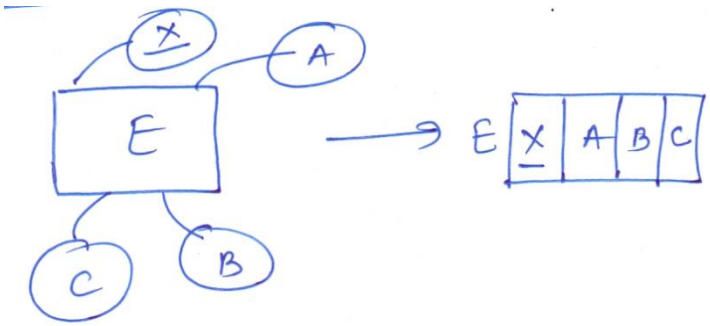
06. Entity Relationship Modeling

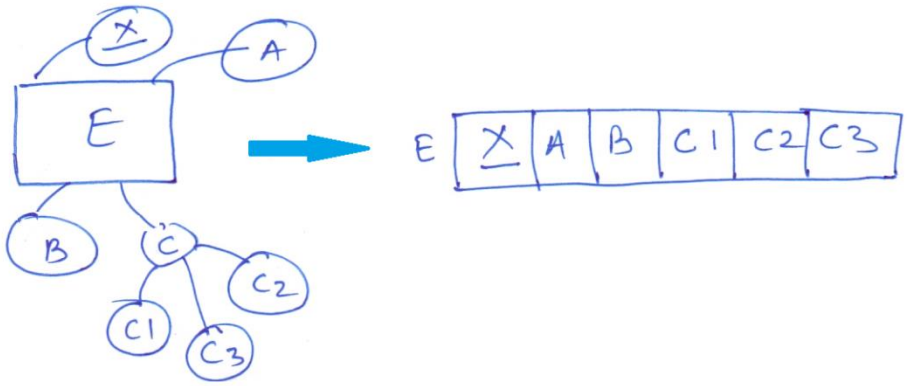
ER to Relational Mapping

- At the end of the day ER models needs to be converted to implementation model like relational. Here we sketch out rule that can be straight forward be used for converting a ERD diagram into relations.

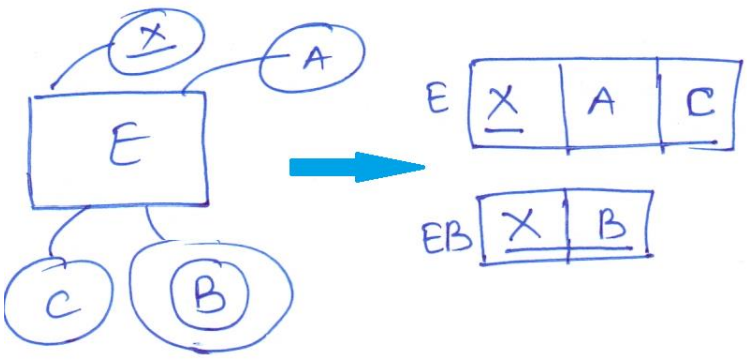
Strong Entity Type



- A strong entity is converted to a relation. Name of relation can be same as entity; key of entity can be key of relation. All attributes (except multi-value and composite, for these separate rules apply) of entity become attributes of the corresponding relation.
- For a composite attribute of an entity type is represented directly by its sub-attributes; example below. For example in Entity E has composite attribute C with sub-attributes C1, C2, and C3; we have three attribute C1, C2, and C3 for C.

