

Types of database and its applications

1. Traditional
   - Info is stored in terms of numeric and text
2. Multimedia
   - Traditional + Audio + Video + Image
3. Analytical
4. Operational
5. Real-time

GOOD WRITE

DATE: __/__/____
PAGE _____

Analytical Database → Traditional + Multimedia + Data ware housing

→ Software used is OLAP
(Online Analytical Processing)
↳ It stores large amount of data with different formats.

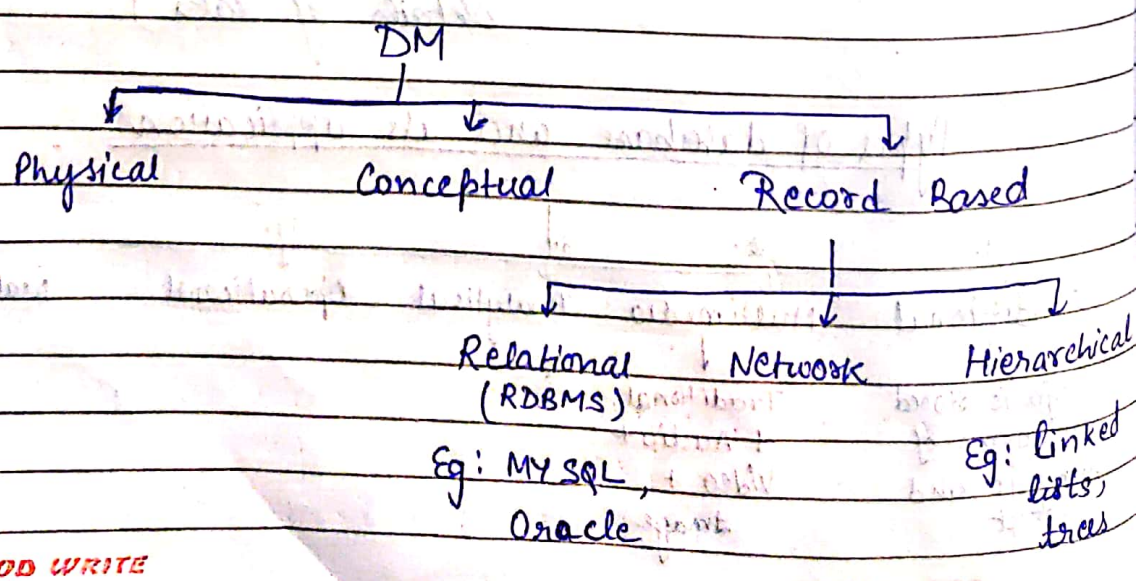
Operational Database → Software used is OLTP
(Online Transactional Processing)

[Real time data working + Dumped data]

Real time database → It only works on real-time data.
Eg: Social media websites, Bigdata

DATA MODELS

They are used for describing, defining and storing the data in DBMS.



GOOD WRITE

DATE: __/__/____
PAGE _____

PRACTICAL (MYSQL series)

```
create database college;
use college;
create table student (
rollno int,
name varchar(20),
age int,
marks int );
insert into student
values ( 1, 'Kanishma', 18, 92);
insert into student
values (2, 'Sachi', 18, 90);
select * from student;
sp_help student;          // for describing the structure of a table
```

Database Management System (DBMS)

→ A DBMS is a collection of inter related data and set of programs to access those data.

→ It is a general purpose application software that provides an environment that is both convenient and efficient to use in each of the following (with respect to the database)

a) Defining a database – i.e. specifying the structure of the database / table / view / index which includes data types, constraints specification.

GOOD WRITE

DDL → Data definition Language
DML → Data Manipulation Language
DQL → Data Query Language

DATE __/__/____
PAGE _____

b) Constructing a database. - Process of storing data in some medium, controlled by DBMS (Hard disk)

c) Manipulating the database - It includes functions as:-

1. Updating the database : i.e inserting, deleting, updating the data from the DB.

2. Querying the DB to retreive some information from the DB.

3. Generating the reports.

d) Sharing the database - It includes access DB to multiple users as well as programs.

e) System Protection - protection from any kind of hardware & or software failure

f) Security Protection - protection from unauthorised access.

g) Backup and recovery

h) Transactions.

Database v/s File Processing System (DB v/s FPS)

Q what are the advantages of database system over file processing system? OR

Q what are the disadvantages of FPS over DB system?

GOOD WRITE

DATE: __/__/____
PAGE _____

1. Controlling the data redundancy
2. Restricting unauthorised access.
3. Providing multiple user Interface
4. Providing backup and recovery
5. Enforcing Integrity constraints
6. Represent complex relationships among data.
7. Providing persistent storage for program objects.
8. Permitting Inferencing and actions via some rules.

Q. How to control data redundancy in DBMS?
→ By using the concept of keys and normalization.

Data Integrity
- consistent data
- Accurate data

⇒ Constraints restrict the unwanted data to be entered.

Structured Query Language (SQL)

① Character datatype

a) ~~Syntax~~ → char(n) {Static character type}
where n is the total no. of bytes

b) varchar(n) or varchar2(n) {Dynamic character type}
variable sized character
Eg: Sname varchar(20);

GOOD WRITE

DATE: __/__/____
PAGE _____

② Numeric datatype

a) Integer type

int, short int, long int, Integer

b) Real no.

↳ float
↳ double

Eg: marks float;
↳ 85.000000

c) Formatted numeric data type

(i) decimal (i,j)

(ii) numeric (i,j)

where i = Precision that is total no. of digits in a number

j = scale
i.e decimal digits after decimal point

Eg: marks numeric (5,2);

000.00 ✓
99.00 ✓
100.00 ✓
99.99 ✓
9999.9 (Invalid)

DATE: __/__/____
PAGE _____

(3) Date and Time datatype:

→ used to store date in 'YYYY-MM-DD' format

Syntax

<I> date
<II> time → hh:mm:ss
<III> datetime → YYYY-MM-DD hh:mm:ss
<IV> timestamp
<V> smalltime

SQL Languages

1. DDL or Data definition language
   ↳ for defining the structure of the database / tables / views / indexes, etc.

   Commands
   - CREATE
   - ALTER
   - DROP

2. DML or Data manipulation language
   ↳ 3 kinds of operations can be applied to data in the DB:-
   - INSERT
   - DELETE
   - UPDATE

3. DQL or Data Query language
   ↳ Command
   - SELECT

4. DCL or Data control language
   ↳ 2 commands
   - GRANT
   - REVOKE

GOOD WRITE

Q Difference between DELETE and DROP commands

DROP	DELETE
1. It is a DDL command.	1. It is a DML command
2. It deletes the data as well as structure of the table.	2. It deletes only the data of the table.

SQL Operations

Arithmatic operators

+, -, *, /, %

→ It is applied to two operands.

Logical operators

AND, OR, NOT, BETWEEN, IN, EXISTS, IS NULL, ANY, LIKE

↳ pattern searching operator

Relational Operators

>, <, >=, <=, !=, =

→ They are used for comparisons.

GOOD WRITE

DATE: __/__/____
PAGE _____

DDL

I Create Command

→ Syntax for Creating a table :-

```
Create table tablename ( fieldname datatype [constraint],
                         fieldname datatype [constraint],
                         :
                         fieldname datatype [constraint] );
```

Structure of Student is as-

Student

Fieldname	Datatype	constraint(if any)
Rollno	int	Primary key
Name	varchar(20)	NOT NULL
Age	int	
Mobile	numeric(10)	
Address	varchar(20)	

```
Create table Student ( Rollno int primary key,
                       Name varchar(20) NOT NULL,
                       Age int,
                       Mobile numeric(10),
                       Address varchar(20) );
```

Dept

Deptno	Dname
1	VSIT
2	VSBS
3	DSB

GOOD WRITE

create table dept (deptno int primary key identity(1,1),
dname varchar(20));

auto updation
identity((1),(1)),
& starting value, increment

Identity constraint should be used at the time of defining the structure of the table

II ALTER Command

→ used to modify the structure of the table

① We can add more column(s) to the existing table.
② We can change the data types of the column / fieldname
③ We can change the size of the character data type.
④ We can add constraint to the field of the table.
⑤ We can modify constraint of the field of the table.

