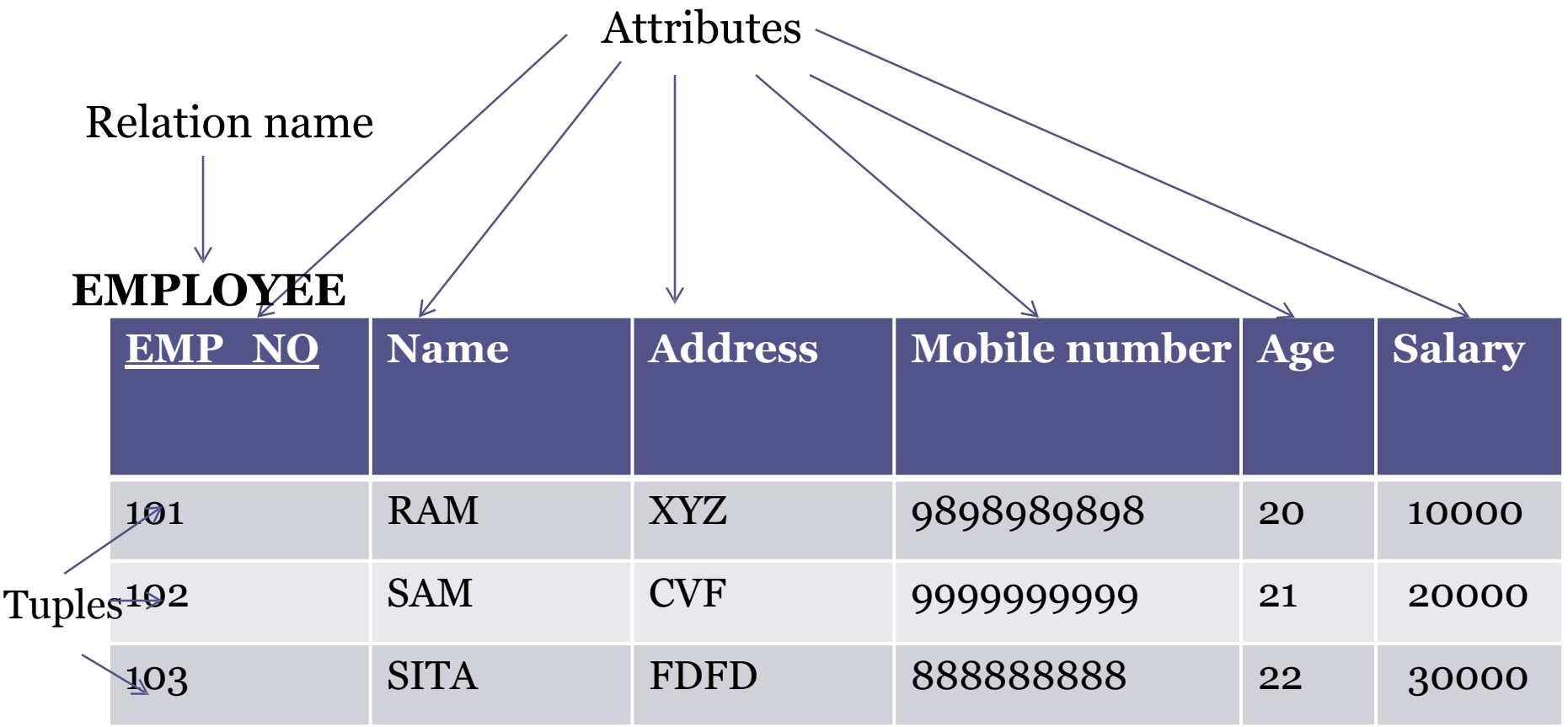
MODULE 2: RELATIONAL MODEL PART 1

SYLLABUS

- Structure of Relational Databases - Integrity Constraints, Synthesizing ER diagram to relational schema
- Introduction to Relational Algebra - select, project, cartesian product operations, join - Equi-join, natural join. Query examples
- Introduction to Structured Query Language (SQL), Data Definition Language (DDL), Table definitions and operations – CREATE, DROP, ALTER, INSERT, DELETE, UPDATE.

Relational data model

- First introduced by Ted Codd in 1970.
- Represent model represents database as a collection of relations.
- Relation is a table which has values and rows in table is a collection of related data values
- Each row represents a real world entity & its related values.
- Row in relational table is called a tuple, column header is attribute and table is a relation


Attributes
Relation name
EMPLOYEE
EMP_NO
Name
Address
Mobile number
Age
Salary
101
RAM
XYZ
9898989898
20
10000
192
SAM
CVF
9999999999
21
20000
103
SITA
FDFD
888888888
22
30000
Tuples

- In the relational model, all data is logically structured within **relations** (also called **table**)
- Informally a relation may be viewed as a named two-dimensional table representing an entity set.
- A relation has a fixed number of named columns and variable number of rows.

Components of relational database

- The main components of relational database structure are as follows:

1. **Domain**
2. **Tuples (rows)**
3. **Attributes (Columns)**
4. **Keys**
5. **Relations (Tables)**

Domain

- It has three parts
  - Name
  - Data type
  - Format
- A Domain is a set of atomic values.
- Atomic means each value in the domain is indivisible to the relational model.

- A **domain** has a logical definition: e.g.“USA_phone_numbers” are the set of 10 digit phone numbers valid in the U.S.
- A domain may have a data-type or a format defined for it. The USA_phone_numbers may have a format: (ddd)-ddd-dddd where each d is a decimal digit.
- E.g., Dates have various formats such as month name, date, year or yyyy-mm-dd, or dd mm,yyyy etc

Figure 2.2 shows the structure of an instance or extension, of a relation called EMPLOYEE. The EMPLOYEE relation has six attributes (field items), namely EMP-NO, LAST-NAME, FIRST-NAME, DATA-OF-BIRTH, SEX, TEL-NO and SALARY. The extension has seven tuples (records). Each attribute contains values drawn from a particular domain.