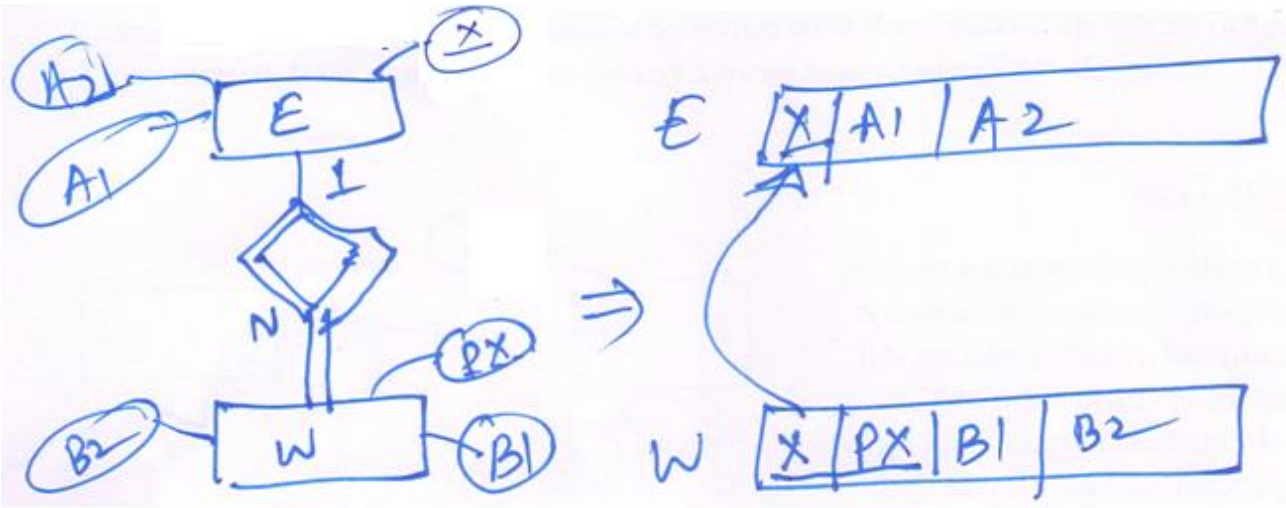


- For multi-value attributes, we create separate relation as following. We have separate relation for each multi-value attribute. Key of such a relation is UNION of key of the key and multi-value attribute. Key of entity act as FK referring to main entity relation. For example we have entity E having multi-value attribute B, we have relation EB, and Key of EB is XB; X is FK referring to E.



Weak Entity types

- We have a separate relation for each weak entity type. Key of such relation is union of partial key of weak entity and key of owner entity. Key of owner entity acts as FK referring to owner entity. For example:



Relationships

- In relational model relationships are represented using foreign keys,

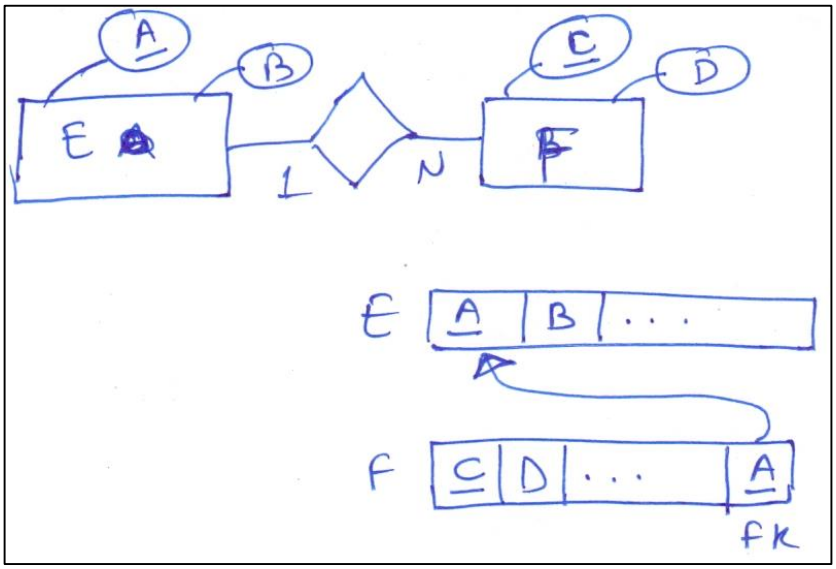
Relationship - One to One (1:1)

- Two relations for both the entities as per the rules talked about. For relationship, we place key of one relation as foreign key in either of relation. Preferable side is entity having total participation.

