uc3m | Universidad Carlos III de Madrid

OpenCourseWare

Database

2.1. Design of a database. Relational model

Lourdes Moreno López
Paloma Martínez Fernández
José Luis Martínez Fernández
Rodrigo Alarcón García

HULAT
Human Language & Accessibility Technologies

Departamento de Informática

Content

1. Relational model
2. Relational Model constrains
3. Update operations and dealing with Constraint Violations

Relational model

Relational model

- It proposed by E.F.Codd (1970)
- The **relational model** is based on the mathematical concept of a **relation**
- The relational model represents the **database as a collection of relations**

GRADE_REPORT

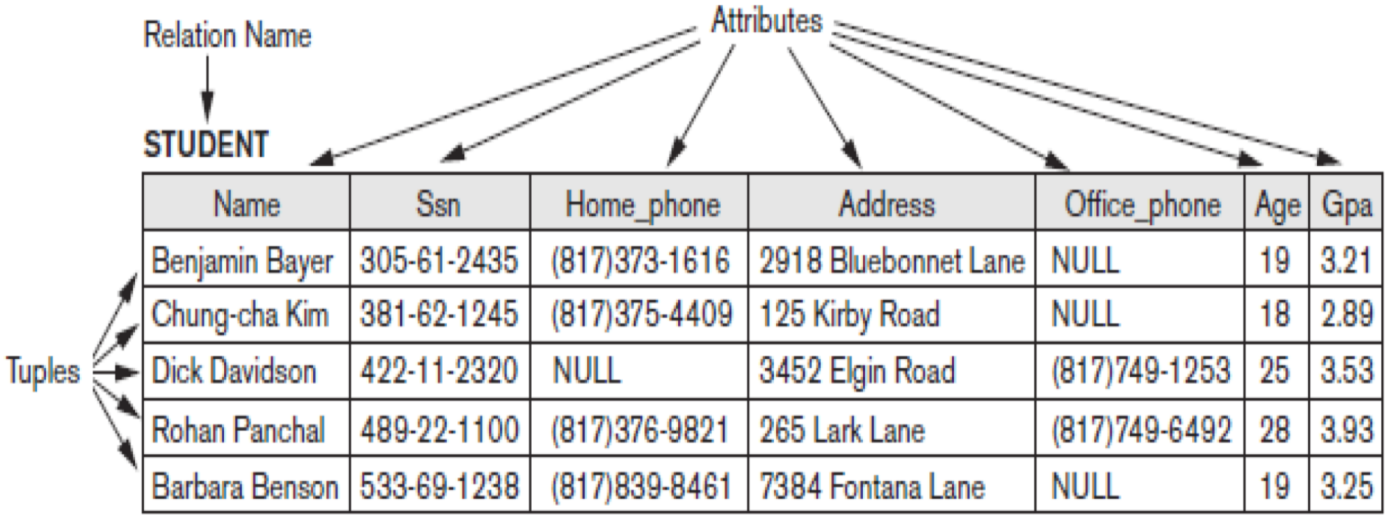
Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Formal Terminology

- A table is called a **relation**
- A row is called a **tuple**
- A column header is called an a**ttribute**

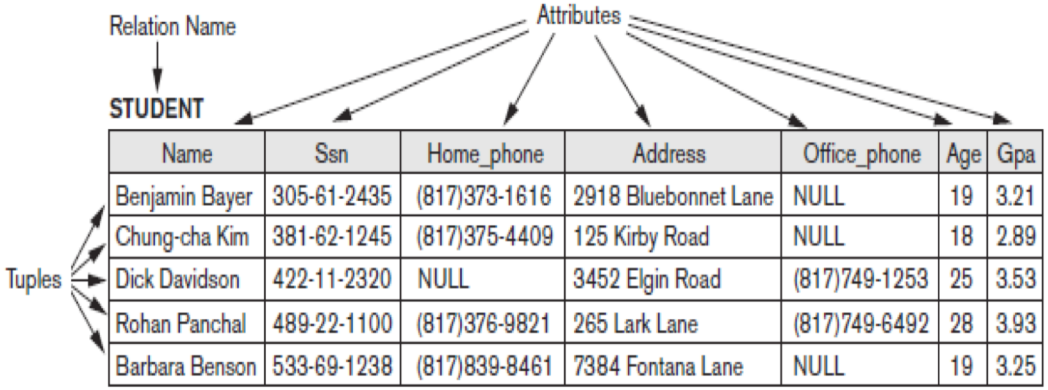


Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	(817)373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	(817)375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	(817)749-1253	25	3.53
Rohan Panchal	489-22-1100	(817)376-9821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

Fuente: Fundamentals of Database Systems, 7 ed., Ramez Elmasri Y Shamkant B. Navathe

Formal Terminology

- A **relation** is a set of **tuples**
- Each **tuple** in the table represents a collection of related data values
- **Relation** and **attributes** names are used to help to interpret the meaning of the values in each row