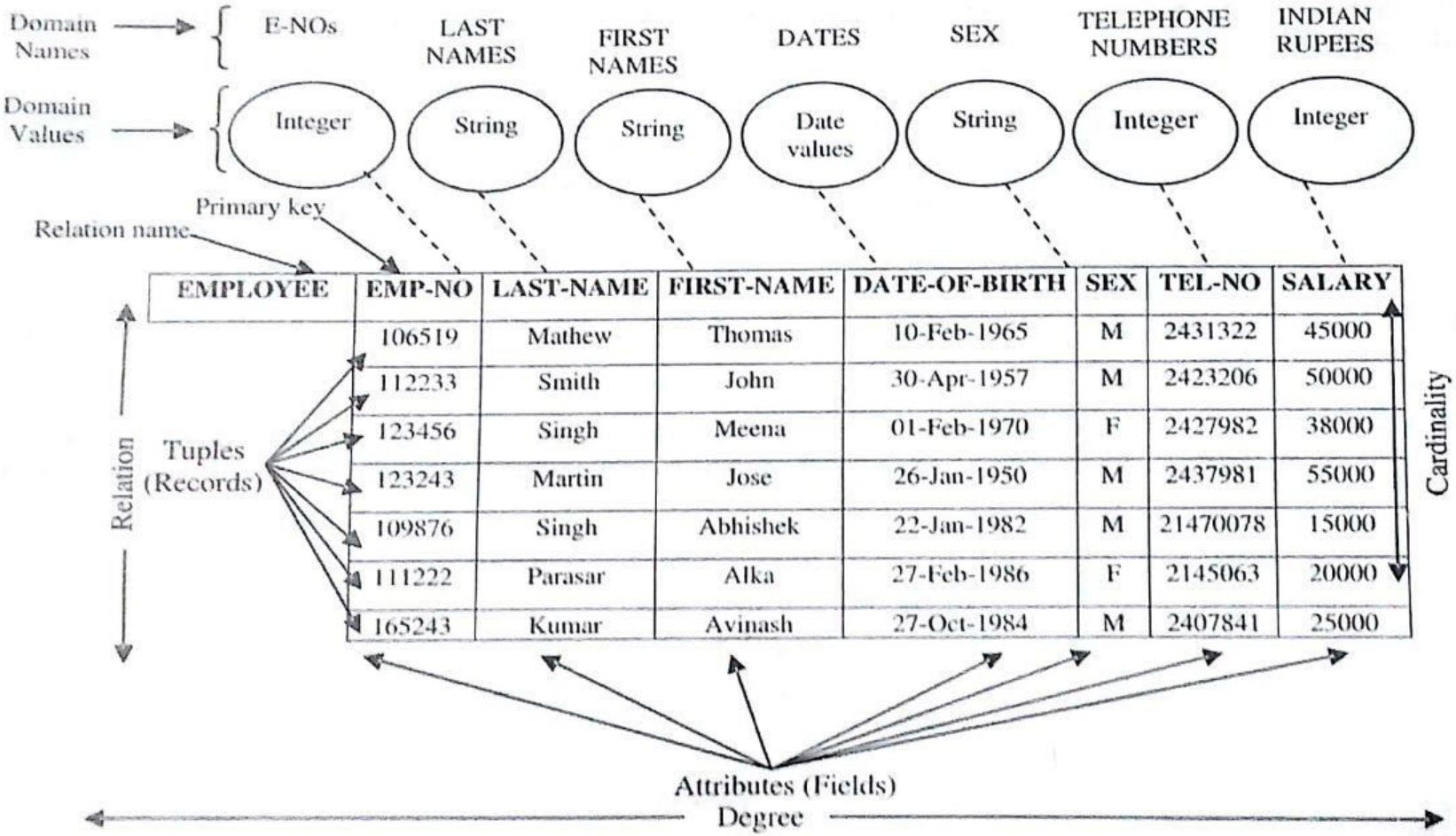
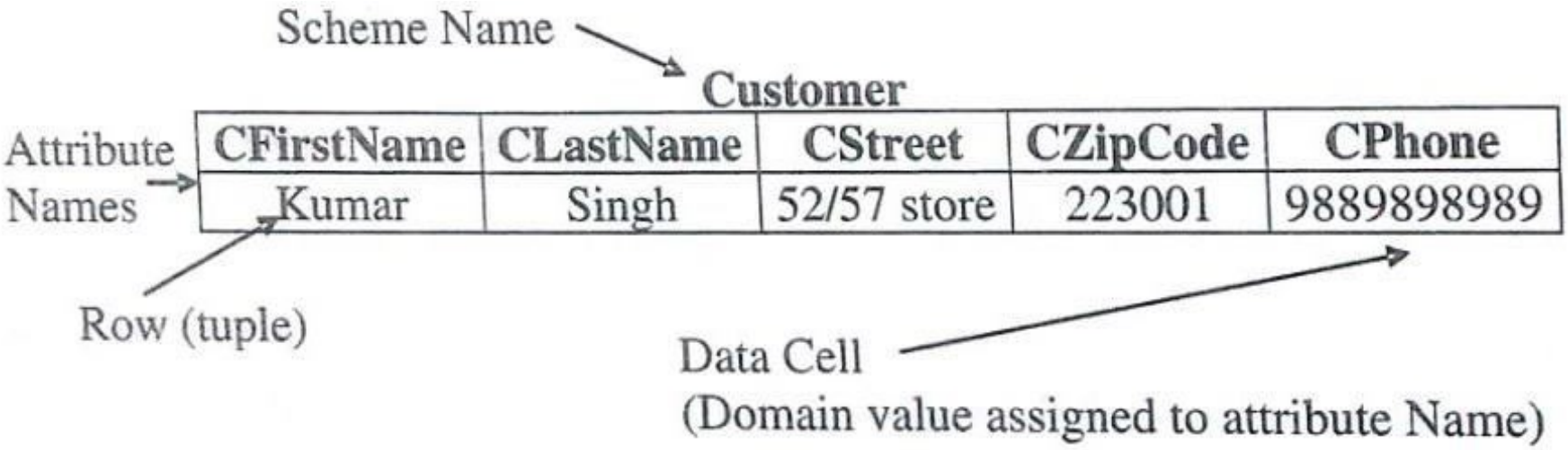


