

数据库原理及应用复习

- 一、 考试要求：
- 二、 客观题：选择 10 分、判断 10 分、填空
- 三、 名词解释、问答：20 分
- 四、 关系操作 25 分：关系代数、SQL 语言、编程
- 五、 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

- 一、 数据：描述事物的符号记录
 - a) 数据管理：收集、整理、存储、更新、检索、分析
- 二、 DBMS：数据库管理系统
 - a) DBS：数据库系统
- 三、 数据模型：抽象、表示、处理现实世界书局和信息的工具
 - a) 分类：概念模型（便于理解交流）、基本数据模型（易于实现）
 - b) 概念模型
 - i. 实体：客观存在并相互区别的事物，具体或者抽象
 - ii. 实体型：同类实体具有的共同特征和性质
 - iii. 码：唯一标志实体的属性集合
 - iv. 域：属性的取值范围
 - v. 联系：实体内部联系、**实体之间的联系**
 - c) 基本数据模型
 - i. 组成要素：数据结构、操作、约束条件
 - ii. 分类：关系型模型、非关系型模型、面向对象模型、NoSQL 模型
 - iii. 关系模型约束：实体完整性、参照完整性、用户自定义完整性

2. 数据库系统结构

- 一、 概念
 - a) 型：某一类数据的结构和属性说明
 - b) 值：型的具体赋值
 - c) 模式：DB 中全体数据的逻辑结构和特征，只涉及型不涉及值
 - d) 实例：模式的一个具体值
- 二、 DDL（data definition language）数据定义语言
 - a) 例如：CREATE {DATABASE | SCHEMA} db_name（就是平常的 SQL 语句）

第二章 关系数据库

1. 关系数据库的形式化定义

- 一、 关系的形式化定义

- a) 域：一组具有相同数据类型的值的集合
- b) 笛卡尔积
- c) N 元组
- d) 分量
- e) 基数 M：元组的个数

二、概念理解

- a) 候选码：可以作为主码的码
- b) 主码：能够唯一区分数据库中各个实体
- c) 主属性：候选码中出现的属性都是主属性
- d) 非主属性：不再候选码中属性

2.关系的完整性约束

- 一、**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
- 二、**参照完整性**：若 A 是关系 R 的非主属性，且与关系 S 的主码对应，则 A 是 R 的外码，R 称为参照关系，S 为被参照关系（R 和 S 可以是同一个关系），则 R 在 A 上的取值必须：要么为空、要么等于 S 中的某个元组主码
- 三、用户自定义完整性

3.关系操作：关系代数

- 一、操作类型：增删改查（插入、删除、修改、查询）
- 二、运算三大要素：运算对象、运算符、运算结果
- 三、集合运算：交、并、差、广义笛卡尔积
- 四、专门关系运算
 - a) 选择 selection
 - b) 投影projection
 - c) 连接join：一般连接、等值连接、自然连接
 - d) **除法division**：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**
- 五、关系代数扩展
 - a) 赋值运算
 - b) 广义投影运算
 - c) 集函数
 - d) 插入运算
 - e) 删除运算
 - f) 修改运算

第三、四、五章 SQL语言

1.语言概述

- 一、三级模式结构
 - a) 模式：对应基本表的定义和操纵
 - b) 外模式：对应视图的定义和操纵
 - c) 内模式：对应存储文件和数据索引的定义

2.基本DDL：定义表、约束、视图

- 一、基本表定义

- a) 基本表创建: Create Table [database_name. [schema_name.]] table_name
- b) 数据类型: 字符串、数值型、时间日期、货币类型、二进制、位类型
- c) 唯一字段: identity (起始值, 步长), 自动赋值, 禁止更新
- d) 实体完整性
 - i. 主键定义: primary key, primary key(s1,s2,...)主键组
- e) 参照完整性
 - i. 外键定义: [constraint <约束名>] [foreign key (列名)] references <表名> (列名)
- f) check约束
- g) 修改表alter
 - i. 增加一列: Alter Table student Add ID Char(16)
 - ii. 删除一列: Alter Table student Drop Column sgender
- h) 删除表: drop table

3.基本DML：增删改查（单表、连接、嵌套）

一、数据查询select...from...

