数据库原理与设计

课程总结

4

课程内容分五个部分

第一部分 数据库的基本概念(1、2)

第二部分 关系数据库(3、4、5)

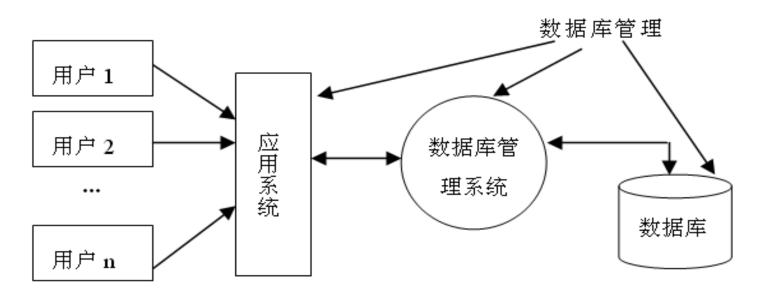
第三部分 数据控制(6、7、8、9)

第四部分 数据库设计(10, 11, 12)

第五部分 数据库新技术(17)

数据库的基本概念

什么是数据库、数据库管理系统、数据库系统?数据库管理系统的主要功能,数据库系统的组成

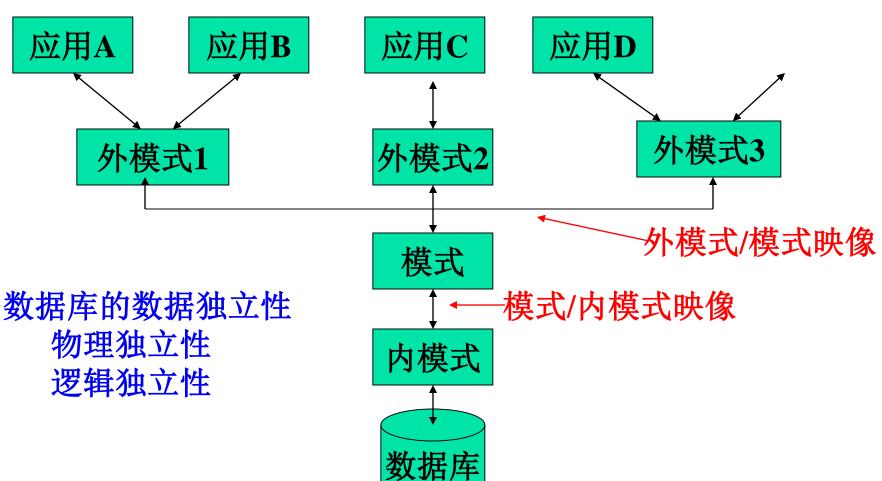


DBS=DB+DBMS+...

- 4
 - 数据管理的三个阶段:
 - 人工管理(20世纪40年代中-- 50年代中)
 - 文件系统(50年代末-- 60年代中)
 - 数据库系统(60年代末--现在)
 - 各个阶段数据和应用程序的关系
 - 各个阶段的特点
 - 文件系统管理数据有何缺点?
 - 数据库系统管理数据有何优点?



■ 体系结构——三级模式结构(二级映像)





数据模型

概念数据模型(概念模型)

数据模型

基本数据模型 (数据模型)

