



Logical Database Design



Main Topics

- Translating the E-R model to A logical data Model
- Normalization
 - Functional Dependency
 - Normal Forms

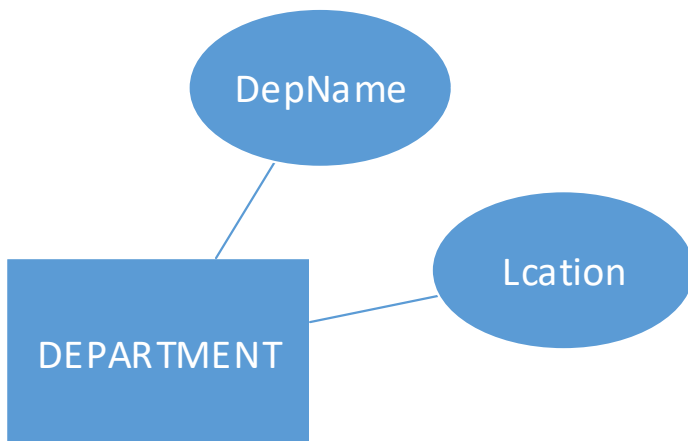


Building a Logical Data Model

- Derive relations for logical data model
 - To create relations for the logical data model that represent the entities, relationships, and attributes that have been identified.
- Mappings
 - Entity will be mapped to table
 - Attributes will be mapped into columns of the respective tables.
 - Relationships will be mapped according to their cardinality type

Mapping of Entities

- Entity names will be automatically be Table names
- The attributes of the entity will be the columns of the table
- Example



