



知识点**18**: 关系数据库

关系数据库简介



- 关系数据库是采用**关系模型**作为数据组织方式的数据库。
- 目前广泛使用的数据库软件基本上都是基于关系模型的关系数据库管理系统，例如SQL Server、Oracle、Sybase、Informix、DB2等，都是典型的关系数据库管理系统。

关系



- 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
- 建立在集合代数的基础上

关系模型的基本概念



■ 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

1. 在关系模型中，数据是以二维表的形式存在的，这个二维表就叫做**关系**
2. 关系理论是以**集合代数理论**为基础的，因此，我们可以用集合代数给出二维表的“关系”定义

关系模型的基本概念(续)



■ 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

1. 关系代数

用代数的方法表示关系模型

2. 关系演算

用逻辑方法表示关系模型

3. SQL

具有关系代数和关系演算双重特点的语言

关系模型的基本概念（续）



■ 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

1. 实体完整性

2. 参照完整性

3. 用户定义的完整性



知识点**19**: 关系的性质

关系的性质



- 尽管**关系**与**二维表格**、传统的**数据文件**是非常类似的，但它们之间又有重要的区别。
- 严格地说，**关系**是种规范化了的**二维表中行的集合**，关系具有如下特性：
 - 1. **关系中不允许出现相同的元组。**
 - 因为数学上集合中没有相同的元素，而关系是元组的集合，所以作为集合元素的元组应该是唯一的。

关系的性质（续）



- 2. 关系中元组的顺序（即行序）是无关紧要的，在一个关系中 can 任意交换两行的次序。
 - 因为集合中的元素是无序的，所以作为集合元素的元组也是无序的。根据关系的这个性质，可以改变元组的顺序使其具有某种排序，然后按照顺序查询数据，可以提高查询速度。
- 3. 关系中属性的顺序是无关紧要的，即列的顺序可以任意交换。
 - 交换时，应连同属性名一起交换，否则将得到不同的关系。

关系的性质（续）



- 4. 同一属性名下的各个属性值必须来自同一个域，是同一类型的数据。
- 5. 关系中各个属性必须有不同的名字，不同的属性可来自同一个域，即它们的分量可以取自同一个域。

关系的性质（续）



- 6. 关系中每一分量必须是不可分的数据项。
 - 或者说所有属性值都是原子的，即是一个确定的值，而不是值的集合，即不可“表中有表”。满足此条件的关系称为**规范化关系**，否则称为**非规范化关系**。
 - 属性值可以为空值，表示“未知”或“不可使用”

姓名	籍贯	
	省	市/县
张强	吉林	长春
王丽	山西	大同



姓名	省	市/县
张强	吉林	长春
王丽	山西	大同



知识点**20**: 关系的键

关系的键（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
 - “学生关系”中的学号能唯一标识每一个学生，则属性学号是学生关系的候选键。
 - 在“选课关系”中，只有属性的组合“学号+课程号”才能唯一地区分每一条选课记录，则属性集“学号+课程号”是选课关系的候选键。

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

主键



- 如果一个关系中有多个候选键，可以从中选择一个作为查询、插入或删除元组的操作变量，被选用的候选键称为**主关系键(Primary Key)**，或简称为**主键、主码、关系键、关键字**。
 - 例如，假设在学生关系中没有重名的学生，则“学号”和“姓名”都可作为学生关系的候选键。如果选定“学号”作为数据操作的依据，则“学号”为主关系键。
- 在关系模式中表示主键
 - 学生（学号，姓名，性别，年龄，系别）

■ 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)
- 假设有教师授课关系TCS，分别有三个属性教师 (T)、课程 (C) 和学生 (S)。一个教师可以讲授多门课程，一门课程可以为多个教师讲授，同样一个学生可以选听多门课程，一门课程可以为多个学生选听。
- 在这种情况下，T，C，S三者之间是多对多关系，(T,C,S)三个属性的组合是关系TCS的候选码，称为**全码**，T,C,S都是主属性。

主属性与非主属性



- **主属性 (Prime Attribute)**：包含在候选码中的的各属性称为主属性。
- **非主属性 (Non-Prime Attribute)**：不包含在任何候选码中的属性称为非码属性。
 - 在最简单的情况下，一个候选码只包含一个属性，如学生关系中的“学号”，教师关系中的“教师号”。
 - 最极端情况，全码关系中所有属性都是主属性

外部关系键



- 如果关系 R_2 的一个或一组属性 X 不是 R_2 的主码，而是另一关系 R_1 的主码，则该属性或属性组 X 称为关系 R_2 的**外部关系键**或**外键**（**Foreign key**）。

