Chapter 3
Relational Data Model
Content

- Introduction
- Concepts
- Constraints
- From E/R diagram to relational design
Introduction

- Was first introduced by E. F. Codd
  - “A Relation Model for Large Shared Data Banks”, Communications of ACM, 1970

- Commercial implementation
  - By IBM
  - Oracle (1979)
  - By Sybase
  - By Microsoft
    - SQL Server (1989)
    - Access (1992)
Introduction

- Open source implementation
  - MySQL
    - By MySQL AB, 1995
  - PostgreSQL
    - Ingres project at the University of California, Berkeley, 1980s
    - By many developers, released in 1996
  - SQLite
    - By D. Richard Hipp working for General Dynamics, 2000
Introduction

- Provide a simple way to represent data
  - The relation: a two-dimensional table

- The theoretical background
  - Set theory of mathematical logic
Content

- Introduction

- Concepts
  - Relation
  - Attribute
  - Schema
  - Tuple
  - Domain
  - Characteristics of relation
  - Notations

- Contraints

- From E/R diagram to relational design
**Relation**

- Relational model presents the DB as a collection of *relations*
  
  - A relation = a two-dimensional table

<table>
<thead>
<tr>
<th>FNAME</th>
<th>LNAME</th>
<th>BIRTHDATE</th>
<th>ADDRESS</th>
<th>SEX</th>
<th>SALARY</th>
<th>DNO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tung</td>
<td>Nguyen</td>
<td>12/08/1955</td>
<td>638 NVC Q5</td>
<td>Nam</td>
<td>40000</td>
<td>5</td>
</tr>
<tr>
<td>Hang</td>
<td>Bui</td>
<td>07/19/1968</td>
<td>332 NTH Q1</td>
<td>Nu</td>
<td>25000</td>
<td>4</td>
</tr>
<tr>
<td>Nhu</td>
<td>Le</td>
<td>06/20/1951</td>
<td>291 HVH QPN</td>
<td>Nu</td>
<td>43000</td>
<td>4</td>
</tr>
<tr>
<td>Hung</td>
<td>Nguyen</td>
<td>09/15/1962</td>
<td>Ba Ria VT</td>
<td>Nam</td>
<td>38000</td>
<td>5</td>
</tr>
</tbody>
</table>

Relation name is EMPLOYEE
Relation

- Includes
  - Name
  - Set of columns
    - Fixed
    - Named
    - Has data types
  - Set of rows
    - Changed by time

- A row ~ A real-world entity or relationship
- A relation ~ An entity set or relationship
Attribute

- The names for columns of the relation
- Describes the meaning of entries in the column below

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FNAME</td>
<td>Tung</td>
</tr>
<tr>
<td>LNAME</td>
<td>Nguyen</td>
</tr>
<tr>
<td>BIRTHDATE</td>
<td>12/08/1955</td>
</tr>
<tr>
<td>ADDRESS</td>
<td>638 NVC Q5</td>
</tr>
<tr>
<td>SEX</td>
<td>Nam</td>
</tr>
<tr>
<td>SALARY</td>
<td>40000</td>
</tr>
<tr>
<td>DNO</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FNAME</td>
<td>Hang</td>
</tr>
<tr>
<td>LNAME</td>
<td>Bui</td>
</tr>
<tr>
<td>BIRTHDATE</td>
<td>07/19/1968</td>
</tr>
<tr>
<td>ADDRESS</td>
<td>332 NTH Q1</td>
</tr>
<tr>
<td>SEX</td>
<td>Nu</td>
</tr>
<tr>
<td>SALARY</td>
<td>25000</td>
</tr>
<tr>
<td>DNO</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FNAME</td>
<td>Nhu</td>
</tr>
<tr>
<td>LNAME</td>
<td>Le</td>
</tr>
<tr>
<td>BIRTHDATE</td>
<td>06/20/1951</td>
</tr>
<tr>
<td>ADDRESS</td>
<td>291 HVH QPN</td>
</tr>
<tr>
<td>SEX</td>
<td>Nu</td>
</tr>
<tr>
<td>SALARY</td>
<td>43000</td>
</tr>
<tr>
<td>DNO</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FNAME</td>
<td>Hung</td>
</tr>
<tr>
<td>LNAME</td>
<td>Nguyen</td>
</tr>
<tr>
<td>BIRTHDATE</td>
<td>09/15/1962</td>
</tr>
<tr>
<td>ADDRESS</td>
<td>Ba Ria VT</td>
</tr>
<tr>
<td>SEX</td>
<td>Nam</td>
</tr>
<tr>
<td>SALARY</td>
<td>38000</td>
</tr>
<tr>
<td>DNO</td>
<td>5</td>
</tr>
</tbody>
</table>

