考试要求：

客观题：选择10分、判断10分、填空

名词解释、问答：20分

关系操作25分：关系代数、SQL语言、编程

设计题15分：范式和数据库设计

**第一章 绪论**

1、数据库系统概述

A）数据：描述事物的符号记录

B）数据管理：收集、整理、存储、更新、检索、分析

C）DBMS：数据库管理系统

DBS：数据库系统

定义、构成、职责

数据模型：抽象、表示、处理现实世界数据和信息的工具

分类：概念模型（便于理解交流）、基本数据模型（易于实现）

概念模型：

实体：可观存在并相互区别的事物，具体或抽象

实体型：同类实体具有的共同特征和性质

码：唯一标识实体的属性子集

域：属性的取值范围

联系：实体内部联系（属性间）、实体之间的联系

基本数据模型：

组成要素：数据结构、操作、约束条件

分类：非关系型模型（层次、网状）、关系模型（二维表格）、面向对象模型、Nosql模型

关系模型完整性约束：实体完整性、参照完整性、用户自定义完整性

1、数据库系统结构

概念：

型：某一类数据的结构和属性的说明

值：型的具体赋值

模式：DB中全体数据的逻辑结构和特征，只涉及型，不涉及值

实例：模式的一个具体值

DDL（data definition language）数据定义语言

例如：CREATE {DATABASE | SCHEMA} db\_name（就是平常的SQL语句）

关系数据库

关系数据库结构形式化定义

关系的形式化定义：

域：一组具有相同数据类型的值的集合

笛卡尔积

N元组

分量

基数M：元组的个数

概念理解：

候选码：可以作为主码的码

主码：能够唯一地区分数据库中各个实体

主属性：候选码中出现的属性都是主属性

非主属性：不包含在候选码中的属性

关系的完整性约束

实体完整性：若A是关系R的主属性，则A非空，码的取值不能重复

参照完整性：A是R的非主属性，且与关系S主码对应，则A是R的外码，R成为参照关系，S称为被参照关系（R和S可以是同一个关系） 第二章ppt-p12

R在A上的值必须：或者空值，或者等于S中某个元组主码值（不能出现S中主码没有的值）

用户自定义完整性

关系操作：关系代数

操作类型：增、删、改、查（插入、删除、修改、查询）

运算的三大要素：运算对象、运算符、运算结果

集合运算

交、并、差、广义笛卡尔积（所有实体的笛卡尔积，一般不用）

专门关系运算

选择selection：

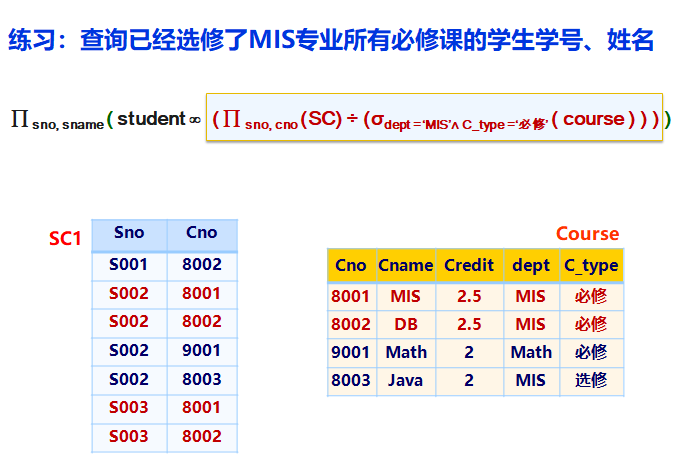
投影projection：，投影会去除重复值

连接join：一般连接、等值连接、自然连接（顺序可变）

除法devision：除运算的含义–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与S的除运算得到一个新的关系P(X)，P是R中满足下列条件的元组在X属性列上的投影：元组在X上分量值x的象集Yx包含S在Y上投影的集合

X、Y不变，Z变化，除法结果不变

被除关系不能扩展、除可以（Z属性可以增加）



关系代数扩展

赋值运算：R1<—R2 如Temp1 ← ∏sno, sname ( student )

