

Chapter-2

- Data Model
- Data Model is a collection of conceptual tools for describing data, data relationships, semantics etc.
- It is a simple representation of complex real world data structure.
- It is a communication tool to facilitate interaction among the designer, the application programmers and the end users.
- It defines how data is connected to each other and how it will be processed and stored inside the system.

The three levels of data Modeling are :-

i) Conceptual data model :-

- Identifies the highest level relationship between different entities.
- Includes important entities and their relationships.
- No attributes and Primary Key is specified.

ii) Logical data Model :-

- Describes the data in as much details as possible, without saying much about their physical implementation.
- It includes all the entities and relationship among them.
- All attributes for each entity are specified.
- Primary Key and foreign key are also specified.

iii) Physical data model :-

- Represents how the Model will be built in the database.
- Shows all table structures, including column name, data type constraints, primary key, foreign key and their relationship between tables.

Features	conceptual	Logical	Physical
Entity Name:	✓	✓	
Entity Relationships:	✓	✓	
Attributes:		✓	
Primary Keys:		✓	
Foreign Keys:		✓	✓
Table Name:			✓
Column Names:			✓
Column Data Types:			✓

Eg:- To store information of students, courses and also keep track of courses taken by students.

Student _____ course _____

Stu_course

Fig:- conceptual Data Model

Student	Course
Roll	CID
Name	Name
Address	Description
Phone	credit hr.

stu-course

St Course-ID

Roll (FK)

CID(FK)

Fig:- Logical data models

tbl_student
Roll: integer (PK)
Name: string
Address: string
Phone: integer

tbl_course
CID : integer (PK)
Name : String
Description : String
credit hr: integer

tbl_stu_course
st-course-ID : integer (PK)
Roll : integer (FK)
CID : integer (FK)

Fig:- Physical ~~data~~^{models}.

Alternate Data Models

i) Hierarchical Data Model

- It organizes data in the tree structure where there is a hierarchy of parent and child data segments.
- It has a single root segment connected to lower level segments.
- Each segments may connect to other lower level segments.
- Here, rule is that one parent can have many children but children are allowed only one parent.
- This allows information to be repeated through the parent child relation.

Advantages:-

- Allows easy addition and deletion of new information.
- Fast access to data at top.
- Relates well to anything that works on one to many relationships.

Disadvantages:-

- Allows data redundancy.
- Searching for lower level information is slow.
- Many to many relationships are not supported.

Student

1 Ram

2 Sita

3 Hari

C₁ DBMSC₂ OOPC₁ DBMSC₁, DBMS C₂ OOP

Fig:- Hierarchical Data Model.

ii) Network Data Model

- It replaces the hierarchical tree with graph.
- Allows to have more than one parent.
- Advantages :-
- Simple and easy to implement
- Can handle one to one and many to many relationships.
- Data access is easier than hierarchical model.

Disadvantages:-

- System complexity so difficult to handle and maintain.
- Lack of structural independence. i.e. change in structure makes change in application too.

Student

1 Ram

2 Sita

3 Hari

C₁ DBMSC₂ OOP

Fig:- Network data models

iii) Relational data models

- All data are stored in tables where each table consists of rows and columns.
- Here, Keys are used to order data or relate data to other tables.
- Also defines how data are manipulated.
- Defines set of rules to enforce data integrity, known as integrity constraints.
- Defines special feature called normalization to ensure efficient data storage.

Advantages :-

- Easy to use and Data independence.
- Security and Flexibility.

Disadvantages:-

- Ease of design can lead to bad design.
- Hardware overheads.

Roll	Name	Course-ID	Name
1	Ram	C1	DBMS
2	Sita	C2	OOP
3	Hari		

STU-Course-ID	Roll	Course-ID
SC1	1	C1
SC2	2	C2
SC3	2	C1
SC4	3	C1
SC5	3	C2

Fig:- Relational Data Models.

iv) Object-oriented Data models

- It is based on object oriented programming paradigm.

⇒ OOP consists of

- Object and object identifier.
- attributes and methods
- class
- class hierarchy and inheritance.

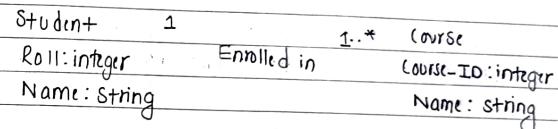


Fig:- Object oriented Data Models

v) E-R (Entity-Relationship) Model

- Defines the conceptual view of database.
- Works around real world objects called entities and relationship among them.

Entity :-

- A real world thing that is easily identifiable and distinguishable.
- Entity set is a collection of similar types of entities.
- Entity sets do not need to be disjoint.
- e.g:- entity set of employee and entity set of customer may have members in common.

Attributes :-

- They are the properties of entities.
- i) **Simple Attribute :-**
• Which can not be divided further eg:- Phone number

ii) Composite Attributes :-

- made of more than one simple attribute .eg:- name where name can be further divided into first name, last name

iii) Derived Attributes :-

- Do not exist in physical database but value is derived from other attributes. eg:- age where age can be derived from date of birth.

iv) Single valued Attribute :-

- Which have only one value .eg:- citizenship number.

v) Multivalued Attribute :-

- Which can have more than one value .eg:- email, phone number.
- Domain of attribute is set of permitted values.
eg:- mobile number. must have 10 digits.

Relationship

- It is association among the entities.
- Relationship set is a set of relationship of same type.

Degree of Relationship :-

- It refers the number of entity sets that participate in a relationship.

- unary relationship is of degree 1 that involves only one entity set.
- Binary relationship is of degree 2.
- Ternary relationship is of degree 3.

Relationships in database are often binary.

- It is possible to replace a non binary (n -ary; $n > 2$) relationship set by a number of distinct binary relationship set.

E-R diagram

- It is a data modeling technique that creates a graphical representation of entities and the relationship between entities, within an information system.

- It illustrates the logical structure of database.

- major components are :-

- Rectangles :-  to represent the entity set

- Ellipse :-  to represent attributes

- Diamonds :- 

to represent relationship set.

- Lines :- 

to link attributes to entity sets and entity sets to relationship set

- Double Ellipse :-



To represent multivalued attributes.

- Dashed Ellipse :-



To represent derived attributes.

- Double Rectangles :-



To represent weak entity.

Eg:-

- Entity:

Driver, Taxi

- Attributes

For Driver

license no

Name

address

Phone

For Taxi

chassis no

company

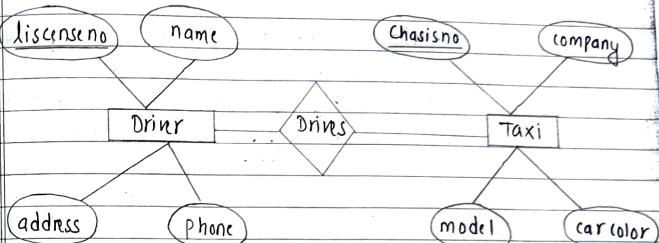
model

car color

- Relationship

Driver Drives taxi. So, relationship between Driver and Taxi is Drives.

ER-diagram



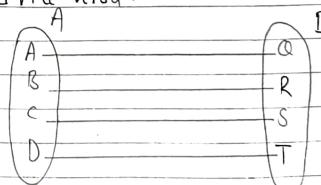
Mapping cardinalities

It express the number of entities to which another entity can be associated through a relationship set.

For a binary Relationship set R between entity A and B, the mapping cardinality must be one of the following:

- One to one

An entity is A is associated with at most one entity in B and vice-versa.

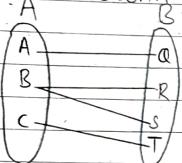


• One to many

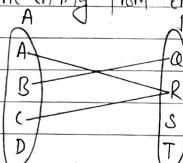
An entity in A can be associated with more than one entities of entity set B but from entity set B one entity can be associated with atmost one entity in A.