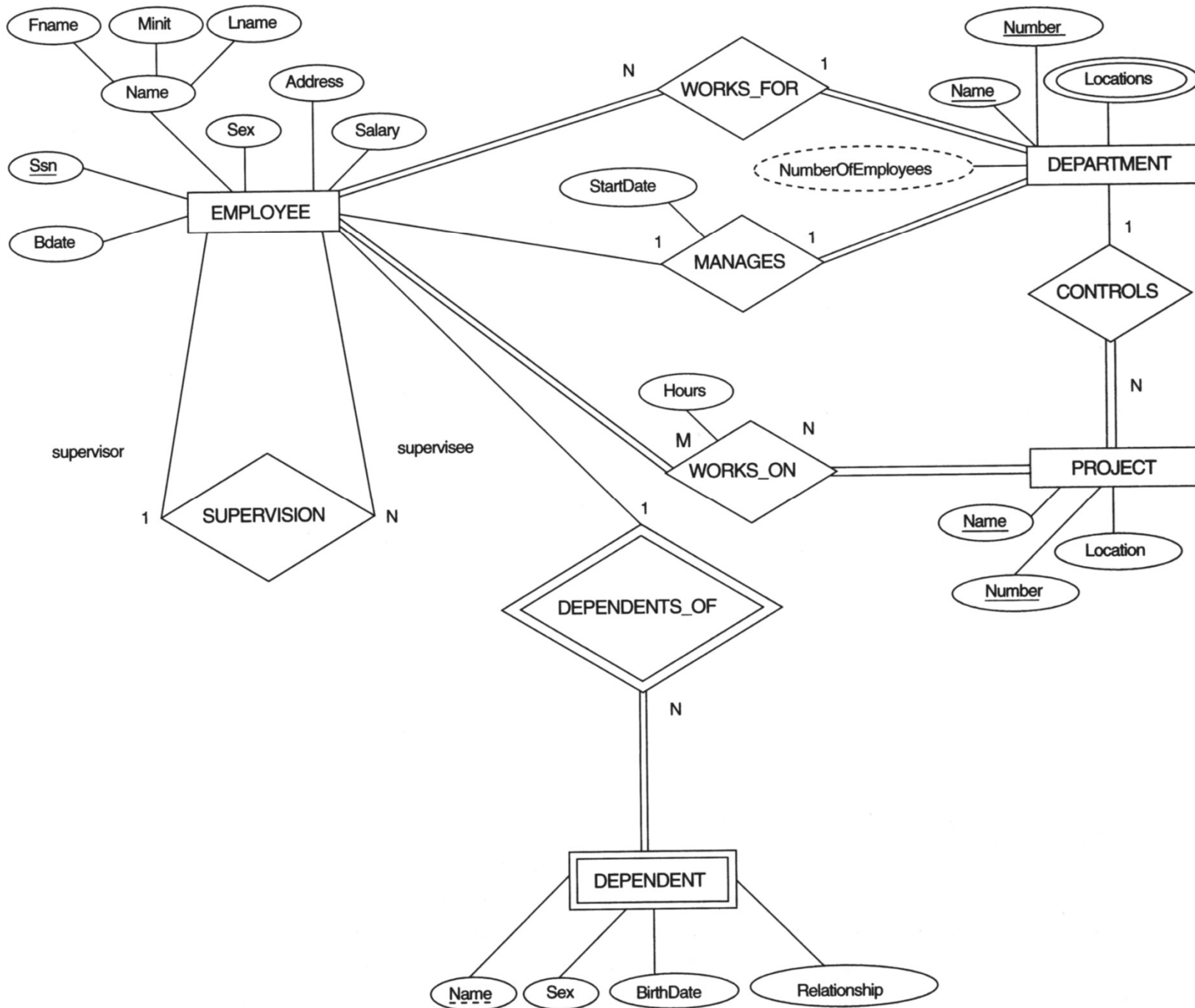
Department	
DepName	Location

- Department = (DepName , Location)



Mapping of Attributes

- Atomic or single-valued attributes will be mapped to individual columns
- Composite attributes
 - The parent attribute will be ignored
 - The decomposed attributes (child attributes) will be columns of the table
- Multi-valued attributes
 - Will be mapped to a new table
 - The primary key of the main table will be posted for cross referencing





Mapping of Strong Entity Types

- For each regular (strong) entity type **E** in the ER Diagram
 1. Create a relation **R** that includes all the simple attributes of **E**.
 2. Choose one of the key attributes of **E** as the primary key for **R**.
 3. If the chosen key of **E** is composite, the set of simple attributes that form it will together form the primary key of **R**.

Example:

- Create the relations EMPLOYEE, DEPARTMENT, and PROJECT in the relational schema corresponding to the regular entities in the ER diagram.
 - SSN, DNUMBER, and PNUMBER are the primary keys for the relations EMPLOYEE, DEPARTMENT, and PROJECT



Mapping of Strong Entities (cont.)

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary
-------	-------	-------	------------	-------	---------	-----	--------

DEPARTMENT

Dname	<u>Dnumber</u>
-------	----------------

PROJECT

Pname	<u>Pnumber</u>	<u>Plocation</u>
-------	----------------	------------------



Mapping of Weak Entity Types

- For each weak entity type W with owner entity type E
 - Create a relation R
 - Include all simple attributes of W as attributes of R .
 - Include as foreign key attributes of R the primary key attribute(s) of the relation that correspond to the owner entity type.
- The primary key of R is
 - The combination of the primary key(s) of the owner(s) and
 - The partial key of the weak entity type W , if any

Mapping Weak Entity Types (cont.)

■ Example:

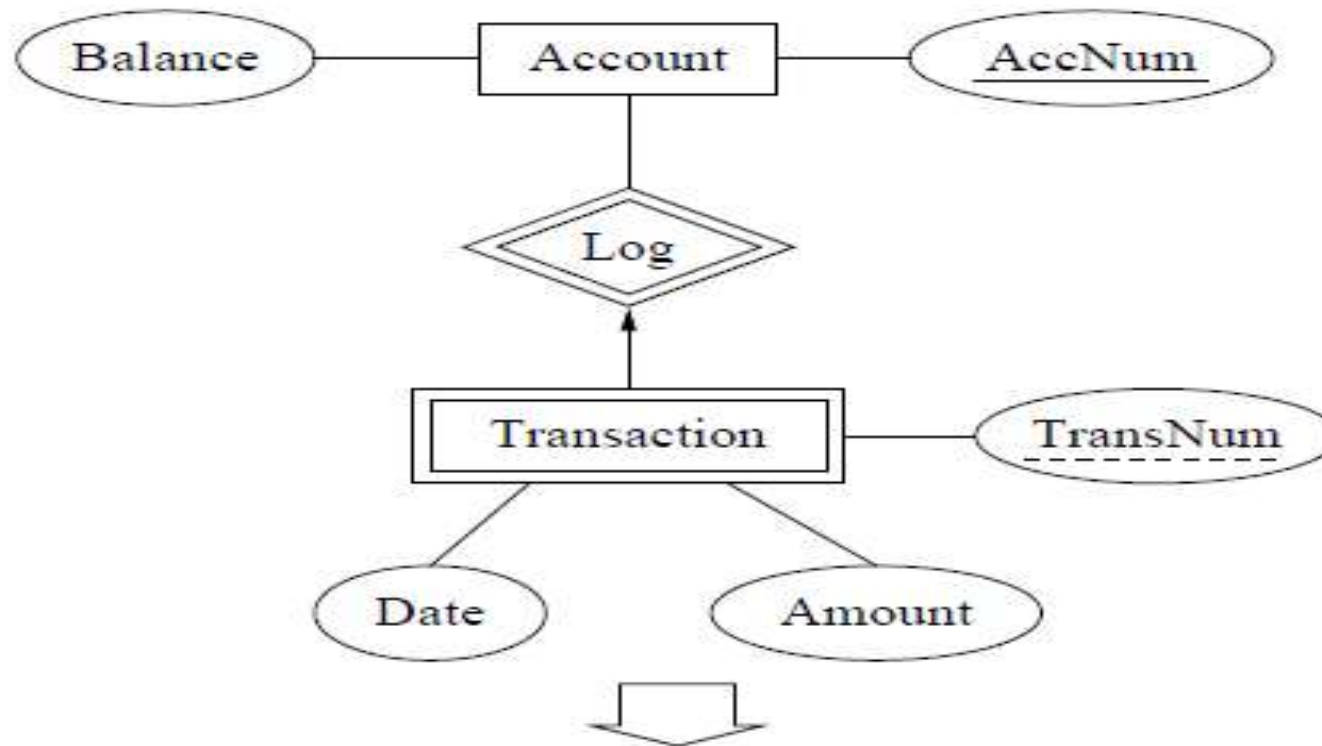
- Create the relation DEPENDENT for the weak entity type DEPENDENT.
- Include the primary key SSN of the EMPLOYEE relation as a foreign key attribute of DEPENDENT
- The primary key of the DEPENDENT relation is the combination {SSN, DEPENDENT_NAME}
 - because DEPENDENT_NAME is the partial key of DEPENDENT.



DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

Mapping Weak Entity Types (cont.)



Account

<u>AccNum</u>	Balance
---------------	---------

Transaction

<u>TransNum</u>	<u>AccNum</u>	Date	Amount
-----------------	---------------	------	--------



Mapping 1:1 Relationship

- One-to-one (1:1) recursive (Unary) relationships
 - represent the recursive relationship as a single relation with two copies of the primary key
 - Example: *Person is married to another person*
Person = (PersonID , Fname , Lname , PartnerID)
- One-to-one (1:1) binary relationships
 - creating two relations and post a copy of the primary key from one relation to the other
 - Example: *A customer can be an employee*
Employee = (EmpID , Fname , Lname , SupID)
Customer = (CustID , Name , CustomerType , EmpID)



Mapping 1:N Relationship

- The entity on the 'one side' of the relationship is designated as the **Parent Entity** and the entity on the 'many side' is designated as the **Child Entity**
- For each regular binary 1:N relationship type
 - identify the relation S that represent the entity at the 1-side of the relationship
 - Include the primary key of S as a foreign key in the entity on the N-side of the relationship
 - Attributes of the relationship(if any) are posted on the many side



Mapping 1:N Relationship (cont.)

