Database Fundamentals

Lecture 3
Relational Model – Data Modeling Mapping

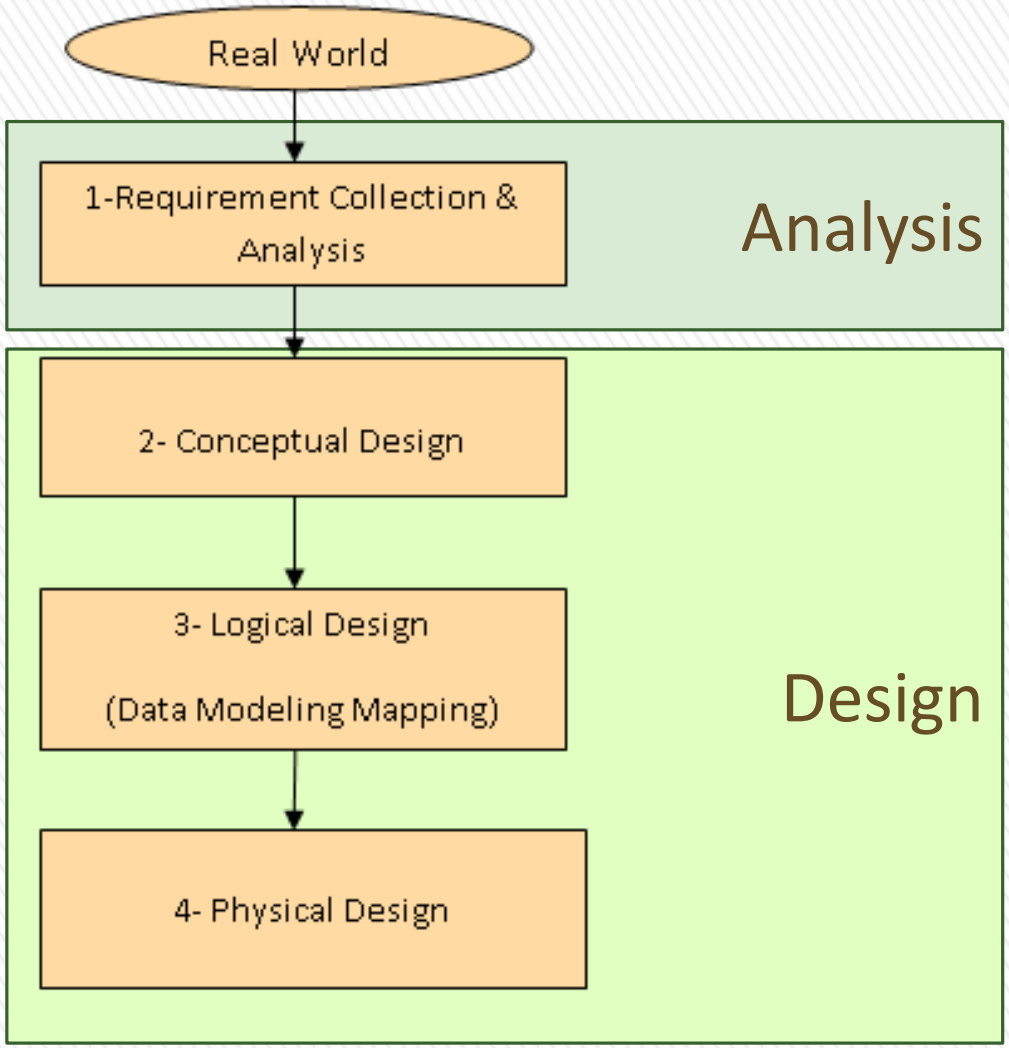
Prepared By: Eng. Doaa Soleiman

Lecture 3
Entity Relationship Diagram (ERD)

1- Relational Data Model

2 – Data Modeling Mapping – Mapping Rules

Database Design Phases

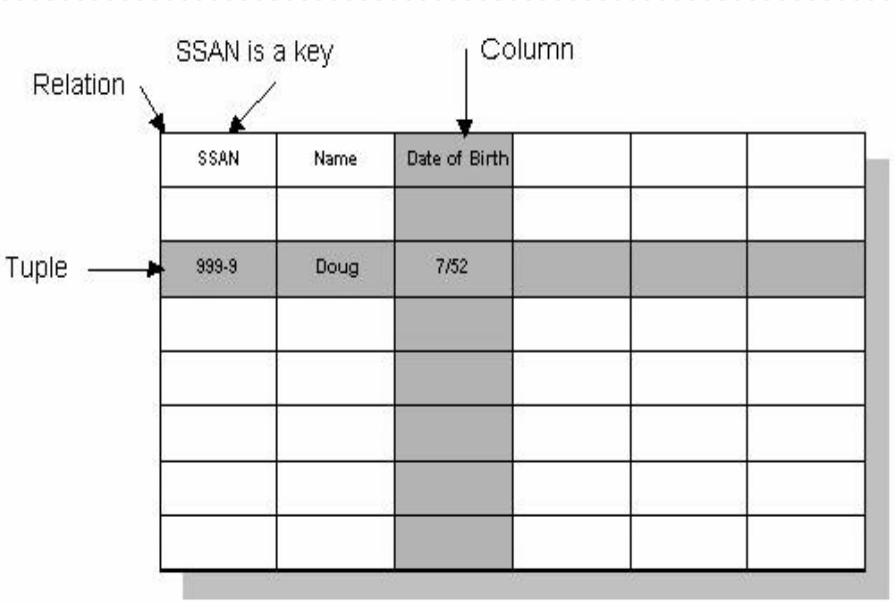


2- Logical Design

Definitions

Relation:

- A Collection of attributes describing the data.
- Relation is what normally called Table.
- Table consists of rows and columns, Data value is stored in the rows and columns intersection



Definitions

Tuple:

Rows of such a table

Attribute:

Columns of such a table.

Primary Key:

Unique identifier for the table, may be single or composite.

Domains:

Pool of values from which one or more attributes draw their actual values.

Definitions

Candidate Key:

Each relation has at least one candidate key, each candidate key K of a relation R has two properties:-

Uniqueness: At any given time no two tuples of R have the same value for K.

Minimality: If K is composite, no component of K can be eliminated without destroying the uniqueness.

Definitions

Primary Key:

Each relation has only ONE primary key, It is chosen from the candidate keys.

Foreign Key:

An attribute in relation R2 whose values are required to match those of the primary key of some relation R1.

Integrity Rules

Entity Integrity Rule:

- No attribute participating in the primary key of base relation is allowed to accept NULL values
- The rows in a relation represent entities and each one must be uniquely identified

Referential Integrity Rule

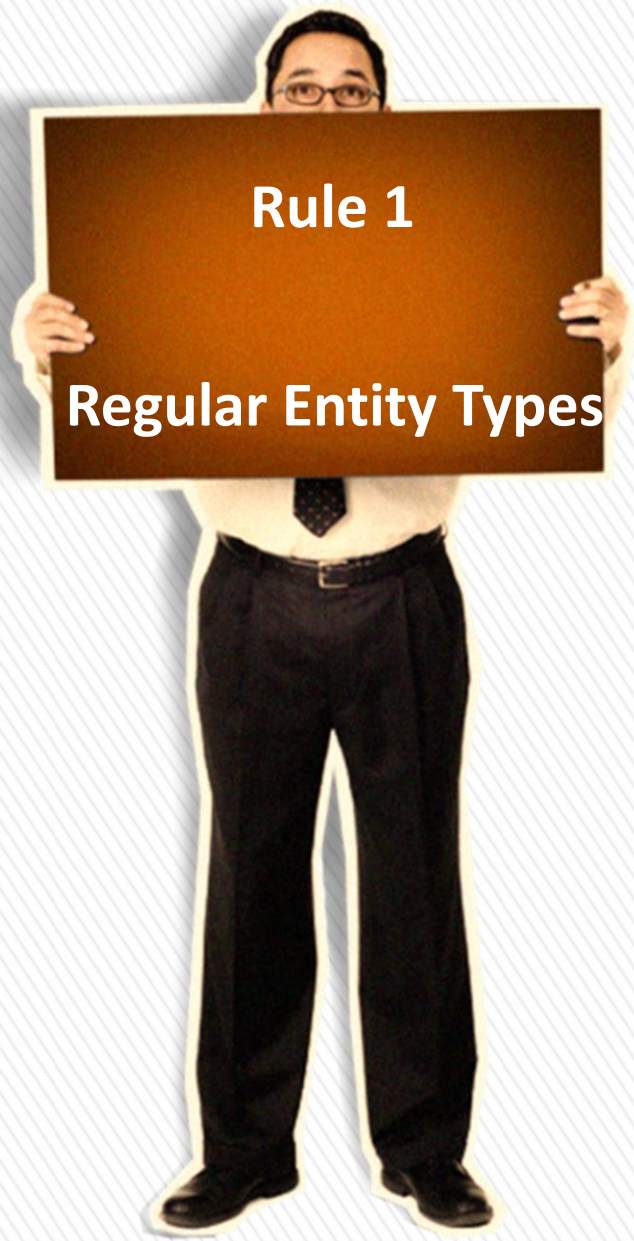
- If base relation R2 includes Foreign Key matching the Primary key of some base relation R1, then every value of Foreign Key must either equal a value of Primary Key in some tuple of R1, or be NULL.

Cascade Update / Delete commands

- Cascade: Delete / Update the target row and all rows that point to it (via foreign keys) are also deleted
- Restricted: The user cannot delete the target row until all rows that point to it (via foreign keys) have been deleted.
- Nullifies: can delete the target row and all foreign keys (pointing to it) are set to null

Mapping Rules 1

- Create table for each entity type include all simple attribute.
- For composite attribute, include only the simple attribute.
- Choose one of key attributes to be the primary key



Mapping Rules 2

- Create table for each weak entity.
- Add foreign key that correspond to the owner entity type.
- Choose the primary key : (FK + weak entity Partial PK if any)

Mapping Rules 3

- Merged two tables if both sides are Mandatory.
- Add FK into table with the total participation relationship to represent optional side.
- Create third table if both sides are optional.

Mapping Rules 4

- Add FK to N-side table
- Add any simple attributes of relationship as column to N-side table.

Mapping Rules 5

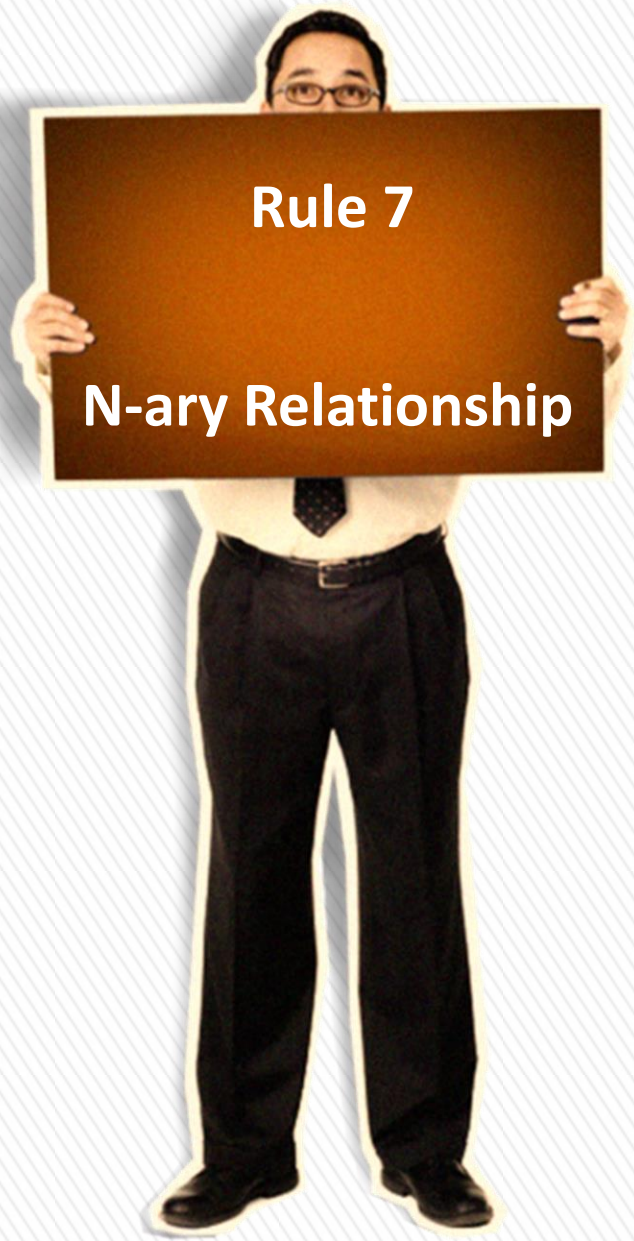
- Create a new third table
- Add FKs to the new table for both parent tables
- PK is the combination of both FKs.
- Add simple attributes of relationship to the new table if any.

Mapping Rules 6

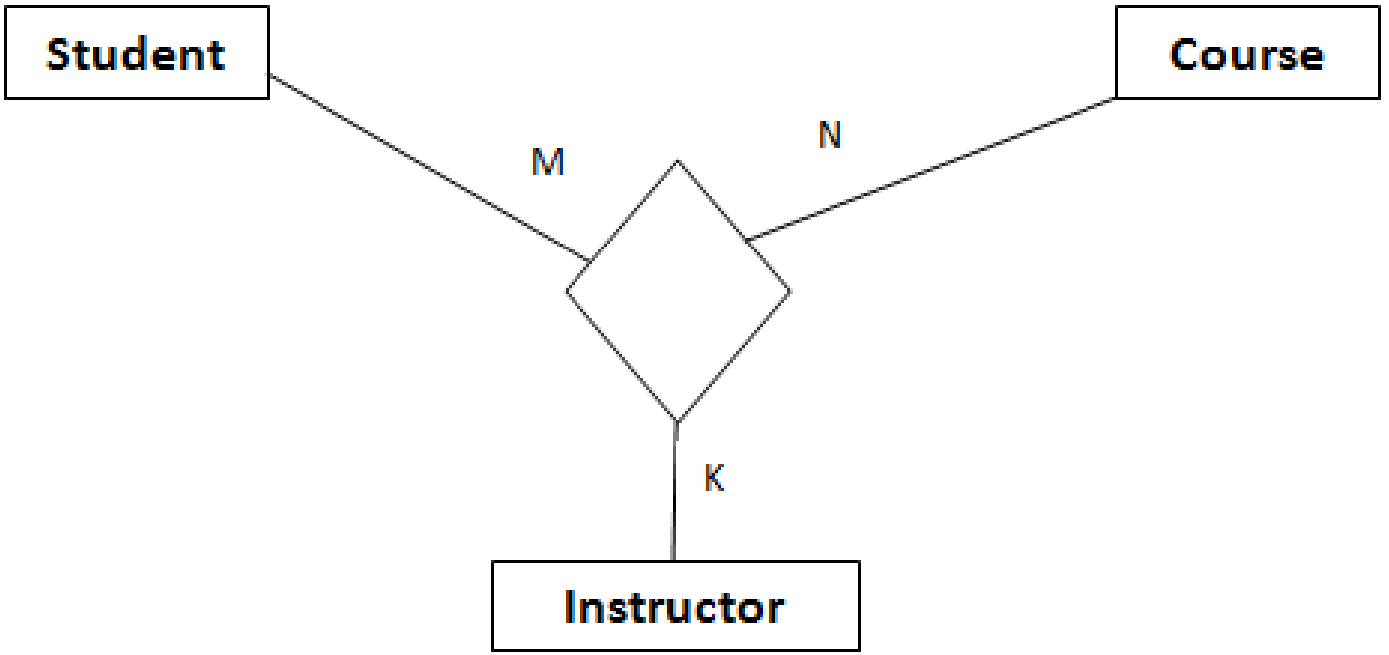
- Create new table for each multi-valued attribute
- Table will include two columns: One for multi-valued attribute + FK column.
- PK is the combination of FK and the attributes.

Mapping Rules 7

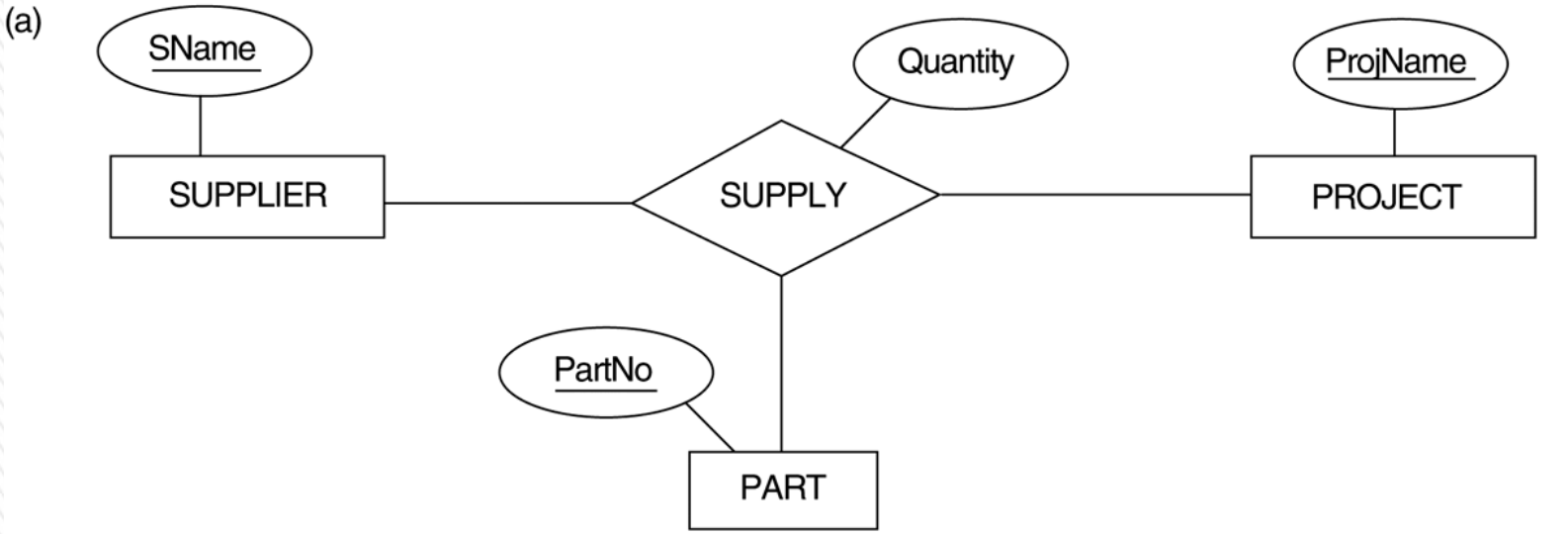
- If n > 2 then : Create a new third table
- Add FKs to the new table for all parent tables
- PK is the combination of all FKs.
- Add simple attributes of relationship to the new table if any.



