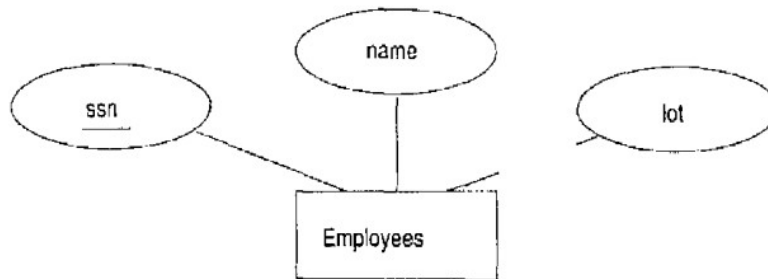


Entity Sets to Tables

An entity set is mapped to a relation in a straightforward way: Each attribute of the entity set becomes an attribute of the table. Note that we know both the domain of each attribute and the (primary) key of an entity set. Consider the Employees entity set with attributes *ssn*, *name*, and *lot* shown in Figure. A possible instance of the Employees entity set, containing three



The Employees Entity Set

Employees entities, is shown in Figure 3.9 in a tabular format.

<i>ssn</i>	<i>name</i>	<i>lot</i>
123-22-3666	Attishoo	48
231-31-5368	Smiley	22
131-24-3650	Smethurst	35

An Instance of the Employees Entity Set

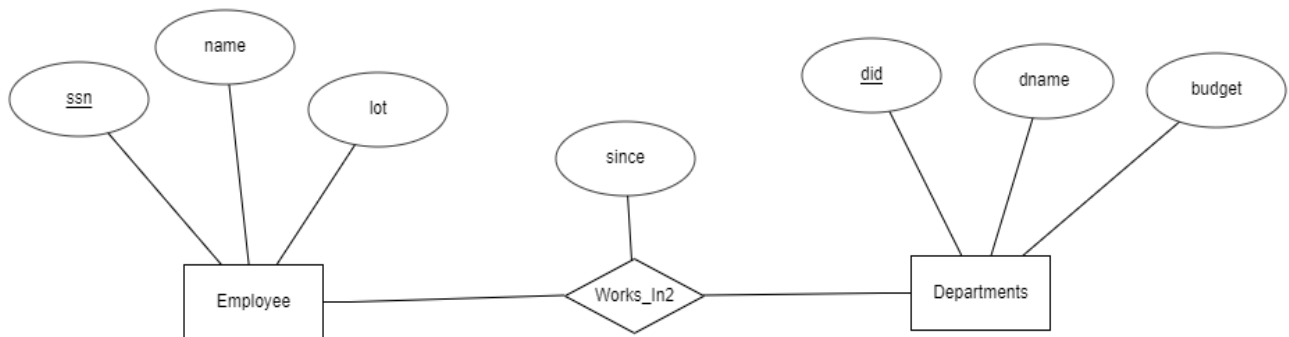
- We just make the **primary key** of the table same as the key of the entity.

Relationship Sets (without Constraints) to Tables

A relationship set, like an entity set, is mapped to a relation in the relational model. We begin by considering relationship sets without key and participation constraints, and we discuss how to handle such constraints in subsequent sections. To represent a relationship, we must be able to identify each participating entity and give values to the descriptive attributes of the relationship. Thus, the attributes of the relation include:

- The primary key attributes of each participating entity set, as foreign key fields.
- The descriptive attributes of the relationship set.

The set of nondescriptive attributes is a superkey for the relation. If there are no key constraints (see Section 2.4.1), this set of attributes is a candidate key. Consider the Works_In2 relationship set shown in Figure 3.10. Each department has offices in several locations and we want to record the locations at which each employee works.

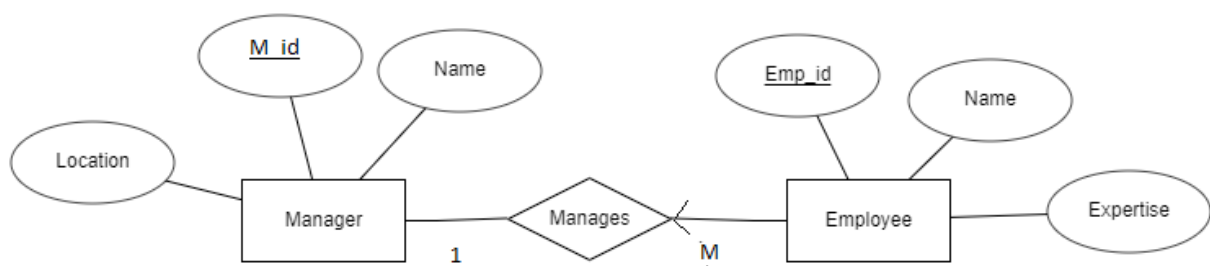


- 3 tables will be created ,one for employee ,one for departments and one for relationship .
- Note that,relation table will have primary keys of related entity sets and discriminatory attribute of relationship as primary key.
- The primary keys from two entities will become the foreign key for relationship.