广义投影运算：如∏title, price \* 10% + 15 (Book)

集函数运算：max、min、avg、sum、count

插入运算：R ← R ∪ Expression

删除运算：R ← R – Expression 如Book ← Book - σprice≥60(Book)

修改运算

四、五章 SQL语言

语言概述

三级模式结构

模式：对应基本表的定义和操纵

外模式：对应视图的定义和操纵

内模式：对应存储文件和数据索引的定义

基本DDL：定义表，约束，视图

基本表定义

基本表创建Create Table  [ database\_name. [schema\_name. ] ] table\_name

数据类型：字符串、数值型、时间日期型、货币类型、二进制串、位类型

唯一字段：identity（起始值，步长），自动赋值、禁止更新

实体完整性：

主键定义：Primary key，primary key(s1,s2,…)

参照完整性：

外键定义：[constraint <约束名>] [foreign key (列名) ] references <表名> (列名）

Check约束：[constraint <约束名>] check (<条件表达式>)

修改表alter：

如增加一列Alter Table student Add SID Char(16)

删除一列Alter Table student Drop Column sgender

增加约束ADD [CONSTRAINT <约束名>] <约束定义>

删除表：drop table

创建索引index

基本DML：增删改查（单表、连接、嵌套）

数据查询select…from…

Select后可跟表达式

列名+空格：为查询结果赋予别名

Select distinct 列名 from 表名：取消重复行

Where用法

模糊匹配：%与任意多个任意字符匹配，\_与一个任意字符匹配

换码字符\，让\后面的通配符失效，变普通字符，后面要加escape’\’

Select \* from course Where cname like ‘DB\\_APP\\_%’ escape ‘\’

结果：“DB\_APP\_........

空值匹配：列名 is NULL（不用=NULL）

结果排序：升序order by asc（默认）…，降序order by desc…，top N查询

集函数：不能独立出现在where子句中（子查询除外），一般和group by一起用

Count：count(distinct name)：统计不同的姓名分量

Sum，avg，max，min作用在数值字段上

Group by，having子句同时出现，对分组后的数据进行筛选

连接查询

内连接（常用）：不满足连接规则的被忽略，其中inner join为默认，可省略

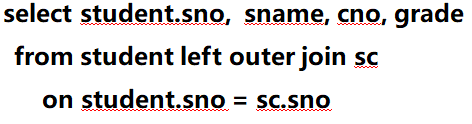
select sc.sno, sname, cno, grade

from （inner join） student, sc

where student.sno = sc.sno

外连接：左连接、右连接、全连接

左连接：保留左表，在右表找匹配项，没有的取NULL

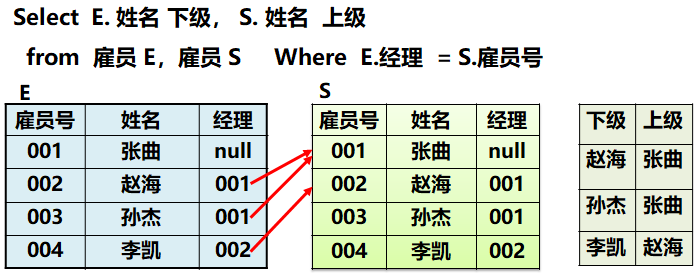


Righter join右连接同理

全连接full outer join：融合左右连接，两个表互相匹配，保留所有信息，匹配不上的NULL

交叉连接cross join（广义笛卡尔积，排列组合）：左表每一行与右表所有行组合

自连接（必须使用别名）：



复合条件连接

Eg：

查询选修全部课程的学生姓名：

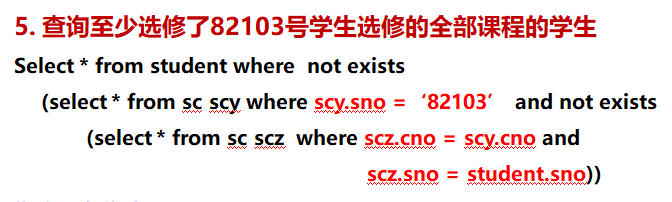
指定一个学生, 存在一门课程，SC中没有该学生选修该课程的记录

指定一个学生, 不存在这样的课程，则符合条件

Select sname from student where not exists

(select \* from course where not exists

(select \* from sc where cno = course.cno and sno = student.sno))



