Lecture 1
Database Fundementals

by:
Ahmed Adel

Basic Definitions

- Database: A collection of related data.
- Database: A Structured collection of Records.

- Database Management System (DBMS): A software package/ system to facilitate the creation and maintenance of a computerized database.

- Database System: The DBMS software together with the data itself. Sometimes, the applications are also included. (Software + Database)

Single Table Data Base

ISBN	Title	AUID	AUName	AUTel	PubID	PubName	PubT
099-9999-1	Emma	1	Austen	222-33333	1	Big House	222-3
091-88881	Ferry	5	Jones	33-555555	1	Big House	222-3
082-44444	Hamlet	2	Fedd	44-777777	2	Alpha Press	666-8
034-22234	Hank	6	Sam	66-444444	1	Big House	222-3
071-22222	Mazi	2	Fedd	44-777777	2	Alpha Press	666-8
043-55555	Rios	8	Mark	55-333333	3	Small House	444-9
043-55555	Rios	9	Joe	22-567894	3	Small House	444-9
054-66666	Hazard	8	Steven	44-555555	1	Big House	222-3
034-56789	Ballon	12	Alberto	33-666666	2	Small House	444-9
034-56789	Ballon	13	Jack	77-999999	2	Small House	444-9

Single Table Disadvantages

- Duplication of Data
- Insert Anomaly
- Delete Anomaly
- Update Anomaly
- Frequent Null Values

DBMS Advantages

- Controlling Redundancy.
- Restricting Unauthorized Access.
- Sharing data.
- Enforcing Integrity Constraints
- Inconsistency can be avoided.
- Providing Backup and Recovery.

DBMS Disadvantages

- It needs expertise to use (which is expensive)
- DBMS itself is expensive
- The DBMS may be incompatible with any other available DBMS

Database Users

- Database Administrator (DBA)
- System Analysts
- Database Designer
- Application programmers
- Testing Unit
- Analyze & Calculate & Summary Data (BI)
- End users

Relational Database

Relational database Lifecycle divided into these stages:

- ERD
- Mapping
- Implementation of database
- Querying Data

Entity Relationship Modeling

Entity-Relationship Diagram (ERD): identifies information required by the business by displaying the relevant entities and the relationships between them.

Entity Relationship Modeling (Cont'd)

- In building a data model a number of questions must be addressed:
  - What entities need to be described in the model?
  - What characteristics or attributes of those entities need to be recorded?
  - Can an attribute or a set of attributes be identified that will uniquely identify one specific occurrence of an entity?
  - What associations or relationships exist between entities?

Definitions

- Entity - An entity is a thing that exists and is distinguishable -- an object, something in the environment. (Types of entities: Weak-Regular)

- Entity Instance - An instance is a particular occurrence of an entity. For example, each person is an instance of an entity, each car is an instance of an entity, etc.

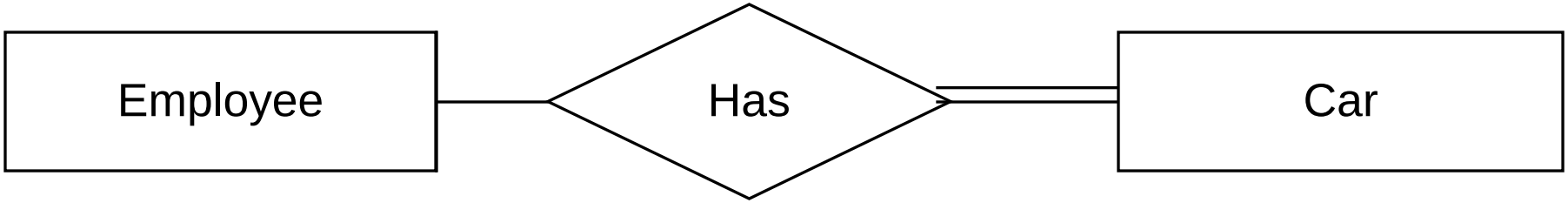
Types of Attributes

- Key
- Multi-valued
- Composite
- Derived

Key Attribute

- An attribute of an entity type for which each entity must have a unique value is called a key attribute of the entity type. For example, SSN of EMPLOYEE.
- A key attribute may be composite. For example, ID is a key of the applicant entity type with components (National_ID, Application_no).
- Candidate Key: An entity type may have more than one key.

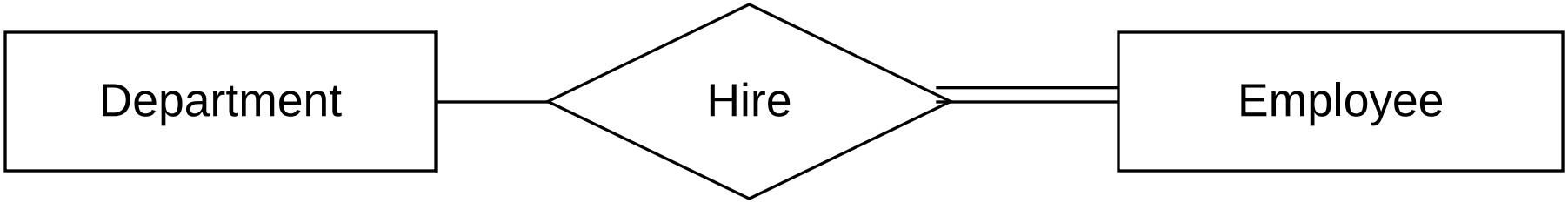
PARTICIPATION CONSTRAINT



-An Employee may have a car.

-A Car must be assigned to particular employee

PARTICIPATION CONSTRAINT



- A department may hire many employees (Zero or more)

- An employee must be employed by a department

(Department membership is Optional, Employee membership is Mandatory)

PARTICIPATION CONSTRAINT

- An employee MUST work for a department
  An employee entity can exist only if it participates in a WORKS_FOR relationship instance
  Thus its participation is TOTAL

  Only some employees manage departments
  The participation is PARTIAL

  A formal constraint: (min,max) where m, n are min and max number of times an entity participates in a relationship instance. For example, (0,10) means partial participation, and (1,max) means total participation.