Figure 2.2: EMPLOYEE Relation

Tuples (rows)

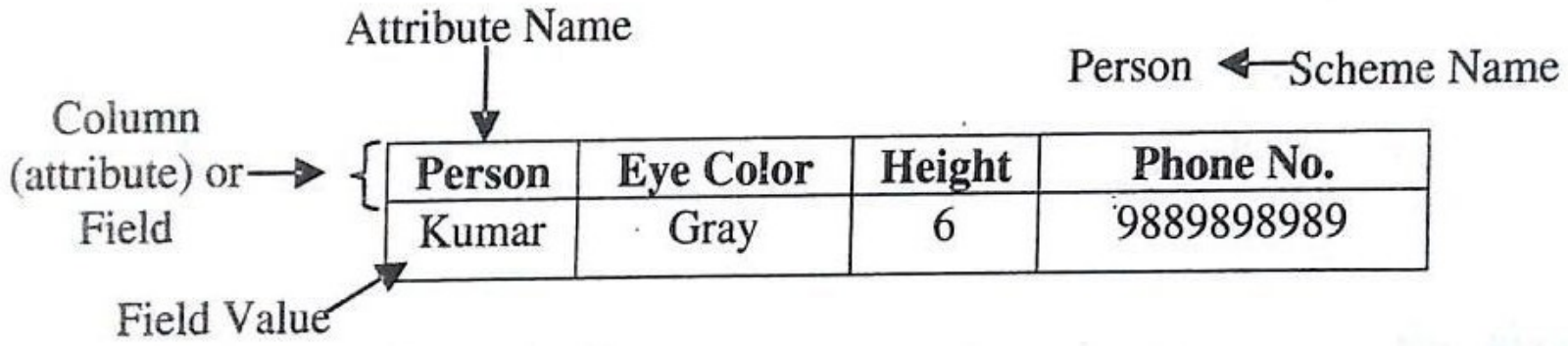
- A tuple is an ordered set of values
- Tuple is a portion of a table containing data that described only entity, relationship, or object
- Also known as record
- Each value is derived from an appropriate domain.



- **<Kumar, Singh, 52/57 store, 223001, 9889898989>** is a tuple belonging to the CUSTOMER relation.

Attributes (Columns)

- Columns in a table are also called **attributes** or **fields** of the relation.
- A single cell in a table is called field value, attribute value or data element.
- For example, for the entity person, attributes could include eye colour and height.



Key of a Relation

- Each row has a value of a data item (or set of items) that uniquely identifies that row in the table
  - Called the *key*
- Sometimes row-ids or sequential numbers are assigned as keys to identify the rows in a table
  - Called *artificial key* or *surrogate key*

Relations (Tables)

- A table of values
- A relation may be thought of as a **set of rows**.
- A relation may alternately be thought of as a **set of columns**.
- That is a table is perceived as a two-dimensional structure composed of rows and columns.
- Each row represents a fact that corresponds to a real-world **entity** or **relationship**.
- Each row has a value of an item or set of items that uniquely identifies that row in the table.

Schema of a Relation

- It is basically an outline of how data is organized
- It is denoted by *R* (A1, A2, ...An)
  - Here R is relation name and
  - it has some attributes A1 to An
- Each attribute have some domain and it is represented by dom(Ai)
- Relation schema is used to describe a relation
- Each attribute has a **domain** or a set of valid values.
  - For example, the domain of PhoneNo is 10 digit numbers.

Degree of a relation

- Degree of a relation is number of attributes in a relation
- Eg. STUDENT(Id, Name, Age, Departmentno)
  - Has degree 4
- Datatype of each the attribute can also be represented.
- Eg. STUDENT(Id:Integer, Name:String, Age:integer, Departmentno:integer)

Relation State

- The **relation state** is a subset of the Cartesian product of the domains of its attributes
  - each domain contains the set of all possible values the attribute can take.

$$r(R) \subseteq (dom(A1) X\ dom(A2)\ X\ \ldots . X\ dom(An))$$

- Example: attribute Cust-name is defined over the domain of character strings of maximum length 25
  - dom(Cust-name) is varchar(25)