例：有如下两个关系模式：

教师信息（教师编号，姓名，性别，院系编号）

院系信息（院系编号，院系名）

R1	
depid	depname
1	计算机
2	外语系
3	电子系

R2		
id	name	depid
2001	张三	1
2002	李四	1
2003	王五	2
2004	钱六	3



知识点**21**: 实体完整性

关系模型的完整性



- 为了维护数据库中数据与现实世界的一致性，对关系数据库的插入、删除和修改操作必须有一定的约束条件，这就是关系模型的三类完整性：
 - 实体完整性
 - 参照完整性
 - 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，由关系系统自动支持
 - 用户定义的完整性
 - 特定的数据库在特定的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。

实体完整性



- **实体完整性**是指关系键的值不能为空或部分为空。
- 若属性 A 是基本关系 R 的主属性，则属性 A 不能取空值

实体完整性的说明



- 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。
- 关系模型中以主键来唯一标识元组。
 - 例如，学生关系中的属性“学号”可以唯一标识一个元组，也可以唯一标识学生实体。
- 如果主键中的值为空或部分为空，即主属性为空，则不符合关系键的定义条件，不能唯一标识元组及与其相对应的实体。
- 主键的值不能为空或部分为空



知识点22: 参照完整性

参照完整性



- 参照完整性也称为引用完整性
- 参照完整性就是描述实体与实体之间的联系。

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

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	

参照完整性



- 形式定义如下：
 - 如果属性集 K 是关系模式 R_1 的主键， K 也是关系模式 R_2 的外键，那么在 R_2 的关系中， K 的取值只允许两种可能，或者为空值，或者等于 R_1 关系中某个主键值。
 - 这条规则的实质是“不允许引用不存在的实体”。
- 关系模式 R_1 的关系称为“参照关系”，关系模式 R_2 的关系称为“依赖关系”。
- “主表”和“子表”。

外键的说明



- 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

参照完整性举例



- 在关系数据库中有下列两个关系模式：

S (S#, SNAME, AGE, SEX)

SC (S#, C#, GRADE)

- 如果关系SC中有一个元组 (S7, C4, 80)，而学号S7却在关系S中找不到
- 在关系SC中引用了一个不存在的学生实体，这就违反了参照完整性规则。



知识点**23**: 用户自定义完整性

用户定义完整性



- **用户定义完整性**是针对某一具体关系数据库的约束条件。
- 它反映某一具体应用所涉及的数据必须满足的语义要求。
 - 例如，选课关系中成绩不能为负数；某些数据的输入格式要有一些限制等
- 关系模型应该提供定义和检验这类完整性的机制，以便使用统一的、系统的方法处理它们，而不要由应用程序承担这一功能。

用户定义完整性（续）



例如学生的年龄定义为两位整数，范围还太大，我们可以写如下规则把年龄限制在15～30岁之间：

```
CHECK (AGE BETWEEN 15 AND 30)
```



知识点24: SQL语言介绍

SQL的产生与发展



- 最早的SQL原型是IBM的研究人员于20世纪70年代开发的
- 1986年10月，美国ANSI（American National Standard Institute，美国国家标准学会）采用SQL作为关系数据库管理系统的标准语言（ANSI X3. 135-1986）
- 后为ISO（International Organization for Standardization，国际标准化组织）采纳为国际标准

SQL语言功能概述



SQL功能	命令
数据定义	CREATE, DROP, ALTER
数据操纵	SELECT, INSERT, UPDATE, DELETE
数据控制	GRANT, REVOKE

SQL语言功能概述（续）



- 数据定义语言（Data Definition Language, DDL）
 - 数据定义语言用来定义数据库的逻辑结构，包括定义表、视图和索引。
 - 数据定义仅定义结构，不涉及具体的数据。

SQL语言功能概述（续）



- 数据操纵语言（Data Manipulation Language, DML）
 - 数据操纵语言包括数据查询和数据更新两大类操作。
 - 数据查询是其核心部分，可以按照一定的条件来获取特定的数据部分呈现给用户；
 - 数据更新包括插入、删除和修改操作。
 - 数据操纵就是对数据库中的具体数据进行存取操作。

SQL语言功能概述（续）



- 数据控制语言（Data Control Language, DCL）
 - 数据控制主要包括数据库的安全性和完整性的控制，以及对事务控制的描述。

SQL的特点



■ 一体化

- SQL语言则集数据定义、操纵和控制功能于一体，语言风格统一，可以独立完成数据库生命周期中的全部活动

■ 高度非过程化

- SQL是一种第四代语言(4GL)
- 用户只需要提出“做什么”，不需要具体指明“怎么做”
- 不但大大减轻了用户负担，而且有利于提高数据独立性

SQL的特点（续）



■ 面向集合

- SQL语言也采用集合操作方式，不仅操作对象和查找结果可以是元组的集合，而且一次插入、删除、更新操作的对象也可以是元组的集合。

■ 能以多种方式使用

- 可以直接以命令方式交互使用
- 可以嵌入到某种高级程序设计语言（如C、C++、C#、Java）中去使用。

■ 语言简洁，易学易用

- 语言十分简洁，核心功能只用了9个命令