Weak Entity Types

- An entity that does not have a key attribute
- A weak entity must participate in an identifying relationship type with an owner or identifying entity type
- Entities are identified by the combination of:
  - A partial key of the weak entity type
  - The particular entity they are related to in the identifying entity type

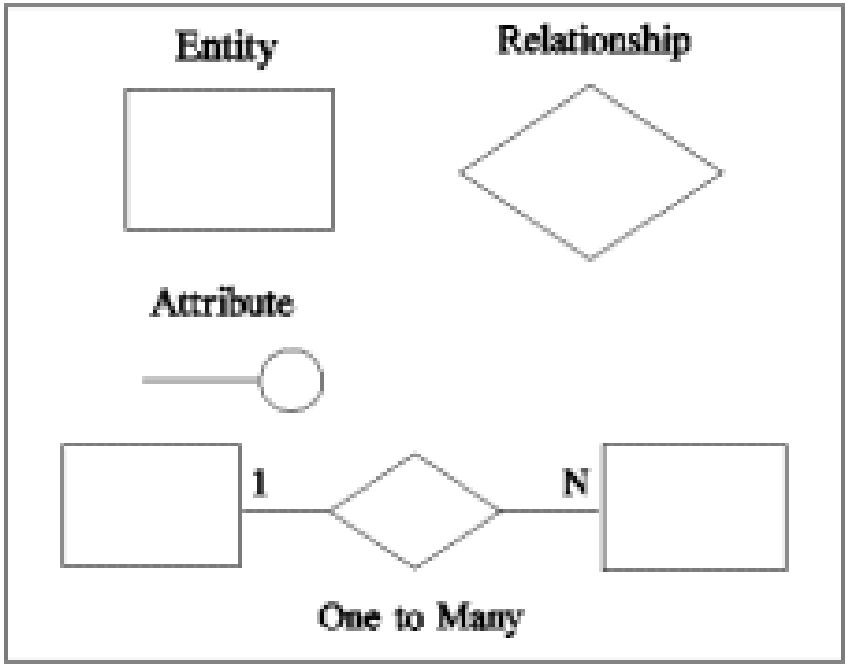
Relationships

- Relationships - A relationship is a connection between entity classes.

- The *cardinality* of a relationship indicates the number of instances in entity class E1 that can or must be associated with instances in entity class E2.

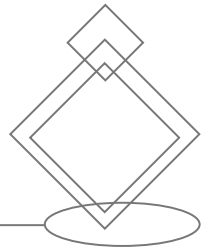
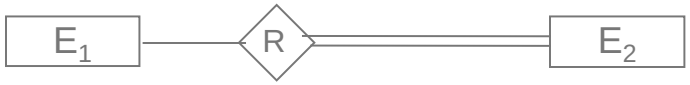
  - One-One Relationship - (citizen – passport ,
  - One-Many Relationship - (student-Advisor, Customer-Order)
  - Many- Many Relationship - (e.g. Student-Organization, Order-Products)
  - Recursive Relationships - A relationship in which the same entity participates more than once.

ERD Notations



- Rectangles represent ENTITY CLASSES
- Circles represent ATTRIBUTES
- Diamonds represent RELATIONSHIPS
- Arcs - Arcs connect entities to relationships. Arcs are also used to connect attributes to entities. Some styles of entity-relationship diagrams use arrows and double arrows to indicate the one and the many in relationships. Some use forks etc.
- Underline - Key attributes of entities are underlined.

SUMMARY OF ERD NOTATION

Symbol	Meaning
	ENTITY TYPE
	WEAK ENTITY TYPE
	RELATIONSHIP TYPE
	IDENTIFYING RELATIONSHIP TYPE
	ATTRIBUTE
	KEY ATTRIBUTE
	MULTIVALUED ATTRIBUTE
	COMPOSITE ATTRIBUTE
	DERIVED ATTRIBUTE
	TOTAL PARTICIPATION OF E_2 IN R
	CARDINALITY RATIO 1:N FOR E_1:E_2 IN R
	STRUCTURAL CONSTRAINT (min, max) ON PARTICIPATION OF E IN R

An Example

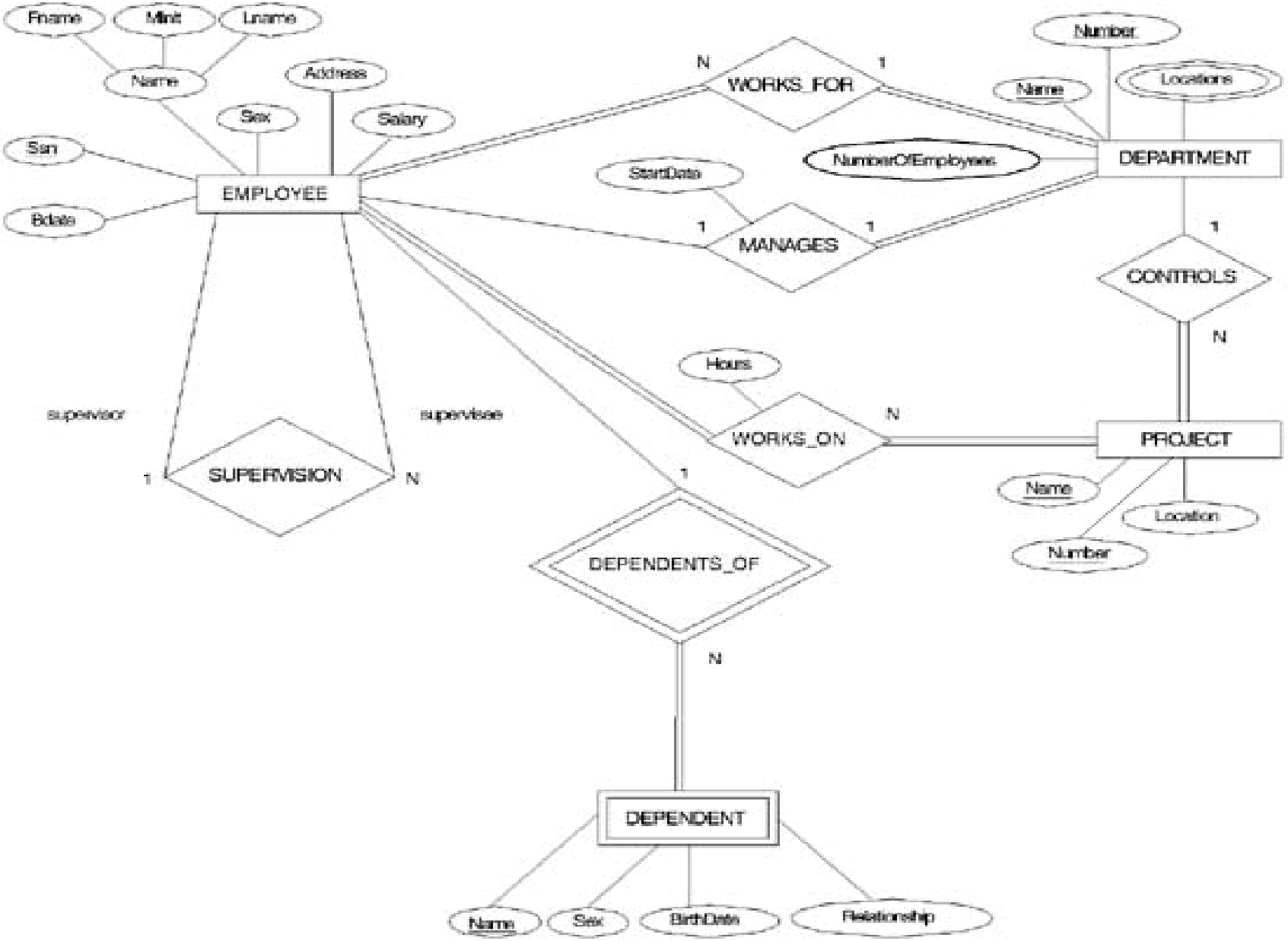
- A company is organized into departments. Each department has a unique name, a unique number, and a particular employee who manages the department. A department may have several locations.

