22/9/21 Database managment system

→ A DBMS is a collection of interrelated data and a set of programs to access the data.

→ The collection of data used only before to as the data base.

→ The primary goal of DBMS is to praise provide away to store and retrieve database information that is both convinient and efficient.

→ The database system designed to manage :-

i) Large bodies of information.

ii) Display the structure of storage information.

iii) Manipulation of information.

iv) Safety & system crash and unauthorised access.

Applications :-

> Airlines
> Banking
> telecommunication
> credit cards, debit cards
> university

Tree structure (or) hierachy - parent & child combination

Database management system :- (A system or organization that has many levels from lowest to highest)

This model is like a hierarchical tree, structure, used to construct a hierarchy of records in the form of nodes and branches The data elements present in the structure have parent-child relationship. closely related information in the parent-child structure is stored together as a logical unit. A parent unit may have many child units, but a child is restricted to have only one parent.

The drawbacks of this model are :-

The hierarchical structure is not flexible to represent all the relationship proportion which occur in the real world.

It can't demonstrate the overall data model for the enterprise because of the non-availability of actual data at the time of designing the data model. It cannot represent the many-to-many relationship.

Network model :-

It supports the one-to-one and one-many types only. The basic objects in this model are data, items. Data ~~aggregents~~

aggregates records and sets.

It is an improvement on the hierarchical model. Here multiple parent-child relationships are used. Rapid and easy access to data is possible in this model due to multiple access paths to data elements.

Relational model :-

- Does not maintain physical connection between relations data is organized in terms of rows and colums in a table.
- The position of a row or column in a table is of no importance. The intersection of a row and column must give a single value.

Features of an RDBMS :-

The ability to create multiple relations and enter data into them an attractive query language.

Retrieval of information, stored in more than one table.

An RDBMS product has to be satisfy at least seven, of the 12 rules of codd to be acceptedas a full fledged RDBMS.

Relational Database management system :-

RDBMS is acronym of relation database management system Dr. E.F Codd first introduced the relational database model in 1970.

The relational model allows data to be represented in a simple row-column. Each data field is considered as a column and each record is considered as a row.

Relational database is more or less similar to database management system. In relational model there is relation between their data elements. Data is stored in tables. Table have columns, rows and names. Tables can be related to each other if each has a column with common type of information. The most famous RDBMS packages are oracle, sybase and informix.

Student Details table :-

Roll.no	s.name	s.Address
1.	Chandralekha	satellite
2.	Harshini	Ambawadi
3.	Akshitha	Naranpura

Student marksheet table:-

Roll no	Sub 1	Sub 2	Sub 3
1.	78	89	94
2.	54	65	77
3.	23	78	46

Here both tables are based on students details. Common field in both tables in Roll no. So we can say both tables are related with each other through Roll no column.

Degree of relationship:-

- One to one (1:1)
- One to many or many to one (1:M/M:1)
- Many to many (M:M)

The degree of relationship indicates the link between two entities for a specified occurence of each.

One to one relationship:-

(1:1)

Student has Roll no

one student has only one Roll no for one occurence of the first entity, there can be at Mostone related occurence of

entity and vice-versa.

One to many or many to one relationship :-

(1 : M / M : 1) 1M

- Course contains students.

As per the institution norm, one student can enroll in one course at a time however in one course, there can be more than one student.

For one occurence of the first entity there can exist many related occurences of the second entity and for every occurence of the second entity there exists only one associated occurence of the first.

Many to many relationship :-

(M : M) MM

- Student appears tests :-

The major disadvantage of the relational model is that a clear-cut interface can't be determined reusability of a structure is not possible.

The relational database now accepted model on which major database system are built

- oracle has introduced added functionality to this by incorporated object-oriented capabilities. Now it is known is as object relational database management system (ORDBMS). object-oriented concept is added in oracle.

Roadway travels :-

"Roadway travels" is in business since 1977 with several buses connecting different places in India. Its main office is located in hyderabad.

The company wants to computerize its operations in the following areas.

Reservations

Ticketing

cancellations.

Reservations :-

Reservations are directly handeled by booking office. Reservations can be made 60 days in advance in either cash or credit. In case the ticket is not available, A wait listed ticket is issued to the customer. This ticket is confirmed against the cancellation.

Cancellation and modification :-

Cancellations are also directly handed at the booking office. cancellation charges will be charged.

Wait listed tickets that do not get confirmed are fully refunded.

23/9/21

Purpose of DBMS :

(duplication)

→ Data redundancy and inconsistency
→ Difficult to access the data.
→ Data Isolation
→ Integrity problems
→ atomicity problems
→ Concurrency access anomilies
→ security.

Advantages :-

- Data
- efficient data access
- Data integrity and security
- Concurrent access and trash recovery
- Reducing application development time
- Data administration

25/9/21

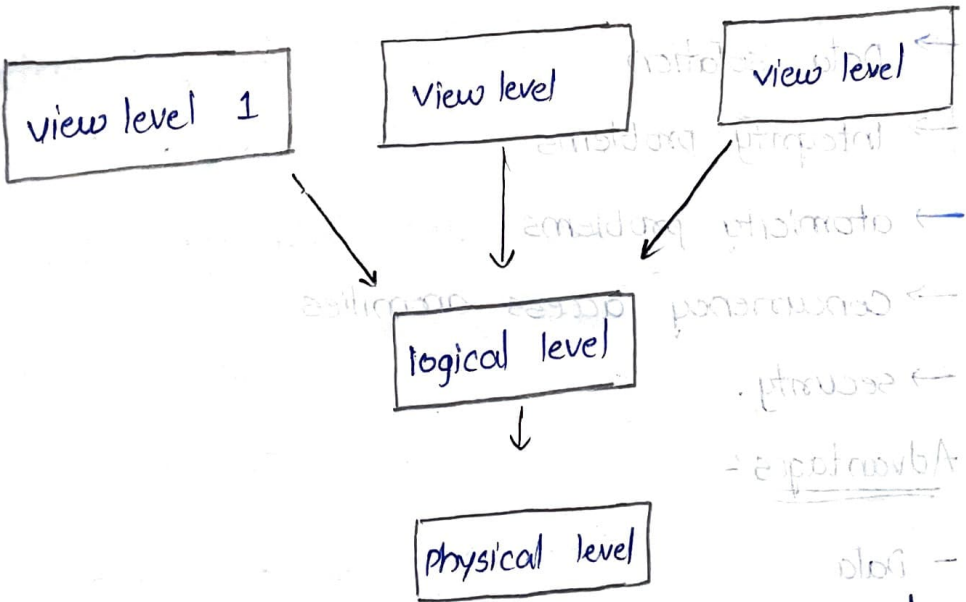
View of data :-

→ A database system is a collection of interrelated data and a set of program that allow users to access & modify the data

→ The major purpose of database system is to provide users with an abstract view of the data. (overview of DB is abstract)

→ That is, the system hides some certain details of how the data are stored and maintained.

Data abstraction or levels of abstraction :-



physical level :- How the data are actually stored

logical level :- what data are stored in database.

view level :- Only part of data of entire database.

29/9/21

Instance and schema :-

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

→ The overall design of database is called database schema.

1. Physical schema - Design of database at physical level
2. logical schema
3. View schema (or) external schema.

Data models :-

1. Relational model - collection of tables
2. Entity relationship model (E-R model) - maintain Relationship b/w tables
3. Object based data model - protection of data by encapsulating
4. semi structured data model (XML).

XML - extendable markup language

Database model :-

SQL (structured query language)

- → DML - data manipulation language
- → DDL - data defination language.

Data manipulation language :-

→ DML requires user to access or manipulate data as organised the appropriate data model.

→ In the DML i) retrieval of the information stored in the database, ii) insertion of new information in DB

iii) Deletion of information from database

iv) modifyication " stored in the database.

→ DML is mainly of two types :-

i) procedured DML requires a user to specify what data are needed & how to get those data

ii) Declarative DML requires a user to specify what data are needed without specifying how to get those data.

→ A query is a statement requesting the retrieval of information, and the portion of DML, that involves information retrieval is called query language.

Data defination language :-

→ specify a the database schema by a set of defination expressed by a special language is called data defination language

→ they specify storage structure used by data base and / access methods used by database system by the set of statements in a special type of DDL called data storage or defination language.

- Domain constraints (data types)
- Refferential Integrity (Relationships & dependencies)
- Assertions (conditional structures)
- Authorizations. (permission

30/9/21

Relational database :-

- collection of tables.

customer table

unique name

Customer id	customer name	customer street	customer city
1	A	AB	AAA
2	B	BC	BBB
3	C	CD	CCC
4	D	DE	DDD

Records or tuples

physical level

customer colums

Account table

Acc number	Acc balance
A-101	500
A-102	600
A-103	700
A-104	500

depositor-table

customer id	Acc number
1	A-101
2	A-102
1	A-103
3	A-104

→ Relational database is based on relational models and use a collection of tables to represent both data and relationships among the data.

→ Data manipulation language (DML) working structure.

Find the name of all customers who reside the customer city AAA.

Table field

Select customer . customer name

from customer

where customer . customer-city = "AAA"

} logical level

View level

Id	name	Street	city
1.	A	AB	AAA

→ The query involves information from more than 1 table.

Find the account numbers and corresponding balance of all accounts hold by the customer with customer id equal to 1.

```
Select   account.account number,
         account.account balance
from     Accounts, depositor
where    depositor.customer id = "1" and
         depositor.account number = account.aecount number;
```

Acc number	Acc balance
A-101	500
A-103	700

Data defination language :-

- SQL provides a rich DDL that allows to define tables, relationships, integrity constraints, assertions, authorizations.

Query :-

i) Create a table customer with (customerid, customername, street, city)

ii) Create a account table

iii) Create a depositor table

→ Create table customer (

customer_id int(15),
customer name varchar (20),
customer street varchar (15),
customer city varchar (30));

→ create table account (

account number varchar(20),
account balance varchar (15));

→ create table depositer (

customer id varchar(15),
account number varchar (20));

Database access for application programs :-

→ Application programs that are used to interact with database example :- Banking sector

→ To access the database, data manipulation need to extended from host language

1/10/21

Database design :-

→ Database design is a structure or system designed to manage large bodies of information

→ The large bodies of information do not exist in isolation

→ The database design mainly involves the design of database schema.