• Many to one

More than one entities from entity set A can be associated with at most one entity from B but one entity from B can be associated with more than one entity from entity set A.



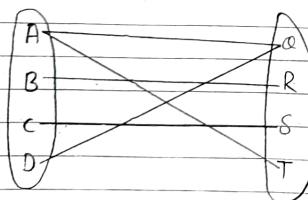
one to many



many to one

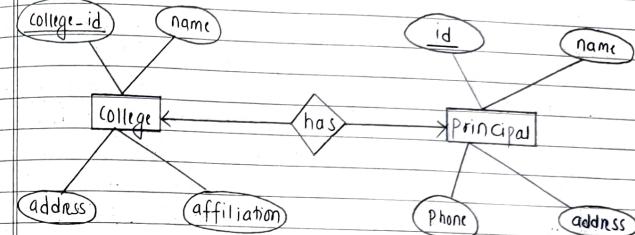
• Many to many

One entity from entity set A can be associated with more than one entity from entity set B and vice-versa.

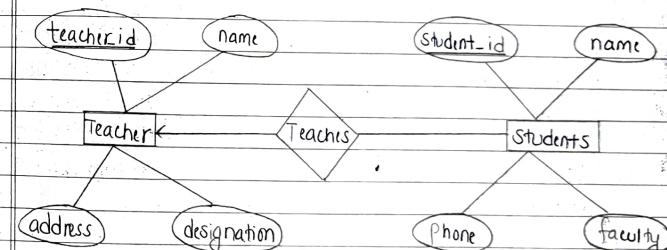


many to many

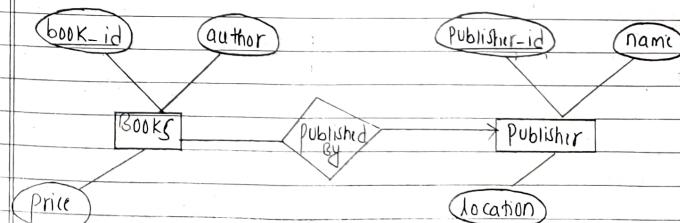
One to one Relationship



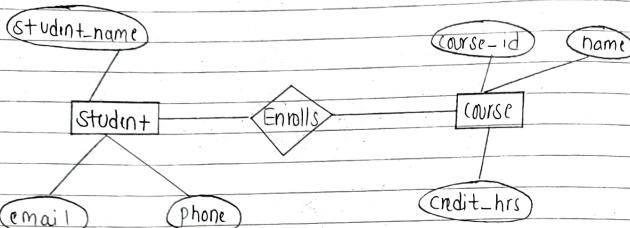
One to Many Relationship



many to one Relationship



Many to many Relationship



Keys

- It is set of one or more attributes whose values are distinct for each individual entity in entity set is called Key.
- A Key is single or combination of fields in table and used to fetch records/data from table according to conditions.

Types:-

- Super Key** :- It is a set of one or more attributes whose value uniquely determine each entity.
- Candidate Key** :- It is a minimal super key.
- Primary Key** :- It is a candidate key chosen as a principle means of identifying entities within entity set.
- Foreign Key** :- It is used to establish and enforce a link between two tables. It is a field in one table that uniquely identifies a row of another table.

Strong Entity set

- It has its own primary key.

- It is represented by a rectangle.

- It contains a primary key represented by underline.

- Its member is called Dominant entity set.

- Primary Key is one of attributes that uniquely identify its members.

- Total participation in relationship may or may not exist.

- Relationship between two strong entity set is represented by diamond symbol.

- Single line connects strong entity set with relationship.

Weak Entity set

- It don't have sufficient attributes to form a primary key on its own.

- It is represented by a double rectangle.

- It contains a partial key or discriminator represented by dashed underline.

- Its member is called subordinate entity set.

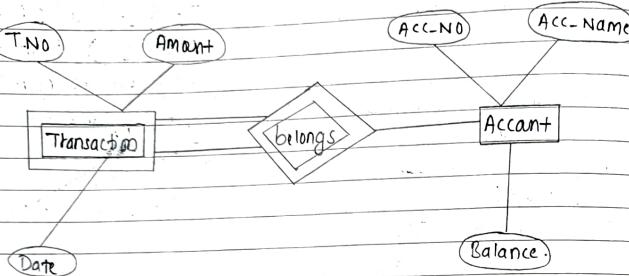
- Primary Key is combination of partial key and primary key of strong entity set.

- Total participation always exists.

- Relationship between one strong and one weak entity set is represented by double diamond symbol.

- Double line connects weak entity set with relationship.

eg:-



- Here, transaction is weak entity which is existence dependent on Account.
- Account is dominant entity while Transaction is subordinate.
- T-NO of Transaction is called discriminator.

eg:-

- A company has several departments. Each department has a supervisor and at least one employee. Employees must be assigned to at least one but possibly more departments. At least one employee is assigned to the project, but an employee may be on vacation and not assigned to any projects. The important data fields are names of departments, projects, supervisors and employees, as well as the supervisor and employee number and a unique project no.

Ans:-

Soln,

⇒ Identify Entities :-

- Company, Department, supervisor, Employee, Project.

⇒ Find Relationship :-

Company	Department	supervisor	Employee	Project
has				
exists in		run by		is assigned
		runs		
		belongs to		works on
				uses

⇒ Draw Rough ER-Diagram.

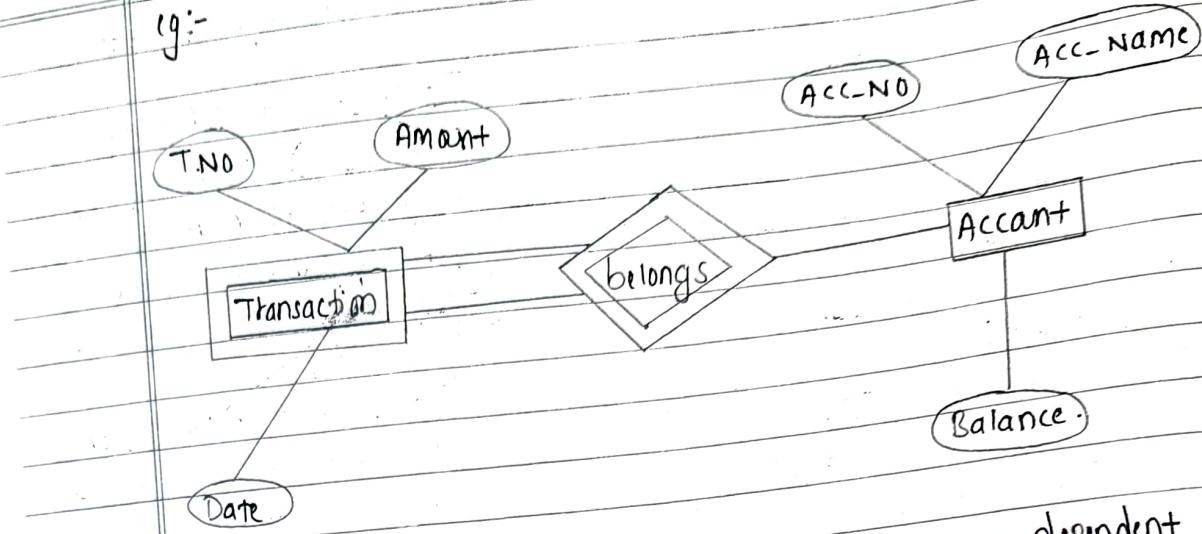
Company	has	Department
---------	-----	------------

Department	run by	Supervisor
------------	--------	------------

Department	is assigned	Employee
------------	-------------	----------

Employee	works on	Project
----------	----------	---------

eg:-



- Here, transaction is weak entity which is existence dependent on Account.
- Account is dominant entity while Transaction is subordinate.
- TNO of Transaction is called discriminator.

eg:-

- A company has several departments. Each department has a supervisor and at least one employee. Employees must be assigned to at least one but possibly more departments. At least one employee is assigned to the project, but an employee may be on vacation and not assigned to any projects. The important data fields are names of departments, projects, supervisors and employees, as well as the supervisor and employee number and a unique project no.

