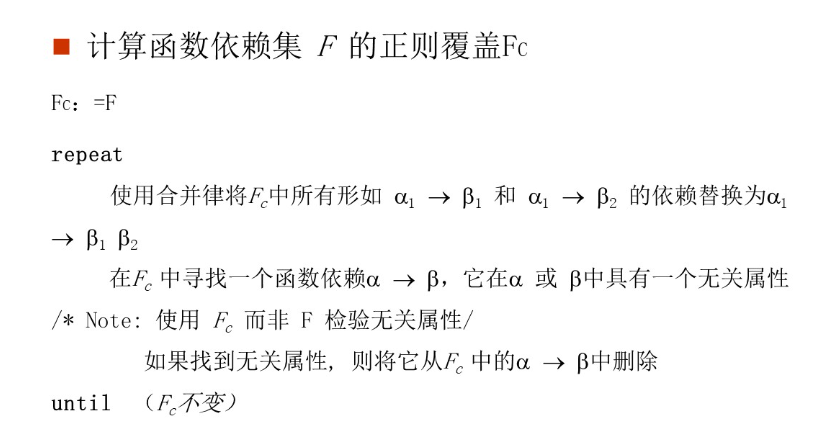
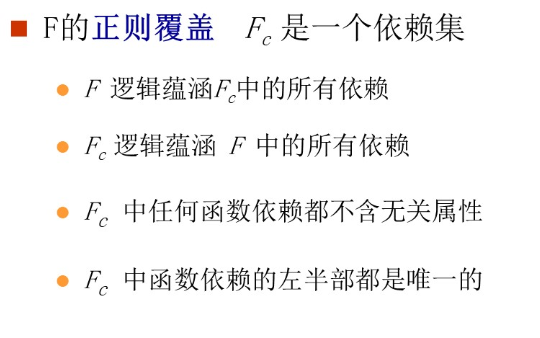
F的正则覆盖是一个与F相等的最小的函数依赖集，没有冗余的依赖，也没有**无关属性**

**正则覆盖算法**





无关属性的判定

1. 先算出Fc，在Fc中进行判定
2. 依赖的右侧属性，是否能同过左侧的闭包来不依靠这个依赖得到，如果能，则无关
3. 依赖的左侧属性，是否能在只有其中部分的情况下推出右边，如果能，剩余部分无关

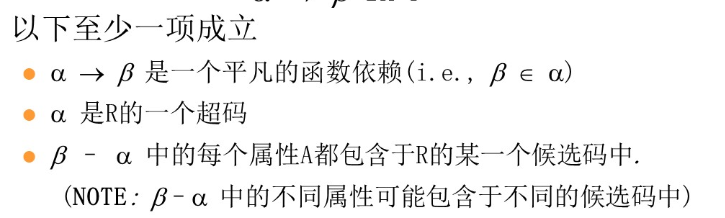
1NF

R的所有属性域都是原子的

2NF

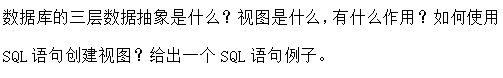


3NF



BCNF

每一个决定属性的因素都包含码



1.2.1.数据抽象

由于许多数据库系统的用户并未受过计算机专业训练，系统开发人员通过如下几个层次上的抽象来对用户屏蔽复杂性，以简化用户与系统的交互;

