关系数据库

聂玉敏
生物信息学系
yumin_nie@njmu.edu.cn

关系数据库

- 关系数据库基于关系模型。
- 提出关系模型的是美国**IBM**公司的**E.F.Codd**

  E.F.Codd, “A Relational Model of Data for Large Shared Data Banks”, 《Communication of the ACM》,1970

- 早期代表系统
  - System R：由IBM研制
  - INGRES：由加州Berkeley分校研制

关系数据库

- 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。

- 目前主流的关系数据库管理系统软件产品
  - IBM DB2 UDB、Oracle、Informix、Sybase、 MS SQL Server
  - MySQL、MariaDB
  - Access、Foxpro、Foxbase

关系模型

- 关系数据库基于关系模型。
- 关系模型是以集合论中的关系概念为基础发展起来的数据模型。

- 关系模型的组成要素
  - 关系数据结构
  - 关系操作
  - 关系完整性约束

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。
- 实体是客观存在并可相互区别的事物，可以是具体的人、事、物或抽象的概念，如学生、一堂课、一次比赛。

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。
- 关系模型的这种简单的数据结构能够表达丰富的含义，描述出现实世界的实体以及实体间的各种联系。

Instructor 关系

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Figure 2.1 The *instructor* relation.

关系模型中术语

- **关系(Relation)**
  - 一个关系对应通常说的一张表(Table)。
- **元组(Tuple)**
  - 表中的一行(Row)即为一个元组。
- **属性(Attribute)**
  - 表中的一列即为一个属性，给每一个属性起一个名称即属性名。

关系模型中术语

- 域（Domain）
  - 属性的取值范围。
- 分量
  - 元组中的一个属性值。
- 关系模式
  - 对关系的描述，相当于“表头”部分。
  - 表示形式：关系名（属性1，属性2，...，属性n）。
  - 例：学生（学号，姓名，年龄，性别，系，年级）。

关系模式与关系

- 关系可看作是关系模式在某一时刻的状态或内容。

- 关系模式是静态的、稳定的，而关系是动态的、随时间变化的，因为关系操作在不断地更新着数据库中的数据。

关系示例

关系名

instructor

关系

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci	75000

属性名

元组(行)

分量

属性(列)

关系模式：Instructor（ID、name、dept_name、salary）

码(key)

- **超码（Super key; SK）**
  - 一个或多个属性的集合。
  - 属性的组合能够唯一的确定一个元组。
- **候选码（Candidate key; CK）**
  - 如果超码中的任一属性去掉后剩余的属性集不能唯一标识一个元组，则该属性集是关系上的候选码。
- **主码（Primary key; PK）**
  - 从候选码中选择一个作为关系的主码。
  - 一般情况下，如不加特别说明，码即指主码。
- **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）。

码(key)

- **外码（Foreign key; FK）**
  - 如果关系R1中的某个属性集是另外一个关系R2的候选码，那么该属性集对于关系R1而言就是它的外码。
- **主属性**
  - 候选码中的属性称为主属性。
  - 习惯上把一个关系模式的主码属性列在其他属性前面，并将主码属性加上下划线。
  - 例：学生（学号，姓名，年龄，性别，系，年级）
- **非主属性（或非码属性）**
  - 不包含在任何候选码中的属性称为非主属性。

码示例

instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000

关系模式：instructor(ID, name, dept_name,salry)
超码： {ID}, {ID, name}, {ID, dept_name}
候选码：{ID}
主码：{ID}
主属性：ID
非主属性：name, dept_name, salary

码示例

instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000

假设同一系中无重名

关系模式：instructor(ID, name, dept_name,salry)

超码： {ID}、{ID, name}、{name, dept_name}

候选码： {ID} 、{name, dept_name}

主码：{ID}或{name, dept_name}

主属性：ID、name、dept_name

非主属性：salary

外码示例

instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000

department

dept_name	*building*	*budget*
Biology	Watson	90000
Comp. Sci.	Taylor	100000
Elec. Eng.	Taylor	85000
Finance	Painter	120000
History	Painter	50000
Music	Packard	80000
Physics	Watson	70000

关系模式：instructor(ID, name, dept_name, salary);
department(dept_name, building, budget)
主码：instructor中的ID；department中的dept_name
instructor的外码：dept_name;
外码的参照关系：instructor
外码的被参照关系：department

全码示例

teaches

ID	*course_id*
10101	CS-101
10101	CS-315
10101	CS-347
12121	FIN-201
15151	MU-199
22222	PHY-101
32343	HIS-351
45565	CS-101
45565	CS-319