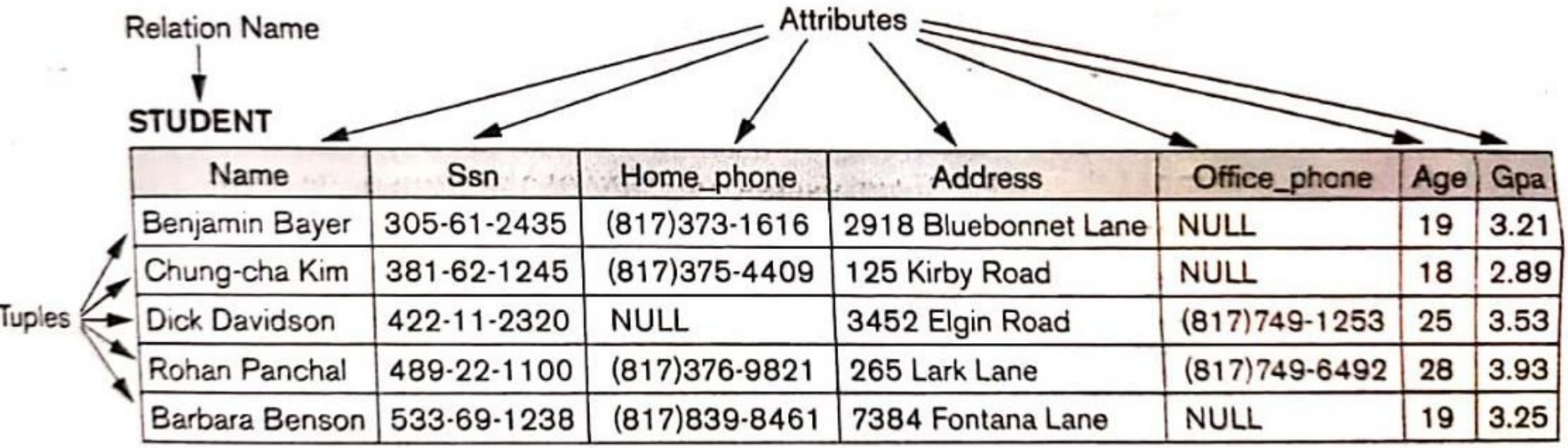
- Cartesian product specifies all possible combination of values from underlying domains
  - Let R(A1, A2) be a relation schema:
    - Let dom(A1) = {0,1}
    - Let dom(A2) = {a,b,c}
  - Then: dom(A1) X dom(A2) is all possible combinations:

    {<0,a> , <0,b> , <0,c>, <1,a>, <1,b>, <1,c> }

  - The relation state r(R) ⊂ dom(A1) X dom(A2)
  - For example: r(R) could be {<0,a> , <0,b> , <1,c> }

Current relation state

- Reflects only the valid tuples that represent a particular state of real world
- As the state of real world changes, so does the relation state, by being transformed into another relation state



Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	(817)373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	(817)375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	(817)749-1253	25	3.53
Rohan Panchal	489-22-1100	(817)376-9821	265 Lark Lane	(817)749-6492	28	3.93
Barbara Benson	533-69-1238	(817)839-8461	7384 Fontana Lane	NULL	19	3.25

Figure 3.1
The attributes and tuples of a relation STUDENT.

Relational Model Notations

- A relation schema of degree n is denoted by:
  R(A1, A2, ……., An), where R – name, Ai - Attributes
- The uppercase letters Q, R, S denotes relation name.
- Relation state is denoted as:
  r(R) = {t1, t2, …, tn} where each ti is a tuple
  $r(R) \subset dom(A1) \times dom(A2) \times \ldots \times dom(An)$
- A tuple in a relation is denoted by:
  $ti = \langle v1, v2, \ldots, vn \rangle$ where each $vj \in dom(Aj)$
- The lowercase letters t, u, v denote tuples.

Definition Summary

Informal Terms	Formal Terms
Table	Relation
Column Header	Attribute
All possible Column Values	Domain
Row	Tuple
Table Definition	Relation Schema
Populated Table	Relation State

Characteristics of Relation

1) Ordering of tuples in a relation r(R):

- The tuples are *not considered to be ordered*, even though they appear to be in the tabular form.

2) Ordering of attributes in a relation schema R (and of values within each tuple):

- We will consider the attributes in R(A1, A2, ..., An) and the values in t=<v1, v2, ..., vn> to be ordered .

Figure 5.2
The relation STUDENT from Figure 5.1 with a different order of tuples.

STUDENT

Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	749-1253	25	3.53
Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	NULL	19	3.25
Rohan Panchal	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
Chung-cha Kim	381-62-1245	375-4409	125 Kirby Road	NULL	18	2.89
Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	NULL	19	3.21

3) Values & nulls in a tuple:

- All values are considered atomic (indivisible).
- Each value in a tuple must be from the domain of the attribute for that column
  - If tuple t = <v1, v2, ..., vn> is a tuple (row) in the relation state r of R(A1, A2, ..., An)
  - Then each *vi* must be a value from *dom(Ai)*

- A special **null** value is used to represent values that are unknown or inapplicable to certain tuples.

4) Interpretation(Meaning) of a Relation:

- STUDENT(Name, SSN, home_phone, Age, GPA)
- From this schema name, we interpret that this relation stores data about students.
- Similarly, from a tuple <“Smith”, 17, 888888,18,8>, we can interpret that there is a student whose name is “Smith”.

Relational Model Constraints

- Constraints are **conditions** that must hold on **all** valid relation states.
- There are *three* main types of constraints in the relational model:
  - Inherent model-based constraints/ Implicit
  - Schema-based constraints/ Explicit
  - Application based constraints/ Semantic

Inherent model-based Constraints

- Constraints that are inherent/ inbuilt in the data model.
- Eg: A relation cannot have duplicate tuples.

Application based Constraints

- Cannot be expressed in a schema.
- Defined in application programs.

Schema-based Constraints

- Domain constraint
- Key constraint
- Constraints on NULLs – whether permitted or not
- Entity Integrity constrains
- Referential Integrity constraints

Key Constraints

- Primary key
- Candidate key
- Secondary/ Alternate key
- Composite key
- Foreign key
- Super key

- Example: Consider a STUDENT relation schema:
  - STUDENT(Ssn, Name, Age, URegNo, DeptId)
  - Candidate key – {Ssn, URegNo}
  - Primary key – Ssn
  - Secondary key – URegNo
  - Composite key – {State, No}
  - Foreign key – DeptId
  - Super key – {Ssn, Name, Age}

- In general:
  - Any *key* is a *superkey* (but not vice versa)
  - Any set of attributes that *includes a key* is a *superkey*
  - A *minimal* superkey is also a key

Entity-Integrity Constraints

- States that no primary key value can be NULL.
- Primary key should be unique.

