



2024年春季学期

数据库系统概论

An Introduction to Database Systems

第二章 关系数据库

中国科学技术大学 大数据学院

黄振亚, huangzhy@ustc.edu.cn



关系数据库简介

2

- 1962年，CODASYL发表“信息代数”
- 1968年，IBM7090机实现集合论数据结构
- 提出关系模型的是美国**IBM**公司的**E.F.Codd**

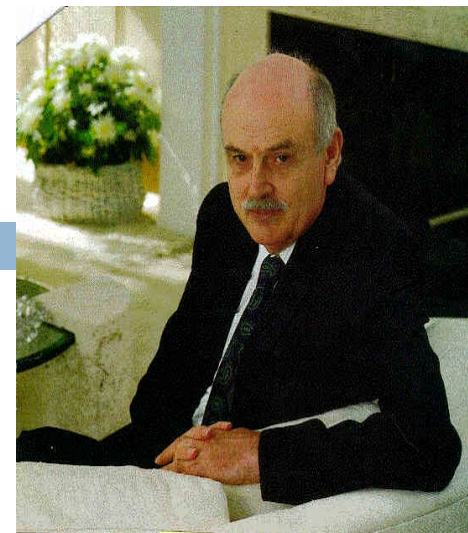
- 1970年，提出关系数据模型

E.F.Codd, “A Relational Model of Data for Large

Shared Data Banks”, 《Communication of the ACM》, 1970

✓ 1958年以来的四分之一世纪具有里程碑意义的25篇研究论文之一

- 之后，提出了关系代数和关系演算的概念
- 1972年，提出了关系的第一、第二、第三范式
- 1974年，提出了关系的**BC**范式





第二章 关系数据库

3

2.1 关系数据结构及形式化定义

2.2 关系操作

2.3 关系的完整性

2.4 关系代数

2.5 关系演算

2.6 小结



2.1 关系数据结构及形式化定义

4

- 2.1.1 关系
- 2.1.2 关系模式
- 2.1.3 关系数据库
- 2.1.4 关系模型的存储结构



2.1.1 关系

5

□ 单一的数据结构----关系

现实世界的**实体**以及**实体间的各种联系**均用关系来表示

□ 逻辑结构----二维表

从用户角度，关系模型中数据的**逻辑结构**是一张二维表

□ 数学基础：建立在**集合代数**的基础上



关系 (续)

6

1. 域 (Domain)
2. 笛卡尔积 (Cartesian Product)
3. 关系 (Relation)



1. 域 (Domain)

7

- **域**是一组具有相同数据类型的值的集合。例：
 - 整数
 - 实数
 - 介于某个取值范围的整数： (0,100)
 - 指定长度的字符串集合： 长度为30
 - { ‘男’ , ‘女’ }
 -



2. 笛卡尔积 (Cartesian Product)

8

□ 笛卡尔积 — 域上的一种集合运算

给定一组域 D_1, D_2, \dots, D_n , 允许某些域是相同的

D_1, D_2, \dots, D_n 的笛卡尔积为:

$$D_1 \times D_2 \times \dots \times D_n = \{(d_1, d_2, \dots, d_n) \mid d_i \in D_i, i=1, 2, \dots, n\}$$

□ 所有域的所有取值的一个组合

□ 不能重复

■ $D_1 =$ 导师集合 **SUPERVISOR** = {张清玫, 刘逸}

■ $D_2 =$ 专业集合 **SPECIALITY** = {计算机专业, 信息专业}

■ $D_3 =$ 研究生集合 **POSTGRADUATE** = {李勇, 刘晨, 王敏}



2. 笛卡尔积 (Cartesian Product)

9

- 笛卡尔积 —— 一个例子
 - 给出3个域
 - $D_1 =$ 导师集合SUPERVISOR = {张清玫, 刘逸}
 - $D_2 =$ 专业集合SPECIALITY = {计算机专业, 信息专业}
 - $D_3 =$ 研究生集合POSTGRADUATE = {李勇, 刘晨, 王敏}
 - D_1, D_2, D_3 的笛卡尔积 $D_1 \times D_2 \times D_3$ 为