  For participation, we can set NOT NULL to the FK. But that captures half of the constraint? Having specified NOT NULL for FK only implies total participation of entity corresponds to relation having FK. Unfortunately we can capture total participation of entity corresponding to other relation.

Relationship - One to Many (1:N)

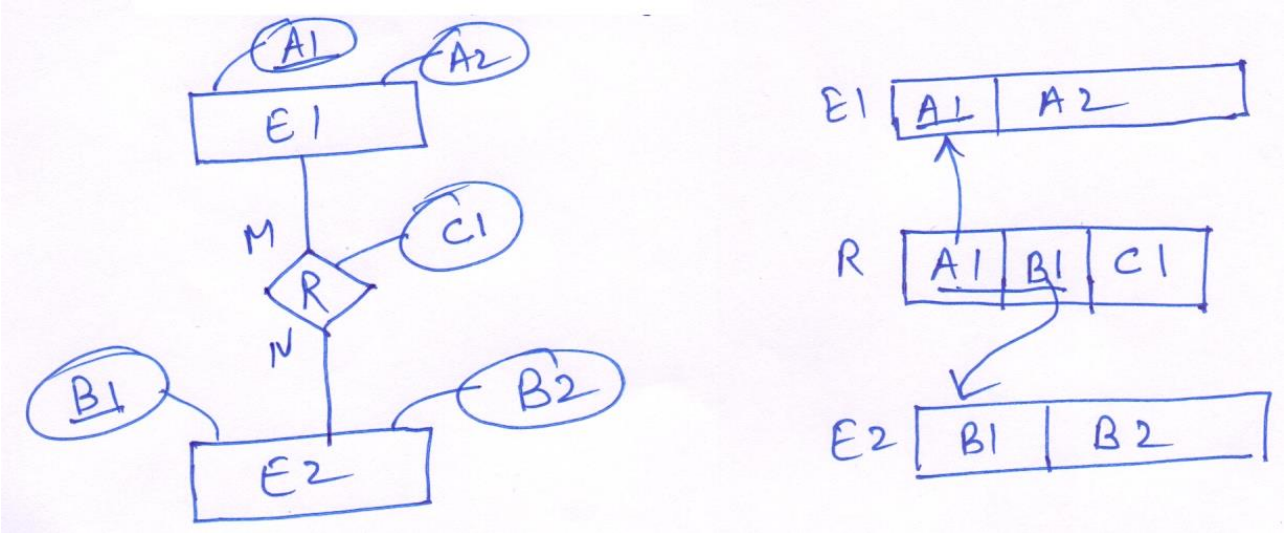
- Two relations for both the entities. We place key of one side entity as foreign key in relation corresponding to N side entity. Example below-
- For Total participation of N side entity, we can have NOT NULL constraint for FK. We cannot capture total participation of entity at one side.



- If there is an attribute of the relationship then that attribute is also mover to relation corresponding to many side.

Relationship - Many to Many (M:N)

- For Many to Many relationships, we create separate relation. Name of the relation can be same as relationship type. If there are attributes of relationship, they are also placed in new relation. Key of relation corresponding to relationship is keys of both participating entities. Both keys are FK referring to their respective relations. We cannot capture participation constraints of entities in case of many to many relationships.

