Chapter 2

The Relational Model

Ch2 The Relational Model

- ▶ 2.1 The CAP Database
- ▶ 2.2 Naming the Parts of a Database
- ▶ 2.3 Relational Rules
- ▶ 2.4 Keys, Superkeys, and Null Values
- ▶ 2.5 Relational Algebra
- ▶ 2.6 Set-Theoretic Operations
- ▶ 2.7 Native Relational Operations
- ▶ 2.8 The Interdependence of Operations
- ▶ 2.9 Illustrative Examples
- ▶ 2.10 Other Relational Operations

2.2 Naming the Parts of a Database

- Terminology (术语)
  - **Table (Relation)**
    - Old: file of records
  - **Column names (Attributes)**
    - Old: field names of records
  - **Rows (Tuples)**
    - Old: records of a file
  - **Table heading (Schema)**
    - Old: set of attributes

2.2 Naming the Parts of a Database

关系模型	关系数据库管理系统(SQL)	文件系统
Relation （关系）	Table （表）	File of Records
Attribute （属性）	Column （列）	Field
Tuple （元组）	Row （行）	Record
Schema （模式）	Table Heading （表头）	Type of Record

2.2 Naming the Parts of a Database

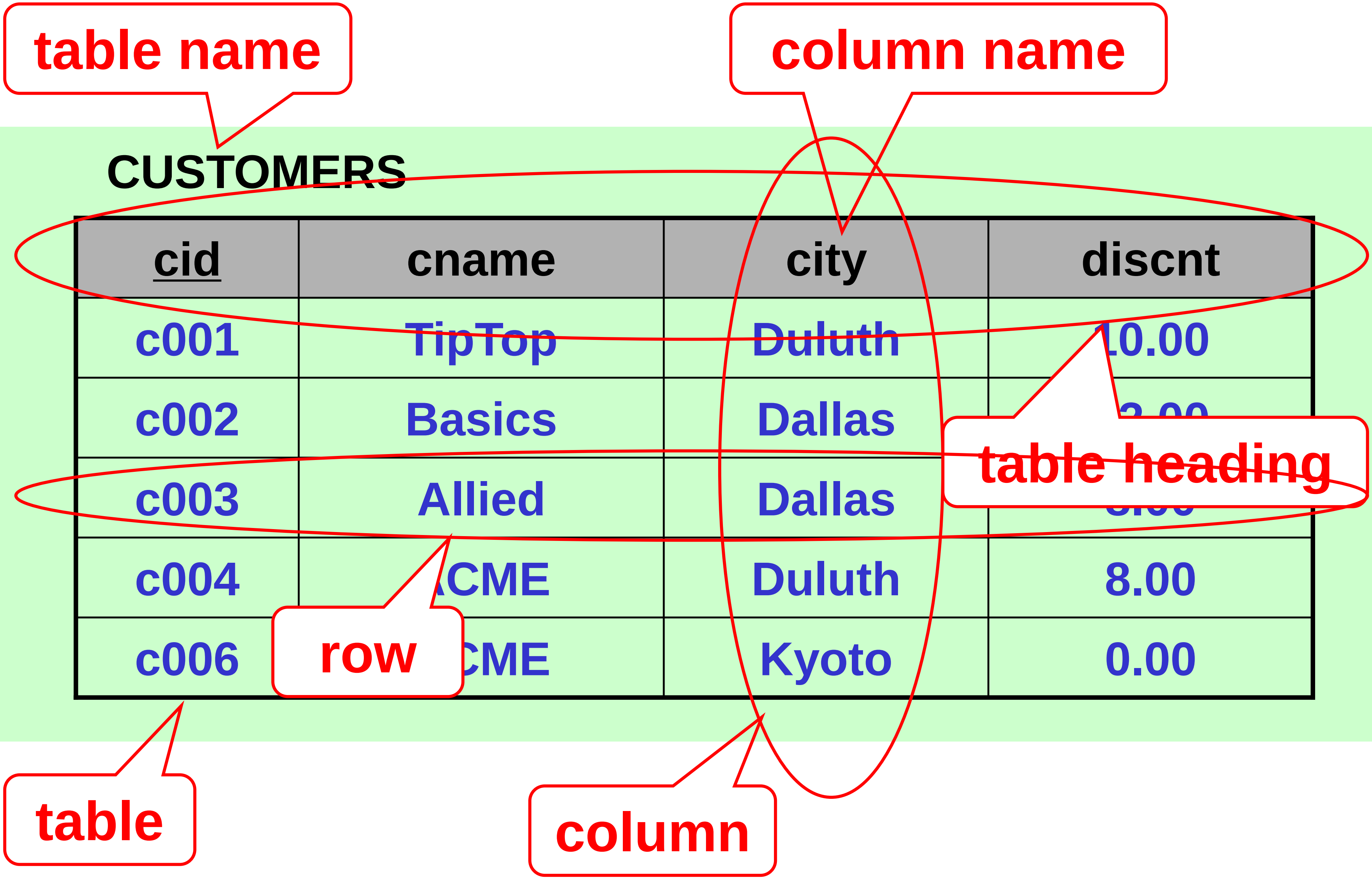