层次数据模型

网状数据模型

关系数据模型

面向对象模型

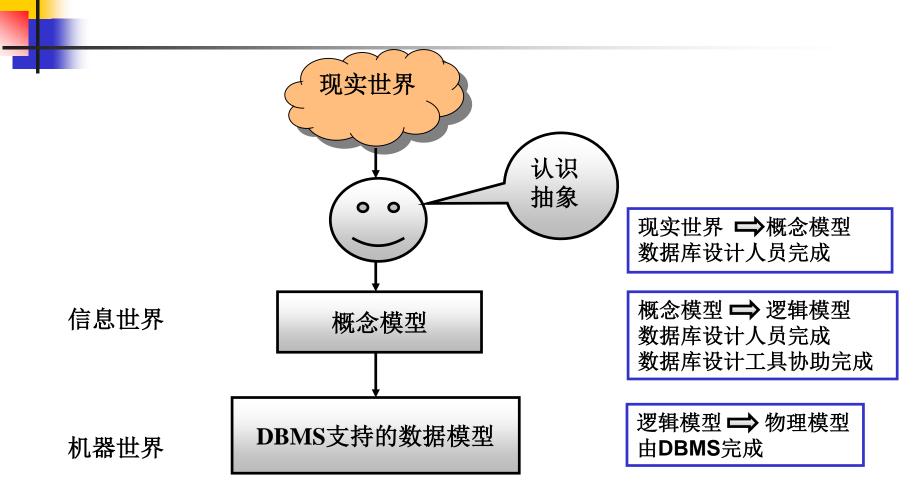
逻辑数据模型

数据结构

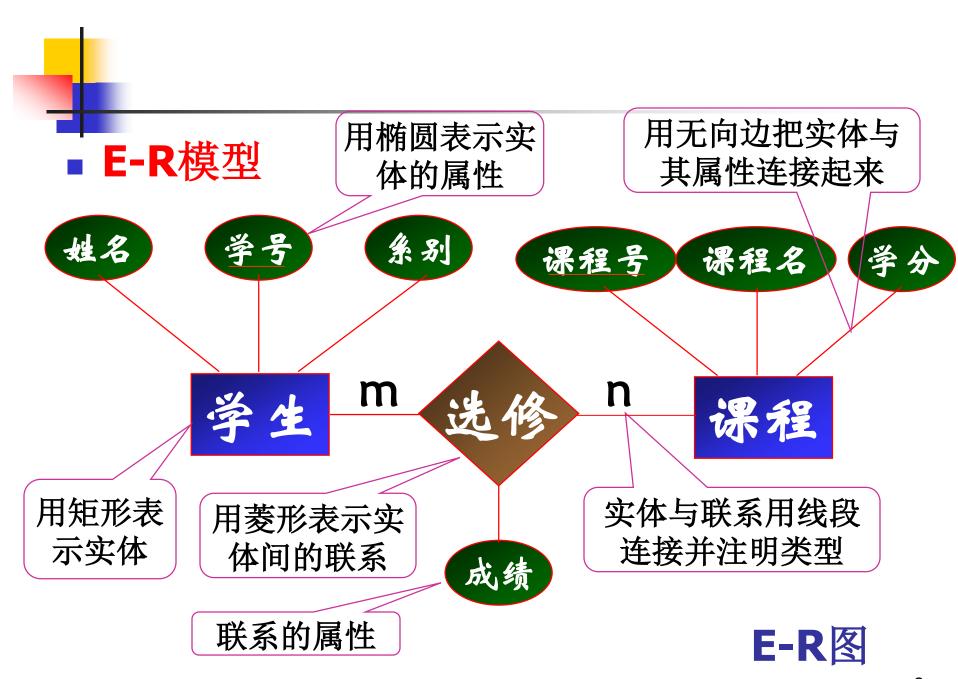
数据操作

数据完整性约束

物理数据模型

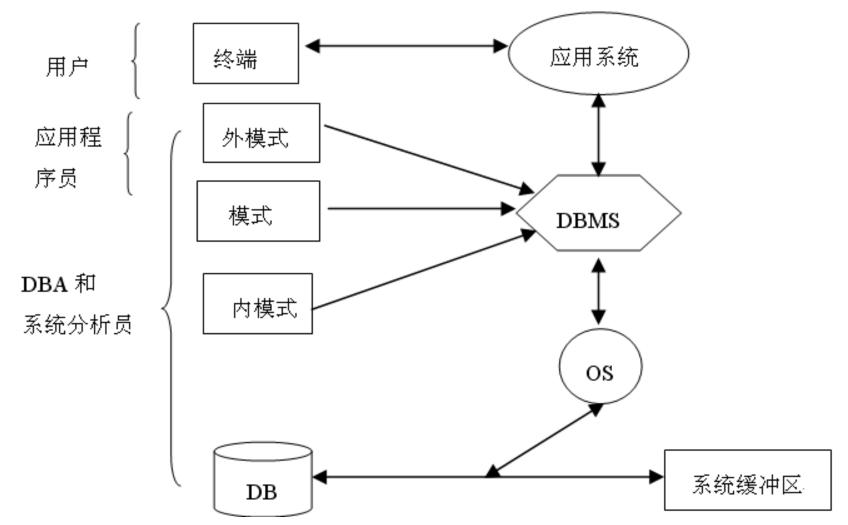


概念模型是现实世界到机器世界的一个中间层次。



- 4
 - 层次数据模型
 - 网状数据模型
 - 关系数据模型
 - 了解层次模型、网状模型、关系模型中对实体及实体间 联系的表示
 - 与层次和网状模型比较,关系模型的优点是什么