·物理层:最低层次的抽象，描述数据存储

·逻辑层:描述存储在数据库中的数据，以及数据之间的关系

·视图层:最高层次的抽象，只描述整个数据库的某部分数据

4.2.**视图**

视图提供一个对某些用户从视图中隐屈某些数据的机制。

任何不是逻辑模型的一部分，但作为虚关系对用户可见的关系称为视图。

**视图特点**

。虚表。是从一个或几个基本表（或视图)导出的关系

·只存放视图的定义，不会出现数据冗余

·基表中的数据发生变化，从视图中查询出的数据也随之改变

·查询时.视图名可以出现在任何关系名可以出现的地方

**视图vs派生查询(with)**

·视图存储在DB数据字典中。是数据库模式的一部分

- with定义的派生关系,仅在所属SQL有效。不属于DB模式

**视图(view) vs表(table)**

。视图和表都是关系，都可在SQL中直接应用。DB中存储表的模式定义和数据

-DB中只存储视图的定义，不存视图的数据·视图教据在使用视图时临时计算

·物化视图是提高计算的一种手段，结果等价于临时计算

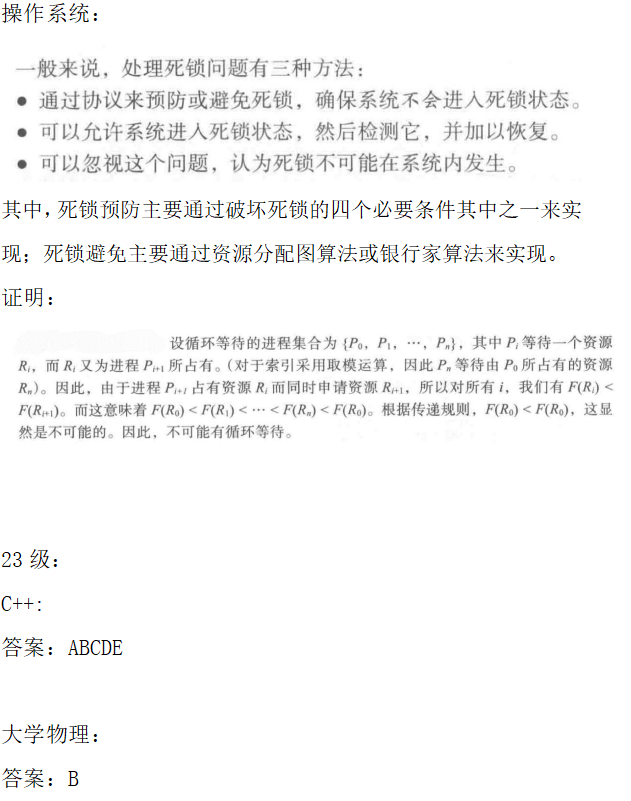
**视图的作用**

·对外模式的支持·安全性、方便性

·适应数据库共亨的需要

·对重构数据库在一定程度上提供了一定程度的逻辑独立性

在死锁处理部分,都有哪几类方式可以用于处理死锁?其中在死锁预防中怎样打破循环等待这一条件的?请证明。



第一章+第二章

1.四个基本概念

数据(Data)

数据是数据库中存储的基本对象

数据的定义：描述事物的符号记录

数据库(Database,简称 DB)

数据库是长期储存在计算机内、有组织的、可共享的大量数据集合

数据库的特征：数据按一定的数据模型组织、描述和储存；可为各种用

户共享；冗余度较小；数据独立性较高；易扩展

数据库管理系统(Database Management System，简称

DBMS)

数据库管理系统由一个互相关联的数据的集合和一组用以访问这些数据

的程序组成，是位于用户与操作系统之间的一层数据管理软件

DBMS 的目标：科学地组织和存储数据、高效方便地获取和维护数据

DBMS功能

（1）数据定义功能：DBMS提供数据描述语言（DDL），用户可通过它来定义数据对象。

（2）数据操纵功能：DBMS还提供数据操纵语言（DML），实现对数据库的基本操作：查询、插入、删除和修改。

（3）数据库的运行管理：这是DBMS运行时的核心部分，它包括并发控制，安全性检查，完整性约束条件的检查和执行，发生故障后的恢复等。

（4）数据库的建立和维护功能： 它包括数据库初始数据的输入及转换，数据库的转储与恢复，数据库的重组功能和性能的监视与分析功能等。

数据库系统(Database System，简称 DBS)