table name

column name

CUSTOMERS

cid	cname	city	discnt
c001	TipTop	Duluth	10.00
c002	Basics	Dallas	12.00
c003	Allied	Dallas	8.00
c004	ACME	Duluth	8.00
c006	ACME	Kyoto	0.00

table heading

row

table

column

2.3 Relational Rules

- Rule 1. First Normal Form Rule
  - Can't have multi-valued fields.

- Rule 2. Access Rows by Content Only Rule
  - Can't say: the third row down from the top.
    - No order to the rows
    - No order to the columns

- Rule 3. The Unique Row Rule
  - Entity Integrity Rule (实体完整性)

2.4 Keys, Superkeys, and Null Values

- Key & Superkey

- Primary Key

2.4 Keys, Superkeys, and Null Values

- Theorem 2.4.2. **Every table T has at least one key.**

- **Proof.** Given a table T with Head(T) = $\{A_1 . . . A_n\}$.

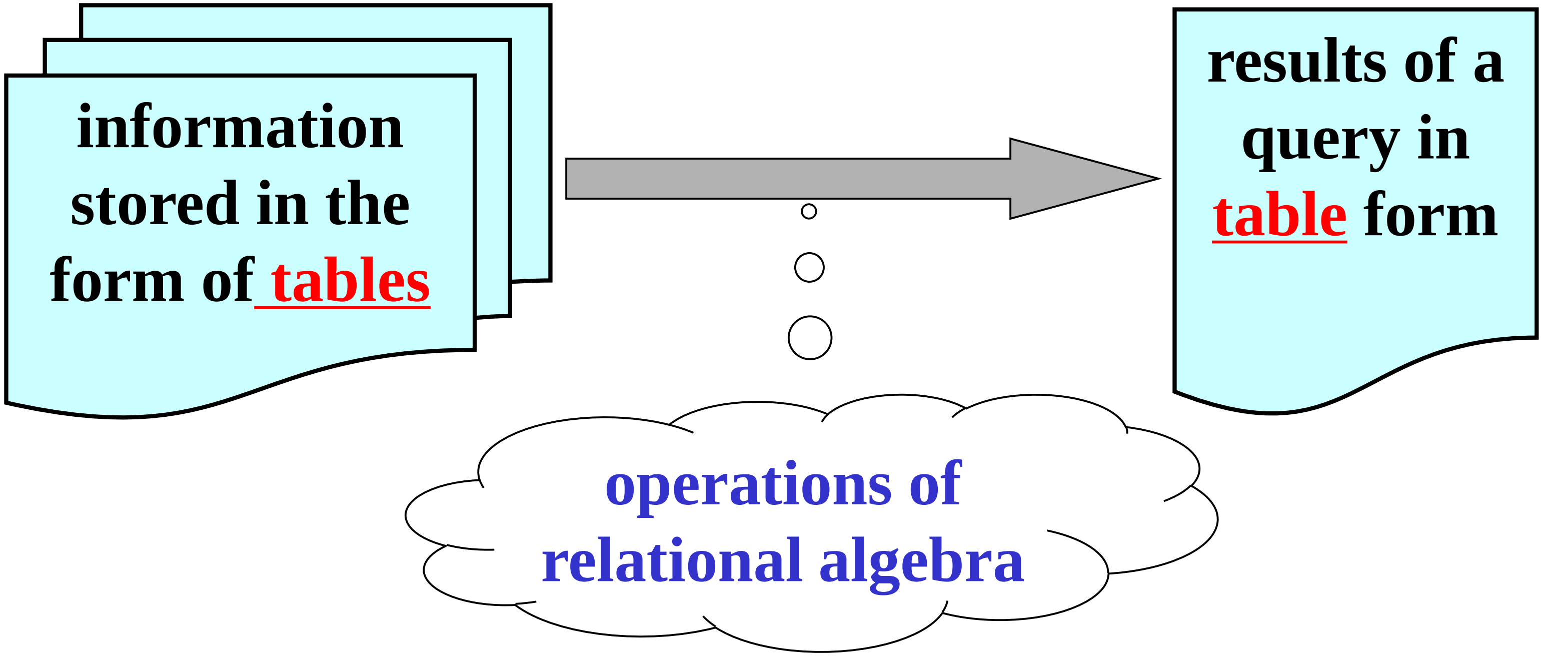
```
令 K := Head(T), 则K是表T的一个超键(superkey)
While (K是表T的超键)
{
    IF (存在K的一个真子集S，且S也是T的超键)
    THEN  { K:=S;  continue; }
    ELSE  break;
}
Return key ‘K’ for T;
```

