1.主键
2.关系模式

关系数据模型基础

关系（Relation）

关系建立在集合代数的基础上。

单一的数据结构：关系

关系数据库系统中，只有“表”这一种数据结构。

8.☆数据模型定义：
对数据库中数据进行精确描述的方法

逻辑结构：二维表

一个关系对应通常说的一张表。

9.☆数据模型三要素
数据结构
数据操作
完整性约束条件

在用户观点下，关系模型中数据的逻辑结构是一张二维表。

元组（Tuple）

- 表中的一行即为一个元组

属性（Attribute）

- 表中的列命名为属性

1、☆主码（Key）

- 也称码、键。表中的某个属性组，它可以唯一确定一个元组

域（Domain）

- 是一组具有相同数据类型的值的集合，属性的取值范围来自某个域

分量

- 元组中的一个属性值
- 元组的每个分量都具有原子性

关系模式

2、Student(Sno, Sname, Ssex)

- 对关系的描述，关系名和其属性集合的组合称为这个关系的模式

关系数据结构及形式化定义

关系

- 单一的数据结构——关系
- 逻辑结构——二维表
- 关系建立在集合代数的基础上

1

域（Domain）

域是一组具有相同数据类型的值的集合，比如整数、实数、介于某个取值范围内的整数、{"男"，"女"}

笛卡尔积（Cartesian Product）

给定一组域$D_1, D_2, \ldots, D_n$，允许其中某些域是相同的

$D_1, D_2, \ldots, D_n$的笛卡尔积为：$\{(d_1, d_2, \ldots, d_n) | d_i \in D_i, i = 1, 2, \ldots, n\}$

它是所有域的所有取值的组合的集合，集合中的元素不能重复

元组（Tuple）

笛卡尔积中每一个元素$(d_1, d_2, \ldots, d_n)$叫做一个n元组（n-tuple）或简称元组

分量（Component）

笛卡尔积元素$(d_1, d_2, \ldots, d_n)$中的每一个值d_i叫做一个分量

基数（Cardinal Number）

若$D_i (i = 1, 2, \ldots, n)$为有限集，其基数为$m_i (i = 1, 2, \ldots, n)$，则$D_1 \times D_2 \times \cdots \times D_n$的基数$M$为：$M = \prod_{i=1}^{n} m_i$。

笛卡尔积的表示方法

笛卡尔积可表示为一张二维表，表中的每行对应一个元组，表中的每列对应一个域

关系（Relation）

关系

$D_1 \times D_2 \times \cdots \times D_n$的笛卡尔积的某个子集才有实际含义

$D_1 \times D_2 \times \cdots \times D_n$的**子集**叫做在域$D_1, D_2, \ldots, D_n$上的**关系**，表示为：

$R(D_1, D_2, \ldots, D_n)$，

其中R为关系名，n为关系的**目或度**（Degree）。

元组

关系中的每个元素是关系中的元组，通常用t表示。

单元关系与二元关系

$n = 1$时，称该关系为**单元关系**（Unary relation）或**一元关系**。

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

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。

属性

关系中不同列可以对应相同的域。

为了区分列，为每列起一个名字，称作**属性**。

n目关系必有n个属性。

码

3.

- 候选码（Candidate key）
    - 若关系中的某一属性组能唯一地标识一个元组，则称该属性组为候选码
    - 最简单的情况：候选码只包含一个属性
- 全码（All key）
    - 最极端的情况：关系的所有属性是这个关系的候选码，称为全码
- 主码（Primary key）
    - 若一个关系有多个候选码，则选定其中一个为主码
- 主属性
    - 所有候选码的属性称为主属性（Prime atttribute）
    - 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）

4.

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
    - 有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

基本关系的性质

- 列是同质的（Homogeneous）
- 不同的列可出自同一个域
    - 其中的每一个列称为一个属性
    - 不同的属性要给予不同的属性名
- 列的顺序无所谓，列的次序可以任意交换
- 任意两个元组的候选码不能相同
- 行的顺序无所谓，行的次序可以任意交换
- 分量必须取原子值（第一范式）

关系模式

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
        - 属性与域之间的映像关系
    - 完整性约束条件

定义关系模式

关系模式可以形式化地定义为：

$R(U, D, DOM, F)$

R	关系名
U	组成该关系的属性名集合
D	U中属性所来自的域
DOM	属性向域的映像集合
F	属性间数据的依赖关系的集合

关系模式通常简记为：$R(U)$或$R(A_1, A_2, \ldots, A_n)$，其中A_i为属性名

2☆ 关系模式与关系

- 关系模式 型
  - 是对关系的描述
  - 静态的、稳定的
- 关系 实例 | 值
  - 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

