CptS 415 Big Data

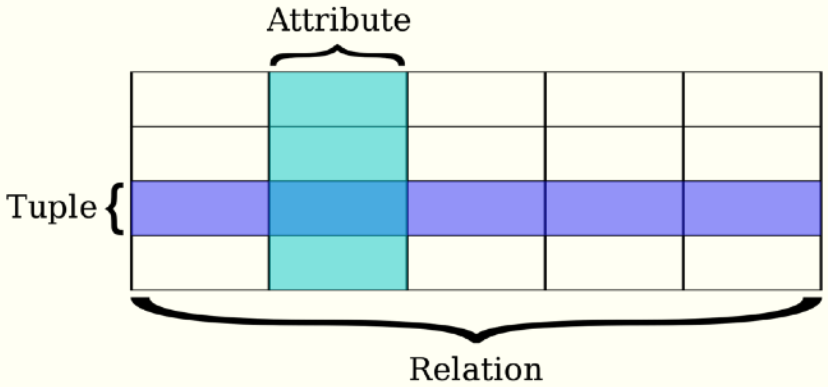
Relational Database

Srini Badri

Acknowledgement: Tinghui Wang

Contents

- Relational Data Model and DBMS
  - Relational Model Concepts
  - Constraints and Schemas
  - Update Operations
  - Constraint Violations



Data Models

- A collection of tools for describing
  - Data
  - Data relationships
  - Data semantics
  - Data constraints
- Relational Model
  - Entity-Relationship Data Model (mainly for database design)
  - Object-based Data Model (Object-oriented and Object-relational)
- Semi-Structured Data Model (XML and Graphs)
- Other Models:
  - Network model
  - Hierarchical model
- Reading: “What goes around comes around”, by Michael Stonebraker

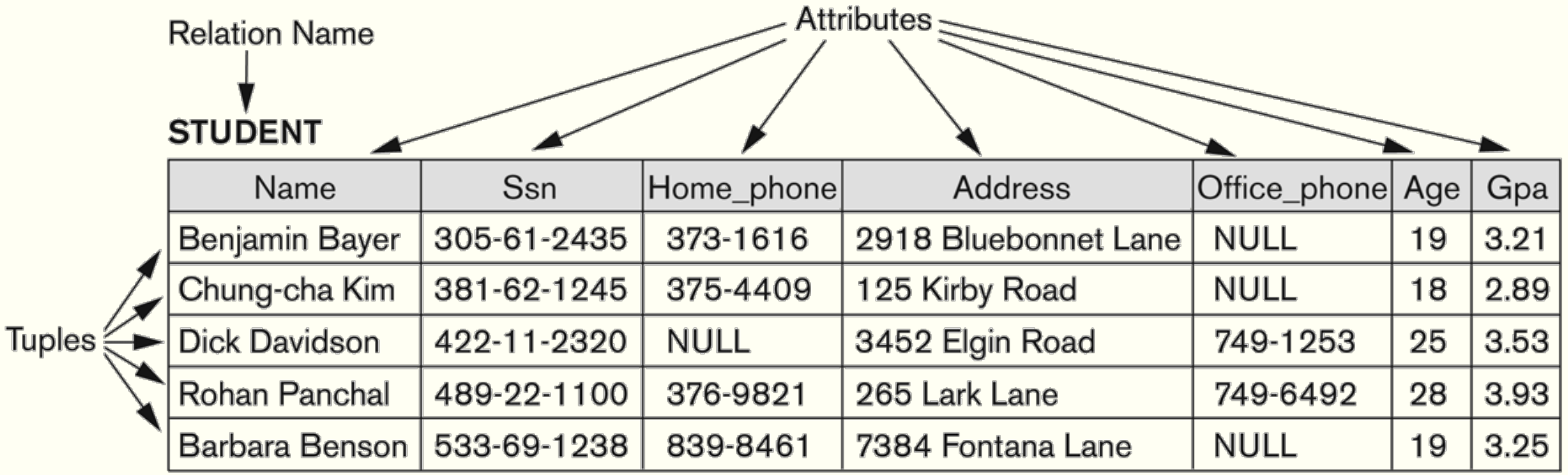
Relational Model Concepts

- A relation is a mathematical concept based on the idea of sets
- The model was first proposed by Dr. E.F. Codd of IBM Research in 1970
  - “A Relational Model for Large Shared Data Banks,” Communications of the ACM, June 1970
- ACM Turing Award winners who are data management researchers
  - 1973: Chales W. Bachman
  - 1981: Edgar F. Codd
  - 1998: Jim Gray
  - 2014: Michael Stonebraker