Syntax:

① Adding a new column to the existing tablename

ALTER TABLE tablename
ADD Columnname datatype [constraint];

Eg: ALTER TABLE Dept
ADD location varchar(20);

GOOD WRITE

DATE: __/__/____
PAGE _____

Dept

Deptno	int Primary Key
Dname	varchar(20)
Location	varchar(20)

→ Structure of the table Dept

② Modify the column's data type to the existing table_name

```
ALTER TABLE tablename
ALTER COLUMN column_name datatype;
```

Eg:
```
ALTER TABLE Dept
ALTER COLUMN dname char(10);
```

Structure →

Deptno	int Primary key
Dname	char(10)
Location	varchar(20)

③ Dropping / Deleting the existing column_name

```
ALTER TABLE tablename
DROP COLUMN column_name;
```

Eg:
```
ALTER TABLE Dept
DROP COLUMN location;
sp_help dept;
```

Structure →

Deptno	int Primary key
Dname	char(10)

GOOD WRITE

III. DROP Command.

→ used to delete the table permanently from the memory.
→ deletes all the records with the structure.

Syntax: DROP table table_name;

DML

I INSERT Command.

→ used to insert records in a table.

Syntax 1:
INSERT INTO table_name
VALUES (value1, value2, ..., value n);

where table_name is an existing table_name and value1, value2, ... value n are the values for the corresponding fieldname / columns.

Syntax 2:

INSERT INTO table_name (field1, field2, ..., field n)
VALUES (value1, value2, ..., value n);

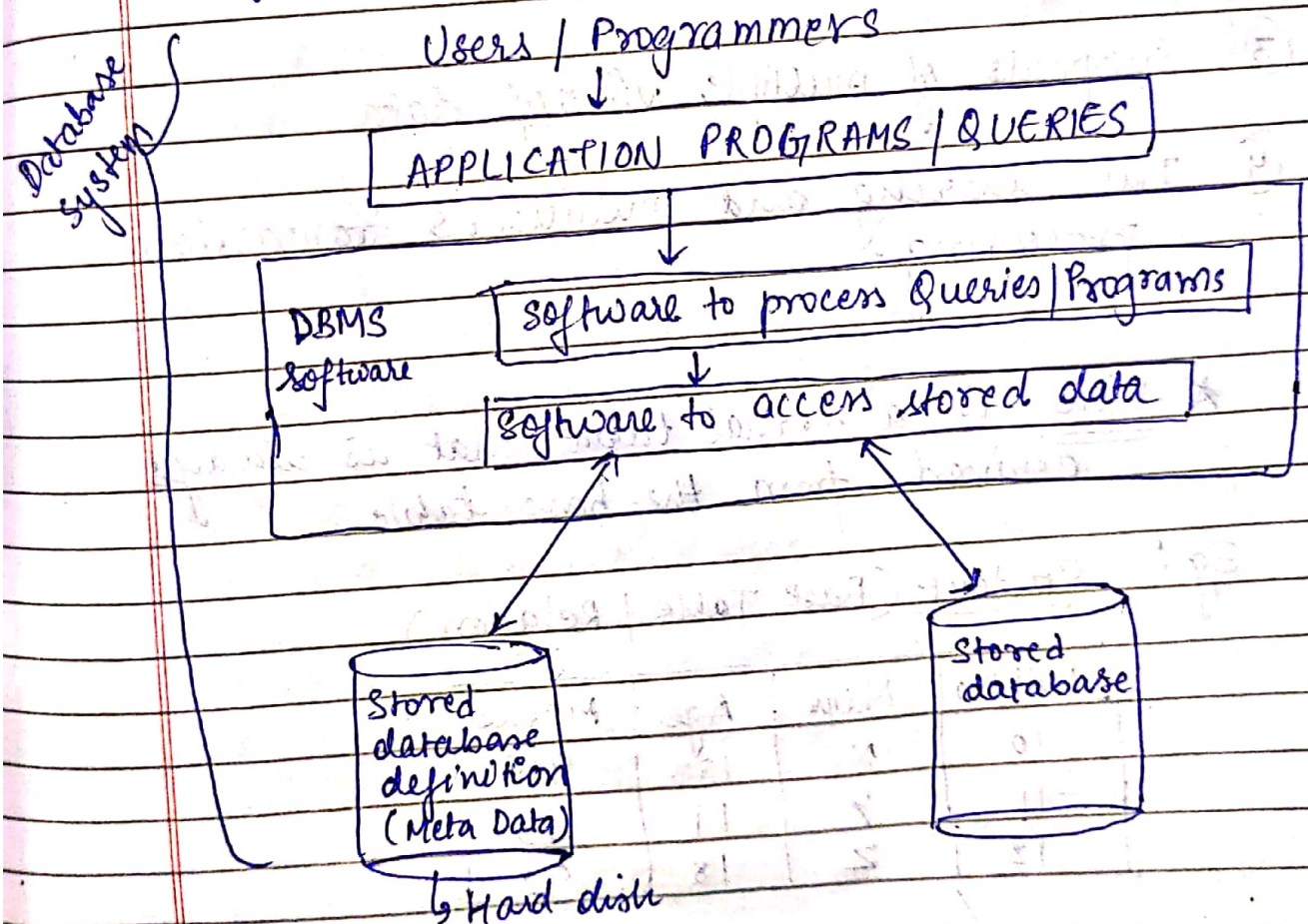
DATE: __/__/____
PAGE _____

[UNIT - I] Theory continue...

Database System:

A database ~~system~~ together with DBMS software is called Database system.

Fig: Database System Environment.



GOOD WRITE

DATE: __/__/____
PAGE _____

[V.V.I] **Characteristics of database system over file processing system**

→ 4 characteristics that distinguish the database approach over file processing system

① **Self describing nature of a database**
↳ changes made at one place is reflected to all the

② Insulation between Programs and data and data Independence.

③ Supports of multiple views of data

④ Data sharing and multiusers transaction processing.

★ View is a virtual table that is always derived from the base table.

Eg: Student (Base Table / Relation)

Roll No	Name	Age	Marks
10	X	18	90
11	Y	19	85
12	Z	18	92

View 1

Roll No	Name
10	X
11	Y
12	Z

View 2 (mark > 90)

Roll No	Name	Age	Marks
10	X	18	90
12	Z	18	92

GOOD WRITE

DATE: __/__/____
PAGE _____

* Transaction processing system is used to control concurrency.
Multiple users can access the database at a single time. They are known as concurrent users.

V.V.I. DBMS ARCHITECTURE / ANSI - SPARK 3-Tier Architecture

- The idea was first described by the ANSI / SPARK committee in 1970's.
- Also called ANSI - SPARK 3-Tier / 3-Scheme architecture.
- The purpose of DBMS architecture is to provide user interface with an abstract view of the data called data abstraction.

A DBMS can be organised into 3-seperate levels of abstraction -:

1. Internal level or physical level
2. Conceptual level or Logical level
3. External level or view level

ABSTRACTION LEVELS:

1. Internal or Physical Level
→ It is the lowest level of abstraction that describes internal scheme i.e. how the data is actually stored in the database.