- a) select后可以跟表达式
- b) 列名+空格, 为查询结果赋予别名
- c) select distinct 列名 from 表名: 取消重复行
- d) where用法
- e) 模糊匹配
 - i. %与任意多个字符匹配, _匹配任意一个字符
 - ii. 换码字符\, 让\后的通配符失效, 变为普通字符, 后面要加escape '\'

```
1 Select * from course Where  cname  like 'DB\_APP\_%' escape '\'
```

```
2 结果: DB_APP_.....
```

- f) **空值匹配**: 列名 is NULL (不用=NULL)
- g) 结果排序: 升序 order by xxx asc, 降序order by xxx desc, top N查询
- h) 集函数: 不能独立出现在where子句中, 一般和group by一起用
 - i. count
 - ii. sum, avg, max, min
- i) group by, having子句一般同时出现, 对分组后的数据进行筛选
- j) 连接查询
 - i. 内连接 (常用): 制定规则进行连接, 不满足连接规则则被忽略

```
1 select sc.sno, sname, cno, grade
2   from (inner join) student, sc
3  where student.sno = sc.sno
```

- ii. 外连接: 左连接、右连接、全连接
 - 1. 左连接: 保留左表, 在右表中找匹配项, 没有的取NULL

```
1 select student.sno,sname,cno,grade
2   from student left outer join sc
3   on student.sno=sc.sno
```

- 2. 右连接righter join: 同理
- 3. 全连接cross join: 融合左右连接, 两个表相互匹配, 左表每一行与右表所有行匹配 (**必须使用别名**)

Select E. 姓名 下级, S. 姓名 上级
from 雇员 E, 雇员 S Where E.经理 = S.雇员号

E			S			下级	上级
雇员号	姓名	经理	雇员号	姓名	经理		
001	张曲	null	001	张曲	null	赵海	张曲
002	赵海	001	002	赵海	001	孙杰	张曲
003	孙杰	001	003	孙杰	001	李凯	赵海
004	李凯	002	004	李凯	002		

二、复合查询、嵌套查询

a) 复合条件查询

```

1 #查询选修全部课程的学生姓名
2 #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
3 #指定一个学生, 不存在这样的课程, 则符合条件
4 Select sname from student where not exists
5     (select * from course where not exists
6         (select * from sc where cno = course.cno and sno = student.sno))

```

b) 嵌套查询

- i. 带有比较运算的查询: 比较运算符对象必须只有一个值, 否则用all或any
- ii. 内外相关的子查询

c) 集合查询

- i. 集合的差except

```

1 #选修了a但未选修b的学生学号
2 select sno from sc where cno = 'a'
3 except
4 select sno from sc where cno = 'b'

```

- ii. 集合的并union

```

1 select sno from sc where cno = 'a'
2 union
3 select sno from sc where cno = 'b'

```

- iii. 集合的交intersect

```

1 select sno from sc where cno = 'a'
2 intersect
3 select sno from sc where cno = 'b'

```

三、数据更新

a) 插入数据

```

1 insert into sc(sno,cno,grade) values('001','A',90)

```

b) 修改数据

```

1 update sc set grade=grade+10 where cno='A'

```

c) 删除数据

```

1 delete from sc where cno='B' and sno='001'

```

4.安全管理: 用户、角色、授权

一、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、破坏

二、数据库用户user：是登录名在数据库中的映射，创建用户时需要制定对应的登录名

a) 特殊数据库用户

i. DBO：默认用户，数据库创建后就存在，拥有在数据库中完成所有操作的权限，默认情况下sa登录名在各个数据库中对应的用户是DBO

ii. Guest：自动存在，不能删除

三、数据库角色：用户是具体到某一个账户个体，角色是某一类账户的集合。比如说仓库管理员，这是一个角色，他们管理着仓库；但是可能公司里有好几个仓库管理员，他们是不同的用户；

a) 特殊数据库角色

i. public：既是服务器角色也是数据库角色，用于使用系统过程，不能删除

四、授权grant

a) 授予、回收语句权限DDL

```
1 GRANT create table, view to user1, user2
2 Revoke create view from user1, user2
```

b) 授予、回收对象权限DML

```
1 Grant select, update on student to u1, u2
2 Revoke update(sno), select on student from u1, Myrole
```

c) 禁止权限

```
1 Deny select on student to u1,u2
2 Deny update, delete, insert on student to role1,role2
```