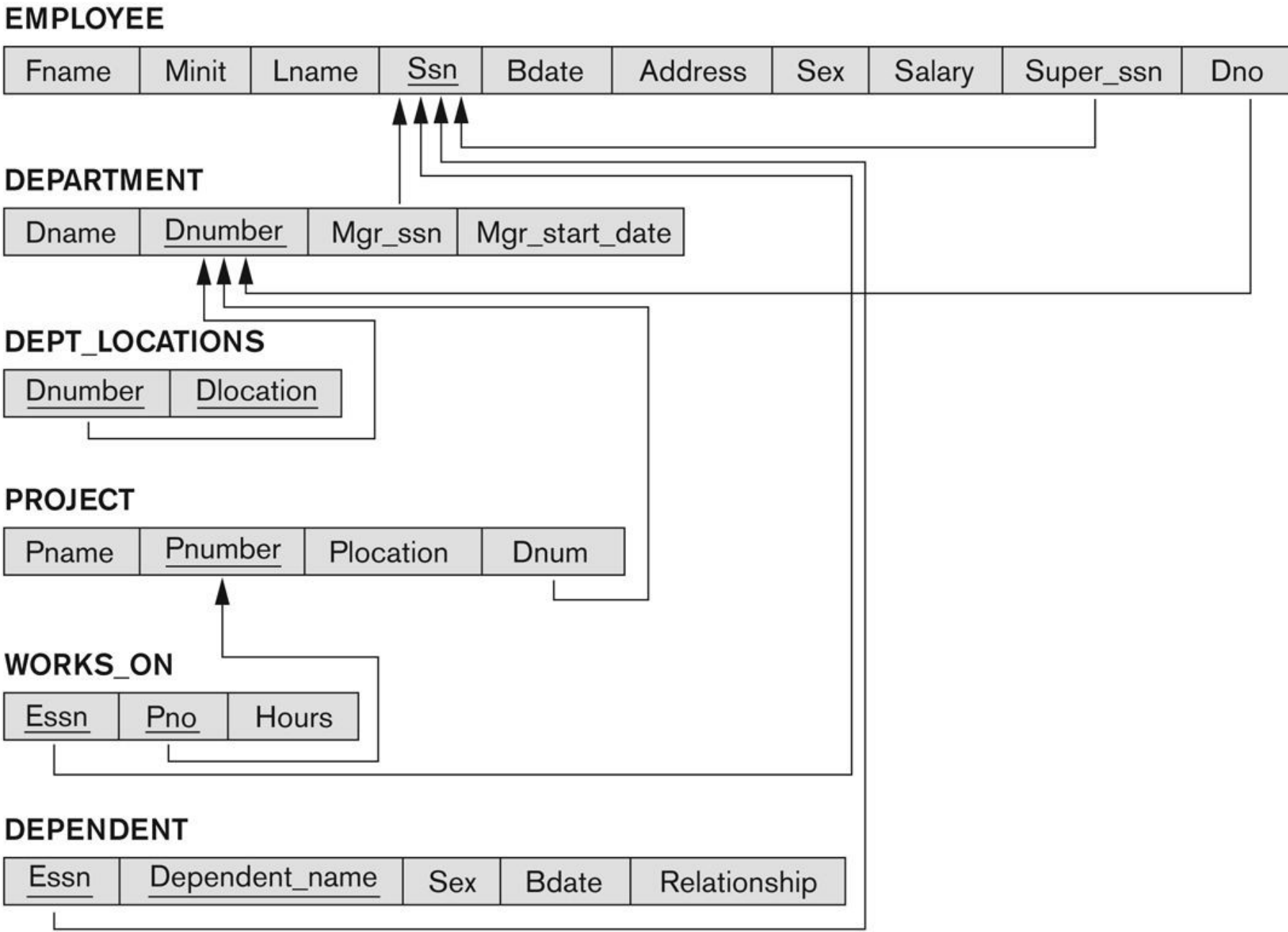
Referential-Integrity Constraints

- It is specified between 2 relations.
- States that foreign key must contain a null value or a valid primary key value.

Displaying a relational database schema and its constraints

- Each relation schema can be displayed as a row of attribute names.
- The name of the relation is written above the attribute names.
- The primary key attribute (or attributes) will be underlined.
- A foreign key (referential integrity) constraint is displayed as a directed arrow from the foreign key attribute to the primary key attribute of referenced table.

Figure 5.7
Referential integrity constraints displayed on the COMPANY relational database schema.



Synthesizing ER diagram to relational schema

- Step 1: Mapping of regular entity types.
- Step 2: Mapping of weak entity types
- Step 3: Mapping of binary 1:1 relationship type
- Step 4: Mapping of binary 1:N relationship type
- Step 5: Mapping of binary M:N relationship type
- Step 6: Mapping of multivalued attributes
- Step 7: Mapping of N-ary relationship types