GOOD WRITE

→ It uses a physical data model.
→ Concerned with physical storage, actual size (in bytes), access mechanism, sequential, random, index, etc.
→ Data Encryption and Decryption techniques.
→ DBA (Data Base Administrator) is required.

2. Conceptual or Logical Level
→ It describes the structure and constraints of the entire database for all users.
→ Describes what type of data is stored in the database.
→ Uses a conceptual model.
→ Normally, the users are application programmers and DBA.

3. External or View Level
→ It is the highest level of abstraction.
→ It describes the user's view.
→ Uses the same model as the conceptual level used.
→ End users uses this.

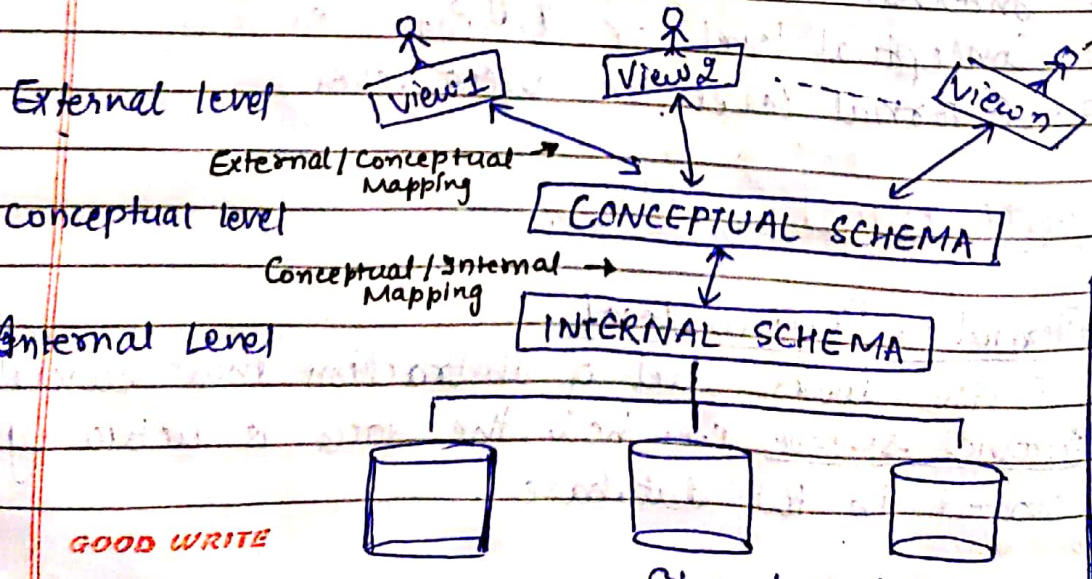
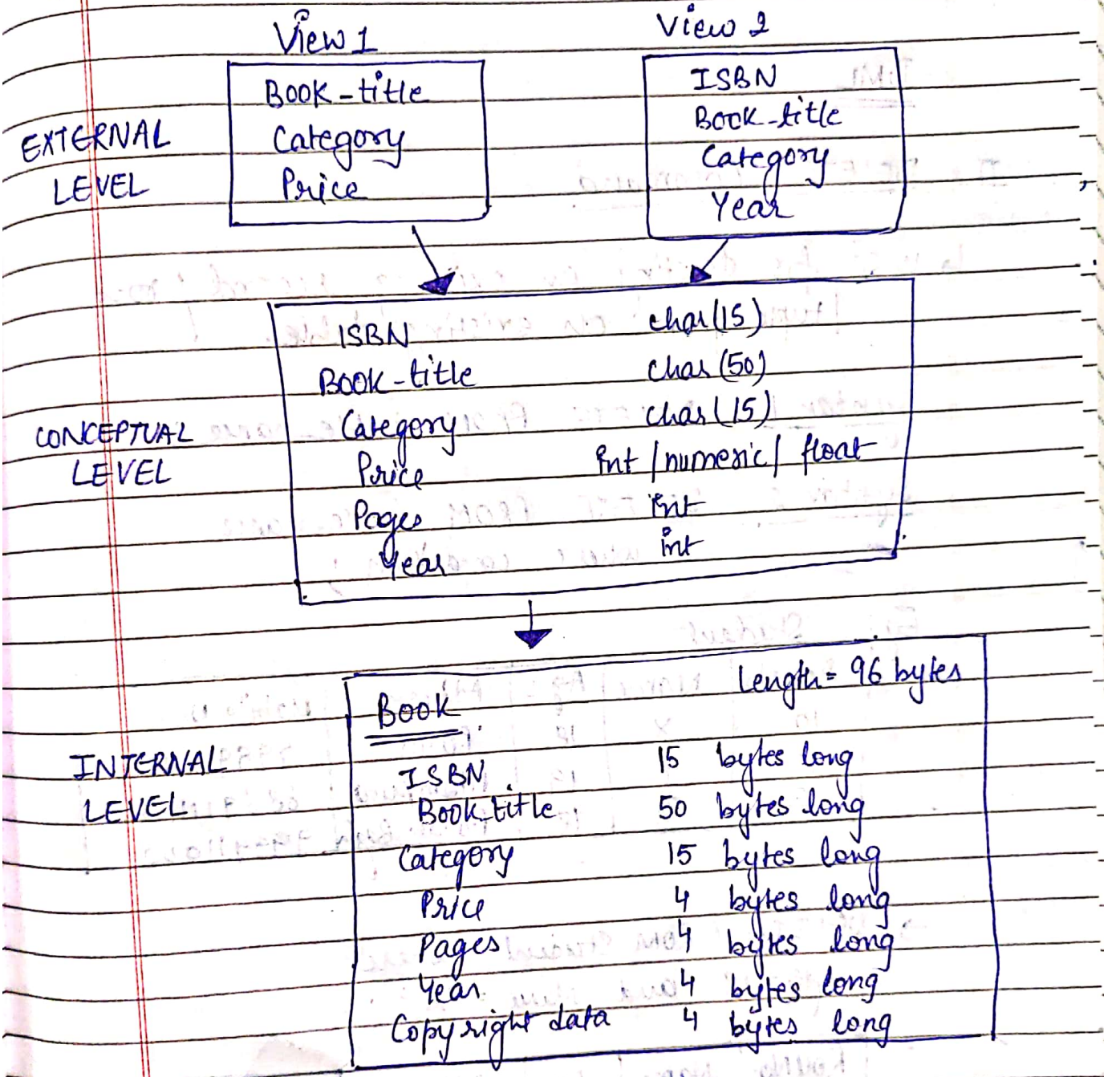


Figure:
ANSI / SPARK 3-Schema / Tier Architecture
Pg 36

GOOD WRITE

DATE: __/__/____
PAGE _____

Eg: Online Book Database (Application)



GOOD WRITE

DATE: __/__/____
PAGE _____

Practical (SQL)

DML

II DELETE Command

↳ used for deleting an existing record / row / tuple from an existing table.

Syntax 1 DELETE FROM table_name ;

Syntax 2 DELETE FROM table_name
where condition ;

Eg: Student

RollNo	Name	Age	Address	Mobile No.
10	X	18	Rohini	99993100
11	Y	19	Pitampura	88891111O
12	Z	18	Punjabi Bagh	999911000

→ DELETE FROM Student where age = 18 and Name = 'X' ;

RollNo.	Name	Age	Address	Mobile No.
11	Y	19	Pitampura	88891111O
12	Z	18	P. Bagh	999911000

GOOD WRITE

DATE: __/__/____
PAGE _____

III UPDATE Command.

↳ used to modify the record / row / tuple of a table / relation.

