2 : Relational Data Model

IT2306 – Database Systems I

Level I - Semester 2

Detailed Syllabus

2.1 Introduction to Relational Data Model:

- Review of database models

- Definition of
  - Relation
  - Attribute
  - Tuple
  - Domain
  - Instance
  - Cardinality
  - Degree
  - Schema
  - Constraints

Detailed Syllabus

2.2 Concepts of keys:

- Candidate key
- Primary key
- Alternate key
- Composite key
- Surrogate key
- Foreign key

2.3 Fundamental integrity rules:

- Entity integrity
- Referential integrity
- Domain constraints
- Key constraints.

2.1 Introduction to Relational Data Model

Review of Data Models

A data model is a collection of concepts that can be used to describe the structure of a database which includes describing data, data relationships, data semantics, and consistency constraints.

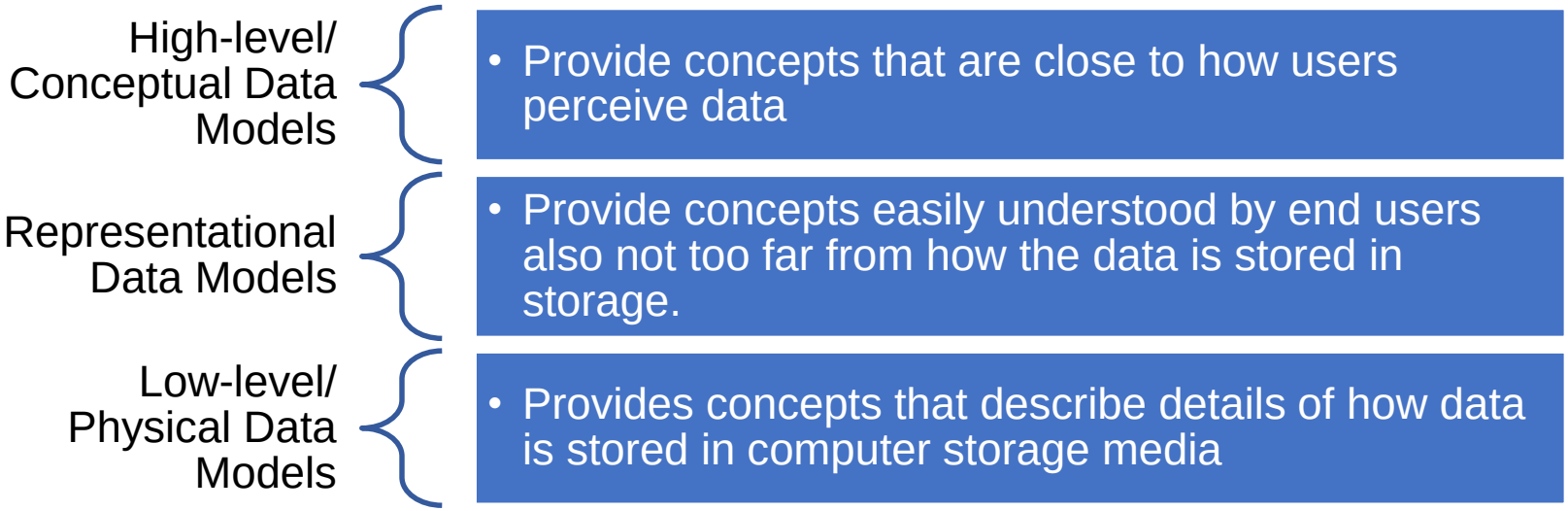
Most data models also include a set of basic operations for specifying retrievals and updates on the database.

In addition they may also allow database designers to specify a set of user-defined operations that are allowed on the database objects.

Review of Data Models

Categories of Data Models

- Data Models can be categorized based on the types of concepts they use to describe the database structure.



Review of Data Models

Conceptual data models:

- Use concepts like entities, attributes and relationships
  - An entity represent a real-world object or concept
    - Ex: employee, department
  - An attribute represent some property of an entity
    - Ex: employee's name
  - A relationship represent an association between two or more entities
    - Ex: employee 'works in' department
- Entity-relationship model is a popular high-level conceptual data model.

