

## **Chapter 3**

# **The Relational Data Model and Relational Database Constraints**

# The Relational Data Model and Relational Database Constraints

- Relational model
  - First commercial implementations available in early 1980s
  - Has been implemented in a large number of commercial system
- Hierarchical and network models
  - Preceded the relational model

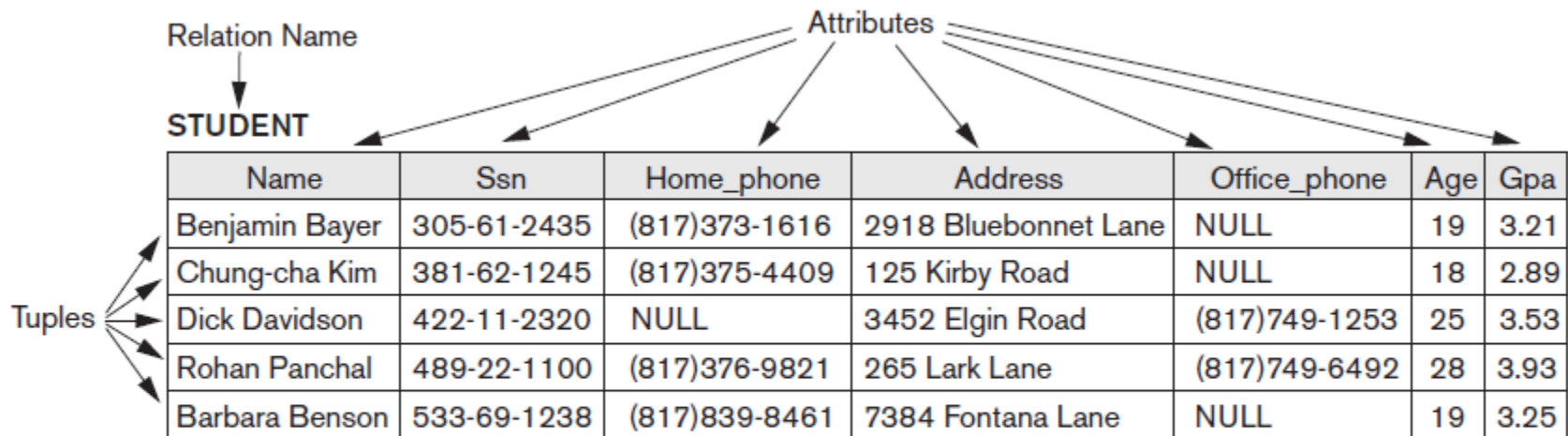
# Relational Model Concepts

- Represents data as a collection of relations
- *A relation* is a table with columns and rows.
  - Only applies to *logical structure of the database*, not the physical structure.
- *Attribute* is a named column of a relation.
- *Domain* is the set of allowable values for one or more attributes

# Relational Model Concepts (cont'd.)

- *Tuple* is a row of a relation.
- *Degree* is the number of attributes in a relation.
- *Cardinality* is the number of tuples in a relation.
- *Relational Database* is a collection of normalized relations with distinct relation names.

# Relational Model Concepts (cont'd.)



**Figure 3.1**

The attributes and tuples of a relation **STUDENT**.

Degree of relation: 7

Cardinality of relation: 5

# Relational Model Concepts (cont'd.)

Alternative terminology for relational model terms.

Formal terms	Alternative 1	Alternative 2
Relation	Table	File
Tuple	Row	Record
Attribute	Column	Field

---

# Domains, Attributes, Tuples, and Relations

- **Domain D**

- A domain is the set of allowable atomic values for one or more attributes

- **Atomic**

- Each value indivisible

- Specifying a domain

- **Data type** specified for each domain

# Relation Schema

- It is used to **describe a relation**
- Named relation defined by a set of attribute and domain name pairs.
- Denoted by  $R(A_1, A_2, \dots, A_n)$
- Made up of a relation name  $R$  and a list of attributes,  $A_1, A_2, \dots, A_n$

## Example

- STUDENT(Name, Ssn, Home\_phone, Address, Office\_phone, Age, Gpa)
- STUDENT(Name: string, Ssn: string, Home\_phone: string, Address: string, Office\_phone: string, Age: integer, Gpa: real)



# Relational database Schema and Relation State

## ■ Relational database schema

- Set of relation schemas, each with a distinct name.

