THE RELATIONAL DATA MODEL

CHAPTER 3 (6/E)

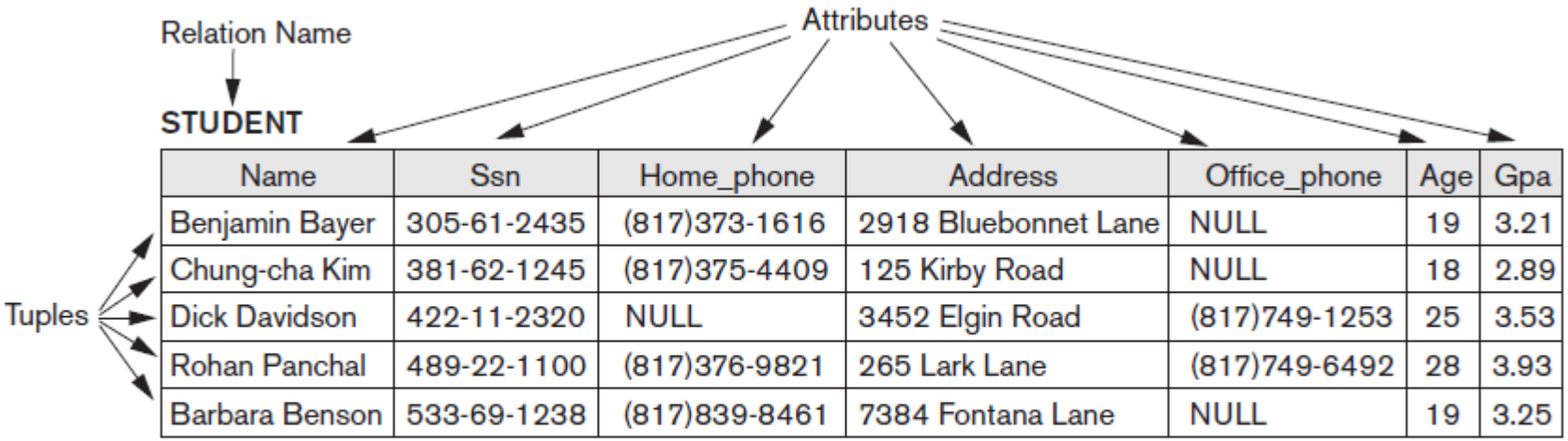
CHAPTER 5 (5/E)

LECTURE OUTLINE

- Relational Model Concepts
- Relational Database Schemas
- Brief History of Database Applications (from Section 1.7)

RELATIONAL MODEL CONCEPTS

- Represent data as a collection of *relations*
  - Think of a relation as a table of values