表2.1 D_1, D_2, D_3 的笛卡尔积

SUPERVISOR	SPECIALITY	POSTGRADUATE
张清玫	计算机专业	李勇
张清玫	计算机专业	刘晨
张清玫	计算机专业	王敏
张清玫	信息专业	李勇
张清玫	信息专业	刘晨
张清玫	信息专业	王敏
刘逸	计算机专业	李勇
刘逸	计算机专业	刘晨
刘逸	计算机专业	王敏
刘逸	信息专业	李勇
刘逸	信息专业	刘晨
刘逸	信息专业	王敏



笛卡尔积 (续)

11

$$D_1 \times D_2 \times \dots \times D_n = \{(d_1, d_2, \dots, d_n) \mid d_i \in D_i, i=1, 2, \dots, n\}$$

□ 元组 (Tuple)

- 笛卡尔积中每一个元素 (d_1, d_2, \dots, d_n) 叫作一个 n 元组 (n-tuple) 或简称**元组(Tuple)**
- 例如: (张清玫, 计算机专业, 李勇)
(张清玫, 计算机专业, 刘晨) 等都是元组

□ 分量 (Component)

- 笛卡尔积元素 (d_1, d_2, \dots, d_n) 中的每一个值 d_i 叫作一个**分量**
- 张清玫、计算机专业、李勇、刘晨 等都是分量



表2.1 D_1, D_2, D_3 的笛卡尔积

SUPERVISOR	SPECIALITY	POSTGRADUATE
张清玫	计算机专业	李勇
张清玫	计算机专业	刘晨
张清玫	计算机专业	王敏
张清玫	信息专业	李勇
张清玫	信息专业	刘晨
张清玫	信息专业	王敏
刘逸	计算机专业	李勇
刘逸	计算机专业	刘晨
刘逸	计算机专业	王敏
刘逸	信息专业	李勇
刘逸	信息专业	刘晨
刘逸	信息专业	王敏

分量

元组



笛卡尔积 (续)

13

□ 基数 (Cardinal number)

- 一个域允许的不同取值个数称为这个域的基数
- 若 $D_i (i=1, 2, \dots, n)$ 为有限集, 其基数为 $m_i (i=1, 2, \dots, n)$, 则笛卡尔积 $D_1 \times D_2 \times \dots \times D_n$ 的基数 M 为:

$$M = \prod_{i=1}^n m_i$$

- 例如, 上述例子中, 基数 $M=2*2*3=12$ 。
 - $D_1 =$ 导师集合SUPERVISOR = {张清玫, 刘逸}
 - $D_2 =$ 专业集合SPECIALITY = {计算机专业, 信息专业}
 - $D_3 =$ 研究生集合POSTGRADUATE = {李勇, 刘晨, 王敏}



笛卡尔积 (续)

14

- 笛卡尔积的表示方法
 - 笛卡尔积可表示为一个二维表
 - 表中的每行对应一个元组，表中的每列对应一个域



笛卡尔积 (续)

表2.1 D_1, D_2, D_3 的笛卡尔积

SUPERVISOR	SPECIALITY	POSTGRADUATE
张清玫	计算机专业	李勇
张清玫	计算机专业	刘晨
张清玫	计算机专业	王敏
张清玫	信息专业	李勇
张清玫	信息专业	刘晨
张清玫	信息专业	王敏
刘逸	计算机专业	李勇
刘逸	计算机专业	刘晨
刘逸	计算机专业	王敏
刘逸	信息专业	李勇
刘逸	信息专业	刘晨
刘逸	信息专业	王敏

分量

元组

$M1 = 2$
 $M2 = 2$
 $M3 = 3$
 $M = 12$



3. 关系 (Relation)

16

□ 关系

$D_1 \times D_2 \times \dots \times D_n$ 的子集叫作在域 D_1, D_2, \dots, D_n 上的关系，表示为

$R(D_1, D_2, \dots, D_n)$

- R : 关系名
- n : 关系的目或度 (Degree)



