28/July/17

Data :-

* Data is any information.
* In software, we manage the data. Manage data means storing, processing and extracting data.
* Data persistence means existence of data, we need to analysis that for how long we want data to persist in the memory.

Simply, we can say data persistence is "life of data". Sometimes, we need data even beyond the life of the program.

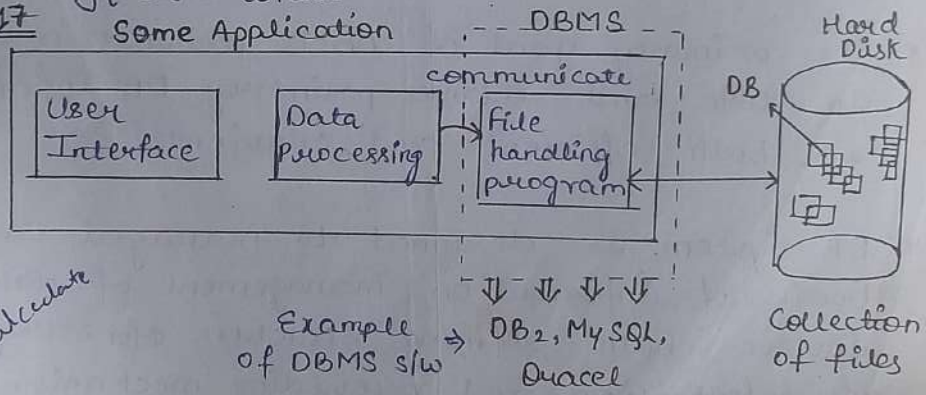
* Data is sometimes, needed to remain available even beyond the life of the program in such a case, data is stored in secondary storage device like a Hard Disk.

* When data is stored in secondary storage device is bound to reside in a file.

29/July/17

110

It is a collection



fine calculate

Example of DBMS s/w ⇒ DB2, MySQL, Oracel

Collection of files

* A database is a place where all your application related data is stored, one application data can be stored in a bunch of file. we can say that DB is a collection of file.

DataBase:-

* DataBase Management System is a collection of interrelated data and set of programs access those data.

* Database is the collection of stored operational data used by application system of some particular enterprise.

* Database + DBMS = Database System

* DBMS is an intermediate layer between programs and data.

* DBMS is a collection of programs that enables user to create and maintain the DB.

* The primary goal of DBMS provides a way to store and retrive primary DB information i.e., both efficient and convient

* DB System is designed to managed large bodies of information, management of data involve both defining structure for storage of information and providing mechanism for

different

manipulation of data.

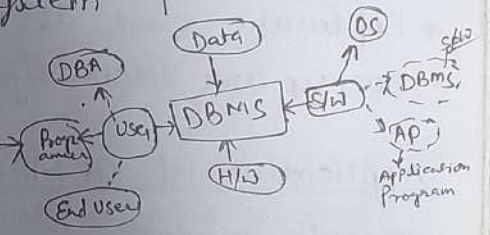
* Note:- It is nothing more than computer based record system, i.e., a system whose overall purpose is to stored and maintain information and it may be necessary to the decision making process involved in the management of the organisation.

Types of DB:-

1) Small size DB
   Personal mailing list
2) Middle size DB
   Hospital management system
   Library management system
3) Large size DB
   Banking system
   Railway reservation system

31/July/17

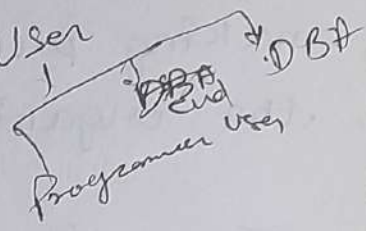
Component of DBMS:-



1. Data
2. Hardware :- secondary storage devices
3. Software :- collection of program
4. User :- we consider 3 class of user -
   * there is an application programmer responsible for writing application program for actual interrelated data

that used in DB.

* Second class of user is end user, access the DB from the terminal.
* The third class of user DBA (DataBase Administrator) which control and manage the DB.



Advantage:-

* Inconsistency can be avoided.
* Redundancy can be reduced.
* Integrity can be maintained.
* Security mechanism can be applied.
* Data can be shared.
* Data persistence.

Disadvantage of DBMS:-

* Hardware cost is very high.
* Software cost is very high.
* Backup and recovery cost is high.

Application of DBMS:-

* Banking System
* Railway reservation System
* Library management system
* Airlines reservation system
* Telecommunication management system
* Hotel management system

differentiate

* Student registration mang. system
* Human Resource
* Traffic control system
* Hostipal management system

DataBase Administorator:-

DBA is a person or group of person responsible for overall control of the DBMS. It means that it control both data and application program.

The DBA responsibilities includes:-

* Decisiding the inf. content of DB.
* Decisiding the storage structure of DB.
* Defining the stragies of backup and recovery process.
* Defining the authorisation check and validation process.

01/Aug/17

View of the Data //

* A DB system is a collection of interrelated files and set of programs that allows users to access and modify these files.
* A major purpose of DBMS is to provide users with abstract view of the data, i.e., the system hide certain details to yhow the data are stored and maintained.

Data Abstraction:-

The property of DBMS that allow program data independence is known as "Data Abstration".

They are used for following purpose:-

* To provide abstract view of the data.
* To hide complexity from user.
* To simplify user interaction with DBMS.

Three level of Data abstraction

1. Physical Level:

* The lowest level of abstraction describes 'How data are actually stored'.
* The physical level describe the complex low level data in details.
* At this level efficient algorithm to access data are defined.

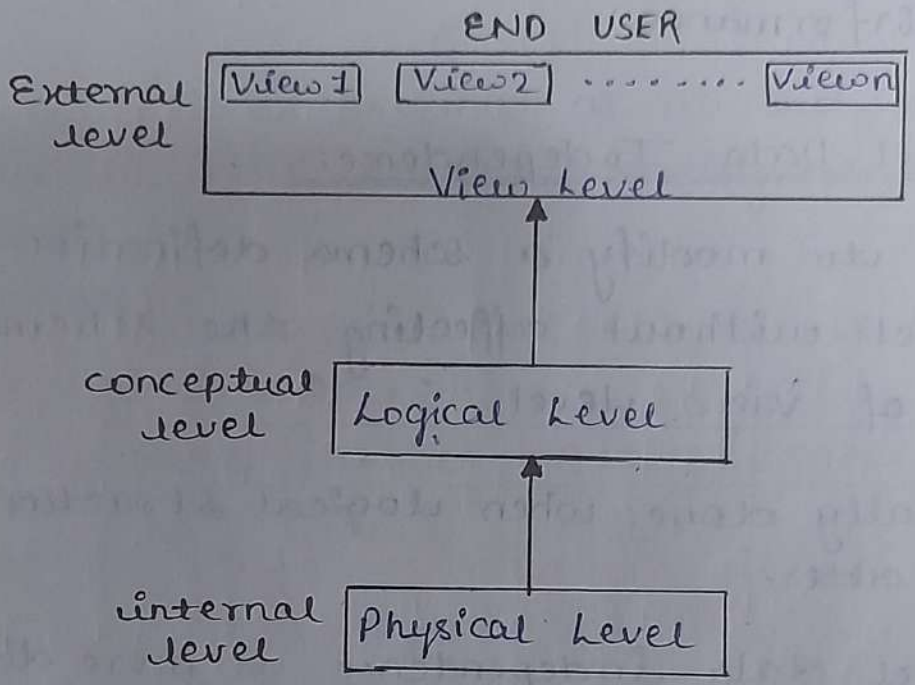
2. Logical or Conceptual Level:

* In this level of abstraction describes 'what type of data are stored in the DB and what is the relationship among those data'.
* DBA, who must decide 'what inf. to keep in the DB' use logical level of abstration.

differentiate

3. View or External Level:

* The highest level of abstraction describes only 'part of entire DB'.
* View level exist to simplify their interaction with the system.
* The system may provide many view for the same DB.