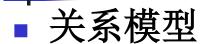
数据库系统的不同视图



二、关系数据库

- 1.关系模型的基本概念、关系运算
- 2.关系数据库标准语言SQL语言
- 3.关系数据库的查询优化

关系模型



- 数据结构: 关系(二维表)
- 数据操作
- 数据完整性约束
 - 实体完整性
 - ▶参照完整性
 - 用户自定义完整性
- 关系的基本概念:

关系、域、笛卡尔积、属性、元组、键、候选键、主键、 全键、联合键、主属性、非主属性、外键、参照关系、被 参照关系、空值



关系的基本性质

- 列是同质的,每一列中的值是同一类型的数据,来自同一个域。
- 不同的列可出自同一个域,每一列称为一个属性,不同的属性要给予不同的属性名。
- 列的次序是无关紧要的,列的次序可以任意交换
- 元组的每个分量都是不可分的数据项。
- 元组的次序是无关紧要的
- 各个元组都是不同的



■ 关系运算

关系代数

传统的集合运算

专门的关系运算

扩充的关系运算

关系运算

元组关系演算

关系演算

域关系演算

5种基本运算:并、差、选择、投影、笛卡儿积 用基本运算表示其他运算

R

A	В	С
a1	<i>b1</i>	5
a1	<i>b</i> 2	6
<i>a</i> 2	<i>b3</i>	8
<i>a</i> 2	<i>b4</i>	12

- 专门的关系运算一连接
 - 条件连接
 - 等值连接
 - ■自然连接
- 外连接
 - 左外连接 S
 - 右外连接
 - 完全外连接

В	E
<i>b1</i>	3
<i>b</i> 2	7
<i>b</i> 3	10
<i>b</i> 3	2
<i>b</i> 5	2

自然连接R⋈S

$m{A}$	В	C	$\boldsymbol{\mathit{E}}$
a_1	b_1	5	3
a_1	b_2	6	7
a_2	b_3	8	10
a_2	b_3	8	2

■ 用关系代数表达式表示查询。

$$\prod_{Sname} (\sigma_{Cpno='5'}(C \bowtie SC \bowtie S))$$

$$\prod_{Sname} (\sigma_{Cpno='5'}(C) \bowtie SC \bowtie \prod_{Sno, Sname} (S))$$

$$\prod_{Sname}(\prod_{Sno}(\sigma_{Cpno='5'}(C)\bowtie SC)\bowtie\prod_{Sno,\ Sname}(S))$$



关系数据库标准语言SQL

SQL数据库的体系结构 SQL 的组成

SQL 的数据更新

SQL模式的创建和撤消

SQL 提供的基本数据类型

SQL的数据定义〈基本表的创建、修改和撤消

视图的创建和撤消

关系数据库 SQL 语言

SQL 的数据查询(关于 SELECT 语句的详细介绍)

对视图的更新操作

嵌入式 SQL 的使用规定 嵌入式 SQL 的使用技术

- 4
 - SQL语言的特点。
 - SQL中的基本概念:基本表和视图。
 - 了解SQL的DDL语句,会用SQL语言定义基本表、定义视图。
 - 了解"唯一索引"和"聚集索引"的概念,会定义索引。
 - 掌握SQL的数据查询功能,各种查询条件的表示,聚集函数的使用,多表连接查询,外连接查询,单表和多表的分组查询,用IN 和 EXISTS于查询,蕴含关系的实现等。

- 4
 - 掌握数据更新操作语句:插入、删除和修改。
 - 理解视图消解的概念,掌握通过视图进行查询, 了解视图的优点。
 - SQL中不同对象上具有的权限,能够用GRANT语 句给不同用户授权,用REVOKE撤销。
 - 嵌入式**SQL**中,通信区、主变量、游标的概念和 作用。
 - 不用游标和使用游标的SELECT语句对数据的不同 处理。
 - 熟练掌握各个SQL语句的语法、各个关键词的含义,别写错用错。



- 了解查询处理的过程,优化的意义。
- 理解实现基本关系操作的各种算法,及其适用条件。
- 理解查询优化步骤,查询执行的代价。
- 理解查询优化的一般准则。
- 掌握基于关系代数表达式的优化算法。会画关系 代数表达式的语法树,能够优化语法树。