Ans:-

Soln,

GENIUS

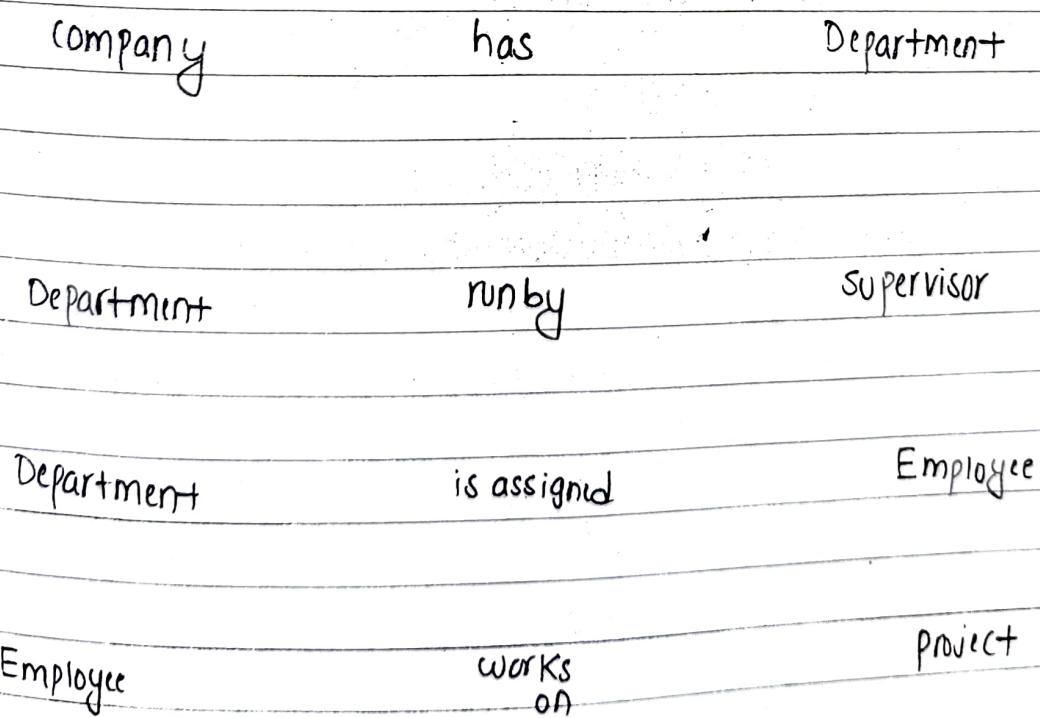
⇒ Identify Entities :-

- Company, Department, supervisor, Employee, Project.

⇒ Find Relationship :-

	company	Department	supervisor	Employee	Project
company		has			
Department	exists in		run by	is assigned	
supervisor		runs			
Employee		belongs to			works on
Project				uses	

⇒ Draw Rough ER-Diagram.



company has Department run by supervisor

is assigned

Project works on Employee

Filling cardinality:

one and only one one or more

zero or more zero or one

A B Each instance of A is related to minimum of zero and maximum of one instance of B.

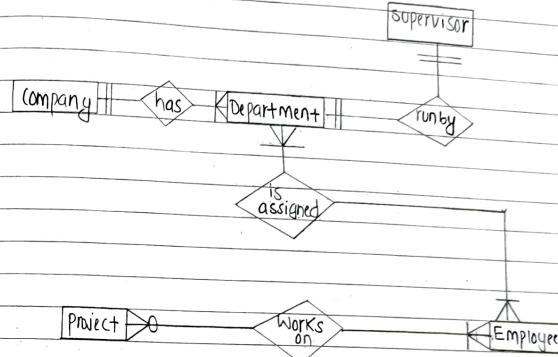
A B Each instance of B is related to minimum of one and maximum of one instance of A.

A B Each instance of A is related to minimum of one and maximum of many instances of B.

A B Each instance of B is related to minimum of zero and maximum of many instances of A.

Solution:

Fill cardinality:

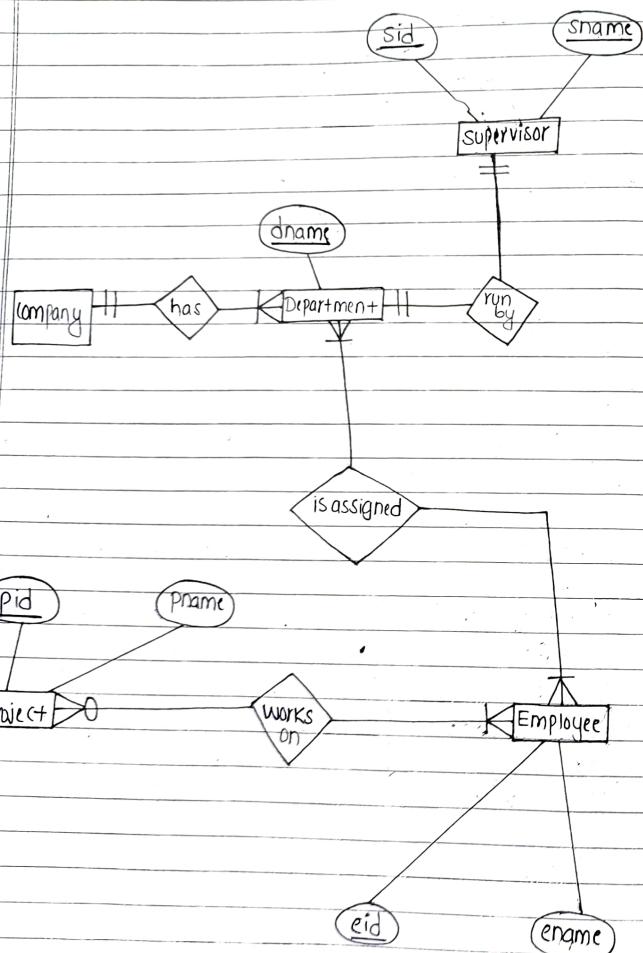


Identify attributes and define primary keys:-

- Employee-name (ename)
- Project-name (pname)
- Supervisor-name (sname)
- Department-name (dname)
- Employee-no (eid)
- Project-no (pid)
- Supervisor-no (sid)

Now,

Mapping and drawing fully attributed ER-Diagram.



ER Model

EER (Extended or Enhanced ER) Model

- IS the ER model combined with the additional features of semantic concepts.
- Shows complex relationship between Objects in a database.
- Some additional concepts in EER Model are:-
 - specification
 - Generalization
 - categorization
- Superclass and subclass
- Superclass is the entity type whose attributes are shared among subclass.
- Subclass is the one whose attributes are unique from other subclasses.