- All values in a column are of the same data type
Schema

- Schema of a relation
  - Name
  - Set of attributes

Relation schema

EMPLOYEE(SSN, FNAME, LNAME, BIRTHDATE, ADDRESS, SEX, SALARY, DNO)

a set, not a list
Database schema

- A design consist of one or more relational schemas

**Database schema**

EMPLOYEE(SSN, FNAME, LNAME, BIRTHDATE, ADDRESS, SEX, SALARY, DNO)
DEPARTMENT(DNUMBER, DNAME, MGRSSN, MGRSTARTDATE)
DEPT_LOCATION(DNUMBER, DLOCATION)
DEPENDENT(SSN, DEPENDENT_NAME, Sex, BDate, Relationship)
PROJECT(PNAME, PNUMBER, PLOCATION, DNUM)
Tuple

- Row of a relation
  - Except the header row containing the attribute names

- Contains many components
  - One component for each attributes of the relation

$$\langle \text{Tung, Nguyen, 12/08/1955, 638 NVC Q5, Nam, 40000, 5} \rangle$$
Domain

- Each attribute of a relation associates with a **domain**
  - A particular elementary type

- A component of each tuple
  - Is **atomic**
  - Has a **value** that belongs to the domain of the corresponding attribute

- Example
  - FName: string, DOM(FName): the set of strings
  - Salary: integer, DOM(Salary): the set of integers
Characteristics of relation

- The order of tuples in a relation is not important

<table>
<thead>
<tr>
<th>LNAME</th>
<th>FNAME</th>
<th>BIRTHDATE</th>
<th>ADDRESS</th>
<th>SEX</th>
<th>SALARY</th>
<th>DNO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nguyen</td>
<td>Tung</td>
<td>12/08/1955</td>
<td>638 NVC Q5</td>
<td>Nam</td>
<td>40000</td>
<td>5</td>
</tr>
<tr>
<td>Bui</td>
<td>Hang</td>
<td>07/19/1968</td>
<td>332 NTH Q1</td>
<td>Nu</td>
<td>25000</td>
<td>4</td>
</tr>
<tr>
<td>Le</td>
<td>Nhu</td>
<td>06/20/1951</td>
<td>291 HVH QPN</td>
<td>Nu</td>
<td>43000</td>
<td>4</td>
</tr>
<tr>
<td>Nguyen</td>
<td>Hung</td>
<td>09/15/1962</td>
<td>null</td>
<td>Nam</td>
<td>38000</td>
<td>5</td>
</tr>
</tbody>
</table>

- The order of values in a tuple is important

\(<\text{Nguyen, Tung}, 12/08/1955, 638 \text{ NVC Q5}, \text{Nam}, 40000, 5>\)

Differs from

\(<\text{Nguyen, Tung}, 12/08/1955, 638 \text{ NVC Q5}, 40000, \text{Nam}, 5>\)
Characteristics of relation

- Each value of components in a tuple
  - Atomic or
  - NULL

- Relations are sets of tuples, not lists of tuples
  - There are no identical tuples
Relational model notation

- Relation schema
  - Given $A_1, A_2, ..., A_n$ are attributes
  - Has domains $D_1, D_2, ..., D_n$ respectively
  - Is denoted by $R(A_1:D_1, A_2:D_2, ..., A_n:D_n)$
  