判断查询语句是否符合SQL语法规则

根据数据字典,检查查询语句中数据库对象的有效性、用户权限和完整性约束

将查询转化为系统内部的表示形式(建立在扩展关系代数基础上的语法树或语法图)

选择一个高效的查询处理策略

根据查询优化获得的策略,生成查询计划,检索数据,输出查询结果



SELECT Student.Sname

FROM Student, SC, Course

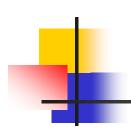
WHERE Student.Sno=SC.Sno and

Course.Cno = SC.Cno and

Student.Dept='计算机学院' and

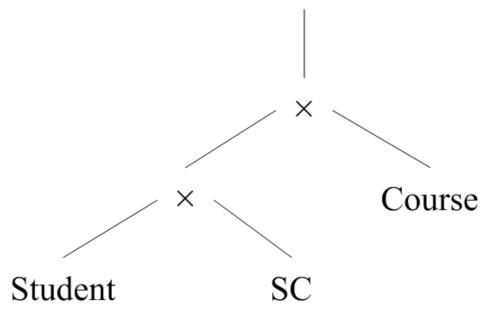
Course.Cname='DataBase'

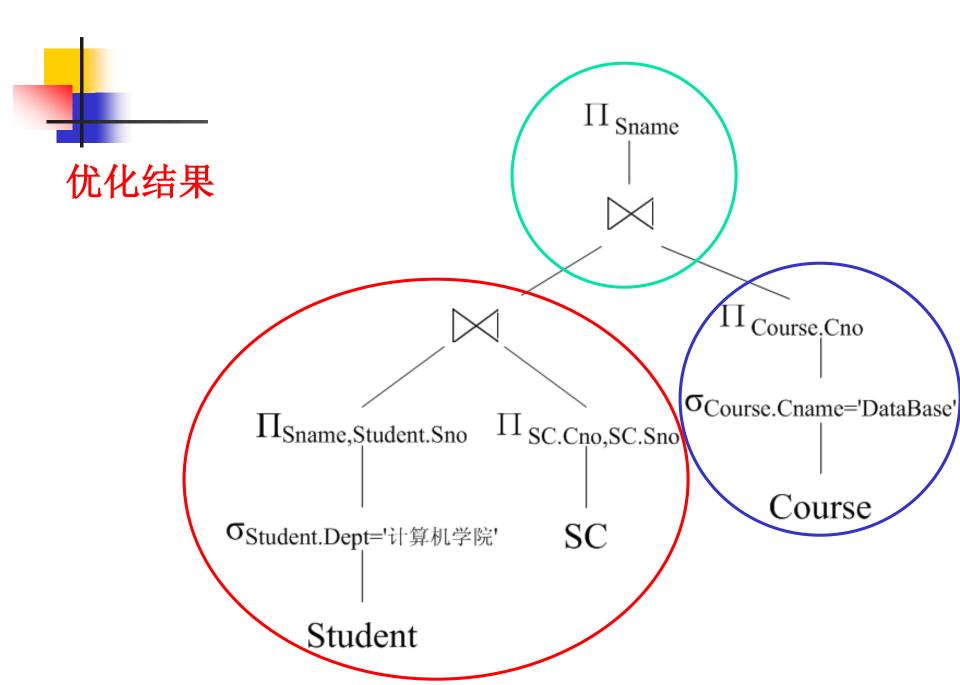
 Π_{Sname} ($\sigma_{Student.Sno=SC.Sno}$ \wedge Course.Cno=SC.Cno \wedge Student.Dept='计算机学院' \wedge Course.Cname='DataBase' ((Student×SC)×Course))



Π_{Sname}

oStudent.Sno=SC.Sno∧Course.Cno=SC.Cno∧Student.Dept='计算机学院'∧Course.Cname='DataBase'







三、数据库控制技术

数据库的恢复

数据库的并发控制

数据库保护

数据库的完整性控制

数据库的安全性控制

数据库安全性

- 了解数据库安全性的含义、安全级别的划分
- 数据库系统中所采取的安全性保护措施有哪些
- 了解存取控制中用户在不同对象上有哪些权限
- 掌握存取控制中自主存取控制方法
- 掌握存取控制的强制存取控制方法
- 如何利用视图机制实现安全性保护
- 什么是角色,角色对权限管理有何用处