数据库系统是指在计算机系统中引入数据库后的系统构成。在不引起混

淆的情况下常常把数据库系统简称为数据库

数据库系统的构成：数据库管理系统（及其开发工具）、应用系统、数

据库管理员（和用户）

数据库完整性

定义：是指数据库中数据的正确性和相容性，由各种各样的完

整性约束来保证，是数据库设计的重要组成部分，完整性约束可以

通过 DBMS 或应用程序来实现

2.关系模型完整性

实体完整性和参照完整性：关系模型必须满足的完整性约束条件，称为

关系的两个不变性，由关系系统自动支持

用户定义的完整性：应用领域需要遵循的约束条件，体现了具体领域中

的语义约束，用户定义后，由关系系统自动支持

实体完整性(Entity Integrity)：关系的主码中的所有属性值不能为

空值

参照完整性(Referential Integrity)：如果关系 R 的外部码 Fk 与关

系 S 的主码 Pk 相对应(R 和 S 可以是同一个关系)，则 R 中的每一个元组的 Fk

值或者等于 S 中某个元组的 Pk 值，或者为空值

用户定义的完整性(User-defined Integrity)：用户针对具体的应用

环境定义的完整性约束条件

3.数据模型的三大要素是什么？

数据操作，数据结构，完整性约束

4.简述文件系统和数据库系统之间的区别与联系。

区别：

文件系统 数据库系统

程序和数据有一定的联系 程序和数据分离

用操作系统中的存取方法对数据进行管理 用DBMS对数据进行管理

实现以文件为单位的数据共享 实现以记录和字段为单位的数据共享

用文件将数据长期保存在外存上 用数据库统一存储数据

联系：均为数据组织的管理技术，均有数据管理软件管理数据，程序与数据之间用存取方法进行转换，数据库系统是基于文件系统的。

第三章

1.空值：

空值就是表示“无意义”，当实体在某个属性上没有值时设为null；或者表示“值未知”，即值存在，但目前没有获得该信息。

第四章

1.试述视图的作用？

（1）视图能够简化用户的操作。

（2）视图使用户能以多种角度看待同一数据。

（3）视图对重构数据库提供了一定程度的逻辑独立性。

（4）视图能够对机密数据提供安全保护。

2.说明视图与基本表的区别和联系。

视图是从一个或几个基本表导出的表，它与基本表不同，是一个虚表，数据库中只存放视图的定义，而不存放视图对应的数据，这些数据存放在原来的基本表中，当基本表中的数据发生变化，从视图中查询出的数据也就随之改变。视图一经定义就可以像基本表一样被查询、删除，也可以在一个视图之上再定义新的视图，但是对视图的更新操作有限制。

3.物化视图：

即实体化视图，它确实存放有物理数据。物化视图包含定义视图的查询时所选择的基表中的行。对物化视图的查询就是直接从该视图中取出行。

物化视图目的：

使用物化视图的目的是为了提高查询性能，是以空间换时间的一种有效手段，更少的物理读/写，更少的 cpu 时间，更快的响应速度；规模较大的报表适合使用物化视图来提高查询性能。

4.简述等值连接与自然连接的区别和联系。

连接运算符是“=”的连接运算称为等值连接。它是从关系R与S的广义笛卡尔积中选取A，B属性值相等的那些元组，即等值连接为：R∞S={trts| tr∈R∧ts∈S∧tr[A]=ts[B]}A=B

自然连接是一种特殊的等值连接，它要求两个关系中进行比较的分量必须是相同的属性组，并且在结果中把重复的属性列去掉。

第五章

1.动态SQL与嵌入式SQL：

动态sql：通用程序设计语言可以通过函数或者方法来连接数据库服务器并与之交互。利用动态sql可以在运行时以字符串形式构建sql查询，提交产寻，然后把结果存入程序变量中，每次一个元组。动态sql的sql组件允许程序在运行时构建和提交sql查询。

嵌入式sql：提供了另外一种使程序与数据库服务器交互的手段。嵌入式SQL必须在编译时全部确定，并交给预处理器。预处理程序提交sql语句到数据库系统进行变异和优化，然后将sql语句替换成相应代码和函数，最后调用程序语言的编译器进行编译。

第六章

1.关系中的元组有先后顺序吗？为什么？

答：没有。关系是元组的集合，而集合中的元素是没有顺序的，因此关系中的元组也就没有先后顺序。

第七章

1.弱实体集定义：一个实体集的所有属性都不足以形成主码，弱实体集必须依赖于强实体集，称作它的标识实体集或者属主实体集。

2． 数据库设计分哪几个阶段？

数据库设计分以下六个阶段：

（1）需求分析

（2）概念结构设计

（3）逻辑结构设计

（4）物理结构设计

（5）数据库实施

（6）数据库运行和维护

第十章

1.常见的存储介质

①高速缓冲存储器(Cache)

②主存储器(Main memory)

③快闪存储器 (Flash memory)

④光学存储器(CD-ROM/DVD)

⑤磁盘

⑥磁带存储器

注：①②都为易失性存储介质，cache 和主存配合工作。

③④⑤⑥都为非易失性存储介质，磁盘是主要的辅存，磁带主要用来脱机

备份。

2.存储层次

①基本存储

访问速度最快的存储介质，但是易失(cache, 主存)，可以直接被 cpu 访问。

②辅助存储

层次结构中基本存储介质的下一层介质, 非易失, 访问速度较快。如：闪存,

磁盘

③第三级存储

层次结构中最底层的介质, 非易失，访问速度慢。如：磁带, 光学存储器

3.RAID技术：

RAID是一种把多块独立的物理硬盘按不同的方式组合起来形成一个硬盘组，从而提供比单个硬盘更高的存储性能和提供数据备份技术。组成磁盘阵列的不同方式成为RAID级别。

通过引入冗余来提高可靠性,存储正常情况下不需要的额外信息，以便在数据故障时修复数据。通过并行来提高性能，将数据按位级拆分或块级拆分拆分到多张磁盘，从而达到并行读取数据，提高传输速率。