■ Example: WORKS_FOR

- **Works_For** is a 1:N relationship between EMPLOYEE and DEPARTMENT
- Include the primary key of DEPARTMENT in the EMPLOYEE relation and call it DNO

Employee = (EmpID , Fname , Lname , DNo)

Department = (Dno , DepartmentName, Location)

■ Example: Supervised_By

- **Supervised_By** is a 1:N relationship between EMPLOYEE and EMPLOYEE

Employee = (EmpID , Fname , Lname , SupervisorID)



Mapping M:N Relationship

- For each regular binary M:N relationship type R
 - Create a new relation S to represent the relationship R.
 - Include as foreign key attributes of S
 - the primary keys of the relations that represent the participating entity types
 - The combination of the foreign will be the primary key of S.
 - Include any simple attributes of the M:N relationship type as attributes of S.



Mapping M:N Relationship (cont.)

Example:

- The WORKS_ON is a M:N relationship between PROJECT and EMPLOYEE
- Create a relation R for WORKS_ON
 - The primary keys of the PROJECT and EMPLOYEE relations are included as foreign keys in WORKS_ON
 - Optionally rename them as PNO and ESSN, respectively.
- The primary key of the WORKS_ON relation is the combination of the foreign key attributes {ESSN, PNO}.
- HOURS will be an Attribute of WORKS_ON relation

Mapping M:N Relationship (cont.)

Example:

- The WORKS_ON relationship

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

PROJECT

Pname	<u>Pnumber</u>	<u>Plocation</u>	Dnum
-------	----------------	------------------	------

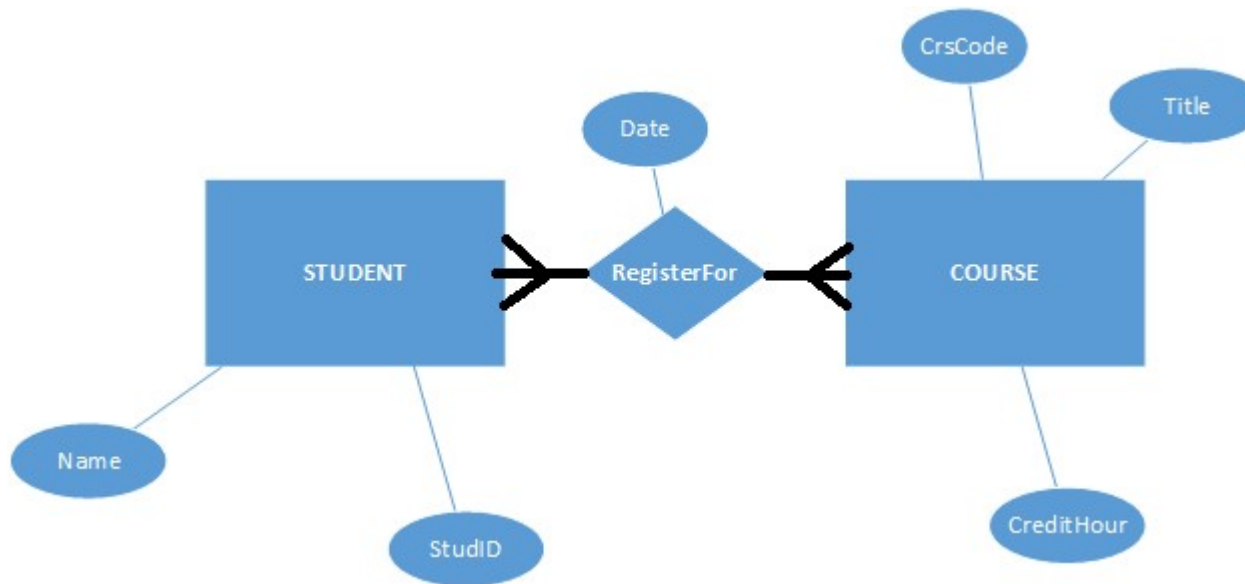
WORKS_ON

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

Mapping M:N Relationship (cont.)

