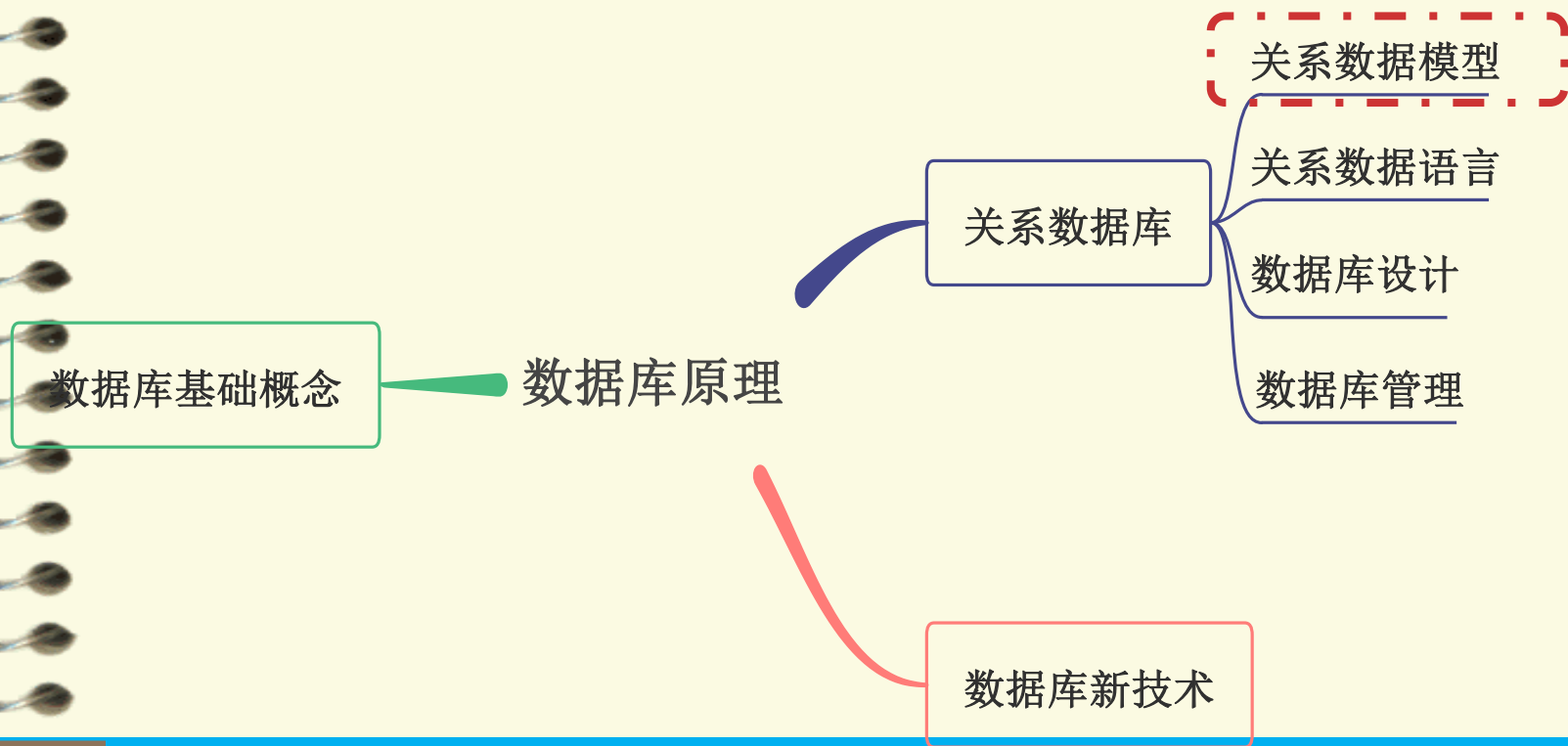


数据库原理

第2章 关系数据模型

辽东学院 鲁 琴

本章内容



数据库的世界

◆ 关系型

– 表、SQL



ORACLE®



◆ 非关系型

– noSQL, 21世纪, 特殊需求



mongoDB



redis



Cassandra

关系数据库系统

- ◆ 关系数据库系统是支持关系模型的数据库系统
- ◆ 1970年IBM公司的E.F.Codd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