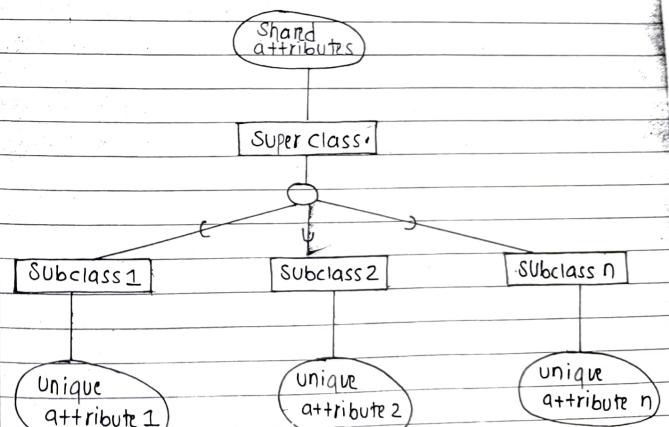


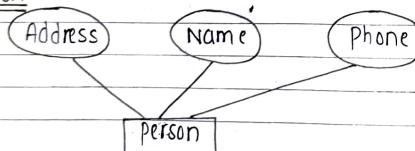
Fig:- Basic notation for superclass / sub class relationship.

Specialization

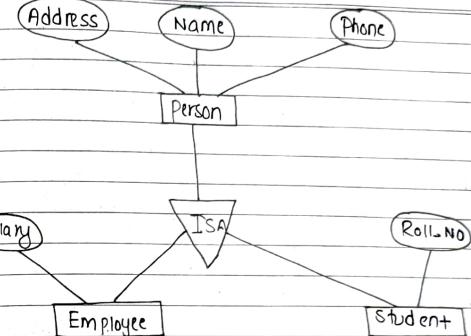
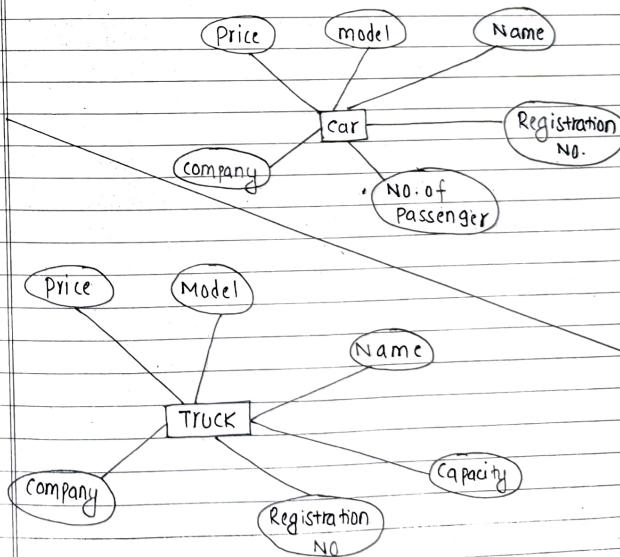
- Is a means of identifying sub groups within an entity set which have unique attributes.
- Is a top down process.
- Is a process of defining one or more subclasses from the superclass and forming superclass subclass relationship.
- In EER diagram, it is represented by a triangle component labeled ISA or simply circle with U symbol.

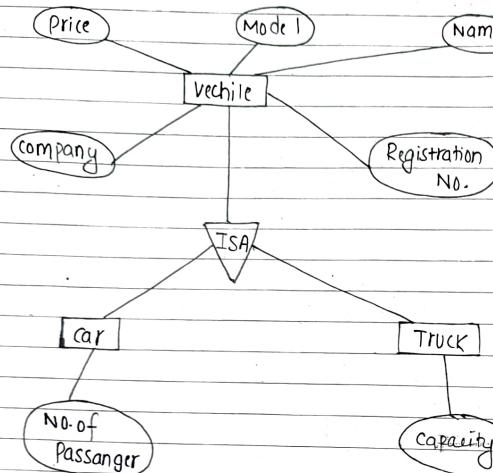
Generalization

- Is a bottom up approach opposite to specialization
- Is a process of forming a superclass.
- It emphasize the similarities among lower level entity set and hide the difference.

Specitization

Now,

Generalization:



Attribute inheritance :-

- Is a crucial property of superclass and subclass.
- Attributes of superclass are said to be inherited by subclass.

Single inheritance :-

- Also called as hierarchy.
- Defines that every subclass has only one super class.

Multiple inheritance :-

- Also called as lattice ..
- Defines that subclass can have more than one super class.
- A subclass with more than one superclass is called Shared Subclass.

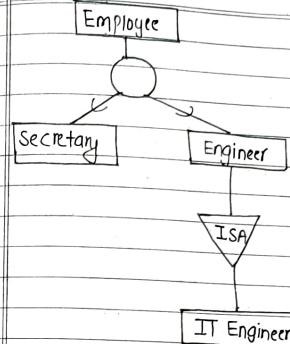


Fig:- single Inheritance (Hierarchy)

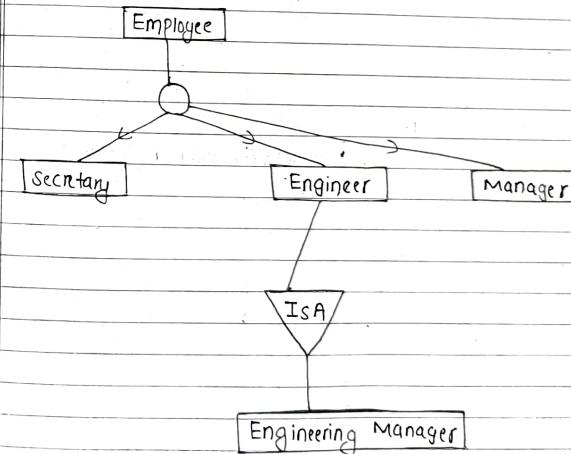
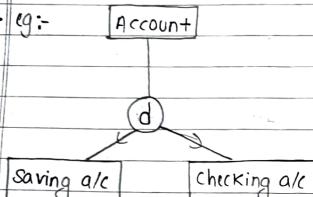


Fig:- Multiple Inheritance (Lattice)

Disjoint constraint

- Subclass belongs to no more than one lower level entity set.
- Is represented by placing 'd' in circle or simply by ISA relationship.

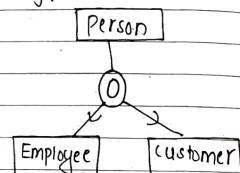
eg:-



overlapping constraint

- Subclass may belong to more than one lower level entity set.
- Is represented by placing 'd' in circle or simply by ISA relationship.

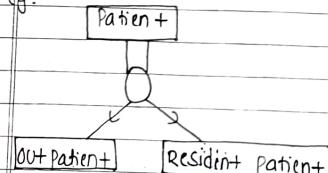
eg:-

Total generalization/
Specialization

- Every entity is superclass must be member of at least one subclass.

Double lines are used to connect superclass to the circle.

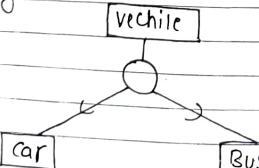
eg:-

Partial generalization/
specialization

- Every entity in superclass may not belong to subclass.

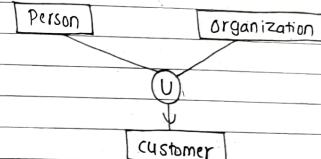
Single line is used to connect superclass to the circle.

eg:-



categorization (Union Types)

- Models a class/Subclass with more than one superclass of distinct entity types.
- Subclass is called category.
- only one Subclass exists in categorization.
- Attribute inheritance is selective.
- category members must exist in at least one of its superclass.

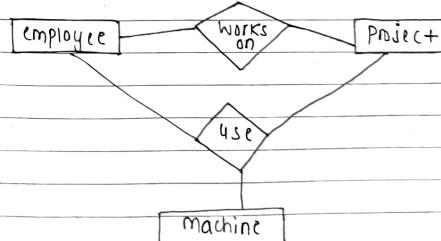


Here, category customer is subclass of union of person, organization.