NOTE: In this example entities are participating in many to many relationship. So we create 3 tables(Employee, Department, works_In).

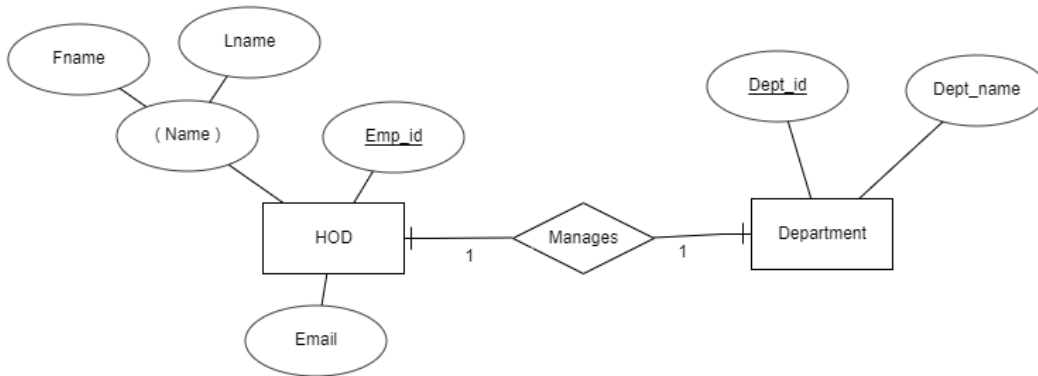
Translating Relationship Sets with Key Constraints

If a relationship set involves n entity sets and some m of them are linked via arrows in the ER diagram, the key for anyone of these m entity sets constitutes a key for the relation to which the relationship set is mapped. Hence we have m candidate keys, and one of these should be designated as the primary key.



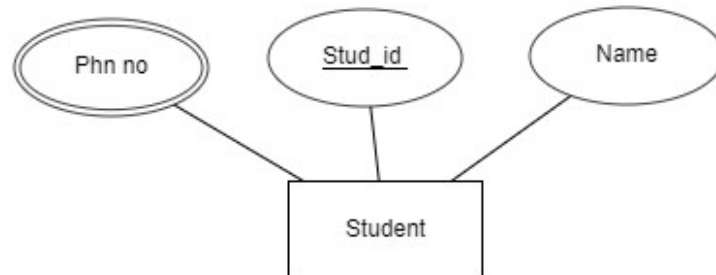
- Here we can merge the relationship table with entity table on many side.
- Primary key for the combined table would be same as the M sided entity's key and the primary key of one sided entity becomes the foreign key for the combined table.

ONE TO ONE CARDINALITY



- For “one-to-one cardinality”, we have key constraints on both the sides.
- Based on our convenience, we can merge the relationship with anyone of the entity set.
- Primary key of the remaining entity becomes the foreign key for the combined table.
- NOTE: Here we have one composite attribute “Name”, which has ‘Fname’ and ‘Lname’ as its simple attributes. So, while creating table, we need to use these simple attributes.

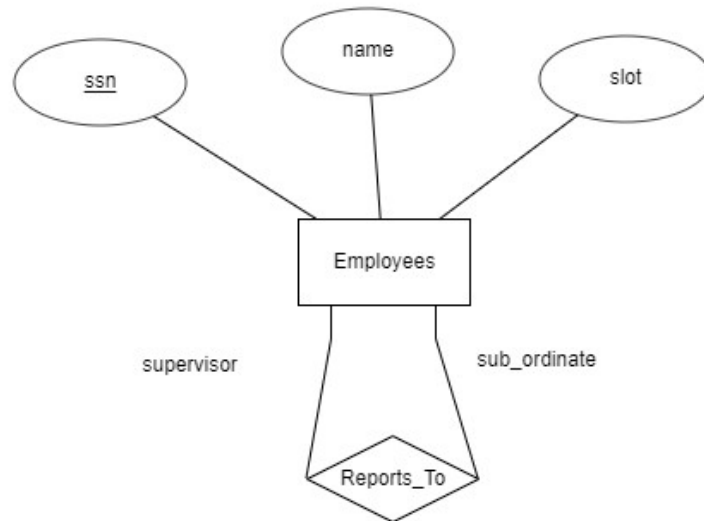
MULTI-VALUED ATTRIBUTE



- Let's deal with multi valued attribute. For a student, there can be multiple contact numbers.
- We cannot insert multiple entries for a single attribute in a tuple for a table.
- So we create **two tables** one with simple attributes and another with multivalued attribute and key of the entity.

