Introduction to Database (CSE-315)

(ID)

Dt-21.04.22

① Silbertz Korth & Sudharsan 6th edition Database System Concept

② Allen Beaulieve, 2nd edition.

---

① Basic Introduction

② ER-Model

③ Basic keys

④ Normalization

⑤ Introduction control & concurrency

⑥ SQL & Relational Algebra.

⑦ Indexing

SQL :-

* collection of information called data.
  or raw facts

Data (Any facts which is can be recorded,
e.g. text, image, video, audio, number)

↓

Database (collection of related data)

↓

(collection of above data)

DBMS

DBS

RDMS

↓

DBMS

↓

IDB

DB + DBMS = DBS

Text + number = Traditional Database

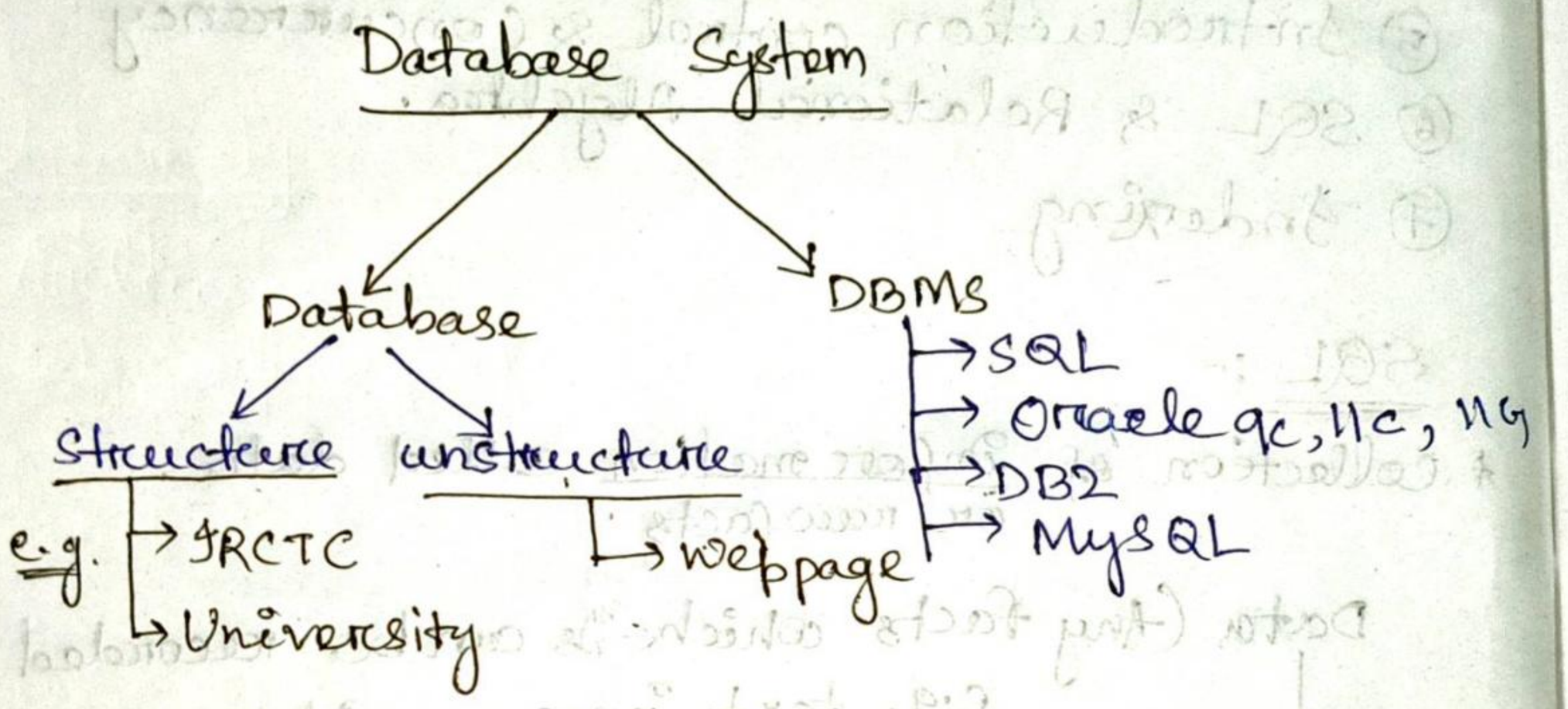
Video + Audio = Meltimedia Database

image (Satelite / Airline information). = Geogireaphic Information System (GIS) Database

Real life Database.

Date - 26.04.22

* Collection of relational data called Database.



→ A database system consists of a collection of interrelated data and a set of programs to access that data.

→ It is a software that assists in maintaining and utilizing a database.

→ A database management System Consists of;

(i) A collection of interc-related data & Persistant data. If some process enters to the data into the database then it can be subsequently removed from the database only by some explicit request

to remove then that data is said to be presistent.

→ This part of DBMS is usually refferred to as the database.

→ A set of applicationprogram used to access, update & manage that data. This is known as Database management System.

Database System Application :-

Enterprise Information

- Sale → for customer, for sale, for purchase, for product - information.
- Accounting → for payment, receipt, account balance, assets and other accounting information.
- Human Resources :→ for information about employees, salary, benefits, and for generation of pay checks.
- Manufacturing :- for management of the supply chain. and for tracking production of item in factories, inventories of items in. Warehouse and stored and order for items.

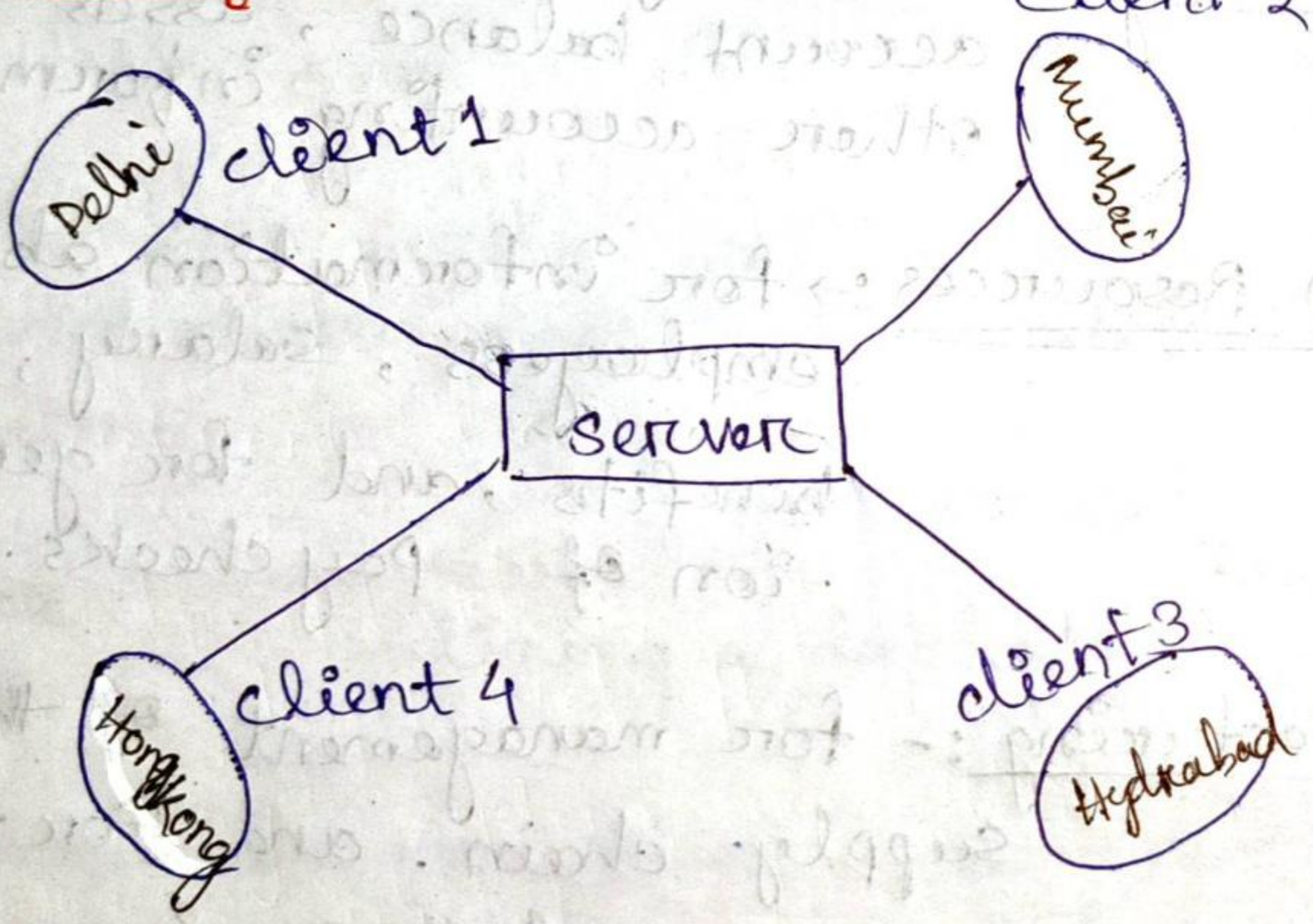
- Online Retailers :- for sales data + online order tracking, generation of recommendation list, maintainance of online product.

- Banking and Finance :-

① Banking.
② credit card transaction
③ finance

- University :-
- Airlines
- Telecommunication

File System Vs DBMS :-



Advantages :-

→ In file system user wants to search data of 1kb, they need to search from the

large amount of data. So thats why we have to stored data in secondary memory such as a Harddisk or a tape and as and when requeired we have to bring a portion of data to the main memory.

↳ The programmer carries out this part explicitly.

① Concurrency :-

② Redeendancy & Data consistency.

↓

(disadv. for file system)

File System vs DBMS

① Difficulty in accessing data.

→ Suppose that an university clerck needs to find out the name of all the student to live in a particular area.

→ Therefore conventional file system do not allow needed data to be retrieed in convinient and efficient manner.

② Data redeendancy and consistency.

since different programmer create the files and application programs over a long period, the various files are likely to have different streecture & the

Program may be written in several Programming language.

→ A redundancy leads to higher storage and access cost. In addition it may lead to data consistency.

③ Data isolation:

→ The data are scattered in various files and files may be in different formats. writting any new application program to retrieved the data is difficult incase of file system.

④ Integrity Problem:

→ The data value stored in the Database must satisfied certain types of consistency constraints.

i.e. In file system we don't apply any constraints whereas in DBMS we apply.

⑤ Concurrent access anamalies:

→ for the shake of overall performance of the system and the faster response of many system are allows multiple uses to update the data simultaneously.

⑥ Atomicity problem:

→ A computer system, like any other device

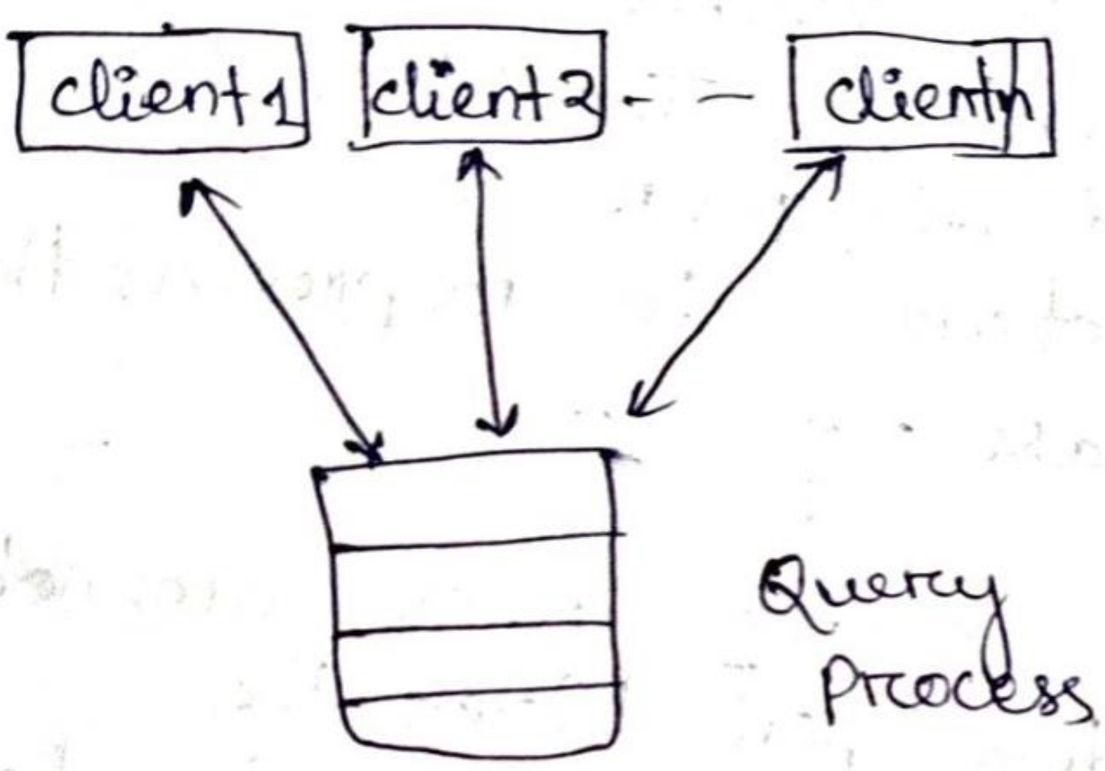
is subject to failure in many application it is crucial that if a failure occurs the data be restored to the consistent state that existed prior to the failure.

⑦ Security Problem :

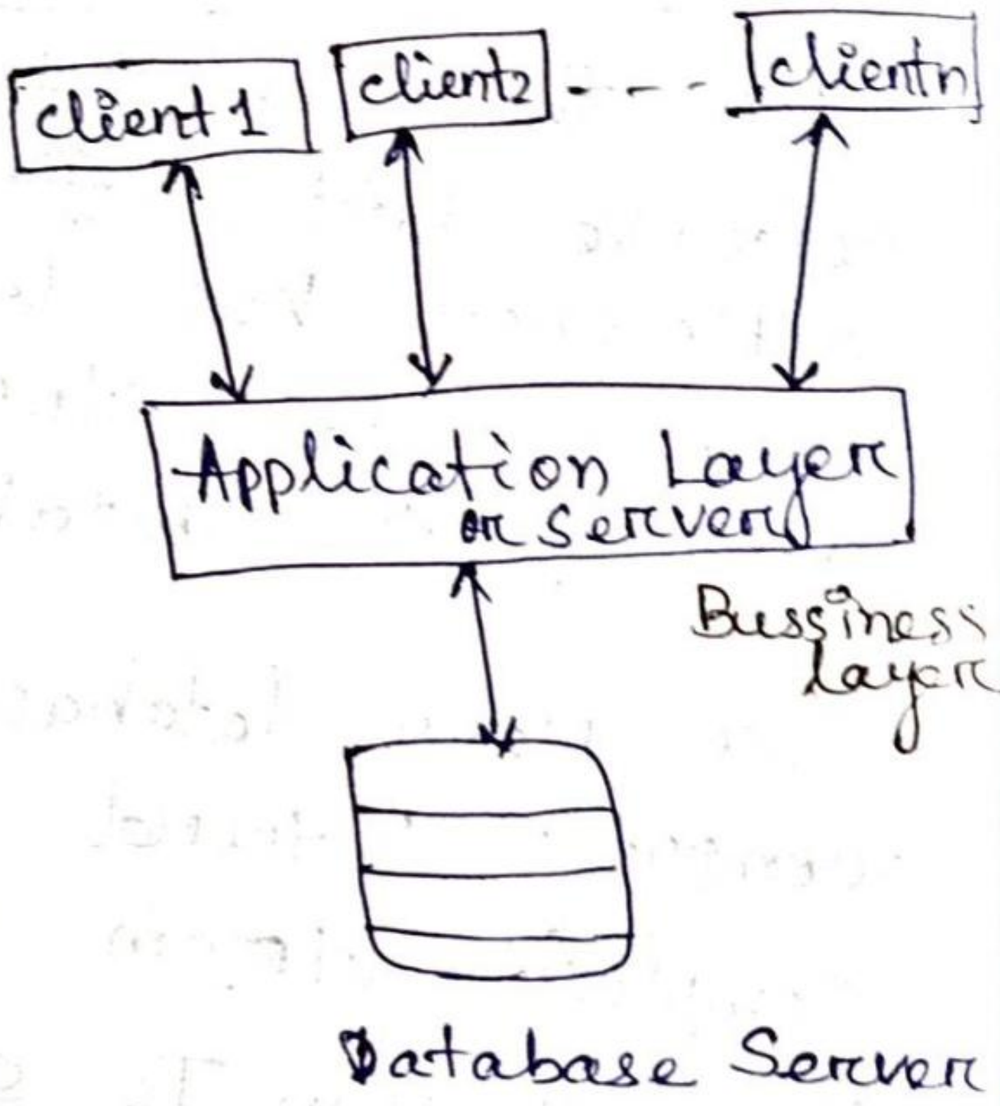
→ Not every user of the database system should be able to access all the data.

DBMS Architecture

2-tier Architecture



3-tier Architecture



Disadvantages :-

→ Security
→ Scalability

Dt- 28.04.22

View of Data

→ A database system is a collection of interrelated data and a set of program that allow users to access a modifier these data.