- 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

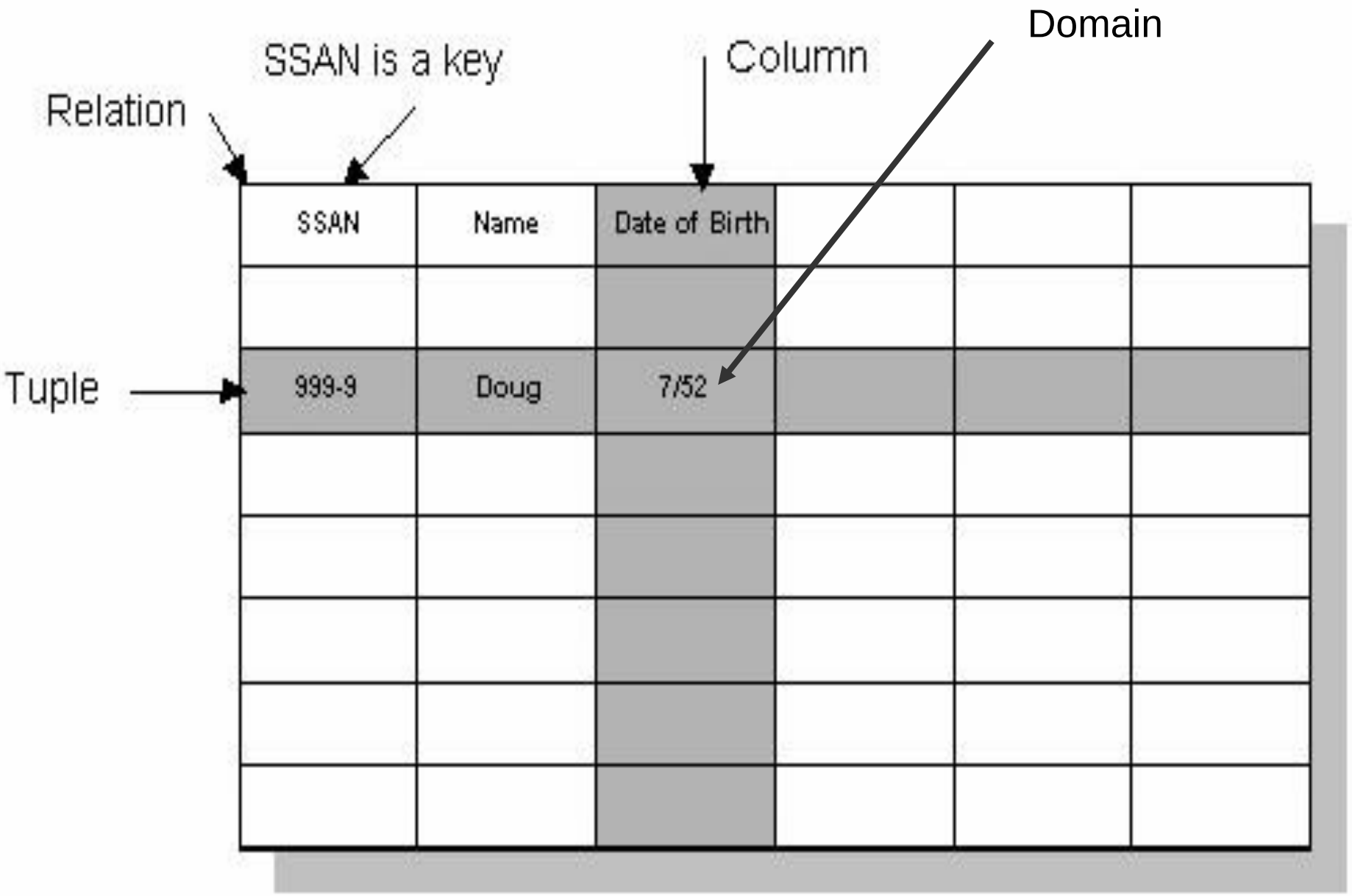
An Example (Cont'd)

- 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.


Fname
Minit
Lname
Name
Address
Sex
Salary
Ssn
Bdate
EMPLOYEE
N
WORKS_FOR
1
Number
Name
Locations
DEPARTMENT
NumberOfEmployees
StartDate
1
MANAGES
1
1
CONTROLS
N
Hours
N
WORKS_ON
PROJECT
Name
Location
Number
supervisor
supervisee
1
SUPERVISION
N
1
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
BirthDate
Relationship

Relational Database



ER-to-Relational Mapping

Step 1: Mapping of Regular Entity Types

Step 2: Mapping of Weak Entity Types

Step 3: Mapping of Binary 1:1 Relation Types

Step 4: Mapping of Binary 1:N Relationship Types.

Step 5: Mapping of Binary M:N Relationship Types.

Step 6: Mapping of Multi-valued attributes.

Step 1: Mapping of Regular Entity Types

- Create table for each entity type

- Choose one of key attributes to be the primary key

Step 2: Mapping of Weak Entity Types

- 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)

Step 3: Mapping of Binary 1:1 Relation Types

- 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.

Step 4: Mapping of Binary 1:N Relationship Types.

- Add FK to N-side table

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

Step 5: Mapping of Binary M:N Relationship Types.