  - Example
    - EMPLOYEE(SSN:DOM(integer), FNAME:DOM(STRING), LNAME:DOM(STRING), BIRTHDAY:DOM(DATE), ADDRESS:DOM(STRING), SEX:DOM(STRING), SALARY:DOM(INTEGER), DNO:DOM(INTEGER))

- The degree of a relation is the number of attributes of its relation schema
  - EMPLOYEE is a relation schema of degree 8
Relational model notation

- Relation instances
  - A relation \( r \) of relation schema \( R(A_1, A_2, \ldots, A_n) \), denoted by \( r(R) \), is a set of tuples \( r = \{t_1, t_2, \ldots, t_k\} \)
  - Where each \( t_i \) is an ordered list of \( n \) values \( t_i = \langle v_1, v_2, \ldots, v_n \rangle \)
    - Each \( v_j \) is a member of \( \text{DOM}(A_j) \) or NULL value

<table>
<thead>
<tr>
<th>FNAME</th>
<th>LNAME</th>
<th>BIRTHDATE</th>
<th>ADDRESS</th>
<th>SEX</th>
<th>SALARY</th>
<th>DNO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tung</td>
<td>Nguyen</td>
<td>12/08/1955</td>
<td>638 NVC Q5</td>
<td>Nam</td>
<td>40000</td>
<td>5</td>
</tr>
<tr>
<td>Hang</td>
<td>Bui</td>
<td>07/19/1968</td>
<td>332 NTH Q1</td>
<td>Nu</td>
<td>25000</td>
<td>4</td>
</tr>
<tr>
<td>Nhu</td>
<td>Le</td>
<td>06/20/1951</td>
<td>291 HVH QPN</td>
<td>Nu</td>
<td>43000</td>
<td>4</td>
</tr>
<tr>
<td>Hung</td>
<td>Nguyen</td>
<td>09/15/1962</td>
<td>null</td>
<td>Nam</td>
<td>38000</td>
<td>5</td>
</tr>
</tbody>
</table>
Summary of denotations

- The relation schema $R$ of the degree $n$
  - $R(A_1, A_2, \ldots, A_n)$
- The attribute set of $R$
  - $R^+$
- Relations
  - $R, S, P, Q$
- Tuples
  - $t, u, v$
- The domain of the attribute $A$
  - $\text{DOM}(A)$
- The value at the attribute $A$ of the $t^{th}$ tuple
  - $t.A$ or $t[A]$
Content

- Introduction
- Concepts
- **Constraints**
  - Superkey
  - Key
  - Primary key
  - Reference
  - Foreign key
- From E/R diagram to relational design
Constraint

- Integrity constraint
  - Rules, conditions need to satisfy for all of instances of relational database

- Constraints
  - Defined when the relation schema is modeled
  - Checked when the data in relations are modified
Superkey

- **Definition**
  - Assume SK is a subset of attributes of R, SK ≠ ∅
  - SK is the super key if
    \[ \forall r, \forall t_1, t_2 \in r, t_1 \neq t_2 \Rightarrow t_1[SK] \neq t_2[SK] \]
    Any two distinct tuples have the different values at the superkey

- **Remark**
  - No two tuples in any state r of R can have the same value for superkey
  - Every relation has at least one default superkey
Example

- Find all superkeys of R

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>1</td>
<td>10</td>
<td>a</td>
</tr>
<tr>
<td>x</td>
<td>2</td>
<td>20</td>
<td>a</td>
</tr>
<tr>
<td>y</td>
<td>1</td>
<td>40</td>
<td>b</td>
</tr>
<tr>
<td>y</td>
<td>1</td>
<td>40</td>
<td>c</td>
</tr>
<tr>
<td>z</td>
<td>1</td>
<td>50</td>
<td>d</td>
</tr>
</tbody>
</table>
Key

Definition
- Assume $K$ is a subset of attributes of $R$, $K \neq \emptyset$
- $K$ is a key if
  - $K$ is a superkey of $R$
  - $\forall K' \subseteq K, K' \neq K', K'$ is not the superkey of $R$
  - $\forall K', K' \subset K, K' \subset K$, $K'$ is not the superkey of $R$

