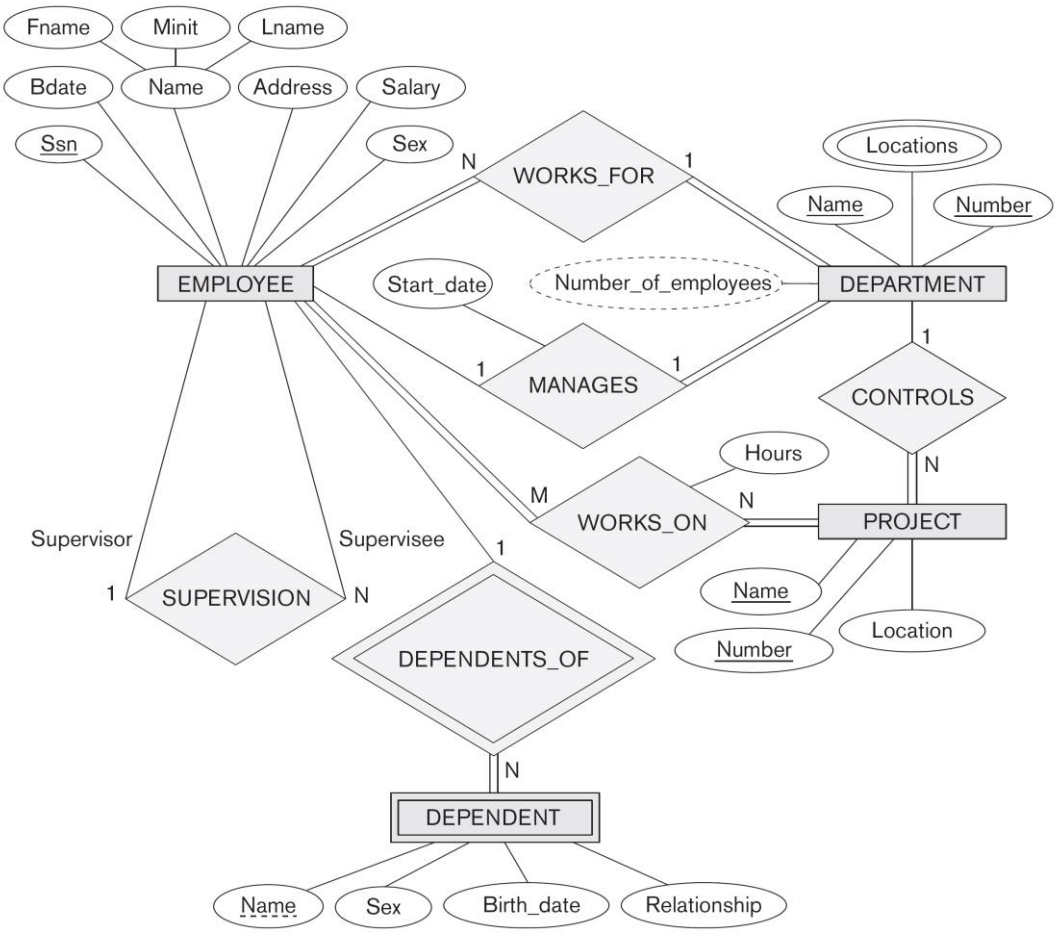


Figure 7.2
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter and is summarized in Figure 7.14.