Unary Relation

Consider the Reports_To relationship set shown in Figure 3.11. The

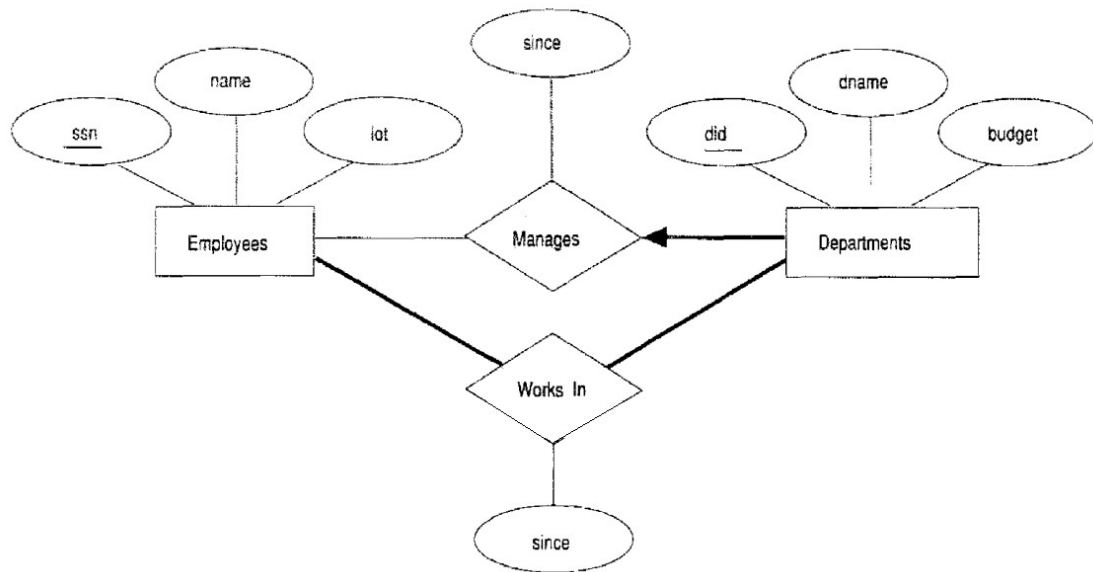


role indicators supervisor and subordinate are used to create meaningful field names in the CREATE statement for the Reports.

- As in Binary relationship, here also ,the table creation will depend on the cardinality of the relationship.
- Two attributes will be created for ssn ,viz. ssn_emp , ssn_sub.

Observe that we need to explicitly name the referenced field of Employees because the field name differs from the name(s) of the referring field(s).

Translating Relationship Sets with Participation Constraints

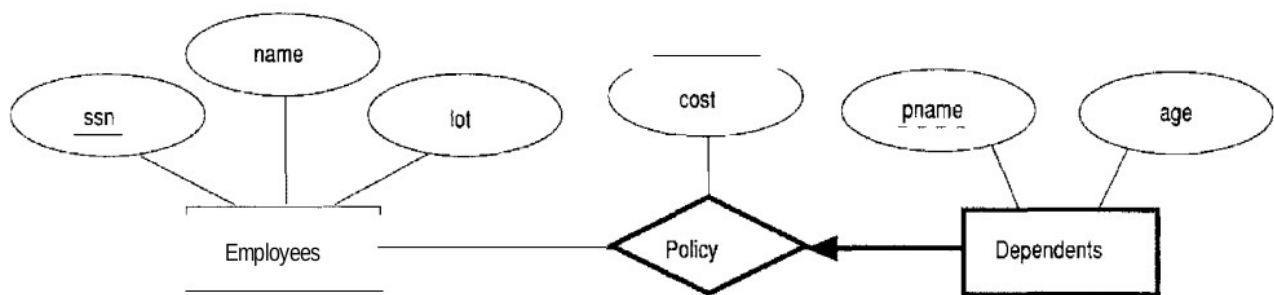


- For total participation ,along with the basic stated rules above, one more constraint need to be taken care i.e
- Along with foreign key constraint ,there should be NOT NULL constraint on the foreign key attribute.

Translating Weak Entity Sets

A weak entity set always participates in a one-to-many binary relationship and has a key constraint and total participation.

Consider the Dependents weak entity set shown in Figure with partial key pname. A Dependents entity can be identified uniquely only if we take the key of the owning Employees entity and the pname of the Dependents entity, and the Dependents entity must be deleted if the owning Employees entity is deleted.



We can capture the desired semantics with the following definition of the Dep_Policy relation:

- Primary key of weak entity cannot be defined by only by its attributes.
- So ,the primary key of weak entity would consists of disriminatory attribute and the primary key of owning entity .