→ The database design mainly considers

i) Design process

ii) Data requirements of data base.

iii) Choose the data models

iv) Relational models

Entity-relationship models :- [E-R model]

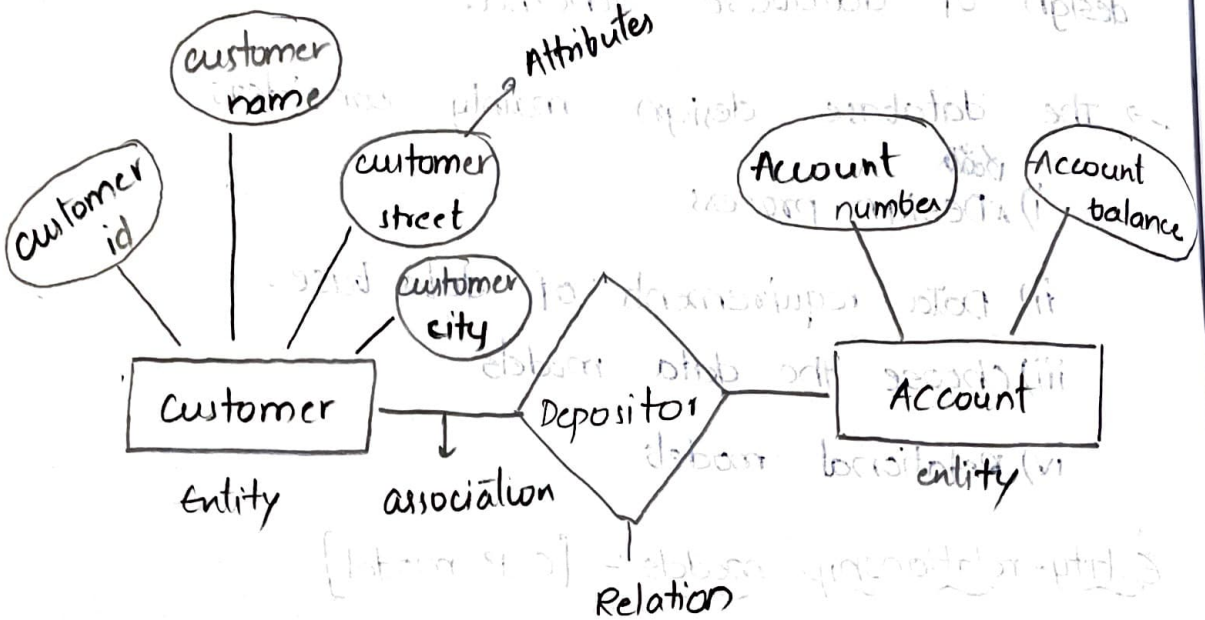
The collection of basic objects called entity. The entity is thing or object

Ex :- student, Bankaccount, employes.

→ Entities are described in a database by set of attributes

→ Relationship is an association among several entities

→ The set of all entities of same kind and set of all relationships of same kind are called An entity set and relationship set.



Rectangle - entity

ellipse - attributes

Rhombus - Relation

lines - association between entity and relationship

4/10/21

Normalization :-

→ The relational database is used to a process is known as normalization.

→ Normalisation is a process of organizing a data in a database and it is reduce data redundancy and eliminates undiseriable characters like insert, update, delete and anominies.

→ In normalization, one solution to be deformed problem is introduced null values. This value indicates that the value doesn't exist. Unknown may be missing or not known

Object based and semi structure data models :-

→ The object oriented model can be seen as extending the entity-relationship model (e-R model) with notations of encapsulation, methods and object identities.

→ The object relational data models, extends traditional relational model with a variety of features such as structured and collection types. As well as object orientation.

semi structured data model :-

→ semi structured data model permits the specification of data where individual data items of the same type di may have different set of attributes.

→ XML - extentable markup language.

Data storage and Query:-

- **Storage manager:-**

A storage manager is a program module that provides interface between the low level data or physical level stored in data base and the application programs and queries submitted (logical level) (view level) to the system.

components of storage manager:-

1. Autherization and integrity manager
2. Transaction manager
3. File manager
4. Buffer manager

Data structures:-

- Data files
- Data dictionary -
- Indies (Index)

5/10/21

- **Transaction management:-**

A transaction is collection of operations that performs a single logical function in database application.

→ one of the major use of DBMS is to protect the users data from system failures.

→ There are three operations can be performed in transaction management is

Read - access the data

write - change the data

commit

→ Example : Transfer of $500 from Account A to Account B. Initially Account A = 500, Account B = 800

R(A) = 500 / access the data

A = A - 50 / dedecting 50 from A

W(A) = 450 / updated in RAM

R(B) = 800 / Access the data

B = B + 50 / 50 is added to the B's account

W(B) = 850 / updated in RAM

commit / the data in RAM is taken back to ROM

→ A transaction in, DB system, A transaction must maintain ACID properties are :

A - Atomicity

C - consistency

I - Isolation

D - durability

These properties in order to ensure accuracy, completeness and data integrity

Atomicity :- A transaction must be treated as atomic unit that is either all of its operations executed or not.

consistency :- The database must be consistent before and after the transaction.

Isolation :- Multiple transactions occur independently without interference.

Durability :- The change of successful transaction occurs even if the system failures occurs.

6/10/21

Data mining and analysis :-

→ Data mining is a process of extract ~~use~~ useable data from a larger of raw data.

→ The datamining is a subset of data analysis.

→ A datamining an efficient and continuous methods of recognizing & discovering hidden patterns and data throughout a huge data sets.

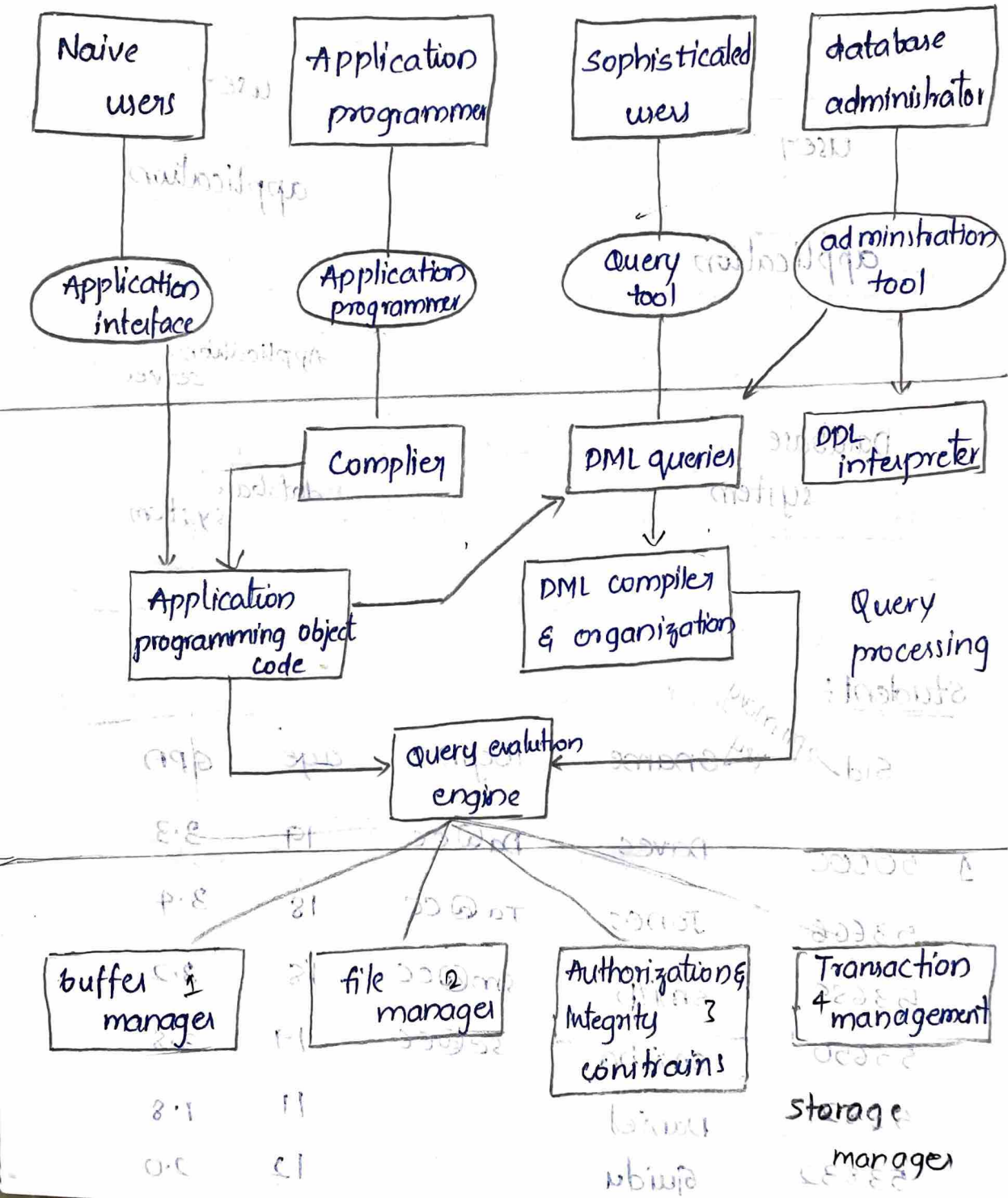
→ It is used to built machine learning models, that are further used in artificial intelligence