3. 关系 (Relation)

17

□ 关系

$D_1 \times D_2 \times \dots \times D_n$ 的子集叫作在域 D_1, D_2, \dots, D_n 上的关系，表示为

$$R(D_1, D_2, \dots, D_n)$$

■ R : 关系名

■ n : 关系的目或度 (Degree)

□ 元组: 关系中的每个元素是关系中的元组，通常用 t 表示

□ 单元关系与二元关系

■ 当 $n=1$ 时，称该关系为单元关系 (Unary relation)
或一元关系

■ 当 $n=2$ 时，称该关系为二元关系 (Binary relation)



关系 (Relation)

关系的表示

- ✓ 关系是笛卡尔积的有限子集
- ✓ 关系也是一个二维表，表的每行对应一个元组，表的每列对应一个域

表 2.2 SAP 关系

SUPERVISOR	SPECIALITY	POSTGRADUATE
张清玫	信息专业	李勇
张清玫	信息专业	刘晨
刘逸	信息专业	王敏



关系 (Relation)

19

□ 属性

- 关系中不同列可以对应相同的域
- 为了加以区分，必须对每列起一个名字，称为属性 (Attribute)
- n 目关系必有 n 个属性



关系 (Relation)

20

□ 码

□ 候选码 (Candidate key)

若关系中的某一属性组的值能唯一地标识一个元组，则称该属性组为候选码

简单的情况：候选码只包含一个属性

□ 全码 (All-key)

最极端的情况：关系模式的所有属性组是这个关系模式的候选码，称为全码 (All-key)



关系 (Relation)

21

□ 码

□ 主码 系统建立index

- 若一个关系有多个候选码，则选定其中一个为主码 (Primary key)

□ 主属性

- 候选码的诸属性称为主属性 (Prime attribute)
- 不包含在任何侯选码中的属性称为非主属性 (Non-Prime attribute) 或非码属性 (Non-key attribute)



关系 (Relation)

□ D_1, D_2, \dots, D_n 的笛卡尔积的某个真子集才有实际含义

例：表2.1 的笛卡尔积没有实际意义

取出有实际意义的元组来构造关系

关系：SAP(SUPERVISOR, SPECIALITY, POSTGRADUATE)

假设：导师与专业：1:1， 导师与研究生：1:n

主码：POSTGRADUATE（假设研究生不会重名）

表2.2 SAP关系

SAP关系可以包含三个元组

- { (张清玫, 计算机专业, 李勇),
- (张清玫, 计算机专业, 刘晨),
- (刘逸, 信息专业, 王敏) }

SUPERVISOR	SPECIALITY	POSTGRADUATE
张清玫	计算机专业	李勇
张清玫	计算机专业	刘晨
刘逸	信息专业	王敏



关系 (Relation)

23

□ 三类关系

□ 基本关系 (基本表或基表)

实际存在的表，是实际存储数据的逻辑表示

□ 查询表

查询结果对应的表

□ 视图表 (View)

✓ 由基本表或其他视图表导出的表，是虚表，不对应实际存储的数据

✓ 实际不存在



关系 (Relation)

SUPERVISOR	SPECIALITY	POSTGRADUATE
张清玫	计算机专业	李勇
张清玫	计算机专业	刘晨
刘逸	信息专业	王敏

24

基本关系的性质

- ① 列是同质的 (同一个类型数据)
- ② 不同的列可出自同一个域
 - 其中的每一列称为一个属性
 - 不同的属性要给予不同的属性名
- ③ 列的顺序无所谓, 列的次序可以任意交换
- ④ 任意两个元组的候选码值不能相同 (增加id)
- ⑤ 行的顺序无所谓, 行的次序可以任意交换



基本关系的性质(续)