Review of Data Models

Representational/ Implementation data models

- Also called as record based data models.
- These models represent data by using record structures.
- The database contains a several types of fixed format records.
- Used frequently in traditional commercial DBMSs.
- Relational data model is an example of representational data models

Review of Data Models

Object Data Models

- An example for of a new family of high-level data models.
- Development of object-oriented programming software-development methodology has led to the development of an object-oriented data model.
- It can be seen as extending the E-R model with notions of encapsulation, methods (functions), and object identity.

Review of Data Models

Physical data models

- Describe how data is stored as files in computer by representing information.
  - Ex: record formats, ordering and access paths (indexes, hashes)

Review of Data Models

Self-describing data models.

- The data storage in systems based on these models combines the description of the data with the data values themselves.
- In traditional DBMSs, the description (schema) is separated from the data.
- These models include XML as well as many of the key-value stores and NOSQL systems that were recently created for managing big data.

The Relational Model

- The relational model uses a collection of relations (or tables) to represent both data and the relationships among those data.
- Each relation (or table) in a database has a unique name.
- Each relation has multiple attributes (or columns) with a unique name for each attribute.
- Each row is unique: no two rows in a relation are identical.
- An entry at the intersection of each row and column is atomic (or single valued)
- There can be no multi-valued attributes in a relation.

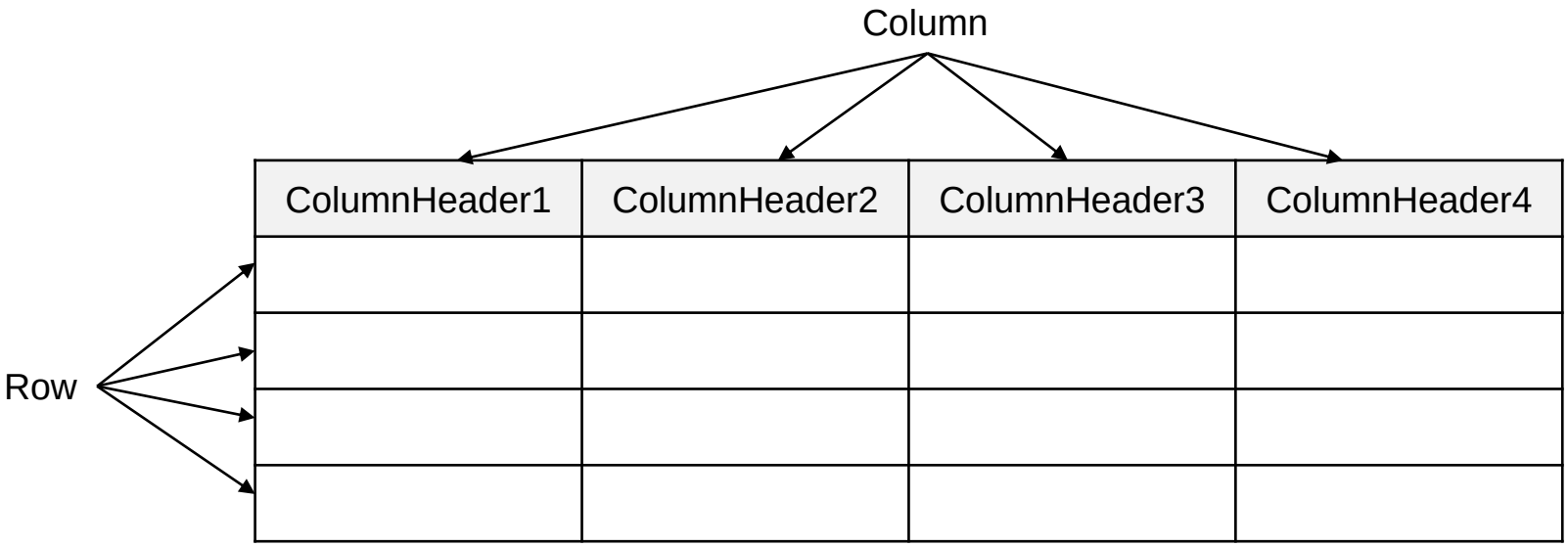
The Relational Model

The relational model of data has three major components:

- Relational database objects
  - allows to define data structures
- Relational operators
  - allows manipulation of stored data
- Relational integrity constraints
  - allows to defines business rules and ensure data integrity

The Relational Model

- Data is represented to the user as tables:
  - Tables are comprised of rows and a fixed number of named columns



The Relational Model

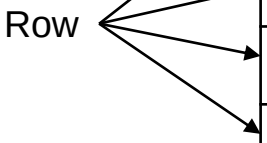
- Data is represented to the user as tables:
  - Columns are attributes describing an entity.
  - Each column must have an unique name and a data type.