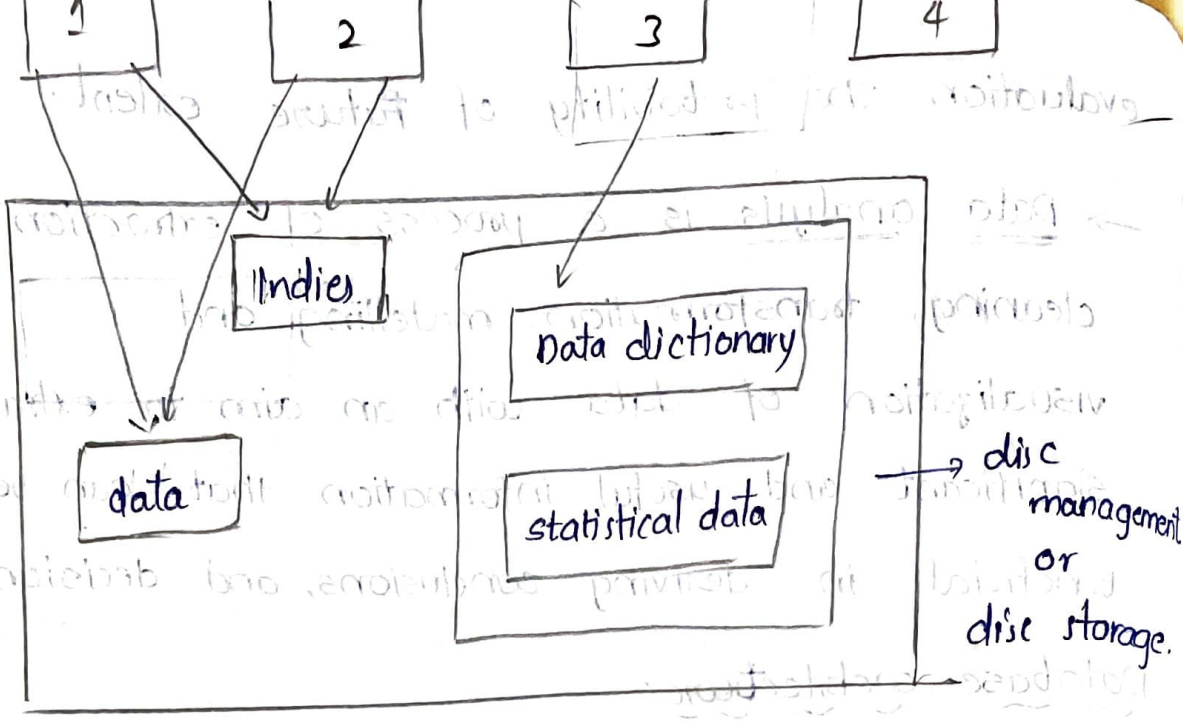
→ It is used sophisticated mathematical algorithms for segmenting that data and

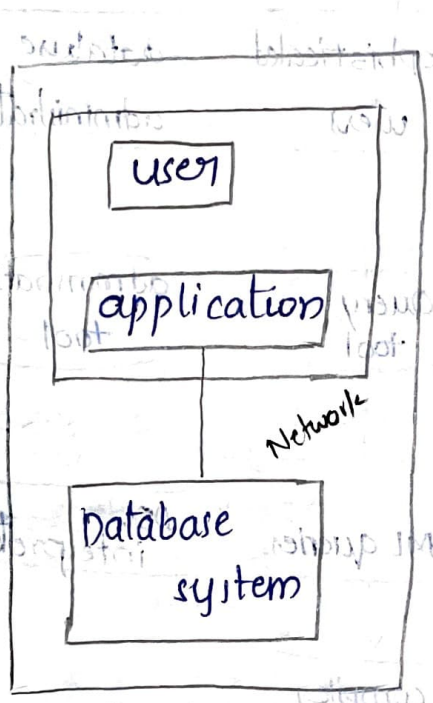
evaluation the probability of future extent.

→ Data analysis is a process of extraction, cleaning, transformation, modelling and visualization of data with an aim to extract significant and useful information that can be beneficial in deriving conclusions, and decisions.

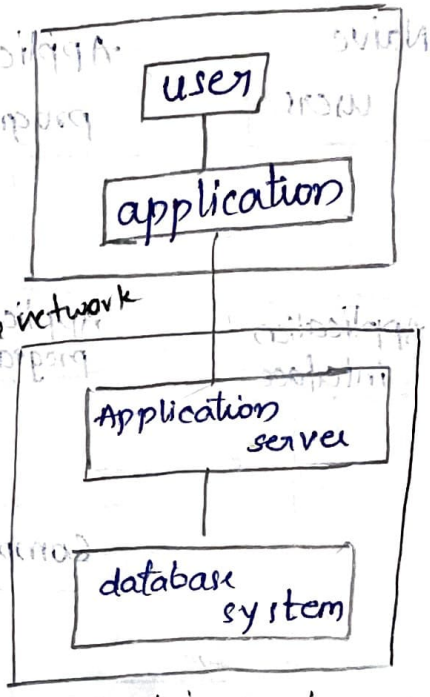
Database architecture :-







Two tier



3 tier

g.mail.com

Student :-

	Sid → primary key	Sname	login	age	GPA
1	50000	Daves	Da@cc	19	3.3
	53666	Jones	Ja@cc	18	3.4
	53688	smith	sm@cc	18	3.2
	53650	smitha	se@cc	19	3.8
	53821	Daniel		11	1.8
	53832	Guidu		12	2.0

DDL:

```
Create table student (sid char(10),
                      sname char(10),
                      login varchar(10),
                      age int(10),
                      GPA varchar(10));
```

```
Insert into student values('50000', 'Dave', 'Da@cc',
                           '19', '3.3');
```

we can,

Delete all student tuples with name = smith

(Records)

Delete from student where sname = smith.

condition

Modification command:-

We can increment the age and decrement the GPA of the student with sid = 53688

→ update student set age = age+1, gpa = gpa-1,

where sid = 53688

update student set s.age = age+1, s.gpa = gpa-1,

where s.side = " ";

Integrity constraints over the relations :- 21/10/21

→ The integrity constrains is a condition specified on database schema and restricts the data that can be stored in an instance of database.

constraint = condition

key constraints :- (unique)

A key constraint is a statement that is a certain minimal subset of fields of a relation is a unique identifier for the tuble (records).

for example :- primary key, total participation

In the student table no two students have same id's

Candidate key :- primary key undochu unda kpovachu

A set of fields that unqiuely identifies a tuble acording to the key constrain is called candidate key. all (may) or no

super key :- (Record is unique)

A super key is a set of single or multiple keys which identify rows in a table.

{sid, name} sid - super key

{sid, login} unique id + other fields

{sid, name, login}

→ Super key is a super set of candidate key.

Key constraints in SQL:-

A subset of the columns of a table composed a key by using unique constraints

- At most one of candidate key can be declared to be a primary key
- The primary key is always unique and not null.
- create table student (sid char(20),
  name char(20),
  login char(20),
  age integer,
  gpa real,
  unique (name, age),
  constraint student key primary key (sid));

Foreign key:-

enrolled table - enrolled referencing relation

Cid	grade	studid → foreign key
Car101	C	53831
Reg 203	B	53832
Top 112	A	53650
Hist 105	B	53666

Student table - student referred relation

primary key

sid	name	login	age	GPA
50000	Dave	d@cs	19	3.3
53666	Jones	d@cs	18	3.4
53688	smith	d@ce	18	3.2
53650	smith	d@math	19	3.8
53831	Madyan	d@music	11	1.8
53832	Guidu	g@music	12	2.0

(primary key)

Information stored in a relation is linked to the information stored in another relation (enrolled table) is called foreign key

```
create table enrolled (studid char(20),
                       cid    char(20),
                       Grade  ~~char(20)~~ integer,
                       primary key (studid, cid)
                       foreign key (studid) reference
                       (student) );
```

Enforcing (rules) integrity constraints :-

Integrity constraints are specified when a relation is created (may created & modified enforced when a relation is modified.

→ The primary key and unique constraints are straight forward

→ If an insertion, deletion, updation commands cause of violation and it is rejected.

for example :-

- Insertion, violation of primary key constraints
- update, the update violates the primary key constraints because there is already a tuble with key constraint

Impact on foreign keys :-

- Deletions of enrolled table (rows) donot violate referential integrity
- But insertion of enrolled tubles effected.
- Insertion of student tuble don't violate and deletion of student table could cause violations.
- SQL provides several alternative ways to handle foreign key violations.

1. What should we do if an enrolled growth is inserted.

(violates) - NO

tuble = rows

2. What should we do if a student row is deleted.

3. What should we do if the primary key values of students row is updated.

A. When student row is deleted, all enrolled rows that are referred to it are to be deleted. but when sid column of a student row is modified this update is to be rejected and if an enrolled row refers to be modified student rows.

```
create table enrolled ( studid char(20),
                        cid  char(20),
                        grade char(20),
                        primary key (studid, cid),
                        foreign key(studid) Reference students
                            ON Delete CASCADE
                            ON ~~violate~~ NOACTION);
                            update
```

The default actioption is

Noaction - is that the action of updations & deletions is to be rejected.

CASCADE - If the students row is deleted, All enrolled rows that refers to it are to be deleted as well.

→ If the update class, the sid column of students rows is updated. This updation is also carried out in each row that refers to updated student rows.

→ SQL also allows use of NULL, as the default value by specifying "ON delete set NULL"

25/10/21

Query relationship data:-

query:- Find the all students younger than 18.

A relational database query is a question about the data and the answer consist of new relation, containing the results

```
select  *                    All students → *
from    student S
where   S.age < 18;
```

fields

Sid	name	login	age	GPA
53631	madyan	d@music	11	1.8
53632	Guidu	g@music	12	2.0

} Records

Here, the symbol of * is known as we retain all fields of selected tables/records/ values in the results.

The symbol of s is a variable that takes on the value of each tuple in the student table.

The where class condition specifies that we want to select only tables of the table or relation.