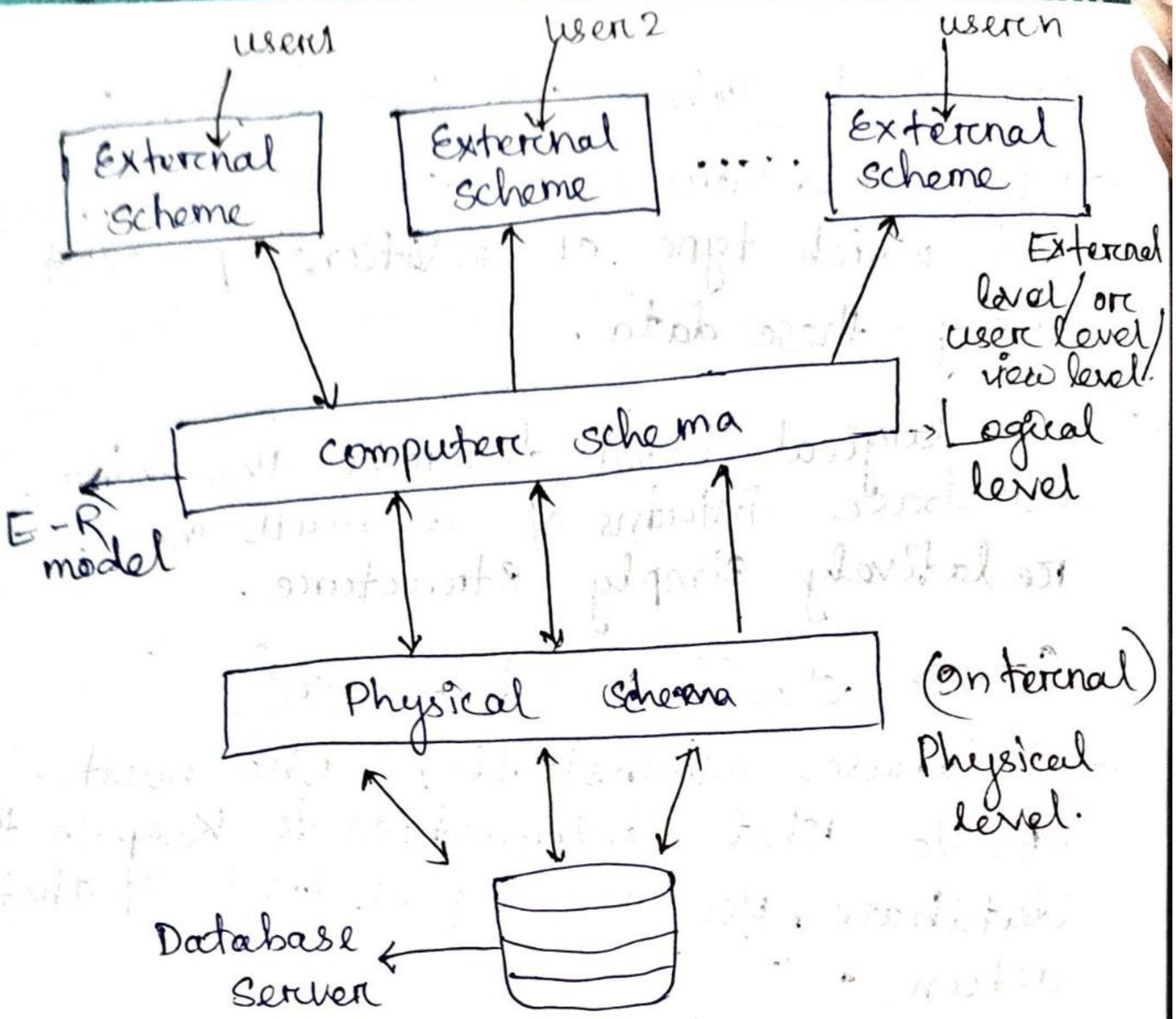
→ A measure objective of a databa system is to provide user with an abstract view of the data. i.e. the system hides certain details of how the data are stored and maintained.

Data Abstraction :-

→ for the system to be usable it must retrieve data efficiently, the need for efficiency has led designer to use complex data structure to represent the data in the database.

→ Since many database system user are not computer trend developer hides the complexity from user ~~for~~ several levels of abstraction. To simplify users interaction with the system.

→ Three level of Data Abstraction



Def^n of Schema :-

→ Database change overtime as information is inserted or deleted.

→ The collection of information is stored in the database at perticular moment is called instance of Database.

→ The overall design of the database is called database Schema.

→ Schema are changed in frequently at all.

External Schema in View Level -

→ the highes level of abstraction describes only part of the entire database.

Conceptual schema or Logical Level -

→ What data are stored in the database and which type of relationship exist among those data.

→ The logical level describes the entire database interms of a small no. of relatively simply structure.

(Table should be interrelated)

→ Database administration who must decide what information to keep in the database, use the logical level of abstraction.

→ Although implementation of the simple structure at the logical level may involve complex physical level structure.

→ the users of the logical level doesn't need to be aware of this complexity. This is refer to as physical data independence.

Physical schema :-

→ The lowest level of abstraction describes how, where the data are actually stored.

Database Language :-

① DDL :- This is used to specifying db schema. CREATE, ALTER, DROP,

, RENAME (conceptual schema)

② DML :- This is used for accessing and manipulating data in the database

SELECT, INSERT, UPDATE, DEL

(Physical schema)

*Structure of a table is called schema.

Dt - 04.05.22

DDL

What is Database language?

→ We specify a database schema by a set of def^n expressed by a special language called DDL.

→ It is used to specify additional properties of data. We specify the storage structures and accessed method use by the database system by a set of statements

in a special type of DDL called a data storage and defination language.

→ These statements defines the implementation details of the database schema which are usually hidden from the user.

→ the data values stored in the database most satisfied certain consistancy constraints.

Types of constraints :-

① Domain Constraints -

- A domain of possible values mostly associated with every attribute.
- Declaring an attribute to be a perticular domain acts as a constraints on the values that it can take.
- Domain constraints are the most elementary form of integrity constraints. There tested ~~by a~~ easily by the system whenever a new data item is entered into the database.

② Referential Integrity constraints -

Student

Name	roll	Course

Teacher

Name	roll	Course

→ There are some cases that we wish to ensure that a value that appears in one relation for a given set of attributes also appear in a certain set of attributes in ~~a~~another relation.

③ Assertion :-

→ An assertion is any condition that the database must always satisfy.

→ A Domain and Referential integrity constraints are special forms of assertion.

④ Authorization :-

→ User may want to differentiate according to the type of access to the database in terms of authorization. like; read, insert, update, delete.

* The DDL is just like any other Programming language gets input as some instruction & generates the of output.

* The output of the DDL is placed in Data dictionary, which contains meta data.

Data about product

* The Data dictionary considered to be a special type of table. That can only be access and updated by the database system itself.

DML :-

→ Data Manipulation language is a language that enables users to access or manip-ulate data as organized by the appropriate data model:

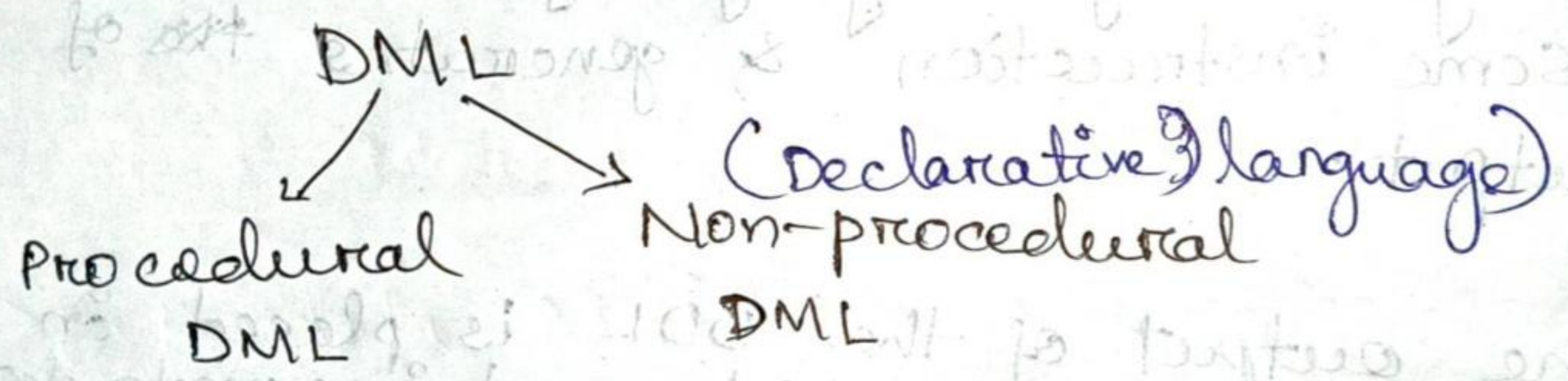
→ The types of access are ;

① Retrieval of Information

② Insertion of new information to from the database.

③ Deletion of Information from the Database

④ Modification of information stored in the database.



→ Procedural DML required a user to specify what data are needed and how to get those data.

→ Non-Procedural DML requires a user to specify what data are needed without specifying how to get those data.

* A query a statement requesting the retrieval of information. The part

of a DML which involves in information retrieval is called query language (DML).

Dt-05.04.22

Data Model in DBMS

→ The Data Model is a collection of conceptual for describing data relationship, data semantic & consistancy constraints.

→ It provides a way to describe the design of the database of the - - - - - - -

Physical level, Logical level, view level.

Types of Data Model

1. Relational Model
2. Entity Relationship Model (E-R Model)
3. Object oriented Model.
4. Semistructured Model

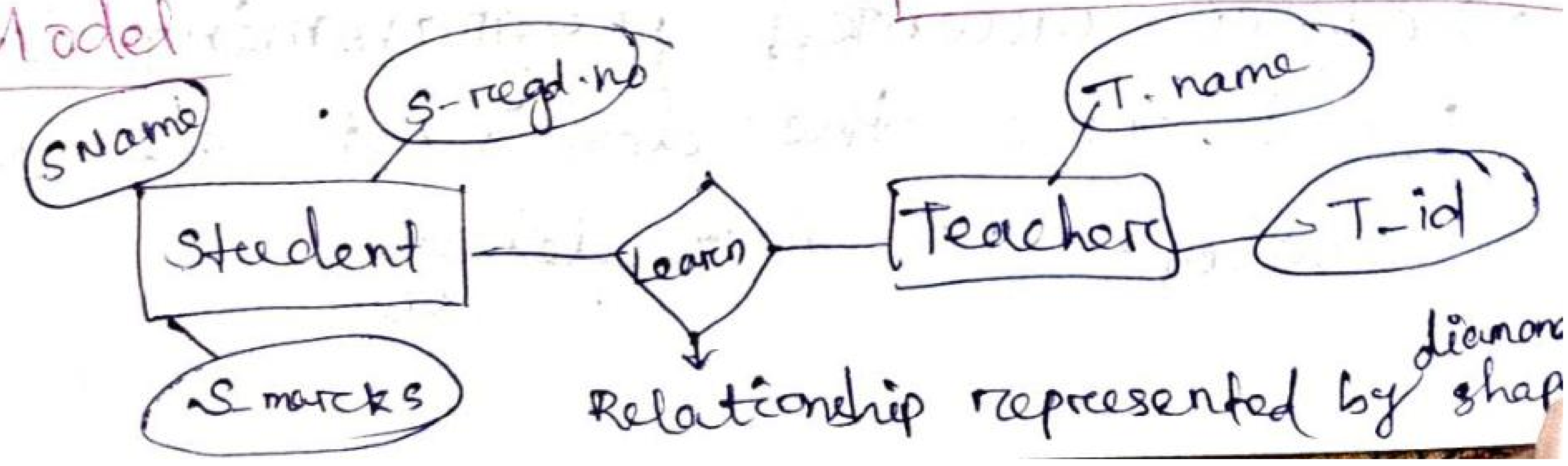
Relational Model :-

Row → Tuple
Col → attribute/field

Acc no.	Balance	Bank name
1037	25000	SBI

Entity - means, things or Object.(

E-R Model



Relationship represented by diamond shape

Relational / Record Based Model :-

→ It uses a collection of table to represent both data and relationship among these data.

→ Each table has multiple columns & each column has a unique name.

→ Tables are also known as relation.

→ Record based Models are so named because database is structured in fixed format records of several type.

→ Each table contains records of a particular type.

E-R Model

→ E-R data Model uses a collection of basic objects called entities and relationship among these objects.

→ A entity is a thing or object in the real world, that is distinguishable from other object.

Object-Oriented Model :-

→ object-oriented programming (c++, java) has become the dominant software developement methodology.

→ Extending ER-Model with notions ~~of~~.
of encapsulation, method and object.

→ In this model using common attribute we are making the relationship.

Semistructured Model

→ It permits the specification of data. Where individual data item of the same type may have different set of attribute.

→ We can implement this data model using XML (Extensible mark-up language)

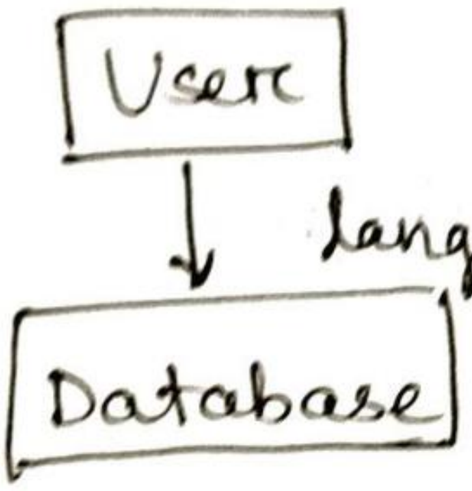
→ In this datamodel is entity may have Missing attribute or additional attributes.

ID - LAB

Dt- 22.04.22

SQL (Structure Query Language)

1970 → EF Code → Relational Algebra & Relational calculus



IBM
↓
SQL
↓
SEQUEL

① SQL is a Domain Specific language.

② SQL is also known as Declarative language.

③ DDL (Data Defination language)
DML (Data Manipulation language)
DCL (Data Control language)
TCL (Translation control language)
Contraints

④ Operators

⑤ Keys

⑥ Aggregation operation

⑦ Join

⑧ PLSQL

DDL

```
CREATE TABLE STUDENT
(
    NAME    VARCHAR (10),
    REGDNO  NUMBER (20),
    SECTION VARCHAR (10).
);
```

Attribute — Datatype.

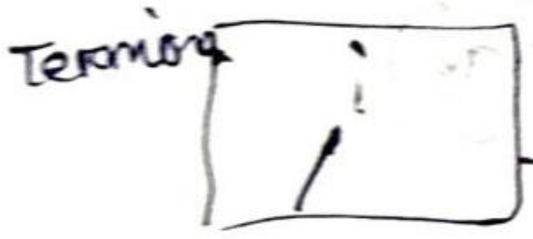
↳ variables are called attribute.

* Open terminal , iSQLPLUS

Username - scott
Password - tiger

→ DESC <Table name>

→ ED (for edit code)



Assignment-1

1// Create Tables with theire respective columns; datatypes and size.

(i)
```
CREATE TABLE INSTRUCT_SK
(
ID NUMBER(5), // NUMERIC
NAME VARCHAR2(30),
DEPT_NAME VARCHAR2(20),
SALARY NUMBER(6)
);
```

```
SQL> desc INSTRUCT_SK
```

(ii) CREATE TABLE COURSE-SK
(
COURSE_ID VARCHAR2(10),
TITLE VARCHAR2(30),
DEPT_NAME VARCHAR2(20),
CREDITS NUMBER(2).
);

SQL> DESC COURSE-SK

(iii) CREATE TABLE PREREQ-SK
(
COURSE-ID VARCHAR2(10),
PREREQ_ID VARCHAR2(10)
);

SQL> /

SQL> DESC PREREQ-SK

(iv) CREATE TABLE DEPARTMENT_SK
(
DEPT-NAME VARCHAR2(20),
BUILDING VARCHAR2(20),
BUDGET NUMBER(10)
);

SQL> DESC DEPARTMENT_SK

(v) CREATE TABLE TEACHES_SK
(
ID. NUMBER(5),
COURSE-ID VARCHAR2(10),
SEC-ID NUMBER(2),
SEMESTER VARCHAR2(10),
YEAR NUMBER(4)
);

SQL> /

SQL> DESC TEACHES-SK

Q// CREATE TABLE EMPLOYEE_SKD
(
PERSON_ID INT (255),
LASTNAME VARCHAR2 (20),
FIRSTNAME VARCHAR2 (20),
ADDRESS VARCHAR2 (20),
CITY VARCHAR2 (10)
);

Dt-30.04.22

Q// What is Oracle?

→ • Oracle is company made mostly usable server based or meltiuser based RDBMS.

• Oracle Supports SQL.
• It supports a client server architecture.

DDL:- (Data Defination language)

→ It is a set of SQL Commands used to create, modify and delete database structures but not data.

CREATE , ALTER , DROP , TRUNCATE

↓ (Create object of a Database)

↓ (Alter the Structures of a Database)

↓ (Delete object)

↓ (Remove all records from a table including all spaces allocated for records are removed.)

COMMENT

↓ (Add comment to the data dictionary)

DML (Data Manipulation Language)

Insert.
Update — allows to change the data within database.
Delete

Data type :-

Char(M)	Date / Date, time
Varchar(M)	Number
Varchar2(M)	INT / Integer.
Float	SmallInt
Decimal / Decimal(P), Decimal (P,S)	BIGINT;
Long.	

> username :-
> csej19411012833
> Password :- ~~nigam~~ " "

Create Table SecJ
(today date, datetime ;

- today → col^n name.
- date, datetime → Datatype

Inserting Data into table :-

→ Using insert query you can add data into a table one row at a time. You can't create one column or table in one Command

```
create table employee
( name Varchar2 (10),
  age int ;
  emp-id number (10));
);
```

*Insert into employee values ("vijaya", 35, 4041);

Syntax (type-1)

INSERT INTO <table-name> (col1, col2, ---
VALUES (val1, val2, ---, valn); ---coln)

Syntax (type-2)

INSERT INTO <table-name>
VALUES (val1, val2, ---, valn);

Syntax (type-3)

INSERT INTO <table-name>
VALUES (&COLUMN1, &COLUMN2, ---
&COLUMNn);

ex:- Insert into Employee
Values ('&name', &age, &Emp-id);

When we will run it, it will ask for values like scanner class.