⑥ 分量必须取原子值（不可分）
这是规范条件中**最基本**的一条

表2.3 非规范化关系

SUPERVISOR	SPECIALITY	POSTGRADUATE	
		PG1	PG2
张清玫	信息专业	李勇	刘晨
刘逸	信息专业	王敏	

小表

关系的每一个分量必须是一个不可分的数据项



2.1 关系数据结构

2.1.1 关系

2.1.2 关系模式

2.1.3 关系数据库

2.1.4 关系模型的存储结构



2.1.2 关系模式

27

1. 什么是关系模式
2. 定义关系模式
3. 关系模式与关系



1. 什么是关系模式

28

关系数据库中

- 关系模式（Relation Schema）是**型**
- 关系是**值**
- 关系模式是对关系的描述
 - 元组集合的结构
 - 属性构成
 - 属性来自的域
 - 属性与域之间的映象关系
 - 完整性约束条件
 - 属性间数据的依赖关系集合
 - 如，取值范围，相互关联



2. 定义关系模式

29

定义2.4 关系模式可以形式化地表示为:

$R (U, D, DOM, F)$

R 关系名

U 组成该关系的属性名集合

D 属性组 U 中属性所来自的域

DOM 属性向域的映象集合

F 属性间的数据依赖关系集合

第6章 关系数据理论



定义关系模式 (

SUPERVISOR	SPECIALITY	POSTGRADUATE
张清玫	计算机专业	李勇
张清玫	计算机专业	刘晨
刘逸	信息专业	王敏

例: SAP模式:

SAP(SUPERVISOR, SPECIALITY, POSTGRADUATE)

□ 定义域

□ D1: PERSON。 D2: SPECIALITY

□ 导师和研究生出自同一个域——人

□ 定义DOM

□ 取不同的属性名，并在模式中定义属性向域的映象，即说明它们分别出自哪个域:

DOM (SUPERVISOR)

= DOM (POSTGRADUATE)

= PERSON



定义关系模式 (续)

关系模式通常可以简记为

$R(U)$ 或 $R(A_1, A_2, \dots, A_n)$

- R : 关系名
- A_1, A_2, \dots, A_n : 属性名

注: 域名及属性向域的映象常常直接说明为属性的类型、长度

```
CREATE TABLE Student 关系R  
  (Sno CHAR(9) PRIMARY KEY,  
   Sname CHAR(20) UNIQUE,  
   Ssex CHAR(2)  
   Sage INT)  
  
  属性名      DOM
```



3. 关系模式与关系

32

- 关系模式
 - 对关系的描述
 - 静态的、稳定的
- 关系
 - 关系模式在**某一时刻**的状态或内容
 - 动态的、随时间不断变化的
- 注：关系模式和关系往往统称为关系

通过上下文加以区别



2.1 关系数据结构

33

2.1.1 关系

2.1.2 关系模式

2.1.3 关系数据库

2.1.4 关系模型的存储结构



2.1.3 关系数据库

34

- 关系数据库
 - 在一个给定的应用领域中，所有关系的集合构成一个关系数据库
- 关系数据库的型与值



2.1.3 关系数据库

35

- 关系数据库的型: 关系数据库模式
对关系数据库的描述。
- 关系数据库模式包括
 - 若干域的定义
 - 在这些域上定义的若干关系模式
- 关系数据库的值: 关系模式在某一时刻对应的关系的集合, 简称为关系数据库



2.1.3 关系数据库（示例）

- 学生表: Student (Sno, Sname, Ssex, Sage, Sdept)
- 课程表: Course (Cno, Cname, Ccredit)
- 学生选课表: SC (Sno, Cno, Grade)

Sno	Sname	Ssex	Sage	Sdept
001	李勇	男	20	CS
002	刘晨	女	19	CS

Cno	Cname	Cpno	Ccredit
1	数据库	3	3.5
2	数学		4
3	数据结构		4

Sno	Cno	Grade	Sno	Cno	Grade
001	1	92	002	2	90
001	2	85	002	3	80
001	3	88			



2.1 关系数据结构

37

2.1.1 关系

2.1.2 关系模式

2.1.3 关系数据库

2.1.4 关系模型的存储结构



2.1.4 关系数据库的存储结构

38

- 关系数据库的逻辑结构表示：“表”
- 关系数据库的**物理结构**
 - 一个表对应一个操作系统文件，物理结构组织交给操作系统完成；
 - 从操作系统那里申请若干个大的文件，自己划分文件空间，组织表、索引等结构，并进行存储管理