关系数据库

定义

在一个给定的应用领域中，所有关系的集合形成一个关系数据库

关系数据库的型

关系数据库模式，是对关系数据库的描述

关系数据库的值

关系模式在某一时刻对应的关系的集合，通常称为关系数据库

关系模型的存储结构

有的RDBMS中一个表对应一个操作系统中的文件，将物理数据组织交给OS完成

有的RDBMS从OS那里申请若干个大的文件，自己进行数据的组织

关系操作

常用的关系操作

- 查询操作
  - 选择、投影、连接、除、并、差、交、笛卡尔积

选择投影乘 并 求差

5、

- 选择、投影、并、差、笛卡尔积是五种基本操作

- 数据更新
  - 增删改

6、关系操作的特点

- 集合操作方式：操作的对象和结构都是集合，**一次一集合**的方式

关系代数语言

- 用对关系的运算达到查询要求
- 代表：ISBL

关系演算语言

- 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 Qurey Language）

7、关系的完整性

实体完整性（Entity Integrity）

主键 非空且不重复

- 若属性A是基本关系R的主属性，则属性A不能取空值（空值就是不知道或不存在或无意义的值）

参照完整性

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，自然存在着关系与关系间的引用

外码（Foreign Key）

R的外键在R中不是主键，在S中是主键.

- 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F是R的**外码**
  - 关系R和S不一定是不同的关系
- 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）
  - 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不属于不同关系时，往往取相同的名字，以便于识别）

参照完整性规则

若属性（或属性组）F是基本关系R的**外码**，且它与基本关系S的主码K_s相对应，则R中每个元组在F上的值必须为**空值**或等于S中某个元组的**主码值**

自定义完整性

not null unique check

关系代数

关系代数是一种抽象的**查询语言**，它用对关系的运算来表达查询。

- 运算对象是关系
- 运算结果也是关系
- 关系代数的运算符有两类
    - 集合运算符
        - 传统的集合运算是从关系的行的角度进行的
    - 专门的关系运算符
        - 专门的关系运算不仅涉及行而且涉及列
- 运算有8个
    - 并、差、交、笛卡尔积、投影、选择、连接、除
- 基本运算有5个
    - 并、差、笛卡尔积、投影、选择
- 交、连接、除可以用5种基本运算来表达

传统的集合运算

并（Union）

- 条件
    - R和S具有相同的度n（即两个关系都有n个属性）
    - 相应的属性取自同一个域
- $R \cup S$
    - 仍为n目关系，由属于R或属于S的所有元组组成
    - $R \cup S = \{t|t \in R \vee t \in S\}$

差（Difference）

- 条件
    - R和S具有相同的度n（即两个关系都有n个属性）
    - 相应的属性取自同一个域
- $R - S$
    - 仍为n目关系，由属于R而不属于S的所有元组组成
    - $R - S = \{t|t \in R \wedge t \notin S\}$

交（Intersection）

- 条件
  - R和S具有相同的度n（即两个关系都有n个属性）
  - 相应的属性取自同一个域
- $R \cap S$
  - 仍为n目关系，由既属于R又不属于S的所有元组组成
  - $R \cap S = \{t|t \in R \wedge t \in S\}$
  - $R \cap S = R - (R - S)$

笛卡尔积（Cartesian Product）

严格的地讲，应该是广义的笛卡尔积（Extended Cartesian Product）

- 条件
  - R：n目关系，k_1个元组
  - S：m目关系，k_2个元组
- $R \times S$
  - 列：$n+m$列元组的集合
    - 元素的前n列是关系R的一个元组
    - 后m列是关系S的一个元组
  - 行：$k_1 \times k_2$个元组
    - $R \times S = \{\widehat{t_r t_s}|t_r \in R \wedge t_s \in S\}$

专门的关系运算

符号引入

引入几个符号，设关系模式为$R(A_1, A_2, \ldots, A_n)$

- $R, t \in R, t[A_i]$
  - R为一个关系
  - t为R中的一个元组
  - $t[A_i]$表示元组t中相应于属性A_i的一个分量
- $A, t[A], \bar{A}$
  - 若$A = \{A_{i1}, A_{i2}, \ldots, A_{ik}\}$，其中$A_{i1}, A_{i2}, \ldots, A_{ik}$是$A_1, A_2, \ldots, A_n$中的一部分，则$A$称为属性列或属性组
  - $t[A] = \{t[A_{i1}], t[A_{i2}], \ldots, t[A_{ik}]\}$表示元组$t$在属性组$A$上诸分量的集合
  - $\bar{A}$则表示$A_1, A_2, \ldots, A_n$中去掉$\{A_{i1}, A_{i2}, \ldots, A_{ik}\}$后剩余的属性组
- $\widehat{t_r t_s}$
  - R为n目关系，S为m目关系
  - $t_R \in R, t_S \in S$，$\widehat{t_r t_s}$称为元组的连接
  - $\widehat{t_r t_s}$是一个$n+m$列的元组，前n个分量为R中的一个n元组，后m个分量为S中的一个m元组
- 象集Z_x

  给定一个关系$R(X, Z)$，X和Z是属性组
  - 当$t[X] = x$时，x在R中的象集（Images Set）为：$Z_x = \{t[Z] \mid t \in R, t[X] = x\}$