Syntax 1. UPDATE table-name SET column1 = value1, column2 = value 2, ... column n = value n;

Syntax 2
```
UPDATE table_name
SET column= value
where condition;
```

Q: Create a table named EMP with the following info

Field Name	Datatype	Constraints (if any)
Empno	int	Primary Key and auto increment
Name	varchar(20)	NOT NULL
Age	int	Between 18 and 65
Gender	char(1)	either 'M' or 'F'

Ans.
```
CREATE TABLE EMP ( Empno int primary key
identity (1,1), name varchar(20) NOT NULL,
age int CHECK ( age >=18 AND age <= 65),
gender char(1) CHECK ( gender ='M' OR gender='F')
);
```

GOOD WRITE

DATE: __/__/____
PAGE _____

III UPDATE Command

↳ used to modify the record / row / tuple of a table / relation.

Syntax 1: UPDATE table-name SET column1 = value1, column2 = value2, ... column n = value n;

Syntax 2:
```
UPDATE table_name
SET column = value
where condition;
```

Q: Create a table named EMP with the following info

Field Name	Datatype	Constraints (if any)
Empno	int	Primary key and auto increment
Name	varchar(20)	NOT NULL
Age	int	Between 18 and 65
Gender	char(1)	either 'M' or 'F'

Ans.
```
CREATE TABLE EMP ( Empno int primary key
identity (1,1), name varchar(20) NOT NULL,
age int CHECK ( age >= 18 AND age <= 65),
gender char(1) CHECK ( gender = 'M' OR gender = 'F')
);
```

DQL

→ used for querying database i.e. getting / retrieving / querying data from the database.

GOOD WRITE

Scanned by CamScanner

DATE: __/__/____
PAGE _____

I SELECT Command

Syntax 1 : SELECT * from table_name;

Syntax 2 : SELECT column 1, column 2, ... column n from table-name;

Syntax 3 : SELECT * from table-name WHERE condition;

Syntax 4 : SELECT column 1, column 2, ... column n from table_name where condition;

The main clauses used in select command are -:

1. SELECT
2. FROM
3. WHERE
4. ORDER BY (for sorting a particular field)
5. GROUP BY
6. HAVING

SQL LIKE Operator [very very Imp]

↳ used for pattern searching
↳ uses two wildcard characters
1. % Symbol (percentage) → 0 or more characters are replaced
2. _ Symbol (underscore) → only single character can be replaced.

GOOD WRITE

Scanned by CamScanner

DATE: __/__/____
PAGE _____

Student

Roll No	Name	Age	Address	Marks
10	Sumit	18	Rohini	90
11	Sanvi	19	Pitampura	85
12	Jatin	18	Janakpuri	60

domain

Q. Find out the details of each student -
a) where name starts with S.
b) where marks ends with 0.

Ans a) SELECT * from student where Name LIKE 'S%';
b) SELECT * from Student where Marks LIKE '%0';

SQL ORDER BY Clause

↳ used for arranging / sorting a particular field.

Syntax : SELECT * from table_name [where condition]
ORDER BY colum_name [ASC / DESC];

Eg: SELECT * from Student ORDER BY NAME;

Student

Roll No	Name	Age	Address	Marks
12	Jatin	18	Janakpuri	60
11	Sanvi	19	Pitampura	85
10	Sumit	18	Rohini	90

Scanned by CamScanner

SQL Integrity Constraints

→ applied to a column of a table
→ Constraints are the rules or the conditions applied to a column for data accuracy, integrity and consistency.

These are 6 types of integrity constraints :-

1. PRIMARY KEY → defines uniqueness of a row
2. NOT NULL
3. UNIQUE
4. CHECK
5. DEFAULT
6. REFERENTIAL INTEGRITY / FOREIGN KEY

Q. Difference b/w UNIQUE and PRIMARY KEY constraints.

Ans. → UNIQUE Constaint column can have NULL value
→ PRIMARY KEY constraint column cannot have NULL value. It should contain some value.

DATA MODELS

Q. What is a data model ?

→ is a collection of concepts / conceptual tool used to design the structure of the Database.
→ Structure means data types, relationships and Constraints applied to the fieldname for consistency and accuracy.
→ Also used for basic operations such as data retrievals and updation.

GOOD WRITE

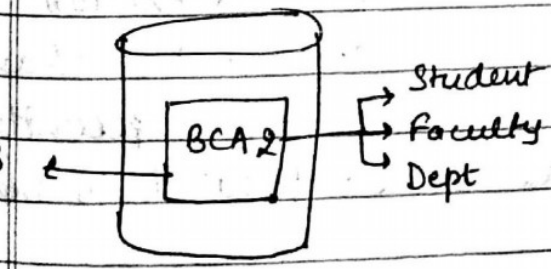
Scanned by CamScanner

Categories of Data Model

1. Conceptual or High Level DM
   - → includes all major concepts such as entity (object), attribute and relationship
   - → Eg: ER-Model, Object Oriented Model
2. Physical or Low-level DM
   - → includes operations i.e. how data are stored in the database.
3. Implementation or Representation DM
   - → used to implement the ~~Constraint~~ conceptual design to any one of the following implementation model.
     - Relational Data Model
     - Network Data Model
     - Heirarchical DM

Database Schema

↳ overall structure of the database



Database Schema for BCA2

Student

RollNo	SName	Age	City	MobileNo

Dept

Deptno	Dname	Location

Faculty

Id	Name	Date_of_birth	Address	Gender

Overall DB Schema
OR
Intension

GOOD WRITE

Scanned by CamScanner

PAGE

Database State (Extension)

↳ No. of records in a database table.

	No. of records
Student	3
Faculty	3
Dept	5

Extension

Instances

↳ Single row or a tuple in a table is known as an instance. whose data is related to each other.

Q. Difference b/w Database Scheme and database state.

Ans.

Database schema	Database State
1. Intension	1. Extension
2. less frequently altered by the user	2. Can be changed frequently
3. Related to the structure	3. Related to the records.

DATABASE USERS

↳ self study

Scanned by CamScanner

Data Modeling using the ER-Model.

(Entity Relationship)

CONCEPTUAL DATABASE DESIGN

What are the phases of database design?

1. Requirement Analysis / Information Gathering
2. Conceptual Design → implemented by ER-Model using pen and paper.
3. Implementation Design
4. Physical

The diagram used to represent ER-Model is called ER-Diagram.

(1) E = Entity → (2) Attribute

(3) R = Relationship

ER-Model

- The ER-Model is used at the conceptual design phase of the database design.
- It was implemented by Peter Chan in 1976.
- ER-diagram is used to implement ER-Model
- Based on 3-concepts :- Entity, Attribute and Relationship

Entity :- An entity is an object or thing that exists in the real world (tangible or non tangible)

Eg: student, person, company, college, etc.

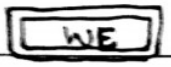
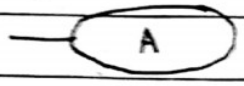
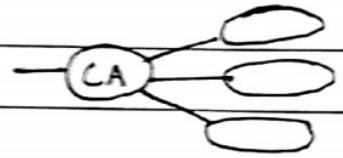
Attributes :- It identifies the characteristics / features / properties of an entity.