5.游标、触发器、存储过程

一、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

a) 创建游标

```
1 declare s_cur cursor for select sname from student, sc
2 where student.sno = sc.sno and cno='A'
```

b) 使用游标：open、fetch

i. 游标状态：@@fetch_status：0成功，-1失败或不在结果中，-2该行不存在

c) 动态游标更新

```
1 update *** set sname =@v_new_name where current of s_cur
```

d) 关闭和释放：close关闭、deallocate释放

二、存储过程（类似函数）：可带输入输出，调用方便，预编译性能高

a) 声明变量

```
1 declare @name char(30) ,@grade int
```

b) 带参数的存储过程

```
1 Create procedure prt_stu_NM @v_cno char(10), @s_cnt int output As
2 begin
3     Declare @v_name char(30), @v_grade int
4     Declare s_cur cursor for Select sname, grade from student, sc
5     where student.sno = sc.sno and cno = @v_cno
6     select @s_cnt=0
7     Open s_cur
```

```

8   Fetch s_cur into @v_name, @v_grade
9   While (@@fetch_status = 0)
10  Begin
11      select @s_cnt = @s_cnt+1
12      if (@v_grade >= 80)
13          print @v_name + '成绩优良'
14      else if (@v_grade >= 60)
15          print @v_name + '成绩合格'
16      else
17          print @v_name + '成绩不合格'
18      Fetch s_cur into @v_name, @v_grade
19  end
20  Close s_cur
21  Deallocate s_cur
22 end

```

c) 过程执行

```

1  Declare @sc_num int
2  Exec prt_stu_NM 'c1', @sc_num output
3  Select @sc_num

```

三、触发器

- a) 定义：与特定表关联，由特定时间出发的特殊存储过程
- b) 结构：时间、条件、动作
- c) 类型
 - i. before：先执行触发器动作，再执行触发事件
 - ii. after
 - iii. instead of
- d) DML触发器

```

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

```

第六章 关系理论

1.函数依赖和多值依赖：基本概念

一、函数依赖

- a) 定义：不存在一个x对应两个y（单射），记做 $x \rightarrow y$ ，称作x函数决定y或y函数依赖于x
- b) 分类
 - i. 平凡函数依赖： $X \rightarrow Y$ ，但Y是X的子集，如（性别，年龄） \rightarrow 年龄
 - ii. 非平凡函数依赖： $X \rightarrow Y$ ，Y不是X的子集
 - iii. 完全函数依赖： $X \rightarrow Y$ ，对X的任意真子集 X' ， X' 都不能函数决定Y，则Y函数依赖X
 - iv. 部分函数依赖：存在 X' ，能够函数决定Y，如（学号，班级） \rightarrow 姓名，学号 \rightarrow 姓名
 - v. 相互依赖
 - vi. 传递依赖： $X \rightarrow Y$ 非平凡， $Y \rightarrow Z$ ，但Y不函数依赖X，则Z对X传递依赖，如学号 \rightarrow 系号，系号 \rightarrow 班主任，学号 \rightarrow 班主任
- c) 码：若X完全函数决定Y，则X为候选码

二、多值依赖

- a) 定义：多值依赖 $X \twoheadrightarrow Y$ 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

例子：R（课程，教员，参考书）

课程 X	教员 Y	参考书 Z
物理	李勇	光学原理
物理	李勇	普通物理学
物理	李勇	物理习题集
物理	王军	光学原理
物理	王军	普通物理学
物理	王军	物理习题集

b) 分类

- 平凡多值依赖：Z是空集
- 非平凡多值依赖：Z非空
- 函数依赖被看作是特殊的多值依赖，多值依赖没有分解性

2.1NF, 2NF, 3NF, BCNF, 4NF

一、第一范式1NF

- a) 每个分量都是不可再分的最小数据项

S#	C#
S1	{C1, C2, C3}

S#	C#
S1	C1
S1	C2
S1	C3

姓名	工资	
	基本工资	奖金
李秘	1000	500

姓名	基本工资	奖金
李秘	1000	500

二、第二范式2NF

- a) 满足第一范式条件下，每一个非码属性完全函数依赖于码（在第一范式基础上消除非码属性对码的部分依赖）
- b) 对非2NF分解转为2NF

■ 非主属性有两类

- 第一类：完全依赖于码
- 第二类：部分依赖于码

■ 分解方法：将第二类属性分离，加上决定他们的码的子集组成新关系

c) 特殊

- 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
- R是2NF一定是1NF，反之不成立

三、第三范式3NF

- a) 满足第一范式条件下，不存在这样的码X，属性组Y，非码属性Z（Z不是Y的子集），使得X→Y（Y不决定X），Y→Z成立（在第二范式基础上消除非码属性对码属性的传

递依赖)