03/Aug/17

Data Independence

The ability to modify the schema defination in one level without affecting the schema defination in next higher level is called data independence.

There are two types of data independence in DBMS :-

1. Physical ~~do~~ level data independence
2. Logical level data independence.

Physical Level Data Independence :-

* The ability to modify the schema defination of physical level without affecting schema defination of logical level is known as Physical level data independence.

* The modification of this level are usually to improve performance.

Logical Level Data Independence :-

* The ability to modify a schema defination in logical level without affecting the schema defination of view level.

* It is usually done when logical structure of DB is alter.

* Logical level data independence is more difficult to achieve than the physical data independence because the application program are always dependented on logical structure of DB.

* Data independence is important characteristic of DBMS as it allow the changing of the structure of the DB without making any changes in the application program that use the DB.

Schema and Instances:

* Database change over time as information is inserted and deleted, the collection of information stored in the DB at a particular moment is called "instances".

* The overall design of the DB is called "schema of DB".

* A schema is defined as an outline or plan that describe the record existing at a particular level.

Example:-

Stu_info

Name	Branch	Address	Mobile No.
X	CS	LKO	9307

* Schema is called as the 'intention of DB' while instance is called as the 'extention of DB'.

04/Aug/17

DataBase Language

* DBMS provides two types language, one to specify the DB scheme and other express query and update.

* These language are used to create and maintain DB on the computer system. These types of

language are called "Structured Query Language" (SQL). {VDL, SDL}

Data Defination Language:

* DDL is used for defining the DB scheme.
* A DB scheme is specified by set of defination expressed by a set of language is called "Data defination language".
* It is language that allow the user to define data and their relationship do other type of data.
* DDL is a set of SQL commands used to create, modify and delete the DB structure but not data.
* These command are normally not used by general user, they are used by DBA, designer and developer of the DB.
* The DDL statement is used to specify the integrity rule.

DDL Command:

1. Create : create the database.
2. Alter: it is the structure of the database
3. Drop: delete object (column) from the database

4. Rename.

For example:

⇒ Create database SIET;
Use SIET;

⇒ Create table Student_record (Stu_id int,
name varchar(20),
branch char(15),
address varchar(50),

Data Manipulation Language:

* The DBMS provides DML that enables user to retrive and manipulate the data, the statement that is used to retreive the information is called "Query Language".

* It is defined as a language that provides a set of operation to support the basic data manipulation operation on the data held in the DB. These language can be used for —

1. The retreival of information stored in the DB.
2. Insert the new information into the DB.
3. Deletion of information from the DB.
4. The modification of information stored in the DB.

Types of DML:

There are 2 types of DML:-

1. Procedural DML
2. Non-procedural DML

Procedural DML:-

It requires user to specify whata data are needed and how to get that data.

For example:

Relational Algebra

Non-procedural DML:-

It requires user to specify what data are needed without specifying how to get those data.

For example:

SQL

DML Command:

1. Insert
2. Delete
3. Update
4. Select

1. Create table Stu_record (id int, name char(20), address varchar(50));

2. insert into Stu_record into (101, 'Rahul', '135/113 Teliyarganj, Alld.');

3. Select * from Stu_record;

Database Interface: An Interface is a program that allocos users to query the DBMS without writing the code in query language. * An interface can be used to maintain manipulate the DB either for adding the data or deleting some data or updating some data for viewing the data present in the DB.

There are various types of DB interface:

1. Stand-alone query language interface.
2. Programming language interface.
3. User interface.
   * Menu based interface
   * Form based interface
   * Graphical user interface
   * Interface for parametric user
   * Interface for DBA

HomeWork

Difference b/w Data File System and DBMS:

* Database and File System are two methods used to store, retrieve, manage and manipulate data. Both systems can be used to allow the user to work with data in a similar way.

* A file system is a collection of raw data

files stored in the hard-drive, whereas a database is intended for easily organizing, storing and retrieving large amounts of data.

* File System are used to store data while, a database is a collection of organized data.

* A file system will lead to problems like data integrity, data inconsistency and data security, but these problems could be avoided by using a database.

* A file based system is a collection of data stored in an orderly manner in a file. It is a file packed with data, with no metadata and thus no organizing structure. A DB is a self-organizing collection of integrated records, whose metadata gives it structure.

Differentiate

Explain Data Model.

* Data models are fundamental entities to introduce abstraction in DBMS.
* Data models define how data is connected to each other and how they are processed and stored inside the system.
* A data model explicitly determines the structure of data. Data models are specified in a data modeling notation, which is often graphical to form.
* The main aim of data models is to support the development of information systems by providing the definition and format of data.

Types :-

*. Flat model

The flat (or table) model consists of a single, 2-D array of data elements, where all members of given column are assumed to be similar values, and all members of a row are assumed to be related to one another.

* Entity - Relational model

ER model is based on the notion of real-world entities and relationships among them.

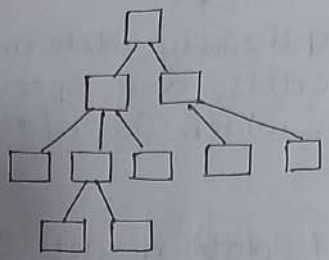
ER mode based on -

⇒ Entities and their attributes : real-world entity having properties called attributes.

⇒ Relationship among entities : logical association.

2. Hierarchical model

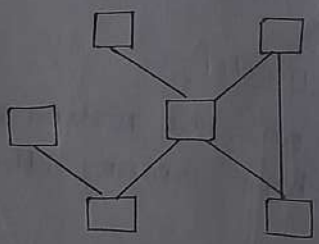
Hierarchical model is similar to the network model except that links in the hierarchical model form a tree structure, while the network model allows arbitary graph.



3. Network model

This model organizes data using two fundamental constructs, called records and sets. Records contain fields, and sets defines one-to-many relationship between records : one owner, many members.

The network model is an abstraction of the design concept used in the implementation of databases.



differentiat...

4. Relational model

Relational model is a DB model based on first-order predicate logic. Its core idea is to describe a DB as a collection of predicates over a finite set of predicate variables, describing constraints on the possible values and combinations of values.

The power of relational data model lies in its mathematical foundations and a simple user-level paradigm.

5. Object-Relational model

Similar to relational DB model, but objects, classes and inheritance are directly supported in DB schemas and in the query language.