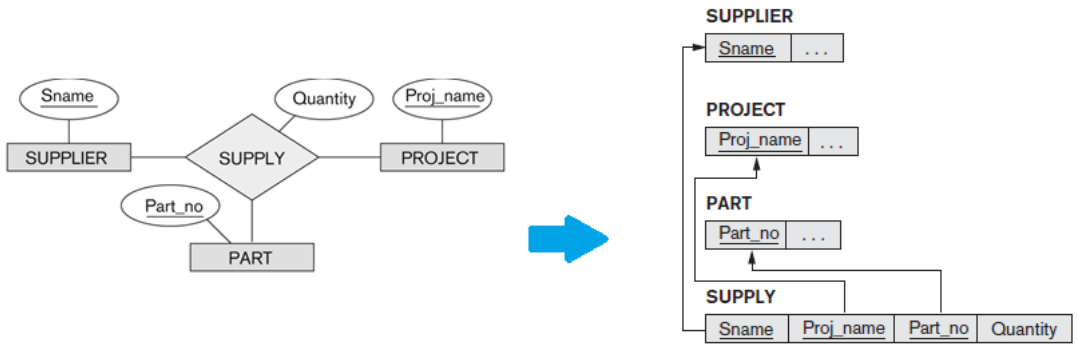


Recursive Relationship

Same as any other binary; for 1:N type cardinality, we have a FK, interpretation of FK is as following – the tuple is treated as being in N side, it refers to entity's PK in 1-Side. For example in Supervises relationship tuple is treated employee as supervisee, and value in it refers to its supervisor.

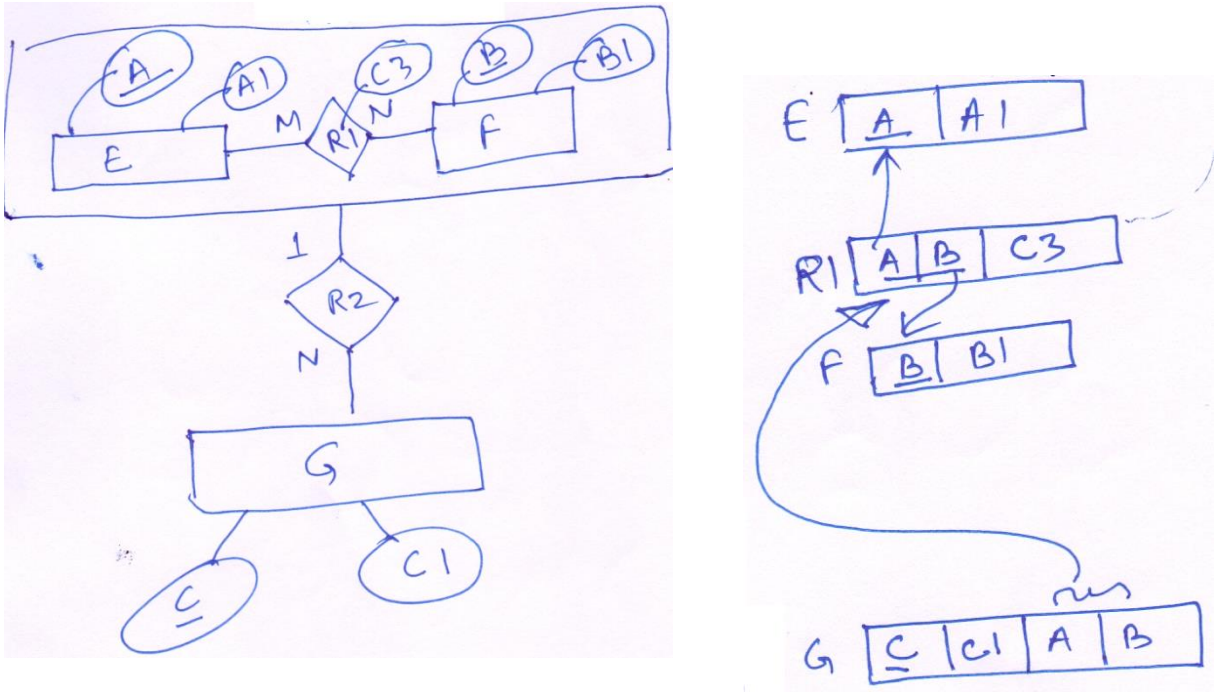
Ternary Relationship

- For ternary relationship too, we create separate relation. Name of the relation can be same as relationship type. If there are attributes of relationship, they are also placed in new relation. Key of relation corresponding to relationship is keys of all three participating entities. All three keys are FK referring to their respective relations. We cannot capture participation constraints of entities in case of ternary relationships. Example from Elmasri/Navathe -