关系模式：teaches(ID, course_id);
候选码：{ID, course_id}
主属性： ID, course_id

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，关系是规范化了的二维表中行的集合。为了使相应的数据操作简化，在关系模型中，对关系作了种种限制。

关系的性质

1. 关系中不允许出现相同的元组

- 因为在数学上集合中没有相同的元素，而关系是元组的结合，所以作为集合元素的元组应该是唯一的。

ID	name	dept_name	salary
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
12121	Wu	Finance	90000
45565	Katz	Comp. Sci.	75000

不允许

关系的性质

- **2. 关系中元组的顺序（即行序）是无关紧要的，在一个关系中可以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。

示例

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Figure 2.1 The *instructor* relation.

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

Figure 2.4 Unsorted display of the *instructor* relation.

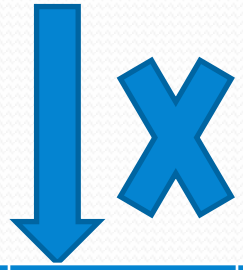
关系的性质

- **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。

ID	name	salary
22222	Einstein	95000
12121	Wu	90000
45565	Katz	75000

ID	salary	name
22222	95000	Einstein
12121	90000	Wu
45565	75000	Katz

ID	name	salary
22222	Einstein	95000
12121	Wu	90000
45565	Katz	75000



ID	salary	name
22222	Einstein	95000
12121	Wu	90000
45565	Katz	75000

关系的性质

- **4.** 同一属性名下的各个属性值必须来自同一个域，是同一个类型的数据。

ID	name	dept_name	salary
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
98345	Kim	Elec. Eng.	Physics
45565	Katz	Comp. Sci.	75000

不允许

关系的性质

- 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。

course_id	*prereq_id*
BIO-301	BIO-101
BIO-399	BIO-101
CS-190	CS-101
CS-315	CS-101
CS-319	CS-101
CS-347	CS-101
EE-181	PHY-101

Figure 2.3 The *prereq* relation.

关系的性质

- **6. 关系中每一个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。

工资表

职工号	姓名	职 称	工 资			扣 除		实 发
			基 本	津 贴	职务	房 租	水 电	
86051	陈 平	讲 师	1305	1200	50	160	112	2283

关系及表术语对比

关系术语	一般表格的术语
关系名	表名
关系模式	表头（表格的描述）
关系	（一张）二维表
元组	记录或行
属性	列
属性名	列名
属性值	列值
分量	一条记录中的一个列值
非规范关系	表中有表（大表中嵌有小表）

2. 关系操作

查询

- 选择（Select）、投影(Project)、连接(Join)、除(Divide)、并(Union)、交(Intersection)、差(Difference)等
- 选择、投影、并、差、笛卡尔积是五种基本操作
- 关系的查询表达能力很强，是最主要的部分

数据更新

- 插入（Insert）、删除(Delete)、修改(Update)

关系数据库语言

- 关系模型与其他模型相比，最有特色的是它的数据库语言
- 关系数据库所使用的语言一般都具有定义、查询、更新和控制一体化的特点，而查询是最主要的部分
- 关系数据库的核心部分是查询，而查询的条件要使用关系运算表达式来表示
- 按表达查询的方法不同，关系运算可分为关系代数和关系演算两大类

关系数据库语言的分类

- 关系代数语言
  - 用对关系的运算来表达查询要求
  - 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）

3. 关系完整性约束

- **关系模型的完整性规则是对关系的某种约束条件，这些约束条件实际上是现实世界的要求。**
  - 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

- **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

- **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。
  - 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。
  - 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，这与实体的可区分性矛盾，因此这个规则称为实体完整性。

参照完整性

- 若属性（或属性组）***F***是关系***R***的外码，它与关系***S***的主码K_s相对应（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

外码

instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000

department

dept_name	*building*	*budget*
Biology	Watson	90000
Comp. Sci.	Taylor	100000
Elec. Eng.	Taylor	85000
Finance	Painter	120000
History	Painter	50000
Music	Packard	80000
Physics	Watson	70000

instructor关系中每个元组的dept_name属性只取两类值：

（1）空值，表示该教师所在院系不明

（2）非空值，这时该值必须是department关系中某个元组的dept_name值，表示该教师不可能分配一个不存在的院系中

用户定义的完整性

- 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

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Figure 2.1 The *instructor* relation.

不能为负数

关系完整性约束

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，由关系系统自动支持
- 任何关系数据库系统都应该支持这两类完整性

用户定义的完整性

- 应用领域需要遵守的约束条件，体现了具体领域中的语义约束

The END