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

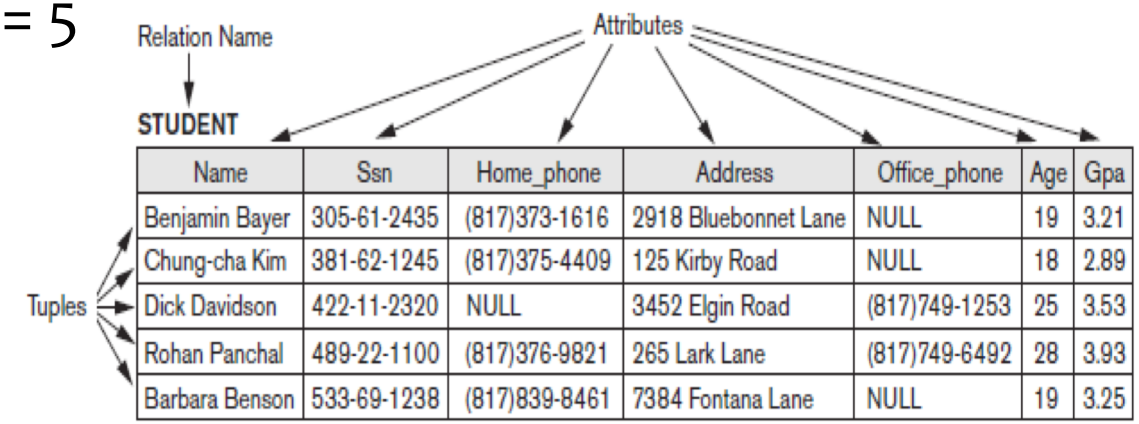
Fuente: Fundamentals of Database Systems, 7 ed., Ramez Elmasri Y Shamkant B. Navathe

Features

- The **degree** of a relation is the number of attributes it contains
- The **cardinality** of a relation is the number of tuples it contains

  Degree = 7

  Cardinality = 5



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

Fuente: Fundamentals of Database Systems, 7 ed., Ramez Elmasri Y Shamkant B. Navathe

Domain

- **Domain**
  - A finite set of homogeneous and atomic values, characterized by a name.
  - The set of allowed values for each attribute is called the domain of the attribute
  - It is useful to specify a name for the domain, to help in interpreting its values

Attribute	Domain definition	Domain name
City	Character: size 15	DomainCity
Sex	Character: size 1, value M or F	DomainSex
salary	Monetary: 7 digits, range: 8000.00-50000.00	DomainSalary

Schema

- A relational database usually contains many relations, with tuples in relations that are related in various ways
- A relational database schema is a set of relation schemas $S = \{R1, R2, \ldots, Rm\}$ and a set of integrity constraints
- It is a structural description of relations in database

Schema example: Company

EMPLOYEE (Fname, Minit, Lname, Ssn ,Bdate, Address, Sex, Salary. Super_ssn, Dno)

DEPARTMENT (Dname, Dnumber, Mgr_ssn, Mgr_start_date)

DEPT_LOCATIONS (Dnumber,
Dlocation)
PROJECT (Pname,Pnumber,Plocation, Dnum)

WORKS_ON (Essn, Pno, Hours)

Relational Model Constrains

Relational Model Constrains Types

- In the relational database, there are many relations
- **Relations are related among themselves in different ways**
- There are **constrains** on the actual values in a database state
  - **Inherent model-based** constrains
  - **Schema-based** constrains
    - **Domain** constraints
    - **Key** constraints
    - Constraints **on NULLs**, **entity integrity** constraints, **Referential integrity** constraints**, Foreign keys**
  - **Semantic integrity** constraints

Relational Model Constrains
Inherent model-based constrains

- A relation **cannot have duplicate** tuples (mandatory primary key)
- The **order of the tuples** and the order of the attributes is **not relevant**

Relational Model Constrains
Schema-based constrains

- **Domain** constraints
- **Key** constraints
- Constraints **on NULLs**, **entity integrity** constraints,
- **Referential integrity** constraints**, Foreign keys**

Schema-based constrains
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).
- **Each attribute can only take a single value from the domain over which it is defined**
- Data types associated with domains:
  - Numeric data types for integers (integer) and real numbers (float)
  - Characters, Booleans, fixed-length strings, and variable-length strings
  - Date, time, timestamp
  - Subrange of values from a data type or as an enumerated data type in which all possible values are explicitly listed.

Schema-based constrains
Key constraints

- A relation is defined as a set of tuples
- All tuples in a relation must also be distinct
  - **Two tuples can not have the same combination of values for all their attributes.**

Schema-based constrains
Key constraints

- **Superkey:** An attribute, or set of attributes, that uniquely identifies a tuple within a relation
- **Candidate key:** a superkey such that no proper subset is a superkey within a relation (minimal)

STUDENT

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

Superkey = {Name, Office_phone, Home_phone}, {Name, Home_phone, Address}
Candidate Key ={Name, Home_phone}, {Ssn}

Fuente: Fundamentals of Database Systems, 7 ed., Ramez Elmasri Y Shamkant B. Navathe

Schema-based constrains
Key constraints

- **Primary key:** a candidate key chosen as the principal means of identifying tuples within a relation
  - **Choose an attribute that almost never changes**
  - E.g. email address is unique, but may change
- **Alternate Key:**
  - A relation always has a primary key.
  - In the worst case, the entire set of attributes could serve as the primary key, but usually some smaller subset is sufficient to distinguish the tuples. The candidate keys that are not selected to be the primary key are called alternate keys.