Exercise:

- STUDENT registers for COURSE



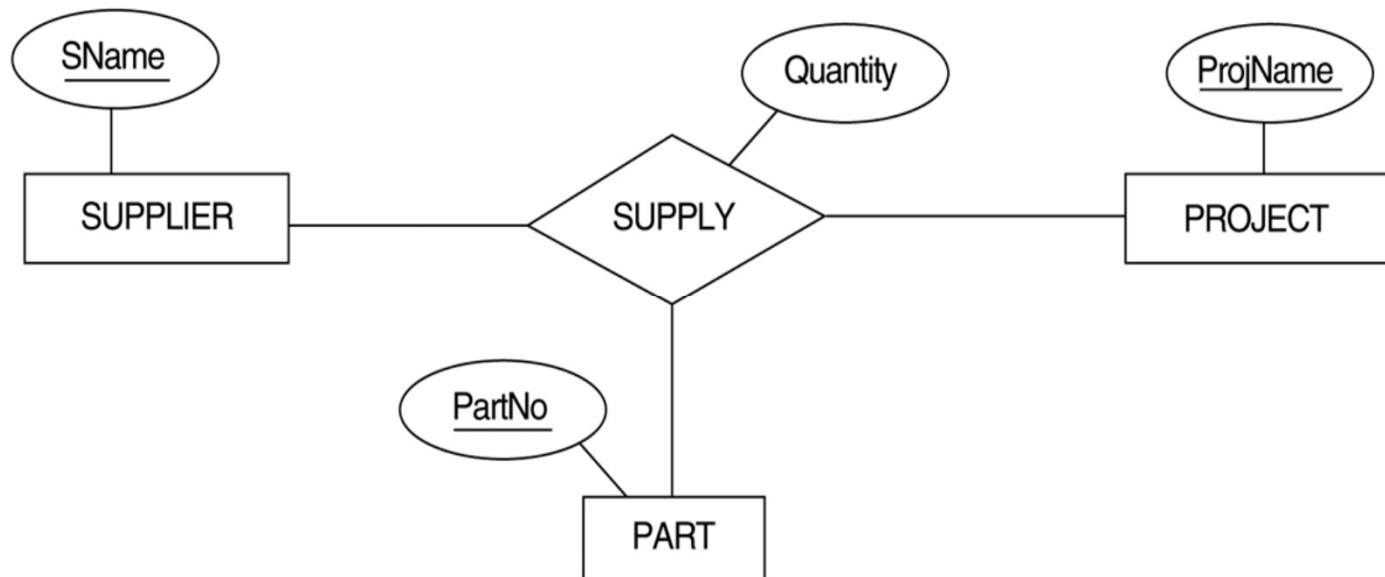


Mapping N-ary Relationships

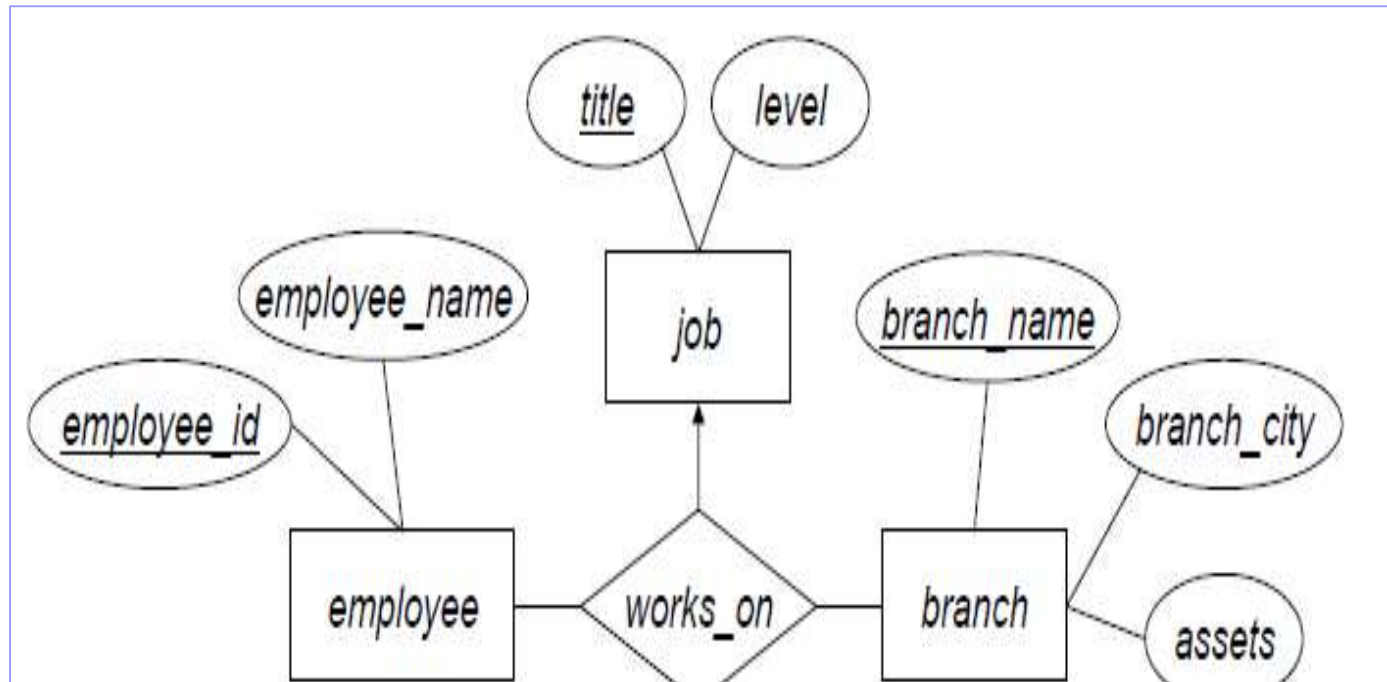
- For each n-ary relationship type R, where $n > 2$
 - create a new relationship S to represent R.
 - Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types.
 - Also include any simple attributes of the n-ary relationship type as attributes of S.

N-ary Relationships (cont.)

- Example: The relationship SUPPY is a ternary relationship
- This is mapped to the relation SUPPLY
 - primary key is the combination of the three foreign keys {SNAME, PARTNO, PROJNAME}
 - Quantity will be an attribute of SUPPLY



N-ary Relationships (cont.)



■ Relations:

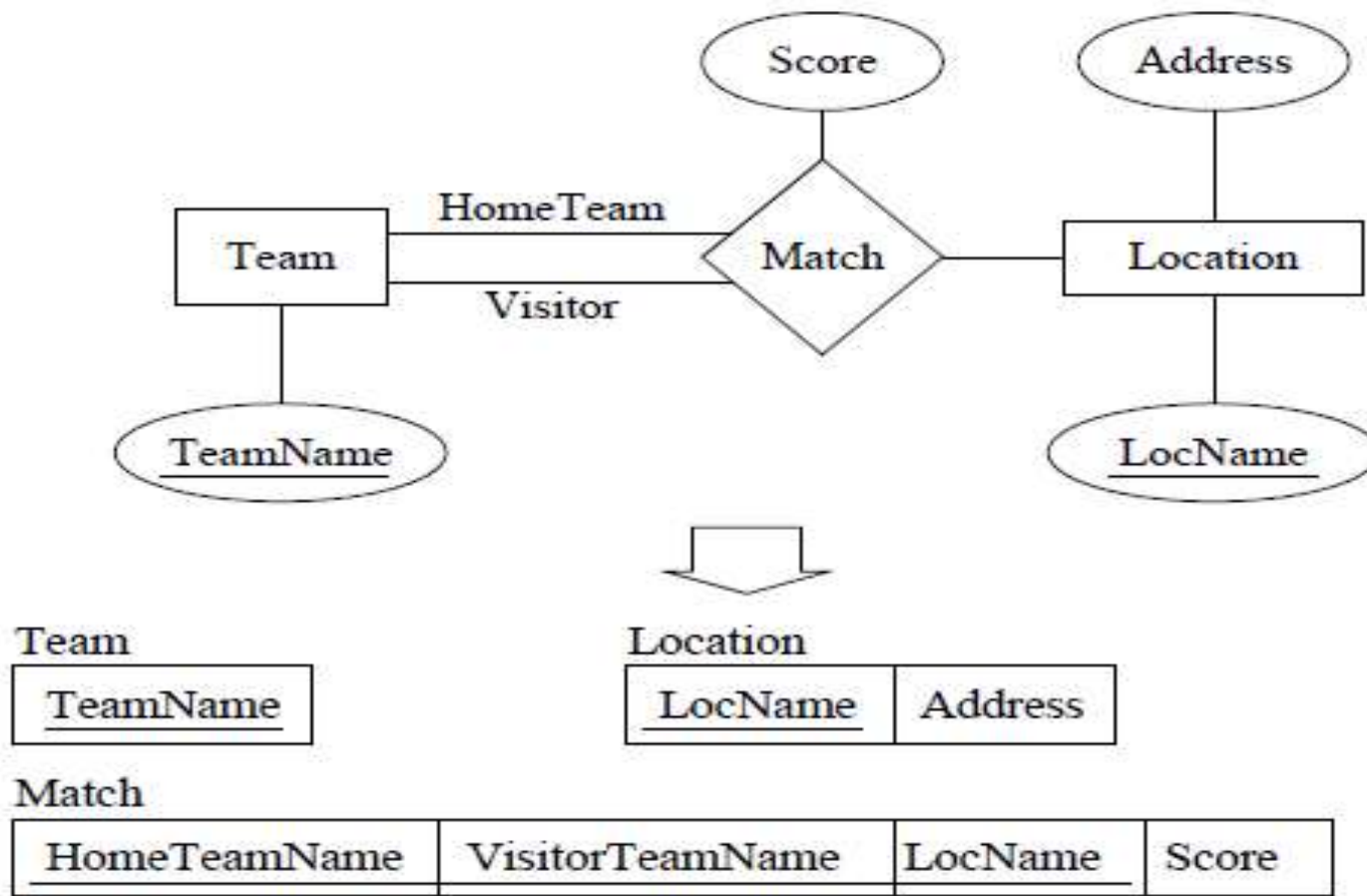
- job(title, level)
- employee(employee_id, employee_name)
- branch(branch_name, branch_city, assets)
- Works_On (title , employee_id , branch_name

■ Primary key

- employee_id, branch_name, title

N-ary Relationships (cont.)