Veiwing Data in Tables

Select * from Employee;

~~DESC~~ Table Details Check

DESC TABLE_NAME;

SQL QUERY

Syntax: COMMIT;
↳ will permanently save any changes made to any table in the database!

Basic Data types :-

① CHAR(M): This is used to store charactere string values of fixed length where M is the length of the string. This data type can holds maximum number of 255 characters.

② VARCHAR(M) / VARCHAR2(M): It is more flexible form of the character datatype. It can store variable length alphanumeric data. It can holds maximum upto 4000 characters.

③ DATE / DATETIME: It is used to represent Date and Time. The standard format for Date is DD-MM-YY as in 21-Aug-17. DateTime stores date in 24 hours format DD-MM-YY, HH-MM-SS.

④ DECIMAL / DECIMAL(P), DECIMAL(P,S):

Like Number specification but the storage length is a minimum specification i.e. Greater length are acceptable but smaller lengths are not.

⑤ FLOAT: Maximum precision 38 digits. Only use when the number has decimal form.

⑥ RAW / LONG RAW: It is used store binary data such as digitized picture or image.

⑦ LONG: To store variable length character Strings containing upto 2 GB Long data. It can be used to store binary data in ASCII format.

⑧ INT/INTEGER: Can store whole numbers can't store decimal numbers upto 11 digits.

⑨ SMALLINT: Integers upto 6 digits can be stored.

⑩ BIGINT: Integers upto 20 digits can be stored.

⑪ NUMBER/Number(P), Number(P,S) :

The Number data type is used to store numbers (fixed or floating point) including sign. Numbers of virtually any magnitude may be stored upto 38 digits of precision.

Ex:- number(4,2) = 12.32, number(5,2) = 132.25

P = determines max^m length of data.

S = scale determines the number of places to the right of the decimal.

User id = S1941012833

Password = Iter1234##

* If error go to Databases (double click) → then S194.. → New Query (right click) → write → execute

* See tables, S1941.. → Tables → dbo.EMS (right click) → Design.

→ Extending ER-Model with notions of encapsulation, method and object.

→ In this model using common attribute we are making the relationship.

Semistructured Model

→ It permits the specification of data where individual data item of the same type may have different set of attribute.

→ We can implement this data model using XML (Extensible mark-up language)

→ In this datamodel is entity may have missing attribute or additional attributes.

Database User and Database Administrate

Dt - 10.05.22

→ Database users are catagorised based up on there interaction with the database

① DBA (Database Administrative)

→ DBA is a person or team to defines the schema and also controls the 3 levels of database.

→ DBA will then create a new account id and password for the user if

he/she needs to access the database.

→ DBA is also responsible for providing the security to the database and he allows only the authority users to access or modify the database.

→ DBA also monitors recovery and back up & provide technical support.

→ The DBA has a account in the DBMS which is called a System / Super user account.

→ DBA repairs damage caused due to hardware and software Failure.

② Naive / parametric end user :-

→ Parametric End Users are the unsophisticated who do not have any DBMS knowledge but they frequently used the database application in their daily life to get the desired result.

③ System Analysis :-

→ System analysist is a user who analyses the requirements of parametric end user. ~~are satisfied or not~~.

→ They check whether all the requirements of end user are satisfied or not.

④ Sufficticated User :-

→ Sufficticated users can be a engineer, scientist, business analyst who are familiar with (db) database.

→ they can develop their own database application according to their requirement.

⑤ Database Designer :-

→ Database designer who designs the structure of database which includes tables, indexes, constraints, triggers, stored, procedures.

→ He also controls what data must be stored & how the data items to be related.

⑥ Application Programmer :-

→ Application programmer are the backend programmer who writes the code for an application ~~teacy~~ programmer.

→ This programs should be written as, such as Java, C, ---.

⑦ Casual / Temporary User :-

→ Casual user who occasionally use or access database but each time when they access the database they requires the new information.

e.g:- Middle or Higher level Manager.

⑧ Relational Database

→ It is based on relational model and uses a collection of table to represent both data and relationship among those data, it also includes DML and DDL.

e.g Select * from instructor;
select name-ID from instructor;

Database Access From Application Program:

- ODBC, JDBC.

→ SQL is not powerful as a universal turing machine that is there are some computations that are possible using a general purpose programming language but not possible using SQL.

→ SQL also doesn't support actions such as input from user and output to display, communicate over the network.

→ Such computation and action must be written in a host language such as: C, C++, Java, ---- with embedded SQL Queries that access the data in the database.

→ Application programs are programs that are used to interact with db in this fashion.

Ex:- for a university systems are programs that allows student to register for courses, calculate student CGPA generate payroll cheques.

JDBC, ODBC :-

→ To access the database DML statement needs to be executed from the host language.

→ There are 2 ways to do this;

(i) By programming some application program interface that can be used to send DML & DDL statement to the db & retrieve the results.

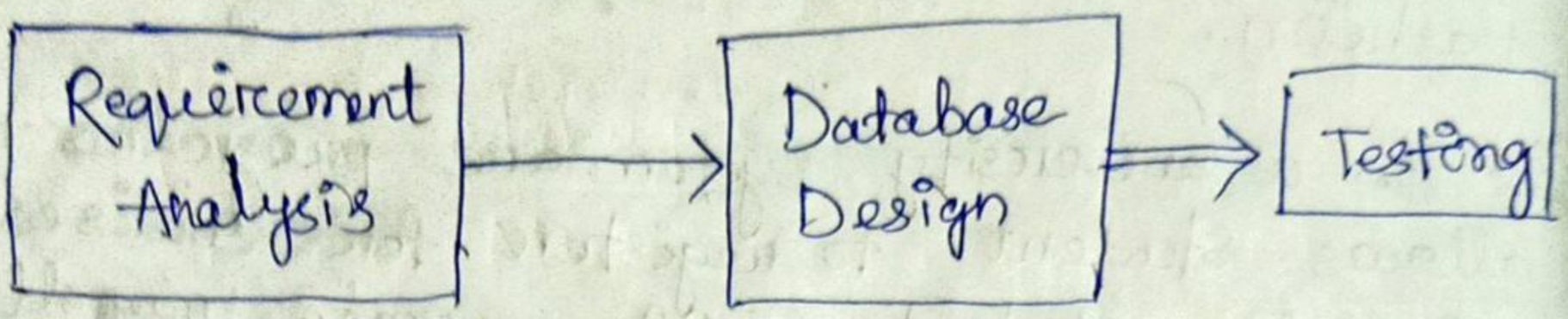
→ The ODBC standard for use with the c language is a commonly used application Program interface standard.

→ the JDBC standard provides corresponding features to the java language.

(ii) By extending the host language syntax to embed DML call within the host language program. Usually a special character Prefaces DML calls & every processor called DML precompiler, converts the DML statement to normal procedure called in the host language.

Dt- 11.05.22

Database Design



→ Database Design is the collection of processes that facilitate the designing, developement implementation and maintainance of enterprice data management System.

→ Properly design database are easy to maint-ain, improves data consistency, and are cost effective interms of storage space.

→ The database design are decide how the data elements co-relate and what data must be stored.

→ The main objective of database design in DBMS are to produce logical and physical design model of the proposed database System.

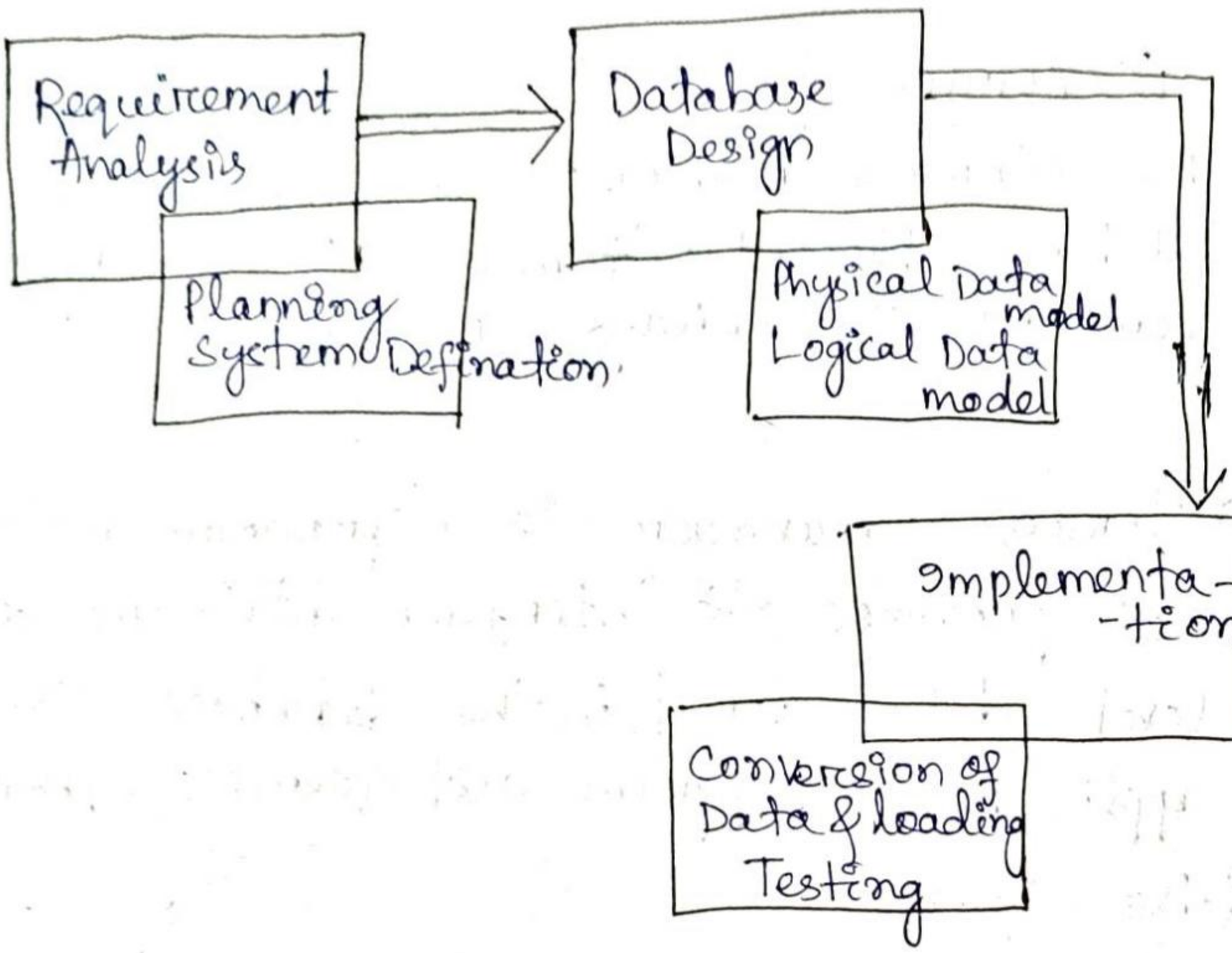
Q// Why database design is important?

Ans:- Database design is important because,

(i) It meets the requirement of user.

(ii) It is used to increase the performance of the database system.

Database Developement life cycle (DDLC)



→ 2 types of Database Design Approach.
① ER-Model (next chap.)
② Normalization

② What is Normalization?

→ Using Normalization Technique Database designer normalize what designs are bad and test for them.

→ Allows to store information without any redundancy.

Data Storage Management and Quercying

(Query Processing)

→ the functional components of Database system can be divided into 2 types;

(i) Storage Manager
(ii) Query Processing

(i) Storage Manager :-

→ The storage Manager is important b'coz database typically requires a large amount of storage space.

→ Storage manager is a program module that provides the interface betⁿ the low level data stored in the database and application program and queries submitted to the system.

Different Components of Storage Management:

① Authorization & Integrity Manager :-

→ It tests integrity constraints and checks authority of users to access data.

② Transaction Manager :-

→ Transaction manager ensure that database remains in the consistent (correct state) despite system failure and Transaction failure.

③ File Manager :-

→ Manages allocation of space on disk storage.

(i) Buffer Manager :-

↳ Buffer Manager fetching data from disk to memory.

* The storage manager implements following data structures.

(i) Data files → where the database is stored.

(ii) Data Dictionary → Meta data about schema of database.

(iii) Indices → Fast access to data.

(ii) Query Processor :-

→ The Query processor is important b'coz it helps the database system to access the data.

Different Components of Query Processor :-

① DDL Interpreter →

→ Which interpretes DDL statement and records the definition in the data dictionary.

② DML Compiler →

→ Which translates DML statements into an evaluation plan consisting of low level instruction that the Query evaluation engine understand.

Query evaluation Engine →

Which executes low level instructions

generated by the DML compiler and returns the answer to the query.

DT- 12.05.22

Chapter-2 Entity Relationship Model / E-R Model

Entity	Person (name, salary, Age)
Attribute	Student (name, id, Age)
Relationship	Employee (name, id, Age)

* Entity :-

→ An Entity is an object that exist and distinguishable from othere objects.

or//,

→ An Entity is a thing in the Real world with an independent existance.

or//

→ An entity is an object on real world that has certain properties.

Entity Types :-

→ Entity type is an abstraction defining the properties (attributes) of similar set of Entity.

Entity Instance :-

→ Entity instances are instantiation of entity types.

Entity Set :-

→ An Entity set is a set of entity instances that act of the same type.

Ex:- All persons having an account in SBI is an Entity set.

→ All students belong to CSE is an Entity set.

* Attribute :-

→ Attributes are used describe either entities or relationship.

More Attribute ⟶ Better information.

→ Each attribute has a domain from which the values for this attribute are domain.

Different types of Attributes

① Single value Vs	Multivalue Attribute
→ for any student where only one roll no., and one name.	→ Means which take more than one value. ex:- phone no. & address

② Simple Attribute Vs	Composite Attribute
→ That attribute are not divisible, are called simple or	→ composite attributes can be divided into smaller sub parts.

atomic attribute.

for example :- address of a Student can be Sub divided into plot no., street, city, Pin code.

---

③ Stored Attribute VS Derived Attribute

(DOB)

→ The Attribute which can not be derived is called stored attribute.

(Age)

→ the attribute which can be derived from other attribute is known as derived attribute.

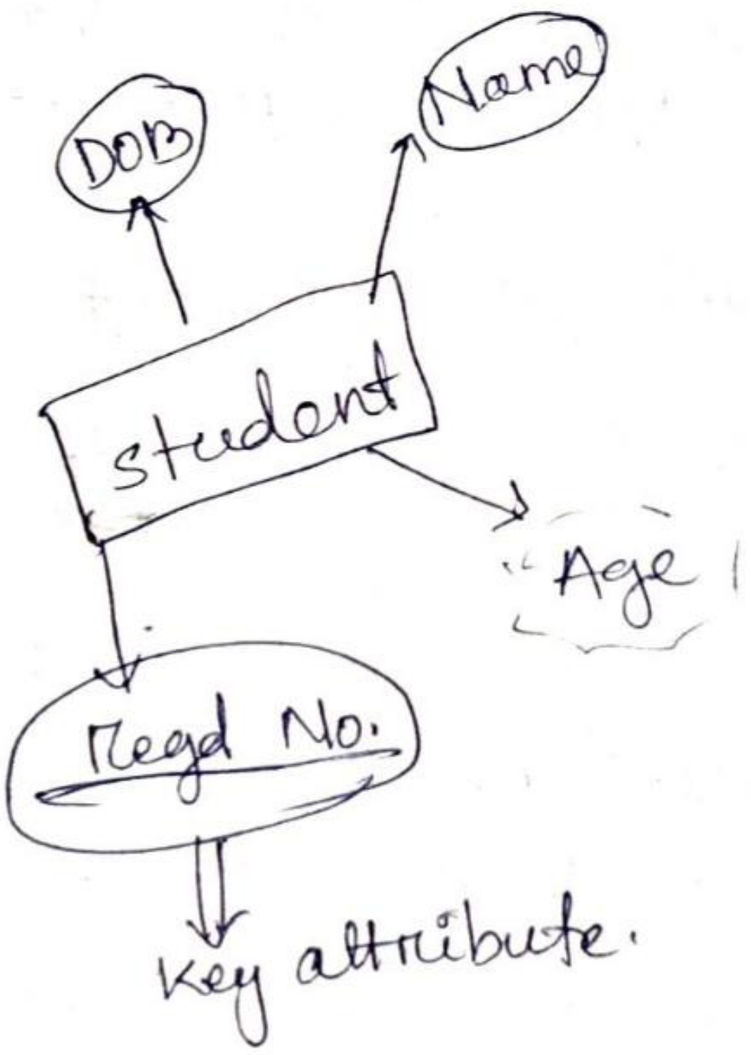
---

④ Key Attribute VS Nonkey Attribute

→ Key attribute always represented by underline.

→ Key (unique) attribute is used to uniquely identified each and every row.

⇒ It is having a unique value. (that will be distinguish from any other value)



⑤ Required Attribute Vs Optional Attribute

→ Required Attributes mean which value are mandatory. or optional.

