四川大學
SICHUAN UNIVERSITY

Database System Concepts

Relational Model

伍元凯

College of Computer Science (Software), Sichuan University

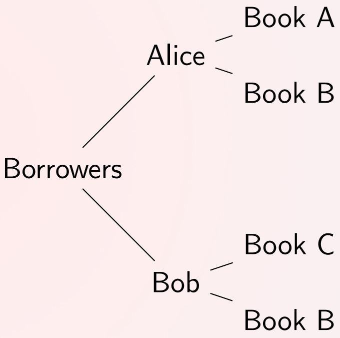
wuyk0@scu.edu.cn

2024/03/01

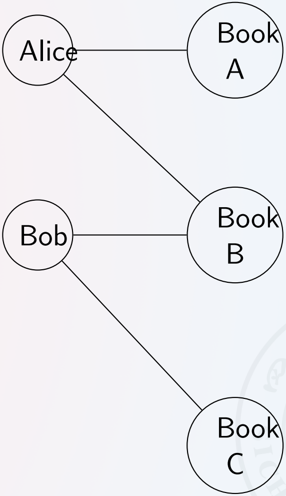
海纳百川 有容乃大

In 1970 there were two popular approaches used to construct database management: the tree-based (called hierarchical) and graph-based (network)

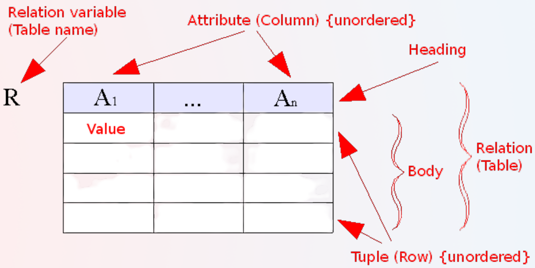
- To answer a specific database request, an application programmer, skilled in performing disk-oriented optimization, must write a complex program to navigate through the database.
- When the structure of the database changes, as it will whenever new kinds of information are added, application programs usually need to be rewritten.



Tree-based



Netwo



Relational model concepts

- the term relation（关系） is used to refer to a table(表)
- the term tuple（元组、记录） is used to refer to a row(行)
- the term attribute（属性） is used to refer to a column(列)

- The current values (relation instance) of a relation are specified by a table
- An element t of r is a tuple, represented by a row in a table

Relation instance for "Sales"

Sales_ID	Product_ID	Product_Name	Category	Quantity	Price
1	101	Product A	A	10	100
2	102	Product B	B	20	200
3	103	Product C	A	30	300

Order of tuples is irrelevant (tuples may be stored in an arbitrary order)

Customer ID	Customer Name	City
101	Alice	New York
102	Bob	Boston
103	Charlie	San Francisco

Customer ID	Customer Name	City
102	Bob	Boston
101	Alice	New York
103	Charlie	San Francisco

The set of allowed values for each attribute is called the domain（域） of the attribute

The special value null （空值） is a member of every domain

The null value causes complications in the definition of many operations

DOMAIN Salary $\in \mathbb{R}^+$.

```
CREATE TABLE Employees (
    Employee_ID INT PRIMARY KEY,
    First_Name VARCHAR(50),
    Last_Name VARCHAR(50),
    Hire_Date DATE,
    Salary DECIMAL(10,2),
    Department_ID INT,
    CONSTRAINT salary_range CHECK (Salary >= 0)
);
```

Attribute values are (normally) required to be atomic（原子的）; that is, indivisible

Table 1: Violating atomic

Customer ID	Order Details
101	Pen, Pencil
102	Notebook, Pen, Pencil

Employee			
Attribute	**Data Type**	**Domain**	**Description**
Employee_ID	INTEGER	$\mathbb{N}$	Unique identifier for each employee
First_Name	VARCHAR(50)	Σ^{50}	First name of the employee
Last_Name	VARCHAR(50)	Σ^{50}	Last name of the employee
Gender	CHAR(1)	{'M', 'F'}	Gender of the employee
Date_of_Birth	DATE	$\mathbb{D}$	Date of birth of the employee
Salary	DECIMAL(10,2)	$\mathbb{R}^+$	Salary of the employee