关系模型

1. 关系数据结构
2. 关系操作
3. 完整性约束

1. 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。

学号	姓名	年龄	性别	籍贯
S1	WANG	20	M	北京
S4	LIU	18	F	山东
S2	HU	18	M	上海
S3	XIA	19	F	四川

相关概念

◆ 关系模型建立在集合代数的基础上

① 域

② 笛卡尔积

③ 关系

④ 关系模式

⑤ 关系数据库

学号	姓名	年龄	性别	籍贯
S1	WANG	20	M	北京
S4	LIU	18	F	山东
S2	HU	18	M	上海
S3	XIA	19	F	四川

① 域(Domain)

◆ 域是一组具有相同数据类型的值的集合

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

学号	姓名	年龄	性别	籍贯
S1	WANG	20	M	北京
S4	LIU	18	F	山东
S2	HU	18	M	上海
S3	XIA	19	F	四川

② 笛卡尔积 (Cartesian Product)

◆ 给定一组域 D_1, D_2, \dots, D_n

- 这些域可以完全不同
- 也可以部分或全部相同

◆ 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

笛卡尔积举例

$D_1 = \text{SUPERVISOR} = \{ \text{张清玫}, \text{刘逸} \}$

$D_2 = \text{SPECIALITY} = \{ \text{计算机专业}, \text{信息专业} \}$

$D_3 = \text{POSTGRADUATE} = \{ \text{李勇}, \text{刘晨}, \text{王敏} \}$

分量

则 $D_1 \times D_2 \times D_3 =$

元组

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

基数

$D_1: 2$

$D_2: 2$

$D_3: 3$

$D_1 \times D_2 \times D_3 : 12$

笛卡尔积的二维表表示方法

$$D_1 \times D_2 \times \dots \times D_n =$$

◆ 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

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)

◆ $D_1 \times D_2 \times \dots \times D_n$ 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)

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)

关系举例

◆在笛卡尔积中取出有实际意义的元组来构造关系

SAP(SUPERVISOR, SPECIALITY, POSTGRADUATE)

假设：

- 专业与导师：1:n
- 导师与研究生：1:n

元组

于是：**SAP**关系可以包含三个元组

{ (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) }

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

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

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）

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

码 (Key, 键)

由一个或多个属性组成

- ✓ 候选码 (Candidate Key): 在关系中能唯一标识元组的属性或属性集
- ✓ 主属性 (Prime Attribute): 候选码的各个属性
- ✓ 主码 (Primary Key): 用户选作元组标识的候选键
- ✓ 全码 (All-key): 关系的所有属性是这个关系的候选码

学号	姓名	年龄	性别	籍贯
S1	WANG	20	M	北京
S4	LIU	18	F	山东
S2	HU	18	M	上海
S3	XIA	19	F	四川

(学号)是候选码
(学号)是主码

讨论: A banking Database

acctNo	type	balance
12345	savings	12000
23456	checking	1000
34567	savings	25

The relation **Accounts**

firstName	lastName	idNo	account
Robbie	Banks	901-222	12345
Lena	Hand	805-333	12345
Lena	Hand	805-333	23456

The relation **Customers**

The key of each relation?

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

- 由基本表或其他视图导出的表，是虚表，不对应实际存储的数据

基本关系的性质

学号	姓名	出生日期	性别	联系电话
S1	张三	1999/3/3	女	14541555656
S2	李四	1998/12/23	男	12547894545
S3	王二	1998/4/5	男	12345678978

◆ 列是同质的

– 每一列中的分量是同一类型的数据，来自同一个域

◆ 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

◆ 列的顺序无所谓

◆ 任意两个元组不能完全相同

◆ 行的顺序无所谓

◆ 分量必须取原子值

学号	姓名	出生日期	性别	联系电话	
				手机	小号
S1	张三	1999/3/3	女	14541555656	765089
S2	李四	1998/12/23	男	12547894545	765024
S3	王二	1998/4/5	男	12345678978	745033

讨论

一共有多少种
等价形式？

下面各关系是否等价？

学号	性别	出生日期	姓名	联系电话
S1	女	1999/3/3	张三	14541555656
S2	男	1998/12/23	李四	12547894545
S3	男	1998/4/5	王二	12345678978

学号	姓名	出生日期	性别	联系电话
S1	张三	1999/3/3	女	14541555656
S2	李四	1998/12/23	男	12547894545
S3	王二	1998/4/5	男	12345678978

学号	姓名	出生日期	性别	联系电话
S3	王二	1998/4/5	男	12345678978
S2	李四	1998/12/23	男	12547894545
S1	张三	1999/3/3	女	14541555656