⑥ Complex Attribute :-

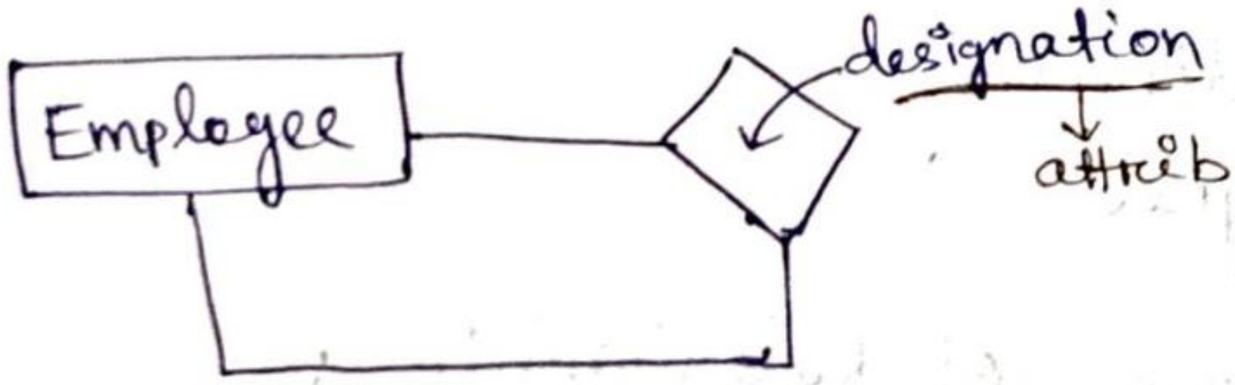
composite + Multivalue = Complex

e.g. Pin code, address

Dt - 17.05.22

Relationship :-

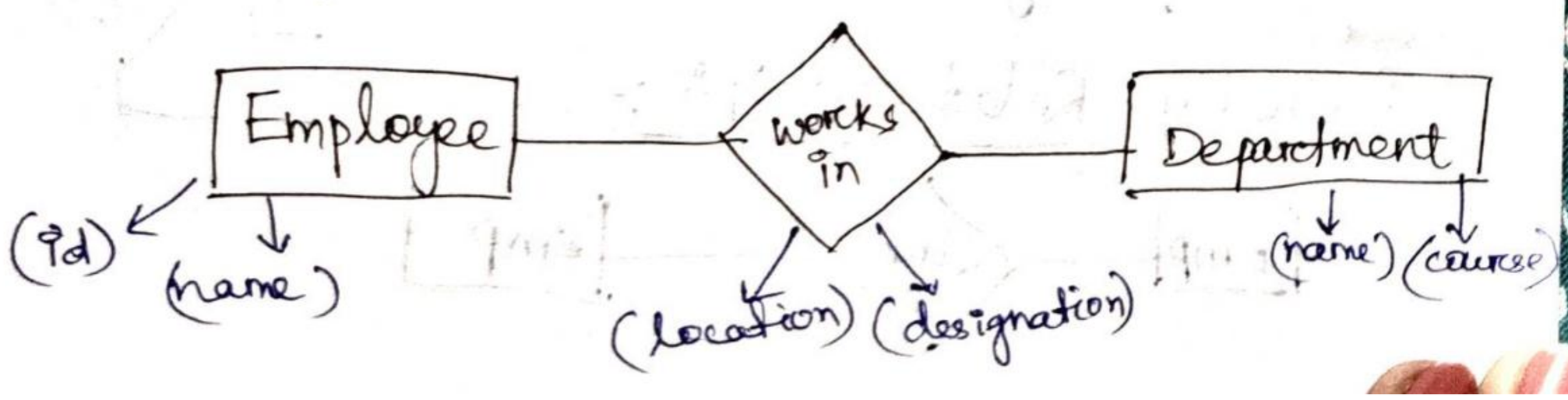
→ Relationship is association between one or more entities.

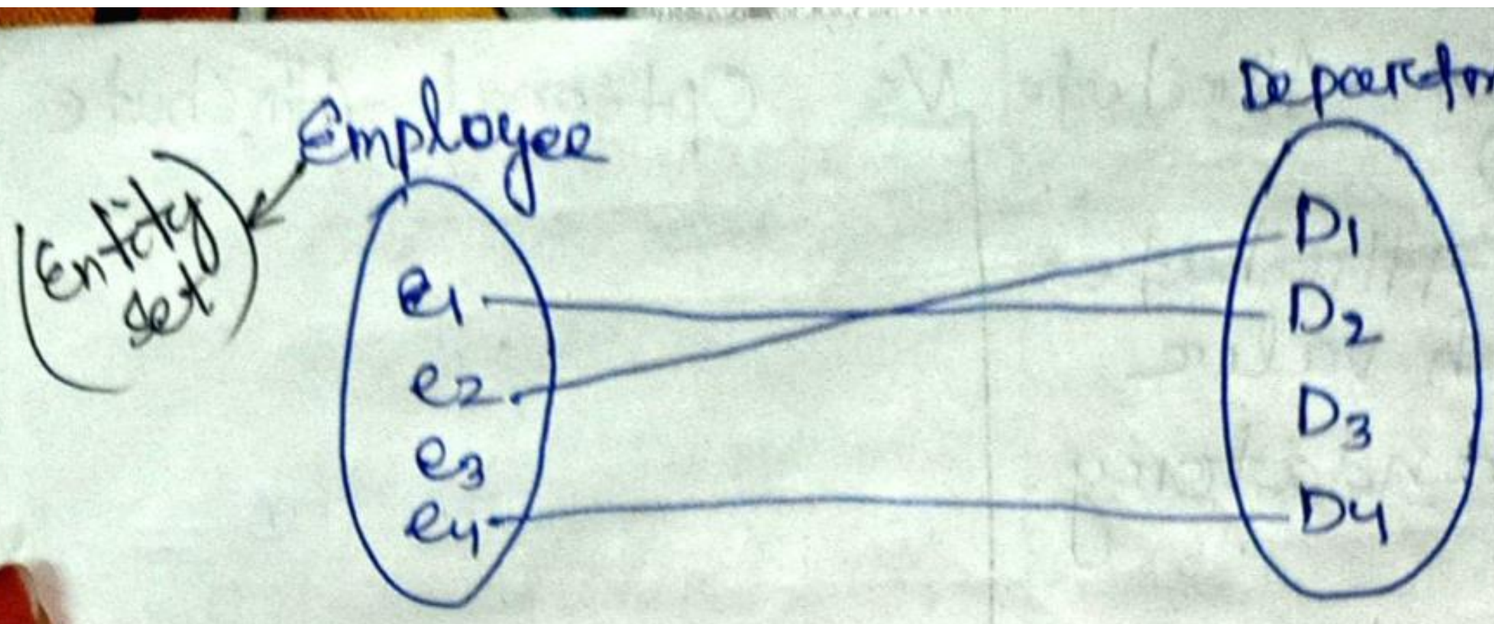


* Unary relationship is there or not? Yes, there

→ Association among several entities is known as Relationship.

→ It connects different entities through a meaningful relation.





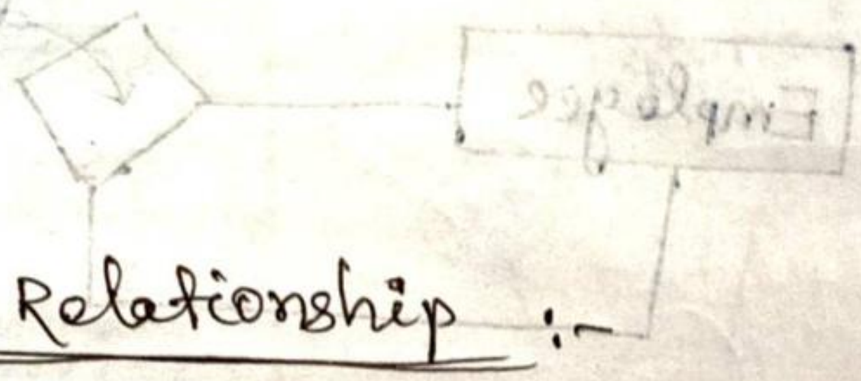
Relationship Set :-

→ Relationship set is a set of relationship on same type.

(e1) works in (D2)

→ Attribute of any relationship set is known as descriptive attribute.

→ Attribute which connects two entity set.
→ Every relationship has 3 identity;
(i) Unique name
(ii) Cardinality
(iii) Degree

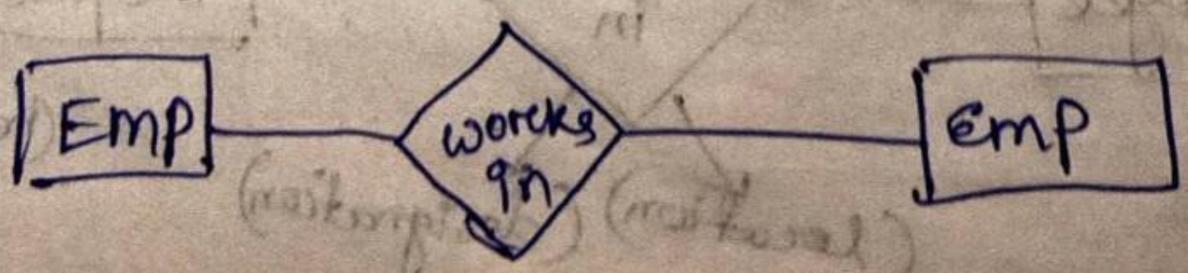


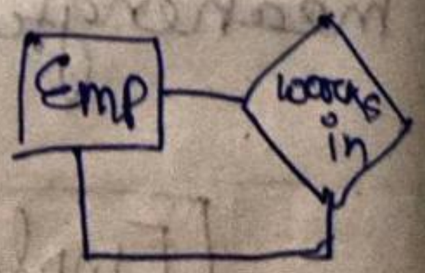
Degree of Relationship :-

4 types of Degree of relationships.

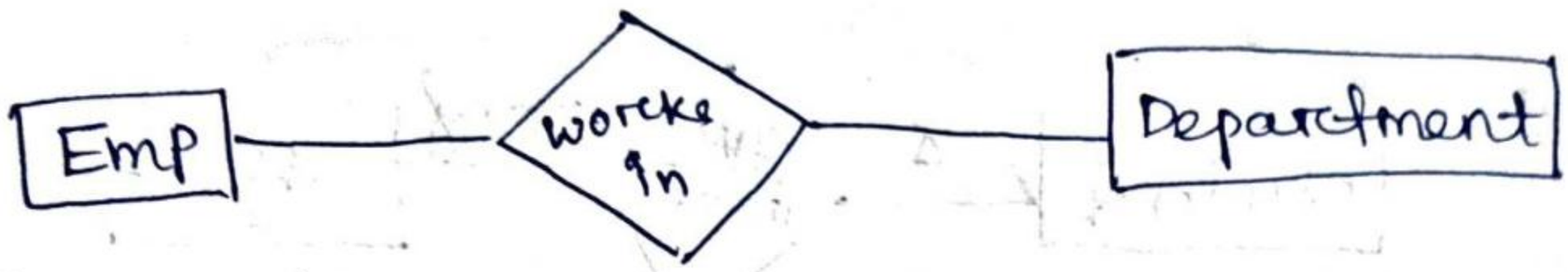
① Unary relationship
② Binary "
③ Ternary "
④ N-ary "

① Unary Relationships :-

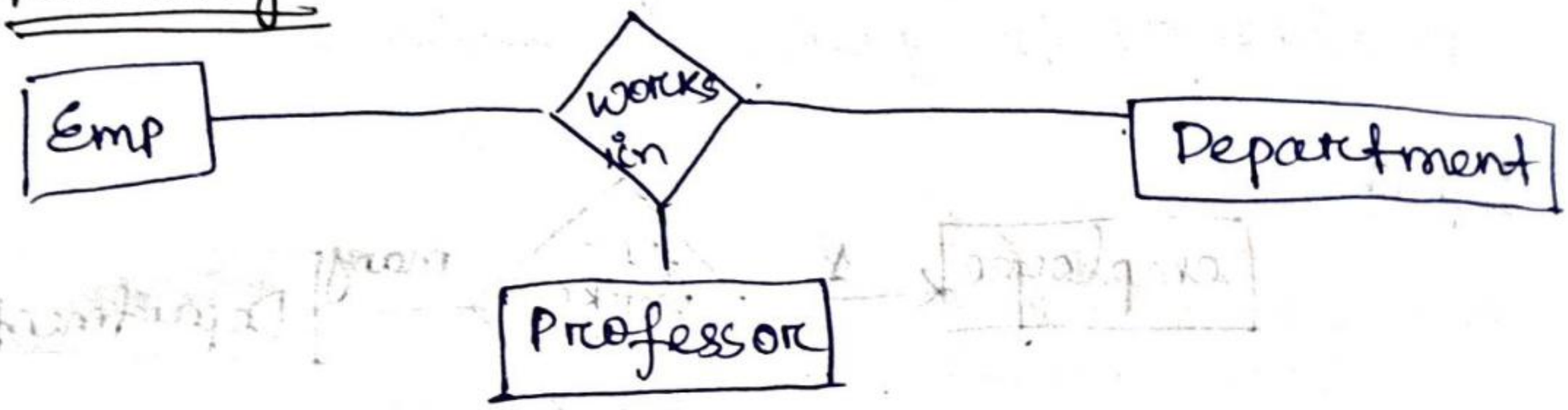




② Binary Relationship :-



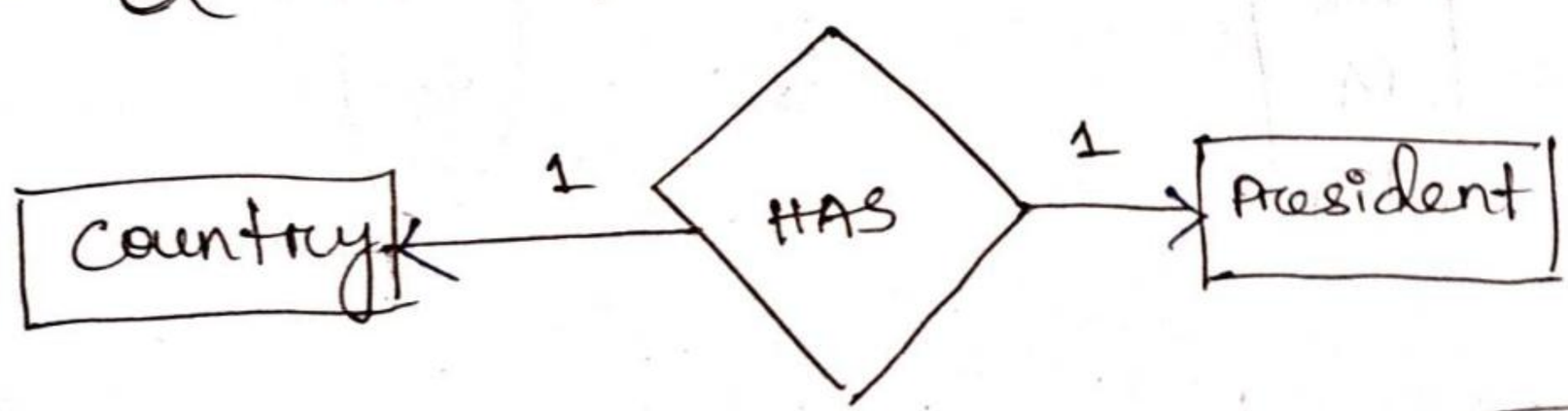
③ Ternary :-

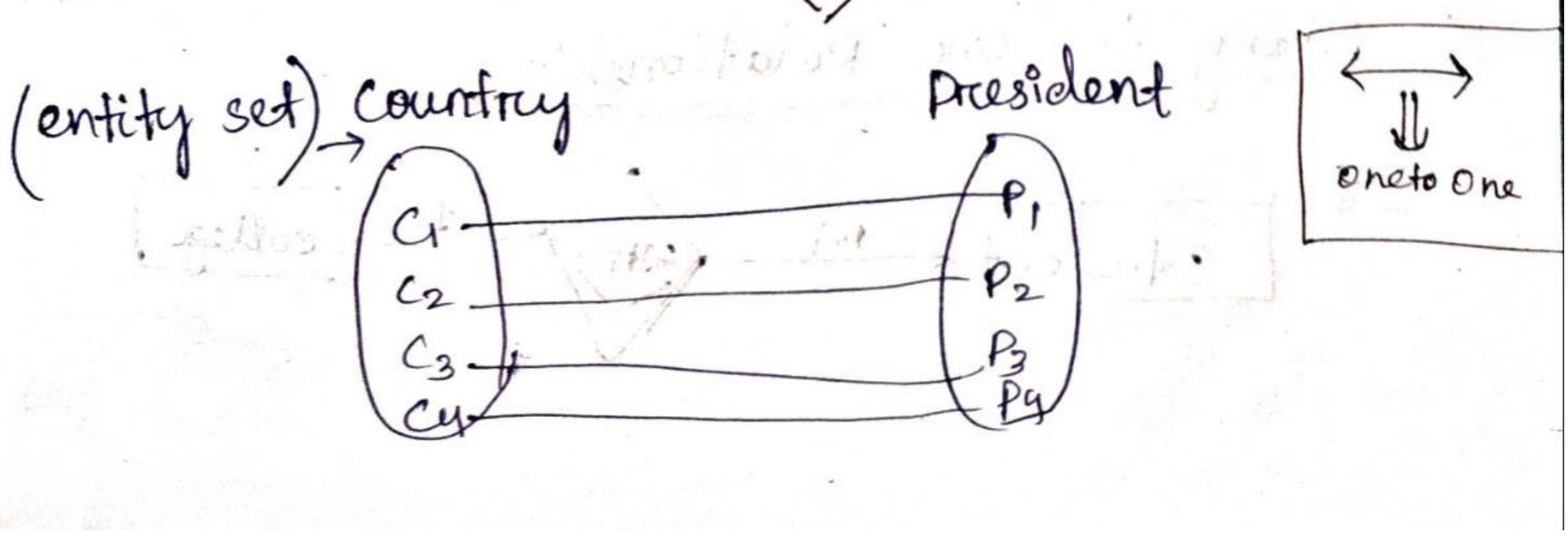


→ If total no. entity set participate in a relationship set is known as degree of relationship.

