# Chapter 9.1 Relational Database Design from ER (Entity-Relationship model).

## What is a relation?

* A relation is a subset of Cartesian product of some sets.

Let see an example:

S1 = {Bill, Jeff, Elon}

S2 = {Musk, Gates, Bezos}

Cartesian product of S1 and S2 = S1 \* S2 = {(Bill, Musk), (Bill, Gates), (Bill, Bezos),

(Jeff, Musk), (Jeff, Gates), (Jeff, Bezos),

(Elon, Musk), (Elon, Gates), (Elon, Bezos)}

A subset of S1 \* S2 = {(Bill, Gates), (Jeff, Bezos), (Elon, Musk)}

That is a relation between S1 and S2. You can reformat it as:

|  |  |
| --- | --- |
| Bill | Gates |
| Jeff | Bezos |
| Elon | Musk |

That is a table! Therefore a relation is a table and vice versa!

From chapter one:

* A database is a set of tables.

Now we can say: A database is a set of relations!

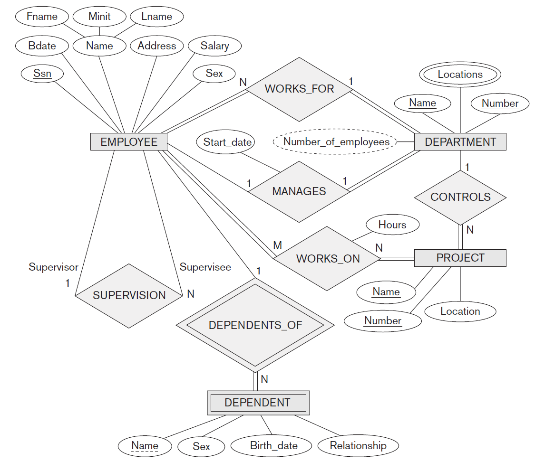
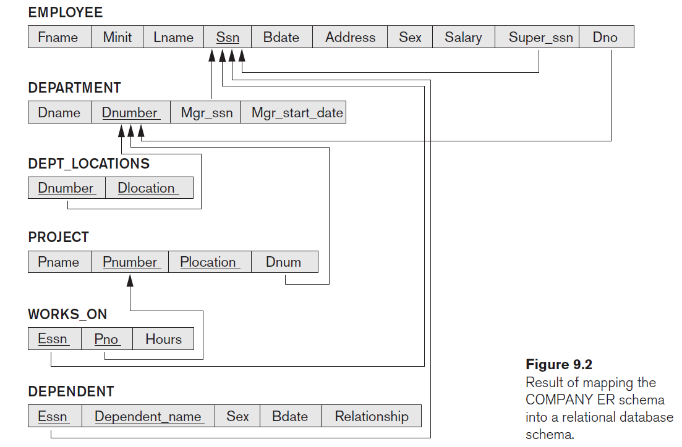
If you define your database as a set of relations, that is called Relational database, or Relational Model (RM).

Other equivalent terminologies:

* A tuple of a relation = a row of the table = a record of the database.

So far, we modeled a mini-world with ER model; in order to develop a database application for the mini-world by a DBMS software, we need to convert its ER to RM.

## Converting ER to RM; creating a tabular model for an ER model.



→

How to do it? There is a 7-step algorithm.

### ER-to-Relational Mapping Algorithm

Step 1: Mapping of Regular Entities

Step 2: Mapping of Weak Entities

Step 3: Mapping of Binary 1:1 Relationships

Step 4: Mapping of Binary 1:N Relationships

Step 5: Mapping of Binary M:N Relationships

Step 6: Mapping of Multivalued attributes.

Step 7: Mapping of N-ary Relationships.

Goals while running the algorithm

* Preserve all information, including all attributes and relationships
* Maintain the constraints to the extent possible

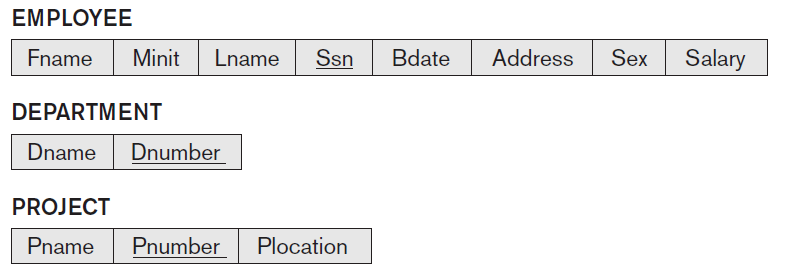
Relational Model cannot preserve all constraints;

RM cannot enforce min-max cardinality ratio such as (2, 4) of ER.