- - 安全级别的划分:四组7个等级:
 - 从低到高的顺序为D,C(C1,C2),B(B1,B2,B3),A(A1)
 - 按系统可靠或可信程度逐渐增高
 - 各安全级别之间具有一种偏序向下兼容的关系,即较高安全性级别提供的安全保护要包含较低级别的所有保护要求,同时提供更多或更完善的保护能力



- 数据库安全性控制的常用技术和方法
 - 访问控制技术(用户标识与鉴别)
 - 存取控制技术
 - ■视图
 - ■数据加密
 - ■数据库审计



- 存取控制机制可分为:
 - 自主存取控制 (Discretionary Access Control, DAC)
 - 同一用户对于不同的数据对象有不同的存取权限
 - 不同的用户对同一对象也有不同的权限
 - 用户还可将其拥有的存取权限转授给其他用户
 - 比较灵活,C2级
 - 强制存取控制(Mandatory Access Control, MAC)
 - 每个数据对象都被标以一定的安全类别或安全级别
 - 每个用户也被标以一定的安全类别或安全级别
 - 对于任意一个对象,只有具有合法分类级别的用户 才可以存取
 - ■比较严格,**B1**级

数据库完整性

- 什么是数据库的完整性
- 关系模型必须满足的完整性约束条件
- 了解完整性约束条件作用的对象及不同对象上施加的完整性约束
- 用户定义的完整性有哪些,如何定义
- 如何通过触发器定义完整性
- 违约处理,具体措施。



- 关系数据库的完整性约束分为:
 - 实体完整性约束
 - 参照完整性约束
 - 其他 (用户定义的) 完整性约束
- 完整性约束条件的作用对象
 - 关系(关系约束)
 - 元组(元组约束)
 - 属性列(列级约束)



- 用户自定义完整性约束的方式
 - 默认值(Default)(非编程的方式)
 - 规则(Role)
 - 约束 (Check)
 - 触发器(Trigger)(编程方式)
 - 存储过程(Procedure)

• 破坏参照完整性的情况及其处理

被参照表publishers	参照表titles	违约处理
可能破坏参照完整性	插入不存在外键值 的元组	拒绝
可能破坏参照完整性	修改外键值	拒绝
删除元组	可能破坏参照完整 性	拒绝/级连删除/设置为空
修改主键值	可能破坏参照完整 性	拒绝/级连修改/设置为空

4

■ 触发器的定义--标准SQL

CREATE TRIGGER SC T AFTER UPDATE OF Grade ON SC REFERENCING **OLD row** AS OldTuple, **NEW row AS NewTuple** FOR EACH ROW WHEN (NewTuple.Grade >= 1.1*OldTuple.Grade) **INSERT INTO** SC_U(Sno, Cno, OldGrade, NewGrade) VALUES(OldTuple.Sno, OldTuple.Cno, OldTuple.Grade, NewTuple.Grade)

■ 触发器的定义--TSQL(SQL Server) **CREATE TRIGGER tri_grade** ON SC FOR UPDATE AS **IF UPDATE (Grade)** IF EXISTS (SELECT * FROM INSERTED JOIN **DELETED ON INSERTED.Sno** = **DELETED.Sno** WHERE INSERTED.Grade >= 60 AND **DELETED.Grade < 60) BEGIN** RAISERROR ('不许将不及格的成绩改为及格!') **ROLLBACK**

END

数据库恢复技术

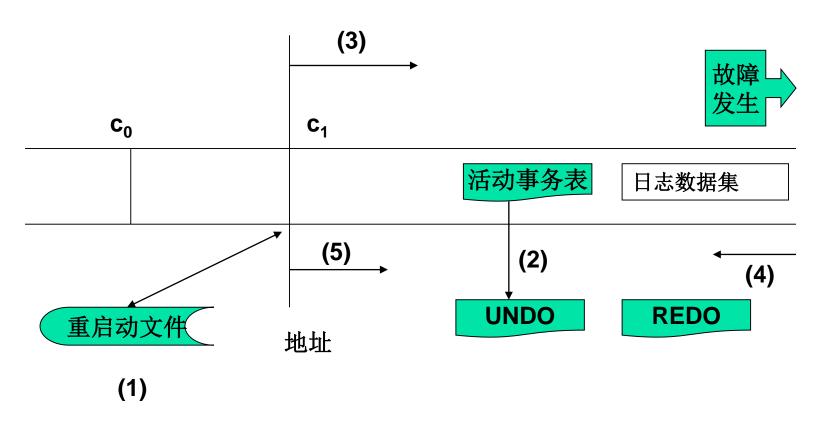
- 事务的基本概念
- 在SQL中如何定义一个事务
- 理解和掌握事务的四个性质(ACID)
 - 原子性 (Atomicity)
 - 一致性(Consistency)
 - 隔离性 (Isolation)
 - 持久性(Durability)