姓名	性别	出生日期	学号	联系电话
张三	女	1999/3/3	S1	14541555656
李四	男	1998/12/23	S2	12547894545
王二	男	1998/4/5	S3	12345678978

讨论: A banking Database

acctNo	type	balance
12345	savings	12000
23456	checking	1000
34567	savings	25

The relation **Accounts**

firstName	lastName	idNo	account
Robbie	Banks	901-222	12345
Lena	Hand	805-333	12345
Lena	Hand	805-333	23456

The relation **Customers**

- a) 每个关系的属性有哪些?
- b) 每个关系的元组有哪些?
- c) 从每个关系中举例说出一个元组的分量有哪些?
- d) 每个属性的域?
- e) 写出每个关系的另一个等价形式?

④ 关系模式

◆ 什么是关系模式

- 关系模式（**Relation Schema**）是对关系的描述
- 关系模式是**型**，关系是**值**

- 关系

- **学生**

学号	姓名	年龄	性别	籍贯
S1	WANG	20	M	北京
S4	LIU	18	F	山东
S2	HU	18	M	上海
S3	XIA	19	F	四川

- 关系模式

- **学生**(学号,姓名,年龄,性别,籍贯)

关系模式的表示

- ◆ 关系模式通常可以记为

$R(U)$

或 $R(A_1, A_2, \dots, A_n)$

R 关系名

A_1, A_2, \dots, A_n 属性名

学生(学号,姓名,年龄,性别,籍贯)

关系模式 VS. 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

学生(学号,姓名,年龄,性别,籍贯)

学号	姓名	年龄	性别	籍贯
S1	WANG	20	M	北京
S4	LIU	18	F	山东
S2	HU	18	M	上海
S3	XIA	19	F	四川

⑤ 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

- 例：教学管理数据库中有四个关系：

教师关系T，课程关系C，学生关系S，选课关系SC

- ◆ 关系数据库模式是关系数据库的型，是对关系数据库的描述

- 例：教学管理数据库中有四个关系模式：

T(TID,TNAME,TITLE)

C(CID,CNAME,TID)

S(SID,SNAME,AGE,SEX)

SC(SID,CID,SCORE)

关系模式

数据库模式

讨论: A banking Database

acctNo	type	balance
12345	savings	12000
23456	checking	1000
34567	savings	25

The relation **Accounts**

firstName	lastName	idNo	account
Robbie	Banks	901-222	12345
Lena	Hand	805-333	12345
Lena	Hand	805-333	23456

The relation **Customers**

- 每个关系的属性有哪些?
- 每个关系的元组有哪些?
- 从每个关系中举例说出一个元组的分量有哪些?
- 这个数据库有几个关系? 写出这个数据库的数据模式?
- 每个属性的域?
- 写出每个关系的另一个等价形式?

讨论：A banking Database

一个关系实例的属性个数和元组个数如下所示，如果考虑到元组的顺序和属性的顺序，则这个关系实例有多少种不同的表示形式？

- a) 3个属性，3个元组。例如关系Account?
- b) 4个属性，5个元组？
- c) n 个属性， m 个元组？

关系模型

1. 关系数据结构
2. 关系操作
3. 完整性约束

2. 关系操作

◆ 常用的关系操作

- 查询、插入、删除、修改

◆ 关系操作的特点

- 集合操作方式

◆ 关系数据语言的种类

- 关系代数语言
- 关系演算语言
- 具有上两者双重特点的语言.如 SQL

早期的
抽象的

关系数据库的标准语言

◆ 关系数据语言的特点

- 是一种高度非过程化的语言
- 能够嵌入高级语言中使用

关系模型

1. 关系数据结构
2. 关系操作
3. 完整性约束

3. 关系的完整性约束

- ◆ 关系模型的完整性规则是指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

① 实体完整性(Entity Integrity)

◆ 主码唯一且不能为空

◆ 主属性不能取空值。

— 例

SAP(SUPERVISOR, SPECIALITY, POSTGRADUATE)

POSTGRADUATE属性为主码（假设研究生不会重名），则其不能取空值。

— 例：

T(T#,TNAME,TITLE)

C(C#,CNAME,T#)

S(S#,SNAME,AGE,SEX)

SC(S#,C#,SCORE)