2] Find the all the students enrolled Req.203

```
select *
from student s, enrolled e
where s.sid = e.studid AND
      e.cid = Req203;
```

53832	Guidu	q@music	12	2.0

3] Find the name and login of students who are younger than 18.

```
select  *s.name, s.login
from    student s;
where   s.age < 18 ;
```

name	login
Madyan	d@mwic
Guidu	g@mwic

4] Find the name and cid of all students who have obtained 'A' Grade

```
select  s.name, e.cid
from    student s, enrolled e
where   e.grade = A and s.sid = e.studid ;
```

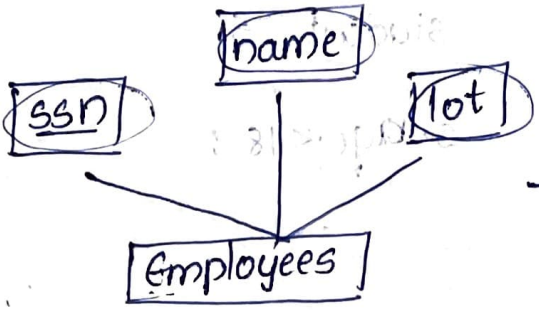
Name	cid
Smith	Top112

Logical database design:-

How to translate an E-R model into a collection of tables with associated

1. Entity sets to tables :-

An entity set is mapped to a relation and each attribute of the entity set become an attribute of the table.



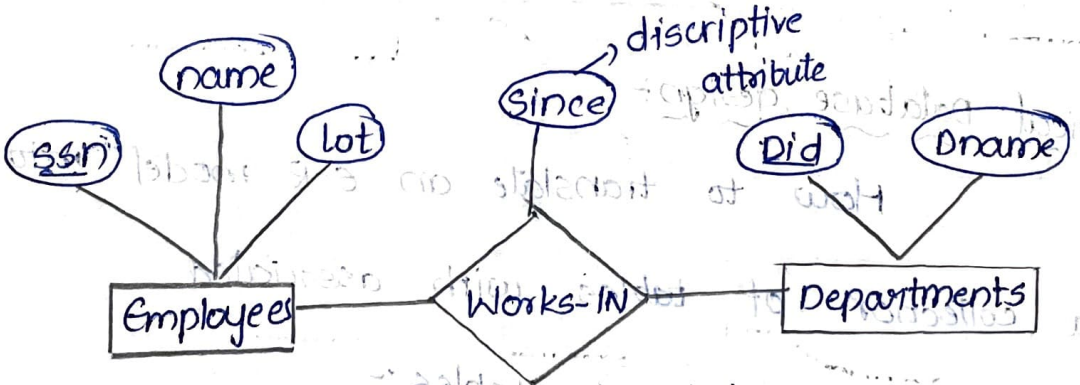
→ Entity set of employees

ssn	name	lot
123-22-3666	Attishoo	48
231-31-5368	smiley	22
131-24-3650	Smithrust	35

→ Relation of employees

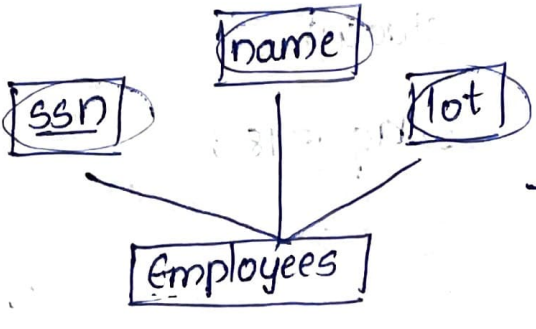
Create table employee (ssn varchar(20),
name char(20),
lot varchar(20),
primary key (ssn));

E-R model :-



Binary relationship

Create table works-in (ssn char(10),
Did char(10),
since datechar(100),
primary key (ssn, Did),
foreign key (ssn) reference employee



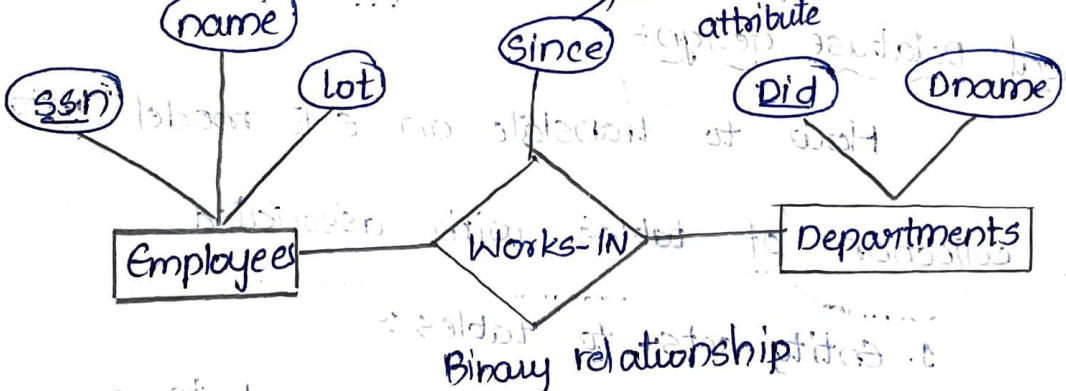
→ Entity set of employees

ssn	name	lot
123-22-3666	Attishoo	48
231-31-5368	smiley	22
131-24-3650	Smithrust	35

→ Relation of employees

```
Create table employee (ssn varchar(20),
                       name char(20),
                       lot varchar(20),
                       primary key (ssn));
```

E-R model :-



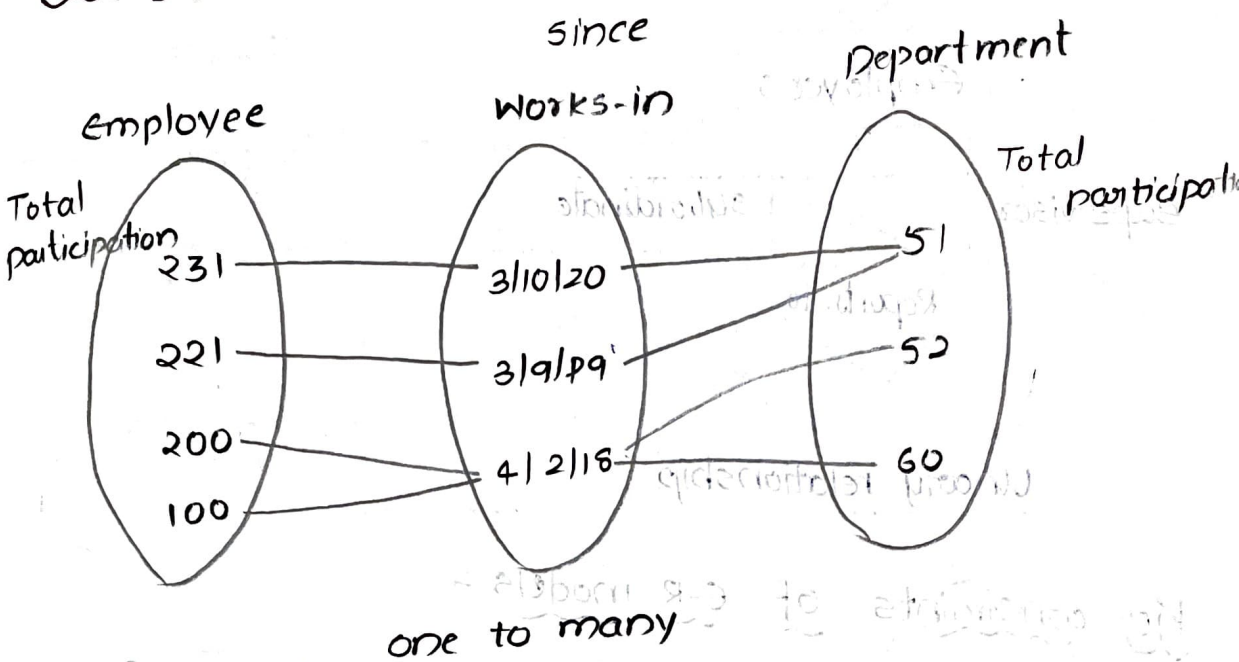
Binary relationship

```
Create table works-in (ssn char(10),
                       Did char(10),
                       since datechar(10),
                       primary key (ssn, Did),
                       foreign key (ssn) reference employee
```

foriegn key (Did) reference departments
);