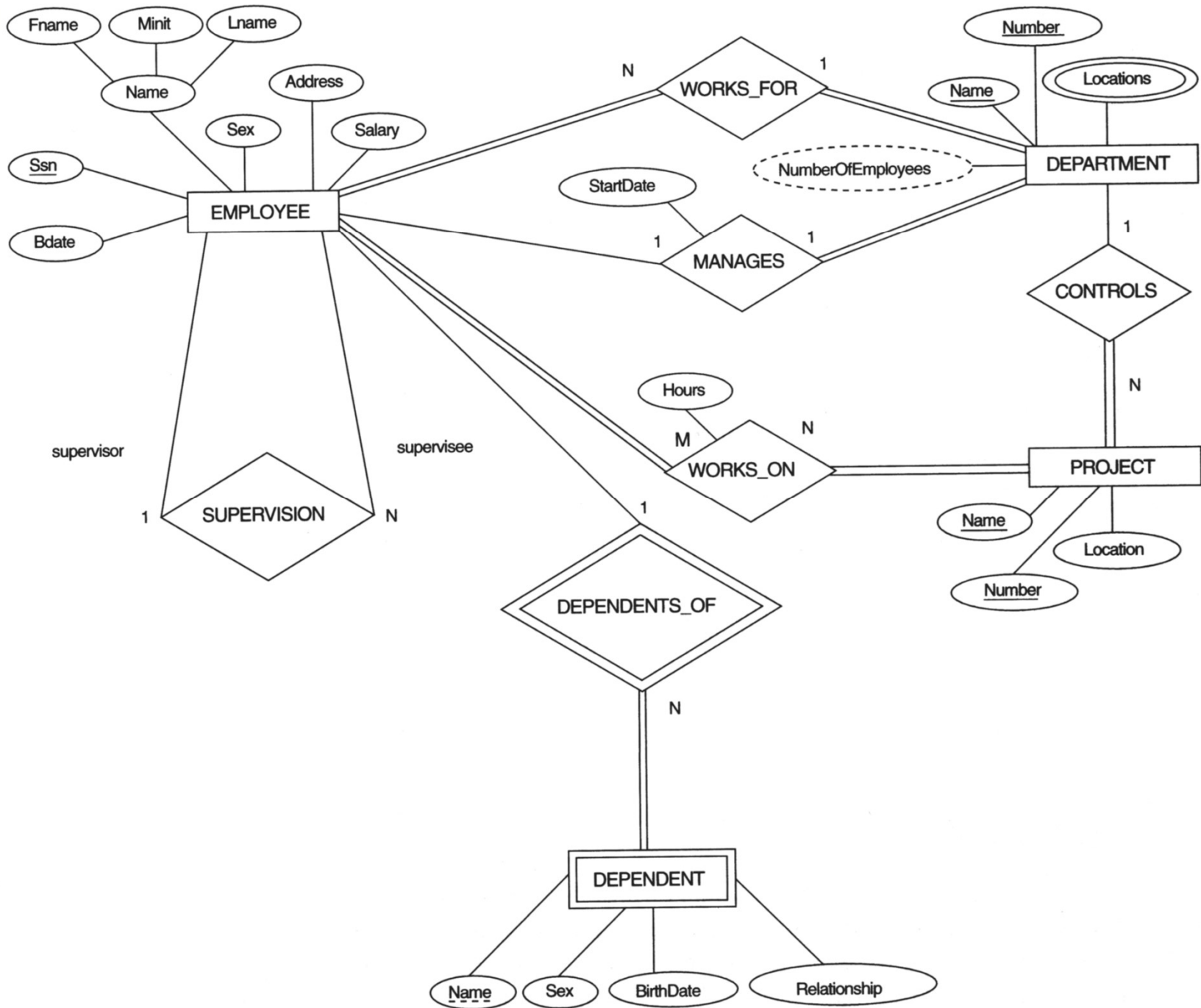
- Example





Mapping - Summary

ER Model	Relational Model
Entity type	“Entity” relation
1:1 or 1:N relationship type	Foreign key (or “relationship” relation)
M:N relationship type	“Relationship” relation and two foreign keys
<i>n</i> -ary relationship type	“Relationship” relation and <i>n</i> foreign keys
Simple attribute	Attribute
Composite attribute	Set of simple component attributes
Multivalued attribute	Relation and foreign key



EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
-------	----------------	---------	----------------

DEPT_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

PROJECT

Pname	<u>Pnumber</u>	<u>Plocation</u>	Dnum
-------	----------------	------------------	------

WORKS_ON

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------