

IT420: Database Management and Organization

The Relational Model
18 January 2006

Adina Crăiniceanu
www.cs.usna.edu/~adina

1

Database Design Process

- Requirements analysis
- Conceptual design: Entity-Relationship Model
- Logical design: transform ER model into relational schema
- Schema refinement: Normalization
- Physical tuning

2

Goals

- Understand:
 - The relational model
 - Relational model terminology

3

The Relational Model

- Introduced in 1970 by E.F. Codd (IBM)
- Turing Award in 1980
- Now the standard model for commercial DBMS products
 - Vendors: IBM, Oracle, Microsoft, Sybase, etc

4

Relational Database

- A **relational database** is a set of relations
- A **relation** is a two-dimensional table
- **Relation schema** describes the column heads for the table
 - Relation name
 - Column names
 - Column types

5

A Relation

EMPLOYEE(EmployeeNumber:integer,
FirstName:string,
LastName:string,
Department:string,
Email:string,
Phone:integer)

EmployeeNumber	FirstName	LastName	Department	Email	Phone
100	Jerry	Johnson	Accounting	JJ@somewhere.com	236-9987
200	Mary	Abemathy	Finance	MA@somewhere.com	444-8898
300	Liz	Smathers	Finance	LS@somewhere.com	777-0098
400	Tom	Caruthers	Accounting	TC@somewhere.com	236-9987
500	Tom	Jackson	Production	TJ@somewhere.com	444-9980
600	Eleanore	Caldera	Legal	EC@somewhere.com	767-0900
700	Richard	Bandalone	Legal	RB@somewhere.com	767-0900

6

Relation

- All entries in a column are of the same kind
 - Each column has a unique name
 - Cells of the table hold a single value
 - The order of the columns is not important
 - The order of the rows is not important
 - No two rows may be identical
-
- Rows contain data about entity instances
 - Columns contain data about attributes of the entity

7

A Relation with Values of Varying Length

EmployeeNumber	FirstName	LastName	Department	Email	Phone	Comment
100	Jerry	Johnson	Accounting	JJ@somewhere.com	236-9987	Joined the Accounting Department in March after completing his MBA at night. Will sit for CPA exam this fall.
200	Mary	Abernathy	Finance	MA@somewhere.com	444-9898	
300	Liz	Smathers	Finance	LS@somewhere.com	777-0098	
400	Tom	Canthers	Accounting	TC@somewhere.com	236-9987	
500	Tom	Jackson	Production	TJ@somewhere.com	444-9990	
600	Eleanore	Caldera	Legal	EC@somewhere.com	767-0900	
700	Richard	Bandalone	Legal	RB@somewhere.com	767-0900	Is a full time consultant to legal on a retainer basis.

8

Tables That Are Not Relations: Multiple Entries per Cell

EmployeeNumber	FirstName	LastName	Department	Email	Phone
100	Jerry	Johnson	Accounting	JJ@somewhere.com	236-0000
200	Mary	Abernathy	Finance	MA@somewhere.com	444-9898
300	Liz	Smathers	Finance	LS@somewhere.com	777-0098
400	Tom	Canthers	Accounting	TC@somewhere.com	236-0000, 236-0991, 236-0991
500	Tom	Jackson	Production	TJ@somewhere.com	444-9990
600	Eleanore	Caldera	Legal	EC@somewhere.com	262-0000
700	Richard	Bandalone	Legal	RB@somewhere.com	767-0900, 767-0011

9

Tables That Are Not Relations: Table with Required Row Order

EmployeeNumber	FirstName	LastName	Department	Email	Phone
100	Jerry	Johnson	Accounting	JJ@somewhere.com	236-9987
200	Mary	Abemathy	Finance	MA@somewhere.com	444-8898
300	Liz	Smathers	Finance	LS@somewhere.com	777-0098
400	Tom	Caruthers	Accounting	TC@somewhere.com	236-9987
					Fax: 236-9987
					Home: 555-7171
500	Tom	Jackson	Production	TJ@somewhere.com	444-9980
600	Eleanore	Caldera	Legal	EC@somewhere.com	767-0900
					Fax: 236-9987
					Home: 555-7171
700	Richard	Bandalone	Legal	RB@somewhere.com	767-0900

10

Alternative Terminology

- Although not all tables are relations, the terms table and relation are normally used interchangeably
- The following sets of terms are equivalent:

Table	Column	Row
Relation	Attribute	Tuple
File	Field	Record

11

Integrity Constraints (IC)

- IC: condition that must be true for any instance of the database; e.g., domain constraints
- ICs are **specified** when schema is **defined**
- ICs are **checked** when relations are **modified**
- A **legal instance** of a relation is one that satisfies all specified ICs
- DBMS should not allow illegal instances

12

Keys

- A **key** is a combination of one or more columns that is used to identify rows in a relation
- A **composite key** is a key that consists of two or more columns

13

Keys (cont.)

- A set of columns is a **key** for a relation if :
 1. No two distinct rows can have same values in all key columns, and
 2. This is not true for any subset of the key
- Part 2 false? A **superkey**
- E.g., EmployeeNumber is a key for EMPLOYEE (What about LastName? SSN?)
- The set {EmployeeNumber, Email} is a superkey

14

Candidate and Primary Keys

- A **candidate key** is a key
- A **primary key** is a candidate key selected as the primary means of identifying rows in a relation:
 - There is one and only one primary key per relation
 - The primary key may be a composite key
 - The ideal primary key is short, numeric and never changes

15

Surrogate Keys

- A **surrogate key** as an artificial column added to a relation to serve as a primary key:
 - DBMS supplied
 - Short, numeric and never changes – an ideal primary key!
 - Has artificial values that are meaningless to users
- Remember Access

16

Surrogate Keys

NOTE: The primary key of the relation is underlined below:

- RENTAL_PROPERTY without surrogate key:
`RENTAL_PROPERTY (Street, City,
State/Province, Zip/PostalCode, Country, Rental_Rate)`
- RENTAL_PROPERTY with surrogate key:
`RENTAL_PROPERTY (PropertyID, Street, City,
State/Province, Zip/PostalCode, Country, Rental_Rate)`

17

Foreign Keys

- A **foreign key** is the primary key of one relation that is placed in another relation to form a link between the relations
- 'Refer' to a row in another relation
- 'Logical pointer'
- The term refers to the fact that key values are *foreign* to the relation in which they appear as foreign key values

18

Foreign Keys

NOTE: The primary keys of the relations are underlined and any foreign keys are in *italics* in the relations below:

DEPARTMENT (DepartmentName, BudgetCode, ManagerName)
EMPLOYEE (EmployeeNumber, EmployeeName, *DepartmentName*)

19

The Referential Integrity Constraint

- A **referential integrity constraint**: the values of the foreign key must exist as primary key values in the corresponding relation
- No 'dangling references'

20

Foreign Key with a Referential Integrity Constraint

NOTE: The primary key of the relation is underlined and any foreign keys are in *italics* in the relations below:

DEPARTMENT (DepartmentName, BudgetCode, ManagerName)
EMPLOYEE (EmployeeNumber, EmployeeName, *DepartmentName*)

Where EMPLOYEE.DepartmentName must exist in DEPARTMENT

- Can you name a data model w/o referential integrity?

21