第十一章

1.简述何时需要建立聚簇索引，以及它的优缺点。

如果经常需要按照某搜索码（在文件中查找记录的属性或属性集）做查找或者排序操作，可在该搜索码上建立聚簇索引，使得数据文件的记录顺序按照搜索码的顺序排序。

优点：利用搜索码上的聚簇索引，可以基于搜索码属性做高效的查找和排序操作；聚簇索引可以采用稀疏方式存储，节省索引的存储成本。

缺点：聚簇索引要求数据文件的记录顺序按照搜索码的顺序排序，因而在一个关系上只能建立一个聚簇索引。如果基于非聚簇索引（辅助索引）的搜索码进行查找操作，成本会相对高一些。

2.索引：

索引是一个单独的、物理的数据库结构，它是某个表中一列或若干列值的集合和相应的指向表中物理标识这些值的数据页的逻辑指针清单。 索引提供指向存储在表的指定列中的数据值的指针，然后根据指定的排序顺序对这些指针排序。

索引优点：

第一， 通过创建唯一性索引，可以保证数据库表中每一行数据的唯一性。

第二， 可以大大加快数据的检索速度，这也是创建索引的最主要的原因。

第三， 可以加速表和表之间的连接，特别是在实现数据的参考完整性方面特别有意义。

第四， 在使用分组和排序子句进行数据检索时，同样可以显著减少查询中分组和排序的时间。

第五， 通过使用索引，可以在查询的过程中，使用优化隐藏器，提高系统的性能。索引

索引类型：

顺序与散列：

顺序索引：基于值的顺序排序

散列索引：基于将值平均分布到若干散列桶中。一个值所属的散列桶是由一个散列函数决定的。

聚集索引与非聚集索引：

聚集索引：数据表的物理顺序与索引的制定顺序相同的索引。

非聚集索引：数据表的物理顺序与索引指定的顺序不同的索引。

稠密索引与稀疏索引：

稠密索引：每个搜索码的值都有一个索引项

稀疏索引：只为搜索码的某些值建立索引项，只有当关系按搜索码排列顺序存储时才能使用稀疏索引，也就是只有当索引是聚集索引时才能使用稀疏索引

3.索引顺序文件的缺点及B+树索引优缺点

(1) 索引顺序文件的缺点（主要在结构方面）

①随着文件的增大，由于许多溢出块会被创建，索引查找性能和数据顺序扫描性能都会下降。

②插入和删除时，频繁重组整个文件。

(2) B+树索引文件的优点

①在数据插入和删除时，能够通过小的自动调整来保持平衡。

②不需要重组文件来维持性能。

(3) B+树索引文件的缺点

增加文件插入和删除的时间开销，同时会增加空间开销。这是因为插入和删除可能会引发 B+树的调整，并且树形结构比线性存储需要更大的空间。

总结：相较于顺序存储时频繁地调整整个文件，B+树只需要在局部调整以维持平衡，并且 B+树查找每个叶节点的性能非常稳定。所以我们可以接受一定的时间和空间开销，使用这种数据结构。

第十四章

1.事务的定义及其特性

答：事务是由一系列操作序列构成的程序执行单元，这些操作要么都做，要么都不做，是一个不可分割的工作单位。

事务的ACID特性：

原子性(Atomicity) 事务中包含的所有操作要么全做，要么全不做

一致性(Consistency) 事务的隔离执行必须保证数据库的一致性

隔离性(Isolation) 系统必须保证事务不受其它并发执行事务的影响

持久性(Durability) 一个事务一旦提交之后，它对数据库的影响必须是永久的。

2.

3、试叙述“串行调度”与“可串行化调度”的区别。

如果多个事务依次执行，则称事务串行调度。

如果利用分时的方法，同时处理多个事务，则称为事务的并发调度。如果一个并发调度的结果与某一串行调度执行结果等价，则称这个并发调度是可串行化调度。

第十五章

1.两阶段封锁：

可以保证可串行性。在对任何数据进行读、写操作之前，事务首先要获得对该数据的封锁。两段锁协议要求每个事物分成两个阶段提出加锁和解锁申请：

增长阶段：事物可以获得封锁，不能释放锁；

缩减阶段：事物可以释放锁，但不能获得新锁。

2.证明为什么两阶段封锁协议能够保证事务集合的可串行化。

采用反证法，假定两阶段封锁协议不能保证可串行化，则存在一个遵循两阶段封锁协议的事务集合T0, T1, …, Tn-1产生了一个非串行的调度，那么在其优先图中会有一个环。不失一般性，假定优先图中的环是T0 T1 T2… Tn-1 T0，让ai对应事务Ti获得到最后一个锁的时间（即事务Ti的封锁点）。因而对于所有的事务Ti<Tj，都有ai < aj，对于这个环，我们就有 a0 a1 a2 … an-1 a0，存在矛盾，因此假设不成立。

所以，两阶段封锁协议能够保证事务集合的可串行化。

3.简述三级封锁协议的内容以及不同级别的封锁协议能解决哪些数据不一致性问题?