EMPLOYEE

Name	Designation	Department	Salary

The Relational Model

- Data is represented to the user as tables:
  - Rows are records that present information about a particular entity occurrence



Name	Designation	Department	Salary
De Silva	Manager	Sales	50000
Perera	Secretary	Administration	35000
Dias	Clerk	Finance	40000
Alwis	Clerk	Administration	40000

The Relational Model

The formal relational model terminology:

- A row is called a **tuple.**
- A column header is called an **attribute.**
- A table is called a **relation**.
- The data type describing the types of values that can appear in each column is represented by a **domain** of possible values.

The Relational Model

- Domain
  - A domain D is a set of atomic values.
  - Atomic values in a domain is indivisible as far as the formal relation model is concerned.
  - Domain can be specified by
    - Data types for the data values in the domain
    - A name for the domain to help in interpreting its values.
  - Ex:
    - Grade_point_averages: Possible values of computed grade point averages; each must be a real (floating-point) number between 0 and 4.
    - Local_phone_numbers. The set of 10-digit phone numbers valid in a particular country.

The Relational Model

- Cardinality of the domain
  - Cardinality refers to the total number of values in a domain.
  - For domain D cardinality is denoted by |D|.
  - Example:
    - Domain: Days_of_week
    - |Days_of_week| = 7

The Relational Model

- Relation schema and Attributes
  - A relation schema is used to describe a relation
  - A relation schema is composed of a relation name R and a list of attributes A_1, A_2, … A_n,
  - Each attribute A_i is the name of a role played by some domain D in the relation schema R.
  - Domain of a particular attribute A_i is denoted by dom(A_i).

The Relational Model

- Example: A relation storing information about university students

STUDENT (Name, Ssn, Home_phone, Address, Office_phone, Age, Gpa)

Name of relation

Attributes

- Degree of STUDENT relation is 7 since it has 7 attributes.
- Along with the data types of each attribute we can re-write the above relation as:

STUDENT (Name: *string*, Ssn: *string*, Home_phone: *string*, Address: *string*, Office_phone: *string*, Age: *integer*, Gpa: *real*)

The Relational Model

- Attribute
  - It is possible for several attributes to have the same domain.
  - The attribute names indicate different roles, or interpretations, for the domain.
  - In STUDENT relation in the above example both Home_phone and Office_phone attributes have the same domain: Local_phone_numbers
    - dom(Home_phone) = dom (Office_phone) = Local_phone_numbers

The Relational Model

- Degree of a relation
  - The number of attributes n of its relation schema.
  - Example:

STUDENT (Name, Ssn, Home_phone, Address, Office_phone, Age, Gpa)

7 attributes

  - Degree of STUDENT relation is 7.

The Relational Model

- Relation and Tuple
  - A relation is a set of tuples.
  - Each tuple in the relation represent a particular entity (or an object)
  - A single tuple or a row is called as an instance.

The Relational Model

- Relation and Tuple
  - A relation of the relation schema $R(A_1, A_2, \ldots, A_n)$ is denoted by r(R) is a set of n-tuples

$$r = \{t_1, t_2, \ldots, t_m\}$$

  - Each n-tuple is an ordered list of n values

$$t = \langle v_1, v_2, \ldots, v_n \rangle$$

The Relational Model

- Relation and Tuple
  - Each value v_i is an element from the domain of its corresponding attribute dom(A_i) or a special NULL value.
    - A NULL value represent an attribute whose value is unknown or does not exist for a particular tuple.