- Create a new third table

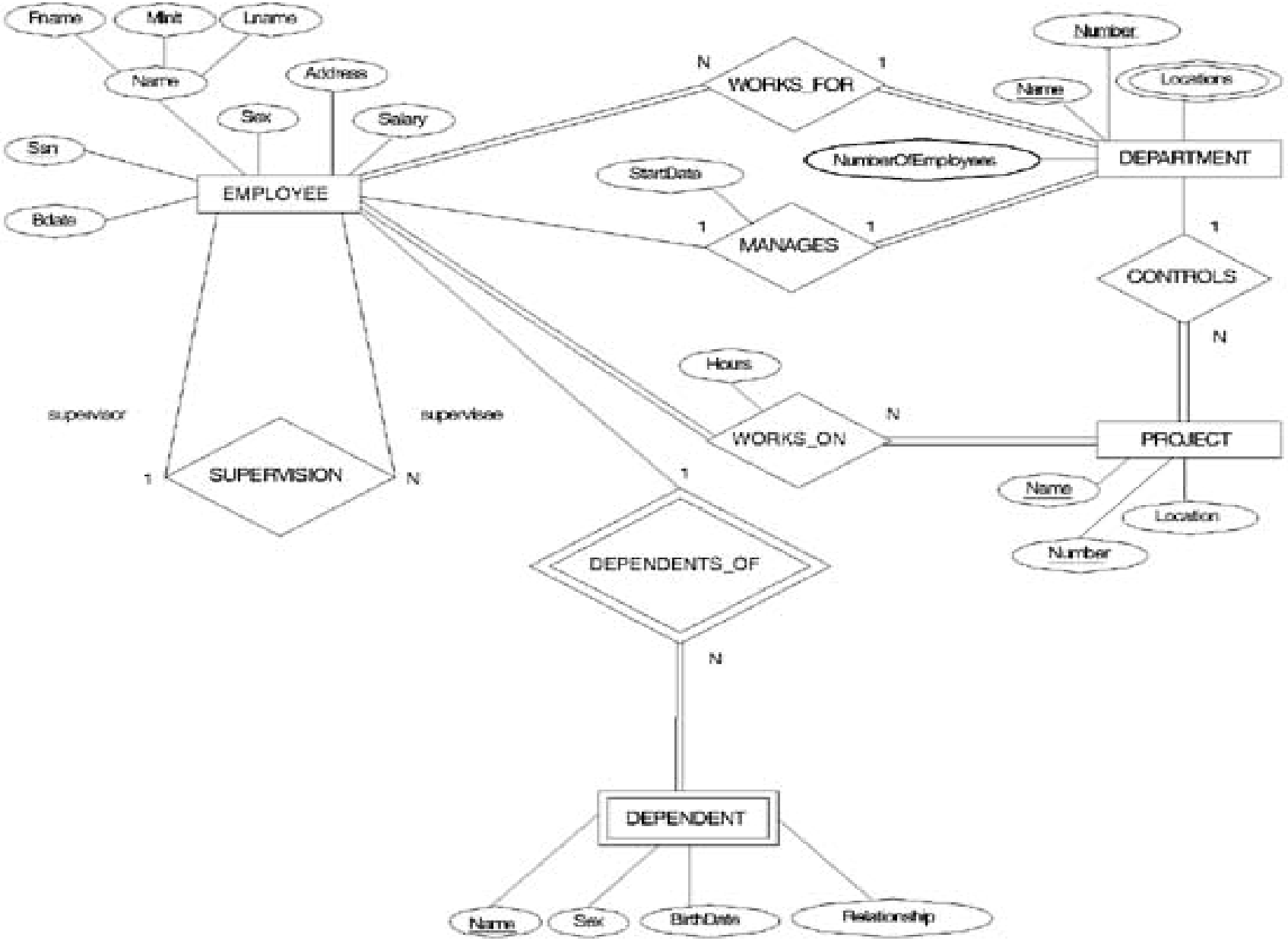
- Add FKs to the new table for both parent tables

- Add simple attributes of relationship to the new table if any .

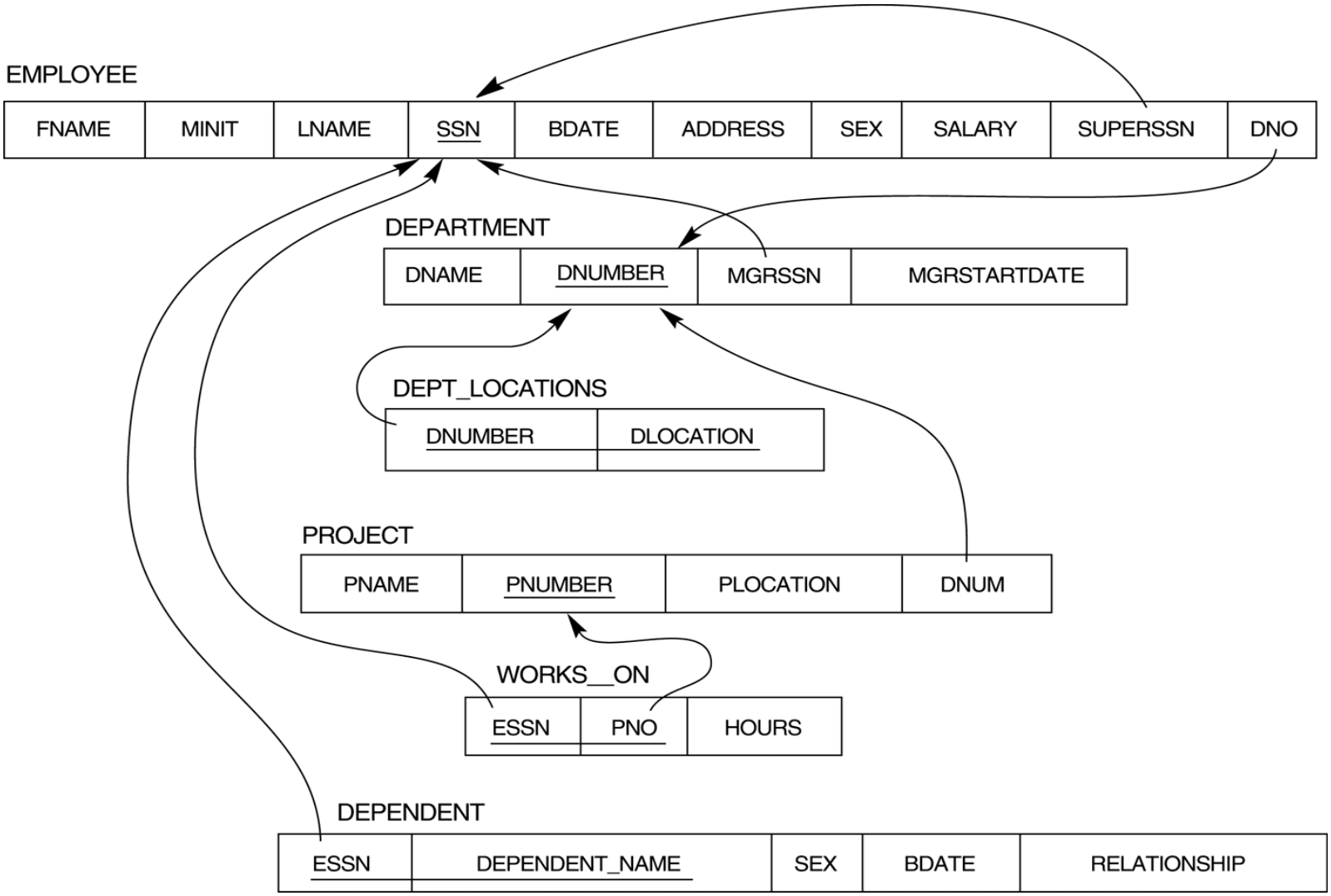
Step 6: Mapping of Multi-valued attributes.