## ■ Relation (or relation state)

- Set of ***n*-tuples**  $r = \{t_1, t_2, \dots, t_m\}$
- Each *n*-tuple *t*
  - Ordered list of *n* values  $t = \langle v_1, v_2, \dots, v_n \rangle$
  - Each value  $v_i$ ,  $1 \leq i \leq n$ , is an element of  $\text{dom}(A_i)$  or is a special NULL value
- Relation state at a given time
- Reflects only the valid tuples that represent a particular state of the real world

# Relation State

**Figure 3.2**

The relation STUDENT from Figure 3.1 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	(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

# Characteristics of Relations

- Ordering of tuples in a relation
  - Relation defined as a set of tuples
  - Elements have no order among them
- Ordering of values within a tuple and an alternative definition of a relation
  - Order of attributes and values is not that important
  - As long as correspondence between attributes and values maintained

# Characteristics of Relations (cont'd.)

- Each tuple is distinct; there are no duplicate tuples.
- Each attribute has a distinct name.
- Values of an attribute are all from the same domain.
  - Multivalued attributes (not allowed)
    - Must be represented by separate relations
  - Composite attributes (not allowed)
    - Represented only by simple component attributes in basic relational model

# Characteristics of Relations (cont'd.)

## ■ NULL values

- Represent the values of attributes that may be unknown or may not apply to a tuple
- Meanings for NULL values
  - *Value unknown*
  - *Value exists but is not available*
  - *Attribute does not apply to this tuple (also known as value undefined)*

# Relational Model Notation

- Name of a relation: STUDENT
  - Indicates the current set of tuples in that relation
- Notation: STUDENT(Name, Ssn, ...)
  - Refers only to relation schema
- Attribute  $A$  can be qualified with the relation name  $R$  to which it belongs
  - Using the dot notation  $R.A$

# Relational Model Constraints

- Constraints
  - Restrictions on the actual values in a database state
  - Derived from the rules in the miniworld that the database represents

# Categories of Constraints

- Inherent model-based constraints or implicit constraints
  - Ex: no two tuples can have the same values
- Schema-based constraints or explicit constraints
- Application-based or semantic constraints or business rules
  - Ex: salary of employees has a different range than the salary of managers



# Schema Based Constraints

- Domain constraints
- Key constraints
- Constraints on Null
- Entity integrity constraints
- Referential integrity constraints

# Domain Constraints

- Specify that within each tuple, the value of each attribute  $A$  must be an atomic value from the domain  $\text{dom}(A)$ .
- Can be enforced by:
  - Data types: numeric, boolean, characters, date and time etc....
  - Specifying sub-range of values
  - Enumerated data type in which all possible values are explicitly listed

# Key Constraints and Constraints on NULL Values

- No two tuples can have the same combination of values for all their attributes.
- **Superkey**
  - No two distinct tuples in any state  $r$  of  $R$  can have the same value for SK
  - SuperKey may contain redundant attributes
- **Key**
  - Superkey of  $R$
  - Removing any attribute  $A$  from  $K$  leaves a set of attributes  $K$  that is not a superkey of  $R$  any more

# Key Constraints and Constraints on NULL Values (cont'd.)

- **Key satisfies two properties:**
  - Two distinct tuples in any state of relation cannot have identical values for (all) attributes in key
  - Minimal superkey
    - Cannot remove any attributes and still have uniqueness constraint in above condition hold
- **A key is also a superkey but not vice versa**

# Key Constraints and Constraints on NULL Values (cont'd.)

- **Candidate key**
  - Relation schema may have more than one key
- **Primary key** of the relation
  - Designated among candidate keys
  - Underline attribute
- Other candidate keys are designated as **unique keys**
- **Another constraint on attributes specifies whether Null values are allowed or not**

# Key Constraints and Constraints on NULL Values (cont'd.)

CAR

<u>License_number</u>	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

**Figure 3.4**

The CAR relation, with two candidate keys: License\_number and Engine\_serial\_number.