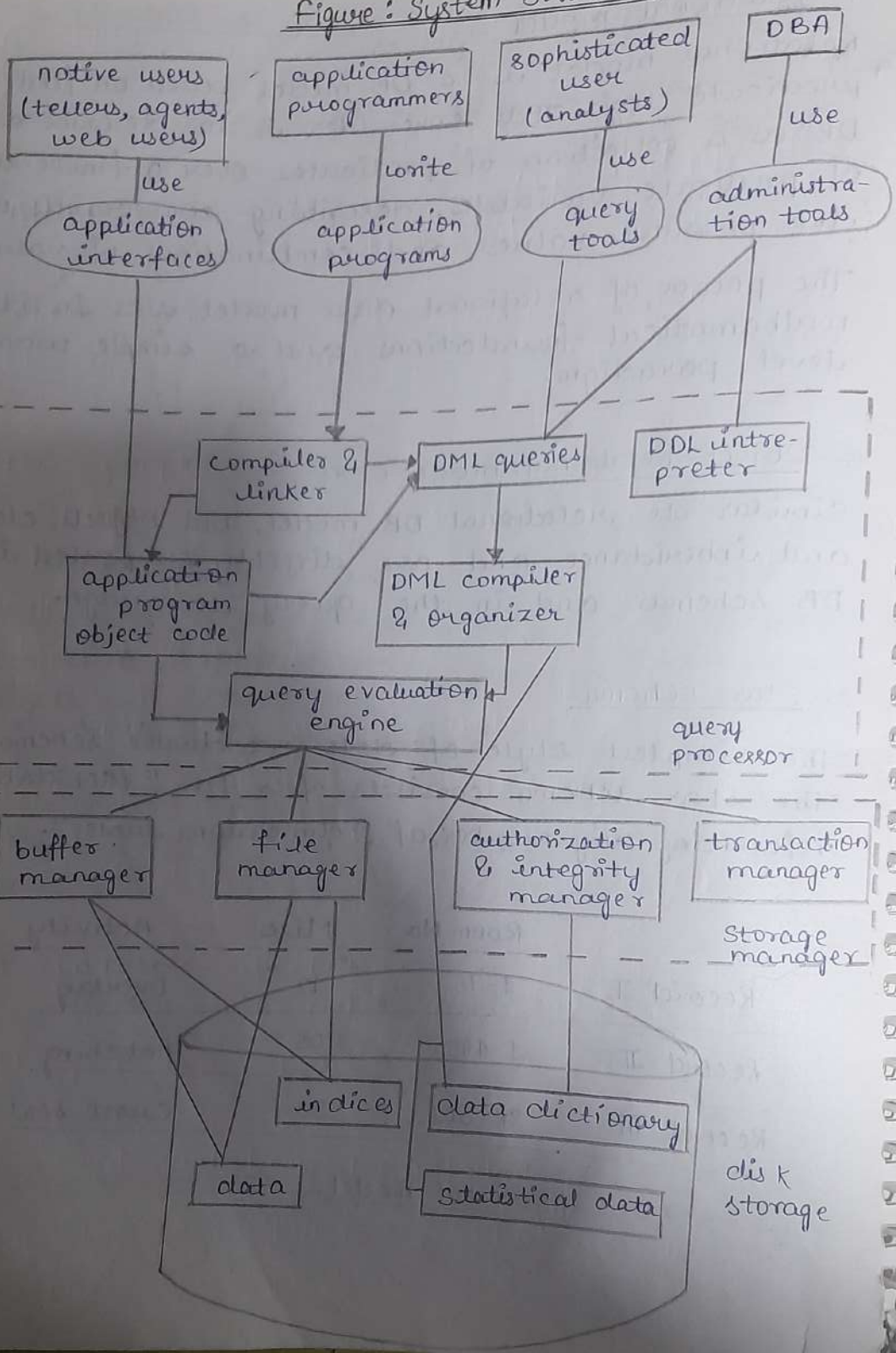
6. Star schema

The simplest style of data warehouse schema. The star schema consists of a few "fact tables" referencing any number of "dimensions tables".

	Room No.	Miles	Activity
Record I	1-95	12	Overlay
Record II	1-495	05	Patching
Record III	SR-301	33	Crack seal

fig: Flat model

Figure : System Structure



10/Aug/17

Entity Relationship Diagram.

* The overall logical schema of DB can be expressed graphical by ER Diagram.
* ER model can be used by DB designer to communicate the design to end user.
* An ER model is defined as conceptual data model that view the real would as entities and relationship
* Visual representation of DB.
* ER diagram describes relationship b/w tables.

Entity: * An entity is a things or object in the r/l world that distinguishable from other object

* An entity is an object that exist in the real world and it may be 'student', 'faculty', 'place', 'object'. * An entity has a set of properties, and values for some set of properties that may uniquely identify an entity.

for example: Each worker working in the company is the entity and the department or project in which worker is working again an entity, all worker working in the company such a group is called "Entity Set".

* An entity set is a set of entities of the same type that share the same property of attributes.

* An entity is a concept or object, entity can sometimes represents the relationship between two or more object, this type of entity is called "association entity".

Relationship:

* Relationship is a meaningful association b/w two or more entities.

* A relnship involve multiple entity set. The number of entity involved in relationship is called "degree of relationship."

Attributes: entity is represented by set of attributes. For each attributes, there is an set of value are called domain. Each entity has attributes, they are properties whose value are data that stored in the DB. Such that entity are described in database by set of attributes.

For example:

Student is entity and Student name, age, branch, phone no., address are the attributes.

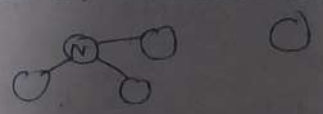
11/Aug/17

Types of Attributes

* Simple and Composite Attributes :-

→ A simple attribute can not be further sub-divided.

For eg: Student_Id, First_name, middle_name, last_name, etc.



⇒ A composite attribute is an attribute that can be divided into other attributes.

For eg: Address, name

* Stored and Derived Attributes :-

⇒ A stored attributes is which does not be require any updation.

of emp -

For eg: Date_of_birth

⇒ The value of derived attribute can be derived by other attribute.

For eg: (Age) (Age)

* Single valued and Multi-valued Attribute :-

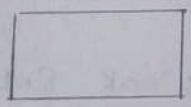
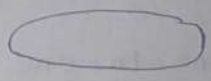
⇒ It is defined as an attribute that hold a single value for a single entity. It means which takes one value per entity

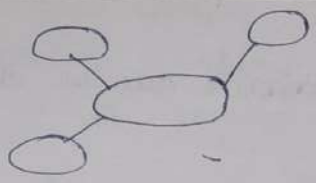
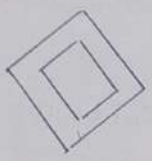
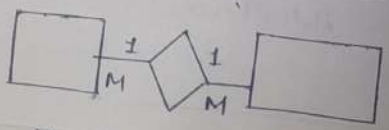
For eg: Gender, Age

⇒ It is defined as an attribute that hold multiple value for a single entity.

For eg: Student_address , phone_no.