- 4
 - 数据库恢复的含义
 - 数据恢复的基本原理: 数据冗余
 - 建立数据冗余的基本方法
 - 数据转储
 - 登记日志文件
 - 了解数据转储的分类方法
 - 日志的内容,记日志的规则
 - 基于日志的恢复技术
 - Redo
 - Undo



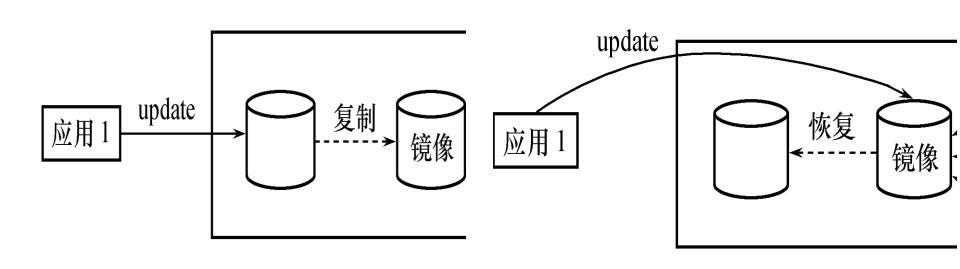
- 故障的种类,对不同故障所采取的恢复策略
 - ■事务故障的恢复
 - 系统故障的恢复
 - 介质故障的恢复

什么是检查点,检查点记录的内容,写检查点的 步骤,基于检查点的恢复技术及恢复过程。



利用日志和检查点进行事务恢复的过程

什么是数据库镜像,数据库镜像在数据库恢复中的作用。



并发控制

- 理解并发控制的含义
- 并发操作带来的数据不一致性
 - 丢失更新 (lost update)
 - 不可重复读(non-repeatable read)
 - 读"脏"数据(dirty read)
- 并发控制的相关概念
 - 调度、串行调度、可串行化调度
 - 冲突操作,调度的冲突等价
 - 调度的可串行性测试,判断一个调度是否可串行化



- 基于封锁的并发控制技术
 - 锁的基本类型: S锁和X锁的含义和作用
 - 锁的使用
- 死锁与活锁
 - 含义
 - 产生条件
 - 解决办法



- 封锁协议
 - 对数据对象加锁时约定的一些规则
 - 一级封锁协议
 - 二级封锁协议
 - 三级封锁协议
 - 两阶段封锁协议(2PL)
- 一到三级封锁协议的内容,解决的问题
- 两段锁封锁协议的内容、作用和特点

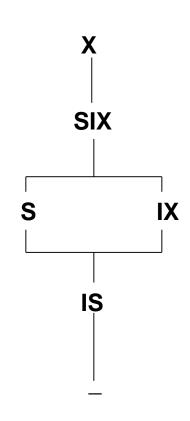


- 多粒度封锁
 - 封锁的粒度,即封锁的对象有哪些,
 - 理解封锁粒度与并发度及并发控制开销的关系
 - 封锁粒度越小,并发度越高,需要的锁越多,系统开销大。
 - 封锁粒度越大,并发度越低,需要的锁就少,系统开销小。
 - 多粒度封锁需要解决的问题
 - 意向锁的含义,锁的相容关系,封锁强度的比较。