# Integrity, Referential Integrity, and Foreign Keys

- **Entity integrity constraint**
  - No primary key value can be NULL
- **Referential integrity constraint**
  - Specified between two relations
  - Maintains consistency among tuples in two relations

# Integrity, Referential Integrity, and Foreign Keys (cont'd.)

- **Foreign key rules:**

- The attributes in FK have the same domain(s) as the primary key attributes PK
- Value of FK in a tuple  $t_1$  of the current state  $r_1(R_1)$  either occurs as a value of PK for some tuple  $t_2$  in the current state  $r_2(R_2)$  or is NULL



# Integrity, Referential Integrity, and Foreign Keys (cont'd.)

- Diagrammatically display referential integrity constraints
  - Directed arc from each foreign key to the relation it references
- All integrity constraints should be specified on relational database schema

# Valid/Invalid Database State

- **Invalid state for relational database**
  - Does not obey all the integrity constraints
- **Valid state for relational database**
  - Satisfies all the constraints in the defined set of integrity constraints IC

# Update Operations, Transactions, and Dealing with Constraint Violations

- Operations of the relational model can be categorized into retrievals and updates
- Basic operations that change the states of relations in the database:
  - Insert
  - Delete
  - Update (or Modify)

**STUDENT**

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

**COURSE**

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

**SECTION**

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

**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

# The Insert Operation

- Provides a list of attribute values for a new tuple  $t$  that is to be inserted into a relation  $R$
- Can violate any of the four types of constraints
- If an insertion violates one or more constraints
  - Default option is to reject the insertion

# The Delete Operation

- Can violate only referential integrity
  - If tuple being deleted is referenced by foreign keys from other tuples
  - **Restrict**
    - Reject the deletion
  - **Cascade**
    - Propagate the deletion by deleting tuples that reference the tuple that is being deleted
  - **Set null or set default**
    - Modify the referencing attribute values that cause the violation

# The Update Operation

- Necessary to specify a condition on attributes of relation
  - Select the tuple (or tuples) to be modified
- If attribute not part of a primary key nor of a foreign key
  - Usually causes no problems
- Updating a primary/foreign key
  - Similar issues as with Insert/Delete

## EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

## DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
-------	----------------	---------	----------------

## DEPT\_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

## PROJECT

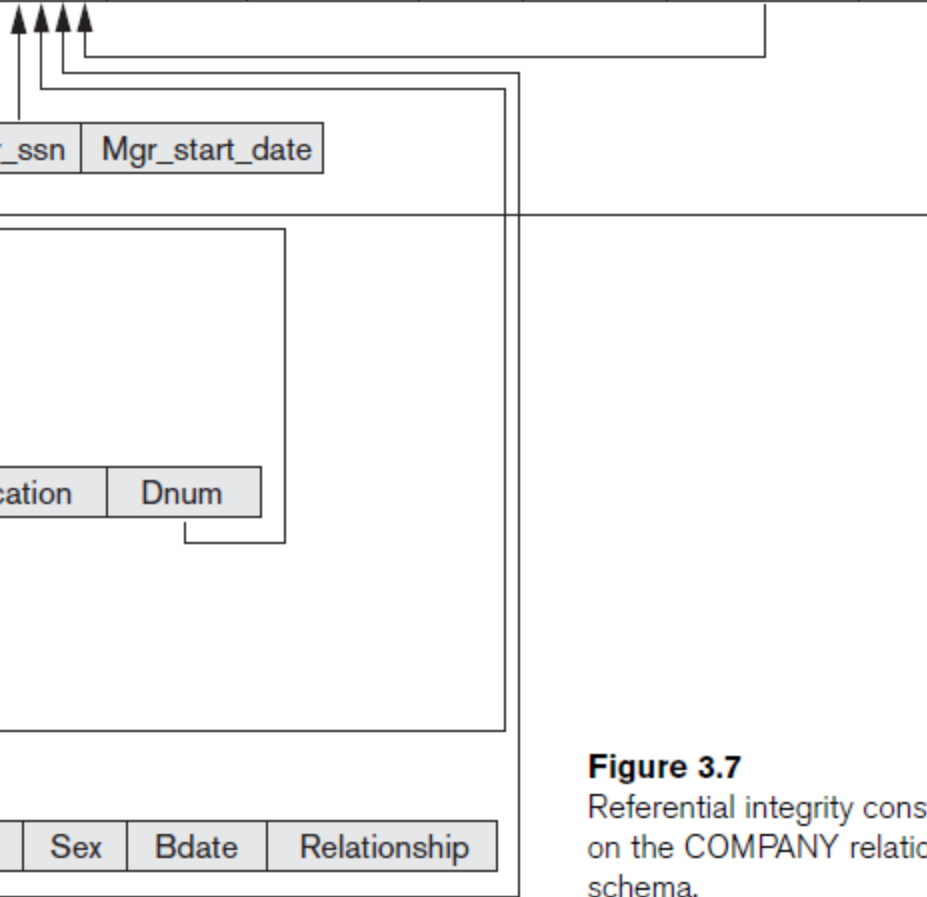
Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