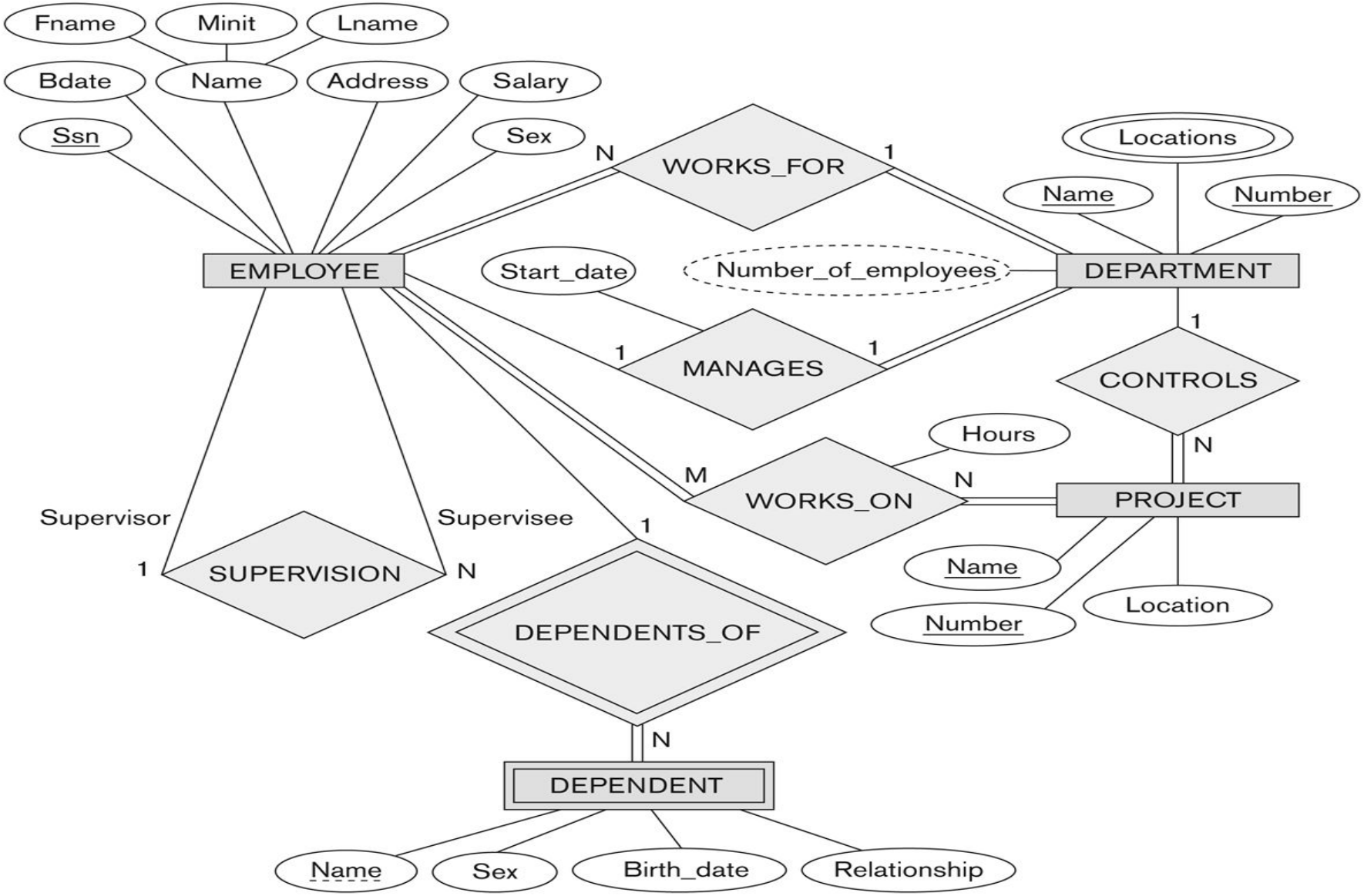


Figure 3.2
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter.

Step 1: Mapping of regular entity types

- For every regular entity type (E), create a relation (R) that includes all simple attributes of E.
- Include only the simple component attributes of a composite attribute.
- Choose one of the key attributes of E as the primary key for R.

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary

DEPARTMENT

Dname	Dnumber

PROJECT

Pname	Pnumber	Plocation

Step 2: Mapping of weak entity types

- For each weak entity type (W), with owner type (E), create a relation (R) and include all simple attributes of W.
- In addition, include the primary key of E as the foreign key of R.
- Primary key of R = Primary key of E + partial key of W.

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

Step 3: Mapping of binary 1:1 relationship type

- 3 approaches:

1) Foreign key approach
2) Merged relation approach
3) Cross-reference approach

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

- Foreign key approach
- Let the 2 relations be S & T.
- Choose one of the relation, say S, & include the primary key of T as foreign key in S. (Its better to choose an entity type with total participation as S.)
- Include all the simple attributes of 1:1 relationship type as attributes of S.

Step 4: Mapping of binary 1:N relationship type

- Let the 2 relations be S & T.
- Let S be the entity type at the 'N' side.
- Add the primary key of T as foreign key in S.

EMPLOYEE

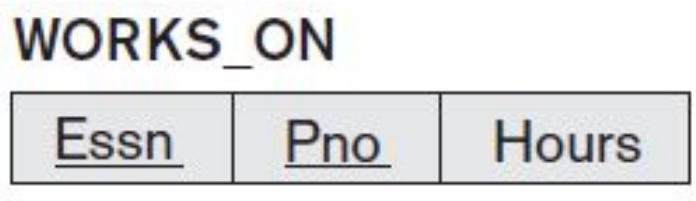
Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

PROJECT

Pname	Pnumber	Plocation	Dnum

Step 5: Mapping of binary M:N relationship type

- For every binary M:N relationship type, create a new relation S.
- Include the primary key of both the relations as foreign keys in S (Same will be the primary keys of S).
- Also include simple attributes of M:N relationship type as attributes of S.