Eg: RollNo, name are the attributes for entity STUDENT

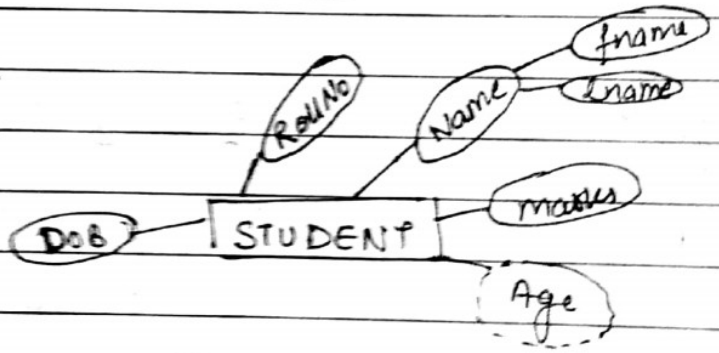
GOOD WRITE

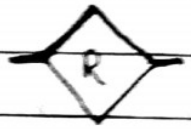
Scanned by CamScanner

Notations used in ER-Diagram

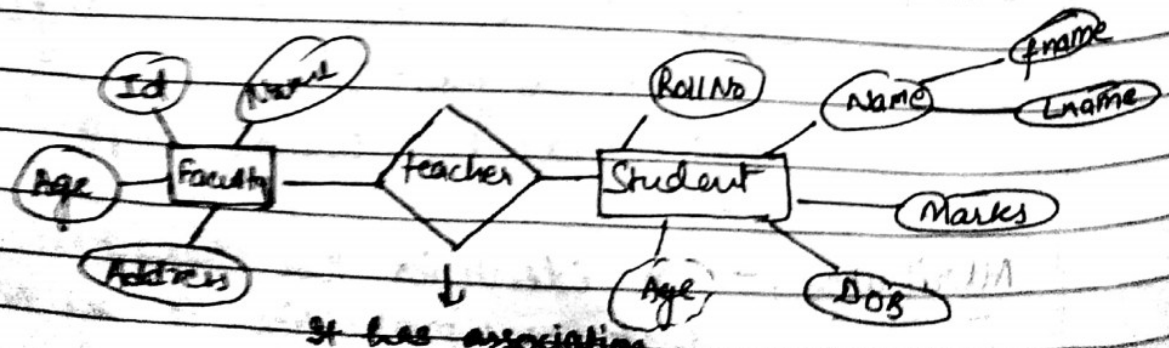
	Symbol	Meaning
1.		Entity / Strong Entity
2.		Weak Entity
3.		Attributes (simple) → terminal attribute Eg: RollNo
4.		Composite Attributes → an attribute that can further be divided. Eg: Name

Eg:



5.		Derived attribute
6.		Relationship Eg: teacher
7.		Identifying ~~Attribute~~ Relationship

Eg:



↓
It has association with two entities.

Scanned by CamScanner

DATE: __/__/____
PAGE _____

DBMS PRACTICAL

SQL Aggregate Functions

→ used for getting consolidated information from a group of tuples / rows / records.

→ 5 types of aggregate functions are provided by SQL.

1. Sum( ) → returns the total values from the corresponding field.

Eg: Select Sum (marks) as totalmarks from student;

2. Max( ) → returns max value of a field
3. Min( ) → returns min value of a field
4. Avg( ) → returns average of all values of field.
5. Count( ) → fieldname or *

⇒ Select count(*) from emp; → display total no. of records.

⇒ Select count (comm) from emp;
↳ It will not count NULL values.

⇒ Not equal to ⇒ '!=' or '<>'

Scanned by CamScanner

DATE: __/__/____
PAGE _____

Theory continue ...

8.  Multi valued attribute

+ Stored and Derived Attributes

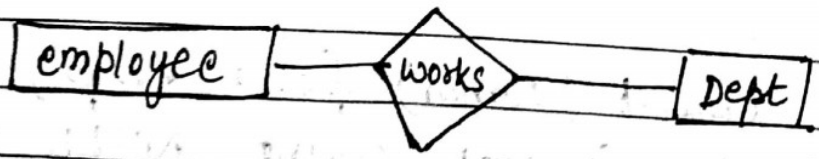
Stored Attribute : They are already stored in the database and from them the value of another attributes can be derived.
Eg : DOB

Derived Attribute : whose value is derived from Stored attributes.

→ They are created usually by a formula applied on stored attributes.
Eg : Age.

→ Key Attribute (Primary Key)
→ NULL Attribute

Q. Identify the entities and relationship
① An employee works in a dept.



Scanned by CamScanner

DATE: __/__/____
PAGE _____

Types of Relationship

Degree

1 → Unary relationship (1 entity involved)
2 → Binary relationship (2 entities involved)
3 → Ternar relationship 3 entities involved)
4 → N-ary relationship (N>3 entities involved)

Relationship Constraint

① Mapping Cardinality

1:1	1:M	M:1	M:M
(one to one)	(one to many)	(many to one)	(many to many)

Scanned by CamScanner

DATE: __/__/____
PAGE _____

Types of Relationship

Degree
1 → Unary relationship → (Recursive Relationship) (1 entity involved)
2 → Binary relationship (2 entities involved)
3 → Ternary relationship 3 entities involved)
4 → N-ary relationship (N>3 entities involved)

Relationship Constraint

① Mapping Cardinality | Cardinality Ratio

- 1:1 (one to one)
- 1:M (one to many)
- M:1 (many to one)
- M:M (many to many)

② Participation

PRACTICAL

SQL GROUP BY AND HAVING CLAUSE

GROUP BY CLAUSE

It is used with aggregate functions to group the records / rows / tuples based on one or more column's value.

HAVING CLAUSE

It is used with GROUP BY clause which enables us to specify conditions that filters which group results appear in the results.

GOOD WRITE

SELECT deptno, sum(Salary) as total_sal from EMP group by deptno having sum(Salary) > 4000;

↳ it will display total salary greater than 4000.

Q1. Find out the total salary dept wise of each dept.

Q2 Find out the maximum salary from EMP grp by deptno.

Ans 2. SELECT deptno, max(salary) as max_sal from EMP group by deptno;

Ans 1. SELECT deptno, sum(salary) as total_salary from EMP group by deptno;

Using all aggregate functions:

```
SELECT deptno, sum(salary) as total sum,
               min(salary) as minimum,
               max(salary) as maximum,
               avg(salary) as total counts;
from EMP GROUP BY deptno;
```

Syntax of GROUP BY and HAVING clause

```
SELECT [column_name], aggregate function(column)
FROM table_name where condition
GROUP BY column_name, HAVING condition;
```

↓ column_name should be same.

FOREIGN KEY

→ used for joining two or more tables / relations
→ A foreign key is a key in an attribute in a table that refers to some another table where it is a primary key.

GOOD WRITE

DATE: __/__/____
PAGE _____

SQL JOIN OPERATION.

Primary Key.

Empno	Ename	Age	Deptno
110	X	22	10
111	Y	18	20
112	Z	30	10
113	E	32	30

EMP

Primary Key.

Deptno	Dname	Location
10	VSIT	Block A
20	VJMC	Block B
30	BSB	Block C

DEPT

```
CREATE TABLE DEPT
(Deptno int Primary key, Dname varchar(10),
 Location varchar(20));

CREATE TABLE EMP
(Empno int Primary key, Ename varchar(20), Age int,
 Deptno int foreign key Reference DEPT(Deptno));
```

Conditions to be checked :

1. The data type of foreign key and primary key must be same, but fieldname can be different.
2. A foreign key can have NULL value but a primary key can never have a NULL value.

GOOD WRITE

DATE: __/__/____
PAGE _____

SUBQUERY

(Nested Query / Inner Query)

→ A subquery is a query within another SQL query and embedded within the WHERE clause.

→ A subquery is used to return the data that will be used in the main query.

→ Subqueries are most frequently used with the SELECT statement.
i.e a subquery is a Select statement that is nested within another Select statement and returns immediate results.