Mapping Constraint / Cardinality Constraint / Cardinality ratio :-

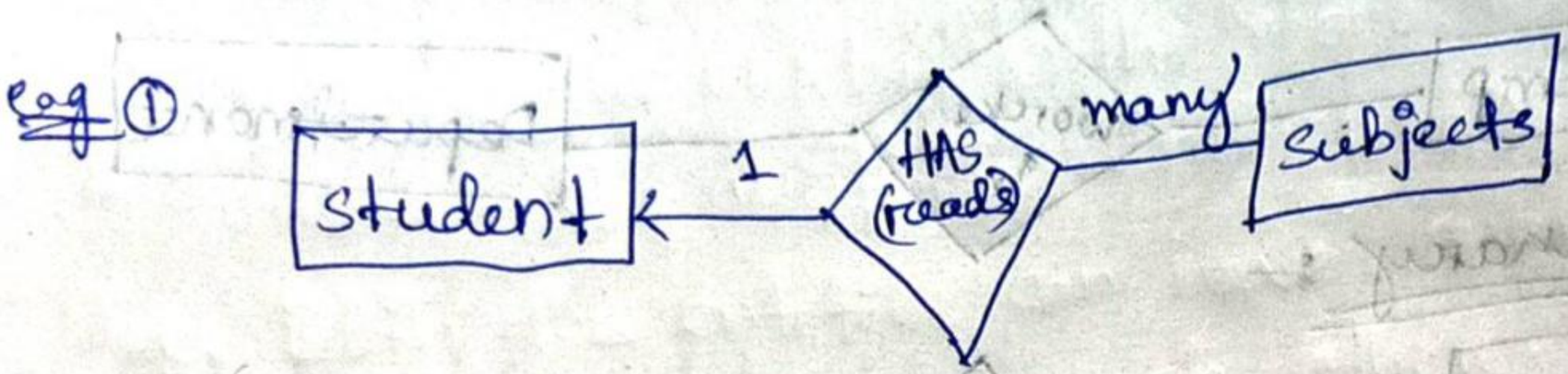
→ It specify the no. of entities of an entity set that are associated entities of another entity set from relationship set. @(Relationship is one to one)



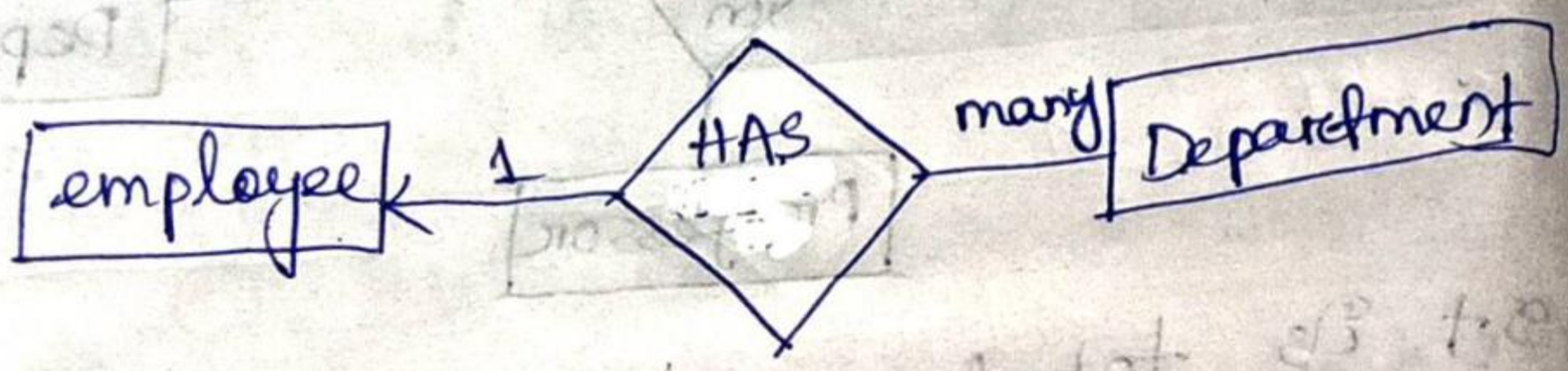


(b) One to many Relationship :-

eg ①

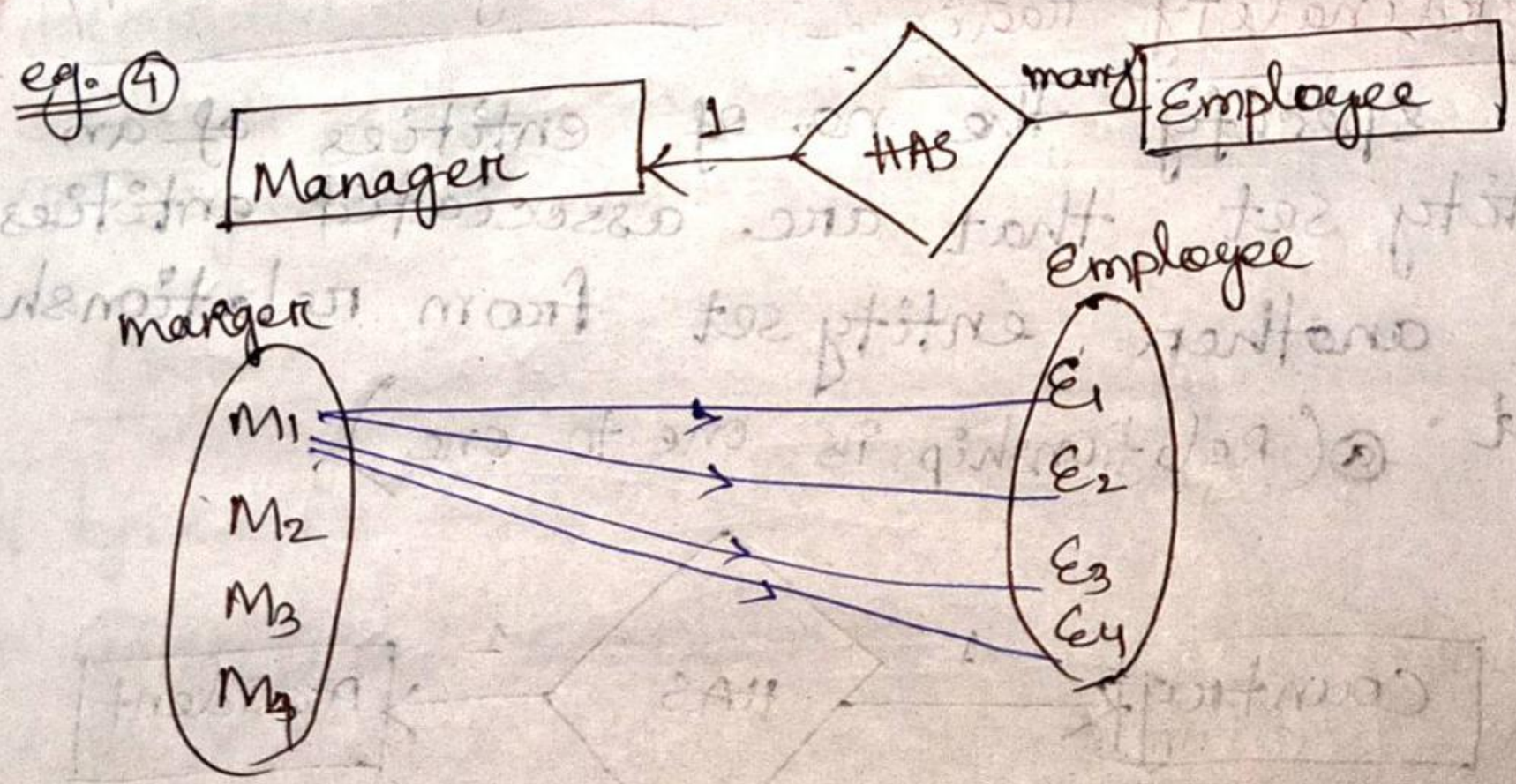


eg. ②



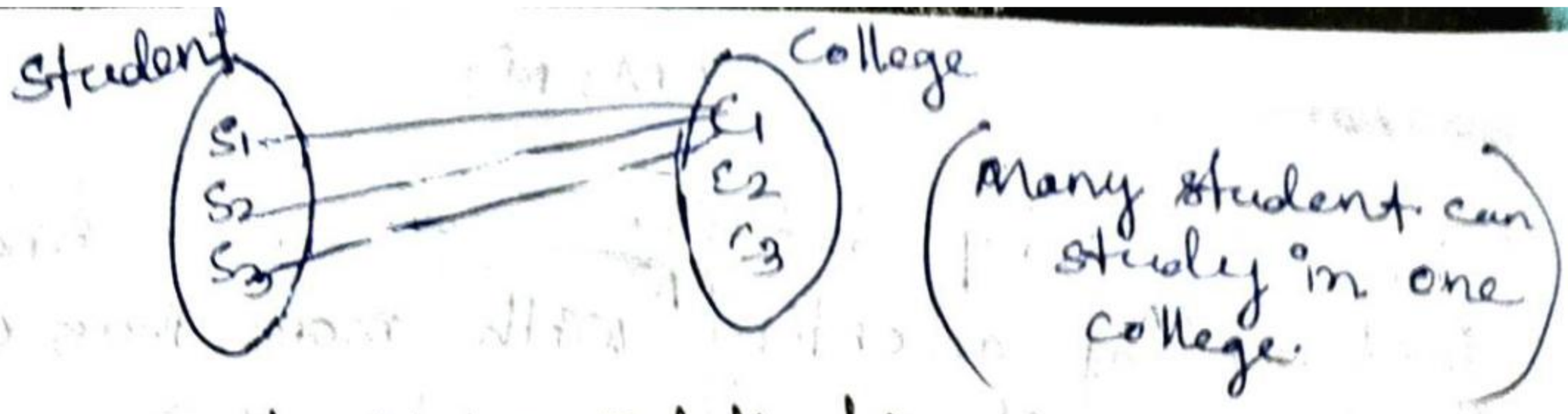
e.g ③

eg. ④

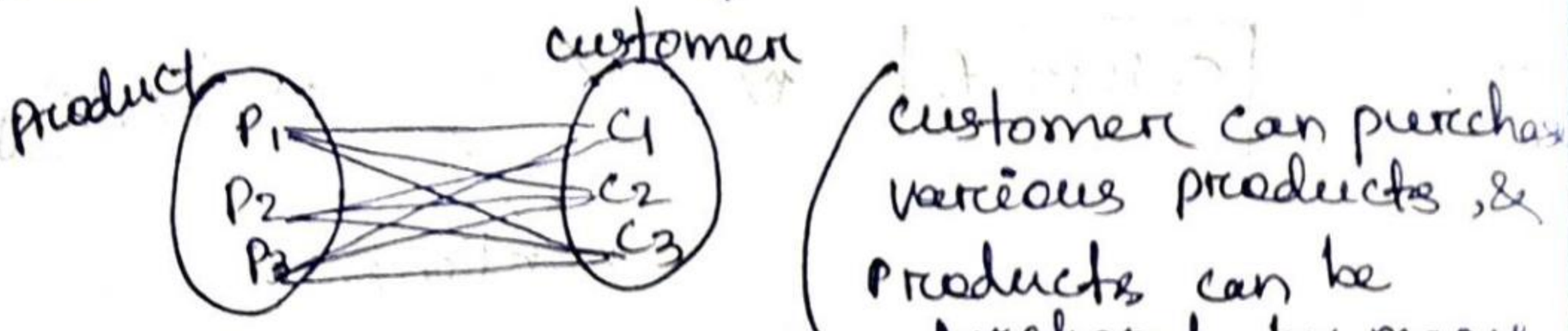


(c) Many to One Relationship

eg



④ Many to Many Relationship :-



One to One :- (1 : 1)

An Entity in A associated with atmost one entity in B and an entity in B associated with atmost one entity in A.

One to Many :- (1 : M)

An entity in A is associated with any no. of entities in B, and an entity in B however can be associated with at most One entity in A.

Many to One :- (M : 1)

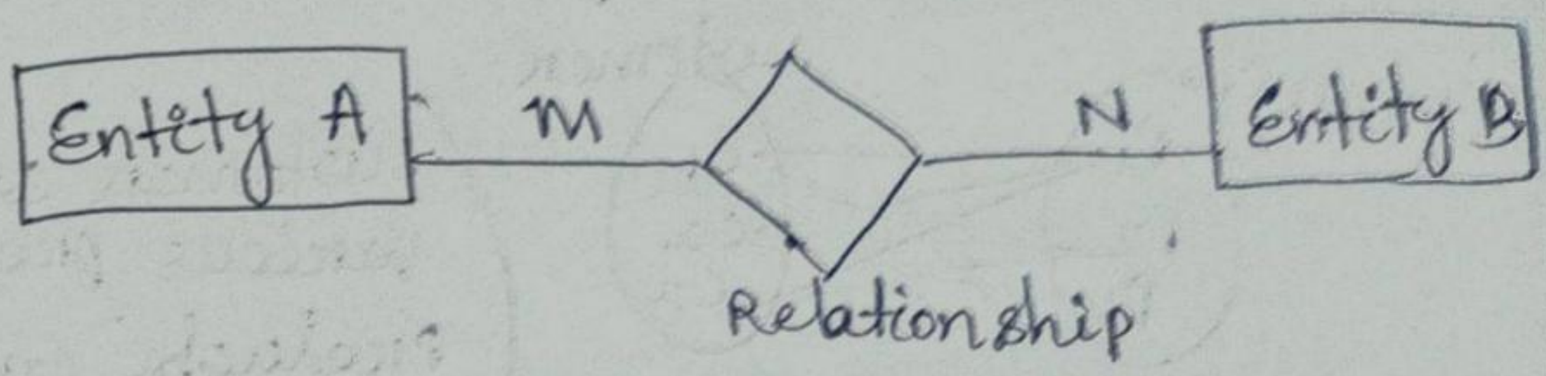
Any no. of entity in A is associated with ~~many no. of or more than~~ atmost one entities in B and the entity B can associated with any no. of entities in A.

or/ more than one instances of an entity with one instance of another entity.

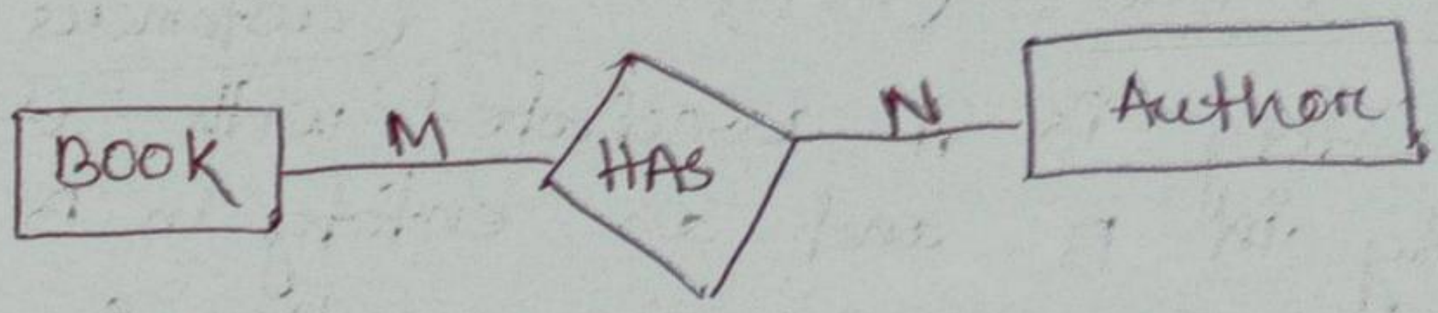
Many to Many :- (M:N)

→ The relationship between more than one instance of an entity (A) with more than one instance of another entity (B).

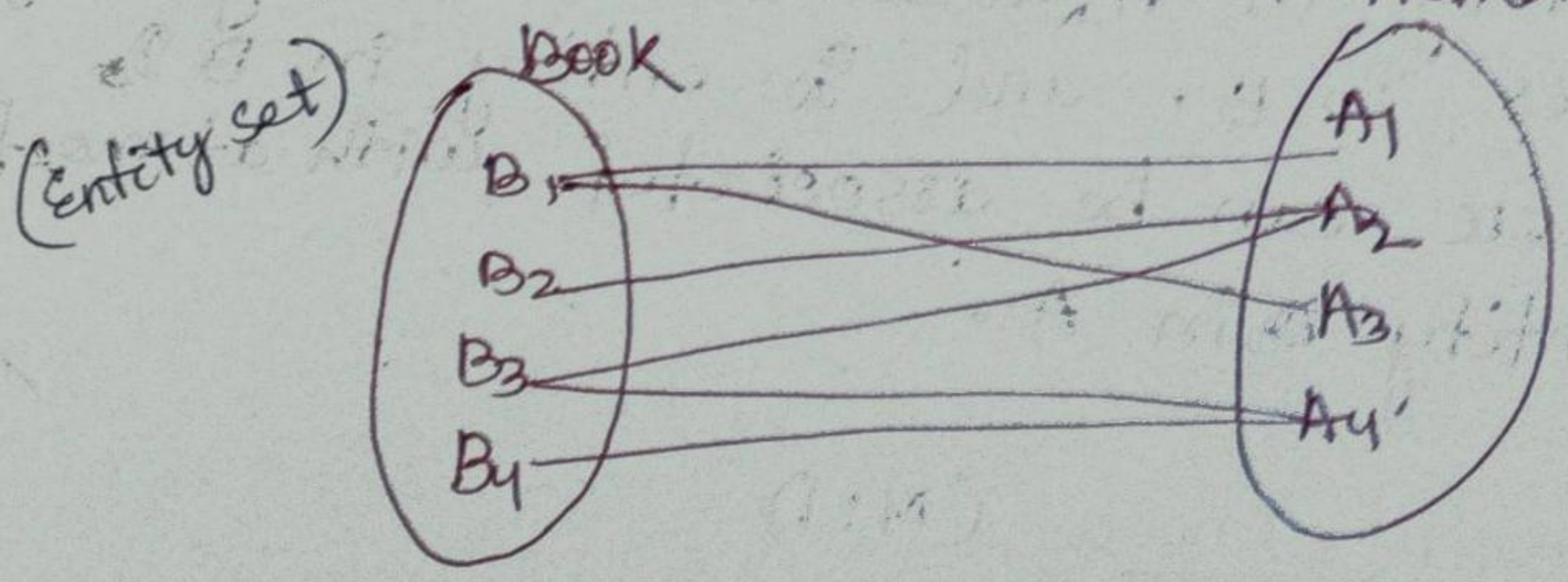
i.e. Both the entities can have many relationship between each other.



e.g.