- b) 比较第三范式条件和传递依赖的条件
 - i. 传递依赖: $X \twoheadrightarrow Y$ (非平凡), $Y \rightarrow Z$, $X \rightarrow Z$
 - ii. 第三范式: $X \rightarrow Y$ (平凡或非平凡), $Y \rightarrow Z$, $X \rightarrow Z$ (不允许出现)
- c) 把非3NF分解成3NF
 - i. 非3NF: 存在 $X \twoheadrightarrow Y$, $Y \rightarrow Z$
 - ii. 把Y, Z拿出来单独成表
- d) 特殊
 - i. 全码关系属于3NF (不存在非码属性)
 - ii. 二目关系属于3NF (不存在部分依赖和传递依赖)
 - iii. R是3NF则一定是2NF, 反之不成立

四、修正的第三范式BCNF

- a) 满足第一范式的条件下, 若 $X \twoheadrightarrow Y$ 且非平凡, X必须含有码 (在3NF基础上消除码属性对其他候选码的部分和传递依赖)
- b) 提炼: 所有决定因素必须包含码 (候选码\组)
- c) 特殊
 - i. 二目关系属于BCNF

五、第四范式4NF

- a) 对于R的每个非平凡多值依赖 $X \twoheadrightarrow Y$, X都含有码
- b) 4NF不允许存在真正的非平凡多值依赖
- c) 特殊
 - i. 若 $X \twoheadrightarrow Y$ 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
 - ii. 4NF必是BCNF

3.关系候选码的确定

一、Armstrong公理系统

- a) 自反律: (姓名, 性别) \rightarrow 性别
- b) 增广律: 学号 \rightarrow 姓名, 则 (学号, 专业) \rightarrow (姓名, 专业)
- a) 传递律: 学号 \rightarrow 专业, 专业 \rightarrow 负责人, 则学号 \rightarrow 负责人

二、属性集的闭包

- a) $X_f = \{A | X \rightarrow A \text{ 能由F根据Armstrong公理导出的全部属性}\}$

- 1 例: 设: 属性集 $U = \{A, B, C\}$
- 2 函数依赖集 $F = \{A \rightarrow B, B \rightarrow C\}$
- 3 则: $AF^+ = \{A, B, C\}$; $BF^+ = \{B, C\}$
- 4 $CF^+ = \{C\}$

三、模式分解

- a) 无损分解: 分解后的关系通过自然连接能够恢复成原关系, 称为无损分解

四、候选码求解算法

- 1 (1) 将R的属性分为 L、R、N和 LR四类:
- 2 L类: 仅出现在函数依赖左部的属性; R类: 仅出现在函数依赖右部的属性
- 3 N类: 左右两边都不出现的属性; LR类: 左右两边都出现的属性。
- 4 (2) 令X代表L、N两类, Y代表LR类
- 5 (3) 求X闭包 X^+ , 若 X^+ 包含全部属性, 则X为唯一候选码, 转(6); 否则转(4)。
- 6 (4) 在Y中逐一取每个属性A, 求 $(XA)^+$, 若 $(XA)^+$ 包含全部属性, 则XA为候选码; 否则调换属性A, 重复过程(3), 直到试完所有Y中属性。
- 7 (5) 在Y中依次取2个、3个、...属性, 求X与它们的属性闭包, 直到其闭包包含R的全部属性。
- 8 (6) 停止, 输出结果。

第七章 数据库设计

1.数据库设计阶段及其任务

一、数据库设计

a) **定义**：对于一个给定的应用环境，构造最优的数据库模式，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）

b) 阶段及任务

- i. 需求分析（最困难、最耗时）：了解用户数据及数据处理方面的需要
- ii. 概念结构设计（关键）：根据应用需求，建立概念模型，如**ER图**
- iii. 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
- iv. 物理结构设计：根据逻辑模型，建立适合应用环境的物理结构
- v. 数据库实施：建立数据库，导入初始数据，试运行
- vi. 数据库运行和维护：运行、评价、调整、修改

2.数据库的概念结构设计

一、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）

二、设计方法：自顶向下，逐层细化；自底向上，逐步集成（广泛采用）；由里向外，逐步扩充

三、实体、属性、联系

a) 实体（矩形）

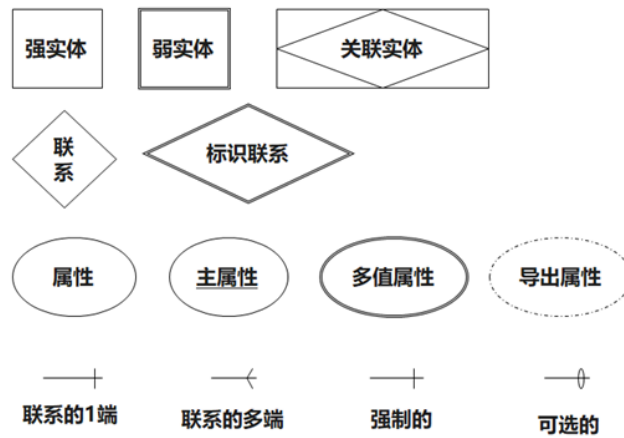
- i. 强实体：不依赖其他实体，可以独立存在，用**单边矩阵**表示
- ii. 弱实体：必须依赖其他实体才能存在，用**双边矩阵**表示