Schema-based constrains
Key constraints

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

Candidate Key ={License_number}, {Engine_serial_number}

=> Primary Key = {License_number}

Fuente: Fundamentals of Database Systems, 7 ed., Ramez Elmasri Y Shamkant B. Navathe

Schema-based constrains
Key constraints

STUDENT

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

Candidate Key ={Name, Home_phone}, {Ssn}

=> Primary Key = {Ssn}

Fuente: Fundamentals of Database Systems, 7 ed., Ramez Elmasri Y Shamkant B. Navathe

Schema-based constrains
Key constraints

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Candidate Key ={FName, Lname, Bdate, Address}, {Ssn}

=> Primary Key = {Ssn}

Fuente: Fundamentals of Database Systems, 7 ed., Ramez Elmasri Y Shamkant B. Navathe

Schema-based constrains Constraints on NULLs, Entity Integrity constraints

uc3m

- No attribute that is part of the primary key of a relation can take a null value (entity integrity constraint)

Schema-based constrains Referential Integrity constraints, and Foreign Keys

uc3m

- The referential integrity constraint is specified between two relations
- It is used to maintain the consistency among tuples in the two relations.
- The referential integrity constraint states that a tuple in one relation that refers to another relation must refer to an existing tuple in that relation.

Schema-based constrains Referential Integrity constraints, and Foreign Keys

uc3m

- To define referential integrity more formally, first we define the concept of a foreign key
- A set of attributes FK in relation schema R1 is a foreign key of R1 that references relation R2 if it satisfies the following rules:
  - 1. The attributes in FK have the same domain(s) as the primary key attributes PK of R2; the attributes FK are said to reference or refer to the relation R2.
  - 2. A value of FK in a tuple t1 of the current state r1(R1) either occurs as a value of PK for some tuple t2 in the current state r2(R2) or is NULL. In the former case, we have t1[FK] = t2[PK], and we say that the tuple t1 references or refers to the tuple t2.

Shema example: Movie Database

- We are going to design a database of movies. The data to be collected in the database are given by the following assumptions.
  - A director is identified by a code, he/she has a first name and a last name.

    DIRECTOR (<u>code</u>, Fname, Lname)

  - A movie is identified by a code, in addition, it must be collected: the title, year, duration, language, and country

    MOVIE (<u>code,</u> Title, Year, Time, Lang, country)

Shema example: Movie Database

- More assumptions (2 different possibilities)
  - Option 1) A movie is directed by one director
  - Option 2) A movie may be directed by more than one director

MOVIE (Id, Title, Year, Time, Lang, Country, Id_Director)

DIRECTOR (Id, Fname, Lname)

Shema example: Movie Database

- Option 1) A movie is directed by one director

MOVIE (code, Title, Year, Time, Lang, Country, ~~code_Director~~)

DIRECTOR (code, Fname, Lname)

MOVIE

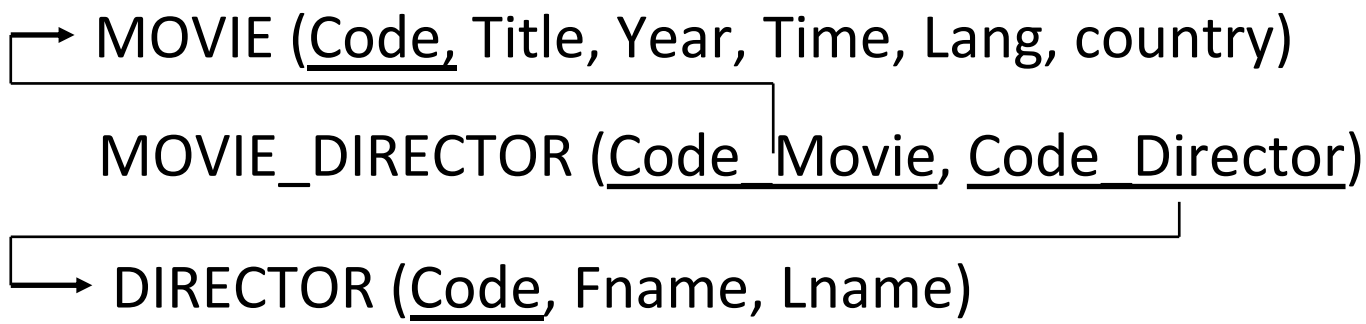
Code	Title	Year	Time	Lang	Country	Code_Director
901	Vertigo	1958	128	English	UK	201
914	American Beauty	1999	122	English	UK	214
915	Titanic	1997	194	English	UK	215