（1）一级封锁协议是：事务T在修改数据R之前必须先对其加X锁，直到事务结束才释放。一级封锁协议能够解决“丢失修改”问题。

（2）二级封锁协议是：一级封锁协议加上事务在读取数据R之前必须先对其加S锁 ，读完后即可释放S锁。二级封锁不仅可以解决“丢失修改”问题，而且可以解决读“脏“数据问题。

（3）三级封锁协议是：一级封锁协议加上事务在读取数据R之前必须先对其加S锁，直到事务结束才释放。三级封锁协议不仅解决了“丢失修改”、读“脏”数据问题，而且进一步解决了“不可重复读”问题。

4.简述数据库并发操作通常会带来哪些问题。（智库p71）

丢失修改（lost update）不可重复读（non-repeatable read）读“脏”数据（dirty read）

第十六章

1.登记日志文件时必须遵循什么原则？

（1）登记的次序严格按并发事务执行的时间次序。

（2）必须先写日志文件，后写数据库。

2．数据库系统的故障有哪些类型？

答：故障主要有下面三种类型：

（1）事务故障（2）系统故障（3）介质故障

3.简述系统故障时的数据库恢复策略。

（1）正像扫描日志文件，找出在故障发生前已经提交的事务，将其事务标识记入REDO队列。同时找出故障发生时尚未完成的事务，将其事务标识记入UNDO队列。（2）对UNDO队列中的各个事务进行撤销处理。（3）对REDO队列中的各个事务进行重做处理。

请解释数据和程序在逻辑上和物理上的独立性

逻辑独立性

逻辑独立性是数据库系统中应用程序与数据库逻辑结构之间的独立性。具体来说，当数据库的逻辑结构（如表结构、字段类型等）发生变化时，如新增关系、字段或修改字段类型，用户的应用程序可以保持不变。这主要得益于数据库的三级模式结构中的外模式/模式映像，它确保了用户视图（外模式）的稳定，即使数据库逻辑结构（模式）有所调整，用户程序也无需修改。这种独立性大大简化了应用程序的维护和修改过程。

物理独立性

物理独立性则是指应用程序与数据库物理存储结构之间的独立性。当数据库的物理存储结构（如存储设备、存储格式等）发生变化时，应用程序仍然可以正常运行，无需修改。这得益于模式/内模式映像，它确保了数据库的逻辑结构（模式）与物理存储结构（内模式）之间的转换。物理独立性的实现使得数据库的物理存储结构可以灵活调整，而不会影响上层的应用程序。

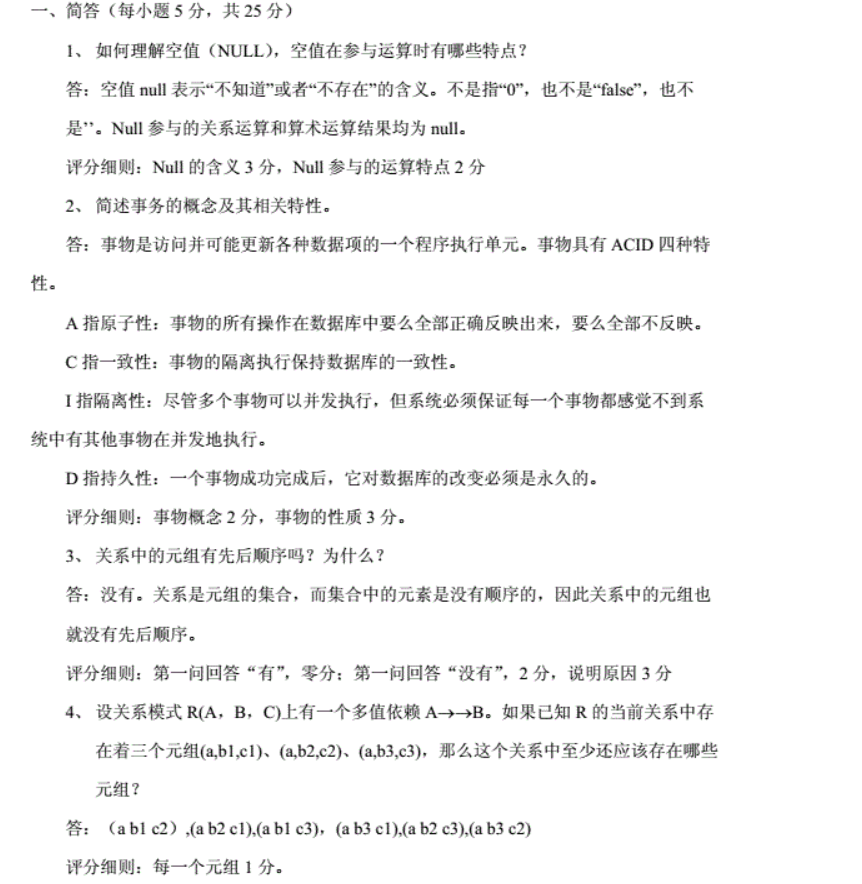
物理独立性是指用户的应用程序与存储在磁盘上的数据库中数据是相互独立的。即，数据在磁盘上怎样存储由DBMS(DataBase Management System 数据库管理系统)管理，用户程序不需要了解，应用程序要处理的只是数据的逻辑结构，这样当数据的物理存储改变了，应用程序不用改变。