嵌套查询

带有比较运算的查询，比较运算符的对象必须只有一个值，如果存在多个则要用any（大于最小、小于最大）、all（大于或小于全体值）

内外相关子查询

集合查询

集合的并union（去除重复元组）、union all（保留重复元组）

集合的交intersect

查询既选a又选b课程的同学的学号

select sno from sc where cno = ‘a’

intersect

select sno from sc where cno = ‘b’

集合的差

查询选修了a课程，但未选修b课程的同学的学号

select sno from sc where cno = ‘a’

except

select sno from sc where cno = ‘b’

10、数据更新

插入数据 insert into sc(sno,cno,grade) values(‘001’,’A’,90)

修改数据update sc set grade=grade+10 where cno=’A’

删除数据delete from sc where cno=’B’ and sno=’001’

11、视图：称为虚表

安全管理：用户、角色、授权

数据库安全：保护数据库，防止非法使用造成的数据泄露、更改或破坏

威胁因素：非授权用户恶意存取、重要数据泄露、安全环境脆弱

TCSEC/TDI安全体系

数据库安全控制

用户身份鉴别：静（动）态口令、生物特征、智能卡

存取访问控制：自主存取控制DAC（C2级别）、强制存取控制MAC（B1级别）

审计（C2以上级别必备）

特殊数据库用户

DBO：数据库默认用户，拥有所有操作权限

guest：自动存在不能删

数据库角色

public：既是服务器角色也是数据库角色，用于使用系统过程，执行不需要权限的语句如print，不能删除

授权：grant

授予语句权限DDL：GRANT create table, view to user1, user2

授予对象权限DML：Grant select, update on student to u1, u2

回收权限语句权限：Revoke create view from user1, user2

回收对象权限：Revoke update(sno), select on student from u1, Myrole

禁止权限：

Deny select on student to u1,u2

Deny update, delete, insert on student to role1,role2

游标、触发器、存储过程

游标：在查询结果集上滚动游标，检查每行数据并完成相应处理

创建游标declare s\_cur cursor for select sname from student, sc

where student.sno = sc.sno and cno=‘A’

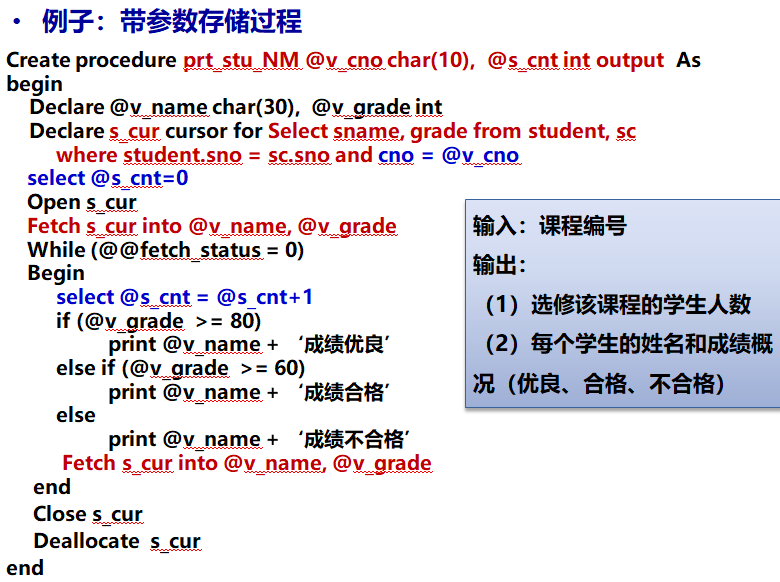
游标使用：open、fetch，游标状态@@fetch\_status：0成功，-1失败或此行不再结果集合中，-2该行不存在