b) 属性（椭圆）

- i. 简单属性：单个元素构成的独立存在的属性
- ii. 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
- iii. 单值属性：每个实体在该属性上只有一个属性值
- iv. 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
- v. 主属性：下用下划线
- vi. 基属性：不能由其他属性导出的属性
- vii. 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

c) 实体间的联系（菱形）

- i. 联系的度：参与联系的实体个数
- ii. 联系的基数约束
 1. 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
 2. 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
- iii. 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，双边菱形标识联系（部分实体不能独立存在）
- iv. 多元联系、关联实体（带矩形框的菱形）



d) 超类、子类

- i. 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）
- ii. 表示：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
- iii. 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**
- iv. 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
- v. 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×
- vi. 可重叠约束

3.数据库的逻辑结构设计

一、逻辑设计的任务：把ER图转换为具体的DBMS支持的逻辑结构

二、ERD向关系模型转换

a) 映射联系

- i. 1：1联系：每个实体都构成一个关系，联系本身加上相关实体的码构成一个关系模式，任一相关实体的码作为该关系的码
- ii. 1：N联系：同上，N端实体码作为关系的码（或者在N端实体中加入1端实体的码和联系的属性，N端实体码作为关系的码）
- iii. M：N联系：同上，M、N端实体的码联合起来作为关系的码

b) 映射多元联系

- i. 1：1：1联系：任意两个1端实体的码作为关系的码
- ii. N：1：1联系：N端实体加上任意一个1端实体的码作为关系的码
- iii. N：N：P联系：N端加上P端实体的码作为关系的码
- iv. M：N：P联系：三个实体的码作为关系的码

第八章 数据库恢复技术

1.事务基本概念和ACID特性

一、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) 事务的特性：ACID

- i. 原子性：是数据库的逻辑工作单元，要么都做要么都不做
- ii. 一致性：使数据库从一个状态变到另一个状态
- iii. 隔离性：并发执行的各个事务不能互相干扰
- iv. 持续性：事务一旦提交，对数据库的修改是永久的

2.数据库恢复的实现技术：冗余技术

一、恢复：DBMS把数据库从错误状态恢复到某一已知正确状态的功能就是数据库恢复

a) 恢复原理：利用冗余恢复（数据转储、登记日志文件）

i. 数据转储：定期将数据库复制到另一个存储设备上

ii. 日志文件：记录事务对数据库的更新操作的文件，日志文件同样需要转储

3.故障恢复策略

一、事务故障

a) 解决方案：逆向扫描日志文件，寻找事务的更新操作，执行逆操作，继续逆向扫描，对所有更新操作执行逆操作，直到事务开始

二、系统故障

a) 无检查点策略

i. REDO（正向重做）队列：故障发生前已经提交的事务

ii. UNDO（逆向撤销）队列：故障发生前未提交的事务

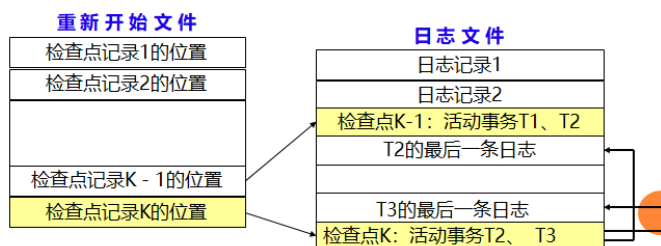
b) 有检查点策略

i. 检查点：检查点时刻，系统强行将缓冲区内容写盘

ii. 恢复策略：正向重做REDO_List，逆向撤销UNDO_List

(5) 恢复过程

- 在重新开始文件中，找最后一个检查点在日志中的位置
- 在日志中取最后一个检查点，得到活动事务清单放入UNDO_LIST
- 从检查点开始，正向扫描日志文件，遇到新开始事务加入UNDO_LIST
遇到结束的事务将其从UNDO_LIST移入REDO_LIST
- UNDO_LIST中的事务，做UNDO处理（从后向前）
- REDO_LIST中的事务，做REDO处理（从前向后）