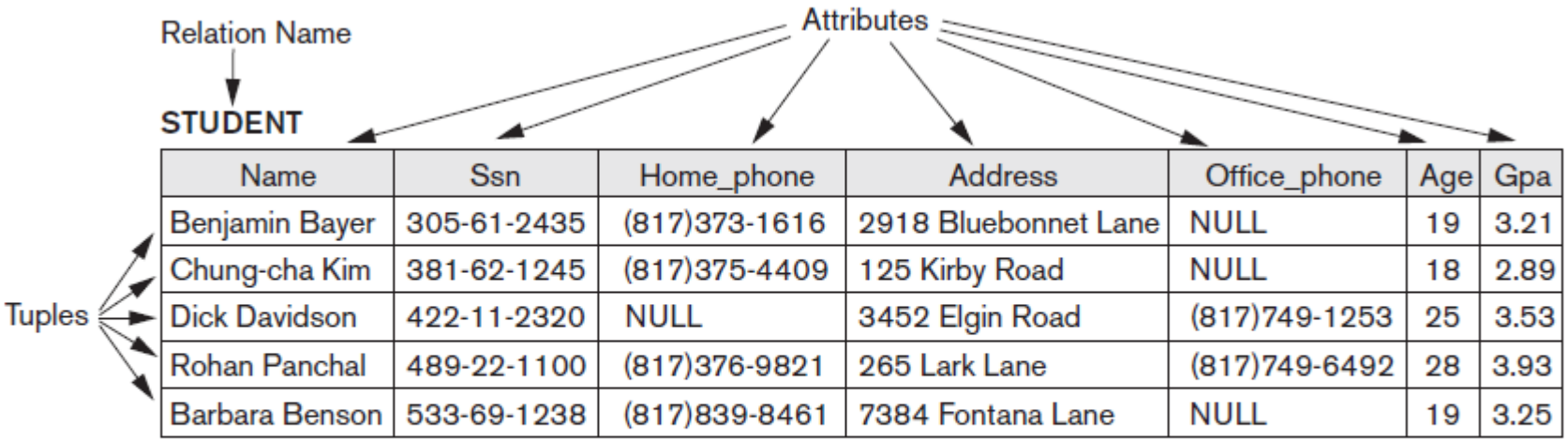


Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	(817)373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	(817)375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	(817)749-1253	25	3.53
Rohan Panchal	489-22-1100	(817)376-9821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

- Each row (*tuple*) represents a record of related data values
  - Facts that typically correspond to a real-world entity or relationship
- Each column (*attribute*) holds a corresponding value for each row
  - Columns associated with a data type (*domain*)
  - Each column header: *attribute name*

RELATIONAL MODEL (CONT'D.)

- Represent data as a collection of *relations*
  - Think of a relation as a table of values



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

- **Schema** describes table
  - Table name, attribute names and types
- **Instance** denotes the current contents of the table
  - The **relation** (or **relation state**)

MEANING OF A RELATION

- **Assertion**
  - Each tuple in the relation interpreted as a **fact**
  - No other similar facts are of interest to the enterprise
  - e.g., a relation Classlist includes only registered students and all registered students are included in Classlist
    - presence in list ⇔ registered student

- **Predicate**
  - Values in each tuple interpreted as values that satisfy predicate
  - e.g., Name of student having ID 83201556 is Lee Wong

DOMAINS

- **Domain** is a set of *atomic* values
  - { 0, 1, 2, … }
  - { Jo Smith, Dana Jones, Ashley Wong, Y. K. Lee, … }

- **Atomic**: Each value indivisible

- Domains specified by **data type** rather than by enumeration
  - Integer, string, date, real, etc.
  - Can be specified by format
    - e.g., *(ddd)ddd-dddd* for phone numbers (where *d* represents a digit)

SCHEMAS AND ATTRIBUTES

- **Relation schema**
  - A relation name R and a list of attributes A_1, A_2, ... , A_n
  - Denoted by $R(A_1, A_2, ..., A_n)$

- **Attribute** A_i
  - Name of a role in the relation schema R
  - Associated with a domain **dom(A_i)**
  - Attribute names do not repeat within a relation schema, but domains can repeat

- **Degree** (or **arity**) of a relation
  - Number of attributes n in its relation schema

FORMALIZATION

- **Relation** (or **relation state**)
  - Instance of relation schema $R(A_1, A_2, A_3, \ldots, A_n)$
  - Set $r = \{ t_1, t_2, \ldots, t_m \}$ of ***n*-tuples** (n is the degree of the relation)
    - Unordered
    - No duplicates
  - Each n-tuple t
    - Ordered list of n values $t = < v_1, v_2, \ldots, v_n >$
    - Each value v_i $(1 \leq i \leq n)$ is an element of $\text{dom}(A_i)$
  - **Finite subset** of the **Cartesian product** of the domains defining R
    - $\text{rel}(R) \subseteq (\text{dom}(A_1) \times \text{dom}(A_2) \times \ldots \times \text{dom}(A_n))$

- Because of updates, relations are **time-varying**
  - rel(R) is relation state at a given time
  - Reflects only (and all) the valid tuples that represent a particular state of the real world