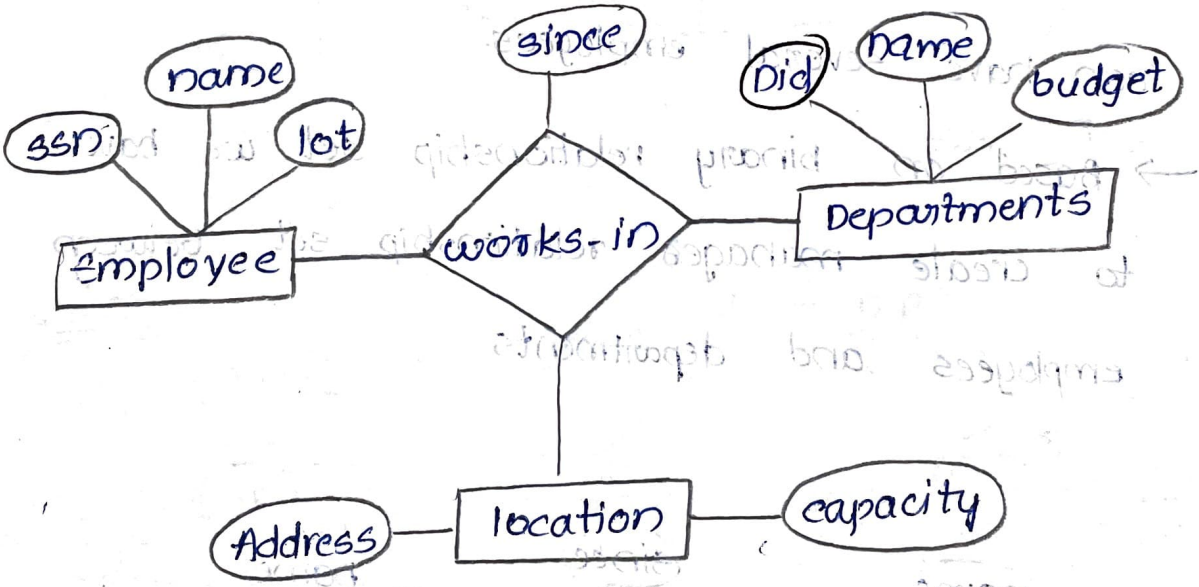
20/10/21

Relationship and relationship sets :-

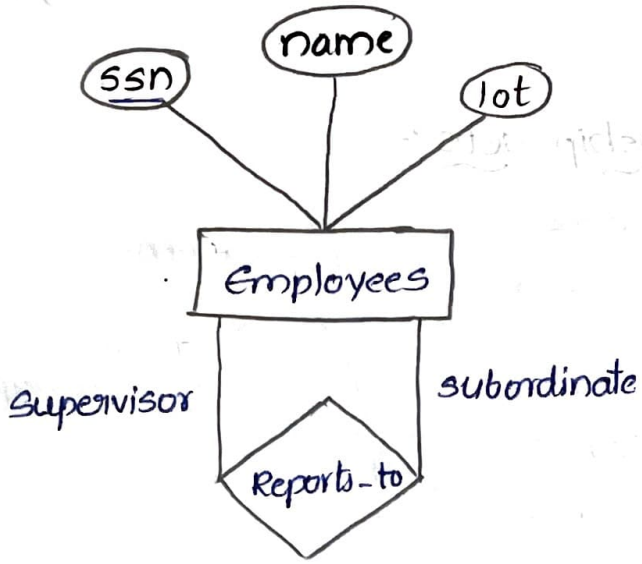


one to many

Instance of relationships



Ternary relationship

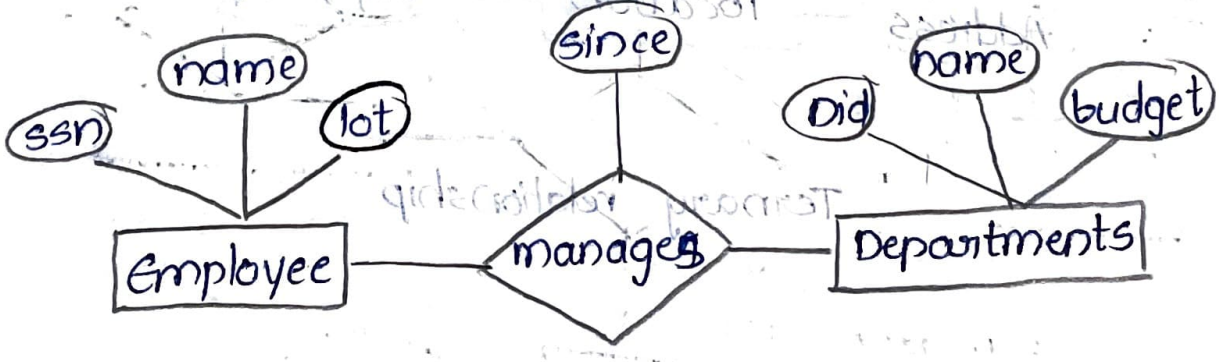


Unary relationship

Key constraints of E-R models :-

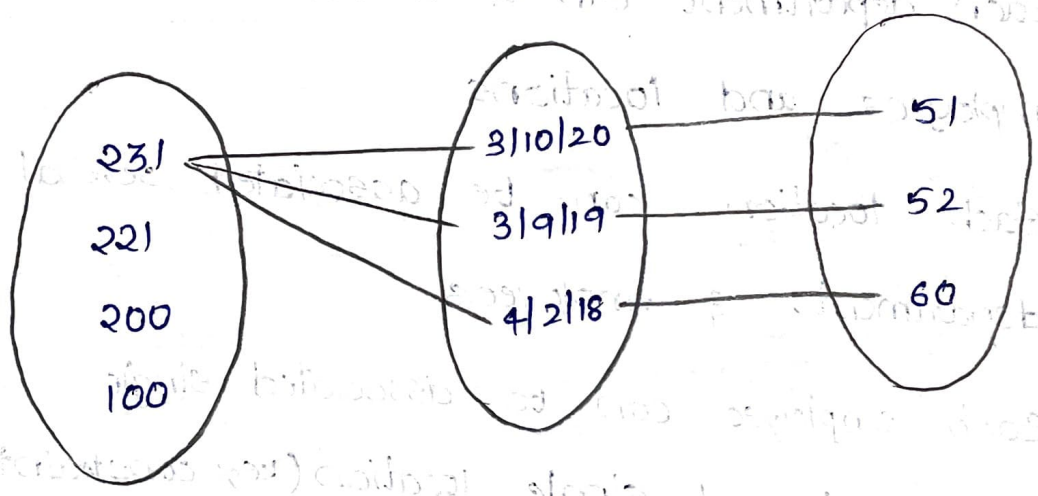
→ In works-in relationship an employee can works-in several departments and department can have several employees.

→ Based on binary relationship set we have to create manages relationship set between employees and departments



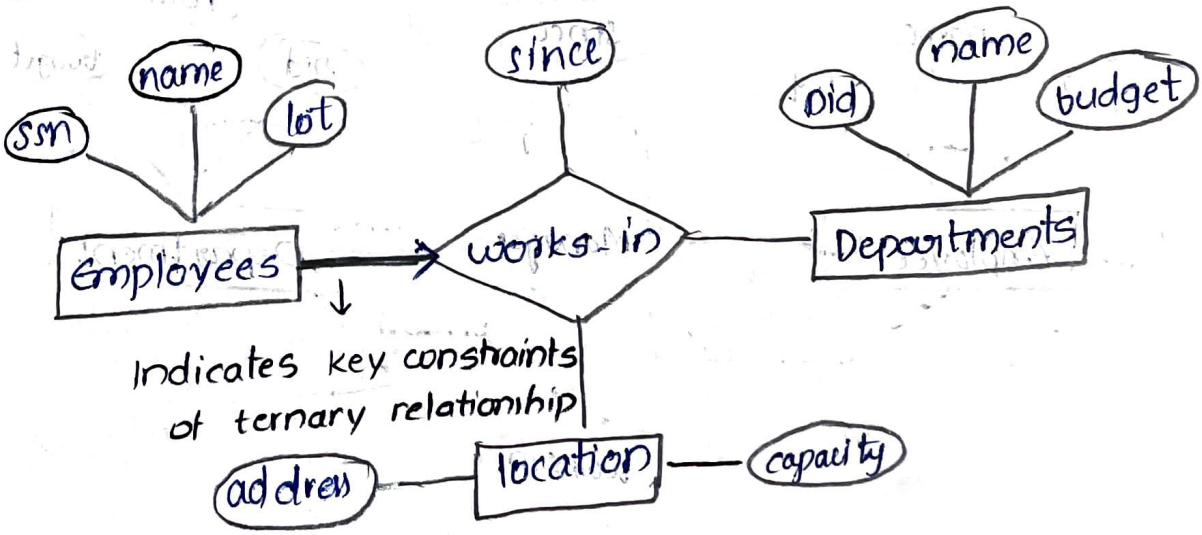
→ In the relationship set, each department atmost one manager and single employee is allowed to manage more than one department.

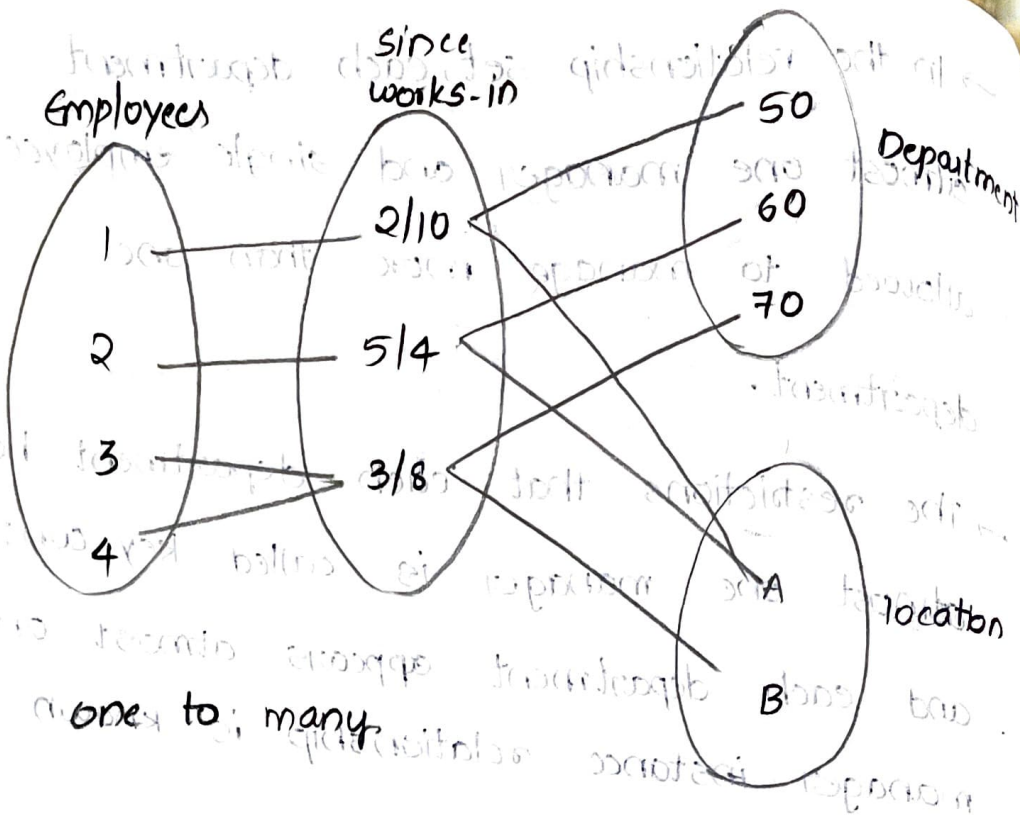
→ The restrictions that each department has atmost one manager is called key constraint
- and each department appears atmost one manager instance relationship is known as one to many.