关系模型必须遵守实体完整性规则

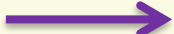
- (1) 实体完整性规则是针对**基本关系**而言的
- (2) 现实世界中的实体和实体间的联系都是可区分的，即它们具有某种**唯一性标识**
- (3) 关系模型中以**候选码**作为唯一性标识
- (4) 所谓**空值**就是“不知道”或“无意义”的值
 - ✓ 如果主属性取空值，就说明存在某个不可标识的实体，即存在不可区分的实体，这与 (2) 相矛盾，因此这个规则称为**实体完整性**

② 参照完整性

- ◆ 关系间的引用
- ◆ 外码
- ◆ 参照完整性规则

关系间的引用

现实世界 关系模型

◆ 实体  关系

◆ 联系  关系

例 学生实体 专业实体 专业与学生间的联系

专业 (专业号, 专业名)

学生 (学号, 姓名, 性别, 专业号, 出生日期)

专业 (专业号, 专业名)

学生 (学号, 姓名, 性别, 专业号, 出生日期)

专业号	专业名
01	信息
02	数学
03	计算机

被参照表
(父表或主表)

学号	姓名	性别	专业号	出生日期
160001	张三	女	01	1995/9/9
160002	李四	男	01	1996/2/20
160003	王五	男	01	1995/4/30
160004	赵六	女	02	1993/5/6
160005	钱七	男	02	1995/8/12

参照表
(子表或从表)

关系间的引用

例 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, 出生日期)

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	出生日期
160001	张三	女	01	1995/9/9
160002	李四	男	01	1996/2/20
160003	王五	男	01	1995/4/30
160004	赵六	女	02	1993/5/6
160005	钱七	男	02	1995/8/12

课程

课程号	课程名
1	信息管理学
2	程序设计语言
3	数据库原理

选修

学号	课程号	成绩
160001	2	85
160001	3	90
160001	4	79
160002	2	68
160003	2	76
160004	2	90
160005	2	79

外码 (Foreign Key)

- 如果一个关系**R**中的一个属性**F**对应着另一关系**S**的主码**K**, 那么**F**在关系**R**中称为**外码**

专业 (专业号, 专业名)

被参照表
(父表或主表)

学生 (学号, 姓名, 性别, 专业号, 出生日期)

参照表
(子表或从表)

说明

- 关系 R 和 S 不一定是不同的关系
- 被参照关系 S 的主码 K 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

学生（学号，姓名，性别，专业号，出生日期，班长）

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
 - 外码的取值必须为：
 - 或者取空值
 - 或者等于被参照表中某个元组的主码值

例

学生关系中每个元组的“专业号”属性的取值：

- (1) 空值(NULL)，表示尚未给该学生分配专业
- (2) 非空值(NOT NULL)，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

专业 (专业号, 专业名)

学生 (学号, 姓名, 性别, 专业号, 出生日期)

讨论：

“选修”关系中“学号”和“课程号”是外码，它们的取值范围分别为？

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

学生（学号，姓名，性别，专业号，出生日期）

课程（课程号，课程名，学分）

选修（学号，课程号，成绩）
~~~~~

## 讨论：

“学生”关系中“班长”是外键，它的取值范围？

“班长”属性值可以取两类值：

- (1) 空值(NULL)，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

学生（学号，姓名，性别，专业号，出生日期，班长）

### 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

# 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}

## 讨论：

“学生”关系可以自定义哪些约束？

---

学生（学号，姓名，性别，专业号，出生日期，班长）

- “姓名”属性取唯一值
- “出生日期”属性的取舍范围为[1980/1/1, 2015/12/12]
- “性别”属性只能取值{'男', '女'}

## 讨论：

有两个关系T和C，问下面的操作能否被执行？为什么？

**T**

| T#  | TNAME | TITLE |
|-----|-------|-------|
| 001 | 张三    | 教授    |
| 002 | 李四    | 讲师    |
| 003 | 王二    | 副教授   |

**C**

| C# | CNAME | T#  |
|----|-------|-----|
| 1  | 数据库   | 001 |
| 2  | 数学    | 002 |
| 3  | 信息系统  | 003 |

其中C#是主码，T#是外码，参照T中的T#

- ①在C中插入行('4', '数据结构', '003')
- ②在C中插入行('3', '操作系统', '001')

- A.①能 ②能
- B.①能 ②不能
- C.①不能 ②能
- D.①不能 ②不能

# 本章小结