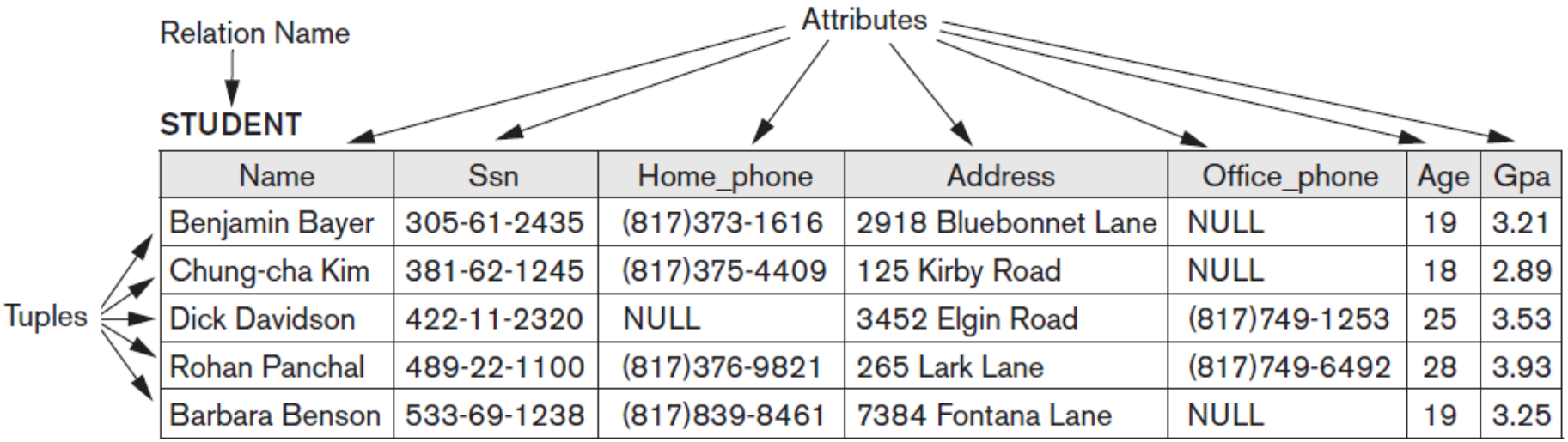
The Relational Model

Cardinality

- The cardinality of a table refers to the number of rows in the table.

The Relational Model

- Example: STUDENT relation



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

Cardinality of the relation is 5, since there are only 5 tuples in the relation.

The Relational Model

Constraints:

- In a typical relational database, there are many relations, and the tuples in those relations are usually related in various ways.
- There are generally many restrictions or constraints on the actual values in a database state.
- These restrictions are derived from the rules in the miniworld that the database represents.
- The various restrictions on data that can be specified on a relational database in the form of constraints

Characteristics of Relations

Ordering of tuples

- Tuples in a relation don't have any particular order.
- How ever records in a file may be physically ordered based on a criteria, this is not there in relational model

Ordering of values within tuple

- If the correspondence between attributes and values is maintained, order of attributes and their values is not that important.
  - In this case we can consider a tuple can be considered as a set of attribute and value pairs.

Values in a tuple are atomic.

2.2 Concepts of keys

Concepts of Keys

Superkey

- A subset of attributes of a relation schema, with the property that no two tuples in a relation have same combination of values in the relation; that subset of attributes is called as ‘superkey’.
- A superkey specifies a uniqueness constraint on tuples.

Concepts of Keys

Key

- A key of a relation is a minimal superkey.
- Minimal superkey means, if any attribute in the superkey k of a relation schema R is removed, the resulting subset of attributes k' is not a superkey of R anymore.
- A key has two properties:
  - Uniqueness
  - Minimality
- A key is determined from the meaning of the attributes.
  - When inserting new tuples to the relation above properties of the key must continue to hold.
- .

Concepts of Keys

Example

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

- {Ssn, Name, Age} is a superkey
  - Since for any tuple combination of values for any of these attributes are unique.
- {Ssn} is a key.
  - Since after dropping other 2 attributes value for Ssn is still unique for each tuple.

Concepts of Keys

Candidate key

- If a relation has more than one key, they are called candidate keys.

Primary key

- The primary key is used to uniquely identify tuples in the relation.
- One of the candidate keys is chosen as the primary key.