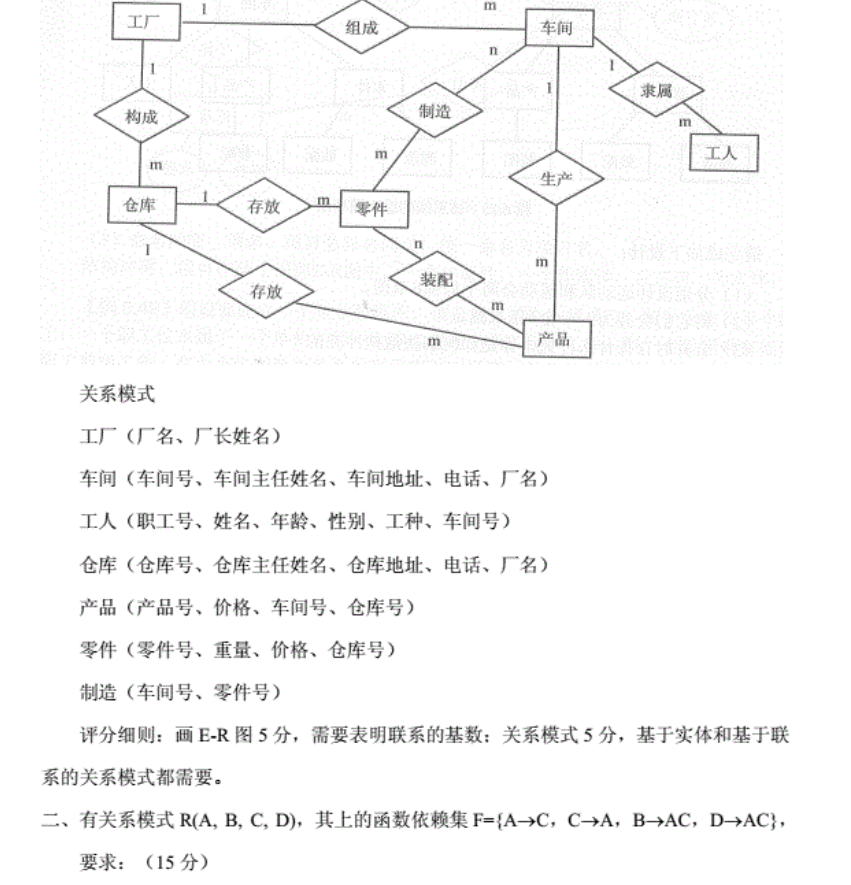
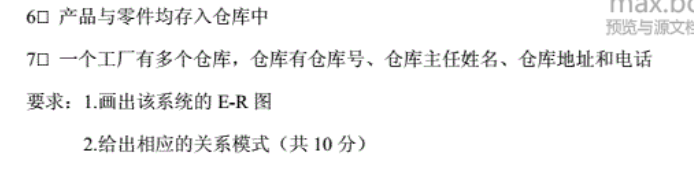
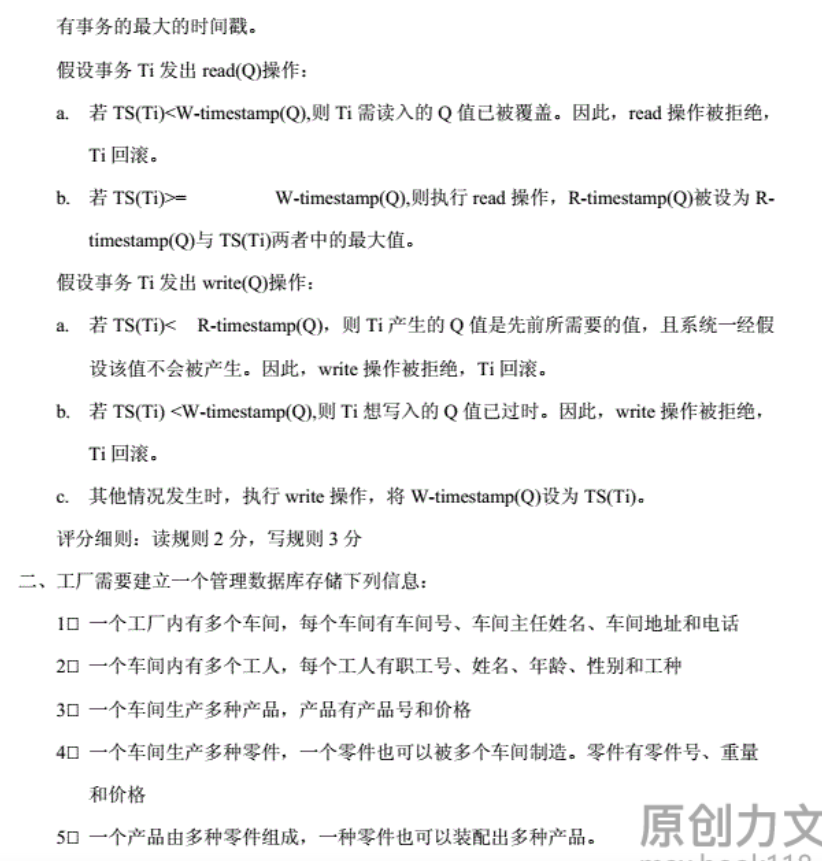
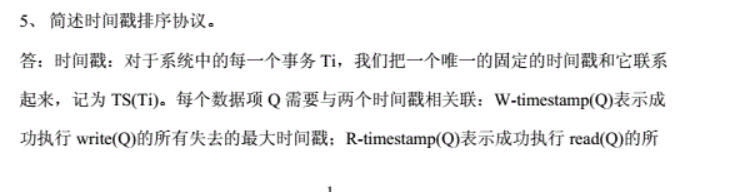
逻辑独立性是指用户的应用程序与数据库的逻辑结构是相互独立的，即，当数据的逻辑结构改变时，用户程序也可以不变数据与程序的独立，把数据的定义从程序中分离出去，加上数据的存取又由DBMS负责，从而简化了应用程序的编制，大大减少了应用程序的维护和修改。