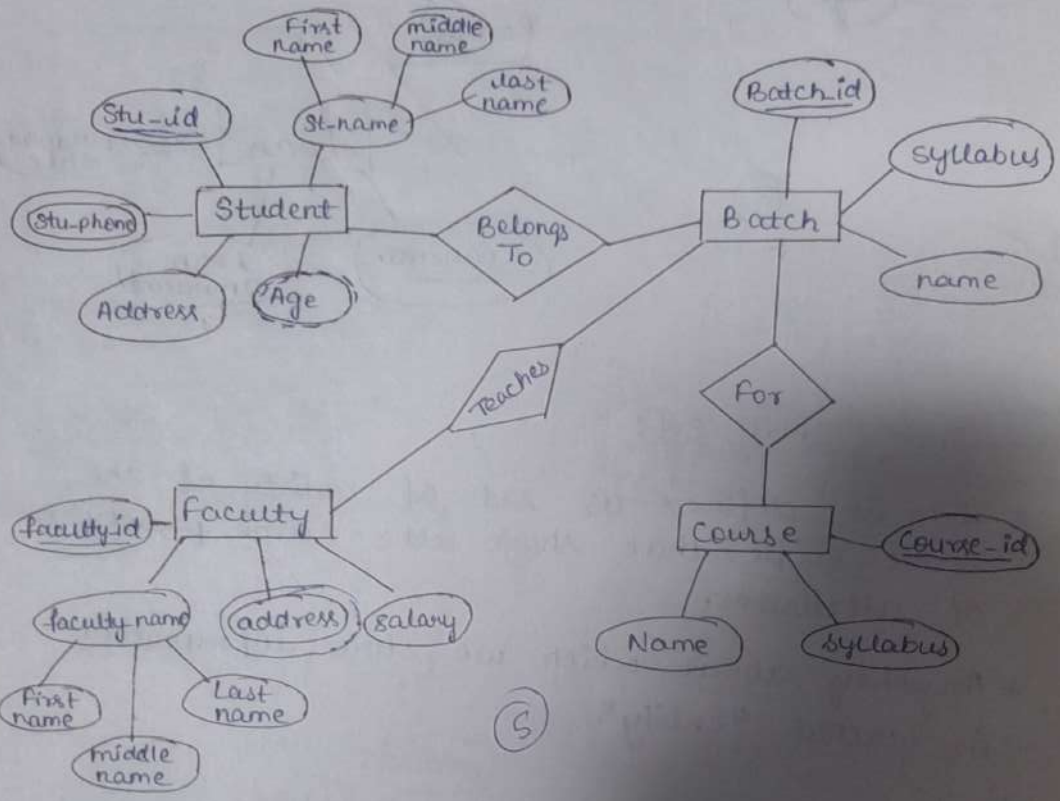
Notation of E-R Diagram

Symbol	Definition
	Entity
	Relationship
	Attribute

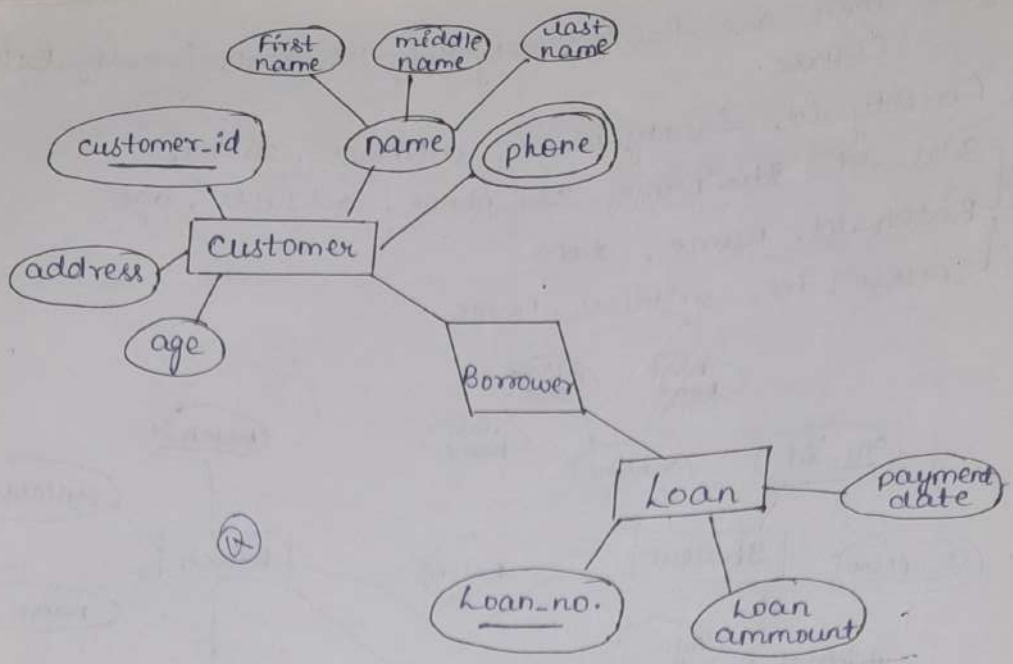
	Composite Attribute
	Multi-valued Attribute
	Key attribute
	Derived attribute
	Identifying relationship
	Weak Entity Set
" "	Link
	One-to-one one-to-many many-to-many

Ques: There are four Entity Set Student, faculty, Batch, Course.

R Faulty_id, faulty-name, address, salary
- R: Stu_id, Stu-name, Stu-phone, address, age
- R: Batch-id, name, date
- R: Course_id, syllabus, name



Ques: Draw the ER diagram which contain Customer and Loan where Customer_id is a primary key, customer_name, phone, address and age And In Loan, loan_no. is a primary key, loan ammount, payment date.



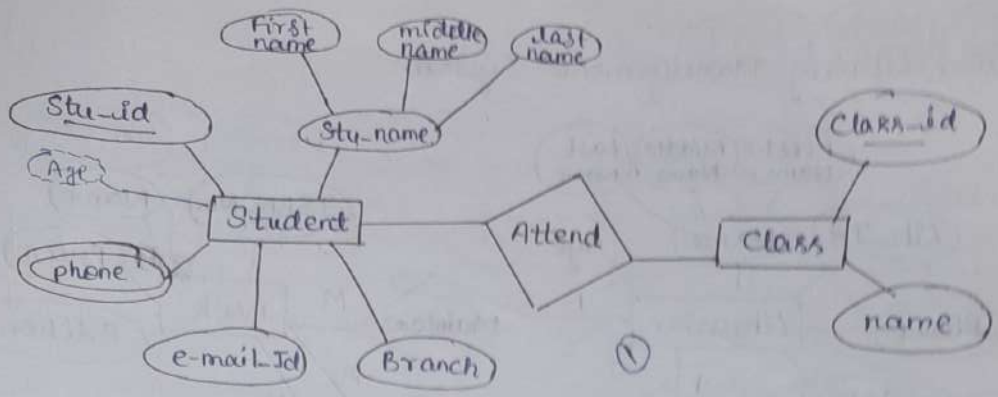
Entity/ Entity Set:

* It is defined as set of entity of the same type that share the same properties of attributes.
* Anything about which we store information is called "entity".

12/Aug/17

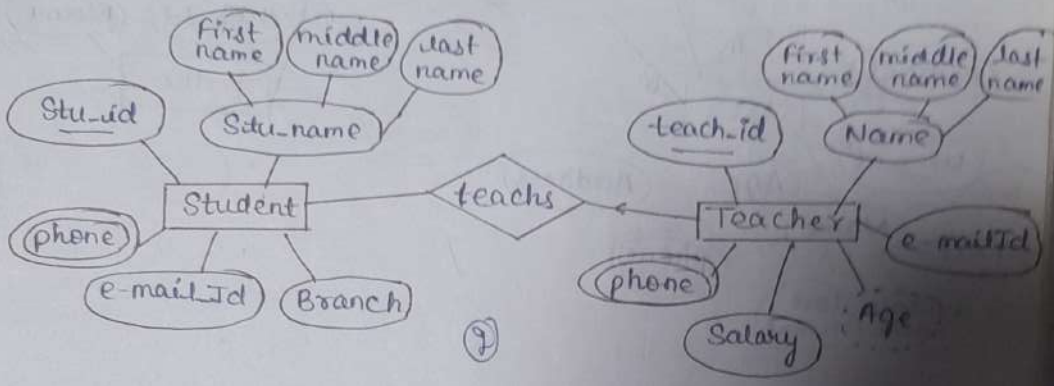
Ques: Student ⇒ Id, name, phone, e-mail_id, branch, ~~subject~~.