2.4 Keys, Superkeys, and Null Values

- **Null Values (空值)**

2.5 Relational Algebra

- **Relational Algebra（关系代数）**
  - **abstract language introduced by E. F. Codd**



information stored in the form of tables

results of a query in table form

operations of relational algebra

2.5 Relational Algebra

❑ **set-theoretic operations**

NAME	SYMBOL	FORM	EXAMPLE
UNION (并)	∪	UNION	R ∪ S
INTERSECTION (交)	∩	INTERSECT	R ∩ S
DIFFERENCE (差)	–	MINUS	R – S
PRODUCT (乘积)	×	TIMES	R × S

❑native relational operations

NAME	SYMBOL	FORM	EXAMPLE
PROJECT (投影)	R [ ]	R [ ]	$R[A_{i1},\ldots,A_{ik}]$
	π		$\pi_{Ai1,\ldots,Aik}(R)$
SELECT (选择)	R where C	R where C	R where A_1=5
	δ	$\delta_C(R)$	$\delta_{A1=5}(R)$
JOIN (联接)	∞	JOIN	R ∞ S
DIVISION (除法)	÷	DIVIDEBY	R ÷ S

Difference between join and division

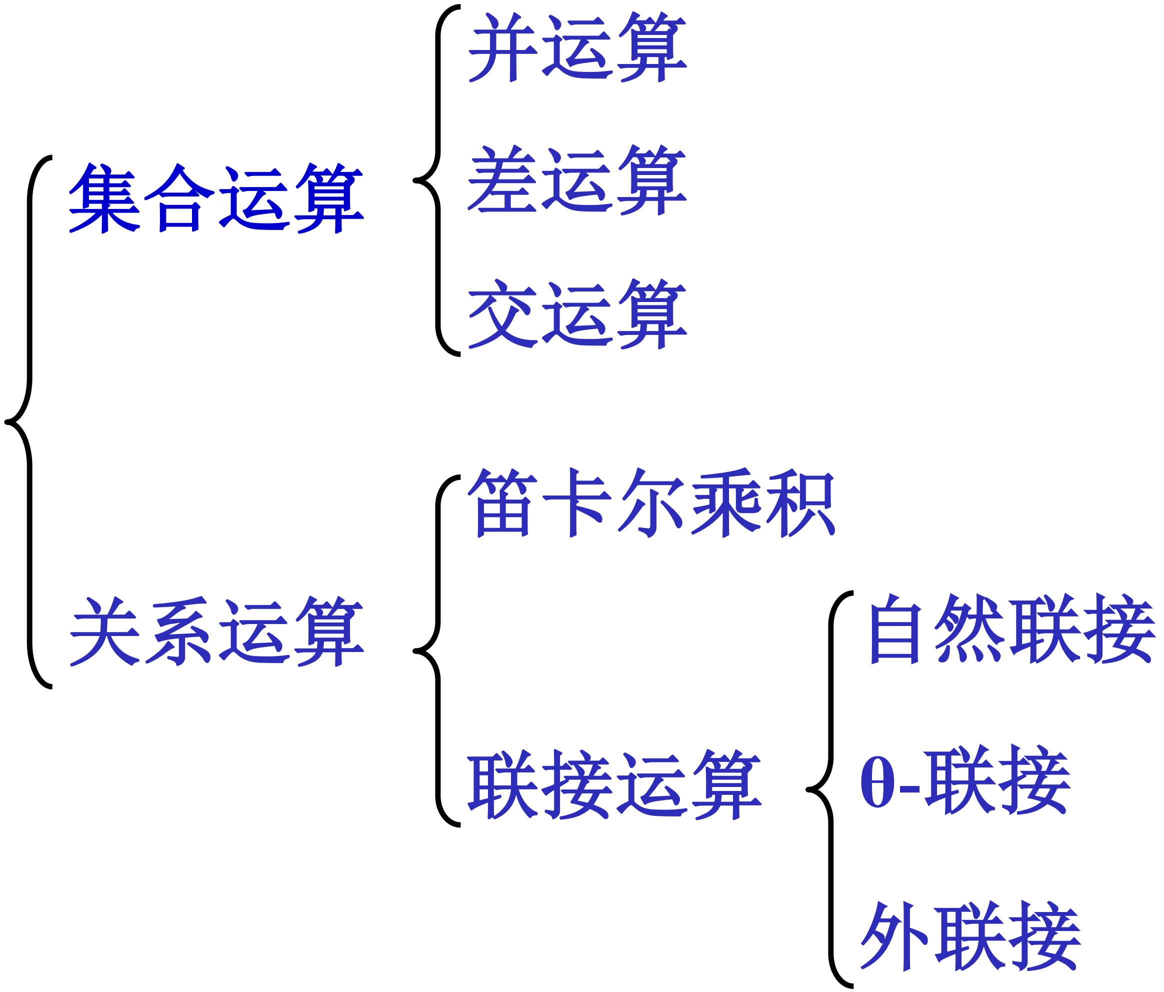
	$T = R \infty S$	$T = R \div S$
Head(R)	$\{A_1,\ldots,A_n,B_1,\ldots,B_k\}$	$\{A_1,\ldots,A_n,B_1,\ldots,B_m\}$