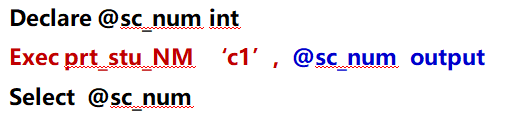
动态游标更新：update \*\*\* set sname =@v\_new\_name where current of s\_cur

关闭和释放：close关闭、deallocate释放

存储过程（类似于函数）：可带输入输出、调用方便、预编译性能高

声明变量declare @name char(30) ,@grade int



过程执行：

触发器

定义：与特定表关联，由特定时间触发的特殊存储过程

结构：事件（insert、update、delete）、条件、动作（执行的脚本）

Before：先执行触发器动作，再执行触发事件操作 after、instead of：同理

DML触发器：Create trigger <触发器名> on <表名> after insert as <sql语句>

Deleted、inserted用于在触发器脚本中读取被删除、插入的信息

关系理论

函数依赖和多值依赖：基本概念

函数依赖：不存在一个x对应两个y（单射），记做X—>Y，称作X函数决定Y或Y函数依赖于X

平凡函数依赖：X—>Y，但Y是X的子集，如（性别，年龄）—>年龄

非平凡函数依赖：X—>Y，Y不是X的子集

完全函数依赖：X—>Y，对X的任意真子集X’，X’不函数决定Y，则

部分函数依赖：存在X真子集，则，如

相互决定：X<—>Y

传递依赖：X—>Y非平凡，Y—>Z，但Y不函数依赖X，则Z对X传递依赖

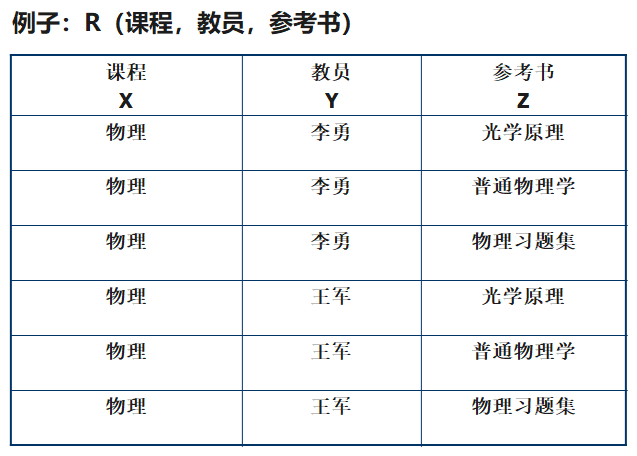
如 学号🡪系号，系号🡪系主任姓名；学号🡪系主任姓名

码

若，则X为候选码

多值依赖：多值依赖X🡪🡪Y成立，当且仅当对R(U)的任意一个关系r，给定一对（x,z）值，有一组y值与之对应，且这组y值仅仅取决于x值，而与z值无关

如：



平凡多值依赖：Z是空集

非平凡多值依赖：Z非空

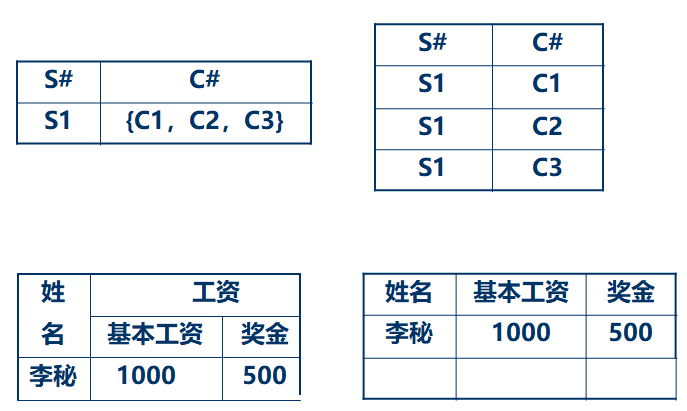
举例：（课程、教员、参考书），（仓库、责任人、设备）

函数依赖被看作特殊的多值依赖

多值依赖没有分解性

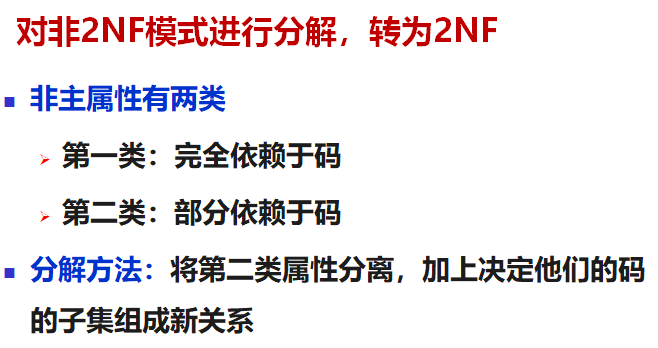
1NF、2NF、3NF、BCNF、4NF：范式判定和规范化

第一范式：每个分量都是不可再分的最小数据项



第二范式：满足第一范式条件下，每一个非码属性完全函数依赖与码（消除非码属性对码的部分依赖）

对非2NF模式进行分解，转为2NF



全码关系属于2NF（不存在非主属性）

单属性做码属于2NF（不存在部分依赖）

二目关系属于2NF（单码 or 全码）

R是1NF但不一定是2NF

第三范式：满足第一范式条件下，不存在这样的码X，属性组Y，及非码属性Z (Z不是Y的子集)，使得X 🡪 Y，（Y 不依赖X），Y 🡪 Z成立。（2NF基础上消除非码属性对码属性的传递依赖）