Aggregation

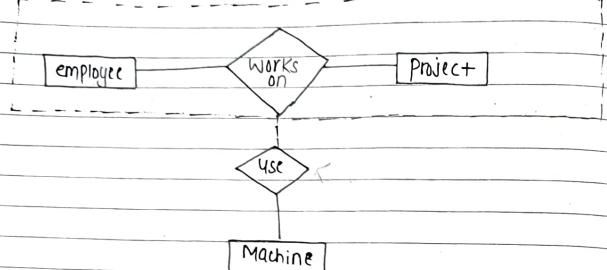
- Is an abstraction through which relationships are treated as high level entities.

eg. employee works on project and use no.of machine during that work.



Here, problem is {employee, project} combination that appears in works on relationship also appear in use relationship.

Now, Solution is using aggregation, as below .



- Here, relationship set works on and entity sets employee and project are treated as high level entity called works.

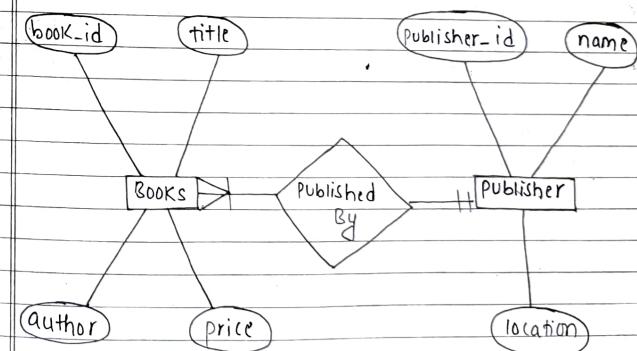
Relation with UML class diagram.

- UML :-
- It stands for unified Modeling language .
- It is a standard language for specifying , visualizing , constructing and documenting that artifacts of software systems .
- UML is a modern approach to modeling and documenting software .
- It is based on diagrammatic representations of software components .
- There are several types of UML diagrams like class Diagram , Use case Diagrams , Sequence Diagram , Activity Diagram etc .
- Class diagram
- It shows the classes in a system , attributes and operations of each class and the relationship between each class .
- In class diagram , Name is at the top , attributes in the middle and operations or methods at the bottom .

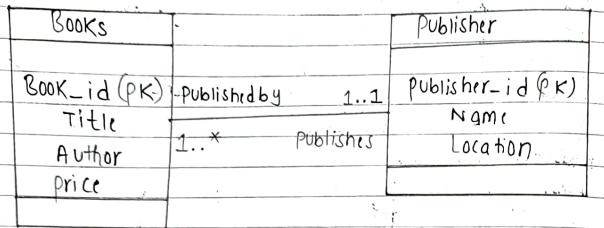
UML Notations for ER Models:-

- Entities are modeled as a class ; The name of the class is the entity name and the attributes of the class are the entity's attributes .
- The name and attributes can be arranged in a box style .
- Relationships are shown as a single solid line connecting the two entities .
- The minimum and maximum cardinalities are shown along the line and verb phrases can be added to completely characterize the nature of the relationship .
eg:-
- One publisher may be publishing one or more Books .
- One Book must be published by one and only one publisher .

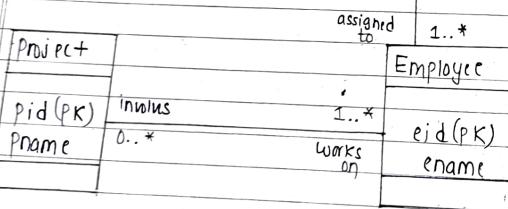
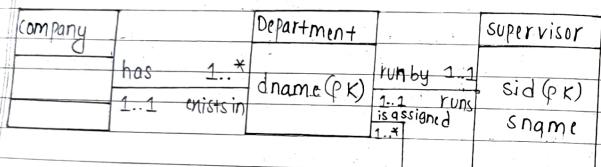
ER model or ER Diagram



UML Class Diagram



UML class Diagram for above ER diagram example.



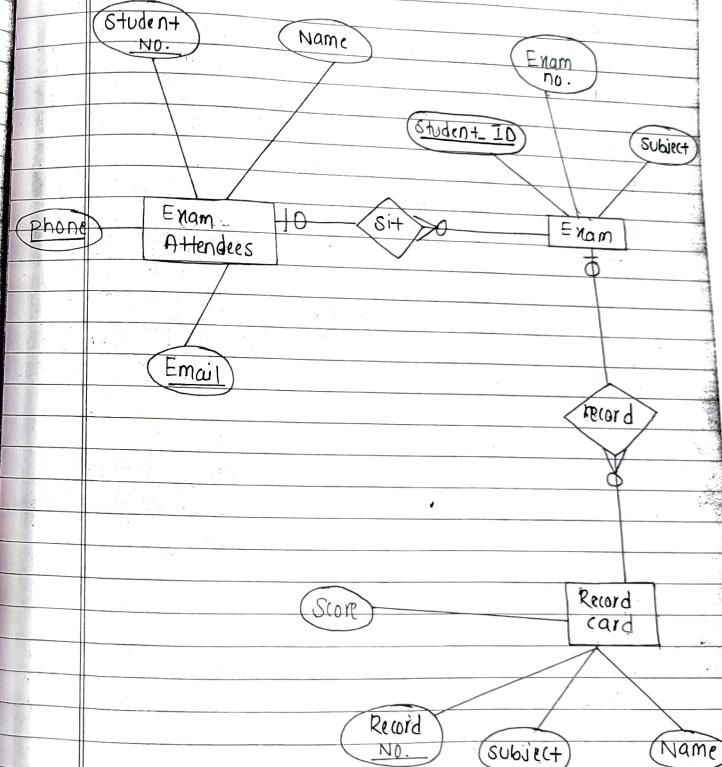
Chapter 2 old is gold solutions

2011 Fall 1b) repeated.

2011 Spring 1b) repeated.

2012 Fall

- 1b) construct an ER diagram for keeping records for exam sections of a college.