Subquery ⇒ (Select statement .... (Select statement));

↓ Outermost Query will be executed last

↓ Innermost subquery that will be executed first.

Syntax

SELECT column [, columns] from table_name
WHERE column OPERATORS (SELECT column [, columns]
from table_name
[where condition]);

→ Relational
→ Logical
→ Between / And
→ IN operator
→ NULL

GOOD WRITE

DATE: __/__/____
PAGE _____

Eg: Q. Find ename of employee who was hired first.

Ans Select ename from EMP where Hiredate IN → OR =
(Select min(Hiredate) from EMP);

Q. Find out the name of employee having 2nd minimum salary.

Ans Select ename, min(Sal) as Second-minimum from EMP where Sal > (Select min(Sal) from EMP);

Q: Find second maximum salary.

Ans Select max(Sal) as second-max from EMP where Sal < (Select max(Sal) from EMP);

Q: List the name and salary of each employee who have above the average of salary.

Ans Select ename, Sal from EMP where Sal >
(Select avg(Sal) from EMP);

GOOD WRITE

DATE: __/__/____
PAGE _____

Case Study Based on ER-Model of COMPANY DATABASE.

Write down the following descriptions :-

① The company is organised into departments, each department has a unique name, a unique number and a particular employee who manages the department. We keep track of the start date i.e. when that employee started managing the department. A department may have several locations.

② The department controls a no. of projects each of which has a unique name, unique number and a single location.

③ We store each employees name, social security number (SSN), address, salary, gender and birth date. An employee is assigned to one department but may work on several projects. We keep track of the no. of hours per week that an employee works on each project.

④ We want to keep the track of the dependence of each employee for insurance purpose. For that we keep each dependence. first name, gender, birth date and relationship to the employee.

GOOD WRITE

DATE: __/__/____
PAGE _____

Q1. Draw the ER-diagram of the above data.
Q2. List out all the strong entities.
Is there any weak entity? If so, find out the name of the weak entity.

Q3 List all the key attributes, simple attributes, composite, stored, derived, single valued, multivalued attributes.

Q4 Find out the various ~~relash~~ relationship constraints

Steps to make ER-diagram

1. Identify the strong entities
2. Identify the weak entities
3. Find out the types of attributes with each entity.
4. Find out the relationship and their constraints.

* strong entity with weak entity, relationship is identifying relationship and is denoted by double diamond.

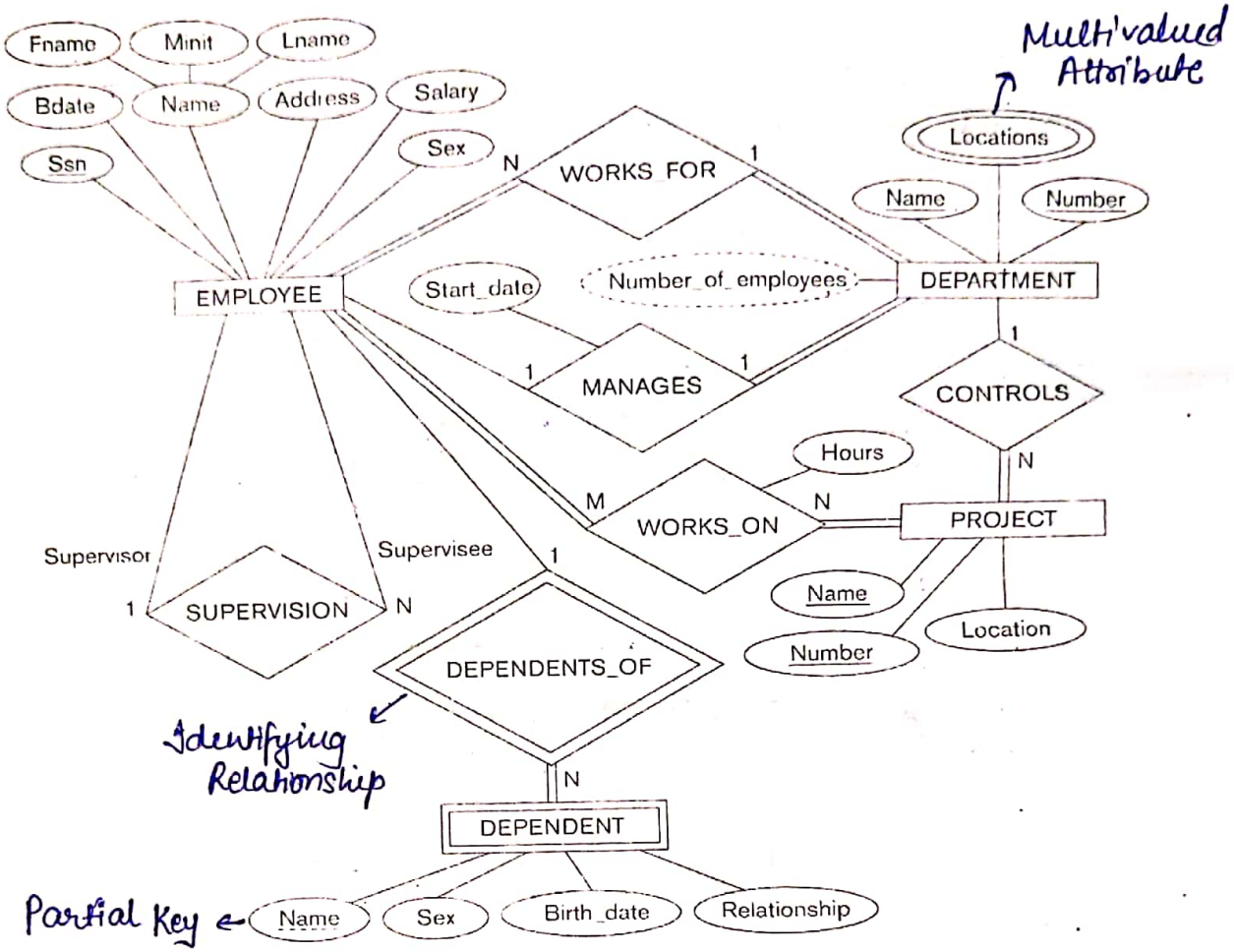
* Partial key. (oval)
↳ It is always associated with weak entity which can be associated with Primary key of strong entity to form a primary key in weak entity.

GOOD WRITE

DATE: ___/___/____

07017702018.

COMPANY ER-Diagram



Find out the following from the above COMPANY ER-diagram:

1. Strong Entity Types EMPLOYEE , DEPARTMENT , PROJECT
2. Weak entity types DEPENDENT
3. Simple attributes
4. Composite attributes Name
5. Stored attributes
6. Derived attributes Number-of-employees
7. Multivalued attributes Locations
8. Key attributes
9. Relationships SUPERVISION , WORKS-FOR , MANAGES , WORKS-ON , CONTROLS
10. Identifying Relationships DEPENDENTS-OF
11. Recursive relationships SUPERVISION

↳ Unary Relationship

DATE: __/__/____
PAGE _____

SQL JOIN

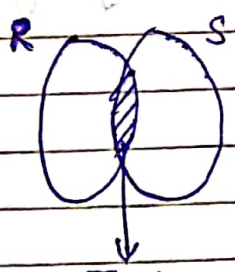
→ Joining is a process of combining two or more tables into a single table.

→ The SQL JOIN clause is used to combine records from two or more tables in a database, based on the common field of the above two tables.