把非3NF分解成3NF

非3NF：X 🡪 Y，（Y 不依赖X），Y 🡪 Z成立

把Y，Z拿出来单独成表

全码关系属于3NF

二目关系属于3NF（不存在部分依赖和传递依赖）

R是3NF，则一定是2NF，反之不成立

修正的第三范式BCNF：满足第一范式条件下，若X🡪Y，且Y不是X的子集时，X必含有码，则 R∈BCNF（3NF基础上消除码属性对其他候选码的部分和传递依赖）

所有决定因素必须包含码

第四范式：对于R的每个非平凡（Z不为空）多值依赖，X都含有码

若X🡪🡪Y是非平凡的多值依赖，且X含有码，则该多值依赖是函数依赖

4NF中，不存在非平凡、非函数依赖的一般意义上的多值依赖

关系候选码确定

Armstrong公理系统

自反律：（姓名，性别）—>性别

增广律：学号—>姓名，则（学号，专业）—>（姓名，专业）

传递律：学号—>专业，专业—>负责人，则学号—>负责人

闭包

模式分解

候选码求解

L、N，R，LR

第七章 数据库设计

数据库设计阶段及任务

数据库设计：对于一个给定的应用环境，够高最优的数据库模式，建立数据库及其应用系统，是指能有效存储数据，满足用户的应用需求（信息需求、处理需求）

数据库设计各阶段及任务：

需求分析：了解用户数据及数据处理方面的需要（最困难、最耗时）

概念结构设计：根据应用需求，建立概念模型如ER图（关键）

逻辑结构设计：建立选定数据库产品所支持的数据逻辑模型

物理结构设计：根据逻辑模型，建立是和应用环境的物理结构

数据库实施：建立数据库，导入初始数据，试运行

数据库运行与维护：运行、评价、调整、修改

数据库概念结构设计

概念设计的目标：产生反映组织信息需求的数据库概念结构（概念模式）

设计方法

自顶向下、逐层细化

自底向上、逐步集成（属性综合法），被广泛采用

有里向外、逐步扩充

混合策略

数据抽象的方法：分类、聚集、概括

实体、实体的属性、实体的码

强实体、弱实体（矩形）

强实体：不依赖其他实体，可以独立存在，用单边矩阵表示

弱实体：必须依赖其他实体才能存在，用双边矩阵表示

简单属性、复合属性，单值属性、多值属性，基属性、派生属性（椭圆）

简单属性：单个元素构成的独立存在的属性

复合属性：多个元素构成的属性，其中每个元素都可以独立存在，用椭圆同实体相连

单值属性：每个实体在该属性上只有一个属性值

多值属性：双边椭圆，实体在该属性上可以有多个值，如联系电话、邮箱地址

主属性：下用下划线

基属性：不能由其他属性导出的属性

派生属性：可以由其他属性导出的属性，虚线椭圆，如是否超期可以由结束日期和还书日期导出

实体间的联系（菱形）

联系的度：参与联系的实体个数

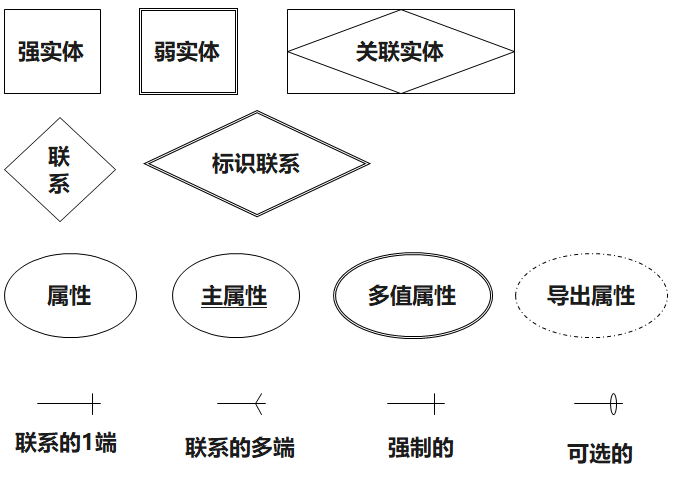
联系的基数约束

最大基数：对实体集A中的每个实体，B中最多有多少个实体与之对应，分为1:1，1:N，M:N联系

最小基数：对实体集A中的每个实体，B中最少有多少个实体与之对应，若为0则B是可选的，为1则B是强制的