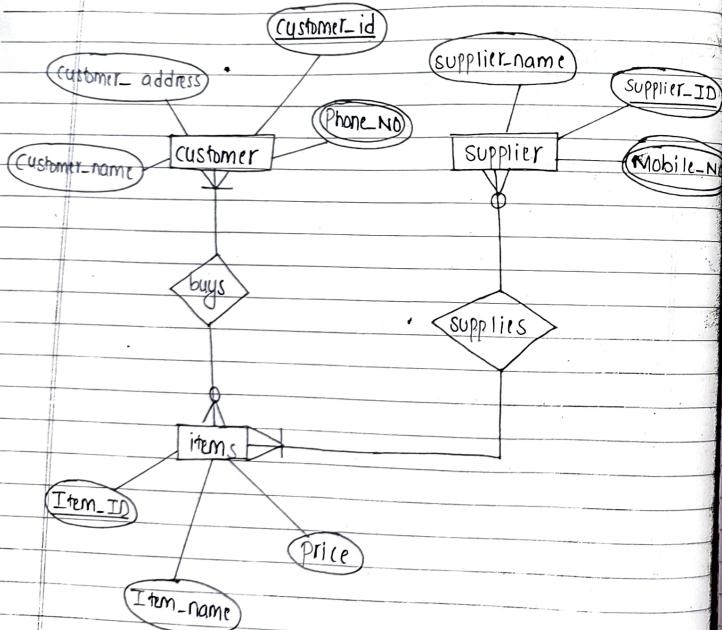


2012 Spring 1b) repeated

2013 spring

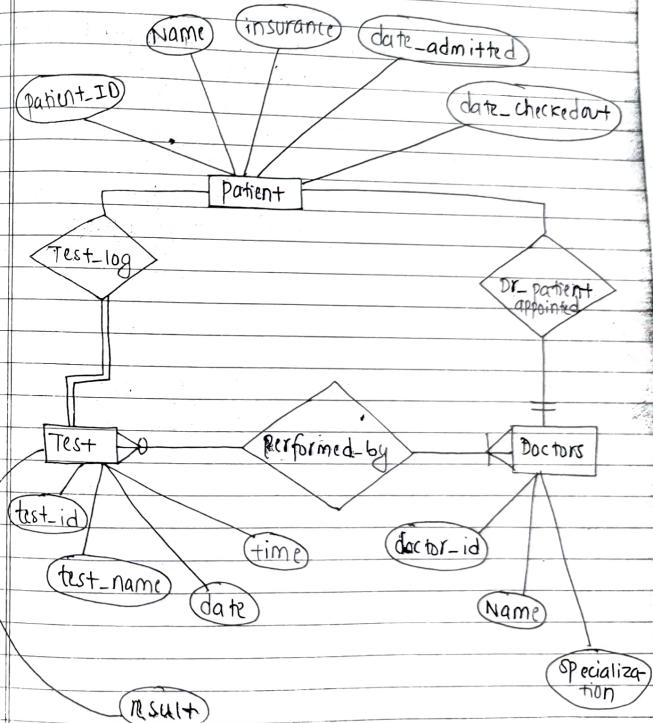
1b) construct an ER Diagram for a departmental store that keeps the information about customer, supplier and items. Associate with a log of various selling and purchasing information. The ER diagram should explain Entity name, some appropriate attributes of the entity relation among the entity, association and directivity among the entities.

Soln Ans is in book Also page No. 13



2014 - Fall

1b) Draw an E-R Diagram for the database of Hospital with a set of patients and a set of medical doctors. With each patient a log of the various tests conducted is also associated. Make your own assumptions if necessary.



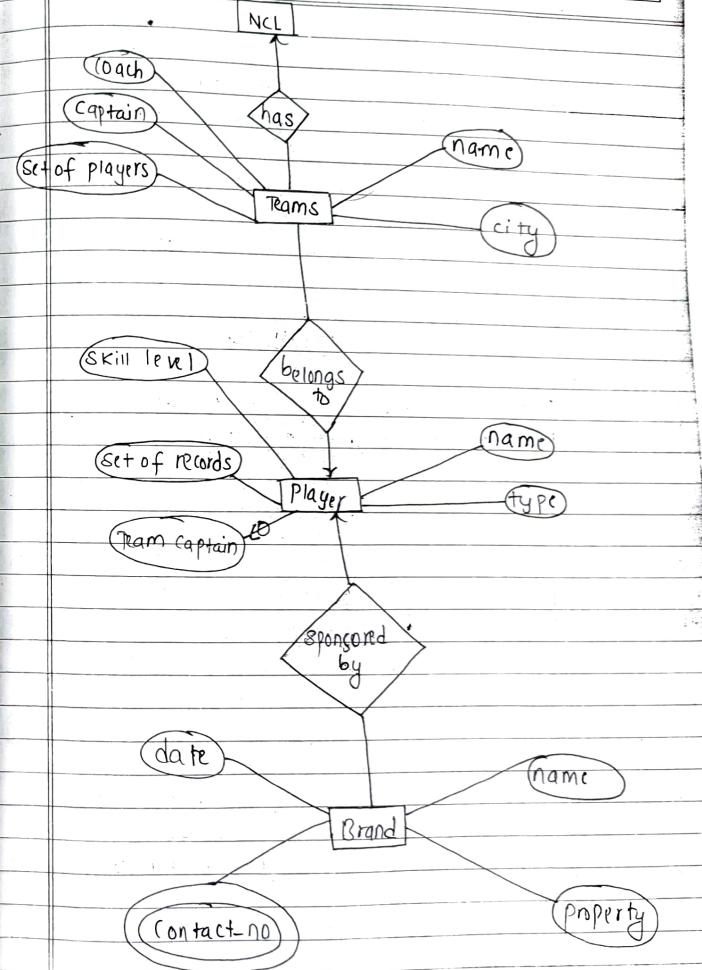
2015 spring

- 16) suppose you are given the following requirements for a simple database for National Cricket League (NCL)

 - the NCL has many teams.
 - each team has a name, a type city, a coach, a captain and a set of players.
 - each player belongs to only one team.
 - each player has a name, a type (such as batsman or bowler), a skill level, and a set of records.
 - a team captain is also a player.
 - each player is sponsored by at least one brand.
 - a brand has its name, established date, property, multiple contact+no.

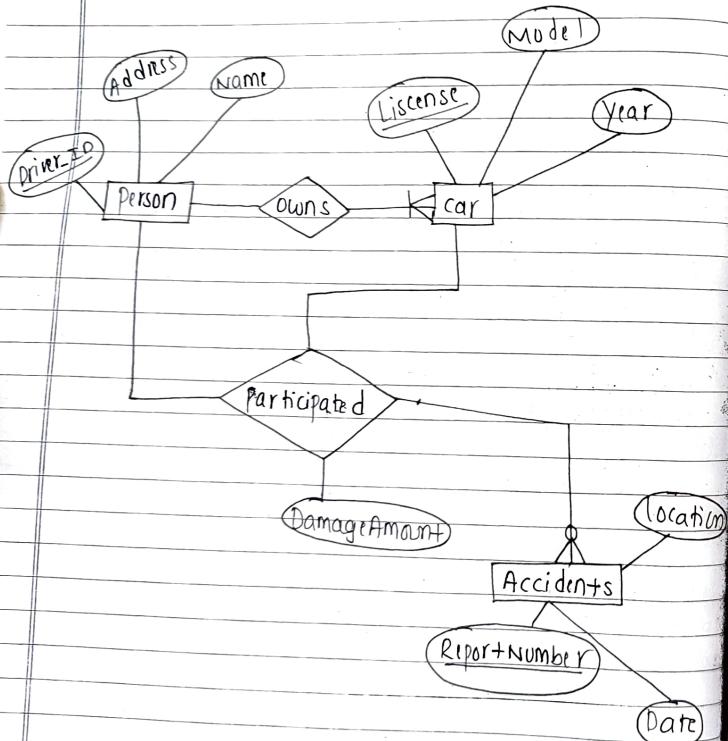
Construct a clean and concise ER-diagram for the NCI database - list your assumptions and clearly indicate the cardinality mapping as well as any role indicators in your ER-diagram.

四〇



2016-Fall

- 1b) construct E-R model for a car insurance company whose customer own one or more cars each. Each cars has associated with it zero to any number of recorded accidents. Also design a relational database corresponding to the E-R diagram.

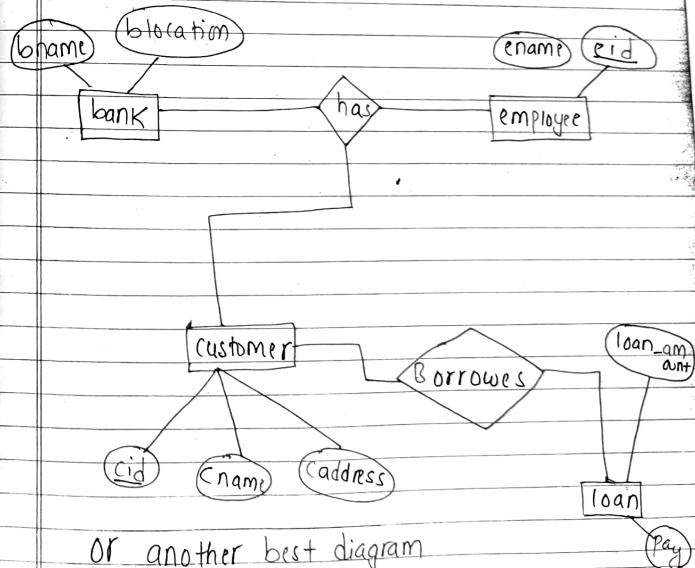


car insurance tables:

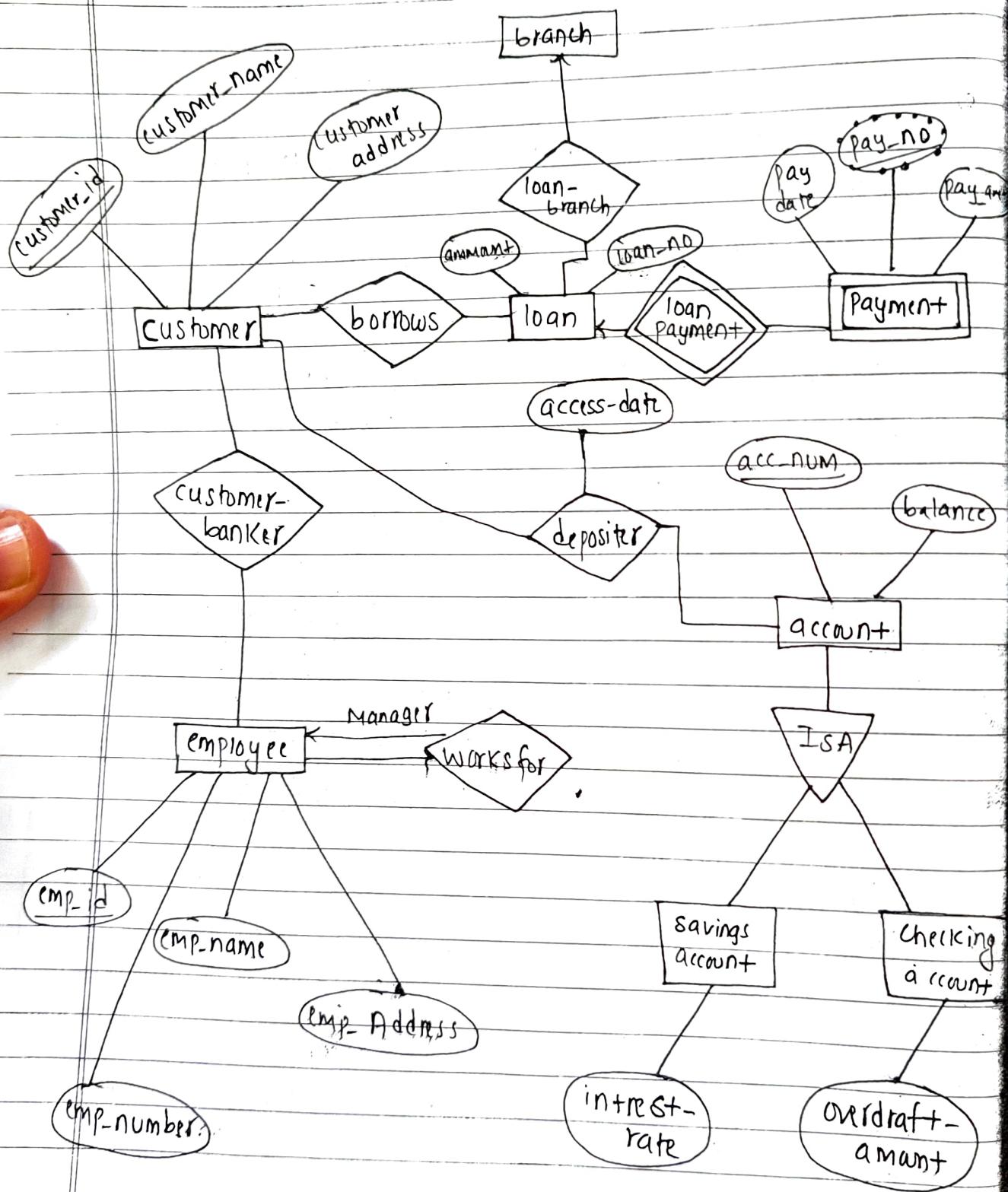
Person (driver_id, name, address)
 car (license, year, model)
 accident (report-number, date, location)
 participated (driver-id, licence, report-number, damage-amount)

2016-spring

- 1b) construct an ER diagram for a banking enterprise that keeps the information about employee, customer, loan, account and payment.

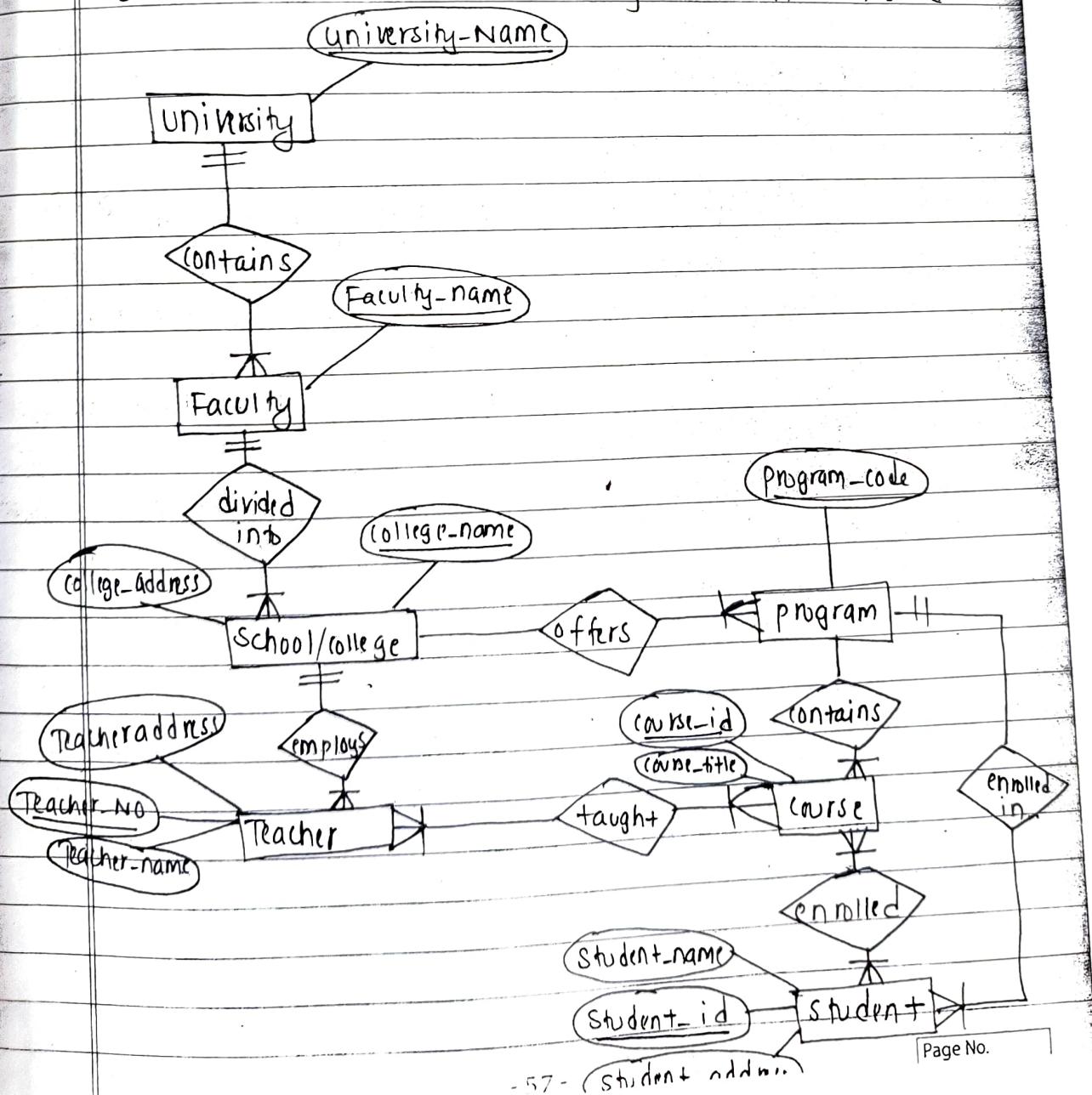


OR another best diagram



2017 Fall

- 1b) A University contains many faculties . The faculties in turn are divided into several colleges . Each college offers numerous programs and each program contains many courses . Teachers can teach many different courses and even the same course numerous times . courses can also be taught by many teachers . A student is enrolled in only one program but a program can contain many students . Students can be enrolled in many courses at a same time and the courses have many students enrolled .