- Create new table for each multi-valued attribute

- Table will include two columns.
  one for multi-valued attribute + FK column.


Fname
Minit
Lname
Name
Address
Sex
Salary
Ssn
Bdate
EMPLOYEE
N
WORKS_FOR
1
Number
Name
Locations
DEPARTMENT
NumberOfEmployees
StartDate
1
MANAGES
1
1
CONTROLS
N
Hours
N
WORKS_ON
PROJECT
Name
Location
Number
supervisor
supervisee
1
SUPERVISION
N
1
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
BirthDate
Relationship

Mapping Result



Normalization

- Why do we use Normalization?

Figure 14.4 Example relations for the schemas in Figure 14.3 that result from applying NATURAL JOIN to the relations in Figure 14.2. These may be stored as base relations for performance reasons.

EMP DEPT

ENAME	SSN	BDATE	ADDRESS	DNUMBER	DNAME	DMGRSSN
Smith,John B.	123456789	1965-01-09	731 Fondren,Houston,TX	5	Research	333445555
Wong,Franklin T.	333445555	1955-12-08	638 Voss,Houston,TX	5	Research	333445555
Zelaya, Alicia J.	999887777	1968-07-19	3321 Castle,Spring,TX	4	Administration	987654321
Wallace,Jennifer S.	987654321	1941-06-20	291 Berry,Bellaire,TX	4	Administration	987654321
Narayan,Ramesh K.	666884444	1962-09-15	975 FireOak,Humble,TX	5	Research	333445555
English,Joyce A.	453453453	1972-07-31	5631 Rice,Houston,TX	5	Research	333445555
Jabbar,Ahmad V.	987987987	1969-03-29	980 Dallas,Houston,TX	4	Administration	987654321
Borg,James E.	888665555	1937-11-10	450 Stone,Houston,TX	1	Headquarters	888665555

EMP PROJ

SSN	PNUMBER	HOURS	ENAME	PNAME	PLOCATION
123456789	1	32.5	Smith,John B.	ProductX	Bellaire
123456789	2	7.5	Smith,John B.	ProductY	Sugarland
666884444	3	40.0	Narayan,Ramesh K.	ProductZ	Houston
453453453	1	20.0	English,Joyce A.	ProductX	Bellaire
453453453	2	20.0	English,Joyce A.	ProductY	Sugarland
333445555	2	10.0	Wong,Franklin T.	ProductY	Sugarland
333445555	3	10.0	Wong,Franklin T.	ProductZ	Houston
333445555	10	10.0	Wong,Franklin T.	Computerization	Stafford
333445555	20	10.0	Wong,Franklin T.	Reorganization	Houston
999887777	30	30.0	Zelaya,Alicia J.	Newbenefits	Stafford
999887777	10	10.0	Zelaya,Alicia J.	Computerization	Stafford
987987987	10	35.0	Jabbar,Ahmad V.	Computerization	Stafford
987987987	30	5.0	Jabbar,Ahmad V.	Newbenefits	Stafford
987654321	30	20.0	Wallace,Jennifer S.	Newbenefits	Stafford
987654321	20	15.0	Wallace,Jennifer S.	Reorganization	Houston
888665555	20	null	Borg,James E.	Reorganization	Houston

Normalization Avoids

- Duplication of Data
- Insert Anomaly
- Delete Anomaly
- Update Anomaly
- Frequent Null Values

Functional dependency

- a constraint between two attributes (columns) or two sets of columns
- A → B if “for every valid instance of A, that value of A uniquely determines the value of B”
- or …A →B if “there exists at most one value of B for every value of A”

