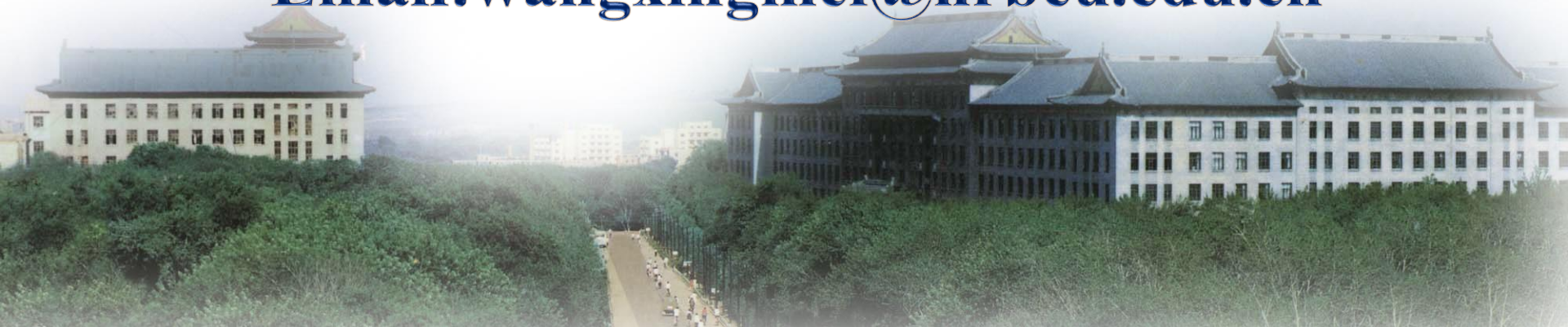
数据库原理

王兴梅
计算机科学与技术学院
Email:wangxingmei@hrbeu.edu.cn



第二章 关系数据库

2.1 关系模型概述

2.2 关系数据结构

2.3 关系操作

2.4 关系的完整性

2.5 关系代数

2.6 小结

关系数据库简介

- 系统而严格地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
  - 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

2.1 关系模型概述

2.2 关系数据结构

2.3 关系操作

2.4 关系的完整性

2.5 关系代数

2.6 小结

2.1 关系模型概述

- 关系数据库系统
  - 是支持关系模型的数据库系统
- 关系模型的组成
  - 关系数据结构
  - 关系操作
  - 关系的完整性

关系模型建立在集合代数的基础上

第二章 关系数据库

2.1 关系模型概述

2.2 关系数据结构

2.3 关系操作

2.4 关系的完整性

2.5 关系代数

2.6 小结

2.2 关系数据结构

- 关系数据结构的基本概念
  - 关系
  - 关系模式
  - 关系数据库

2.2 关系数据结构

- 2.2.1 关系
- 2.2.2 关系模式
- 2.2.3 关系数据库

2.2 关系数据结构

- 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
- 数据的逻辑结构----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。

2.2.1 关系

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

1. 域（Domain）

- **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
  - 指定长度的字符串集合
  - {‘男’，‘女’}
  - 介于某个取值范围的日期

2. 笛卡尔积（Cartesian Product）

- 1) 笛卡尔积

  给定一组域D_1，D_2，…，D_n，这些域中可以有相同的。D_1，D_2，…，D_n的笛卡尔积为：

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

  - 所有域的所有取值的一个组合
  - 不能重复

笛卡尔积

- 2) 元组（Tuple）
  - 笛卡尔积中每一个元素（d_1，d_2，…，d_n）叫作一个*n*元组（n-tuple）或简称元组。

- 3) 分量（Component）
  - 笛卡尔积元素（d_1，d_2，…，d_n）中的每一个值d_i叫作一个分量。

笛卡尔积

- 4) 基数（Cardinal number）
  - 若D_i（$i=1，2，…，n$）为有限集，其基数为m_i（$i=1，2，…，n$），则$D_1\times D_2\times…\times D_n$的基数$M$为：