DIRECTOR

Code	Fname	Lname
201	Alfred	Hitchcock
206	Ridley	Scott
207	Stanley	Kubrick

Shema example: Movie Database

- Option 2) a movie may be directed by more than one director



Shema example: Movie Database

- Option 2) a director can direct several movies

MOVIE (Code, Title, Year, Time, Lang, country)

MOVIE_DIRECTOR (Code_Movie, Code_Director)

DIRECTOR (Code, Fname, Lname)

MOVIE_DIRECTOR

Code_Movie	Code_Director
901	201
900	220
900	221

MOVIE

Code	Title	Year	Time	Lang	Country
901	Vértigo	1958	128	English	UK
900	West Side Story	1961	194	English	UK

DIRECTOR

Code	Fname	Lname
201	Alfred	Hitchcock
220	Robert	Wise
221	Jerome	Robbins

Shema example: COMPANY Database

Design a database that manages the data of employees, departments and projects in a company.

The employees' data should be stored such as the family name, initial of the name, last name, social security number, date of birth, address, sex, and salary.

Each employee is identified by the social security number.

EMPLOYEE (Fname, Minit, Lname, Ssn ,Bdate, Address, Sex)

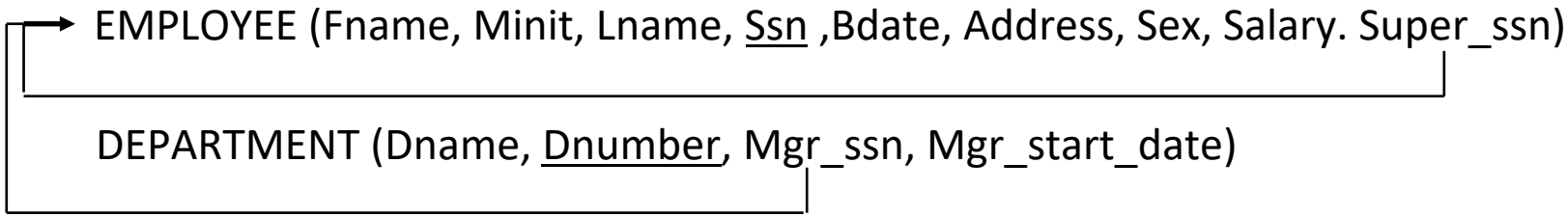
Shema example: COMPANY Database

- Each employee has a supervisor, and this information needs to be collected.

EMPLOYEE (Fname, Minit, Lname, Ssn ,Bdate, Address, Sex, Salary. Super_ssn,)

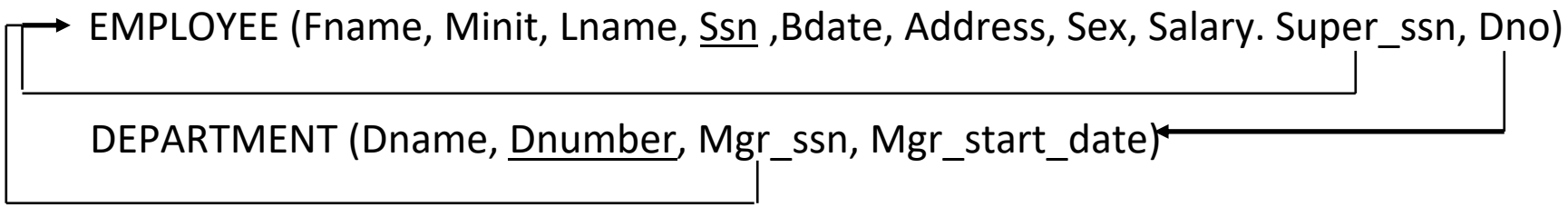
Shema example: COMPANY Database

- On the other hand, it will be necessary to store information about the departments. Thus, the departments are identified by a number, they have a name and it must be collected which employee is the manager of that department and the date since he is the manager. Each department has a unique manager.



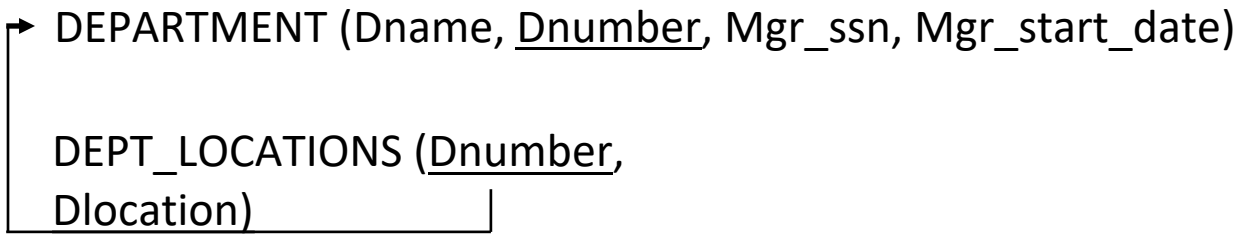
Shema example: COMPANY Database

- Each employee belongs to a alone department.