Many author can write a book, whereas An author has written more-than one book.

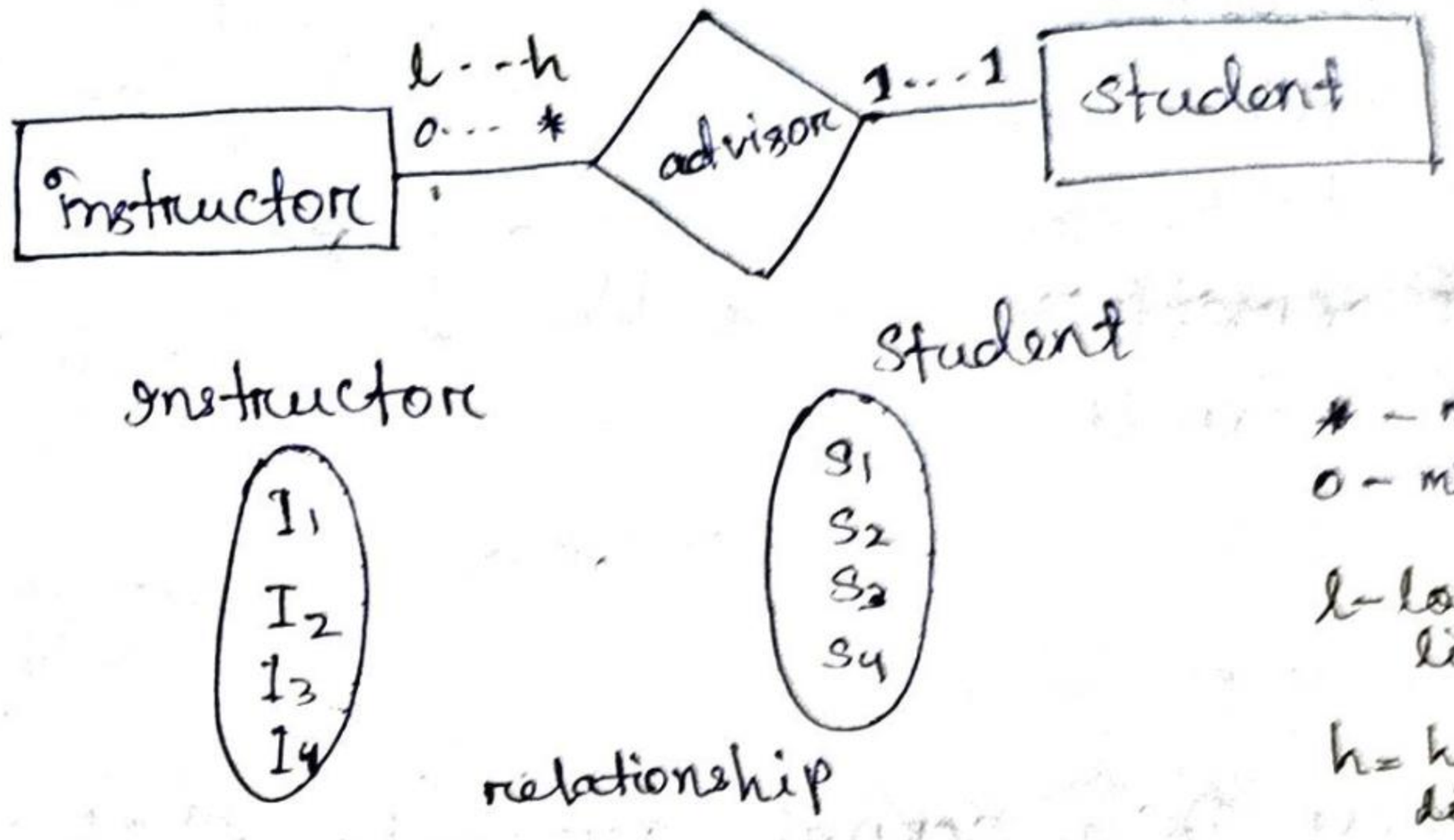


or//

* One entity in A can be associated with any no. of entity in B. similarly, one entity in B can be associated with any no. of entity in A.

Dt - 18.05.22

Mapping Cardinality and participation Constraints



→ On E-R diagram also provide a way to indicate more complex constraint on the number of time each entity participates in relationship in a Relationship set.

→ A line may have an associated min^m & max^m cardinality, shown in the form of l to h. where l is minimum & h is maximum cardinality.

→ In this example, An instructor can have 0 or more students (0..*)
(1..1) ⇒ each student must have exactly one instructor.

V.V.Imp

Database Key

Primary Key = unique + not null

Unique

① Roll no.
② Regd. no.
③ finger Print
④ Adhaar no.
⑤ PAN no.
⑥ ~~email id~~
⑧ Voter id

Student

Regd. No.	Sname	age	City
1	X	13	Delhi
2	Y	20	mumbai
3	X	13	Delhi

unique no.

(If we are choosen one P.K. other key are known as alternate key)

Key :-

→ Key is a normal attribute which is used to uniquely identify any two tuple or row in the table.

→ Different types of keys are there ;

① Primary Key (unique + not null)

→ It refers to a column or a set of colu-mns of a table that help us to identify all the records uniquely present in that table.

→ Each and every relation or table must have one primary key

→ The primary key constraint that we put on a columns or set of columns would not allow these to have a null value or duplicate value.

② Candidate Key :-

→ It refers to those attribute to identi-fy rows uniquely in a table.

→ In a table we select the primary key from a candidate key, there can be multiple candidate key.

③ Alternate Key :-

→ Any table can consists of multiple choice for the primary key. But primary key is only one choose.

→ Hence all those keys that didnot become a primary key are known as alternate key.

④ Super key :-

{Roll no.}
↓
PK

{Roll no, name}
↓ ↓
PK CK

{Roll no, name, age}
↑ ↓ ↓
PK CK CK

* Superkey is subset of Candidate key.

~~→ The candidate~~

→ The Superkey refers to the set of all those keys that help us uniquely identify all of ~~these columns of~~ the rows present in a table.

→ It means that all of these columns present in a table that can identify the columns of that table uniquely, act as the super key.

Eg R(A_1, A_2, A_3), total no. of Super Key?
of A_1 is C.K.

⇒ $\{A_1\}$, $\{A_1, A_3\}$, $\{A_1, A_2\}$, $\{A_1, A_2, A_3\}$

∴ no. of superkey = 4.

* R(A_1, A_2 ---- A_n) (n = no. of attributes)

Total no of Superkey = $\boxed{2^n - 1}$

⑤ Unique Key :-

* Diff. bet^n primary key & Unique key.
  - primary key → not null
  - Unique key → any single null value.

* Every unique key is a Candidate key.

→ A unique key refers to a column or a set of columns that identify every record uniquely in a table.

→ It is different from P.K. b'cox it is only capable of having one null value.

Dt - 19.05.22

Foreign Key

Student

PK

Roll no	name	age
1	A	20
2	B	25
3	C	30

Referenced Table (Base Table)

Course

FK

course no.	course name	Roll no.
1	IDB	1
2	N/W	2
3	DOS	3

PK

Referring Table

Insert → ✓
Delete → ✗
Update → ✗

Insert ✗ (For same roll no. U can)
Delete ✓
Update ✓

→ FK is an attribute or set of attribute that references to primary key of same table or another table.

→ It maintains referencial integrity.

e.g. (Real life example)

* On the Base table or Refrenced table :-

① Insert :- No problem, when we insert in base table

② Delete :- Problem :- If may goes violation. If we delete data from refrencing table.

Solution :- On delete cascade

③ Update (change in data) :- Problem create (inconsistency problem)

Solution :- On update Cascade.

* Refrencing Table

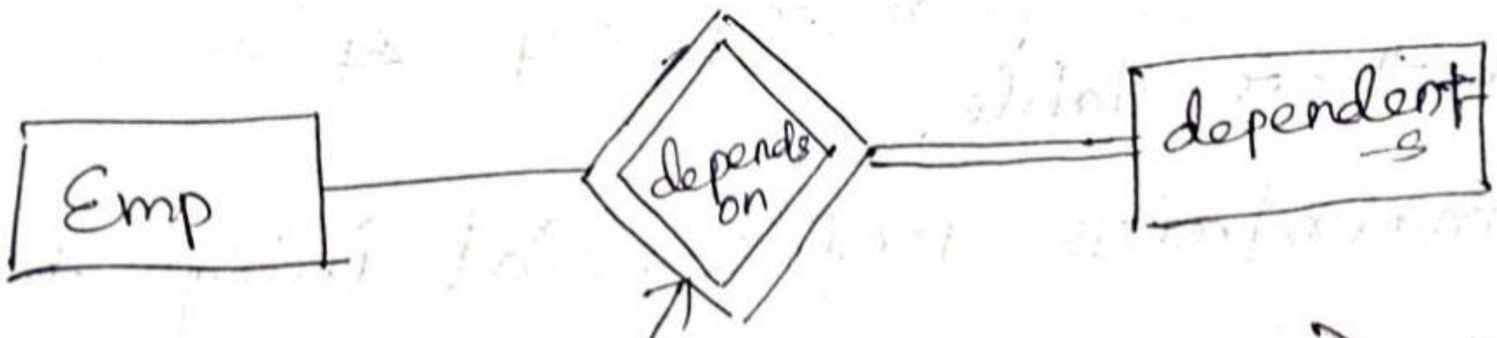
① Insert :- Problem :-
↑ It may cause violation to insert in refrencing table.

② Delete :- No problem, when we delete in refrencing table.

③ Update :- No problem, when we update in refrencing table.

Solution :- 1st insert value in base table then Refrencing table. with diff. PK which is not contained in base table

Key Attribute in Relationship Set



(identifying Relationship)

e.g. dependents completely depends employee but employee not completely depends on dependents.

eg. Me and My parents (not completely depends on me)
↓
(completely depends on parents)

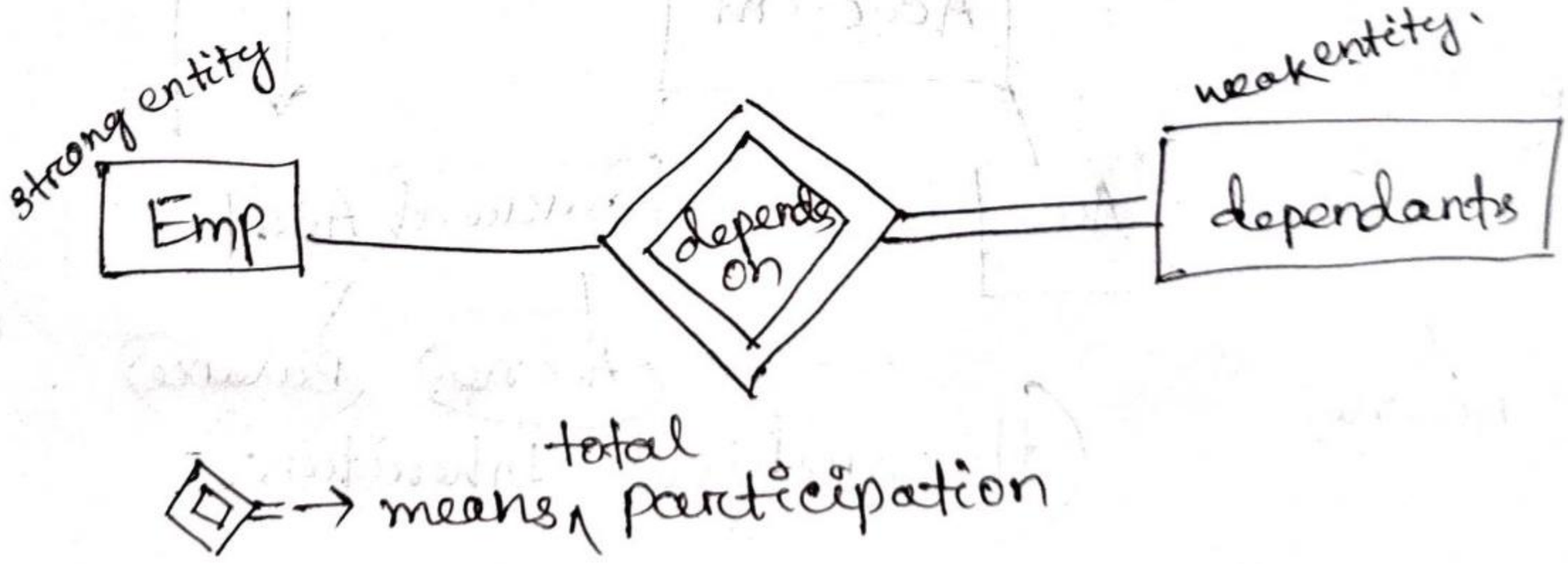
→ An entity types should have a key attribute which uniquely identifies each entity in the entity set. But there exist some entity type for which key attribute can't be defined. these are called weak entity type.

→ The entity set which don't have sufficient attribute to form a primary key are known as weak entity set. And the entity set which have a PK are known as strong entity set.

→ As the weak entities don't have any PK they can't be identified on their own. So that they depends on some other entity (owner entity we can say)

→ The weak entity have total participat-ion constraint in its identifies relation-ship with owner identity.

e.g:- Parent - child relationship
employee - dependants relationship
class - teacher relationship
order - order item

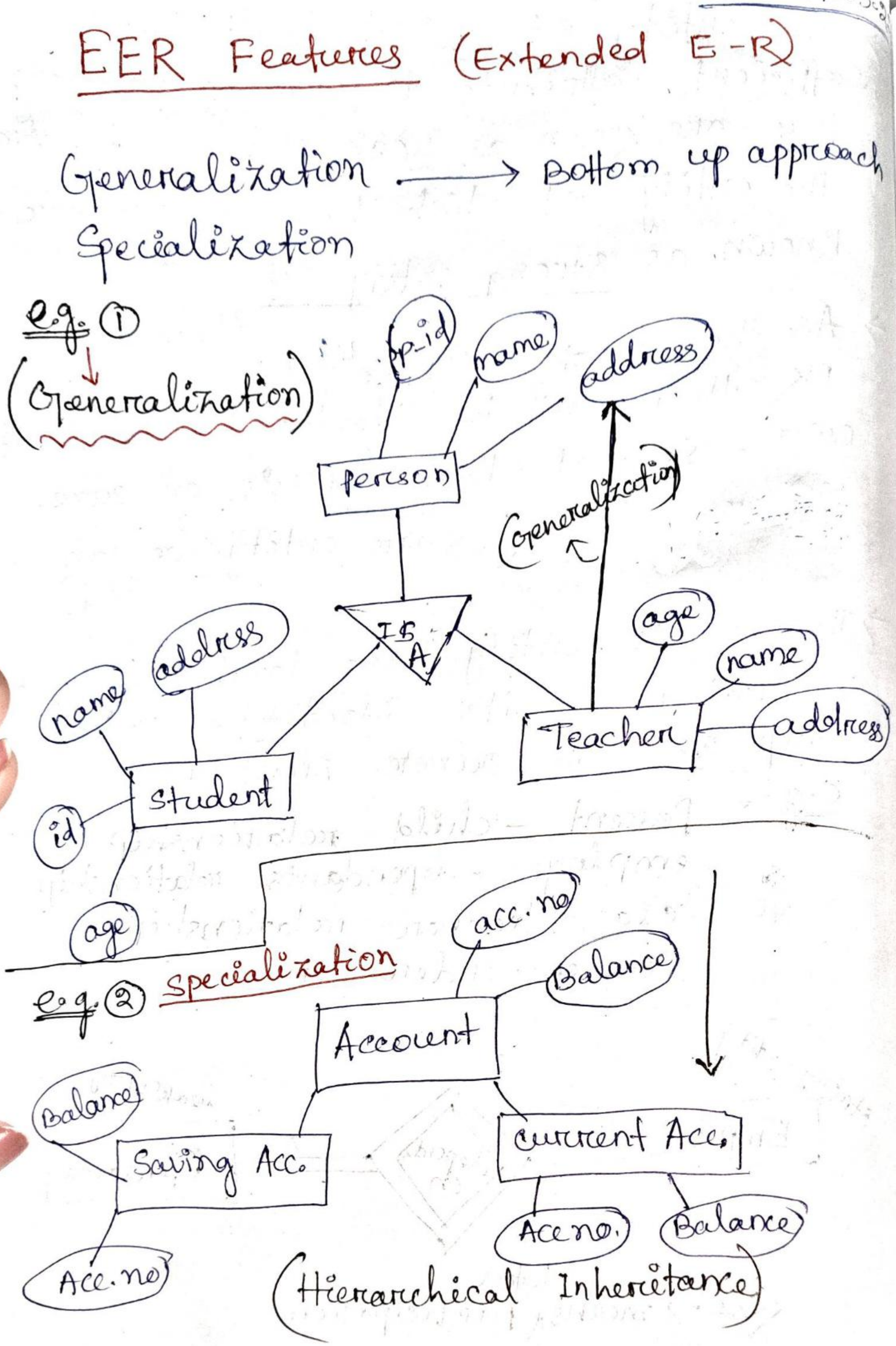


Dt - 24.05.2

EER Features (Extended E-R)

Generalization ⟶ Bottom up approach
Specialization

e.g. ①



Generalization

→ It is the process of extracting common attribute or property from in set of entities and create a generalized entities from it.

→ Multiple entities are combined to form a new attribute.

