

# 第二章 关系数据库

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

# 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

# 关系数据库简介

- 关系数据库应用数学方法来处理数据库中的数据
- 80年代后，关系数据库系统成为最重要、最流行的数据库系统

# 第二章 关系数据库

## 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

- 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
  - 关系操作集合
  - 关系完整性约束

## 2.1 关系数据结构及形式化定义

- 2.1.1 关系
- 2.1.2 关系模式
- 2.1.3 关系数据库

# 关系（续）

## ● 码

### — 候选码

- 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**（Candidate key）。
- 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**非码属性**（Non-key attribute）。
- 在最简单的情况下，候选码只包含一个属性。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（续）

- 码(续)

- 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）。



# 关系（续）

## ● 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（续）

- 基本关系的性质

- ① 列是同质的（**Homogeneous**）

- 每一列中的分量是同一类型的数据，来自同一个域。

- 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

# 关系（续）

- 基本关系的性质（续）

- ③ 列的顺序无所谓

- 列的次序可以任意交换
    - 遵循这一性质的数据库产品（如**ORACLE**），增加新属性时，永远是插至最后一列
    - 但也有许多关系数据库产品没有遵循这一性质，例如**FoxPro**仍然区分了属性顺序

# 关系（续）

- 基本关系的性质（续）

- ④ 任意两个元组不能完全相同

- 由笛卡尔积的性质决定
- 但许多关系数据库产品没有遵循这一性质。例如 **Oracle**, **FoxPro**等都允许关系表中存在两个完全相同的元组，除非用户特别定义了相应的约束条件。

# 关系（续）

- 基本关系的性质（续）

- ⑤ 行的顺序无所谓

- 行的次序可以任意交换
    - 遵循这一性质的数据库产品（如**ORACLE**），插入一个元组时永远插至最后一行
    - 但也有许多关系数据库产品没有遵循这一性质，例如**FoxPro**仍然区分了元组的顺序

# 关系（续）

## ● 基本关系的性质（续）

### — ⑥ 分量必须取原子值

- 每一个分量都必须是不可分的数据项。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	

## 2.1 关系数据结构及形式化定义

- 2.1.1 关系
- 2.1.2 关系模式
- 2.1.3 关系数据库

## 2.1.2 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
  - 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：

$R(U, D, \text{dom}, F)$

$R$  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

$\text{dom}$  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  $R(A_1, A_2, \dots, A_n)$

$R$  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。

### 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。

## 2.1 关系数据结构及形式化定义

- 2.1.1 关系
- 2.1.2 关系模式
- 2.1.3 关系数据库

## 2.1.3 关系数据库

- 1. 关系数据库
- 2. 关系数据库的型与值

# 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

- 关系数据库也有型和值之分
- 关系数据库的型称为关系数据库模式，是对关系数据库的描述
  - 若干域的定义
  - 在这些域上定义的若干关系模式
- 关系数据库的值是这些关系模式在某一时刻对应的关系的集合，通常简称为关系数据库



# 第二章 关系数据库

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

## 2.2 关系操作

- 2.2.1 基本的关系操作
- 2.2.2 关系操作的特点
- 2.2.3 关系数据语言的分类
- 2.2.4 关系数据语言的特点

## 2.2.1 基本的关系操作

- 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

## 2.2.2 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。
  - 非关系数据模型的数据操作方式：一次一记录
  - 文件系统的数据操作方式

## 2.2.3 关系数据语言的分类

- 关系代数语言

- 用对关系的运算来表达查询要求
- 典型代表：ISBL

## 2.2.4 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

- 关系模型的完整性规则是对关系的某种约束条件。
- 任何关系在任何时刻都要满足这些语义约束。



## 2.3 关系的完整性

- 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 关系的完整性

- 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

- 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。
- 用户定义的完整性是应用领域需要遵循的约束条件，体现了具体领域的语义约束。

## 2.3 关系的完整性

- 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

- 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，年龄）

学号属性为主码，则其不能取空值。

## 2.3.2 实体完整性

- 关系模型必须遵守实体完整性规则的原因
  - (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或多对多联系。
  - (2) 现实世界中的实体和实体间的联系都是可区分的，即它们具有某种唯一性标识。
  - (3) 相应地，关系模型中以候选码作为唯一性标识。

## 2.3.2 实体完整性

- 关系模型必须遵守实体完整性规则的原因(续)
  - (4) 候选码中的属性即主属性不能取空值。所谓空值就是“不知道”或“无意义”的值。如果主属性取空值，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，因此这个规则称为实体完整性。

## 2.3.2 实体完整性

### ● 注意

- 实体完整性规则规定基本关系的所有主属性都不能取空值，而不仅是主码整体不能取空值。

例：

选修（学号，课程号，成绩）

“学号、课程号”为主码，

则学号和课程号两个属性都不能取空值



## 2.3 关系的完整性

- 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

## 2.3.3 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

# 关系间的引用(续)

例2 学生、课程、学生与课程之间的多对多联系

专业 (专业号, 专业名)

学生 (学号, 姓名, 性别, 专业号, 年龄)

课程 (课程号, 课程名, 学分)

选修 (学号, 课程号, 成绩)

## 学生

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

## 课程

课程号	课程名	学分
01	数据库	4
02	数据结构	4
03	编译	4
04	PASCAL	2

## 学生选课

学号	课程号	成绩
801	04	92
801	03	78
801	02	85
802	03	82
802	04	90
803	04	88

# 关系间的引用(续)

例3 学生实体及其内部的领导联系(一对多)

学生 (学号, 姓名, 性别, 专业号, 年龄, 班长)

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

## 2. 外码

### ● 外码

- 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**（Foreign Key）
- 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）。



# 外码(续)

## ● 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 $F$ 必须定义在同一个（或一组）域上。
- 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。

### 3. 参照完整性规则

- 参照完整性规则就是定义外码与主码之间的引用规则。
- 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(续)

## ● 例

### 例1

学生关系中每个元组的“专业号”属性只能取下面两类值：

- (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。

# 参照完整性规则(续)

## 例2

由于“学号”和“课程号”是选修关系中的主属性，按照实体完整性和参照完整性规则，它们只能取相应被参照关系中已经存在的主码值。

# 参照完整性规则(续)

## 例3

“班长”属性值可以取两类值：

- (1) 空值，表示该学生所在班级尚未选出班长，或该学生本人即是班长；
- (2) 非空值，这时该值必须是本关系中某个元组的学号值。

## 2.3 关系的完整性

- 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

## 2.3.3 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。

# 用户定义的完整性(续)

例

课程(课程号, 课程名, 学分)

- “课程名” 属性必须取唯一值
- 非主属性 “课程名” 也不能取空值
- “学分” 属性只能取值{1, 2, 3, 4}



# 第二章 关系数据库

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

# 第二章 关系数据库

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

## 2.6 小结

- 关系数据库系统是目前使用最广泛的数据库系统。
- 关系系统只有“表”这一种数据结构。
- 本章系统讲解了关系数据库的重要概念，包括关系模型的数据结构、关系的三类完整性以及关系操作。介绍了关系语言（关系代数）。