… functional dependency

- some examples
- social security number determines employee name

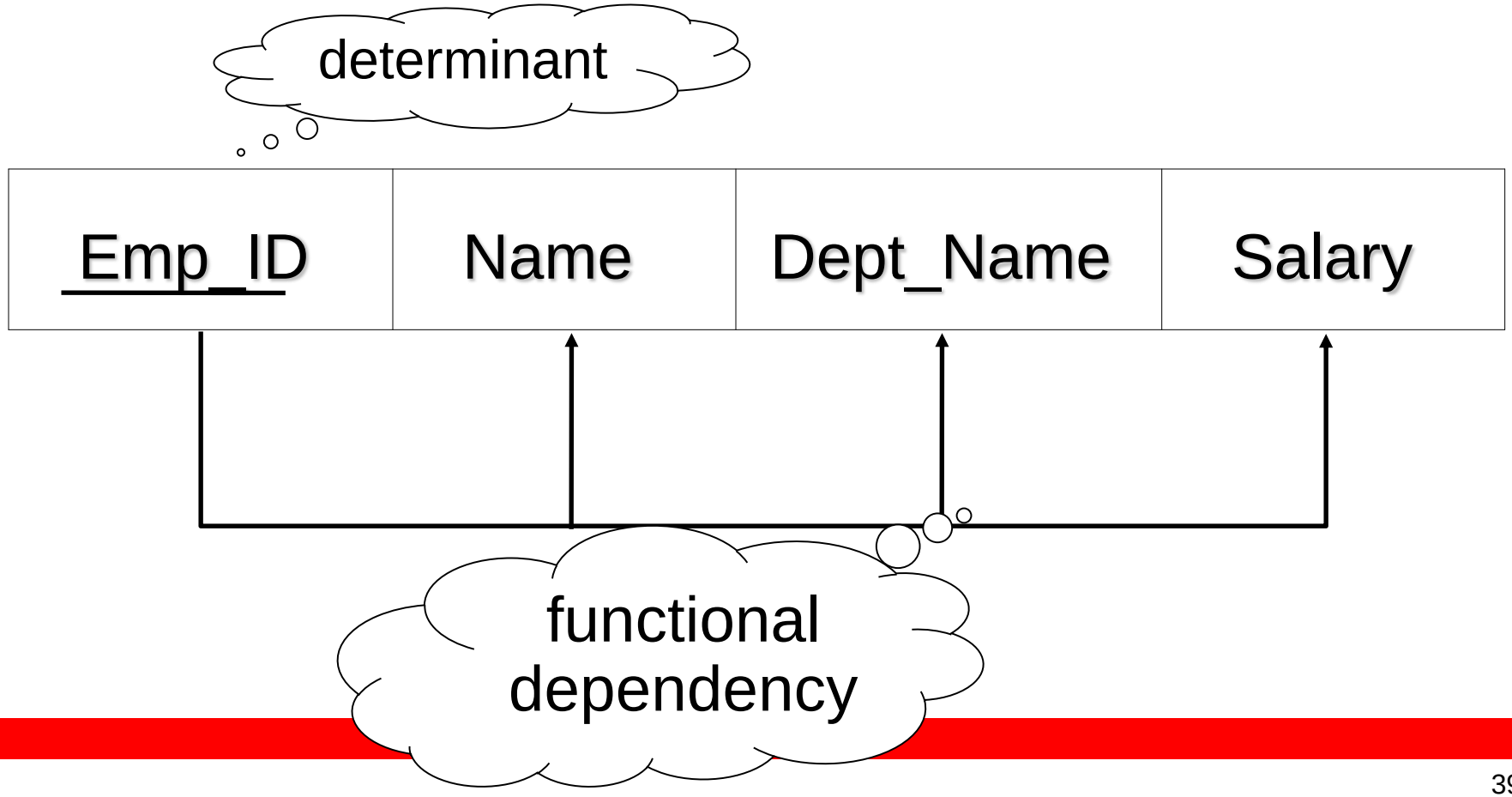
  SSN -> ENAME
- project number determines project name and location

  PNUMBER -> {PNAME, PLOCATION}
- employee ssn and project number determines the hours per week that the employee works on the project

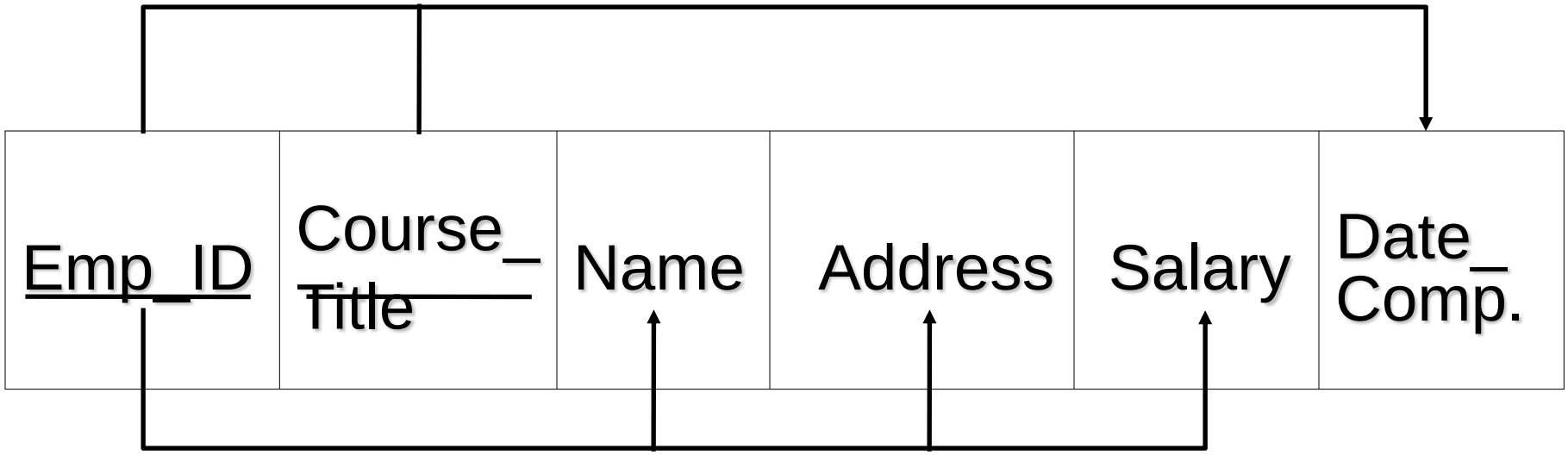
  {SSN, PNUMBER} -> HOURS
- So functional dependency is the technical term for *determines*

keys and dependencies

EMPLOYEE1 (Emp_ID, Name, Dept_Name, Salary)



EMPLOYEE2 (Emp_ID, Course_Title, Name, Address, Salary, Date_Completed)



not fully functionally dependant on the primary key

Definition

- Normalization: The process of decomposing unsatisfactory "bad" relations by breaking up their attributes into smaller relations

- Normal form: Condition using keys and FDs of a relation to certify whether a relation schema is in a particular normal form