RELATIONAL MODEL NOTATION

- Uppercase letters *Q*, *R*, *S* denote relation names
- Corresponding lowercase letters *q*, *r*, *s* denote corresponding relation states
- Uppercase letters *A*, *B*, *C*, … , *H* denote attributes
  - Attribute *A* can be qualified with the relation name *R* to which it belongs using the dot notation
    - e.g., *R.A*
- Lower case letters *t*, *u*, *v* denote tuples

ALTERNATIVE DEFN OF RELATION

- Tuple considered as a function from attributes to values
  - $t_j : \{A_1, A_2, A_3, \ldots, A_n\} \rightarrow \text{dom}(A_1) \cup \text{dom}(A_2) \cup \ldots \cup \text{dom}(A_n)$
  - Use notation $t_j[A_i]$ or $t_j.A_i$ to refer to tuple's value v_i from dom(A_i)
  - Similarly, $t_j[A_u, A_w, ..., A_z]$ and $t_j.(A_u, A_w, ..., A_z)$ refer to the sub-tuple of values $<v_u, v_w, ..., v_z>$ from t_j for attributes $A_u, A_w, ..., A_z$
- Therefore, a tuple is a set of <attribute, value> pairs
- Example: *attendee*(*id*, *givenName*, *surname*, *company*, *dateOfBirth*)
  - t = <10483, John, Doe, IBM, 1978-11-05>
  - t[id] = 10483, t[givenName] = John, t[surname] = Doe, etc.
  - t.id = 10483, t.givenName = John, t.surname = Doe, etc.
  - t = { <id, 10483>, <givenName, John>, <surname, Doe>, <company, IBM>, <dateOfBirth, 1978-11-05> }

VALUES IN TUPLES

- Each value in a tuple is **atomic**
  - **Flat relational model** (as opposed to *nested* relational model)
  - Composite and multivalued attributes not allowed

- *Composite attributes* must be split into simple component attributes
  - e.g., *Waterloo, Ontario* treated as atomic or split into two attributes to store *Waterloo* separately from *Ontario*

- *Multivalued attributes* must be represented by separate relations
  - Recall: *Director* could be stored as attribute of FILM because only one director per film assumed, but multiple characters in a film implies that ROLE must have its own relation.

NULL VALUES

- Each domain may be augmented with a special value called NULL
  - Represent the values of attributes that may be unknown or may not apply to a tuple

- Interpretations for NULL values
  - Nothing is known about the value
  - Value exists but is (currently) not available
  - Value undefined (i.e., attribute does not apply to this tuple)

- If an attribute for a tuple is mapped to NULL, we cannot make any assumption about the value for that attribute (for that tuple)
  - e.g., Ashley's telephone number is NULL could mean
    - Ashley doesn't have a phone
    - Ashley has a phone but we don't know the number (perhaps withheld)
    - Ashley has a phone that has no number
    - Ashley may or may not have a phone, but regardless we don't have a number for Ashley

BRIEF HISTORY

- Relational model
  - Formulated by E.F.Codd (IBM) before 1970
  - First commercial implementations available in early 1980s
  - Predominant database model used today
- (earlier) Hierarchical and network models
  - Preceded the relational model
  - Pointer-based
  - Access relied on record-at-a-time navigation
- (later) Object-oriented applications and more complex databases
  - Object-relational model
  - Used in specialized applications: engineering design, multimedia publishing, manufacturing systems, etc.

RECENT HISTORY

- Interchanging data on the Web for e-commerce
  - Extended markup language (XML) primary standard for interchanging data among various types of DBs and Web pages

- Extending DB (and DBMS) capabilities for new applications
  - Extensions to support specialized requirements for applications
  - Enterprise resource planning (ERP)
    - e.g., SAP
  - Customer relationship management (CRM)
    - e.g., SAP
  - Enterprise content management (ECM)
    - e.g., Open Text
    - includes extensions to information retrieval (IR) to deal with documents (proposals, reports, news articles, etc.)

LECTURE SUMMARY

- Characteristics differentiate relations from ordinary tables or files
- Schemas vs. instances (states)
- Formal definitions for relations and tuples
- NULL values