→ Generealization follows bottom-up approach. (more entity forms a single approach)

e.g. ①

Specialization

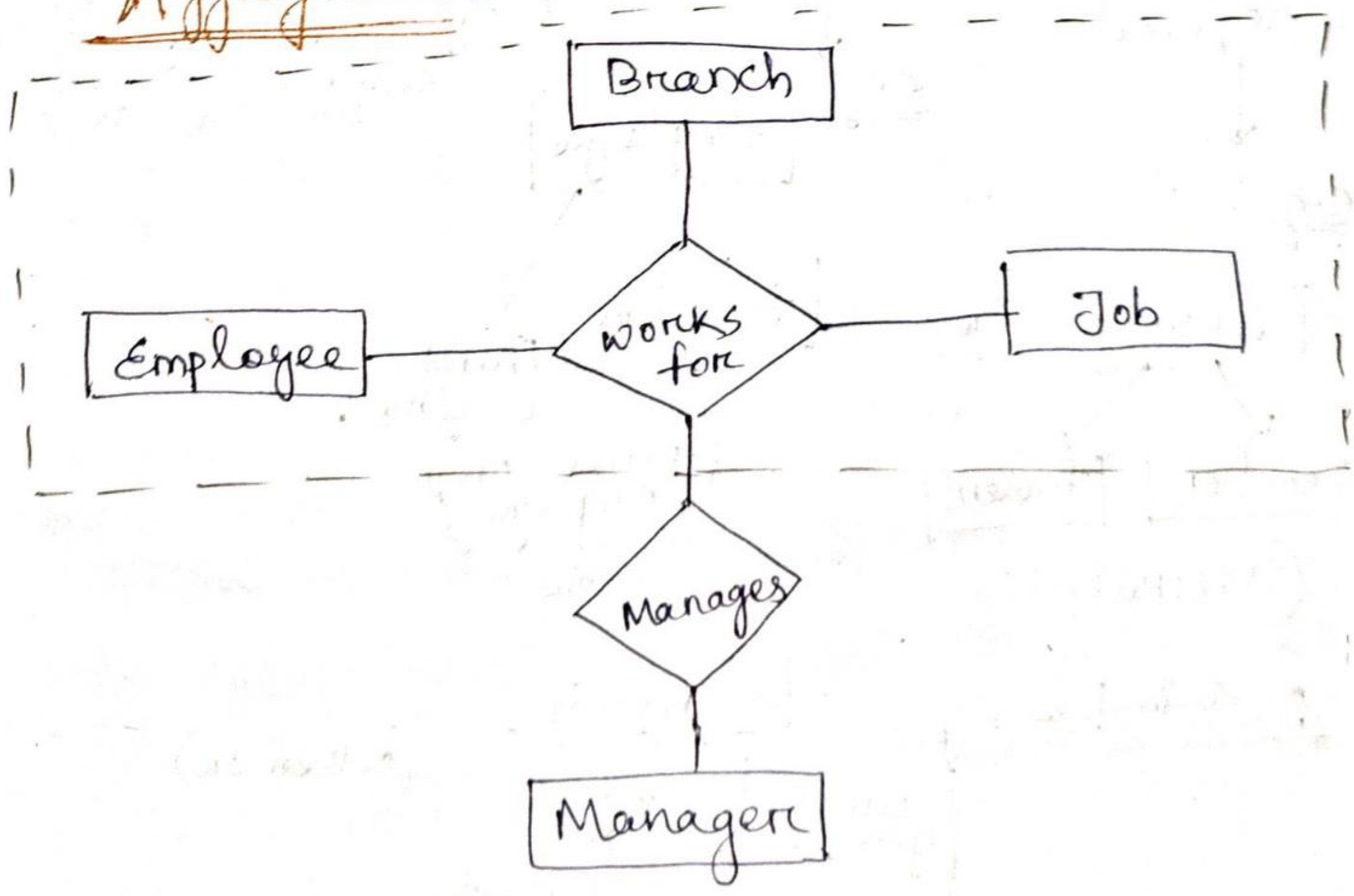
→ An entity is divided into sub entities based on their attributes and properties.

→ Splits a entity into multiple ~~ent~~ new entities.

→ Specialization follows top-down approach.

e.g ②

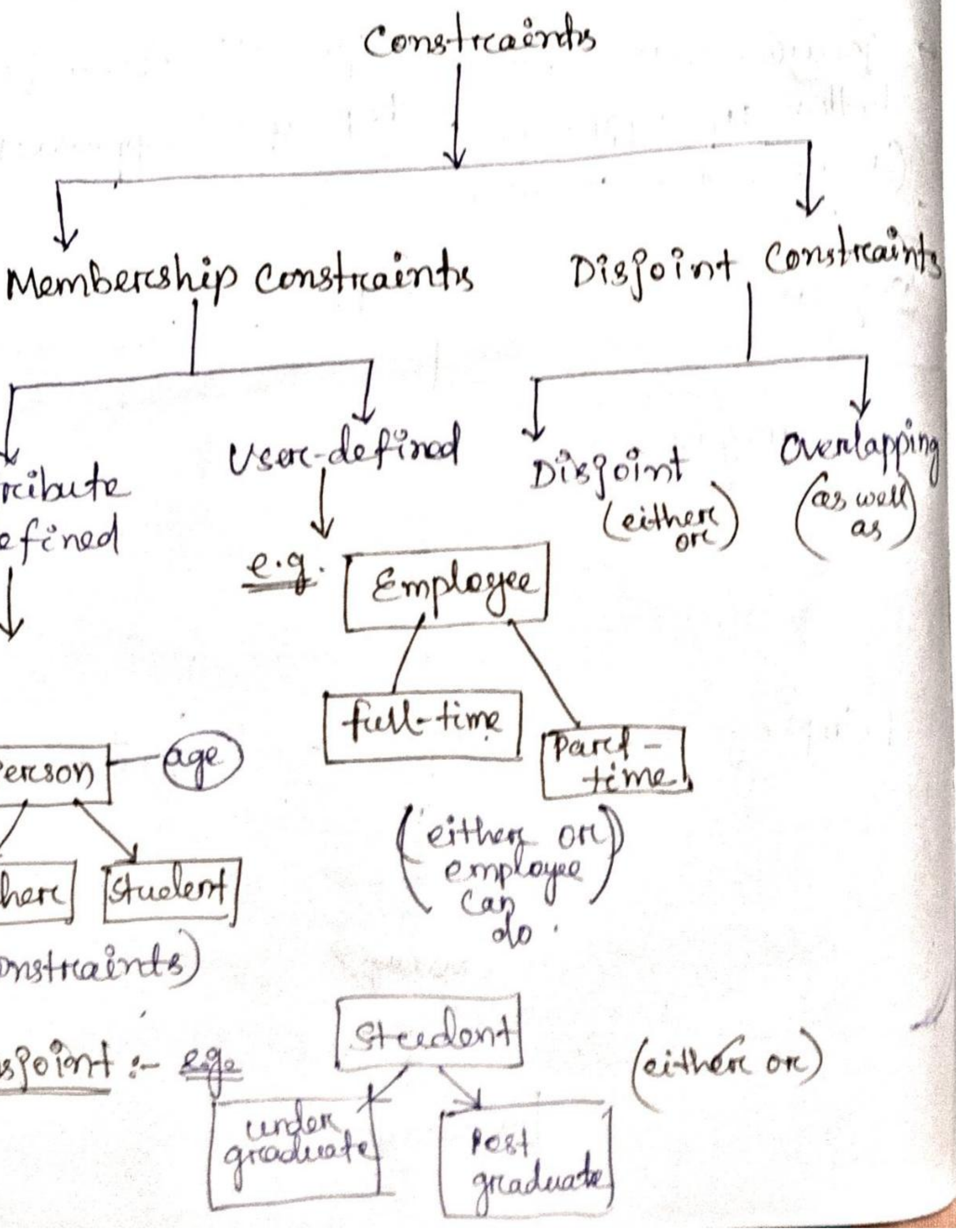
Aggregation :-



→ When we have several relationships then use aggregation.

→ Compiling information on a object there by abstracting higher level objects.

Constraints on Generalization or Specialization



(1) Membership Constraints — Dt- 25.05.22

→ To modern an enterprise more accurately the database designer may choose to place certain constraints on a particular generalized.

→ ... One type of constraints involves determining which entities can be member of a given lower level entity set. Such membership may be one of the following

Attribute-Defined / Condition-Defined :-

→ In condition-defined lower level entity set means student and teacher, membership is evaluated on the basis of whether or not an entity satisfies an explicit condition or predicate or attribute.

User-Defined :-

→ The database user assigned entities to a given entity set.

(2) Disjoint Constraints :-

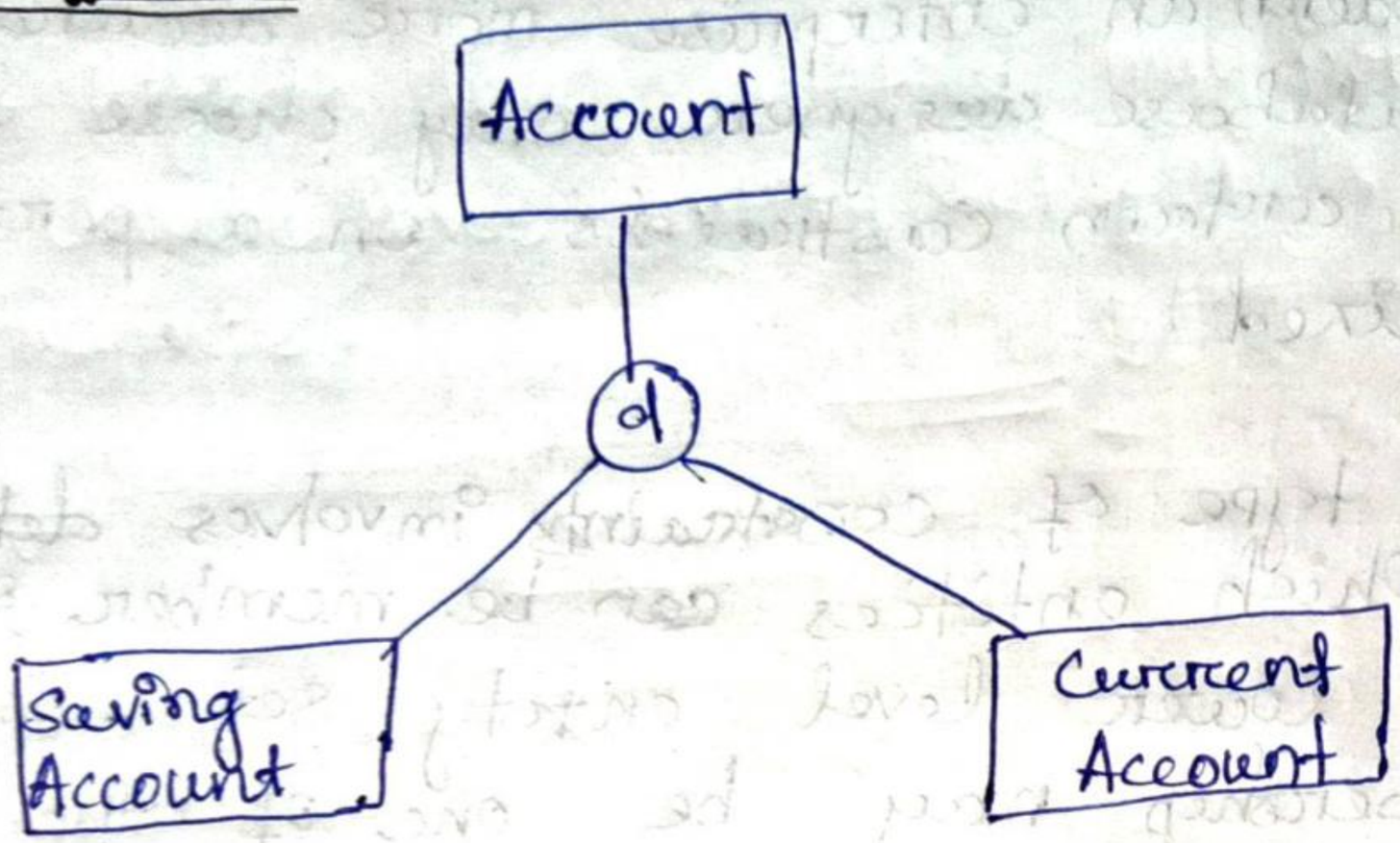
→ A second type of constraints relates to whether or not entities may belong to more than one lower level entity set within a single generalization.

→ The lower level entity set may be one of the following;

(i) Disjoint

(ii) Overlapping

Disjoint :-

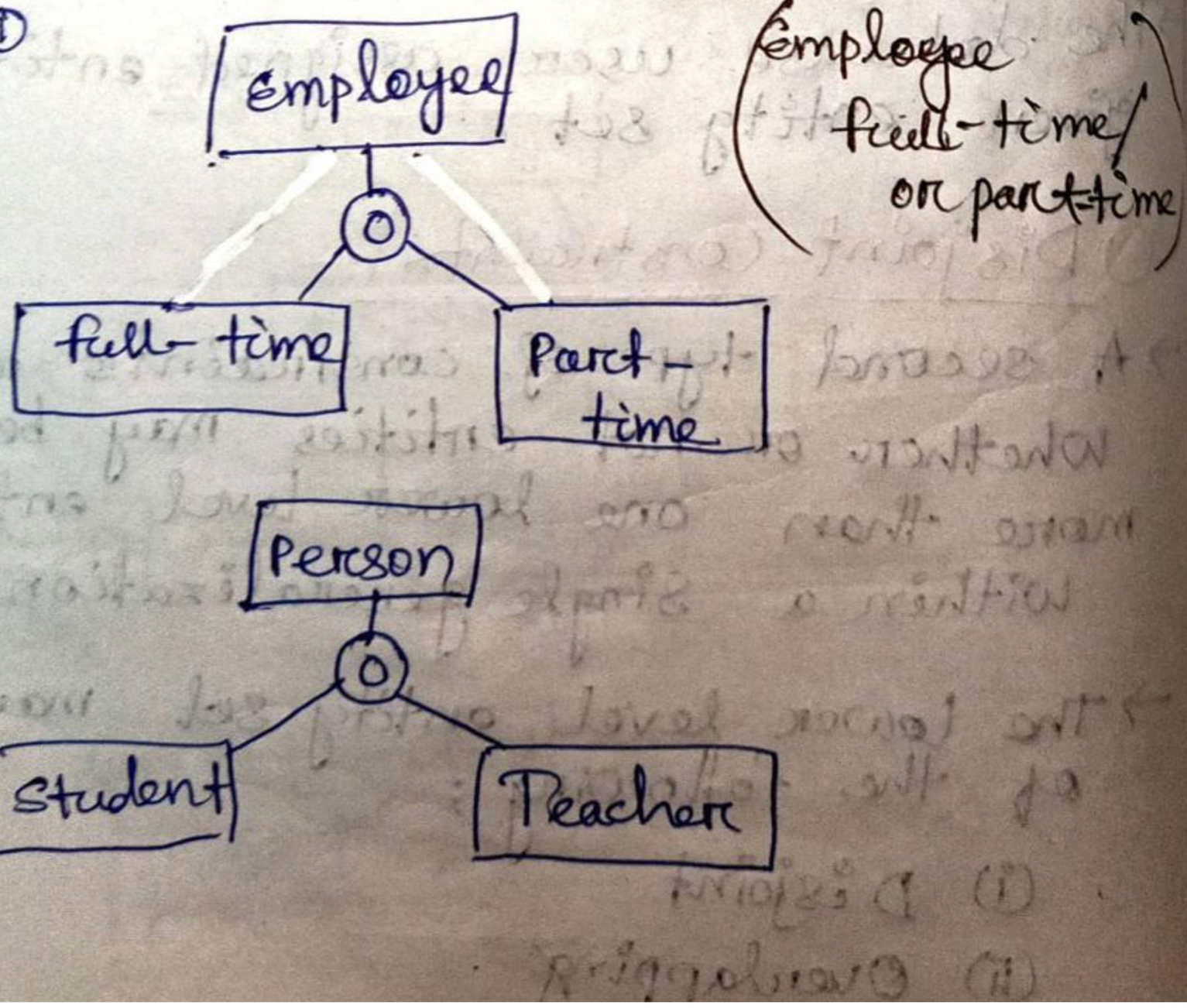


→ An entity belong to no morethan one lower level entity set, that is called

Overlapping :-

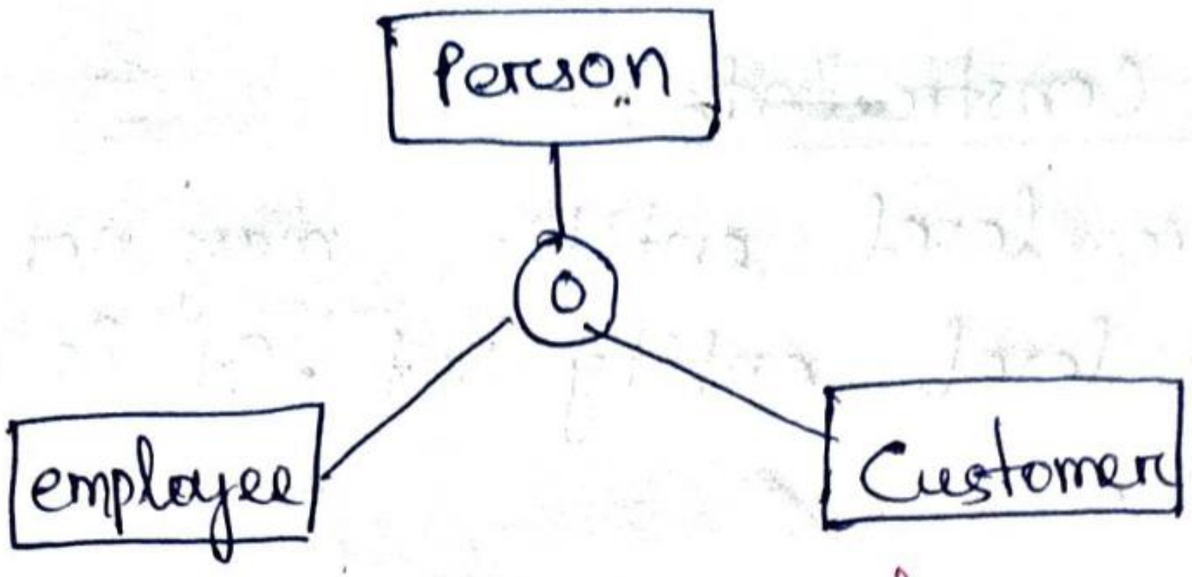
→ The same entity may belong to morethan one lower level entity set within a single generalization.