Class ⇒ Class_id, name,



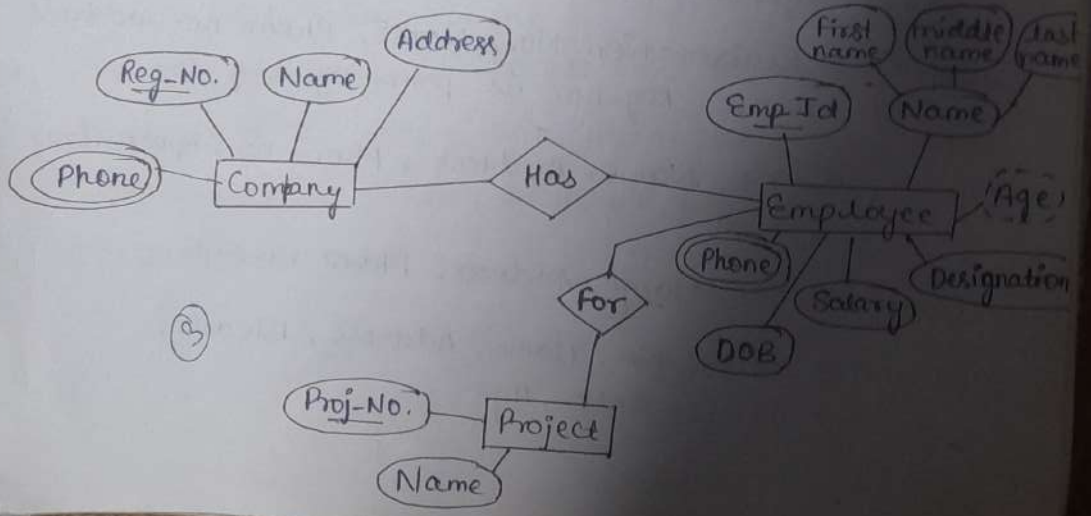
①

Ques: Student and Teacher



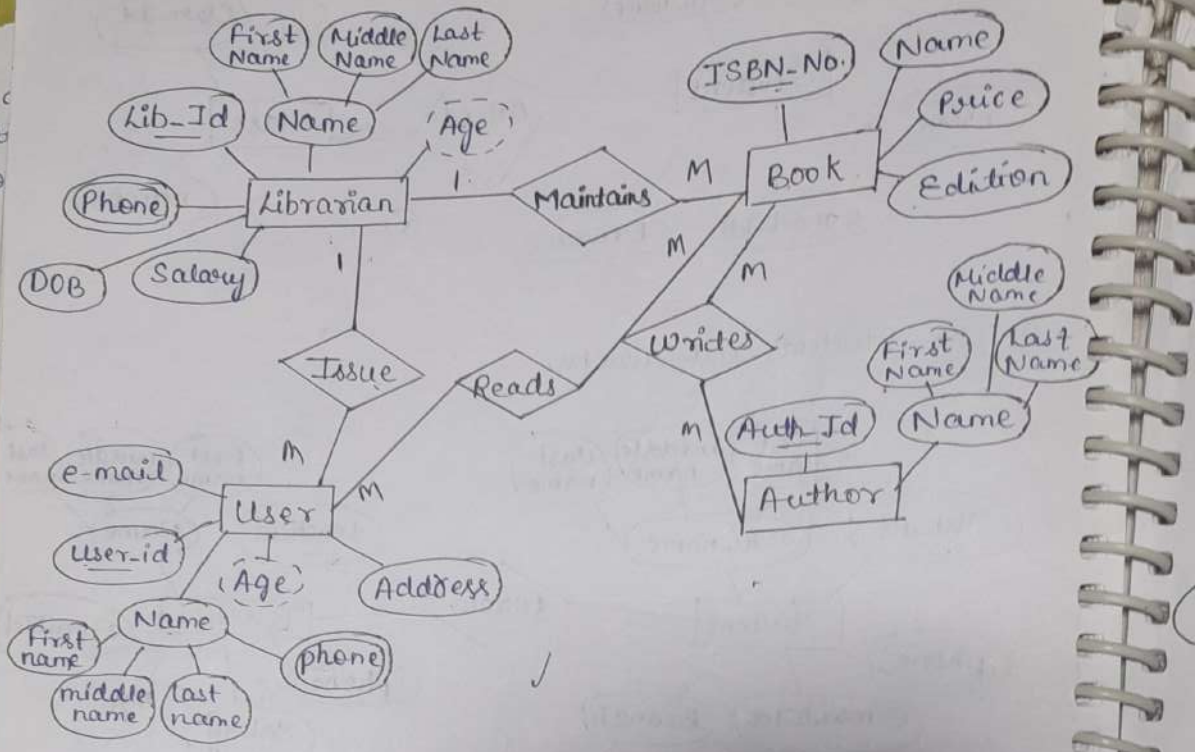
②

Ques: Company; Employee and Project



③

Ques: Library Management System



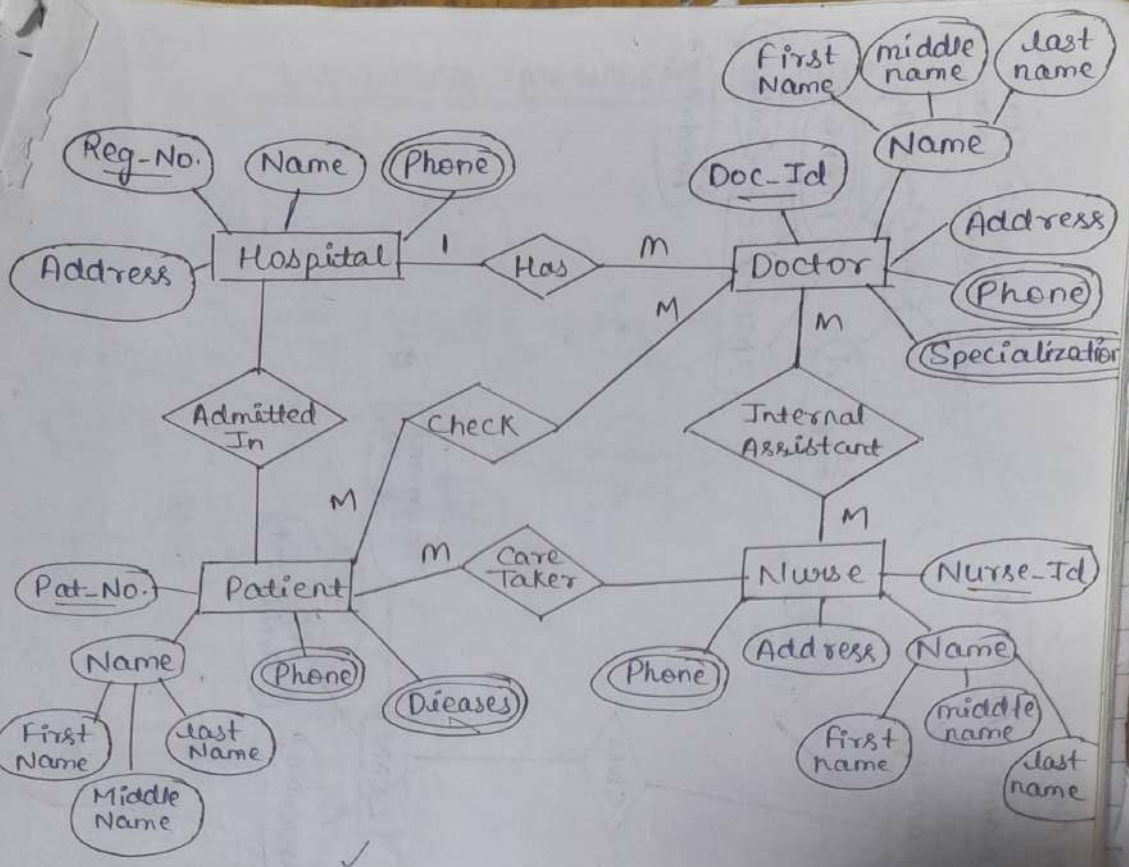
Ques: Draw the E-R diagram for Hospital Management System. Entities are as -

Hospital ⇒ Registration No., Name, Phone no., address where Reg_no. is primary key.

