ER Model to Relational Model

- ER diagram to Relational Model.

① Strong Entity :-

(a) Becomes an individual table with entity name.
(b) Entity's Primary Key (PK) is used as Relation's PK.
(c) FK are added to establish relationships.

② Weak entity :-

(a) A table is formed with all attributes of the entity.
(b) PK of its corresponding strong Entity will be added as FK.
(c) PK of the relation will be a composite PK, & FK.

③ Single value attribute

(a) Represented as columns directly in the tables / relations.

④ Composite attributes

(a) Handled by creating a separate attribute itself in the original relation for each composite attribute.
e.g.:- Address : {Street-name, House-no}.

(5) Multivalued attributes.

(a) New tables (named as original attribute name) are created for each multivalued attribute.

(b) PK of the entity is used as column FK in the new table.

(c) e.g. ⇒ Employee, dependent-Name is a multivalued attribute.

- New table named "dependent-Name" will be formed with columns emp-id, and dname.
- PK : { emp-id, name }
  FK : { emp-id }

(6) Derived attributes :- Not consider in the table.

(7) Generalisation

- There are two ways to do so :-

(1) We make table of generalisation and its lower-level entity set both. We store the common attributes in generalisation table.

e.g. ⇒ Banking System generalisation of account - Saving/Current

T1 :- account (account-number, balance)

T2 :- Savings-account (account-number, interest-rate, daily-withdrawal-limit).

T3 :- Current-account (account-number, overdraft-amount, per-transaction-charges).

- Here we use account-number in all tables as primary key.

(2) Another method is that we remove the generalisation table and make only lower-level entity set table.

Eg.

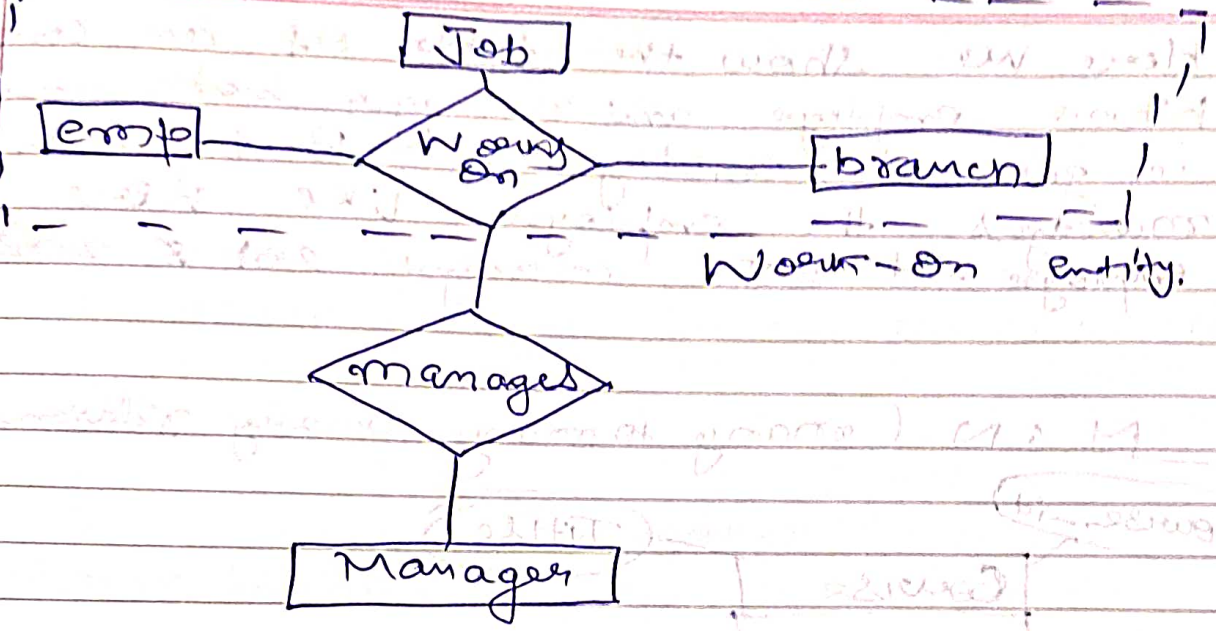
T1 : Savings-account (account-number, balance, ---)

T2 : Current-account (account-number, balance, ---)

- Here we are storing balance in both tables, which is a redundancy.
- Also if some accounts were neither savings nor current accounts - then such accounts cannot be represented with the second method.

(3) Aggregation :-

- let's take an example where we have 3 entities and we made it together to form an entity which is manages by an entity.



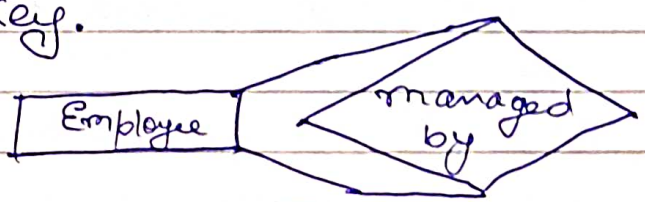
- To make an table of 'aggregation', we make table of relation "manages" and the attributes are the primary key of the other entities and the primary key of manages table is a composite key

- Table:- manages (manager_id, emp_id, job_id, branch_id).

→ Unary relationship

- We will add another attribute in Employee table which will be Foreign Key.

- 1:1 (relationship)

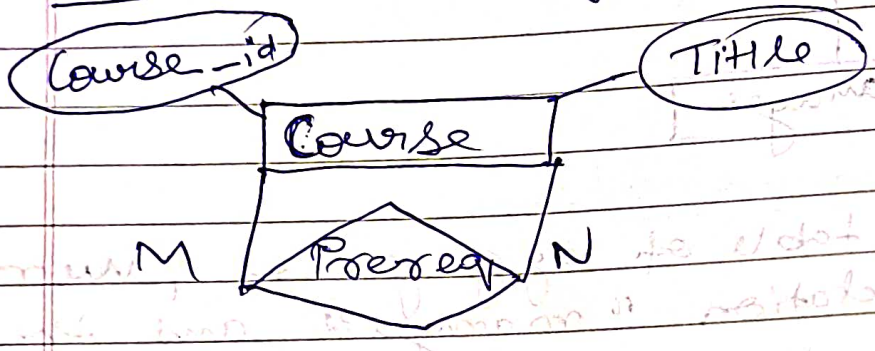


emp_id	name	joining date	emp_mgr_id
201	- -	- - - -	205
202	- - -	- - - -	205
205	- - -	- - - -	NULL

F.k.

- Here we show the table of an company. Where employe and manager both are in general : employees. But manager manages the employee. like here employee -id 205 manages 201 & 202.

- M : N (many to many unary relationship).



(1) Course (Course_id, Title)

(2) Prereq (course_id, Prereq_course_id)
(F.K)

- Here both attributes are foreign key.