Informal Definitions

- Informally, a relation looks like a table of values
- A relation typically contains a set of rows
- The data elements in each row represent certain facts that correspond to a real-world entity or relationship
  - In the formal model, rows are called tuples
- Each column has a column header that gives an indication of the meaning of the data items in that column
  - In the formal model, the column header is called an attribute name or just attribute

Example of a Relation

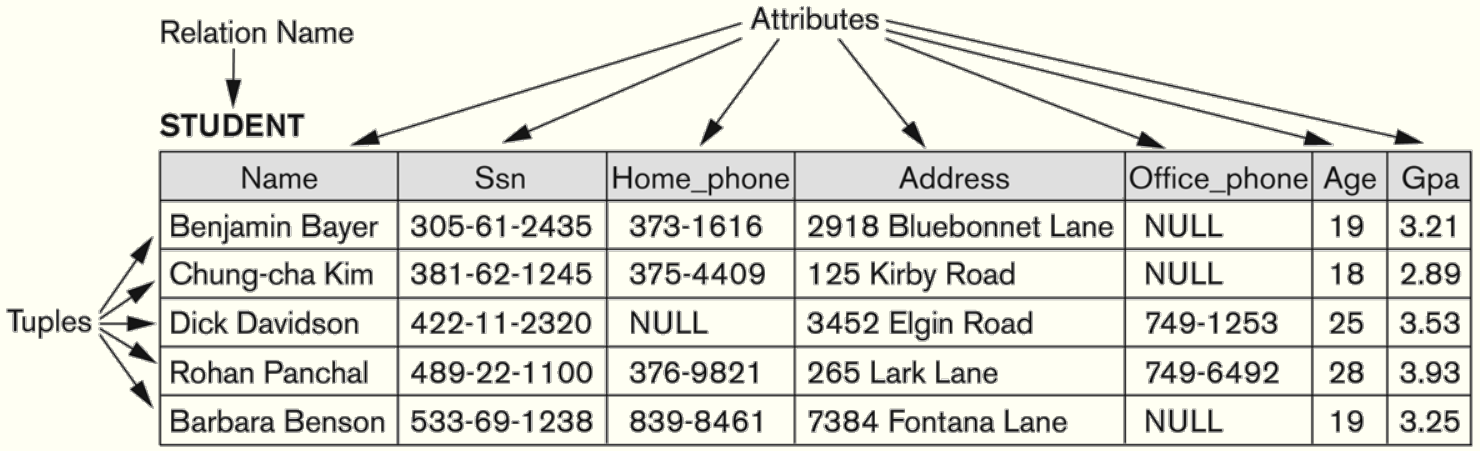


Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	749-1253	25	3.53
Rohan Panchal	489-22-1100	376-9821	265 Lark Lane	749-6492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

Figure 5.1
The attributes and tuples of a relation STUDENT.

Informal Definitions (cont'd)

- 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



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

Figure 5.1
The attributes and tuples of a relation STUDENT.

- Sometimes row-ids or sequential numbers are assigned as keys to identify the rows in a table
  - A.k.a. artificial key or surrogate key

Schema

- The Schema (or description) of a Relation:
  - $R(A_1, A_2, \ldots, A_n)$
  - R is the name of the relation
  - The attributes of the relation are $A_1, A_2, \ldots, A_n$
- Example:

CUSTOMER (Cust-id, Cust-name, Addr, Phone)

Relation Name | Attributes

- Each attribute has a domain or a set of valid values
  - The domain of Cust-id may be 6-digit numbers
  - The customer name can be a string with a maximum length of 255 characters

Tuple

- A tuple is an ordered n-tuple of values
  - Enclosed in angled brackets $< \ldots >$
- Each value is derived from an appropriate domain
- Example

CUSTOMER (Cust-id, Cust-name, Addr, Phone)

A row in the CUSTOMER relation is a 4-tuple, e.g.

<632895, “John Smith”, “101 Main St., Pullman, WA 99163”, “(509)715-2000”>

- A relation is a set of such tuples (rows)

Domain

- 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 also has a data-type or format
  - E.g. “USA_Phone_Numbers” may have a format: (ddd)ddd-dddd, where ‘d’ is a decimal digit
  - Dates have various format, such as yyyy-mm-dd or mm/dd/yyyy, etc.
- The attribute name designates the role played by a domain in a relation
  - Used to interpret the meaning of the data elements corresponding to that attribute
  - E.g. The domain Date may be used to define two attributes named “Invoice-date” and “Payment-date” with different meanings.