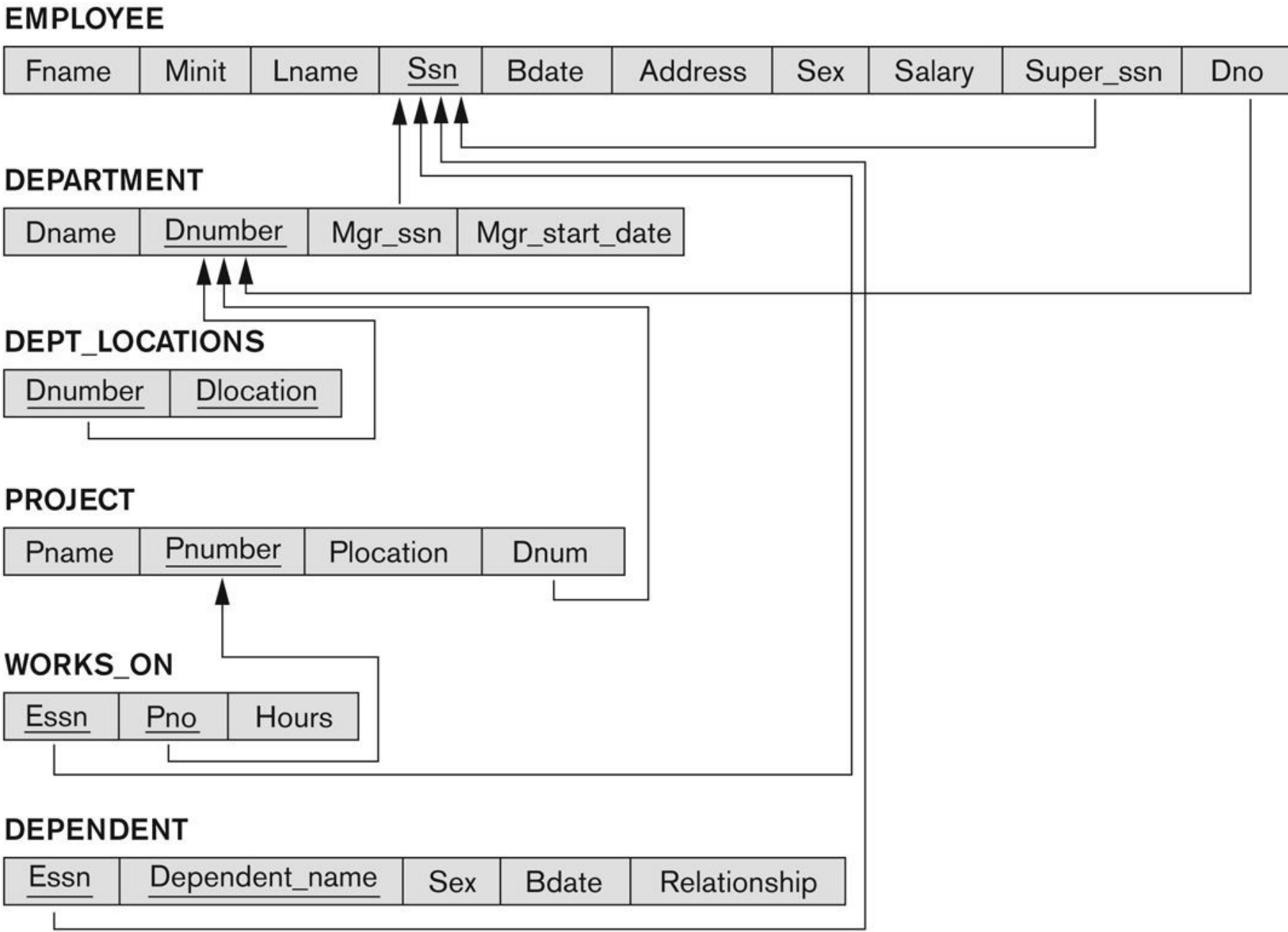
Step 6: Mapping of multivalued attributes

- For each multivalued attribute (A) in S, create a new relation R with that attribute.
- Also include the primary key (K) of S.
- Primary key of R will be the combination of A & K.

DEPT_LOCATIONS

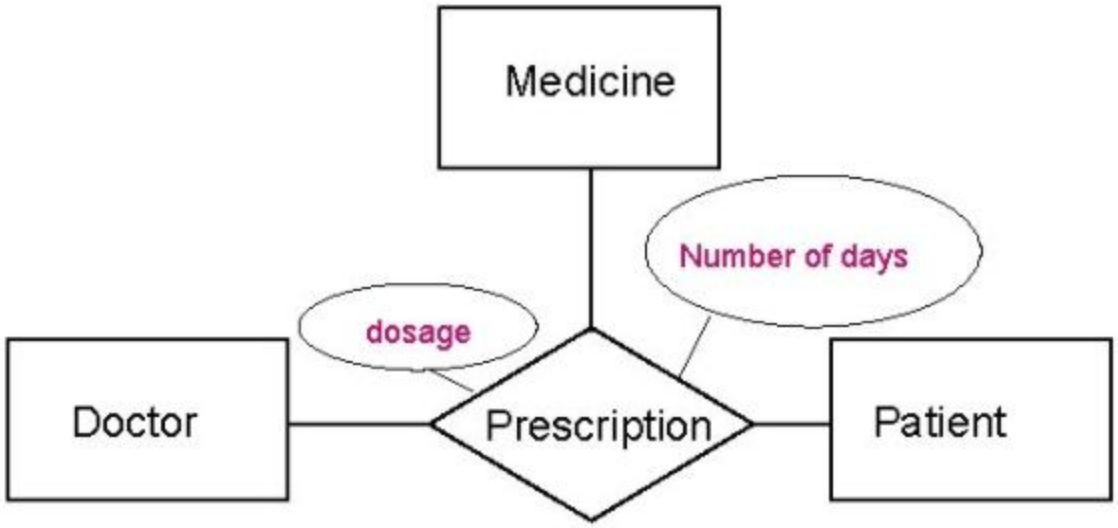
Dnumber	Dlocation

Figure 5.7
Referential integrity constraints displayed on the COMPANY relational database schema.