Doctor ⇒ Id, Name, Address, Phone no., Specialization

Nurse ⇒ Id, Name, Address, Phone no., sp

Patient ⇒ Patient No., Name, Address, Dieases, Admit_date, Age



Ques: Bank, Branch, Employee, Customer, Loan and Account

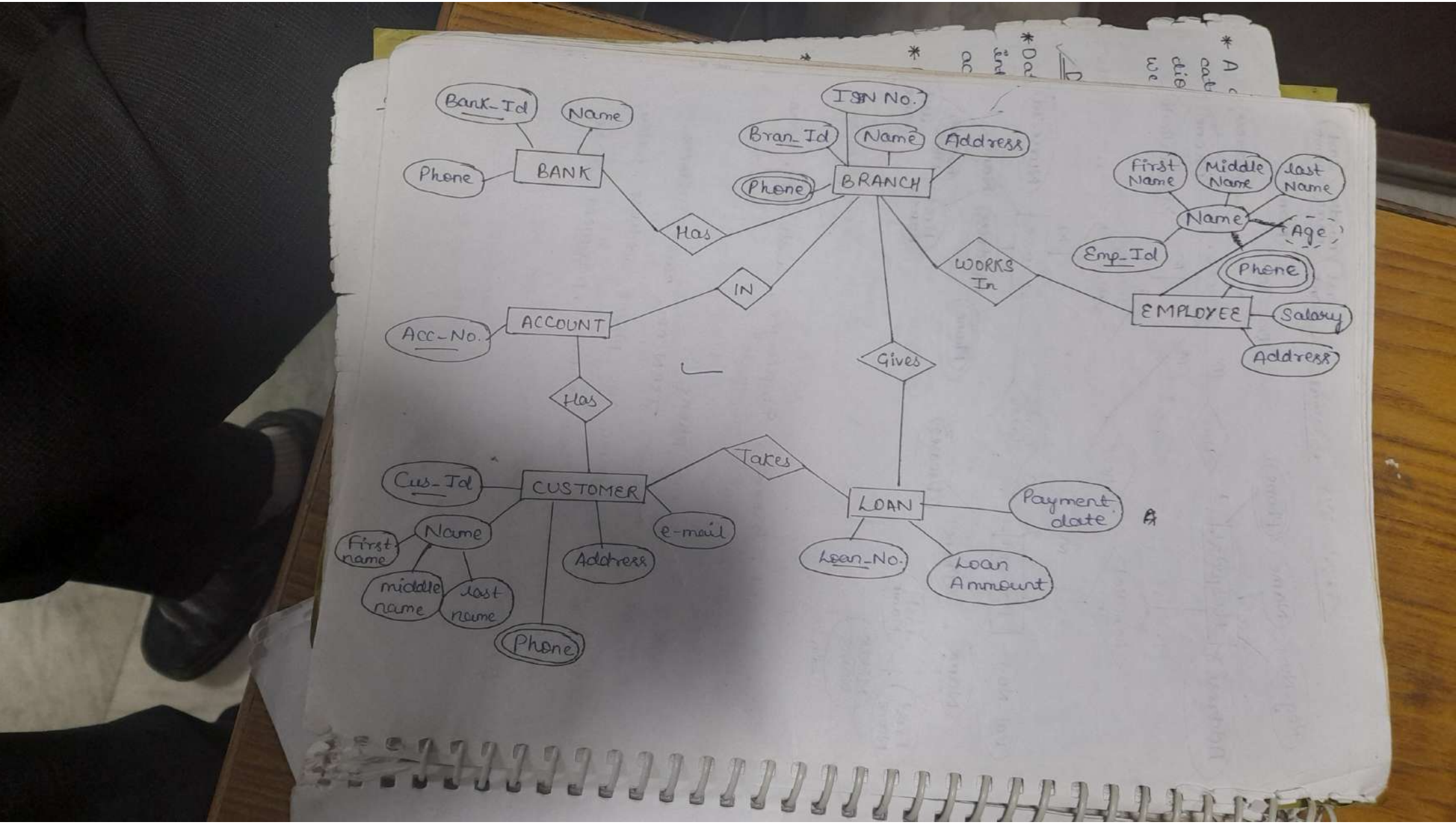
Bank ⇒ Id, name, phone

Branch ⇒ phone, Id, ISBN no., name; address

Employee ⇒ Id, name, age, phone, address, salary

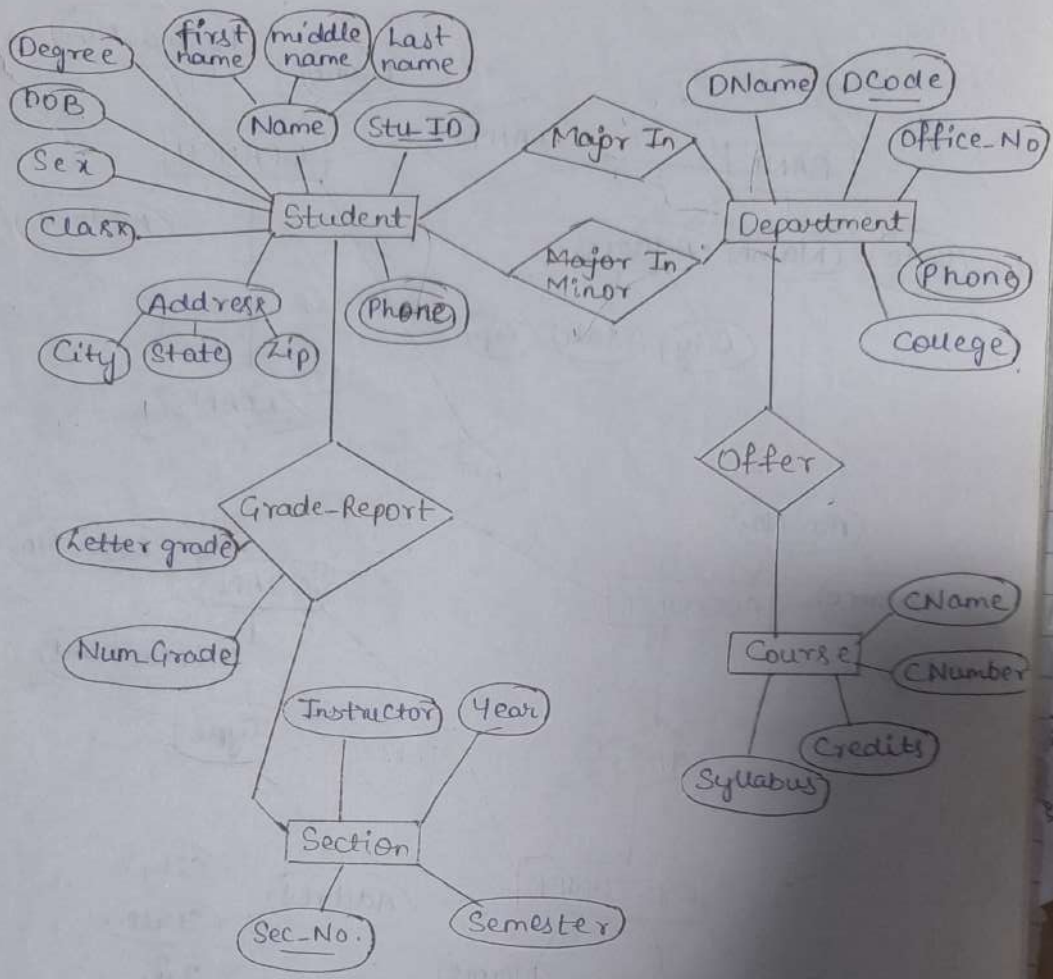
Loan ⇒ Loan_no., Loan ammount, payment date