三、介质故障

a) 恢复最新的数据库备份副本

b) 恢复备份日志文件

i. 重做已完成事务

ii. 撤销未完成事务

第九章 并发控制

1.锁的概念、分类

一、封锁：事物T在对某个数据对象操作之前，先向系统申请对该数据对象加锁，加锁后对该对象有控制权，释放锁之前其他事务对该数据的使用受到限制

a) 分类

i. 排它锁（写锁、X锁）：事务T对数据A加X锁，则其他事务不能对A加任何锁

ii. 共享锁（读锁、S锁）：事务T对数据A加S锁，则其他事务只能对A加S锁，不能加X锁

2.封锁协议、多粒度封锁、锁的相容性

一、封锁协议：运用X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

- a) 一级封锁协议：事务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（防止数据丢失修改，但不能保证不读脏数据和可重复读）
- b) 二级封锁协议：一级封锁协议+事务T在修改数据R前先加S锁，读完立刻释放（防止数据丢失修改和读脏数据，但不能保证可重复读）
- c) 三级封锁协议：一级封锁协议+事务T在修改数据R前先加S锁，事务完成再释放（防止数据丢失修改、读脏数据和不可重复读）

	X锁		S锁		一致性保证		
	操作结束释放	事务结束释放	操作结束释放	事务结束释放	不丢失修改	不读脏数据	可重复读
1级封锁协议		√			√		
2级封锁协议		√	√		√	√	
3级封锁协议		√		√	√	√	√

二、多粒度封锁

- a) 封锁粒度：封锁目标的大小
- b) 意向锁：对某个节点加意向锁，说明该节点的下层节点正在被加锁
 - i. IS锁、IX锁、SIX锁
 - ii. 锁的相容性矩阵

相容矩阵

	S	X	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

3.死锁的预防 and 解决方法

一、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务封锁

二、死锁的预防

- a) 一次封锁法：每个事务必须一次将所有要用的数据全部加锁，否则不能执行（降低并发度）
- b) 顺序封锁法：预先对数据对象规定一个封锁顺序，所有事务按顺序封锁（维护成本高、难实现）

三、死锁的诊断：超时法、等待图法

四、消除死锁：回滚一个或多个事务，使得代价最小

4.串行化调度的判定和二阶段锁协议

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

二、冲突操作和不可交换次序操作

- a) 不同事务间的冲突操作
 - i. 不同事务对同一数据的写写操作

- ii. 不同事务对统一数据的读写操作
 - b) 不可交换次序操作
 - i. 不同事务的冲突操作
 - ii. 同一事务的两个操作
- 三、二段锁协议（不能预防死锁，是可串行化调度的充分非必要条件）
- a) 读写前加锁
 - b) 释放锁后不能再申请锁
 - c) 注意：满足二段锁协议一定是可串行化调度，但可串行化调度不一定满足二段锁协议

第十章 数据仓库

1.数据仓库定义和特点

一、数据仓库

- a) 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

一、外部数据源

- a) ETL (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
 - i. 数据抽取、数据清洗、数据转换
- b) 不同粒度数据的综合

3.多维数据和星型建模

一、多维数据建模：以ER模型为基础，通过引入一些重要约束来直观、准确表达数据特征

- a) 星型建模：一个事实表周围环绕多个维表，形成星型结构
 - i. 事实表、维表
 - ii. 扩展：雪花模式、星片模式（混合使用雪花和星型）

4.OLAP操作

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

- a) OLAP数据构造方式
 - i. ROLAP (relational OLAP)
 - ii. MOLAP (multi-dimensional OLAP)
- b) 多维分析操作：上卷、下钻、切片、切块、旋转

5.NoSQL数据库

一、概念演变和基本理念

- a) 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）
- b) NoSQL数据库特性：易扩展、高可用、高性能、种类平台多

二、主要数据模型

- a) 面向对象的数据模型
- b) XML数据模型
- c) RDF数据模型
- d) NoSQL数据模型

三、内存数据库概念和特点

- a) 概念：将数据放在内存中，可以直接操作的数据库
- b) 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

四、Redis数据库

- a) 概念：Remote Dictionary Server
- b) 特点：持久化、高性能、主从复制、多数据类型、扩展性强
- c) 数据结构：string、hash、list、set
- d) 应用：社交网络好友管理、任务队列、应用排行榜