增加了意向锁的相容矩阵

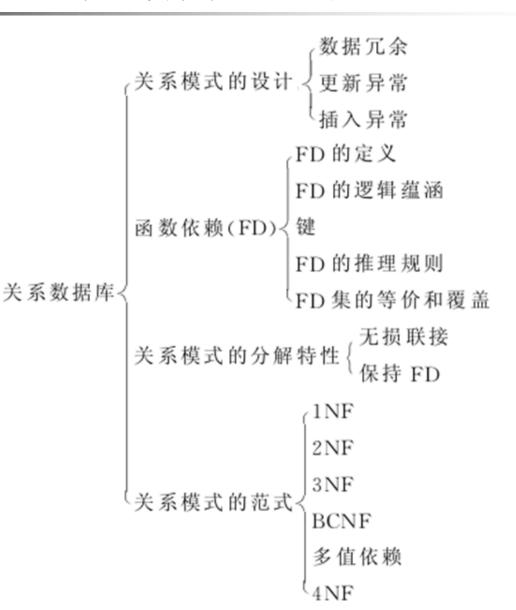
T ₂	S	Х	IS	IX	SIX	-
S	Y	N	Y	N	N	Y
X	N	N	N	N	N	Y
IS	Y	N	Y	Y	Y	Y
IX	N	N	Y	Y	N	Y
SIX	N	N	Y	N	N	Y
-	Y	Y	Y	Y	Y	Y



锁的强弱关系

四、关系数据库设计

- 设计理论
- 设计方法
- 数据库编程



设计理论

- 了解一个设计不好的关系数据库会出现的问题
 - 数据冗余
 - 插入异常
 - ■删除异常
 - 更新异常
- 函数依赖(FD)理解平凡和非平凡的FD、完全和部分FD、传递 FD的概念
- 理解键、主属性、非主属性的概念。



- 函数依赖的逻辑蕴含
- 掌握数据依赖的三条公理、三条推理规则,会用 公理证明三条推理规则以及相关的命题。
- FD的闭包、属性闭包的概念,会求属性闭包。

设关系模式R(U,F), U={A,B,C,D,E,G}, F={AB \rightarrow C,BC \rightarrow D,ACD \rightarrow B,D \rightarrow EG,BE \rightarrow C,CE \rightarrow AG}, 求 (BD)+

 $(BD)^+ = ABCDEG$

BD \rightarrow **ABCDEG**



- 模式分解
 - 数据被分解
 - 函数依赖也被分解
- 模式分解的等价性 (二个特性) 及简单判断。
 - 不损失任何信息,即:对于同样的数据,前后查询结果 应该一样。
 - 保持原来的函数依赖,即:不丢失数据间的约束(语义)。
 - 两者互相独立

- 理解和掌握不同范式的定义,范式间的关系,范式与数据依赖的关系。
 - 1NF
 - 」消除非主属性对候选键的部分函数依赖
 - 2NF
 - ↓消除非主属性对候选键的传递函数依赖
 - 3NF
 - ↓消除主属性对候选键的部分依赖和传递函数依赖
 - BCNF
 - ↓消除非平凡且决定因素不包含候选键的多值依赖
 - 4NF

- 能够正确判断一个关系模式属于几范式,能够将 关系模式规范化到3NF(模式分解算法)
 - 如果要求模式分解具有无损连接性和保持依赖性, 可以达到3NF,但不一定能够达到BCNF。
 - 如果要求模式分解仅具有无损连接性,则可以达到BCNF;如果规范化到BCNF,则不一定能保持函数依赖。
 - 规范化程度并非越高越好,在设计中究竟分解到哪一级取决于实际应用。

设计方法

- 数据库设计的基本步骤(阶段):
 - 需求分析
 - 概念设计
 - ■逻辑设计
 - ■物理设计
 - 数据库实施
 - 数据库运行和维护
- 掌握不同设计阶段的任务、要求和设计内容。
- 了解需求分析阶段的任务,数据流图、数据字典的内容和作用,以及需求分析方法。

- 4
 - 什么是数据库的概念结构,概念结构设计的基本 步骤
 - 在自底向上的设计方法中,视图集成需要解决的问题。
 - 能用E-R图正确表示概念模型。

- 什么是数据库的逻辑设计,数据库逻辑结构设计的内容,掌握E-R模型向关系模型的转换。
 - 一个实体型转换为一个关系模式
 - 一个m:n联系转换为一个关系模式
 - 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
 - 三个或三个以上实体间的一个多元联系转换为一个关系模式

- - 掌握数据库物理设计的内容、方法。
 - 1) 为关系模式选择存取方法(建立存取路径)
 - 2)设计关系、索引等数据库文件的物理存储结构
 - 数据库实施的工作内容
 - 数据库的运行和维护内容
 - 理解数据库重组织和重构造的概念和内容。

数据库编程

- Transact-SQL语法
- Transact-SQL游标
- Transact-SQL存储过程
- Transact-SQL用户定义函数
- Transact-SQL触发器