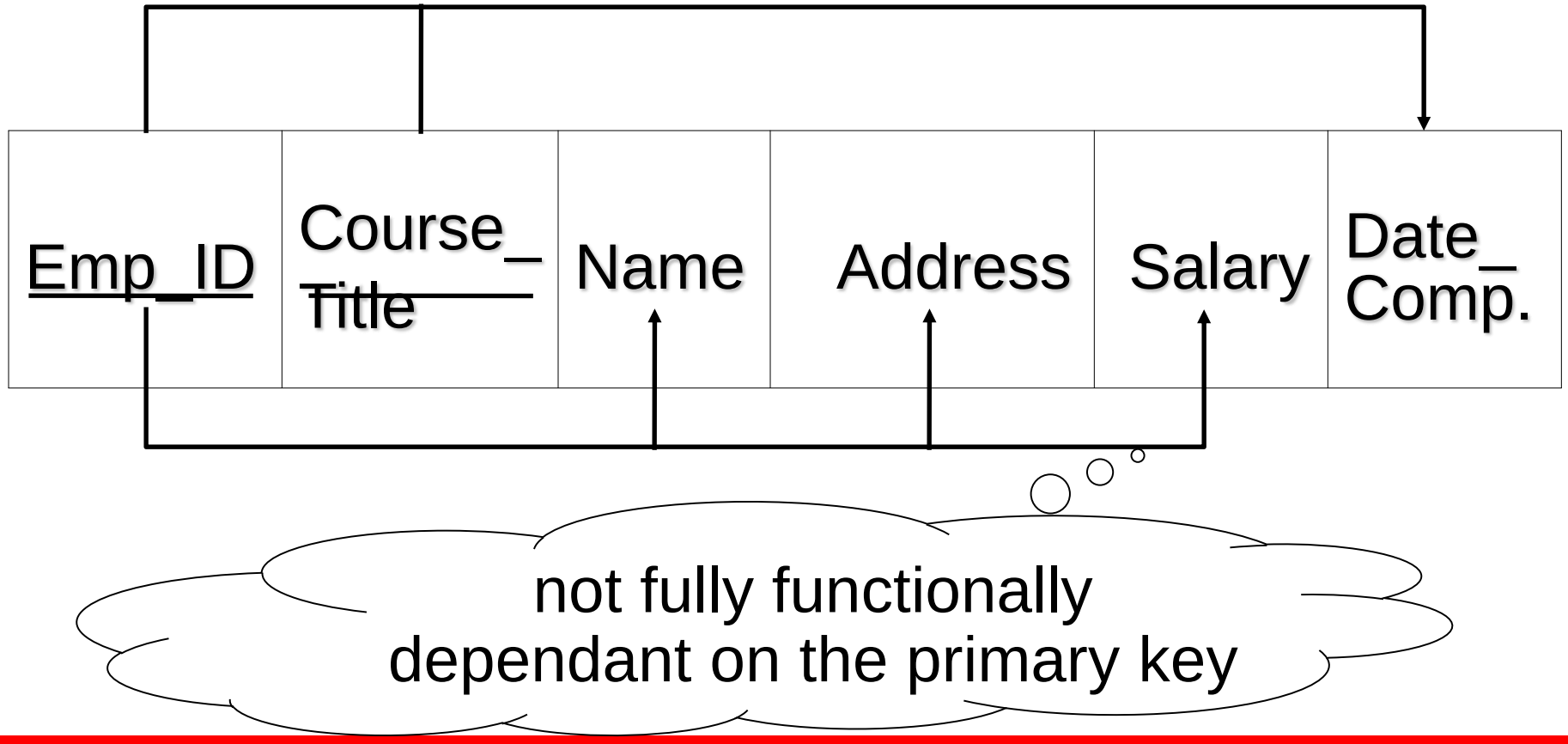
1NF

- relation is in first normal form if it contains no multivalued or composite attributes
- remove repeating groups to a new table as already demonstrated, “carrying” the PK as a FK
- All columns (fields) must be atomic
  - Means : no repeating items in columns

2NF

- a relation is in second normal form if it is in first normal form AND every nonkey attribute is fully functionally dependant on the primary key
- i.e. remove partial functional dependencies, so no nonkey attribute depends on just part of the key

EMPLOYEE2 (Emp_ID, Course_Title, Name, Address, Salary, Date_Completed)



2NF

- a relation is in 2NF if it is in 1NF and any one of these is true:
  - the PK consists of only 1 attribute
  - all attributes are part of the PK (no nonkey attributes)
  - every non key attribute is functionally dependant on the whole PK

1NF →2NF

EMPLOYEE2 (Emp_ID, Course_Title, Name, Address, Salary, Date_Completed) →

EMPLOYEE1 (Emp_ID, Name, adress, Salary)

and

EMP_COURSE (Emp_ID, Course_Title, Date_Completed)