→ The JOIN operation in SQL will be implemented by the SELECT clause

→ Tables are joined on the columns that have the same data types and data width in the table

Suppose R and S are two tables, then the JOIN operation can be shown as :-



JOIN
i.e common records from both the tables

(Simple / EQUI)

Syntax for SQL JOIN :-

```
Select Column_name list from table1, table2 where join-condition;
```

where, column name_list is the list of columns / fields from both the table 1 and table 2, seperated by comma

&

Join condition is -:

~~table1.columnfield = table2.columnfield;~~

table1.commonfield = table2.commonfield

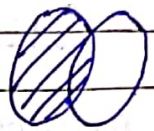
GOOD WRITE

DATE: __/__/____
PAGE _____

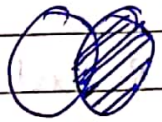
Types of SQL JOIN

1. Simple or EQUI JOIN
2. INNER JOIN
   ↳ same as ① but the syntax is different

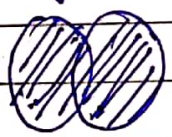
3. OUTER JOIN
   ↳ a) LEFT OUTER JOIN →



   b) RIGHT OUTER JOIN
   ↳



   c) FULL OUTER JOIN
   ↳ Output will be left followed by right
   → Duplicate records will be eliminated.



Syntax for INNER JOIN

Select Columnname-list from table1 INNER JOIN table2 ON table1.commonfield = table2.commonfield

GOOD WRITE

DATE: __/__/____
PAGE _____

Q. Find out the names of each employee, dept name and location.

↓ (SIMPLE JOIN)

Ans. Select ename, dname, loc from emp, dept
where emp.deptno = dept.deptno;

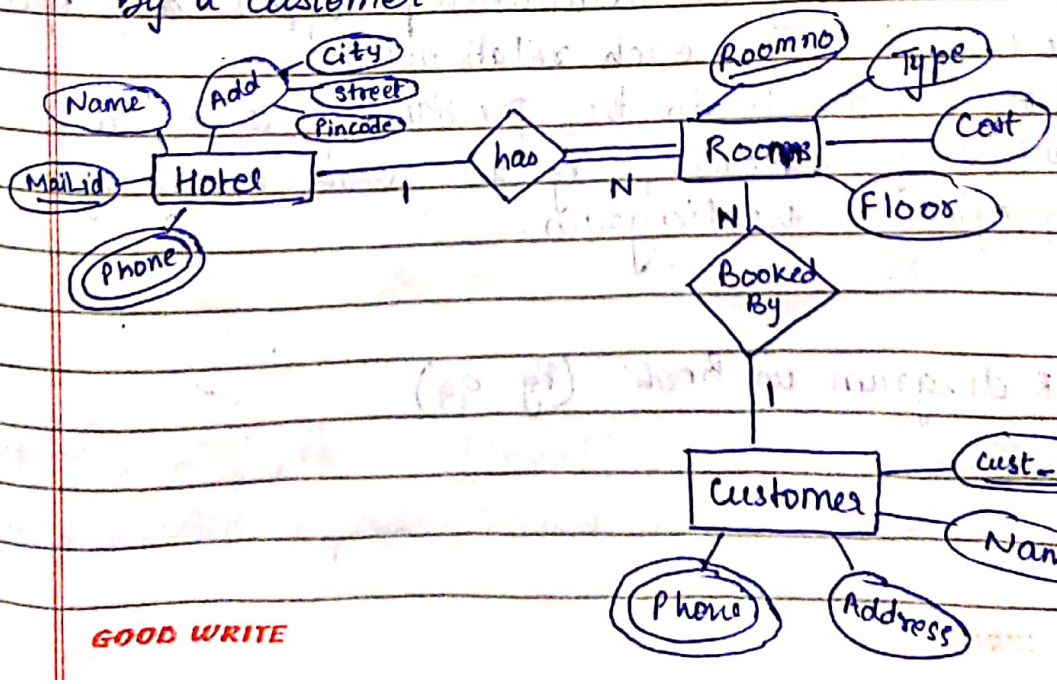
(INNER JOIN)

↳ Select ename, dname, loc from emp INNER JOIN dept ON emp.deptno = dept.deptno;

ER - DIAGRAM

- Car Insurance Company
- Hotel Management System
- University System (Pg 94)
- College System
- Airline booking (Pg 98)

Ques Draw an ER diagram for the following:
A Hotel has a no. of rooms and can be booked by a customer



GOOD WRITE

DATE: __/__/____
PAGE _____

BANK DATABASE (Case Study)

In the bank database, each bank branch can have multiple accounts and loans.
Bank has a code (primary key), name and address as its attributes.
Each branch of the bank has a branch no. and address.
Each account has an account no. (Primary key), balance and type.
Similarly, loan is associated with loan no. (Primary Key), amount and type.
Each customer can have multiple accounts and loans in the bank. Customer has a Cno (Primary key), name, address, phone no. and date-of-birth. Age of the customer can be derived from date-of-birth

Q1. Draw the ER-diagram of above database.
Q2. List all the strong entities and weak entity if any. If there is any weak entity, find out the partial key.
Q3. List the names of all relationship types and specify the constraint on each relationship
Q4 What constraints do the partial key and the identifying relationship of the weak entity type specified in the diagram.

→ ER diagram in Book (Pg 99)

GOOD WRITE

DATE: __/__/____
PAGE _____

EER (Extended E-R Model) / Object Modelling

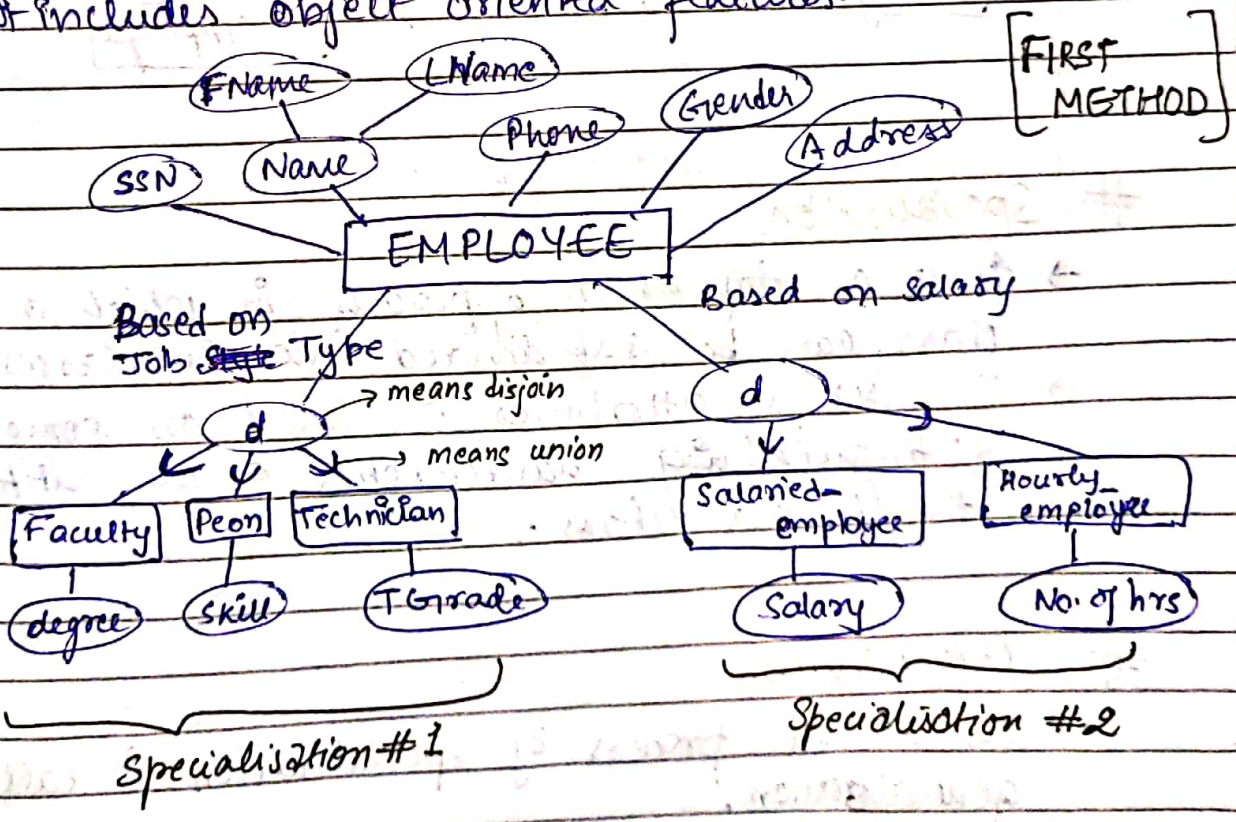
→ It is a higher level conceptual model used for developing complex database applications such as CAD / CAM (Computer Aided Design / Manufacturing).