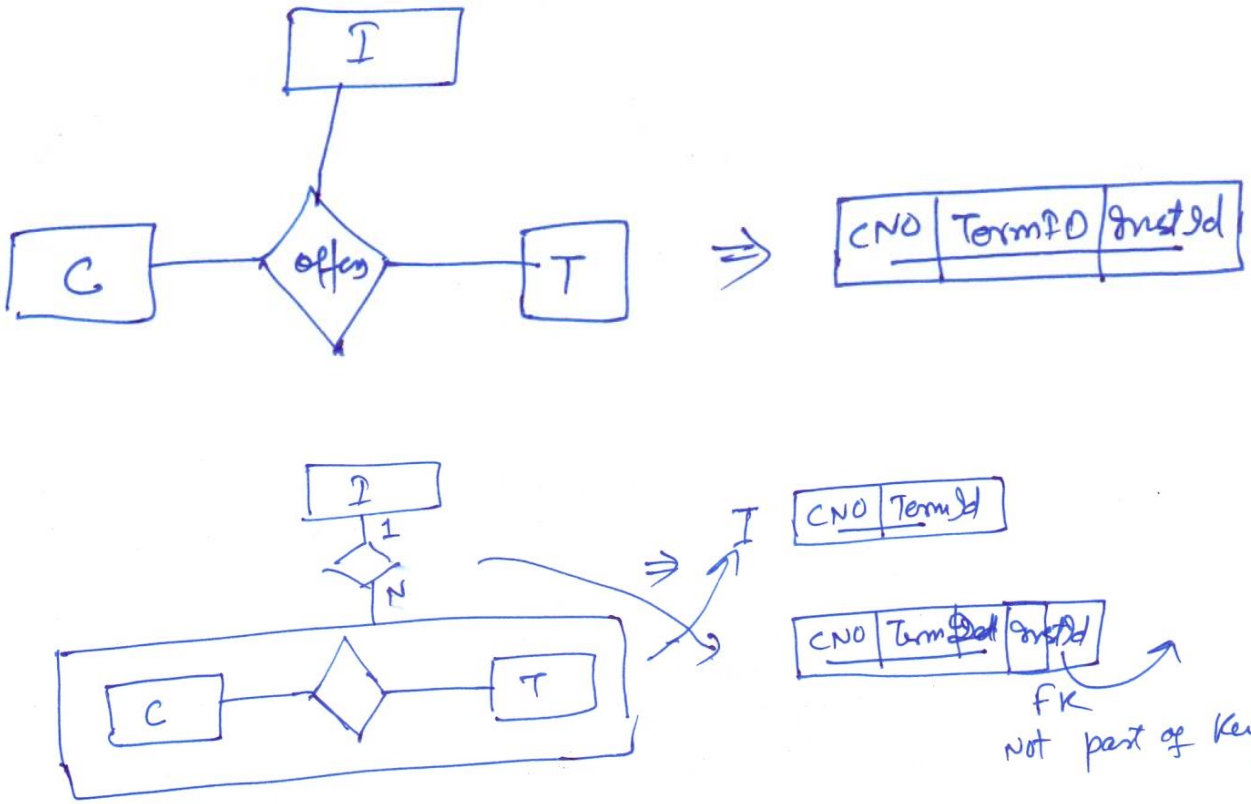
Aggregation

- Aggregation is basically binary only. All binary rules for entity and relationships are applied and corresponding relations are derived. Diagram below depicts an example-



Examples from DA-Acad scenario

- Consider example of Registers as aggregation, we have following relations. Note that resultant relation from aggregation is more accurate than the one resulting from ternary.



Summary: ER to Relational Mapping

ER Model	**Relational Model**
Entity type	Relation
Simple Attribute	Attribute
Key attribute	Key
Composite attribute	Set of simple attributes
Multi-valued attributes	Separate Relation and FK
1:1 relationship	FK in total participation side , either side otherwise
1:1 or 1:N relationship type	FK in Total participation/ FK in N-side relation
M:N relationship type	“Relationship” maps to Relation and two FKs referencing to respective entity relations.
N-ary relationship type	“Relationship” relation and N FKs
Aggregation	Like any other binary relationship is converted