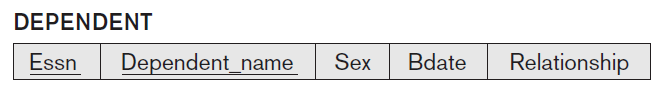
* Minimize null values

Depending on how you do the conversation, the number of null values in table records varies.

### Step 1: Mapping of Regular Entities.

* Create a relation R1 for each regular entity E1:
  + include all simple attributes of E1 as columns for R1.
  + For composite attributes, add only their simple attributes.
  + In ER, entity E1 could have multiple key; but in RM, relation R1 can have only one key, it is called PRIMARY KEY!
    - Choose one of the key attributes of E1 as the primary key for R1.
    - If the chosen key of E1 is composite, the set of component attributes all together, will form the primary key of R1.
* Example for Step 1:

### Step 2: Mapping of Weak Entities

* Similar to Step1, for weak entity E2 with owner entity E\_owner, create a relation R2; Include all simple attributes (or simple components of composite attributes) of E2 as columns of R2.
  + R2 missing something important, what is it?
* E2 is a weak entity, doesn’t have a key!
  + Current R2 could have identical rows.
    - Dependents with the same names.
  + We should add something to make rows unique.
    - To make dependents identifiable.
* Create a primary key for R2:
  + Combine the primary key of E\_owner’s relation with the *partial key* of E2.
* Example for Step 2:

#### FOREIGN KEY

* If you use primary key of R3, in relation R4, it is called a foreign key!

Example: Essn (Employee SSN) in the DEPENDANT relation. SSN is a primary key in EMPLOYEE relation.

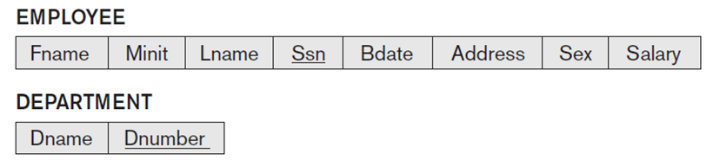
* We draw an arrow from a foreign key to the primary key that it relates to. For example Essn 🡪 SSN (please locate the arrow in the diagram in the previous page).

### Step 3: Mapping of Binary 1:1 Relationships

Assume S and T are participants of 1:1 relationship R.

3 ways:

1) Foreign Key approach:

* Choose one of the relations, let say S and include the primary key of T into S as a foreign key.
* Add all simple/component attributes of R to S.
* Example: EMPLOYEE manages DEPARTMENT
* It is better to choose an entity with total participation in the role of S. Why?

Consider EMPLOYEE manages DEPARTMENT and see the difference between adding Mngr\_ssn to DEPARTMENT vs. Dnumber to EMPLOYEE.

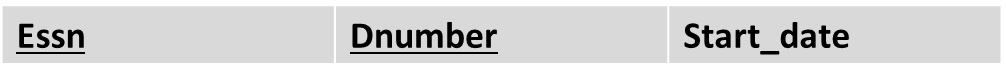
Hint: may 2% of employees manage a department.

2) Merging relations approach.

* + Only if both participants have **total** participations.
    - We need to have the same number of instances (rows) in both participants.
  + If S =R= T (S and T are in relationship R),
    - Merge S, T and R into one relation (table), let say R3.
    - |columns of R3| = |columns related to S| + |columns related to T|

+ |attributes of R|

3) Cross-reference or relationship relation:

* + Set up a third relation R4 to cross-reference the primary keys of the two relations corresponding to S and T.
  + R4 is a relationship relation;
    - Its columns are primary keys of S and T, plus the relationship attributes.
  + Example: EMPLOYEE manages DEPARTMENT
  + R4: