绪论

数据库：永久存储，有组织，可共享

数据模型：概念模型（信息模型）+逻辑模型和物理模型

现实->信息->机器

概念模型：

· 实体

· 属性

· 码

· 实体型：用实体名及其属性集合刻画的实体

· 实体集

逻辑模型

· 层次模型（树）

· 网状模型（图）

· 关系模型（表）

ER图

· 实体-长方形

· 属性-椭圆形

· 关系-菱形

关系模型

学生登记表

学号	姓名	年龄	性别	系名	年级
2013004	王小明	19	女	社会学	2013
2013006	黄大鹏	20	男	商品学	2013
2013008	张文斌	18	女	法律	2013
…	…	…	…	…	…

· 关系：对应一张表

· 元组：表中的一行

· 属性：表中的一列

· 码：主键，某个属性组

· 域：一组具有相同数据类型的值的集合

· 分量：元组中的一个属性

三级模式结构：外模式+模式+内模式

· 外模式：子模式或用户模式，是数据库用户能够看见和使用的局部数据的逻辑结构和特征的描述，是数据库用户的数据视图，是与某一应用有关的数据的逻辑表示

· 模式：逻辑模式，是数据库中全体数据的逻辑结构和特征的描述，是所有用户的公共数据视图

· 内模式：存储模式，一个数据库只有一个内模式。它是数据物理结构和存储方式的描述，是数据在数据库内部的组织方式

二级映像

· 外模式/模式映像：当模式改变时，仅需改变外模式/模式映像，可以使外模式保持不变。应用程序不必修改，保证了**数据与程序的逻辑独立性**

· 模式/内模式映像：当数据库的存储结构改变时，仅需改变模式/内模式映像，可以使模式保持不变，从而应用程序也不用改变。保证了**数据与程序之间的物理独立性**

· 作用：数据与程序之间的独立性使得数据的定义和描述可以从应用程序中分离出去。另外，由于数据的存取由数据库管理系统管理，从而简化了应用程序的编制，大大减少了应用程序的维护和修改。

物理独立性：当数据库的存储结构改变时，由数据库管理员对模式/内模式映像作相应改变，可以使模式保持不变，从而应用程序也不必改变。保证了数据与程序的物理独立性，简称数据的物理独立性。

逻辑独立性：当模式改变时，由数据库管理员对各个外模式/模式的映像作相应改变，可以使外模式保持不变。应用程序是依据数据的外模式编写的，从而应用程序不必修改，保证了数据与程序的逻辑独立性，简称数据的逻辑独立性

关系数据库

关系模式 R（U，D，DOM，F）

· R：关系名

· U：所有属性名

· D：属性来自哪些域

· DOM：属性和域的映射

· F：属性间的依赖关系

连接

自然链接：把共同属性进行等值连接

悬浮元组：两个关系R和S在做自然连接时，关系R中的某些元组在S中不存在对应属性相等的元组

外连接：将悬浮元组也保存在结果关系中，其他属性填空

左外连接：只保留左边关系R中的悬浮元组

右边连接：只保留右边关系S中的悬浮元组

除

保留R中满足S的，且R中列要去掉S的列

关系基本操作

选择，投影，并，差，笛卡尔积

关系的完整性

- 实体完整性规则：若属性A是基本关系R的主属性，则A不能取空值
- 参照完整性规则：若属性F是基本关系R的外码，它与基本关系S的主码K相对应，则对于R中每个元组在F上的值必须
    - 取空值或取等于S中某个元组的主码值

SQL语言

```
SELECT [ALL | DISTINCT] <目标列表达式> [,<目标列表达式>] …
FROM <表名或视图名> [,<表名或视图名>…] | (<SELECT 语句>) [AS] <别名>
[WHERE <条件表达式>]
[GROUP BY <列名 1> [HAVING <条件表达式>]]
[ORDER BY <列名 2> [ASC | DESC]] ;
```

- group by：按<列名1>的值进行分组，该属性列值相等的元组为一个组。如果group by子句带having短语，有满足指定条件的组才予以输出
- order by：结果表还要按<列名2>的值的升序或降序排序