27/10/21

Key constraints on ternary relationship :-





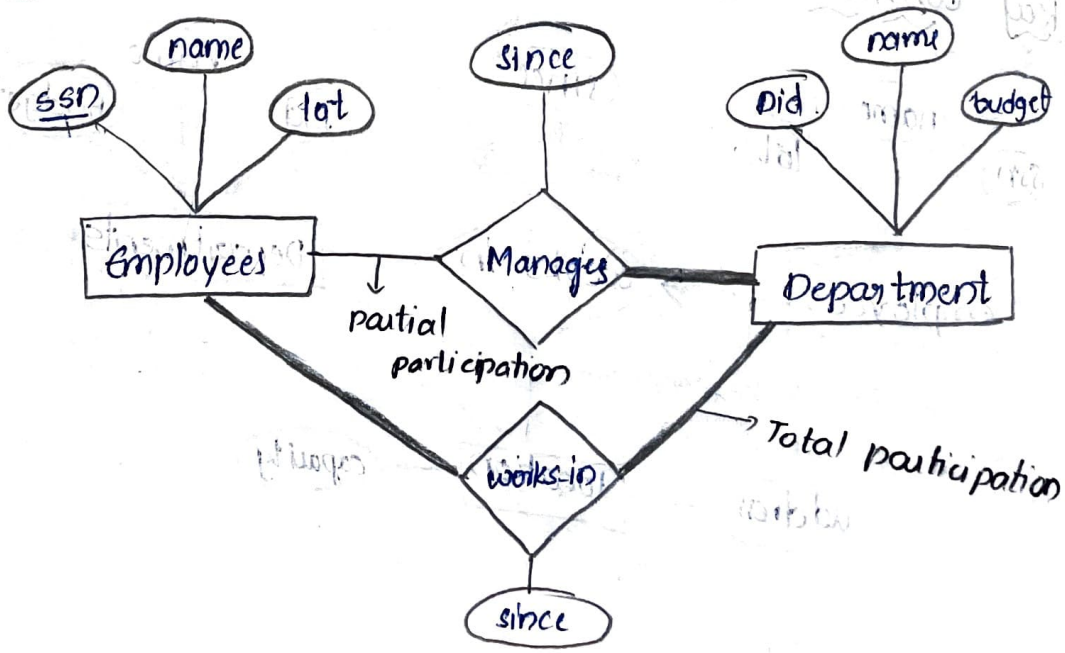
→ Each department can be associated several employees and locations

→ Each location can be associated several departments & employees.

**

→ Each employee can be associated single department and single location (key constraint)

Participation constraints :-

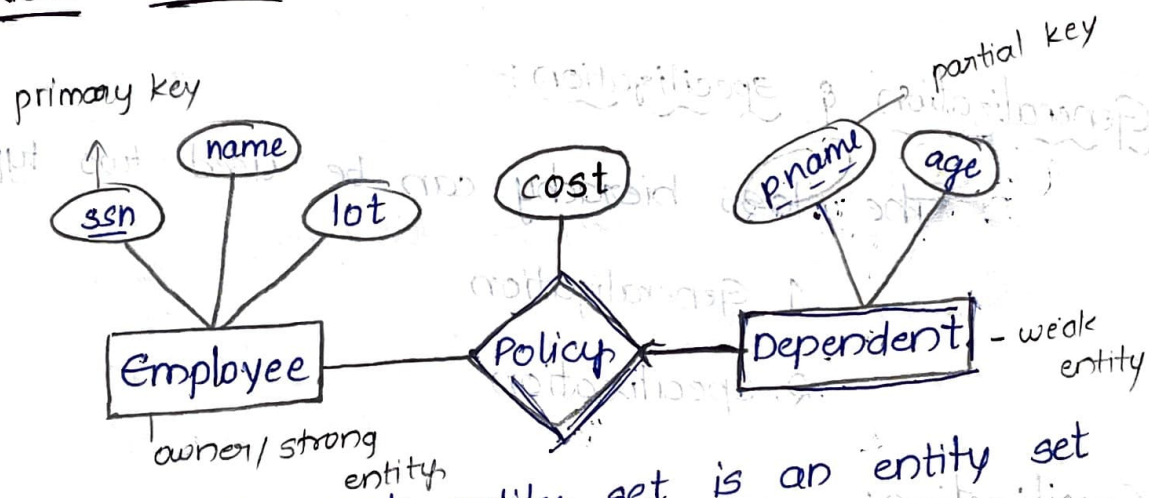


→ Every department is required to have a manager (key constraint)

→ The participation of entity set departments in the relationship set manages is said to be called total participation.

→ The participation of entity set employee in manges is partial participation, since not every employee gets to manages a department.

Weak entities :-



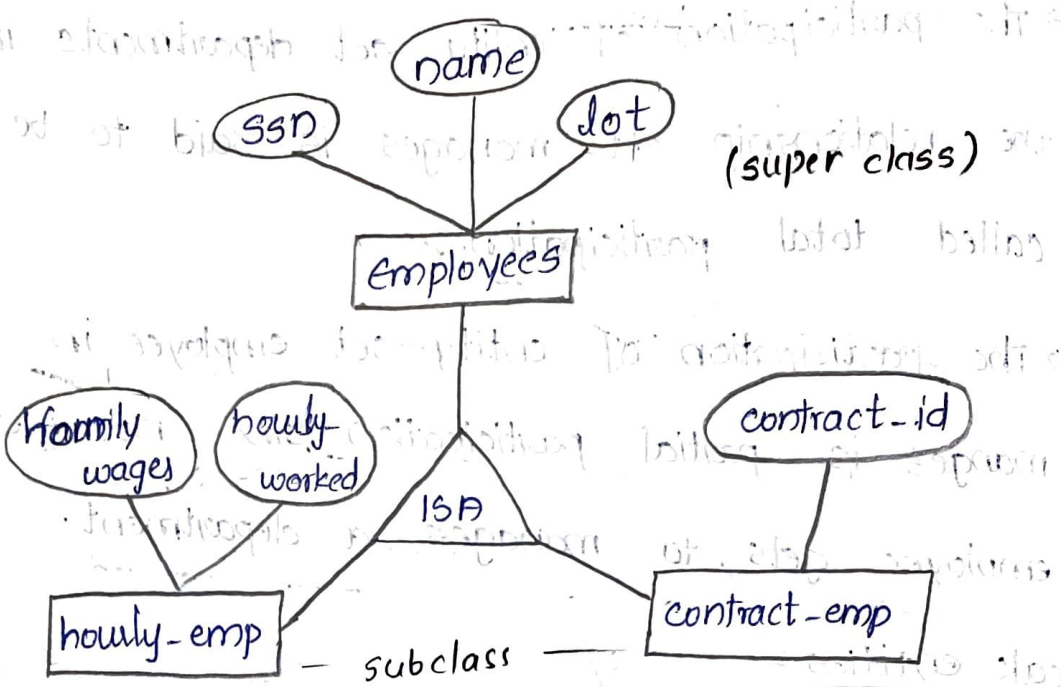
- A weak entity set is an entity set that doesn't contain sufficient attributes to uniquely identify its entities
- The primary doesn't exist weak entity set is known as partial key.

Identifying relationship in weak entity :-

- Strong entity and weak entity sets are associated one to many called identifying (employee) (policies) (dependent) relations

Class hierachy:-

- Entity sets into sub classes is called class hierachy.



Generalization & specilization:-

The class hierachy can be used two types:

1. Generalization
2. Specilization

Specilization:- (top to bottom)

The process of identifying subsets of entity set. (Identify super class)

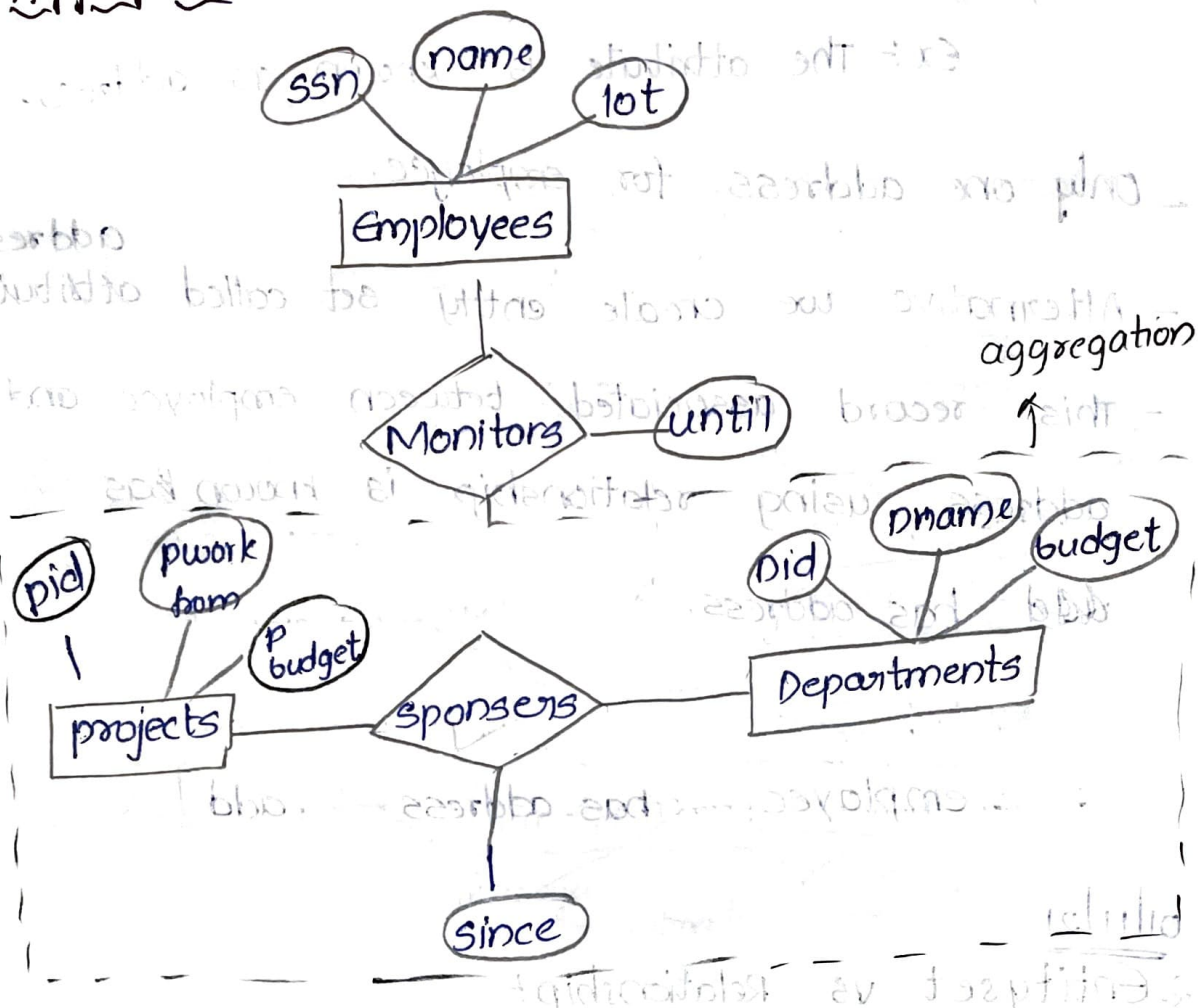
→ The employee is specilized into sub classes

Generalization:- (bottom to top)

- Hourly employees and contract employees are generalized by employees.