State of a Relation

- 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

$$R(A_1, A_2)$$

$$\mathscr{D}(A_1) = \{a,\ b\}$$

$$\mathscr{D}(A_2) = \{c,\ d\}$$

$$\mathscr{D}(A_1) \times \mathscr{D}(A_2) = \{(a,\ c),\ (b,\ c),\ (a,\ d),\ (b,\ d)\}$$

$$\text{Relation State} \subseteq \mathscr{D}(A_1) \times \mathscr{D}(A_2)$$

- Question: What is the difference between a state of a relation and an instance of a relation?
- Answer: A relation instance is a tuple/row in a relation, i.e. one particular combination of attribute values

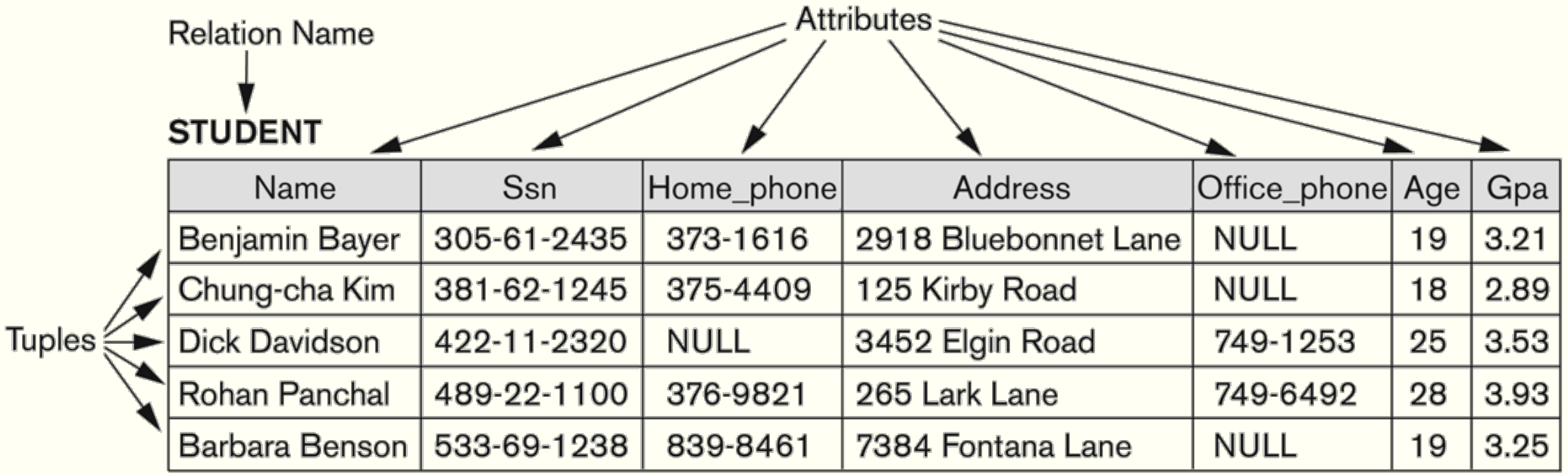
Summary of Formal Definitions

- Schema: $R\left(A_1, A_2, \ldots, A_n\right)$
- Name of the relation: R
- Attributes of the relation: $A_1, A_2, \ldots, A_n$
- State: $\mathrm{r}(R) \subseteq \mathscr{D}\left(A_1\right) \times \mathscr{D}\left(A_2\right) \times \ldots \times \mathscr{D}\left(A_n\right)$
- $\mathrm{r}(R)$ is a specific state (or “value” or “population”) of relation R
- $\mathrm{r}(R)$ is a set of tuples (rows)
  - $\mathrm{r}(R) = \left\{\mathrm{t}_1, \mathrm{t}_2, \ \ldots, \ t_m\right\}$, where each t_i is an n-tuple
  - $t_i = < v_1, v_2, \ldots, v_n >$, where each $v_j \in \mathscr{D}\left(A_j\right)$

Summary: Informal Vs. Formal

Informal Terms	Formal Terms
Table	Relation
Column Header	Attribute
All possible values in a Column	Domain of the Attribute
Row	Tuple
Table Definition	Schema of a Relation
Populated Table	State of the Relation