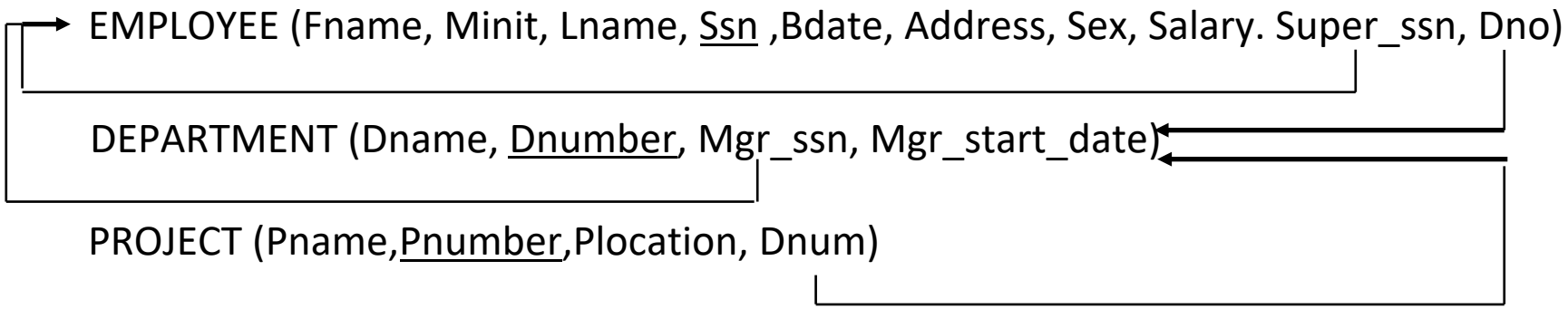
Shema example: COMPANY Database

- The locations of the departments are collected. A department can have different locations.



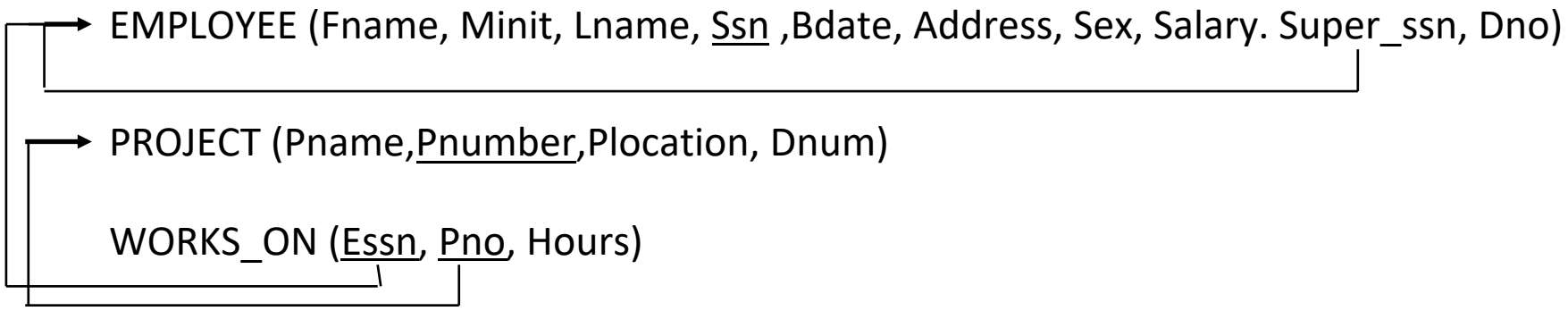
Shema example: COMPANY Database

- In addition, the projects in which the employees work, are stored.
- Each project is identified by number and has a name, they also have a unique location and are associated with a unique department.