Part-of联系：某个实体是另一个实体的一部分，如订单和订单明细，双边菱形表示标识联系（部分实体不能独立存在）

多元联系、关联实体（带矩形框的菱形）



超类、子类

超类：包含一个或多个实体子集的实体型，如高校学生（包含本科生、研究生）

子类

超类和子类的联系为1:1

ER图：通常用单线将超类与三角连接起来，再用单线将三角与各个子类连接起来

完全特化：超类中的每个实体必须属于某个子类的集合，如高校学生要么是本科生、要么是研究生，必为其一，用双线表示

部分特化：超类中某个实体可以不属于任何一个子类集合，用单线。

不相交约束：超类中的实体不能同时属于多个子类，三角中加×

可重叠约束：同理，不用特别描述

ER图设计

数据库逻辑结构设计

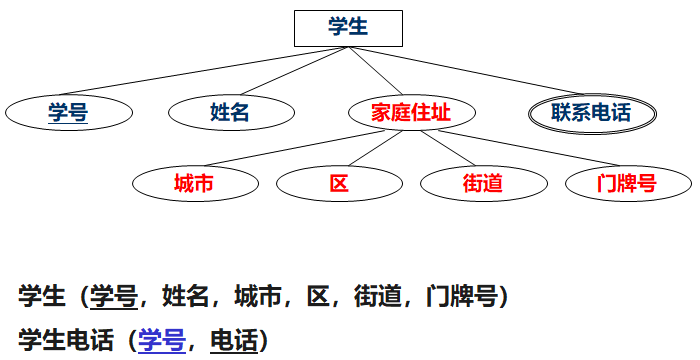
逻辑设计的任务：把ER图转换为具体的DBMS支持的逻辑结构

步骤：ER图——基本数据模型——具体DBMS支持的数据模型——优化的数据模型

ERD向关系模型转换

一个实体一个关系模式，实体属性和码就是关系的属性和码，复合属性直接纳入，多值属性另外创建新关系

一个多值属性对应一个新关系，包括实体型的码和多值属性，两者都是新关系的码，实体型的码是外码



弱实体：属主的码+弱实体的标识符构成关系的码，属主的码是外码，多值属性同常规实体类似

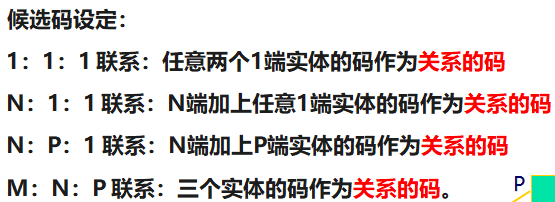
映射联系

1:1联系

1:N联系：N端实体的关系模式中加入1端实体的码和关系本身的属性

M:N联系：相关实体的码和联系本身的属性构成关系属性，相关实体的码联合起来作为关系的码

映射多元联系



映射超类和子类联系：可以将N个子类和1个超类分别转换成关系

数据库的设计（画E-R图，分解关系模型）

物理设计：为逻辑模型选取一个最适合应用要求的物理结构的过程

物理结构：数据库在物理设备上的存储结构和存取方法

物理设计基本概念：

存储访问

数据库文件组织：OS分配文件，由DBMS负责控制

文件组织形式：堆、顺序、hash、聚集

第八章 数据库恢复技术

一、事物基本概念和ACID特性

1、事务transaction：用户定义的一个数据库操作序列，这些操作要么全做要么不做，是一个不可分割的工作单位

a) 显式事务（多条）：begin transaction

b) 隐式事务（单条）：系统自动开始、提交、回滚

2、事务ACID特性

a) 原子性：逻辑工作单元，要么都做要么不做

b) 一致性：使数据库从一个一致性状态变到另一个一致性状态

c) 隔离性：并发执行的各个事务不能相互干扰

d) 持续性：事务一旦提交，对数据库的修改是永久的

二、冗余技术

三、故障恢复策略

1、系统故障

a) 内存缓冲区写盘

b) 系统故障时，数据状态分为已完成和未完成

c) 无检查点策略：

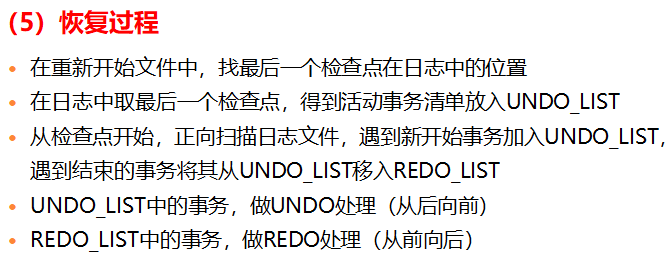
REDO队列：故障发生前已提交的事物

UNDO队列：故障发生前未提交的事物

d) 有检查点策略:

检查点：数据库正常运行的一个时间标志，检查点，系统强行将缓冲区内容写盘

恢复过程



2、介质故障回复策略

a) 介质故障：恢复最新的数据库备份副本、恢复日志文件副本（重做已完成事物、撤销未完成事物）

第九章 并发控制

1、锁的概念、分类、相容性

a) 封锁：事物T在对某个数据对象操作之前，先想系统申请对该数据对象枷锁，加锁后对该对象有了控制权，释放锁之前其他事物对该数据的使用将受到限制

b) 排它锁（写锁、X锁）：事物T对数据A加X锁，则其他事物不能对A加任何锁

c) 共享锁（读锁，S锁）：事物T对数据A加S锁，则其他事物只能对A加S锁，不能加X

相容性：



2、封锁协议、多粒度封锁、锁的相容性

a) 封锁协议：在运用X锁和S锁对数据对象加锁时，需要约定一些规则：何时申请X、S锁，持锁时间，何时释放

b) 一级封锁协议：事物T在修改数据R之前必须先对其加X锁直到事务结束释放（可防止数据丢失、修改，不能保证可重复读和不读脏数据）

c) 二级封锁协议：一级封锁协议+事物T在读数据R前先加S锁，读完释放（可以防止丢失修改和度脏数据，不能保证可重复读）

d) 三级封锁协议：一级封锁协议+事物T在读数据R前先加S锁，直到事务结束释放（可防止丢失修改、度脏数据和不可重复读）



e) 多粒度封锁

意向锁：对某节点加意向锁，说明该节点的下层节点正在被加锁



3、死锁的预防和解决方法

a) 死锁：同时处于等待状态的多个事务中，每个事务在它能够继续执行之前，都在等待对某个数据对象加锁，而这个数据对象已经被另一个等待的事物封锁

b) 死锁预防

一次封锁法：每个事物必须一次将所有要使用的数据全部加锁否则不能执行（降低并发度）

顺序封锁法：预先对数据对象规定一个封锁顺序，所有事物按顺序封锁（维护成本高，难以实现）

c) 检测死锁

超时法：一个事物等待超时，认为发生死锁（实现简单但可能误判）

等待图法

d) 消除死锁：回滚一个或多个事物，且使得代价最小

4、串行化调度的判定和二段锁协议

a) 可串行化调度（对并发事务而言）：多个事务的并发执行是正确的，当且仅当结果与按某一串行次序执行他们时结果相同

b) 冲突可串行化调度

c) 二段锁协议：读写前加锁，释放锁后不能再申请锁

事务遵守两段锁协议是可串行化调度的充分条件，而不是必要条件。

若并发事务都遵守两段锁协议，则对这些事务的任何并发调度策略都是可串行化的

若并发事务的一个调度是可串行化的，不一定所有事务都符合两段锁协议

第十章 数据仓库

1、数据仓库定义和特点

a) 数据仓库定义：一种面向主题的、集成的、不可更新的、随时间不断变化的数据集合，用于支持企业或组织的决策分析处理。本质上和数据库一样，是计算机内有组织、可共享的数据集合