→ It includes all the basic features of ER-Model i.e Entity, Attribute and Relationship

→ Its additional features are -
a) Superclass, subclass and Inheritance
b) Specialisation, Generalisation and Aggregation

→ It includes object oriented features.

Eg:



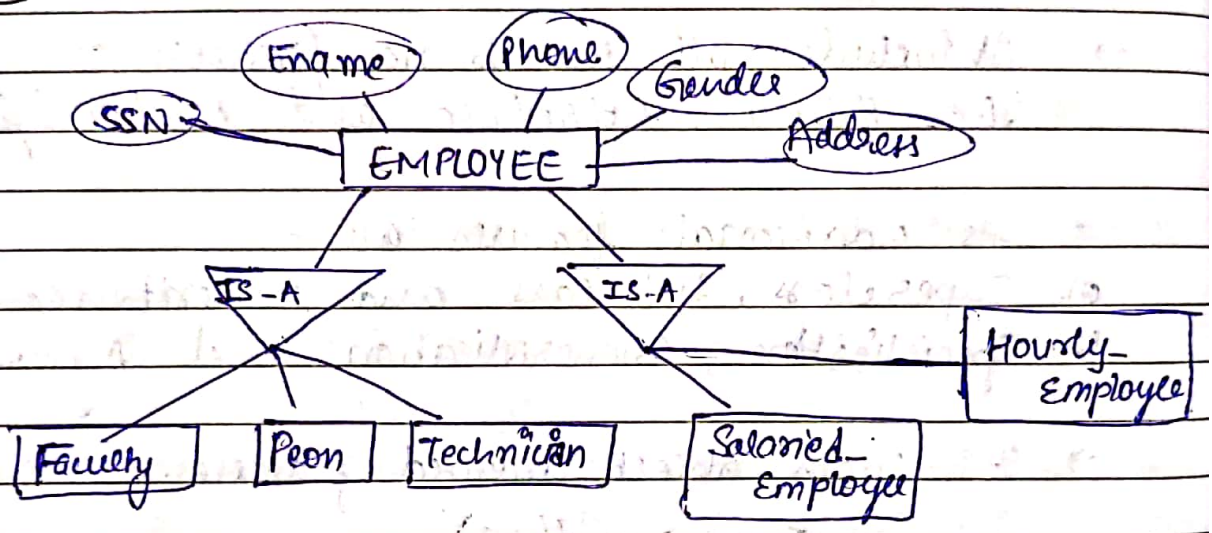
* {Faculty, Peon, Technician} is a specialisation of employee.
* {Salaried Employee, Hourly-Employee} is a specialisation of employee.

GOOD WRITE

Specialisation is a process in which we can categorize super class into different subclasses.

~~(Syntax) Eg.~~

[SECOND METHOD]



Specialisation.

→ It is a top-down approach in which a super-class can be sub divided into subclasses.

→ The set of attributes is based on some distinguishing characteristics of the entities in the super class.

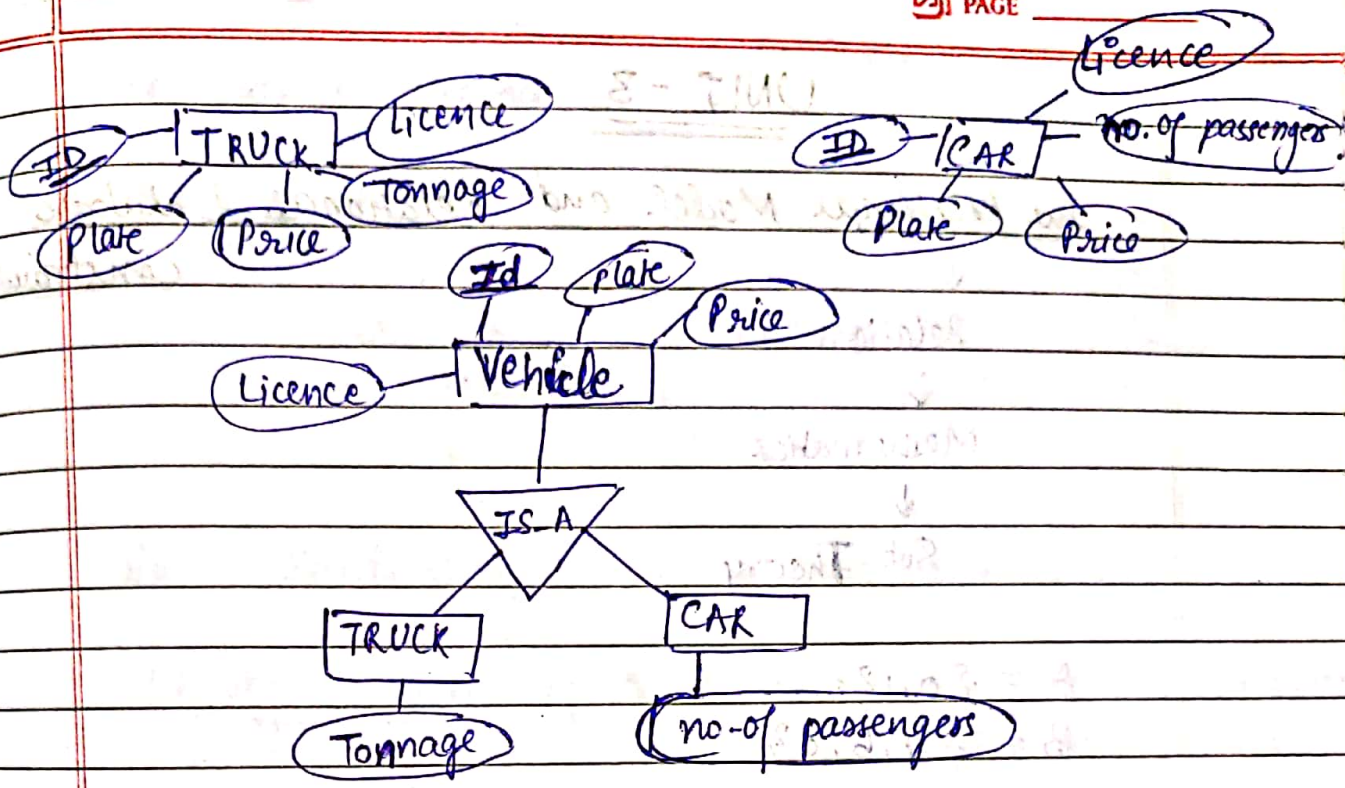
Generalisation.

→ The reverse process of specialisation is called generalisation.

→ It is bottom-up approach.

→ It takes common features and subclasses and then create a superclass.

GOOD WRITE



Superclass and Subclasses