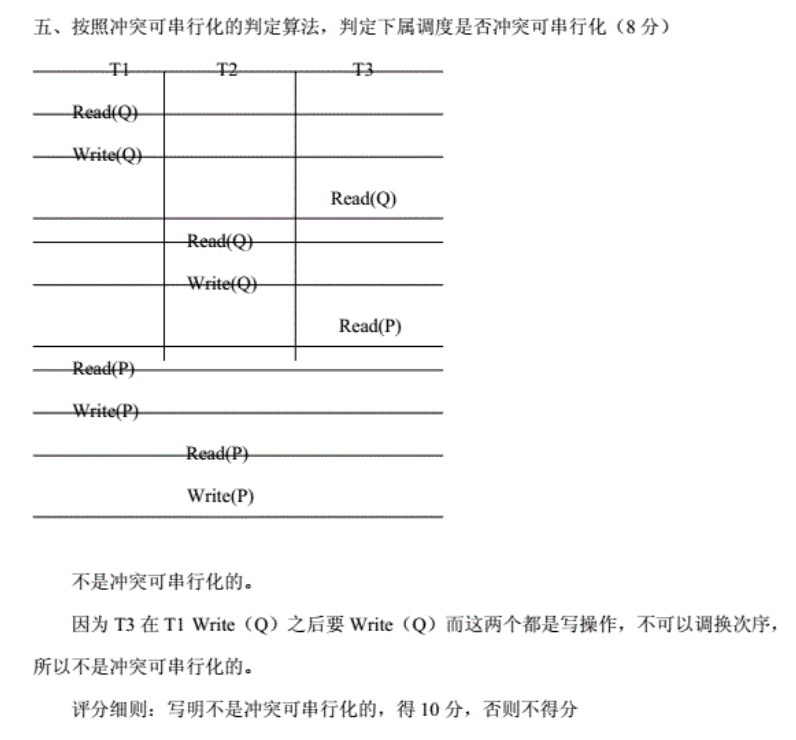
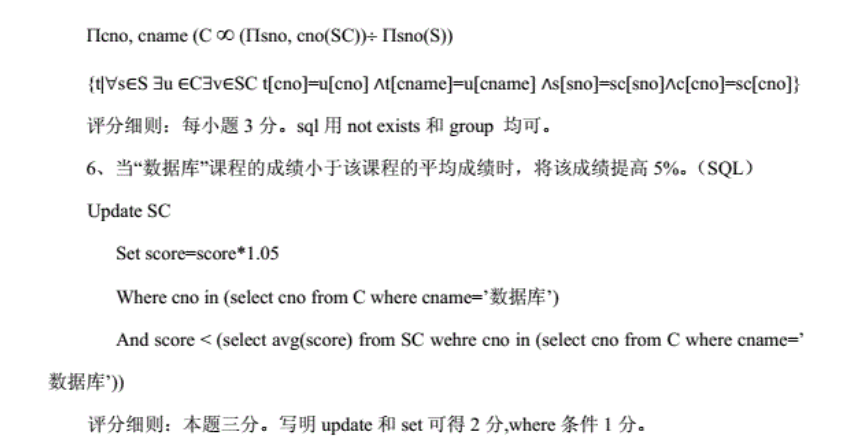