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

举例

D1=导师集合SUPERVISOR={张清政，刘逸}
D2=专业集合SPECIALITY={计算机专业，信息专业}
D3=研究生集合POSTGRADUATE={李勇，刘晨，王敏}

D1、 D2、 D3的笛卡尔积的基数为：2×2×3＝12，即D1×D2×D3共有2×2×3 ＝12个元组

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

笛卡尔积

- 5)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。

在上例中，12个元组可列成一张二维表

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

3. 关系（Relation）

1) 关系

$D_1 \times D_2 \times \ldots \times D_n$的子集叫作在域$D_1$，$D_2$，…，$D_n$上的**关系**，表示为

$$R（D_1，D_2，\ldots，D_n）$$

R：关系名

n：关系的目或度（Degree）

关系

2) 元组

关系中的每个元素是关系中的元组，通常用*t*表示。

3) 单元关系与二元关系

当*n*=1时，称该关系为**单元**关系（Unary relation）。

当*n*=2时，称该关系为**二元**关系（Binary relation）。

关系

4) 关系的表示

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

5) 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。

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

6) 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。

在最简单的情况下，候选码只包含一个属性。

全码（All-key）

在最极端的情况下，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

主码

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

候选码的诸属性称为主属性（Prime attribute）

不包含在任何候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

7) 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

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

关系

8) 基本关系的性质

- 列是同质的（Homogeneous）
- 不同的列可出自同一个域
- 列的顺序无所谓
- 任意两个元组的候选码不能取相同的值
- 行的顺序无所谓
- 分量必须取原子值

2.2 关系数据结构

2.2.1 关系

2.2.2 关系模式

2.2.3 关系数据库

2.2.2 关系模式

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

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
- 元组语义以及完整性约束条件
- 属性间的数据依赖关系集合

2. 定义关系模式

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

定义关系模式 (续)

关系模式通常可以简记为

$R(U)$ 或 $R(A_1, A_2, \ldots, A_n)$

R 关系名

$A_1, A_2, \ldots, A_n$ 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

3. 关系模式与关系

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

关系模式和关系往往统称为关系，通过上下文加以区别

2.2 关系数据结构

2.2.1 关系

2.2.2 关系模式

2.2.3 关系数据库

2.2.3 关系数据库

1. 关系数据库

在一个给定的应用领域中，所有实体及实体之间联系的关系的集合构成一个关系数据库。

2. 关系数据库的型与值

关系数据库的型：也称关系数据库模式，对关系数据库的描述。

关系数据库的值：关系模式在某一个时刻对应关系的集合，简称关系数据库。

第二章 关系数据库

2.1 关系模型概述

2.2 关系数据结构

2.3 关系操作

2.4 关系的完整性

2.5 关系代数

2.6 小结

2.3 关系操作

- 1) 常用的关系操作
- 2) 关系操作的特点
- 3) 关系数据语言的种类
- 4) 关系数据语言的特点

关系操作

- 1) 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

关系操作

- 2) 关系操作的特点
  - 集合操作方式，即操作的对象和结果都是集合。一次一集合的方式。
    - 非关系数据模型的数据操作方式：一次一记录。

关系操作

- 关系数据语言的种类
  - 关系代数语言：用对关系的运算来表达查询要求(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---典型代表：APLHA, QUEL
    - 域关系演算语言

      谓词变元的基本对象是域变量---典型代表：QBE
  - 具有关系代数和关系演算双重特点的语言
    - 典型代表：SQL(Structured Query Language)

关系操作

- 3) 关系数据语言的特点
  - 关系语言是一种高度非过程化的语言
    - 存取路径的选择由DBMS的优化机制来完成
    - 用户不必用循环结构就可以完成数据操作
  - 能够嵌入高级语言中使用
  - 关系代数、元组关系演算和域关系演算三种语言在表达能力上完全等价