N-ary Relationship



N-ary Relationship



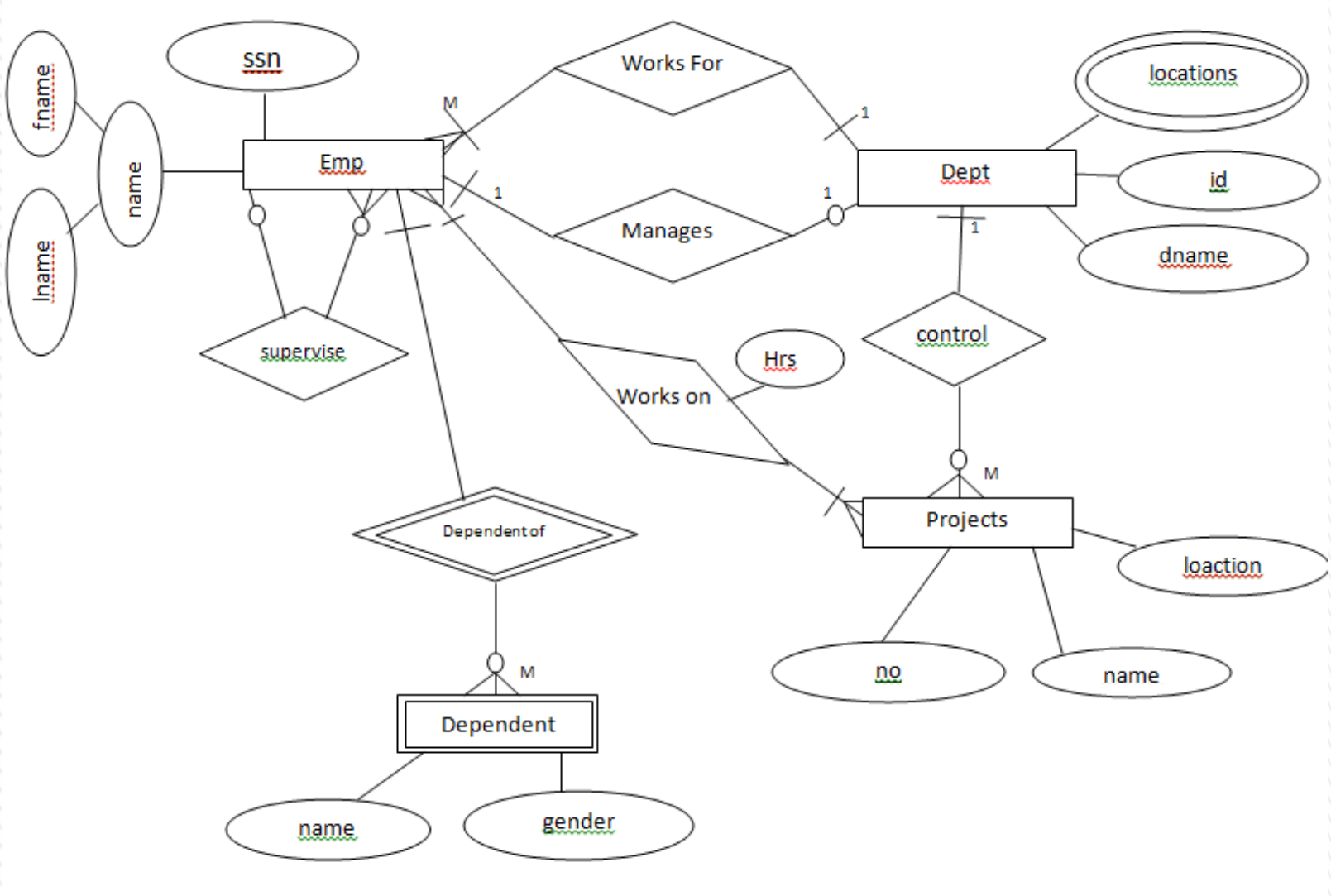
Example

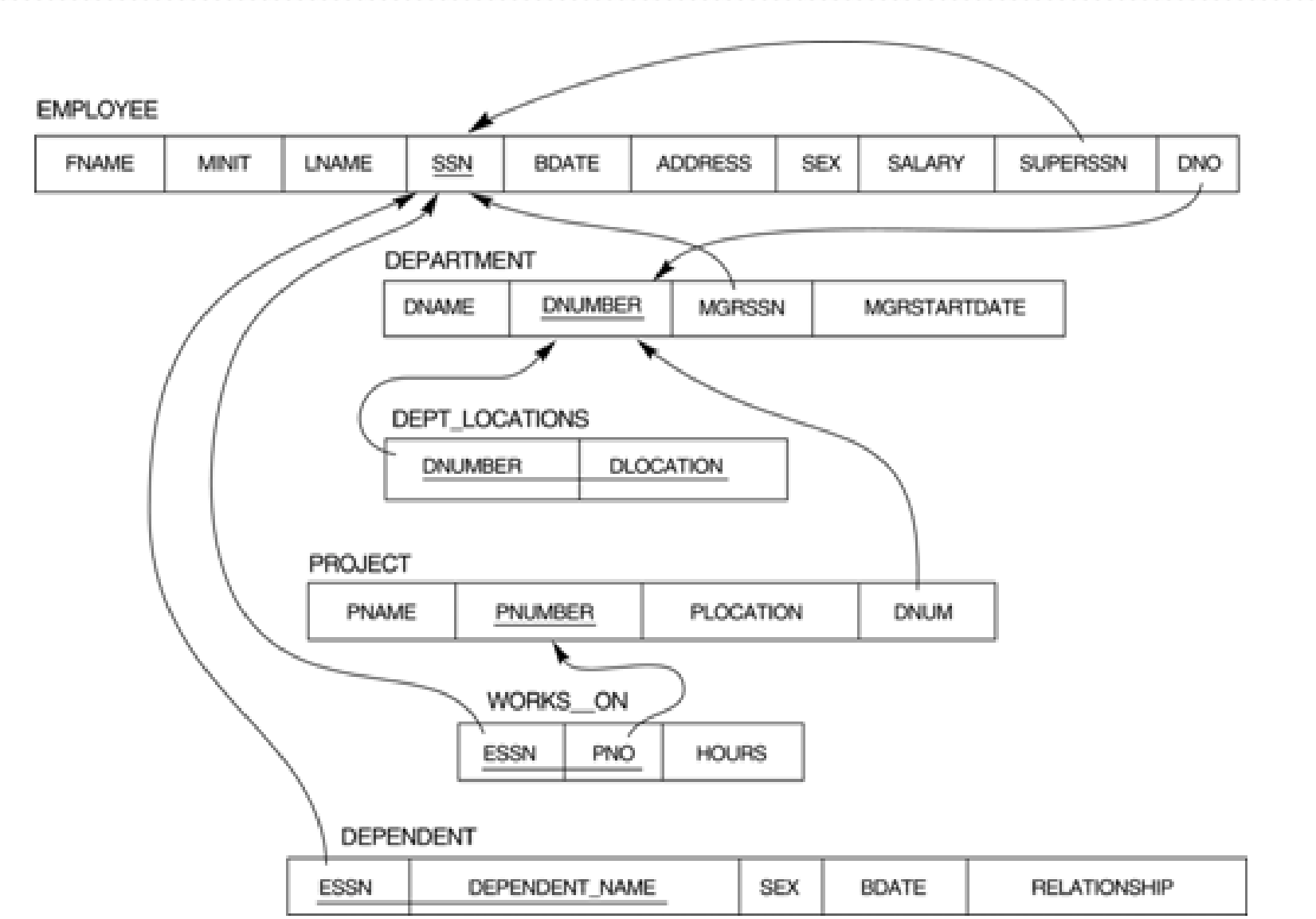
- A company is organized into departments. Each department has a unique name, a unique number, and a particular employee who manages the department. A department may have several locations.
- A department may control a number of projects, each of which has a unique name, a unique number, and a single location. A project must controlled by department

Example (Cont'd)

- We store employee's name, social security number, address, salary, gender and birth date. An employee must be assigned to one department and must work on one or more projects, which are not necessarily controlled by the same department. We keep track of the number of hours per week that an employee works on each project. We also keep track of the direct supervisor of each employee.
- We want to keep track of the dependents of each employee for insurance purposes. We keep each dependent's first name, gender, birth date and relationship to that employee.

Solution


ssn
fname
name
lname
Emp
Works For
Manages
M
1
1
1
Dept
locations
id
dname
supervise
control
Hrs
Works on
M
Projects
loaction
Dependent of
M
Dependent
no
name
name
gender


EMPLOYEE
FNAME
MINIT
LNAME
SSN
BDATE
ADDRESS
SEX
SALARY
SUPERSSN
DNO
DEPARTMENT
DNAME
DNUMBER
MGRSSN
MGRSTARTDATE
DEPT_LOCATIONS
DNUMBER
DLOCATION
PROJECT
PNAME
PNUMBER
PLOCATION
DNUM
WORKS_ON
ESSN
PNO
HOURS
DEPENDENT
ESSN
DEPENDENT_NAME
SEX
BDATE
RELATIONSHIP