SELECT可以是表中的属性列，也可以是表达式

IN可以用来查找属性值属于指定集合的元组，相反为NOT IN

LIKE可以进行字符串的匹配

聚集函数

COUNT(*)	统计元组个数
COUNT([DISTINCT|ALL] <列名>)	统计一列中值的个数
SUM([DISTINCT|ALL] <列名>)	计算一列值的总和（此列必须是数值型）
AVG([DISTINCT|ALL] <列名>)	计算一列值的平均值（此列必须是数值型）
MAX([DISTINCT|ALL] <列名>)	求一列值中的最大值
MIN([DISTINCT|ALL] <列名>)	求一列值中的最小值

只有COUNT（*）处理空值

聚集函数只能用于SELECT子句和GROUP BY中的HAVING子句

数据库的安全性

安全性控制

自主存取控制方法：用户可以自定义和分配其他用户的操作权限

主要通过grant（授权） revoke（回收）进行控制

由两个元素构成：**数据库对象**和**操作权限**

- grant：

  grant <权限> on 表名(列名) to 用户 with grant option

  若给用户分配权限时带 with grant option 子句，则普通用户可把自己的权限授权其他用户
- revoke：

  revoke <权限> on <数据对象> from <数据库用户名>

数据库的完整性

正确性：符合现实世界的描述

相容性：同一对象在不同表里面是符合逻辑的

维护完整性：

- 提供定义完整性的约束条件的机制
- 提供完整性检查的方法
- 进行违约处理

三大完整性

- 实体完整性：主码唯一，且非空
- 参照完整性：没有外码或仅有一个
- 用户定义完整性：非空，列值唯一，满足某一个条件表达式

关系数据理论

关系模式可能存在的问题：

- 数据冗余
- 更新异常
- 插入异常
- 删除异常

函数依赖

平凡函数依赖：X->Y,Y⊆X,即可以推出自己或自己的一部分

非平凡函数依赖：X->Y,Y⊄X