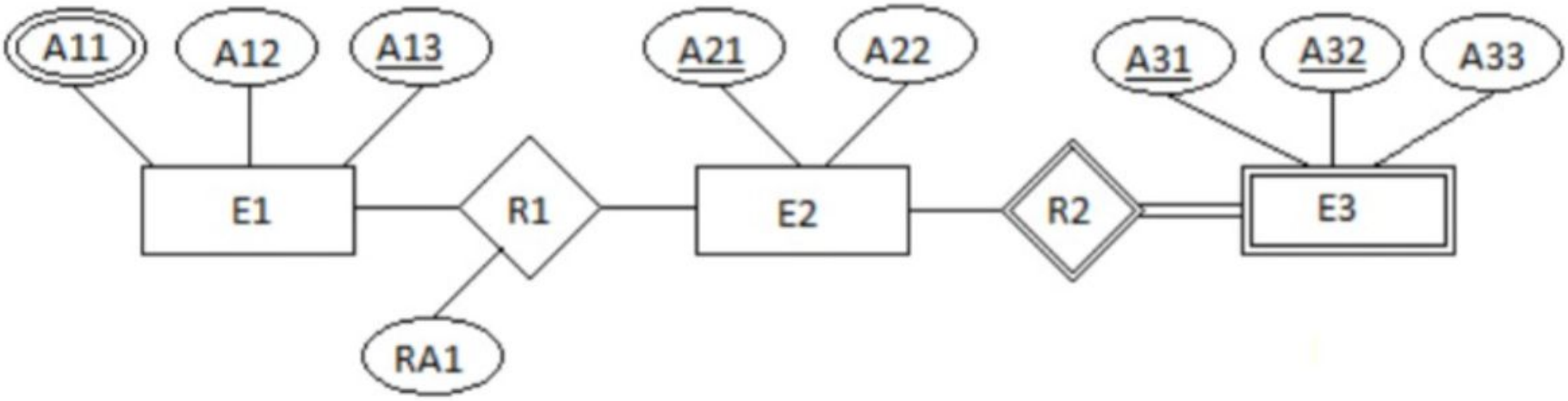
Step 7: Mapping of N-ary relationship type

- For each N-ary relationship type, create a new relation S.
- Include all the primary keys as foreign key in S.
- Also include simple attributes of the relationship type as attributes of S.



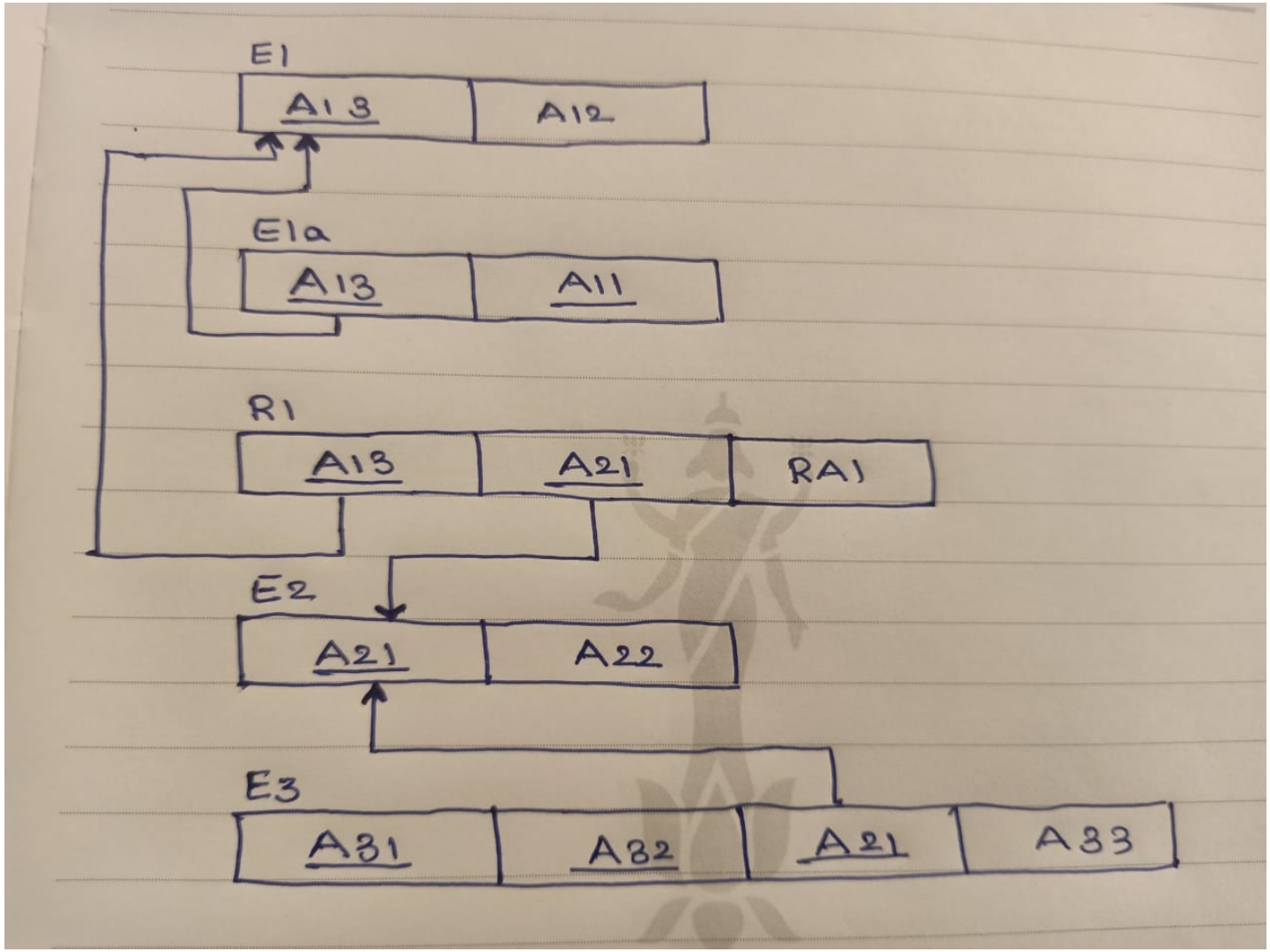
Tutorial

Using the following ER diagram, create a relation database. Give your assumptions.



NB: If cardinality not given, assume it as M:N

Tutorial Answer



PART 1 ENDS