Remark
- The value of a key identifies uniquely each tuple in the relation
- A key is a *property* of the relation schema
  - Time-invariant: a constraint should hold on every valid state
- A key is determined from the meaning of attributes
- A relation has more than one key
Primary key

- Designate one of the key as the primary key (PK)
  - The value for PK is constrained to be not null
  - Underline the attributes of PK when displaying its relation schema

- The choice of PK
  - Influence some implementation issues
  - Usually with a single attribute or a small number of attributes
R refers to S when
- An attribute A of a tuple in relation R receives a value from an attribute B of relation S
  - Must refer to an existing tuple

<table>
<thead>
<tr>
<th>FNAME</th>
<th>LNAME</th>
<th>BIRTHDATE</th>
<th>ADDRESS</th>
<th>SEX</th>
<th>SALARY</th>
<th>DNO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tung</td>
<td>Nguyen</td>
<td>12/08/1955</td>
<td>638 NVC Q5</td>
<td>Nam</td>
<td>40000</td>
<td>5</td>
</tr>
<tr>
<td>Hang</td>
<td>Bui</td>
<td>07/19/1968</td>
<td>332 NTH Q1</td>
<td>Nu</td>
<td>25000</td>
<td>4</td>
</tr>
<tr>
<td>Nhu</td>
<td>Le</td>
<td>06/20/1951</td>
<td>291 HVH QPN</td>
<td>Nu</td>
<td>43000</td>
<td>4</td>
</tr>
<tr>
<td>Hung</td>
<td>Nguyen</td>
<td>09/15/1962</td>
<td>Ba Ria VT</td>
<td>Nam</td>
<td>38000</td>
<td>5</td>
</tr>
</tbody>
</table>

S
- DNAME: Nghien cuu
  - DNUMBER: 5
Foreign key

- Examine two relation schemas R and S
  - Assume FK is a set of attributes of R, FK ≠ ∅
  - FK is a foreign key of R if
    - Attributes in FK have the same domains as the primary key attributes PK of S
    - A value of FK in a tuple t₁ ∈ R
      * Either is a value of PK for some tuple t₂ ∈ S
      * Or is null

- Example

  EMPLOYEE(SSN, FNAME, LNAME, BIRTHDATE, ADDRESS, SEX, SALARY, DNO)
  
  DEPARTMENT(DNAME, DNUMBER)
Foreign key

- Remark
  - An attribute can both participate in PK and participate in FK
  - A FK can refer to its own relation
  - Many FKS might refer to the same primary key
  - Referential constraint = Foreign key constraint
Example
Content

- Introduction
- Concepts
- Constraints

- From E/R diagrams to relational design
  - Rules
Rules

- (1) Entity set
  - Turn each entity set (except weak entity set) into a relation with the same set of attributes
(2) Relationship
- (2a) Many-Many
  - Create a new relation
    * Relation name is the name of the relationship
    * Attributes are the key attributes of connected entity sets
Rules

- (2) Relation
  - (2b) One-Many
    - Adding the key of the many-relation to the one-relation

\[ \text{EMPLOYEE(}SSN, \text{ FNAME, LNAME, BIRTHDATE, ADDRESS, SEX, SALARY, DNUMBER)} \]
Rules

- (2) Relationship
  - (2c) One-One
    - Either adding the key of a relation to another relation
    - Or adding the key to both relations

DEPARTMENT (DNUMBER, DNAME, SSN, STARTDATE)
Rules

- (3) Weak entity set
  - Turn into a relation
    - Has the same name
    - Add the key of related entity sets

**Diagram:**
- EMPLOYEE
  - SSN
  - LNAME
  - FNAME
  - SEX
  - BIRTHDATE
  - SALARY
  - ADDRESS

- DEPENDENT
  - SSN, NAME, SEX, BIRTHDATE, RELATIONSHIP

- Dependents_of
  - (1,1)
  - (1,n)

**Note:**
- CuuDuongThanCong.com
- https://fb.com/tailieudientucntt
(4) Subclass
- Turn into a relation
  • Has the same name
  • Add the key of the superclass