b) 特点：面向主题、集成性、不可更新、时变性

2、数据组织

外部数据源

ETL概念（extract/transformation/load）：针对外部数据源进行的数据抽取、转换、加载，针对加载的数据，进行不同粒度的汇总和综合

1. 数据抽取：是否有用、是否冗余、是否和主题相关
   1. 从外部数据源抽取数据后，先放在缓冲区（抽取存储区）中，以便清洗
2. 数据清洗：处理外部数据的缺陷
3. 数据转换：文件格式差异、数据库平台差异、数据结构差异

不同粒度数据综合

3、多维数据和星型建模

1. 星型建模：一个事实表周围环绕多个维表，形成星型结构
   1. 事实表：描述实际发生的定量业务数据，包含关键字
   2. 维表：不同角度观察事实被称为维度，维表是对维度的描述
   3. 扩展：雪花模式、星片模式（混合使用雪花和星型）

4、OLAP操作（online analytical processing）联机分析处理

1. OLAP数据构造方式：ROLAP（relational OLAP），MOLAP（multi-demensional OLAP）
   1. 事实表：描述实际发生的定量业务数据，包含关键字
   2. 维表：不同角度观察事实被称为维度，维表是对维度的描述
   3. 扩展：雪花模式、星片模式（混合使用雪花和星型）
2. 多维OLAP操作：上卷、下钻、切片、切块、旋转

5、NoSQL数据库

概念演变和基本理念

1. 从反SQL（用新型非关系数据库取代关系数据库）——>Not only SQL，两者各有优劣无法取代
2. NoSQL数据库特性：易扩展、高可用、高性能、种类平台多

主要数据模型

1. 面向对象的数据模型
2. XML数据模型
3. RDF数据模型
4. NoSQL数据模型（Not Only SQL）

内存数据库概念和特点

1. 概念：将数据放在内存中，可以直接操作的数据库
2. 特点：高性能、直接访问内存、并发处理、与磁盘数据库同步具有数据恢复机制

Redis数据库的概念、特点、数据结构、适用性

1. 概念：Remote Dictionary Server
2. 特点
   1. 持久化：内存保存在磁盘，重启可再次加载使用
      1. 两种持久化方式：RDB（指定时间间隔生成快照）、AOF （记录所有操作，重新执行还原数据）
   2. 高性能：读写10W次/s
   3. 主从复制
   4. 多数据类型
   5. 扩展性强
   6. 客户端资源
3. 数据结构：string、hash、list、set、sorted set
4. 应用：社交网络好友管理、任务队列、应用排行榜、网站访问统计