Employee Relation Schema

$Employee(Employee\_ID, First\_Name, Last\_Name, Gender, Date\_of\_Birth, Salary)$

Employee_ID	First_Name	Last_Name	Gender	Date_of_Birth	Salary
001	John	Smith	M	1990-02-08	100,000
002	Jane	Doe	F	1985-06-21	80,000
003	Bob	Johnson	M	1975-10-12	120,000
004	Alice	Lee	F	1995-03-15	70,000
005	Michael	Chen	M	1980-12-01	90,000

Employee Relation Instance

A superkey K is a **set** of attributes $A_1, A_2, ..., A_n$ such that no two tuples (rows) in a relation have the same values for all of the attributes in K. Allow us to identify uniquely a tuple in the relation.

ID	FirstName	LastName	Email	Phone
1	John	Smith	john.smith@example.com	555-1234
2	Jane	Doe	jane.doe@example.com	555-5678
3	Bob	Johnson	bob.johnson@example.com	555-9012
4	Alice	Williams	alice.williams@example.com	555-3456

Superkey: $\{ID\}$ and $\{ID, FirstName\}$

Superkey K is a candidate key(候选健) if K is minimal. Example: $\{ID\}$ is a candidate key for Instructor. One of the candidate keys is selected to be the primary key（主键）.

A set of relations with primary keys underlined:

Employees(<u>EmployeeID</u>, FirstName, LastName, HireDate, Address)
Departments(<u>DepartmentID</u>, DepartmentName, ManagerID)
Projects(<u>ProjectID</u>, ProjectName, Budget)
EmployeeProjects(<u>EmployeeID</u>, <u>ProjectID</u>, HoursWorked)

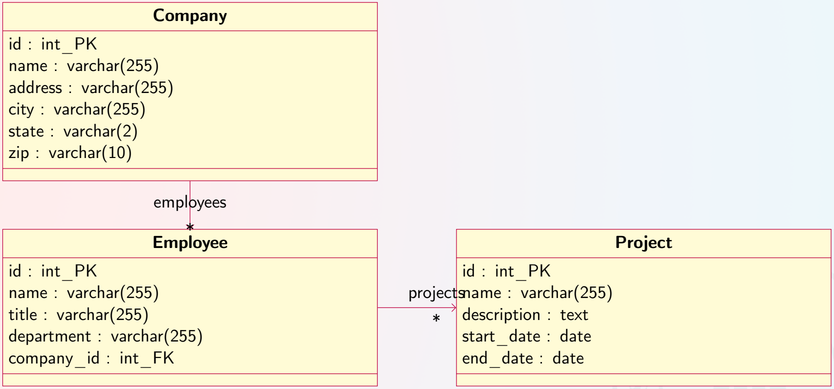
Address should not be the primary key, since it is likely to change.

Foreign key（外键）

constraint: value in one relation must appear in another

Foreign Key.

```sql
CREATE TABLE Students (
  StudentID INT PRIMARY KEY,
  FirstName VARCHAR(50),
  LastName VARCHAR(50),
);
CREATE TABLE Courses (
  CourseID INT PRIMARY KEY,
  CourseName VARCHAR(50),
);
CREATE TABLE Enrollments (
  EnrollmentID INT PRIMARY KEY,
  StudentID INT,
  CourseID INT,
  FOREIGN KEY (StudentID) REFERENCES Students(StudentID),
  FOREIGN KEY (CourseID) REFERENCES Courses(CourseID)
);
```


Company
id : int_PK
name : varchar(255)
address : varchar(255)
city : varchar(255)
state : varchar(2)
zip : varchar(10)
employees
*
Employee
id : int_PK
name : varchar(255)
title : varchar(255)
department : varchar(255)
company_id : int_FK
projects
*
Project
id : int_PK
name : varchar(255)
description : text
start_date : date
end_date : date