Aggregation:-

30/10/21



Aggregation allows to indicates that a relationship set is participate to another relationship set

conceptual design with E-R models:-

1. Entity vs attribute
2. Entity set vs relationship
3. Binary vs Ternary relationship
4. Aggregation vs ternary relationship

1. Entity vs attribute:- Identifying the attributes of an entity set, it is sometimes not clear whether a property should be modelled as an attribute or

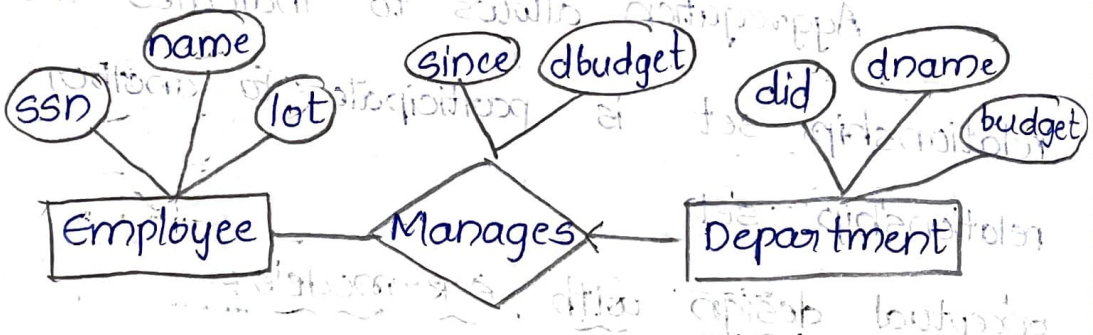
entity set.

Ex:- The attribute is known as address.

- Only one address for employee.
- Alternative we create entity set called address attributes
- This record associated between employee and address, using relationship is known as has_address.

01/11/21

2. Entity set vs Relationship:-

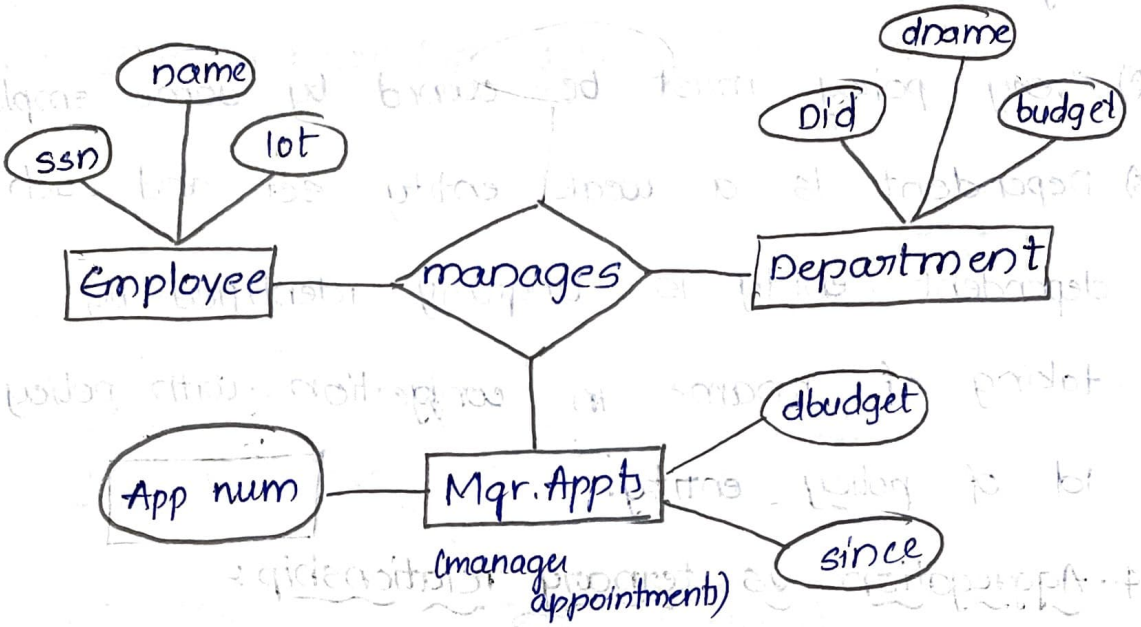


→ Relationship set is called manages, suppose each department manager is given at the dbudget (discreationary budget). Here atmost one employee managing a (more than one) department, but a given employee could manages different departments.

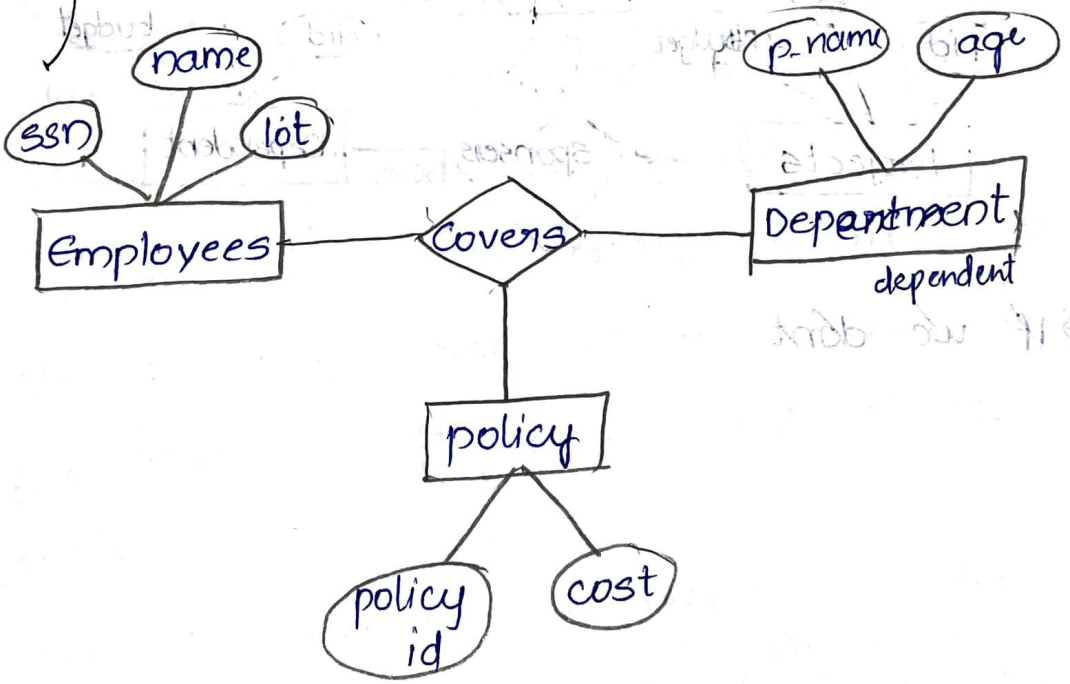
→ The pair of manages and department stores since and dbudget.

→ Here manages has seperated dbudget of department.

→ The same value of dbudget field is converted into one entity set.



3. Binary vs Ternary relationship :-



→ Here, each employee can own several policies, Each policies can own several employees.

→ Each dependent can cover many policies.

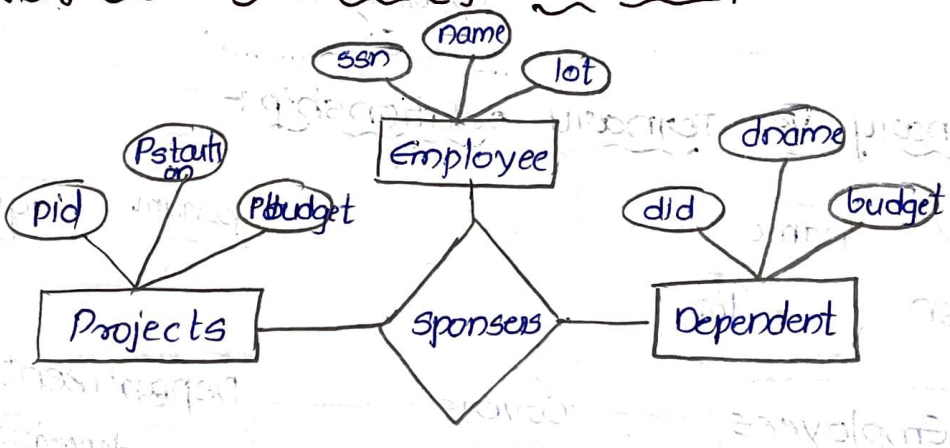
→ Suppose, we have many following additional requirements

1) A policy cannot be own jointly by two or more employees, why because it is not a covered key constraint

2) Every policy must be owned by some employees

3) Dependent is a weak entity set and each dependent entity is unqiuely identified by taking pname in conjgestion with policy Id of policy entity.