State of the STUDENT Relation



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

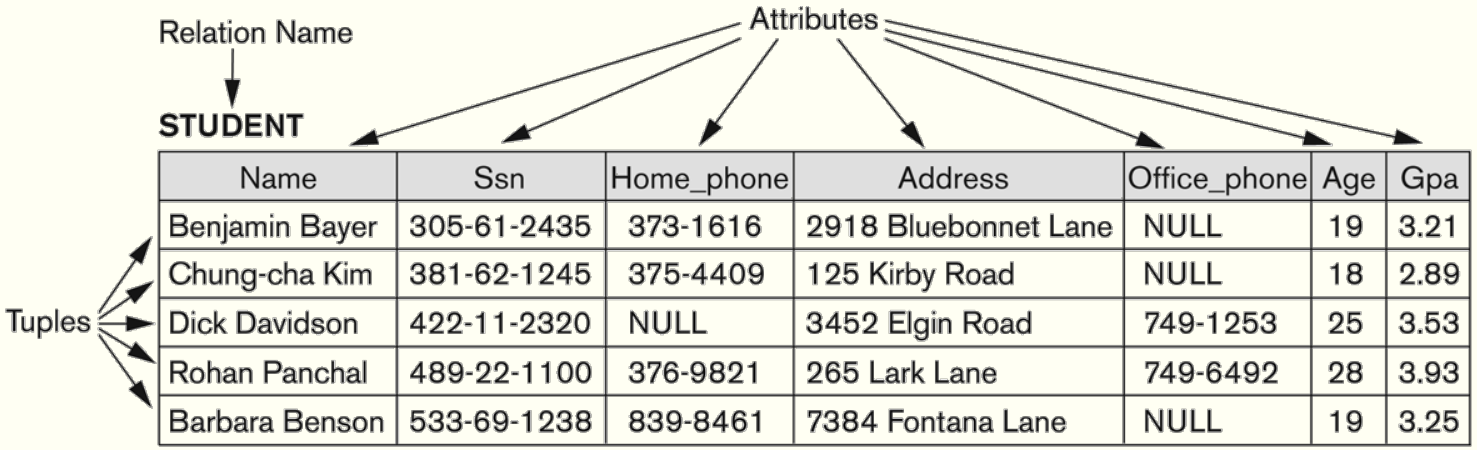
Figure 5.1
The attributes and tuples of a relation STUDENT.

Characteristics of Relations

- Ordering of tuples in a relation state $r(R)$
  - The tuples are not considered to be ordered in a relation state
  - (though they appear in the tabular form)

- Ordering of attributes in a relation schema R (and of values within each tuple):
  - We consider the attributes in a relation $R(A_1, A_2, \ldots, A_n)$ to be ordered
  - We consider the values in a tuple $t_i = < v_1, v_2, \ldots, v_n >$ to be ordered

Same State of STUDENT Relation



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

Figure 5.1
The attributes and tuples of a relation STUDENT.

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

Characteristics of Relations (cont'd)

- Values 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
    - $t = < v_1, v_2, \ldots, v_n >$ is a tuple (row) in the relation state $r(R)$ of a relation $R(A_1, A_2, \ldots, A_n)$
    - $v_j \in \mathcal{D}(A_j)$, i.e. v_j must be a value from domain $\mathcal{D}(A_j)$
  - Component values of a tuple t
    - $v_j = t[A_j] = t.A_j$
  - Similarly, $t[A_u, A_v, \ldots, A_w]$ refers to the sub-tuple of t containing the values $v_u, v_v, \ldots, v_w$ of attributes $A_u, A_v, \ldots, A_w$ respectively in t
  - A special null value is used to represent values that are unknown or inapplicable in certain tuples.