Concepts of Keys

Example

CAR

License_number	Engine_serial_number	Make	Model	Year
Texas ABC-739	A69352	Ford	Mustang	02
Florida TVP-347	B43696	Oldsmobile	Cutlass	05
New York MPO-22	X83554	Oldsmobile	Delta	01
California 432-TFY	C43742	Mercedes	190-D	99
California RSK-629	Y82935	Toyota	Camry	04
Texas RSK-629	U028365	Jaguar	XJS	04

- For CAR relation both attributes License_number and Engine_serial_number are candidate keys.
  - Both of them on their own has distinct values in each tuple
- License_number has been choosen as the primary key. (Only primary key is underlined)

Concepts of Keys

Composite key

- When a key has more than one attributes it is called as a composite key.
- Example:

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

  - For DEPENDENT relation {Essn, Dependent_name} is a key.
  - Since it has 2 attrubutes, it is a composite key.

Concepts of Keys

Foreign keys

- A set of attributes FK in relation schema R_1 is a foreign key of R_1 that references relation R_2 if it satisfies the following rules:
  - The attributes of FK reference (or refers to) the relation R_2:
    - The attributes in FK have the same domain(s) as the primary key attributes PK of R_2
  - The tuple t_1 in relation $r_1(R_1)$ references (or refers to) the tuple t_2 of relation $r_2(R_2)$.
    - A value of FK in a tuple t1 either occurs as a value of PK for some tuple t2 or is NULL.
- R_1 is called the referencing relation
- R_2 is called referenced relation

Concepts of Keys

Example

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

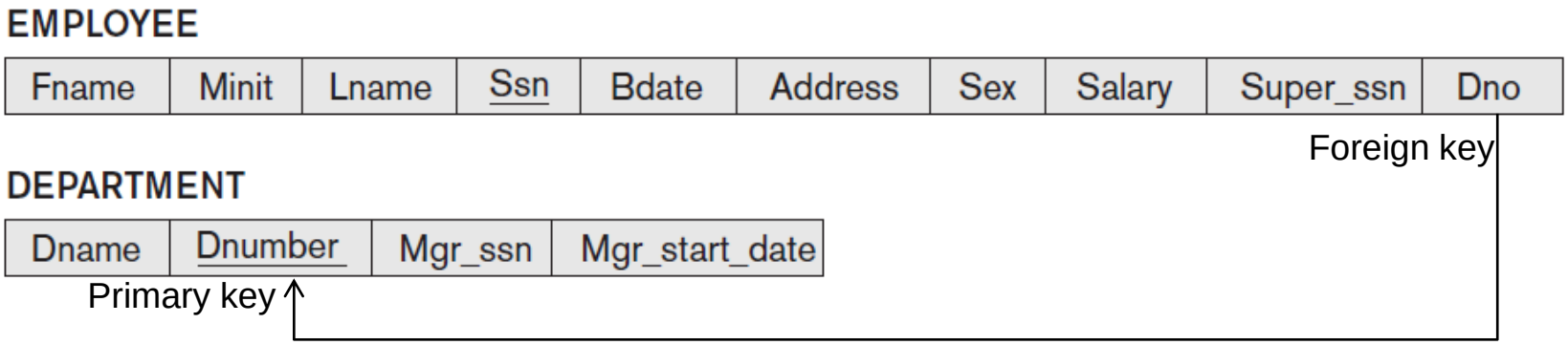
DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

- In the EMPLOYEE relation, the attribute Dno refers to the department for which an employee works for.
- Therefore Dno can be designated as a foreign key of EMPLOYEE referencing the DEPARTMENT relation.