e.g. ①



②

③



③ Completeness Constraints :-

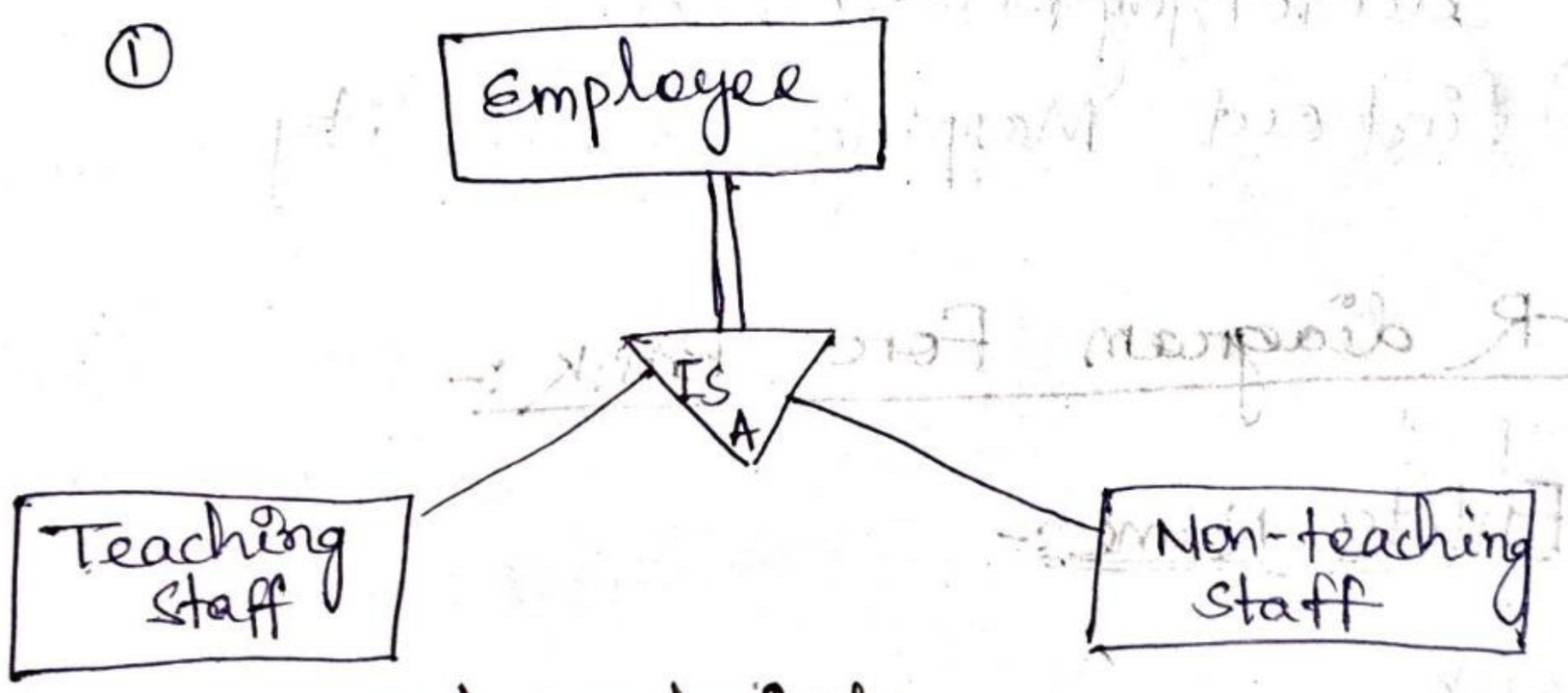
It specifies whether or not an entity in the higher level entity set must belong to at least one of the lower level entity set with in the generalization or specialization.

① Total constraints (=)
② Partial constraints (—)

① Total Constraints On Generalization -

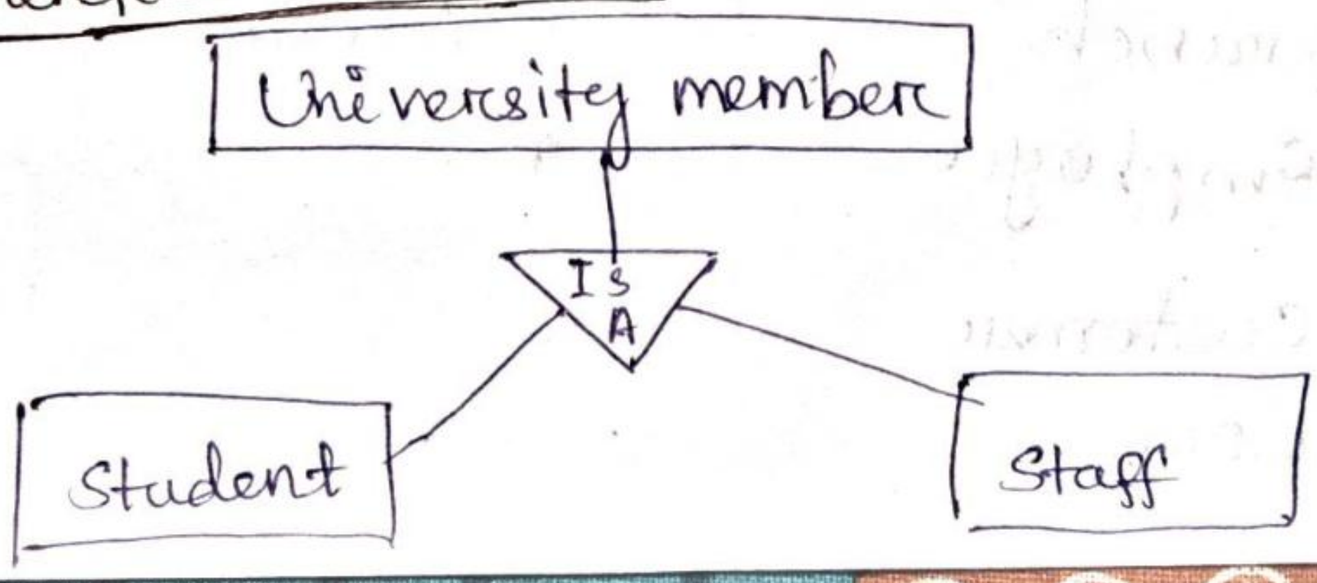
→ Each higher level entity must belong to lower level entity.

eg. ①



e.g. on partial constraints

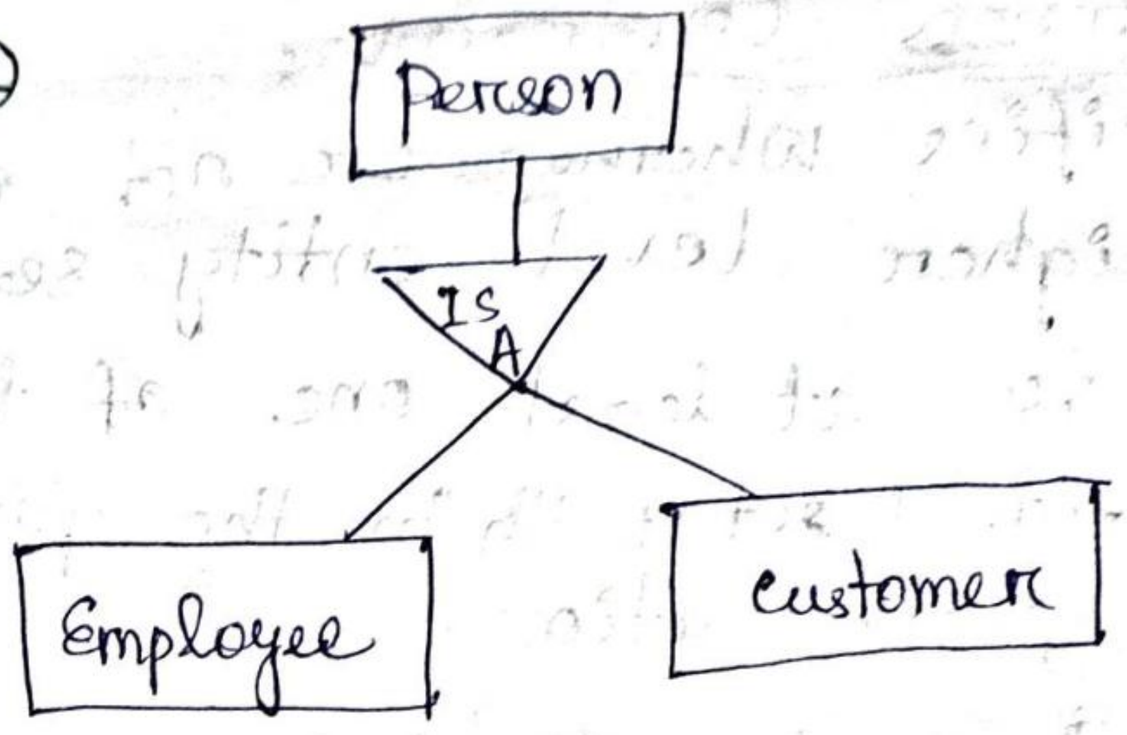
①



Partial Constraints :-

→ Some higher level entities may not belong to lower level entity set. It is default Constraints.

e.g. ②



Steps to Draw E-R :-

① Identify the entity.
② Identify the attribute
③ Identify the key attribute (primary key)
④ Identifying Relationship
⑤ find out Mapping Cardinality.

E-R diagram For Bank :-

Step-1

Entity Name :-

1) Bank
2) Branch
3) Employee
4) Customer
5) Loan
6) Account

Step-2

Attribute

Step-'3'

Primary key

1) Bank: Bname, code
Bank_Location,

code

2) Branch: B-name,
Bank_Location,
B_code
(branch)

B_code
↓
(branch)

3) Employee: E_id
name
Salary
designation

E_id

4) customer: C_id,
C_name,
Address,
DOB

C_id

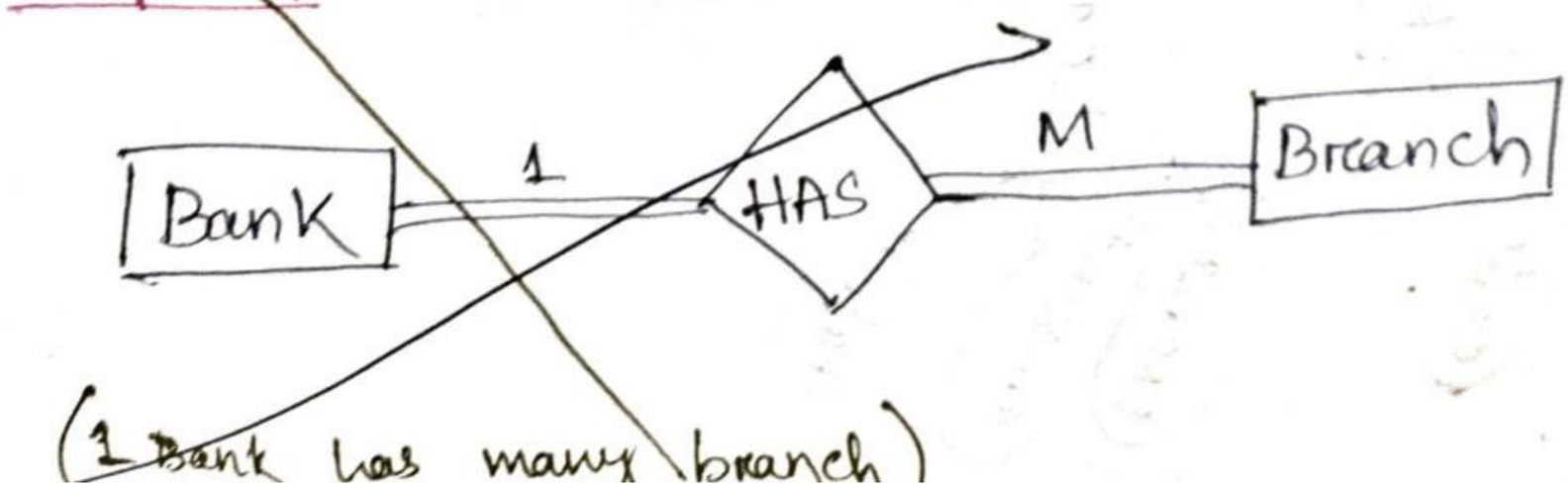
5) Loan: Loan_no,
Amount,
Rate

Loan_no

6) Account: Acc_no,
Acc-type,

Acc_no

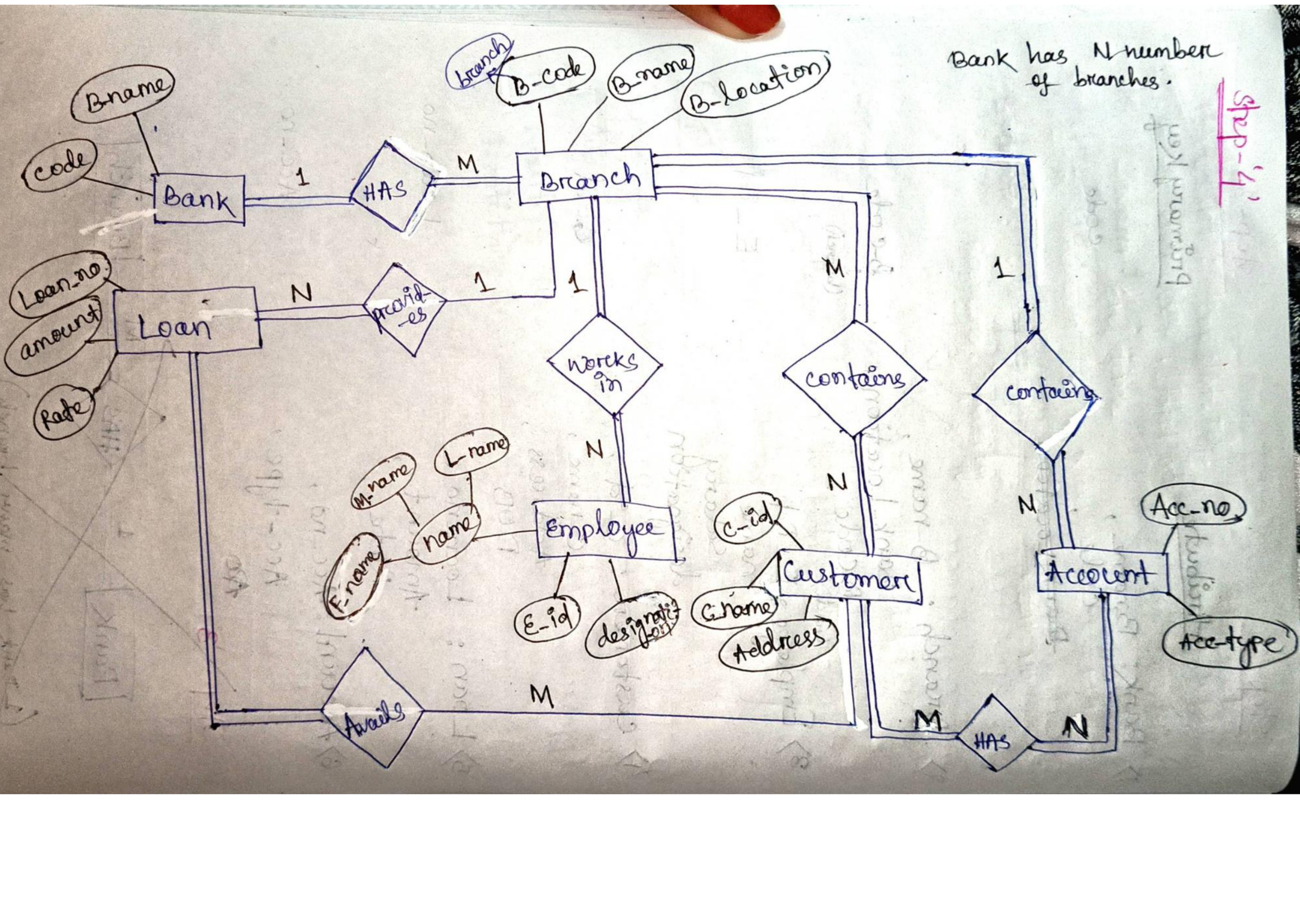
Step-'4'



(1 Bank has many branch)

Bank has N number of branches.

step-4



Dt - 26.05.22

E-R Diagram for University Enterprise

Department (Dept_name, building, budget)
Instructor (id, name, salary)
Course (course-id, title, credits)
Section (sec-id, semester, year)
timeslot (time_slot_id, day, starttime, endtime)
classroom (building, Room_no, capacity)
Student (Id, name)

=

Step-1

1) Department, 2) Instructor, 3) Course
4) Section, 5) timeslot, 6) classroom
7) Student

Step-2

Attribute

Department : Dept_name,
building, budget.

Instructor : id, name, Salary

Course : course_id,
title, credits

Section : sec-id, semester,
Year

Step-3

Primary Key

Dept_name

id

course_id

sec_id

timeslot : timeslot_id, timeslot_id
day, start time,
End time

classroom : building, Room_no
Room_no, Capacity

Student : id, name id

Step-4