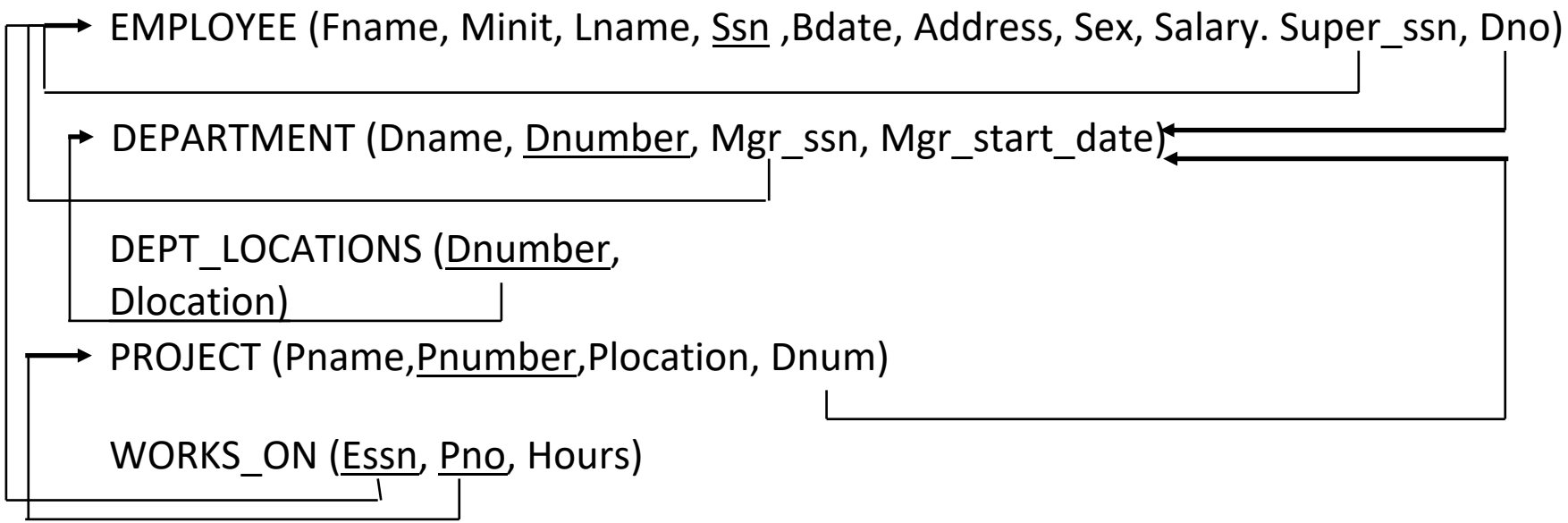


Shema example: COMPANY Database

- Finally, the management of hours worked by employees per project is collected.
- Thus, an employee can work on different projects and different employees can work on a project.



Shema example: COMPANY Database



Semantic integrity constraints Check

- In some cases, it may be necessary to specify a condition that must be met by the values of certain attributes of a database relation apart from the restrictions already seen in primary key, uniqueness and foreign key.

Example:

  - for the EMPLOYEE relation, a restriction could be defined on the SALARY attribute that establishes that "the salary range of an employee can range between 100,000 and 500,000 €".

Semantic integrity constraints Assertion

- Restrictions in which the condition is established on elements of different relations.

Example:

- " there is no employee that works in the accounting department that earns more than 200,000 euros”

Semantic integrity constraints Trigger

- Sometimes it may be interesting to specify an action other than rejection when a certain semantic restriction is not met. In that case, we use the TRIGGERS that allow us to indicate a condition, specify the action we want to take place if the condition becomes True.

Update operations and dealing with Constraint Violations

Database modification or update operations

- Operations:
  - **Insert:** insert one or more new tuples in a relation
  - **Delete:** delete tuples
  - **Update** (or Modify): change the values of some attributes in existing tuples.
- Whenever these operations are applied, the constraints specified on the relational database schema **should not be violated.**

The Insert Operation

- The Insert operation provides a list of attribute values for a new tuple t that is to be inserted into a relation R.
- Insert can **violate** any of the four types of constraints.
  - **Domain constraints** can be violated if an attribute value is given that does not appear in the corresponding domain or is not of the appropriate data type.
  - **Key constraints** can be violated if a key value in the new tuple t already exists in another tuple in the relation r(R).
  - **Entity integrity** can be violated if any part of the primary key of the new tuple t is NULL.
  - **Referential integrity** can be violated if the value of any foreign key in t refers to a tuple that does not exist in the referenced relation.

Example: COMPANY
The Insert Operation

- Operation: Insert <'Alicia', 'J', 'Zelaya', '999887777', '1960-04-05', '6357 Windy Lane, Katy, TX', F, 28000, '987654321', 4> into EMPLOYEE.

- Result: This insertion violates the **key constraint** because another tuple with the same Ssn value already exists in the EMPLOYEE relation, and so it is rejected.

Fuente: Fundamentals of Database Systems, 7 ed., Ramez Elmasri Y Shamkant B. Navathe

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Example: COMPANY
The Insert Operation

- Operation: Insert <'Cecilia', 'F', 'Kolonsky', NULL, '1960-04-05', '6357 Windy Lane, Katy,TX', F, 28000, NULL, 4> into EMPLOYEE.
- Result: This insertion violates the **entity integrity constraint** (NULL for the primary key Ssn), so it is rejected.

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Fuente: Fundamentals of Database Systems, 7 ed., Ramez Elmasri Y Shamkant B. Navathe

Example: COMPANY
The Insert Operation

- Operation: Insert <'Cecilia', 'F', 'Kolonsky', '677678989', '1960-04-05', '6357 Windswept, Katy, TX', F, 28000, '987654321', 7> into EMPLOYEE.

- Result: This insertion violates the **referential integrity constraint** specified on Dno in EMPLOYEE because no corresponding referenced tuple exists in DEPARTMENT with Dnumber = 7.