选择（Selection）

- 选择又称限制（Restriction），从行的角度进行　　1、选择 $\sigma_F(t)$
- 在关系R中选择满足条件F的诸元组
- $\sigma_F(R) = \{t|t \in R \wedge F(t) = '真'\}$

10、投影 $\pi_A(R)$

投影（Projection）

- 从列的角度进行，不仅取消了原关系中的某些列，而且还可能取消某些元组（去除重复行）
- 从R中选择出若干属性组组成新的关系
- $\pi_A(R) = \{t[A] | t \in R\}$，$A$是$R$上的属性组

10、$R \bowtie_{A\theta B} S$

连接（Join）

- 连接也称θ连接
- 从两个关系R和S的笛卡尔积$R \times S$中选取属性组A和B的值满足比较关系θ的元组
- $R \bowtie_{A\theta B} S = \{\widehat{t_r t_s} | t_r \in R \wedge t_s \in S \wedge t_r[A]\theta t_s[B]\}$
  - A和B：分别是R和S上读书相等且可比的属性组
  - θ：比较运算符

θ连接 → 等值连接 → 自然连接

投影 ~~和自然连接~~ 去删除重复元组

两类常用连接运算

- 等值连接（equijoin）
  - θ为=的连接运算称为等值连接
- 自然连接（Natural join）
  - 自然连接是一种特殊的等值连接，将R和S公共属性值全相等的元组进行连接，记作$R \bowtie S$
    - 两个关系中进行比较的分量必须是相同的属性组
    - 在结果中把重复的属性列去掉
- 一般的连接操作是从行的角度去进行，自然连接还要取消重复列，所以同时从行和列的角度进行运算

悬浮元组（Dangling tupie）

略

外连接（Outer join）

略

除（Division）

给定关系$R(X,Y)$和$S(Y,Z)$，其中X,Y,Z为属性组，R中的Y与S中的Y可以有不同的属性名，但必须出自相同的域集。

$R \div S$得到一个新的关系$P(X)$，P是R中满足下列条件的元组在属性组X上的投影：

元组在X上分量值x的象集Y_x包含S在Y投影的集合，记作：

$R \div S = \{t_r[X] \mid t_r \in R \wedge \pi_Y(S) \subseteq Y_x\}$，其中$Y_x$是$x$在$R$中的象集，$x = t_r[X]$

关系演算

选择：$\sigma_F(R)$　F是条件

投影：$\pi_A(R)$　A是属性组

并：$R\cup S$

笛卡尔积：$R\times S$

差：$R-S$

重命名：ρ

θ连接：$R\bowtie_{A\theta B}S$　自然连接 $R\bowtie S$

交：$R\cap S$

选择：$\sigma_{con}(R)$

$\sigma_{name='Jane\ Fonda'}(moviestar)$

$\sigma_{studioName='disney'}(movies)$

投影：$\pi_{A_1,A_2,A_3}(R)$

去删除重复元组

$\pi_{movieTitle,movieYear}(movies)$

重命名：$\rho_{S(A_1,A_2,\cdots,A_n)}(R)$

关系运算的复合：若无括号，从右往左

$\pi_{address}(\sigma_{name='Jane\ Fonda'}(moviestar))$

笛卡尔积：$R\times S$

θ连接：$R\bowtie_\theta S$ → 笛卡尔积+选择（条件为θ）

θ为=时即为等值连接

$\sigma_{studioName=name\ and\ title='Star\ Wars'}(movies\times studio)$

$movies\bowtie_{studioName=name\ and\ title='star\ Wars'}studio$

$\pi_{starName,movieTitle,movieYear,studioName}(movies\bowtie_{movieTitle=title\ and\ movieYear=year}starsIn)$

自然连接 $\bowtie$

将R和S公共属性上相等的元组进行连接

删除重复的公共属性

$\pi_{title,\ year,\ studioName}(\rho_{studio(studioName,\ address,\ presC)}(studio))$

除法 ÷　R÷S

R(X,Y)　S(Y,Z)

设R中元组t

当t[X]=x时，X在R中的象集为x对应的Y的分量的集合

若该象集包括了S中Y的分量集合

则 x∈R÷S

①找公共列Y

②求S中公共列Y的投影(去重)Y'　R中

~~对于每个X上的~~

③求R中X投影X'

④求X'中每个值对应的象集 ~~对于R中每个X的值~~

⑤判断x是否为Y'　对于R中X的每个取值x，①求其在R中的象集

②判断象集是否大于等于S中Y的所有取值

③有则 x∈R÷S

⑥若大于等于，则 x∈R÷S

1.主键
2.关系模式与关系
3.候选码(选一个就是主键)
4.3种关系表
5.5种基本关系操作(选择投
6.关系操作操作对象为集合
7.关系的完整性(3个,实体、参照、自定义)
3种
8.数据模型定义(逻辑模型)→层次概念模型
表
ER,UML
9.数据模型之要素:
10. 选择 $\sigma_F(R)$. 投影 $\pi_A(R)$ θ连接 $R \bowtie_{A\theta B} S$
等值连接
自然连接

除法 $R \div S$