Relational Integrity Constraints

- Constraints are conditions that must hold (true) on all valid relation states.
- There are three main types of constraints in the relational model
  - Key constraints
  - Entity integrity constraints
  - Referential integrity constraints
- Another implicit constraint is the domain constraint
  - Every value in a tuple must be from the domain of its attribute
  - It could be null, if allowed for that attribute

Key Constraints

- Example: Consider the CAR relation schema

CAR (State, Reg#, SerialNo, Make, Model, Year)

Key1: {State, Reg#} Key2: SerialNo

- Super-key is a set of attributes (from the schema) such that there do not exist two tuples of any valid state of the relationship sharing the same super-key.
- What is the largest super-key?
- In general:
  - Any key is a super-key (but not vice versa)
  - Any set of attributes that includes a key is a super key
  - A minimal super-key is a key

Key Constraints (cont'd)

- If a relation has several candidate keys, one is chosen to be the primary key.
  - The primary key attributes are underlined.
- Example:

CAR (State, Reg#, SerialNo, Make, Model, Year)

Primary Key: SerialNo

- The primary key value is used to uniquely identify each tuple in a relation
  - Provides the tuple identity
- Also used to reference the tuple from another tuple
  - General rule: Choose as primary key the smallest of the candidate keys (in terms of size)
  - Not always applicable
  - Choice is sometimes subjective

CAR Table with two candidate keys

Figure 5.4
The CAR relation, with two candidate keys: License_number and Engine_serial_number.

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

COMPANY Database Schema

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

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

Figure 5.5
Schema diagram for the COMPANY relational database schema.

Entity Integrity Constraints

- The primary key attributes PK of each relation schema R in S cannot have NULL values in any tuple of $r(R)$
  - This is because primary key values are used to identify the individual tuple.
  - $t[PK] \neq NULL$ for any tuple $t \in r(R)$
  - If PK has several attributes, NULL is not allowed in any of these attributes

- Note: Other attributes of R may be constrained to disallow NULL values, even though they are not members of primary key

Referential Integrity

- A constraint involving two relations
  - The previous constraints involve a single relation
- Used to specify a relationship among tuples in two relations:
  - The referencing relation and the referenced relation
- Tuples in the referencing relation R1 have attribute FK (called foreign key attributes) that reference the primary key attributes PK of the referenced relation R2
  - A tuple t1 in R1 is said to reference a tuple t2 in R2 if t1[FK] = t2[PK].
- A referential integrity constraint can be displayed in a relational database schema as a directed arc from R1.FK to R2

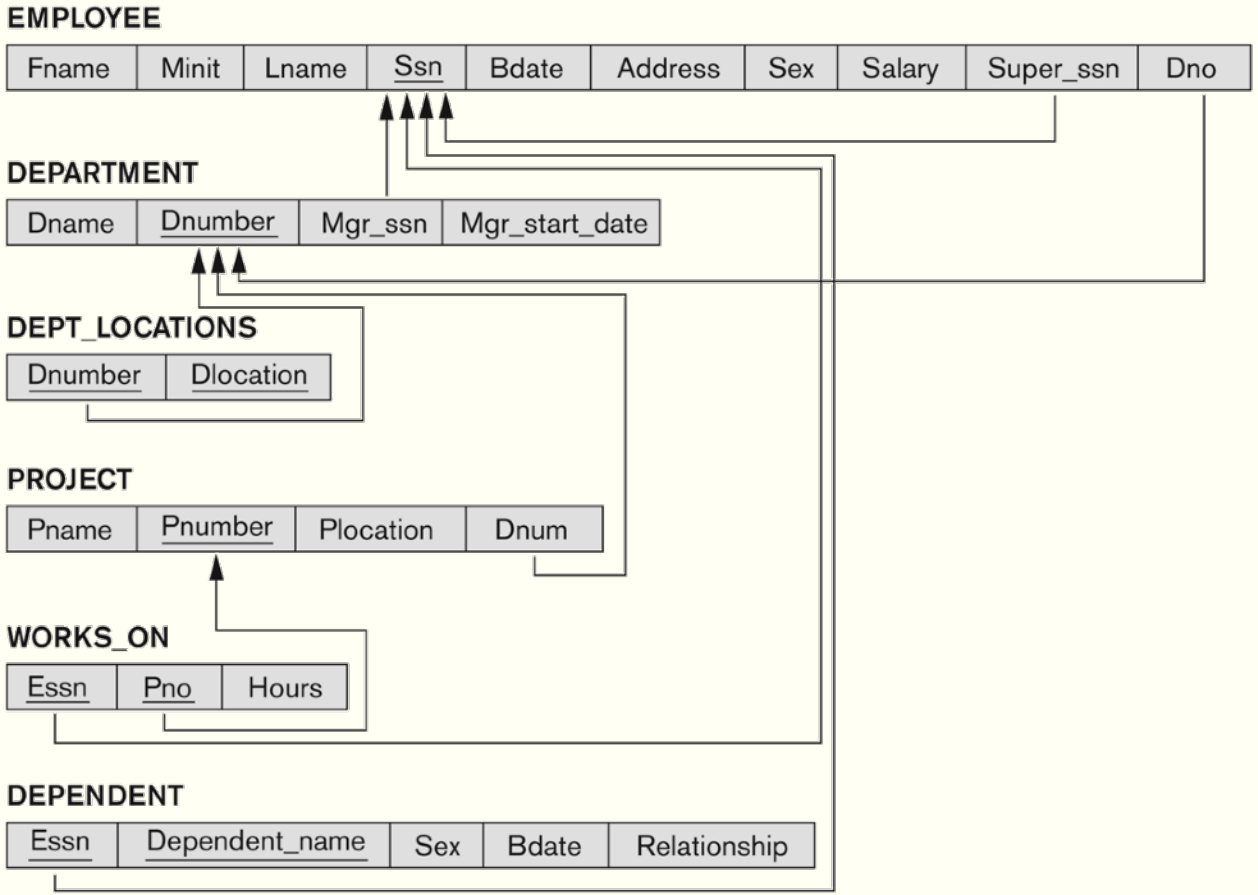
Referential Integrity (or Foreign Key) Constraing