Head(S)	$\{B_1,\ldots,B_k,C_1,\ldots,C_m\}$	$\{ B_1,\ldots,B_m \}$
Head(T)	$\{A_1,\ldots,A_n,B_1,\ldots,B_k, C_1,C_2,\ldots,C_m\}$	$\{ A_1,A_2,\ldots,A_n \}$

Result of T=R∞S	A row t is in the table R∞S if and only if there are two rows u in R and v in S, such that $u[B_i] = v[B_i]$ for all i $(1 \le i \le k)$; then $t[A_i] = u[A_i]$ for all i, $1 \le i \le n$ $t[C_i] = v[C_i]$ for all i, $1 \le i \le m$ $t[B_i] = u[B_i] = v[B_i]$ for all i, $1 \le i \le k$
Result of T=R÷S	T contains *the largest possible set* of rows t such that for each row s in S, we can find one row r in R, where $t(A_i) = r(A_i)$ for $1 \le i \le n$ and $s(B_j) = r(B_j)$ for $1 \le j \le m$

ch2 重点、难点总结

- 1：关系的合并运算
- 2：特殊运算（‘差’运算）的使用方法
- 3：‘除’ 运算与‘联接’运算的区别
- 4：特殊运算（ ‘除’ 运算）的使用方法

总结1：关系的合并运算



- 集合运算
  - 并运算
  - 差运算
  - 交运算
- 关系运算
  - 笛卡尔乘积
  - 联接运算
    - 自然联接
    - θ-联接
    - 外联接

- 正确的表示方法！

总结2：特殊运算（‘差’运算）的使用方法

- ‘差’ 运算
  - 当查询条件带有‘否定’语义，或者具有明显的‘排它性’的时候，通常需要使用两个子查询之间的‘差’运算
  - ‘差’运算的运算对象（关系）中，通常需要包含其关键字