Five Basic Operations of Relational Algebra

1. **Selection（选择）**：selects a subset of rows that satisfy a given condition $\sigma_C(R)$
2. **Projection（投影）**：selects a subset of columns from a table $\Pi_{A_1,A_2,\ldots,A_n}(R)$
3. **Union（并）**：combines two tables with the same schema into one table $R \cup S$
4. **Product（笛卡尔积）**：returns the cartesian product of two tables$R \times S$
5. **Join（连接）**：combines two tables by matching rows based on a common attribute $R \bowtie_C S$

These operations can be combined to form more complex queries, and are used in SQL and other database query languages to manipulate and retrieve data.

Fundamental operations to retrieve and manipulate tuples in a relation.

- Based on set algebra.

Each operator takes one or more relations as its inputs and outputs a new relation.

- We can "chain" operators together to create more complex operations.

σ	Select
Π	Projection
$\cup$	Union
$\cap$	Intersection
$-$	Difference
$\times$	Product
$\bowtie$	Join

目录

Student_ID	Name	Age
1001	John	22
1002	Alice	20
1003	Bob	21
1004	Emma	19
1005	David	20

例 1

Suppose we want to select all the students who are 20 years old.

$$\sigma_{\text{Age}=20}(\text{Students})$$

Student_ID	Name	Age
1002	Alice	20
1005	David	20

Department	Building
Biology	Biology
Chemistry	Chemistry
Computer Science	Engineering
English	Main Building
Mathematics	Main Building

例 2

Comparisons between two attributes

$$\sigma_{Building=Department}(Department)$$

which gives us the following table:

Department	Building
Biology	Biology
Chemistry	Chemistry

Student ID	Course ID	Grade
001	CSC101	A
001	MAT105	B
002	CSC101	C
003	ENG201	A
003	CSC201	B
004	MAT105	B

例 3

To select only the columns for Student ID and Course ID:

$$\Pi_{\text{Student ID, Course ID}}(\text{Grade})$$

This will give us the following table:

Student ID	Course ID
001	CSC101
001	MAT105
002	CSC101
003	ENG201
003	CSC201
004	MAT105

A	B	C
1	2	3
4	5	6
7	8	9

例 4

Suppose we want to project only the A and B attributes, and divide the value of B by 2:

$$\Pi_{A,B/2}(R)$$

which results in the following relation:

A	B/2
1	1
4	2.5
7	4

目录

Given two relations $R(A_1, A_2, ..., A_m)$ and $S(B_1, B_2, ..., B_n)$, their Cartesian product $R \times S$ is defined as:

$$R \times S = \{(r, s) \mid r \in R, s \in S\}$$

In other words, $R \times S$ contains **all possible combinations** of tuples from R and S.

ID	Name	Department
101	Alice	CS
102	Bob	EE

Course	Instructor
CS101	Smith
EE201	Johnson

The Cartesian product of the two tables above is:

ID	Name	Department	Course	Instructor
101	Alice	CS	CS101	Smith
101	Alice	CS	EE201	Johnson
102	Bob	EE	CS101	Smith
102	Bob	EE	EE201	Johnson

目录

Table A:

ID	Name
1	John
2	Mary
3	Bob

Table B:

ID	Salary
1	50000
3	70000
4	60000

To perform a join operation, we need to specify a join condition that determines how the two tables are joined.