- The value in the foreign key column (or columns) FK of the referencing relation R1 can be either
  - A value of an existing primary key value of a corresponding primary key PK in the referenced relation R2, or
  - A NULL
- In second case, the FK in R1 should not be a part of its own primary key.

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) constraints is displayed as directed arc (arrow) from the FK to the referenced table (or PK of the referenced relation)

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



Other Types of Constraints

- Semantic Integrity Constraints:
  - based on application semantics and cannot be expressed by the model per se
  - Example: “the max. no. of hours per employee for all projects he or she works on is 56 hrs per week”
- A constraint specification language may have to be used to express these
- SQL-99 allows triggers and ASSERTIONS to express for some of these

Populated Database State

- Each relation will have many tuples in its current relation state
- The relational database state is a union of all the individual relation states
- Whenever the database is changed, a new state arises
- Basic Operations for changing the database:
  - INSERT: add a new tuple to a relation
  - DELETE: remove an existing tuple from a relation
  - MODIFY: change an attribute value/multiple attribute values of an existing tuple in a relation

COMPANY Database State

Figure 5.6
One possible database state for the COMPANY relational database schema.

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

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

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

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

PROJECT

Pname	Pnumber	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

Essn	Dependent_name	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

Update Operations on Relations

- INSERT
- DELETE
- MODIFY
- Integrity constraints should not be violated by the update operations
- Updates may propagate to cause other updates automatically.

In case of Integrity Violation

- Cancel the operation that causes the violation (RESTRICT/REJECT option)
- Perform the operation but inform the user of the violation
- Trigger additional updates so the violation is corrected (CASCADE option, SET NULL option)
- Execute a user-specified error correction routine

Possible violations: INSERT

- Domain constraint:
  - if one of the attribute values provided for the new tuple is not of the specified attribute domain
- Key constraint:
  - if the value of a key attribute in the new tuple already exists in another tuple in the relation
- Referential integrity:
  - if a foreign key value in the new tuple references a primary key value that does not exist in the referenced relation
- Entity integrity:
  - if the primary key value is null in the new tuple
  - If the primary key value is not unique

Possible Violations: DELETE

- If the primary key value of the tuple being deleted is referenced from other tuples in the database
  - Can be remedied by several actions: RESTRICT, CASCADE, SET NULL
  - RESTRICT option: reject the deletion
  - CASCADE option: delete all tuples that contain the foreign key value
  - SET NULL option: set the foreign keys of the referencing tuples to NULL
- One of the above options must be specified during database design for each foreign key constraint

Possible Violations: UPDATE/MODIFY

- May violate domain constraint and NOT NULL constraint on an attribute being modified
- Updating the primary key (PK):
  - Similar to a DELETE followed by an INSERT
  - Need to specify similar options to DELETE
- Updating a foreign key (FK):
  - May violate referential integrity
- Updating an ordinary attribute (neither PK nor FK):
  - Can only violate domain constraints

Summary

- Relational Model Concepts
  - Definitions
  - Characteristics of relations
- Discussed Relational Model Constraints and Relational Database Schemas
  - Domain constraints’
  - Key constraints
  - Entity integrity
  - Referential integrity
- Described the Relational Update Operations
- Dealing with Constraint Violations