总结3：‘除’运算与‘联接’运算的区别

- ‘除’运算与‘联接’运算的区别
  - 我们将查询的结果关系称为‘目标对象’，用于定义查询条件的关系称为‘条件对象’
  - 在决定某个元组t**是否属于结果关系时，**
    - **如果只需要从条件对象中找到一个元组c并使得查询条件成立，那么就直接使用‘联接’运算**
    - **如果需要条件对象中的所有元组都能使得查询条件成立，那么就使用‘除’运算**

$T=R\infty S$

A row t is in the table $R\infty S$

if and only if

there are two rows u in R and v in S, such that $u[B_i] = v[B_i]$ for all i $(1\le i\le k)$;

then

$t[A_i] = u[A_i]$ for all i, $1\le i\le n$

$t[C_i] = v[C_i]$ for all i, $1\le i\le m$

$t[B_i] = u[B_i] = v[B_i]$ for all i, $1\le i\le k$

$T = R \div S$

T contains ***the largest possible set*** of rows t such that

- for each row s in S, we can find one row r in R, where
  - $t(A_i) = r(A_i)$ for $1 \le i \le n$
  - and
  - $s(B_j) = r(B_j)$ for $1 \le j \le m$

总结4：特殊运算（‘除’ 运算）的使用方法

- ‘除’ 运算表达式的表示方法
  - ‘被除数’ 关系中必须包含‘**目标对象**’**和**‘**条件对象**’**的关键字，**‘**除数**’**关系中只含**‘**条件对象**’**的关键字**
  - ‘**被除数**’**和**‘**除数**’**关系中不能含其它**‘**不必要**’**的多余属性**

Chapter 3

Basic SQL Query Language

Ch3 Basic SQL Query Language

3.1 Introduction

3.2 Setting Up the Database

3.3 Simple Select Statements

3.4 Subqueries

3.5 UNION Operators and FOR ALL Conditions

3.6 Some Advanced SQL Syntax

3.7 Set Functions in SQL

3.8 Groups of Rows in SQL

3.9 A Complete Description of SQL Select

3.10 Insert, Update, and Delete Statements

Basic SQL

- Data Type in SQL
- SQL statement
  - CREATE TABLE
  - Simple Select Statement
    - Relational Algebra vs SQL Query Statement
      - single table, product, natural join, theta join
    - table & column alias

Simple SELECT statement

- SQL查询语句(Select statement)的格式
  - 与关系代数查询表达式的关系
  - 单表查询，多表联接查询
- DISTINCT
- 表与列的换名
- 查询谓词：
  - BETWEEN … AND …
  - IS NULL
  - LIKE
- 结果排序：ORDER BY

Simple Select Statement

- predicate
  - distinct
  - The IN Predicate
    - expr [NOT] IN (subquery)
  - The Quantified Comparison Predicate
    - expr θ SOME|ANY|ALL (subquery)
  - The EXISTS Predicate
    - [NOT] EXISTS (subquery)
  - The BETWEEN Predicate
    - expr [NOT] BETWEEN expr1 AND expr2

Simple Select Statement

- predicate (cont.)
  - The IS NULL Predicate
    - column IS [NOT] NULL
  - The LIKE Predicate
    - column [NOT] LIKE val1 [ESCAPE val2]
      - underscore (_): any single character
      - percent (%): any sequence of zero or more characters

- 子查询(Subquery)
  - 表名的作用域
  - 独立子查询 / 相关子查询
  - 查询谓词：
    - IN,   NOT IN
    - SOME / ANY / ALL
    - EXISTS,   NOT EXISTS
- 复杂查询
  - 自连接：相同表之间的连接查询
  - 关系代数中的除法在SQL中的表示

Subqueries

Relational Algebra	SQL Predicate
natural join	IN
	= SOME
	EXISTS
difference	NOT IN
	<> ALL
	NOT EXISTS

- 统计查询
  - 统计查询
  - 分组统计查询 (GROUP BY)
  - 分组统计选择查询
    (GROUP BY … HAVING …)
- 数据更新语句
  - INSERT, DELETE, UPDATE