Concepts of Keys

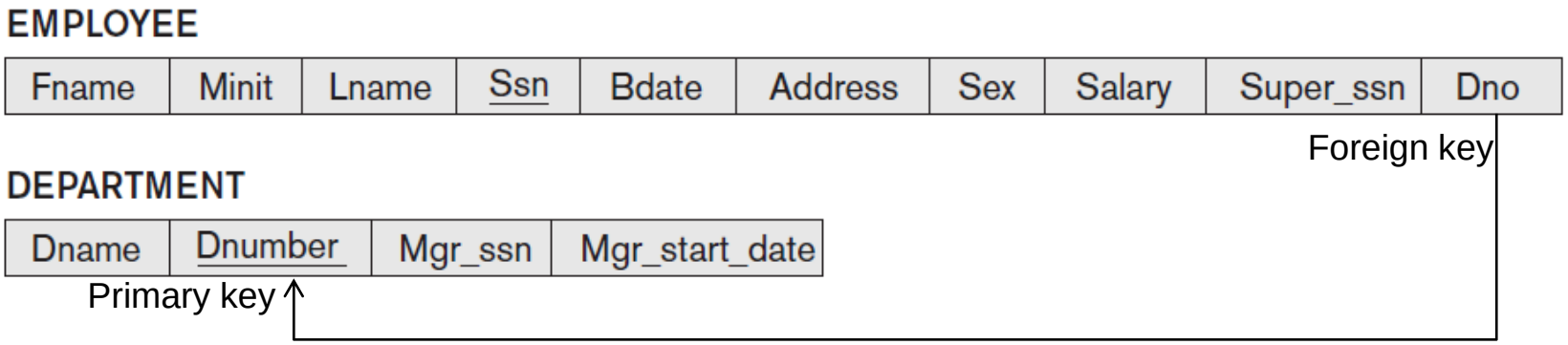
Example



- Referencing relation: Employee
- Referenced relation: Department

Concepts of Keys

Example

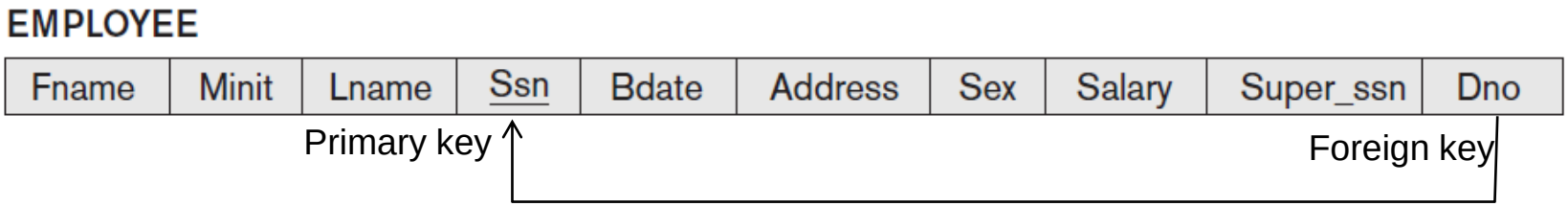


- For any tuple t1 of the EMPLOYEE relation, value of Dno must match a value of the primary key of DEPARTMENT (i.e. the Dnumber attribute) in some tuple t2 of the DEPARTMENT relation
- The value of Dno can be NULL if the employee does not belong to or not yet assigned to a department.

Concepts of Keys

Recursive foreign key

- A foreign key that refer to its own relation
- Example:



- The attribute Super_ssn in EMPLOYEE refers to the supervisor of an employee; supervisor is another employee, represented by a tuple in the same relation
  - Referencing relation: Employee
  - Referenced relation: Employee

2.3 Fundamental Integrity Rules

Relational Model Constraints

Constraints that can be directly expressed in the data model are called as Schema-based constraints.

They can be expressed in a schema of a relational model using Data Definition Languages (DDL).

The schema-based constraints includes

- Domain constraints
- Constraints on NULLs
- Key constraints
- Entity integrity constraint
- Referential integrity constraints

Relational Model Constraints

Integrity constraints

- Integrity constraints are an important part of a relational database schema.
- A database state is valid only if it satisfies all the constraints in the defined set of integrity constraints.

Relational Model Constraints

Domain constraints

- Domain constraints specify that within each tuple, the value of each attribute A must be an atomic value from the domain dom(A).

Constraints on NULLs

- Specifies whether NULL values are or are not permitted.

Relational Model Constraints

Key constraints/ uniqueness constraints

- A relation usually has one or more attributes whose values are distinct for each individual tuple.
- Such a subset of attributes from a relation is called a key attribute.
- Key constraints or uniqueness constraints specify that key attributes' values can be used to identify each tuple uniquely.

Relational Model Constraints

Entity integrity constraint

- States that no primary key value can be NULL.
- If primary key is NULL
  - Tuples cannot be distinguished since the primary key is used to identify individual tuples uniquely.
  - It is also not possible to refer to them from other relations or attributes in the same relation.

Relational Model Constraints

Referential integrity constraints

- The referential integrity constraint between two relations maintains the consistency among tuples in the two relations.
- Referential integrity constraints typically arise from the relationships among the entities
- Referential integrity is defined using foreign keys.
  - These constraint states that a tuple in one relation that refers to another relation must refer to an existing tuple in that relation.