4. Aggregation vs ternary relationship :-

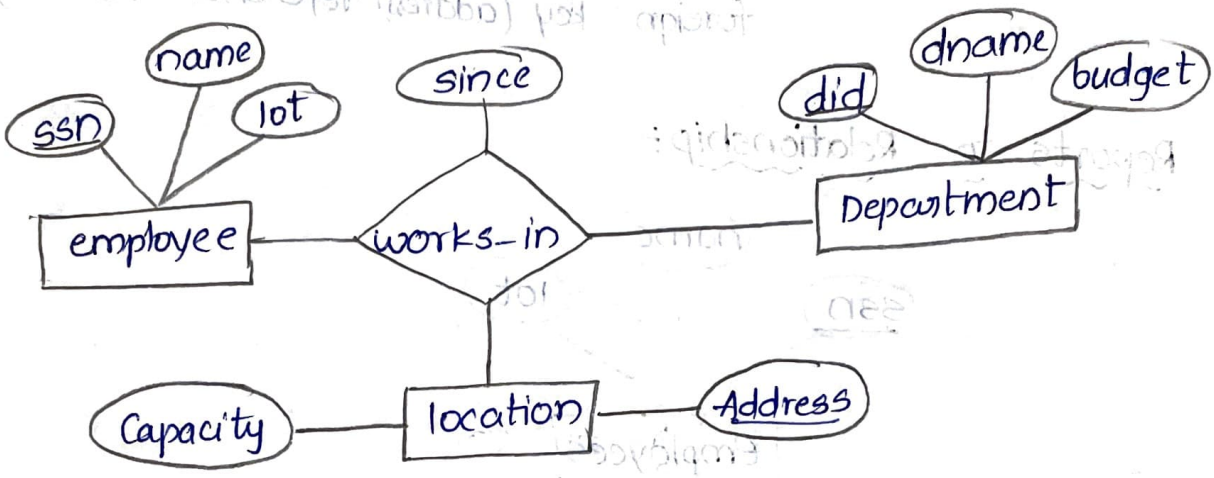


→ If we dont

→ If we dont need to record the untill attribute of monitors then we might reasonable use a ternary relationship says sponsers.

2/11/21

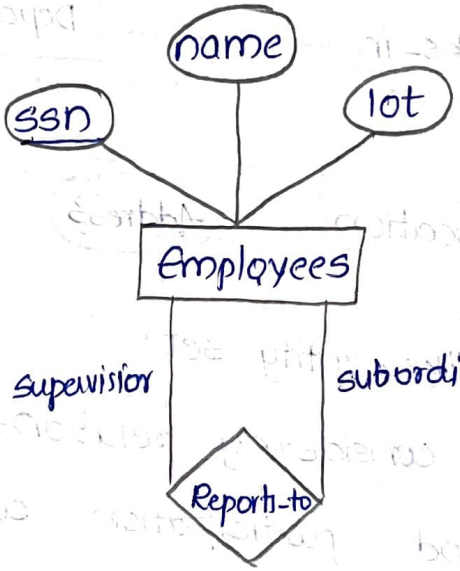
Translating relationship set without key constraints: (total participation ↑)



- Relationship set like a entity set
- We begin by considering relationship set without key and participation constraints.
- To represent a relationship we must be able to identify each participation entity and give value to discriptive attribute of the relationship.
- The alternative of the relation includes, the primary key attribute of each participating entity sets. has foreign key fields and this discriptive attribute ↓ of this relationship.

```
Create table works_in ( ssn varchar(10),
                        Did char(10),
                        since date, address char(10),
                        primary key (ssn, Did, address)
                        foreign key (ssn) references employees,
                        foreign key (Did) references department,
                        foreign key (address) references location);
```

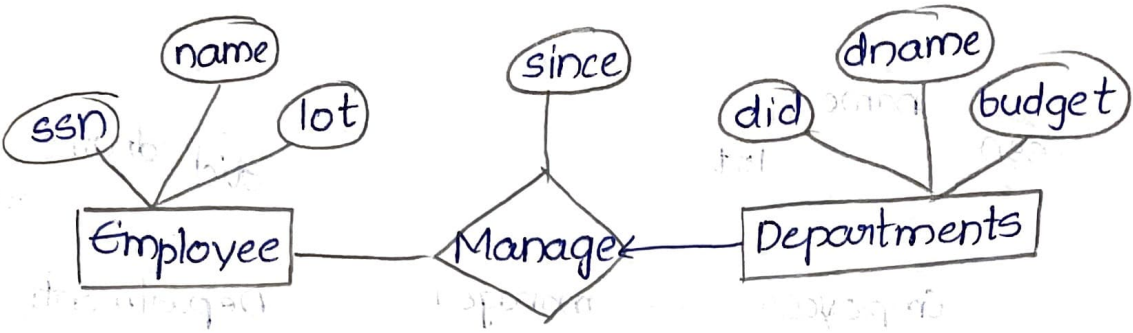
Reports_to Relationship :-




```
Create table Reports_to ( subordinate_ssn varchar(10),
                          supervisior_ssn varchar(10),
                          primary key (supervisior ssn, subordinate
                          ssn)
                          Foreign key (supervision ssn) references
                          employee(ssn)
                          Foreign key (subordinate) references
                          employee(ssn))
```

3/11/21

Relationship set with key constraints :-



Here n entity sets and some m of them are linked via arrows in the E-R model

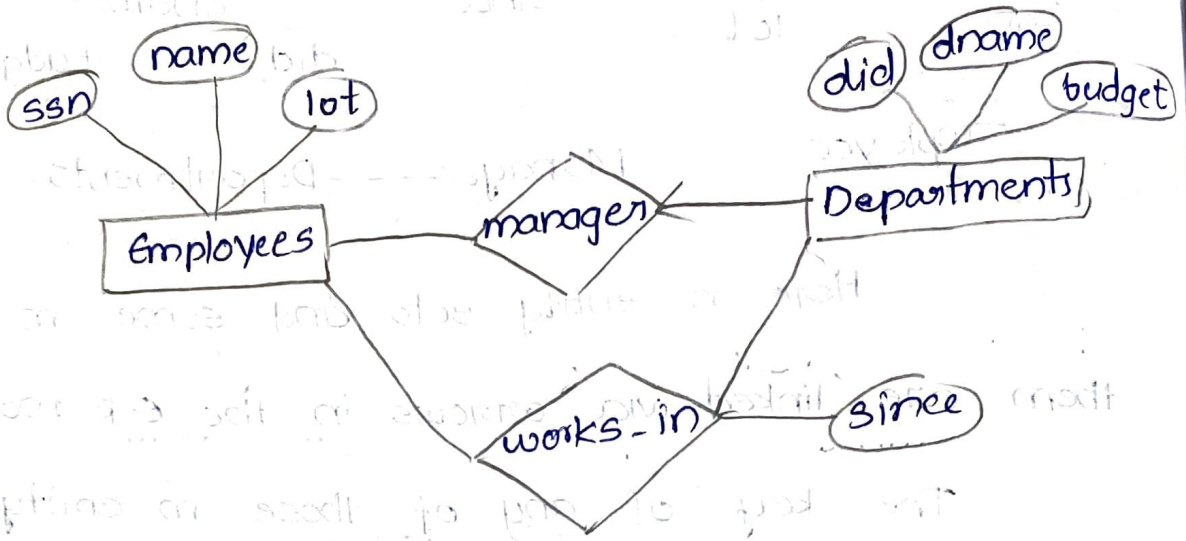
The key of any of those m entity sets of candidate keys and one of those should be a primary key.

The manages attributes are since, did & ssn because each department as atmost manager, no two tuples, can give the same did value but differ on ssn values.

Query :-

```
mysql > create table manager(ssn varchar(10),
                             Did varchar(10), since date,
                             primary key (Did),
                             foreign key(ssn) references employee,
                             foreign key (Did) references departments);
```

Translating relationship sets with participation constraints :



Query :

```
mysql > create table department_manager
        (since date, did varchar(10), dname char(10),
        budget int(20), ssn varchar(10),
        primary key (did) foreign key (ssn),
                references employees,
        ON DELETE NO ACTION);
```

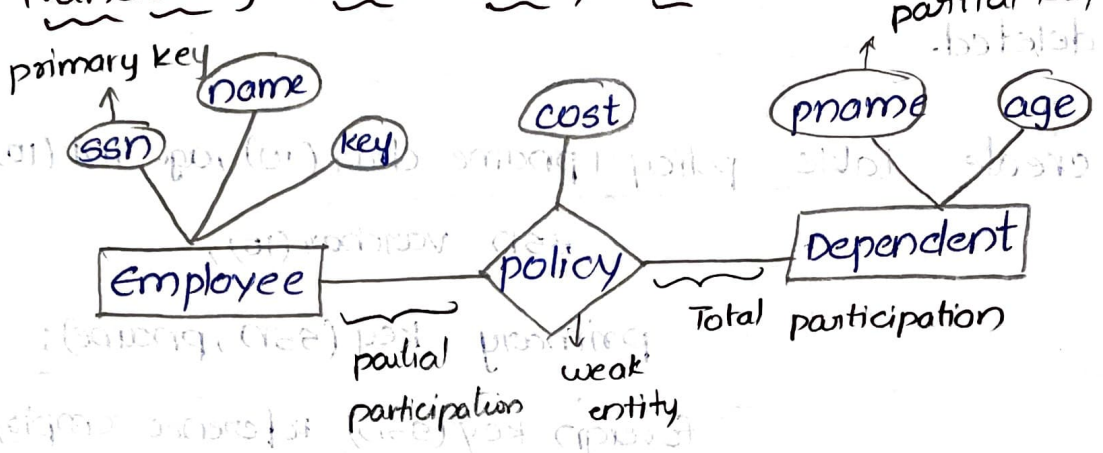
→ The participation constraints every department must have a manager because ssn cannot take on NULL values.

→ ON DELETE NO ACTION means employee tuple cannot delete while it is pointed to by a department manager tuple

→ If we wish to delete such an employee tuple we must first change the department manager tuple to have a new employee as a manager

8/11/21

Translating weak entity set :



create table policy (pname char(10), age int(10),
ssn varchar(10),
primary key (ssn)
Foreign key (ssn) reference employee,
ON Delete CASCADE);

→ A weak entity set always one to many relationship and has a key constraints total participations.

→ The weak entity has only a partial key and also owner entity (employee) is deleted we won't allow owned weak entity to be deleted.

→ A dependent entity can be identified uniquely only it will take the key of owning employee entity and the pname of dependent, and dependent entity must be deleted if owning employee entity is deleted.

```
create table policy (pname char(10), age int(10)
                     ssn varchar(10),
                     primary key (ssn, pname):
                     foreign key (ssn) reference employee
                     ON delete CASCADE):
```