## WORKS\_ON

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

## DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------



**Figure 3.7**

Referential integrity constraints displayed on the COMPANY relational database schema.



**Figure 3.6**

One possible database state for the COMPANY relational database schema.

**EMPLOYEE**

Fname	Minit	Lname	<u>Ssn</u>	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

**DEPARTMENT**

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

**DEPT\_LOCATIONS**

<u>Dnumber</u>	<u>Dlocation</u>
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

**Figure 3.6**

One possible database state for the COMPANY relational database schema.

**WORKS\_ON**

<u>Essn</u>	<u>Pno</u>	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

**PROJECT**

<u>Pname</u>	<u>Pnumber</u>	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

**DEPENDENT**

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

# The Insert Operation Example

*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.

*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.

# The Insert Operation Example

*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.

*Operation:*

Insert <' Cecilia', 'F', 'Kolonsky', '43435', '1960-04-05', '6357 Wind Katy', F, lot of money, '987654321', 1> into Employee

*Result:* This insertion violates domain integrity because salary attribute accepts only numeric values

# The Delete Operation Example

*Operation:*

Delete the EMPLOYEE tuple with Ssn = '999887777'.

*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.

*Operation:*

Delete the EMPLOYEE tuple with Ssn = '333445555'.

*Result:* This deletion will result in even worse referential integrity violations, because the tuple involved is referenced by tuples from the EMPLOYEE, DEPARTMENT, WORKS\_ON, and DEPENDENT relations.

# DB State Example

Employee Relation

EmpID	EmpName	Salary	DeptNo
1234	Ahmed Salem	2000	1
2456	Maha Karim	3000	1
4658	Sherif Maged	2500	2
9840	Yasser Amin	4000	2
3945	Magda Farid	5000	3

Department Relation

DeptNo	DeptName
1	Sales
2	Human Resource
3	Finance
4	Warehouses

# Insert Operation Example

**Operation:** Insert < Null, 'Ashraf Mohamed", 2000,1>  
into Employee

**Result:** This insertion is rejected because it violates  
entity integrity

**Operation:** Insert < 1234, 'Ashraf Mohamed", 2000,1>  
into Employee

**Result:** This insertion is rejected because it violates key  
constraint. No two tuples can have the same value for  
the primary key

# Insert Operation Example (Cont.)

**Operation:** Insert < 8954, 'Ashraf Mohamed", 2000,5>  
into Employee

**Result:** This insertion is rejected because it violates  
referential integrity

**Operation:** Insert < 8678, 'Ashraf Mohamed", lots of  
money,1> into Employee

**Result:** This insertion is rejected because it violates  
domain constraint. Salary attribute accepts only  
numbers



# Delete Operation Example

**Operation:** Delete Department tuple with DeptID = 1

**Result:** Delete operation is rejected because it violates referential integrity

**Operation:** Delete Employee tuple with DeptID = 1

**Result:** Delete operation is accepted