Account ⇒ Account no.,

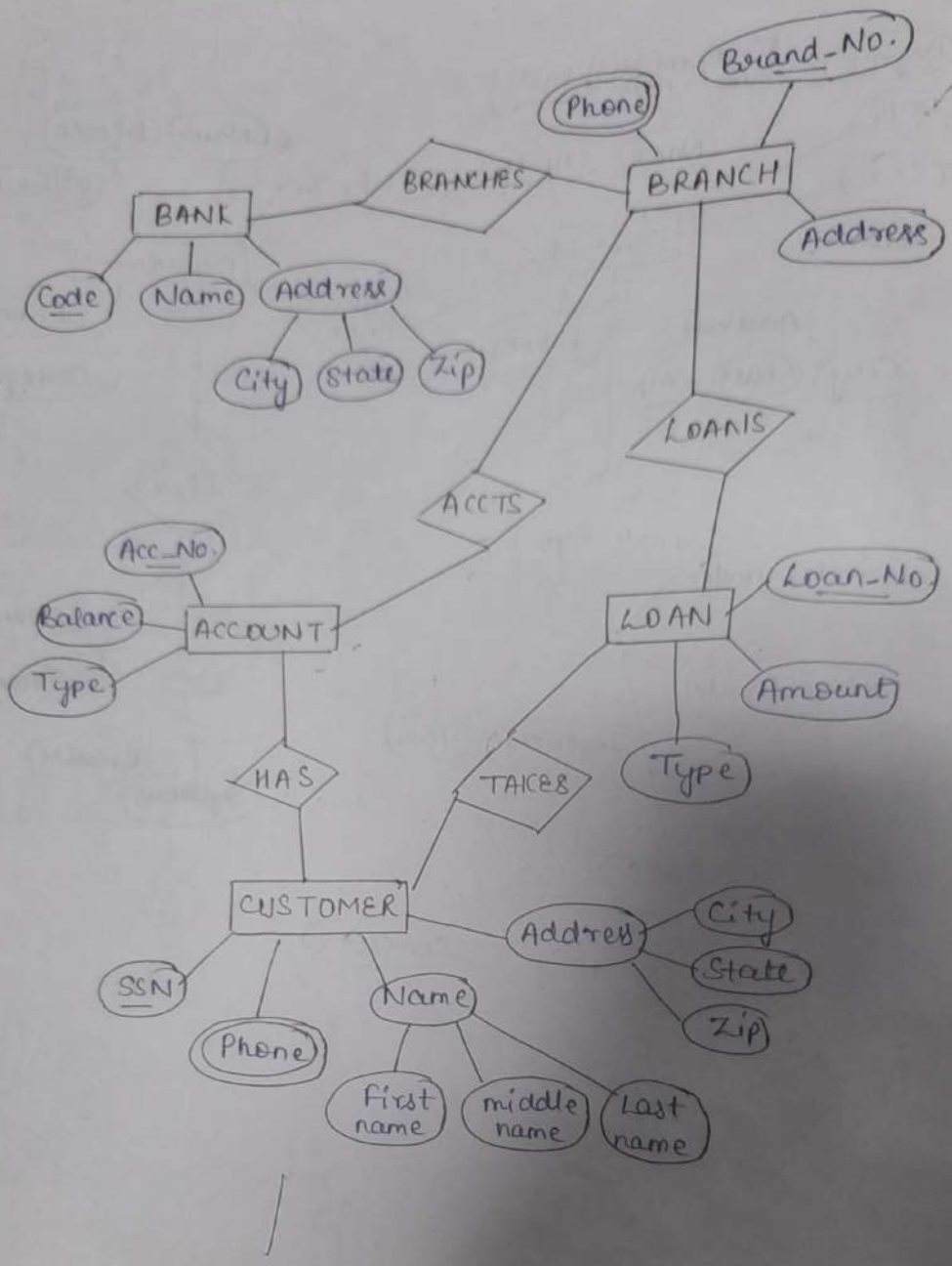

Bank_Id
Name
Phone
BANK
Has
ISN No.
Bran_Id
Name
Address
Phone
BRANCH
First Name
Middle Name
last Name
Name
Age
Emp_Id
Phone
EMPLOYEE
Salary
Address
WORKS In
IN
ACCOUNT
Acc-No.
Has
Gives
Takes
Cus_Id
CUSTOMER
e-mail
Address
Phone
Name
First name
middle name
last name
LOAN
Payment date
Loan_No.
Loan Ammount