EMPLOYEE (Fname, Minit, Lname, Ssn ,Bdate, Address, Sex, Salary. Super_ssn, Dno)

DEPARTMENT (Dname, Dnumber, Mgr_ssn, Mgr_start_date)

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

Fuente: Fundamentals of Database Systems, 7 ed., Ramez Elmasri Y Shamkant B. Navathe

Example: COMPANY
The Insert Operation

- Operation: Insert <'Cecilia', 'F', 'Kolonsky', '677678989', '1960-04-05', '6357 Windy Lane, Katy, TX', F, 28000, NULL, 4> into EMPLOYEE.
- Result: This insertion satisfies all constraints, so it is acceptable.

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987						
James	E	Borg	888665555						

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

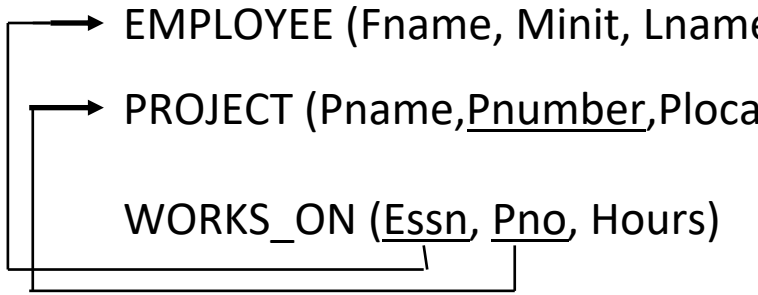
Fuente: Fundamentals of Database Systems, 7 ed., Ramez Elmasri Y Shamkant B. Navathe

The Delete Operation

- The Delete operation can violate only **referential integrity.** This occurs if the tuple being deleted is referenced by foreign keys from other tuples in the database.
- Several options are available if a deletion operation causes a violation.
  - **restrict,** is to reject the deletion.
  - **cascade,** is to attempt to cascade (or propagate) the deletion by deleting tuples that reference the tuple that is being deleted.
  - **set null or set default,** is to modify the referencing attribute values that cause the violation; each such value is either set to NULL or changed to reference another default valid tuple.

Example: COMPANY
The Delete Operation

- Operation: Delete the WORKS_ON tuple with Essn = '999887777' and Pno = 10.
- Result: This deletion is acceptable and deletes exactly one tuple.



WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

Fuente: Fundamentals of Database Systems, 7 ed., Ramez Elmasri Y Shamkant B. Navathe

Example: COMPANY
The Delete Operation

- Operation: Delete the EMPLOYEE tuple with Ssn = '666884444'.
- Result: This deletion is not acceptable, because there are tuples in WORKS_ON that refer to this tuple. Hence, if the tuple in EMPLOYEE is deleted, **referential integrity** violations will result. In this case, **options** must be defined.

EMPLOYEE (Fname, Minit, Lname, Ssn ,Bdate, Address, Sex, Salary. Super_ssn, Dno)

WORKS_ON (Essn, Pno, Hours)

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate				ber_ssn	Dno
John	B	Smith	123456789	1965-01-09				445555	5
Franklin	T	Wong	333445555	1955-12-08				665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5

Fuente: Fundamentals of Database Systems, 7 ed., Ramez Elmasri Y Shamkant B.

Example: COMPANY
The Delete Operation

- **OPTION restrict:**
  - Operation: Delete the EMPLOYEE tuple with Ssn = '666884444'.
  - Result: This deletion is rejected. The delete operations isn't done.

EMPLOYEE (Fname, Minit, Lname, Ssn ,Bdate, Address, Sex, Salary. Super_ssn, Dno)

WORKS_ON (Essn, Pno, Hours)

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate				...er_ssn	Dno
John	B	Smith	123456789	1965-01-09				...445555	5
Franklin	T	Wong	333445555	1955-12-08				...665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5

Fuente: Fundamentals of Database Systems, 7 ed., Ramez Elmasri Y Shamkant B.

Example: COMPANY
The Delete Operation

- **OPTION cascade:**
  - Operation: Delete the EMPLOYEE tuple with Ssn = '666884444'.
  - Result: The deletion in EMPLOYEE is done, also the related tuples in relation WORKS_ON will be deleted.

EMPLOYEE (Fname, Minit, Lname, Ssn ,Bdate, Address, Sex, Salary. Super_ssn, Dno)

WORKS_ON (Essn, Pno, Hours)

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
~~666884444~~	~~3~~	~~40.0~~

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate				...er_ssn	Dno
John	B	Smith	123456789	1965-01-09				...445555	5
Franklin	T	Wong	333445555	1955-12-08				...665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
~~Ramesh~~	~~K~~	~~Narayan~~	~~666884444~~	~~1962-09-15~~	~~975 Fire Oak, Humble, TX~~	~~M~~	~~38000~~	~~333445555~~	~~5~~

Fuente: Fundamentals of Database Systems, 7 ed., Ramez Elmasri Y Shamkant B.