例 5

We can perform a join on the "ID" attribute of Tables A and B as follows:

$$A \bowtie_{A.ID=B.ID} B$$

This will result in the following table:

A.ID	Name	B.ID	Salary
1	John	1	50000
3	Bob	3	70000

Name	Age
John	20
Mary	22

Name	Salary
John	50000
Mary	60000

例 6

Join A and B on Name:

$$A \bowtie_{A.Name=B.Name} B$$

Result without duplication:

A.Name	Age	Salary
John	20	50000
Mary	22	60000

目录

Suppose we have two tables:

ID	Name	Age
1	John	20
2	Mary	22
3	Tom	25

ID	Name	Salary
1	John	50000
2	Mary	60000
4	Bob	70000

$$A_{name} \cup B_{name}$$

Name
John
Mary
Tom
Bob

Result of Set Union Operation on "Name" Attribute

Table A:

Name	Age
John	20
Mary	22

Table B:

Name	Salary
John	50000
Tom	60000

$A = \{John, Mary\}$
$B = \{John, Tom\}$
$A \cap B = \{x | x \in A \text{ and } x \in B\} = \{John\}$

Name
John

Table A:

Name	Age
John	20
Mary	22

Table B:

Name	Salary
John	50000
Tom	60000

$A = \{John, Mary\}$
$B = \{John, Tom\}$
$A - B = \{x | x \in A \text{ and } x \notin B\} = \{Mary\}$

Name
Mary

目录

Assume we have a relation A with attributes x, y, and z, and we want to create a new relation B with attributes x and y where $z > 10$. We can use temporary relation variables to achieve this:

$$T_1 \leftarrow \sigma_{z>10}(A)$$
$$T_2 \leftarrow \Pi_{x,y}(T_1)$$
$$B \leftarrow T_2$$

Finally, we **assign** T_2 to the relation variable B.
Note that we could also combine the selection and projection operators into a single expression:

$$B \leftarrow \Pi_{x,y}(\sigma_{z>10}(A))$$

目录

The **rename** operation is used to rename the attributes of a relation:

$$\rho_{new\_attribute\_name_1/new\_attribute\_name_2/.../new\_attribute\_name_n}(R)$$

where ρ is the rename symbol, $new\_attribute\_name_i$ is the new name for the i^{th} attribute of relation R, and n is the number of attributes in R. For example, suppose we have a relation R with attributes `ID`, `Name`, and `Age`. To rename the `Name` attribute to `FullName`:

$$\rho_{ID/FullName/Age}(R)$$

Note that the rename operation does not modify the original relation R, but creates a new relation with the renamed attributes.

目录

海纳百川 有容乃大

Student:

Student ID	Name
1001	Alice
1002	Bob
1003	Charlie
1004	David

Grade:

Course ID	Student ID	Grade
1	1001	A
1	1002	B
1	1003	C
1	1004	D
2	1001	B
2	1002	C
2	1003	D
2	1004	A

例 7

$$\Pi_{\text{Name}}(\sigma_{\text{Grade} = \text{'A'}}(\text{Grade} \times \text{Student}))$$
$$= \Pi_{\text{Name}}(\sigma_{\text{Grade} = \text{'A'}}(\text{Grade} \bowtie_{\text{Student ID}} \text{Student}))$$

Name
Alice
David

目录

$\div$ operations are performed simultaneously along the horizontal and vertical directions of the relations. Given relations $R(X, Y)$ and $S(Y, Z)$, where X, Y, Z are sets of attributes. $R \div S$ is a vertical operation performed on the projection of attributes X combined with the horizontal join operation over the attribute set Y. The formal definition is as follows:

$$R \div S = \{t_n[X] \mid t_n \in R \wedge \Pi_Y(S) \subseteq Y_x\}$$

where Y_x are objects in R, $x = t_n[X]$, and the result set of $R \div S$ is in attribute X.

X	Y
X1	Y1
X2	Y2
X2	Y3
X2	Y1

R

Y	F
Y1	F1
Y2	F3

S

例 8

Step 1: Finding the same attribute in S and R, and $\Pi_Y(R)$.

Y
Y1
Y2

例 9

Step 2: In the relation R, the attribute column that is different from S is X. The projection of relation R on attribute X without duplicate values is $\{X1, X2\}$.