第二章 关系数据库

39

2.1 关系模型概述

2.2 关系操作

2.3 关系的完整性

2.4 关系代数

2.5 关系演算

2.6 小结



2.2.1 基本关系操作

40

- 常用的关系操作
 - 查询：选择、投影、连接、除、并、交、差
 - 数据更新：插入、删除、修改
- 查询的表达能力是其中最主要的部分
 - 5种基本操作：选择、投影、并、差、笛卡尔积
-



2.2.2 关系数据库语言的分类

41

- 关系代数语言 $\pi_{Sno} (\sigma_{Cno='2'} (SC))$
 - 用对关系的运算来表达查询要求。代表：ISBL
- 关系演算语言：用谓词来表达查询要求
 - 元组关系演算语言
 - 谓词变元的基本对象是元组变量。代表：APLHA, QUEL
 - 域关系演算语言
 - 谓词变元的基本对象是域变量。代表：QBE
- 具有关系代数和关系演算双重特点的语言
 - 代表：SQL (Structured Query Language)
- 关系数据语言 (SQL)
 - 查询、数据定义语言、数据操纵语言、数据控制语言



2.2.2 关系数据库语言

42

- 关系数据库语言的特点
 - 集合操作方式
 - 操作的对象和结果都是集合，一次一集合的方式
 - 语言具有完备的表达能力
 - 非过程化的操作语言，是描述性语言
 - “要什么”，而非“怎么要”
 - 不必请求建立特殊存取路径，存取路径的选择由DBMS完成
 - 例如，顺序查询，or 索引查询
 - 系统自动查询优化



第二章 关系数据库

43

2.1 关系数据结构及形式化定义

2.2 关系操作

2.3 关系的完整性

2.4 关系代数

*2.5 关系演算

2.6 小结



2.3 关系的完整性

44

2.3.1 关系的三类完整性约束

2.3.2 实体完整性

2.3.3 参照完整性

2.3.4 用户定义的完整性



2.3.1 关系的三类完整性约束

45

- 现实世界的要求
- 实体完整性和参照完整性
 - 关系模型**必须满足**的完整性约束条件
 - 称为关系的两个**不变性**，应该由关系系统自动支持
- 用户定义的完整性
 - 应用领域需要遵循的约束条件，体现了具体领域中的语义约束（只需定义）
- **任何关系在任何时刻**都满足这些语义约束



2.3 关系的完整性

46

2.3.1 关系的三类完整性约束

2.3.2 实体完整性

2.3.3 参照完整性

2.3.4 用户定义的完整性



2.3.2 实体完整性

47

规则2.1 实体完整性规则 (Entity Integrity)

- 若属性A 是基本关系R 的主属性，则属性A不能取空值NULL
 - 例：SAP (SUPERVISOR, SPECIALITY, POSTGRADUATE)
POSTGRADUATE: 主码 (假设研究生不会重名)
不能取空值
 - 例：选修 (Sno, Cno, Grade)
Sno和Cno: 两个属性均不能取空值
- 空值就是“不知道”或“不存在”或“无意义”的值。非0
 - 例如，Grade与空值的关系



实体完整性

48

实体完整性规则的说明

- (1) 实体完整性规则是针对**基本关系**而言的。一个基本表通常对应现实世界的一个实体集。
- (2) 现实世界中的**实体是可区分的**，即它们具有某种唯一性标识。
- (3) 关系模型中以主码作为**唯一性标识**。
- (4) **主码中的属性，即主属性**不能取空值。

主属性取空值，就说明存在某个不可标识的实体，即存在不可区分的实体，这与第（2）点相矛盾，

这个规则称为**实体完整性**