

Chapter 2: Introduction to the Relational Model

INFSCI1022: Database Management Systems

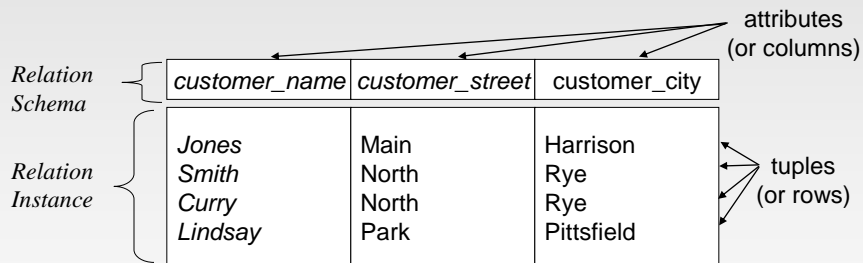
Textbook: Database System Concepts - 6th Edition, 2010

Instructor: Vladimir Zadorozhny, GIST, University of Pittsburgh

1

Relation is a Table

- Relational Database stores data in form of *relations*.
- Roughly speaking relation is a table.
- A data item is represented by a *row* in a table (a *tuple*).
- Order of tuples is irrelevant (tuples may be stored in an arbitrary order).



Relation Name: *customer*

2

Attribute Names and Types

- Each attribute of a relation has a name
- The set of allowed values for each attribute is called the **domain** of the attribute

account:

<i>account_number</i>	<i>branch_name</i>	<i>balance</i>
A-101	Downtown	500
A-102	Perryridge	400
A-201	Brighton	900
A-215	Mianus	700
A-217	Brighton	750
A-222	Redwood	700
A-305	Round Hill	350

- Attribute *balance* is of *integer* domain
- Attributes *account_number* and *branch_name* are *strings*

- Attribute values are (normally) required to be **atomic**; that is, indivisible
 - Multivalued attribute values are not atomic
 - Composite attribute values are not atomic
- The special value *null* is a member of every domain (more about it later)

3

Relation Schema

- *List of attributes is known as a relation schema*

Example: *Customer_schema* = (*customer_name*, *customer_street*, *customer_city*)

customer:

<i>customer_name</i>	<i>customer_street</i>	<i>customer_city</i>
Adams	Spring	Pittsfield
Brooks	Senator	Brooklyn
Curry	North	Rye
Glenn	Sand Hill	Woodside
Green	Walnut	Stamford
Hayes	Main	Harrison
Johnson	Alma	Palo Alto
Jones	Main	Harrison
Lindsay	Park	Pittsfield
Smith	North	Rye
Turner	Putnam	Stamford
Williams	Nassau	Princeton

4

Database

- A database consists of multiple relations
- Information about an enterprise is broken up into parts, with each relation storing one part of the information
 - account* : stores information about accounts
 - depositor* : stores information about which customer owns which account
 - customer* : stores information about customers
- Storing all information as a single relation such as *bank(account_number, balance, customer_name, ..)* results in multiple problems. Will be considered in **Normalization theory (Chapter 7)** that deals with how to design relational schemas

5

Example of a Database

account

<i>account_number</i>	<i>branch_name</i>	<i>balance</i>
A-101	Downtown	500
A-215	Mianus	700
A-102	Perryridge	400
A-305	Round Hill	350
A-201	Brighton	900
A-222	Redwood	700
A-217	Brighton	750

customer

<i>customer_name</i>	<i>customer_street</i>	<i>customer_city</i>
Adams	Spring	Pittsfield
Brooks	Senator	Brooklyn
Curry	North	Rye
Glenn	Sand Hill	Woodside
Green	Walnut	Stamford
Hayes	Main	Harrison
Johnson	Alma	Palo Alto
Jones	Main	Harrison
Lindsay	Park	Pittsfield
Smith	North	Rye
Turner	Putnam	Stamford
Williams	Nassau	Princeton

depositor

<i>customer_name</i>	<i>account_number</i>
Hayes	A-102
Johnson	A-101
Johnson	A-201
Jones	A-217
Lindsay	A-222
Smith	A-215
Turner	A-305

6

Keys

- Let K is a list of attributes of a schema R . K is a **key** (also called a **superkey**) of R if values for K are sufficient to identify a unique tuple of each possible relation r of that schema
 - Example: $\{customer_name, customer_street\}$ and $\{ssn\}$ are both keys of *Customer*, if no two customers can possibly have the same name.

<u>ssn</u>	customer_name	customer_street	customer_city
111111111	Jones	Main	Harrison
222222222	Smith	North	Rye
333333333	Curry	North	Rye
444444444	Lindsay	Park	Pittsfield

7

Candidate Keys and Primary Key

- K is a **candidate key** if K is *minimal* (i.e., no subset of K is a key).
- Relation schema may have more than one candidate key. Example: $\{ssn\}$ and $\{customer_name\}$ are candidate keys for *Customer*, since they are superkeys (assuming no two customers can possibly have the same name, or the same ssn),
- Among all candidate keys we must select one **Primary Key** (e.g. ssn)

<u>ssn</u>	customer_name	customer_street	customer_city
111111111	Jones	Main	Harrison
222222222	Smith	North	Rye
333333333	Curry	North	Rye
444444444	Lindsay	Park	Pittsfield

8

Foreign Keys

- **Foreign key**: Set of fields in one relation that is used to 'refer' to a tuple in another relation. (Must correspond to primary key of the second relation.) Like a 'logical pointer'.
- E.g. *sid* is a foreign key referring to Students:
 - Enrolled(*sid*: string, *cid*: string, *grade*: string)
 - If all foreign key constraints are enforced, *referential integrity* is achieved, i.e., no dangling references.

Enrolled

sid	cid	grade
53666	Carnatic101	C
53666	Reggae203	B
53650	Topology112	A
53666	History105	B

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

Textbook: Database System Concepts - 6th Edition, 2010

V.Zadorozhny

End of Chapter 2

INFSCI1022: Database Management Systems

Textbook: Database System Concepts - 6th Edition, 2010

Instructor: Vladimir Zadorozhny, GIST, University of Pittsburgh

10