The Update Operation

- The Update (or Modify) can violate only **referential integrity**. This occurs if the tuple being updated is referenced by foreign keys from other tuples in the database.
- It is necessary to specify a condition on the attributes of the relation to select the tuple (or tuples) to be modified

The Update Operation

- Options exist to deal with referential integrity violations caused by update
  - **restrict**, is to reject the update.
  - **cascade**, is to attempt to cascade (or propagate) the update by updating tuples that reference the tuple that is being updated.
  - **set null or set default**, is to modify the referencing attribute values that cause the violation; each such value is either set to NULL or changed to reference another default valid tuple.

Example: COMPANY
The Update Operation

- Operation: Update the salary of the EMPLOYEE tuple with Ssn = '999887777' to 28000.
- Result: Acceptable

EMPLOYEE (Fname, Minit, Lname, Ssn ,Bdate, Address, Sex, Salary. Super_ssn, Dno)

Example: COMPANY
The Update Operation

- Operation: Update the Dno of the EMPLOYEE tuple with Ssn = '999887777' to 1.
- Result: Acceptable.

EMPLOYEE (Fname, Minit, Lname, Ssn ,Bdate, Address, Sex, Salary. Super_ssn, Dno)

DEPARTMENT (Dname, Dnumber, Mgr_ssn, Mgr_start_date)

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

Fuente: Fundamentals of Database Systems, 7 ed., Ramez Elmasri Y Shamkant B.

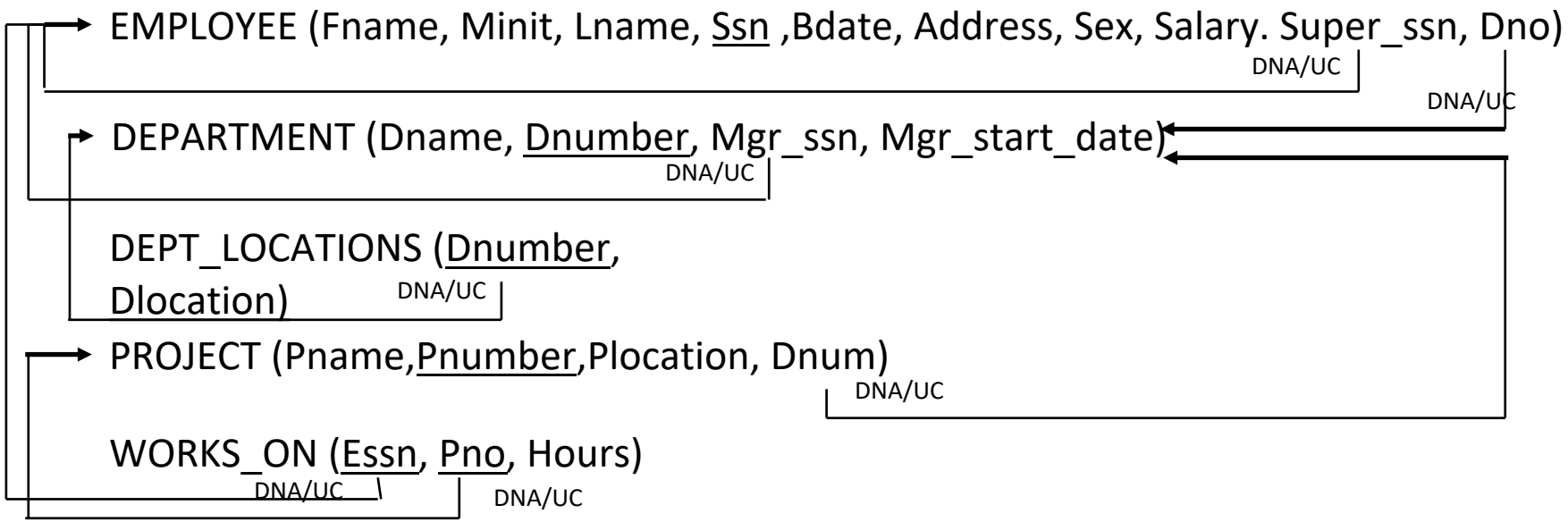
Example: COMPANY
The Update Operation

- Operation: Update the Dno of the EMPLOYEE tuple with Ssn = '999887777' to 7.
- Result: Unacceptable, because it violates **referential integrity**.

Example: COMPANY
The Update Operation

- Operation: Update the Ssn of the EMPLOYEE tuple with Ssn = '999887777' to '987654321'.
- Result: Unacceptable, because:
  - it violates **primary key constraint** by repeating a value that already exists as a primary key in another tuple;
  - it violates **referential integrity constraints** because there are other relations that refer to the existing value of Ssn.

Example: COMPANY Final schema



- by default **DNA/UC**
- or **DC/UC** if there are semantic assumptions

DNA: delete restrict
UC: update cascade
DC: delete cascade

References, Bibliography

- Elmasri Database Fundamentals of Database Systems by Elmasri, Navathe 7th ed. 2017
- Connolly, Thomas M, Begg, Carolyn E. Database systems: a practical approach to design, implementation, and management. Addison Wesley. 2015