完全函数依赖：X->Y，且对于X的任何一个真子集X‘，都有X'!->Y

部分函数依赖：X->Y，但Y不完全函数依赖于X

传递函数依赖：X->Y，Y->Z，则称Z对X传递函数依赖

码

设K为R<U,F>中的属性或属性集合，如果U完全依赖于K，则K为R的候选码

如果U部分函数依赖于K，则K为超码

候选码是最小的超码

如果候选码多余一个，则选择其中一个为主码

包含在任何一个候选码中的属性是主属性，不包含在任何候选码中的属性是非主属性

范式

低一级范式的关系模式通过模式分解可以转换为若干个高一级范式的关系模式的集合，这种过程叫规范化

1NF：每一个分量必须是不可分的数据项

1NF->2NF：消除了非主属性对于码的部分函数依赖

2NF：每一个非主属性完全函数依赖于任何一个候选码

2NF->3NF：消除非主属性对于码的传递函数依赖

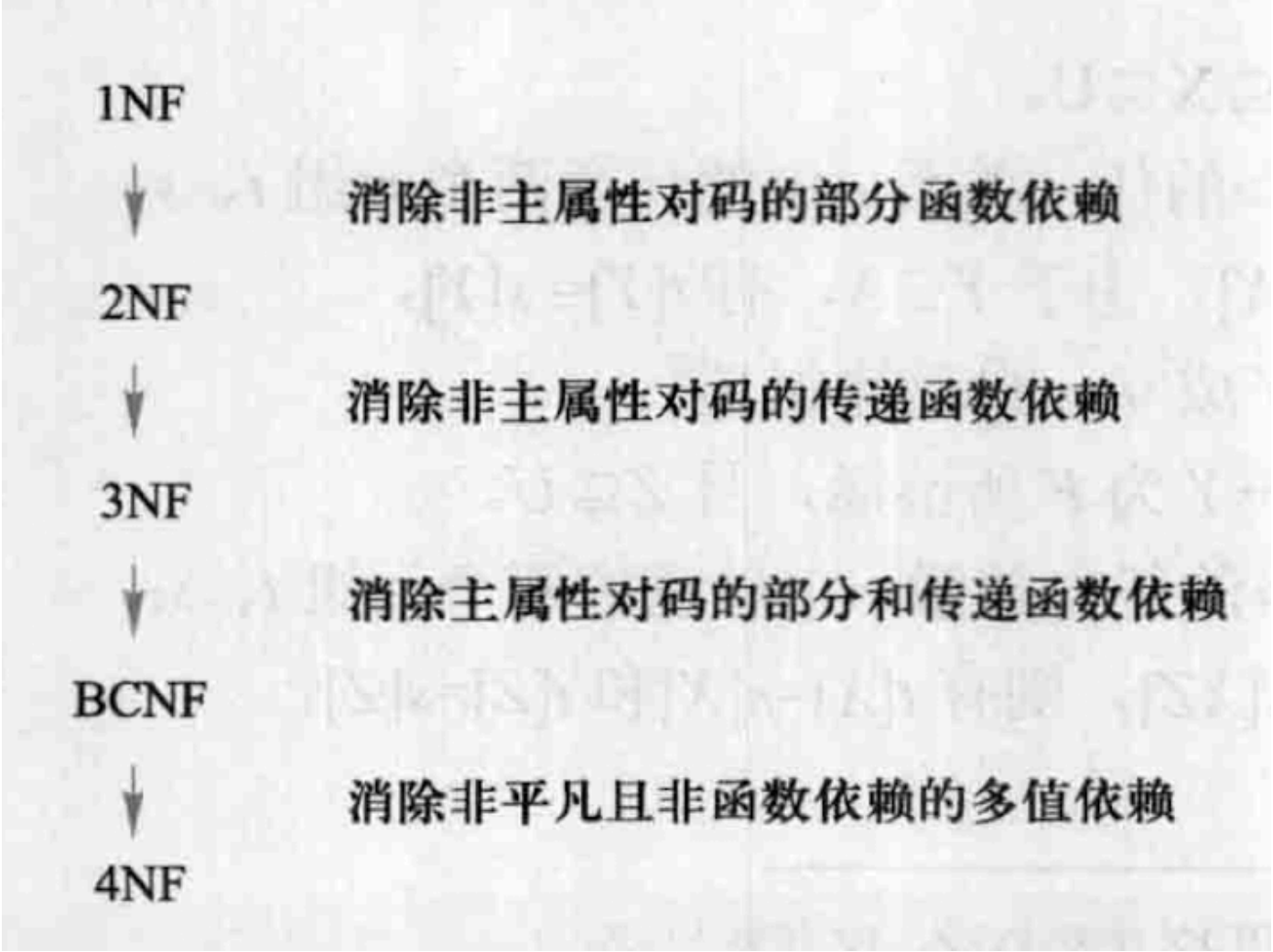
3NF：每一个非主属性既不传递依赖于码，也不部分依赖于码（主属性传递依赖于码仍是3NF）

3NF->BCNF：消除主属性对与码的传递函数依赖

BCNF：

- 所有非主属性对一个码都是完全函数依赖
- 所有主属性对每一个不包含它的码也是完全函数依赖
- 没有任何属性完全函数依赖于非码的任何一组属性

4NF：限制关系模式的属性之间不允许有非平凡且非函数依赖的多值依赖



模式分解

- 分解具有无损连接性
- 分解要保持函数依赖
- 分解既要保持函数依赖，又要具有无损连接性

数据库设计

基本步骤

- 需求分析
- 概念结构设计（E-R图）
- 逻辑结构设计
- 物理结构设计
- 数据库实施
- 数据库运行和维护

E-R图向关系模型的转换

- 一个1:1联系可以转换为一个独立的关系模式，也可以与任意一端对应的关系模式合并
- 一个1:n联系可以转换为一个独立的关系模式，也可以与n端对应的关系模式合并
- 一个m：n联系转换为一个关系模式
- 三个或三个以上实体间的一个多元联系可以转换为一个关系模式
- 具有相同码的关系模式可以合并