Homework

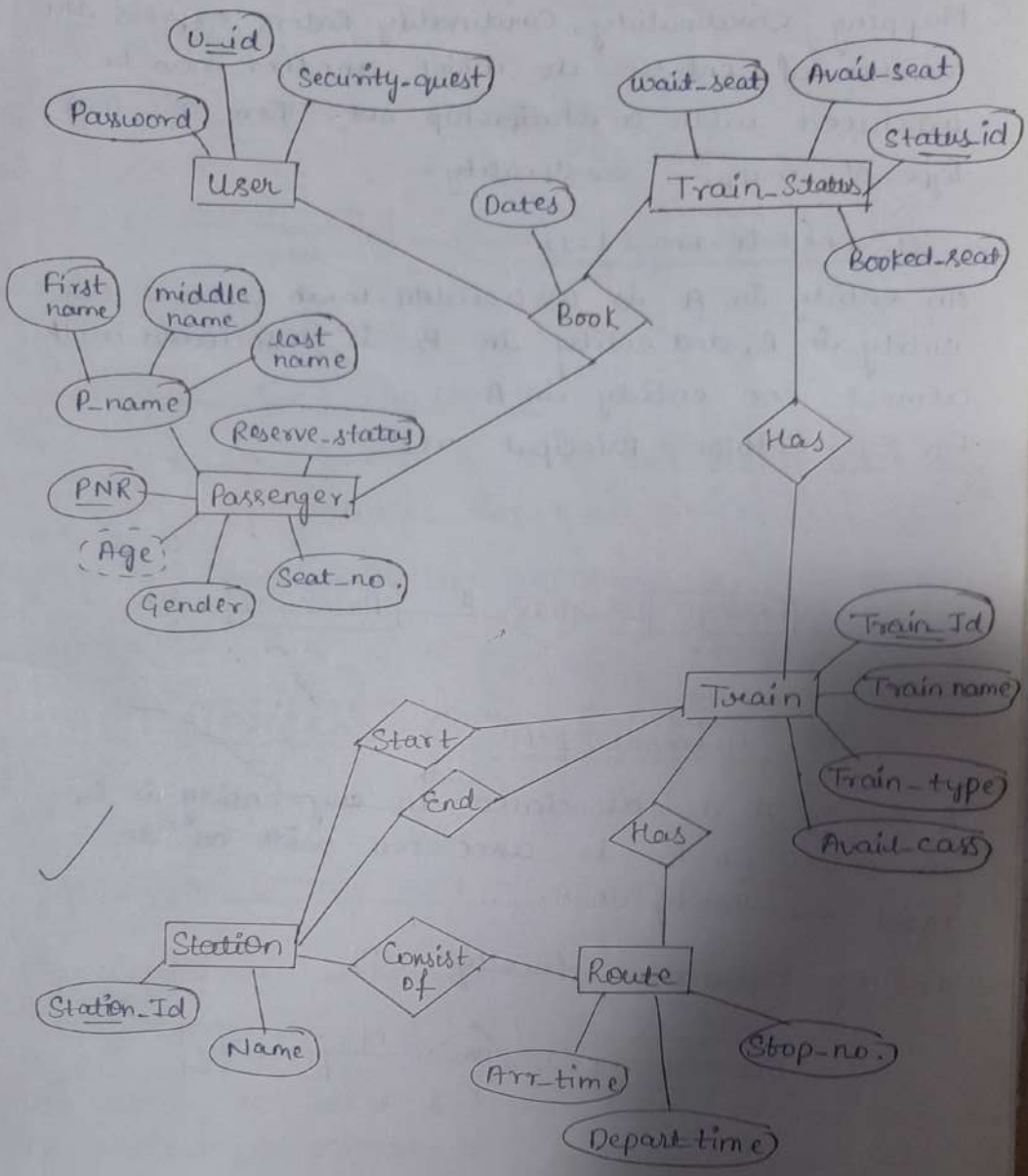
UNIVERSITY MANAGEMENT SYSTEM



BANK MANAGEMENT SYSTEM



RAILWAY MANAGEMENT SYSTEM



14/Aug/17

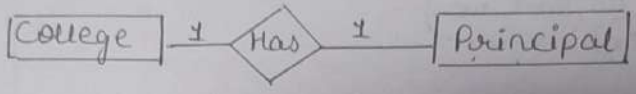
Mapping Cardinality

Mapping Cardinality, Cardinality Ratio, express the number of entities to which another can be associated with a relationship set. There are four type of mapping cardinality -

(1.) One-to-One (1:1)

An entity in A is associated with atmost one entity in B, and entity in B is associated with atmost one entity in A.

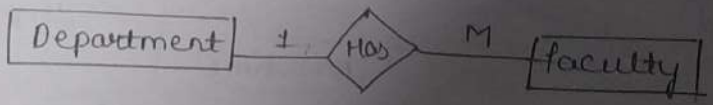
For Eg: College - Principal rel^n



(2.) One-to-many (1:M)

A entity A is associated with any entity in B, an entity in B is associated with atmost one entity in A.

For Eg: Department - faculty rel^n



(3.) Many-to-One (M:1)

A entity in A is associated with atmost one entity in B. An entity in B is associated with any number in entity A.

For Eg: Employee - project relⁿ

(4.) Many-to-many (M:M)

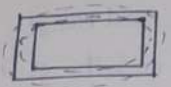
Entities in A and B are associated with any number of entities for each other.

For Eg: Book - Author relⁿship

✓ Strong and Weak Entity set

An Entity Set that has a primary key, it is called "Strong entity set". 

An Entity set that does not have sufficient attributes to perform a primary key, such entity set is called "weak entity set".

A weak entity set is indicated in E-R diagram by  and the corresponding identifying relationship by 

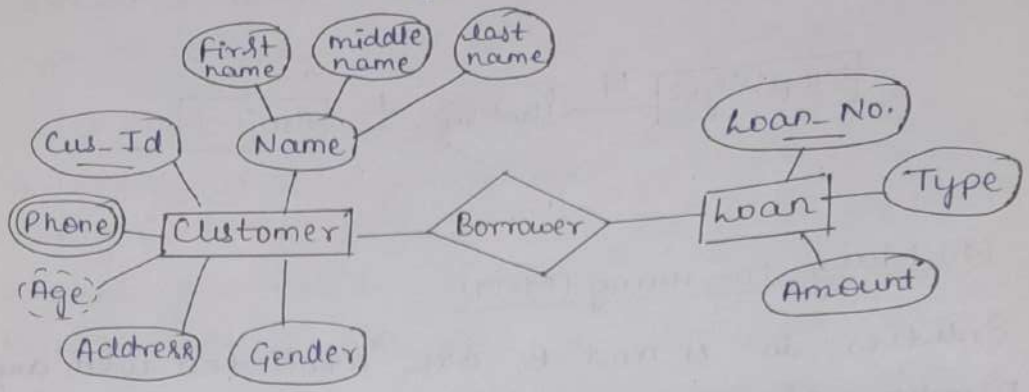
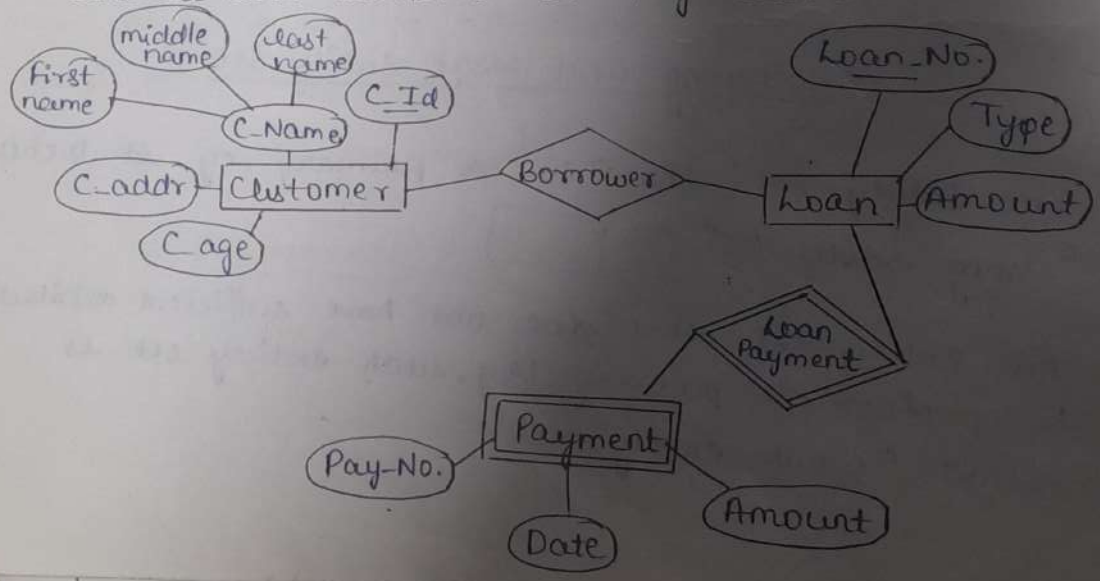


fig: Example of Strong Entity Set

Example of Weak Entity set

there are three entity set :- Customer (Cus_Id, C_Name, C_addr, C_age); Loan (Loan_No., Amount, Type); Payment (Pay_No, Date, Amount) where Cus_Id and Loan_No. is key attribute



17/Aug/17

Extended E-R Model

1. Specialization
2. Generalisation
3. Aggregation

Specifies weather the existence of an entity depends on its being related to another entity via a r/s set.

These constraints specify the min & max no. of r/s instances that each entity can/must participate in.

Specialization

* The process of designating sub-grouping of within a entity set is called "specialization".

  Eg: An entity set 'PERSON' with attributes Name, Address, a person may be further classified as one of the following 'Customer' and 'Employee'.

* It is a top-down process and it is denoted by " 

"

* The symbol 'ISA' for "Is a",

  Eg: That a 'Customer' is a 'PERSON' and 'Employee' is a 'PERSON'.

* The "is a" rel'nship may be reffered as Super class Sub-class rel'nship set.

Generalisation

* It is defined as a process of identifying some common characteristics of a collection of an entity set and creating a new entity set

that contains entity processing there common features.

* The design process in which multiple lower level Entity types are combined on the basis of common features to form higher level Entity type is known as "Generalization".

* This is Bottom-up approach. This approach result in identification of a generalised super class from original sub-class.