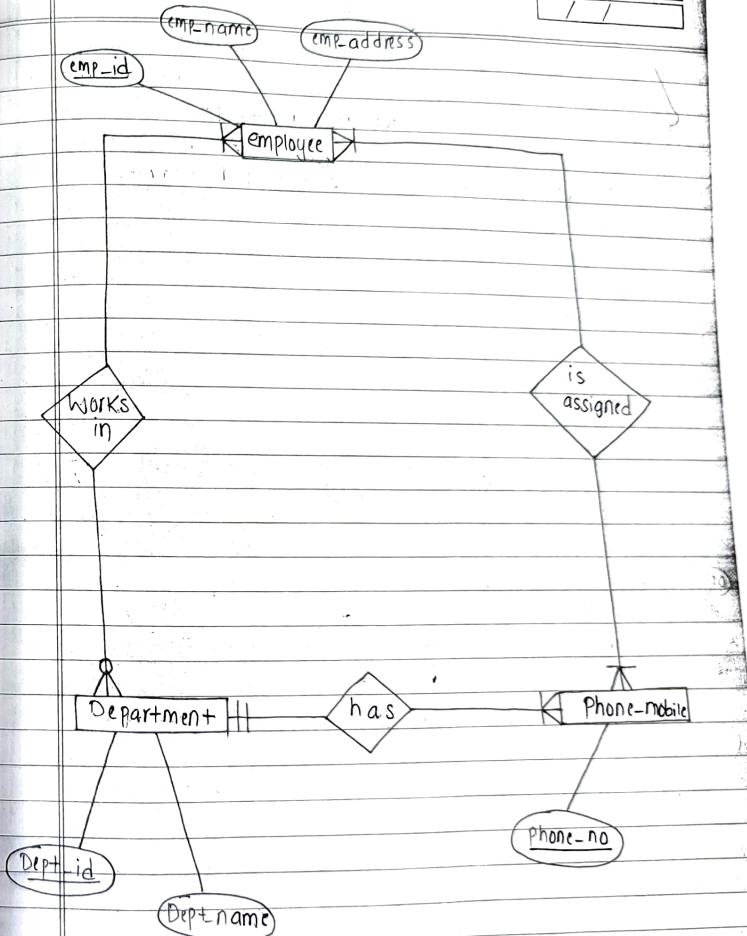
GENIUS

2018 - Fall

1b) suppose you are given the following requirements for a simple database for the employee Management system:

- i) An employee may work in up to two departments or may not be assigned to any department.
- ii) Each department must have one and may have up to three phone numbers.
- iii) Each department can have anywhere between 1 and 30 employees.
- iv) Each phone is used by one, and only one department.
- v) Each phone is assigned to at least one, and may be assigned to up to 30 employees.
- vi) Each employee is assigned at least once, but no more than 5 phones.

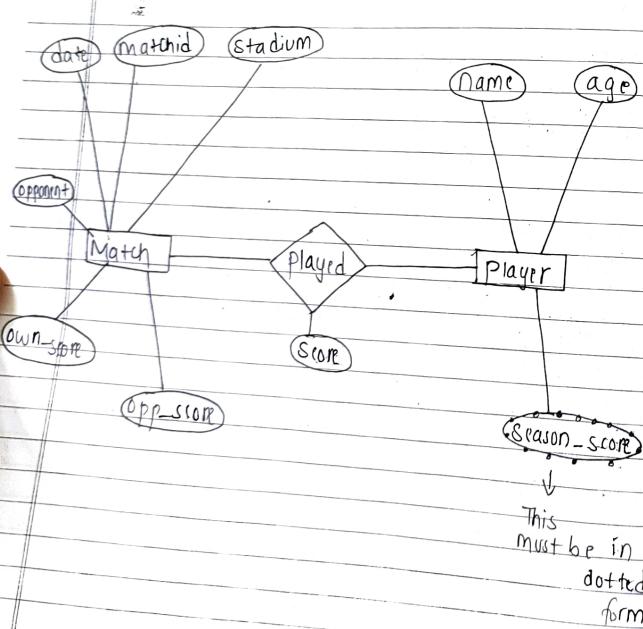
Construct an ER diagram for the database. clearly indicate the cardinality mapping.



- 5b) Design an E-R diagram for keeping track of the exploits of your favourite sports team. You should store the matches played, the scores in each match, the players in each match and individual player statistics for each match. Summary statistics should be modeled as derived attributes.

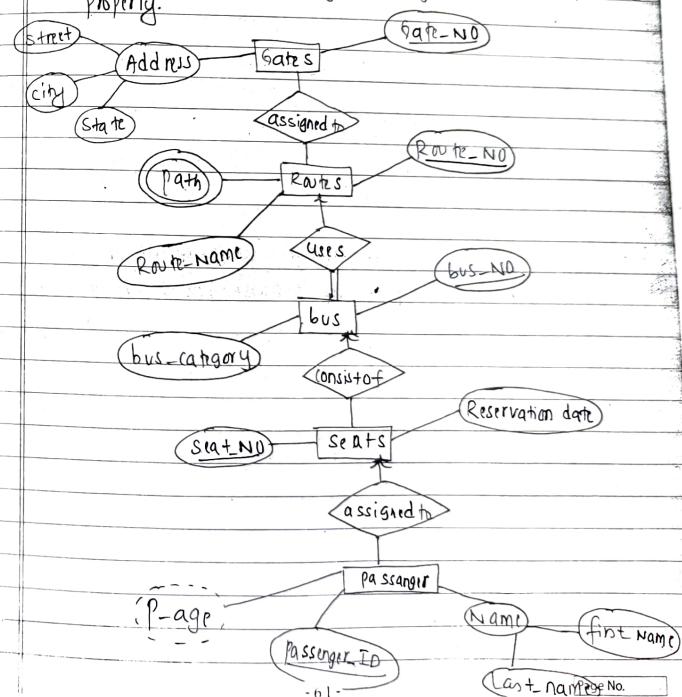
entity

match and player.



2018-spring

- 1b) Construct an ER-diagram for a Metropolitan Bus Park. Different gates are assigned to different routes. Routes are assigned to different buses. Buses consist of different seats which are assigned to different passengers. Frequent travelers are also in passenger. Associate a log of reservation date while reserving seats. The passenger name must have two attributes first-name & last-name. Each of the entities must have primary key attribute as far as possible. The cardinality mapping should be explained properly.



GENIUS

Benefits of data model are:-

- Provides very efficient "high-speed" retrieval.
- Ability to handle more relationship types.
- Ease of data access.
- Data Integrity
- Data Independence
- Higher quality and Faster performance.
- Reduced cost and Fewer data errors.
- Better documentation, Fewer application errors and Improved data quality.

Disadvantages of data models are:-

- Lack of structural independence.
- System complexity.

1) # Draw ER-Diagram for a vehicle management system including primary key, weak entity, composite attribute, derived attribute and multivalued attributes in your ER-diagram.

2) # Construct an ER diagram for Keeping records for Library Management systems

Ans :-