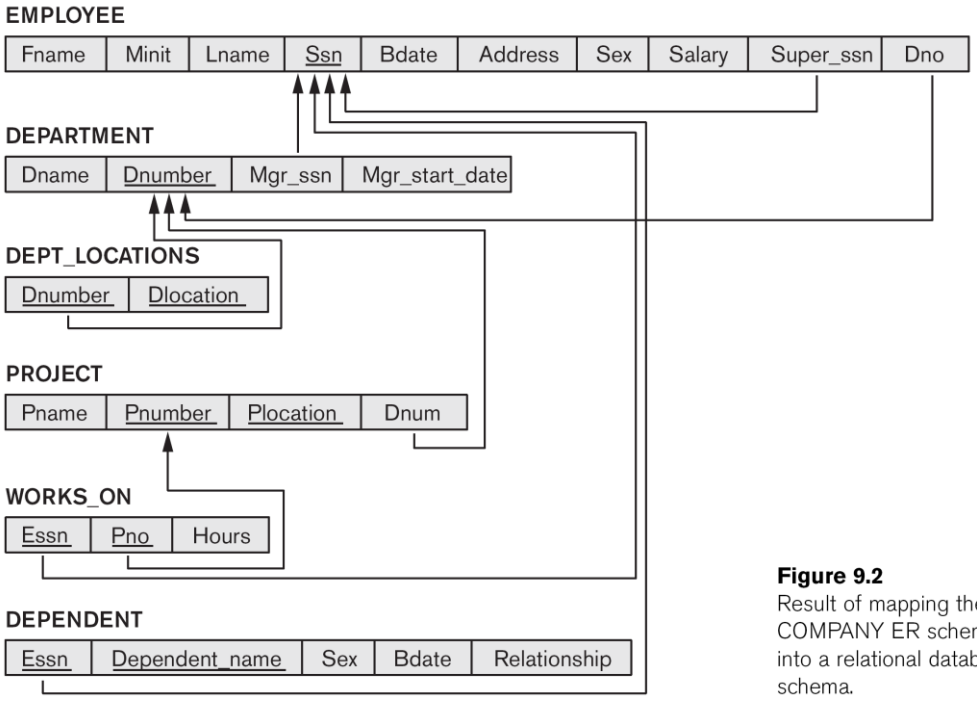


Figure 9.2
Result of mapping the COMPANY ER schema into a relational database schema.

ER-to-Relational Mapping Algorithm

Step 1: Mapping of Regular Entity Types

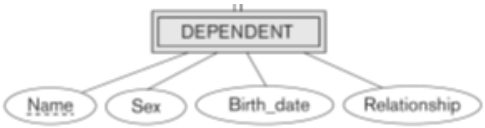
Each ENTITY (strong) becomes a table holds its name, its attributes become columns in this table and the key attribute becomes a PK (except For multivalued attributes)

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary

Step 2: Mapping of Weak Entity Types

Each Weak entity becomes a table holds its name, its attributes become columns in this table and the candidate attribute with Pk of the entity it belongs to **both** become a PK



DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

Step 3: Mapping of Binary 1: N Relationship Types

Every 1: N relationship mapped by moving the PK from the one side to the N side

DEPARTMENT

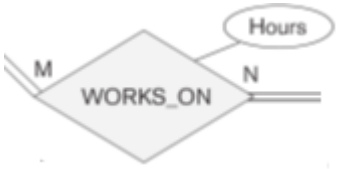
Dname	Dnumber	Mgr_ssn	Mgr_start_date

PROJECT

Pname	Pnumber	Plocation	Dnum

Step 4: Mapping of Binary M: N Relationship Types

Each M: N relationship becomes a table holds its name, its attributes become columns in this table and PKs for the two tables associated with the relationship **both** become a PK.



WORKS_ON

Essn	Pno	Hours

Step 5: Mapping of Multivalued attributes.

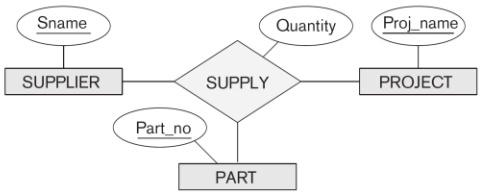
Each Multivalued attribute becomes a table named as a mixture of the attribute and entity it belongs to. The PK for the result table is the attribute itself with key of the ENTITY it belongs to.

DEPT_LOCATIONS

Dnumber	Dlocation

Step 6: Mapping of N-ary Relationship Types.

Each N-ary relationship becomes a table holds its name, its attributes become columns in this table and its PK is the PKs for all tables associated with the relationship.



SUPPLY

Sname	Proj_name	Part_no	Quantity

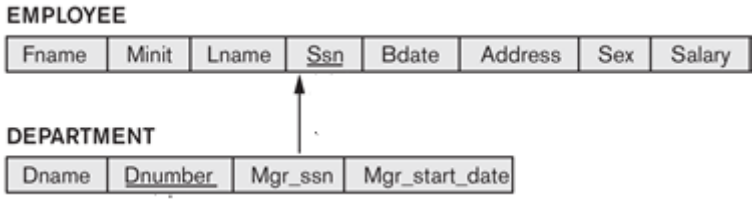
Step 7: Mapping of Binary 1:1 Relationship Types

In the case of 1: 1 relationship we look at the participation on both sides:

If -------: ===== we move the PK from partial side ----- to total side =====

If ====: ===== we merge the two tables and the relationship into one table

If -------: ------- the relation becomes a table holds its name, its attributes become columns in this table and its PK is the PKs for the two tables associated with the relationship.



Important Note: When we move the PK from one table to another by MAPPING, it becomes a foreign key (FK) in the destination table. In other words "strange key" so it is preferable to point to the table of origin for that key.

During mapping, relationship attributes must be moved to the same side where you moved the primary key.