第二章 关系数据库

2.1 关系模型概述
2.2 关系数据结构
2.3 关系操作
2.4 关系的完整性
2.5 关系代数
2.6 小结

2.4 关系的完整性

关系模型的完整性规则是对关系的某种约束条件。

关系模型中三类完整性约束：

实体完整性

参照完整性

用户定义的完整性

实体完整性和参照完整性是关系模型必须满足的完整性约束条件，被称作是关系的两个不变性，应该由关系系统自动支持。

2.4 关系的完整性

- 实体完整性
  - 通常由关系系统自动支持
- 参照完整性
  - 早期系统不支持，目前大型系统能自动支持
- 用户定义的完整性
  - 反映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
  - 用户定义后由系统支持

2.4.1 实体完整性

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

例

学生(学号，姓名，性别)

‘学号’属性为主码,则其不能取空值

2.4.2 参照完整性

1. 关系间的引用
2. 外码
3. 参照完整性规则

1. 关系间的引用

在关系模型中实体及实体间的联系都是用关系来描述的，因此可能存在着关系与关系间的引用。

例1 学生实体、专业实体以及专业与学生间的一对多联系

学生（学号，姓名，性别，专业号，年龄）

专业（专业号，专业名）

学生（学号，姓名，性别，专业号，年龄）

学号	姓名	性别	专业号	年龄
801	张三	女	01	19
802	李四	男	01	20
803	王五	男	01	20
804	赵六	女	02	20
805	钱七	男	02	19

专业（专业号，专业名）

专业号	专业名
01	信息
02	数学
03	计算机

2．外码（Foreign Key）

设*F*是基本关系*R*的一个或一组属性，但不是关系*R*的码。如果F与基本关系*S*的主码K_s相对应，则称F是基本关系*R*的外码，基本关系*R*称为参照关系（Referencing Relation）基本关系*S*称为被参照关系（Referenced Relation）或目标关系（Target Relation）。

例1 学生实体、专业实体以及专业与学生间的一对多联系

学生（学号，姓名，性别，专业号，年龄）

专业（专业号，专业名）

‘专业号’称为学生关系的外码

外码

说明

- 关系*R*和*S*不一定是不同的关系
- 目标关系*S*的主码K_s和参照关系的外码F必须定义在
- 外码并不一定要与相应的主码同名。当外码与相应的主码属于不同关系时，往往取相同的名字，以便于识别。

学生（学号，姓名，性别，所在班级，班长学号）

3. 参照完整性规则

若属性（或属性组）*F*是基本关系*R*的外码它与基本关系*S*的主码K_s相对应（基本关系*R*和*S*不一定是不同的关系），则对于*R*中每个元组在*F*上的值必须为：

- 或者取空值（*F*的每个属性值均为空值）
- 或者等于*S*中某个元组的主码值。

参照完整性规则

学生关系中每个元组的“专业号”属性只取下面两类值：

（1）空值，表示尚未给该学生分配专业

（2）非空值，这时该值必须是专业关系中某个元组的“专业号”值，表示该学生不可能分配到一个不存在的专业中

2.4.3 用户定义的完整性

- 用户定义的完整性是针对某一具体关系数据库的约束条件，反映某一具体应用所涉及的数据必须满足的语义要求。
- 关系模型应提供定义和检验这类完整性的机制，以便用统一的系统的方法处理它们，而不要由应用程序承担这一功能。

小结

- 关系数据结构
  - 关系
    - 域
    - 笛卡尔积
    - 关系
      - 关系，属性，元组
      - 候选码，主码，主属性
      - 基本关系的性质
  - 关系模式
  - 关系数据库

- 关系的数据操作
  - 查询
    - 选择、投影、连接、除、并、交、差
  - 数据更新
    - 插入、删除、修改

- 关系的完整性
  - 实体完整性
    - 主属性-（候选码）
  - 参照完整性
    - 外码
  - 用户定义的完整性