Step 3: Find the image set Y of attribute X in relation R. Based on the records of relation R, we can obtain the records related to the value $X1$, the records related to $X2$ are as shown in the following table.

X	Y
X1	Y1

X	Y
X2	Y1
X2	Y2
X2	Y3

例 10

Step 4: Determine the containment relationship $R \div S$ is actually about determining whether the image set Y for each value of X in the relation R contains all the values of attribute Y in the relation S. By comparison, it can be found that: The image set of $X1$ only has $Y1$, which cannot contain all values of attribute Y in relation S, so $X1$ is excluded; The image set of $X2$ contains all values of attribute Y in relation S, so the final result of $R \div S$ is $X2$.

X
X2

例 11

We have the following relations:

- Employee (*Employee Number, Name, Gender, Position, Home Address, Department Number*)
- Department (*Department Number, Department Name, Address, Telephone*)
- Health (*Health Card Number, Employee Number, Physical Examination Date, Health Status*)

1. Write a relational algebra expression to find the names and home addresses of all female section chiefs.
2. Write a relational algebra expression to find the names and home addresses of the section chiefs in the "Office".
3. Write a relational algebra expression to find the names and home addresses of the employees in the "Finance Section" who are in "Good" health.

例 12

1. $$\Pi_{\text{Name, Home Address}}(\sigma_{\text{Position='section chiefs'} \wedge \text{Gender='Female'}}(\text{Employee}))$$

2. $$\Pi_{\text{Name, Home Address}}(\sigma_{\text{Department Name='Office'}}(\text{Department}) \bowtie \sigma_{\text{Position='section chiefs'}}(\text{Employee})))$$

3. $$\Pi_{\text{Name, Home Address}}(\text{Employee} \bowtie \sigma_{\text{Health Status='Good'}}(\text{Health}) \bowtie \sigma_{\text{Department Name='Finance Section'}}(\text{Department}))$$

例 13

We have the following relations:

- S(*SNO,SNAME,AGE,SEX,SDEPT*)
- SC(*SNO,CNO,GRADE*)
- C(*CNO,CNAME,CDEPT,TNAME*)

1. Retrieve the course number and course name of the courses taught by Professor Wu
2. Retrieve the student number and name of male students older than 23 years:
3. Retrieve the course names and instructor names of the courses taken by the student with the student number S3:
4. Retrieve the names of female students who have taken at least one of the courses taught by Professor Wu:
5. Retrieve the course numbers of the courses not taken by the student named Wang:
6. Retrieve the course numbers and course names of the courses taken by all students:
7. Retrieve the student numbers of students who are taking courses taught by Professor Liu:

例 14

1. $$\Pi_{\text{CNO, CNAME}}(\sigma_{\text{TNAME}='Wu'}(C))$$

2. $$\Pi_{\text{SNO, SNAME}}(\sigma_{\text{AGE}>23 \wedge \text{SEX}='M'}(S))$$

3. $$\Pi_{\text{CNAME, TNAME}}(\sigma_{\text{SNO}='S3'}(SC) \bowtie (C))$$

4. $$\Pi_{\text{SNAME}}(\sigma_{\text{SEX}='F'}(S) \bowtie SC \bowtie \sigma_{\text{TNAME}='Wu'}(C))$$

5. $$\Pi_{\text{CNO}}(C) - \Pi_{\text{CNO}}(\sigma_{\text{SNAME}='WANG'}(S) \bowtie SC)$$

6. $$\Pi_{\text{CNO, CNAME, SNO}}(C \bowtie SC) \div \Pi_{\text{SNO}}(S)$$

7. $$\Pi_{\text{CNO, SNO}}(SC) \div \Pi_{\text{CNO}}(\sigma_{\text{TNAME}='Wu'}(C))$$

Reading

Codd, Edgar F. "A relational model of data for large shared data banks." Communications of the ACM 13.6 (1970): 377-387.
Codd, E. F. "The 1981 ACM Turing Award Lecture." Communications (1982).

Thanks
End of Chapter 2