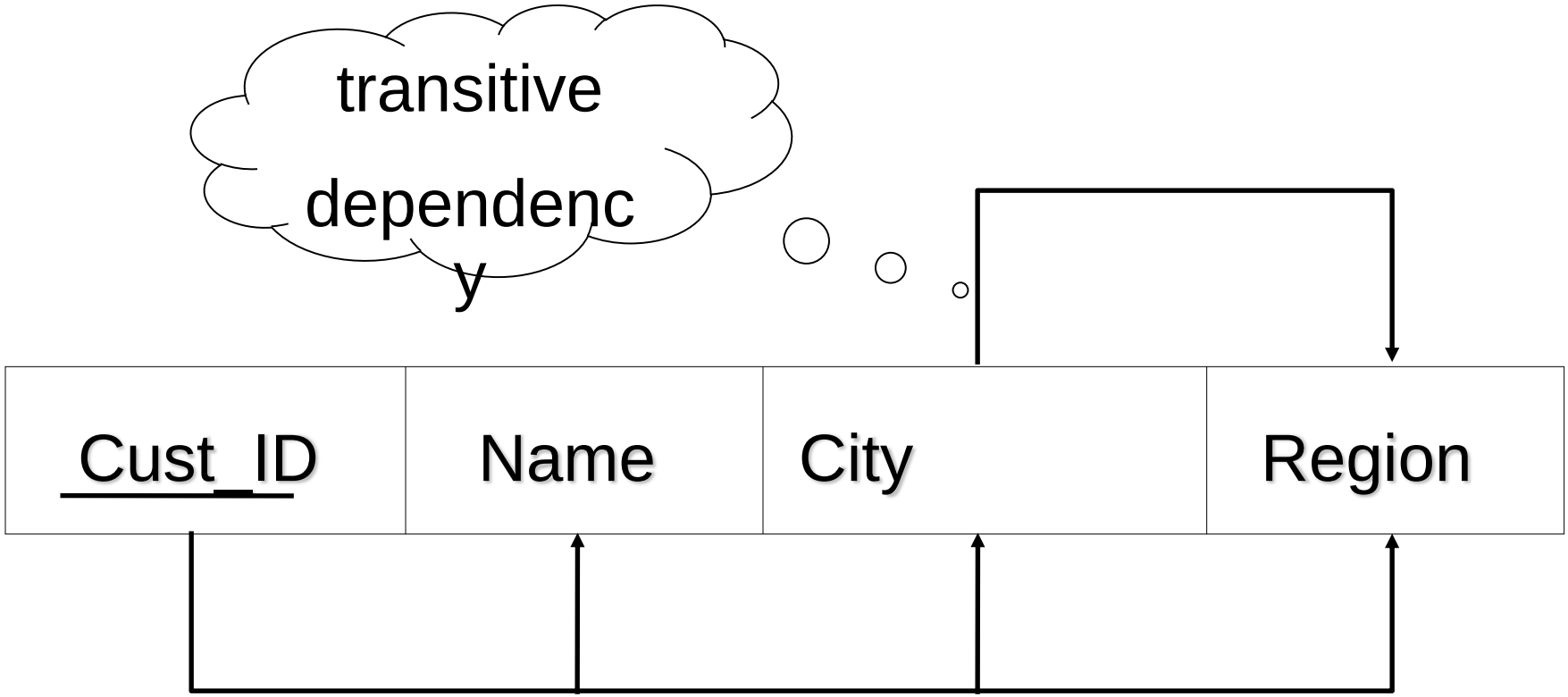
EMPLOYEE1 satisfies condition1

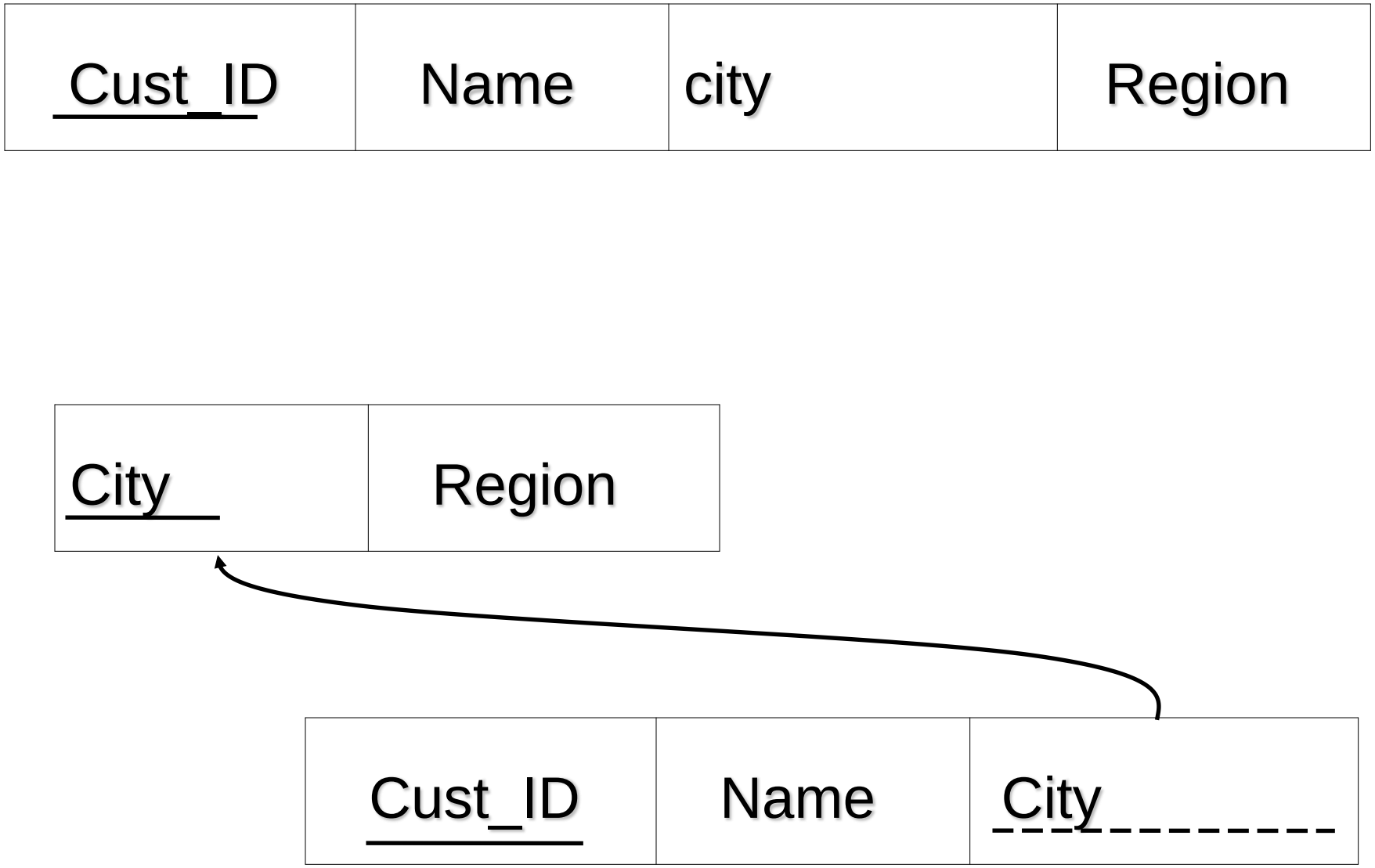
- EMP_COURSE satisfies condition3

3NF

- a relation is in third normal form if it is in 2NF, AND no *transitive dependencies* exist
- transitive dependency is a functional dependency between nonkey attributes

transitive dependency




Cust_ID
Name
city
Region
City
Region
Cust_ID
Name
City

ITI Example

ITI Students Sheet

Platform Name : SWE Platform Description: Software Engineering

Graduate Profile: ALL

Appno	Name	F-code	Faculty	Major	Address	Telno	Found. Grade	Total Att. Hrs	Start date
123	Ahmed	SC-phy	Science	Physics	Haram	3386842 0	A	600	14 Sep
124	Mona	Eng-cs	Engineering	Computer	Dokki	3389745 5,3389 7445	B	591	15 Sep
127	Ali	Com-ac	Commerce	Accounting	Nasr City	2241593 9	A	550	21 Sep
223	Karim	Med-bio	Medicine	Biochemistry	Sheraton	2286845 6	C	600	14 Sep

1st Normal Form

- Normalization: First Normal Form
- Separate Repeating Groups into New Tables.
- *Repeating Groups* Fields that may be repeated several times for one document/entity
- Create a new table containing the repeating data
- The primary key of the table (repeating group) is always a composite key; Usually document number and a field uniquely describing the repeating line, like an item number.

1NF :

- Platform :pfname , pfdesc , pfgraduate
- Students: pfname, appno, name , faculty , major , address, Foundgrade, attd , start_date
- Std_Tel: appno, telno

2NF

- Students: appno, name , f-code, faculty , major , address
- Students_pf: pfname,appno, Foundgrade, attd , start_date

- Unchanged Tables
- Platform :pfname , pfdesc , pfgraduate
- Std_Tel: appno, telno

3NF

- Students: appno, name , f-code, address
- Fac_majors: f-code, faculty , major

- Unchanged Tables
- Platform :pfname , pfdesc , pfgraduate
- Std_Tel: appno, telno
- Students_pf: pfname,appno, Foundgrade, attd , start_date