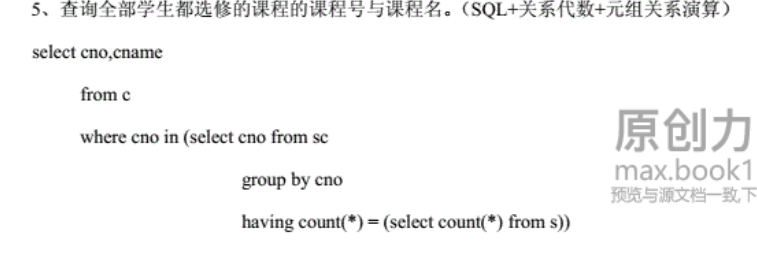
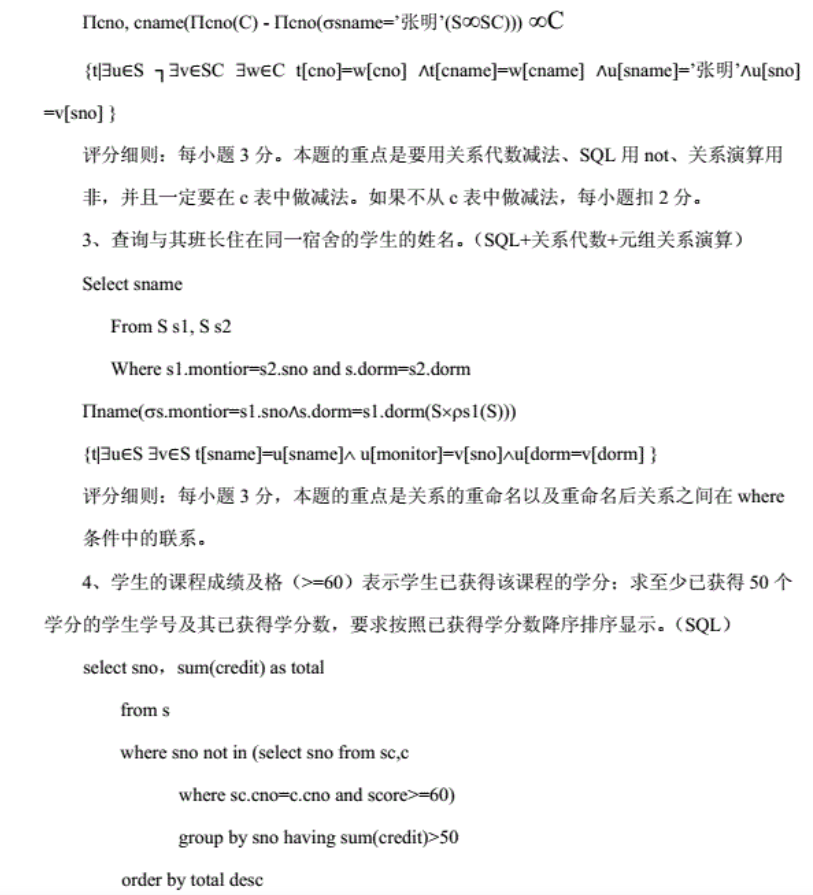
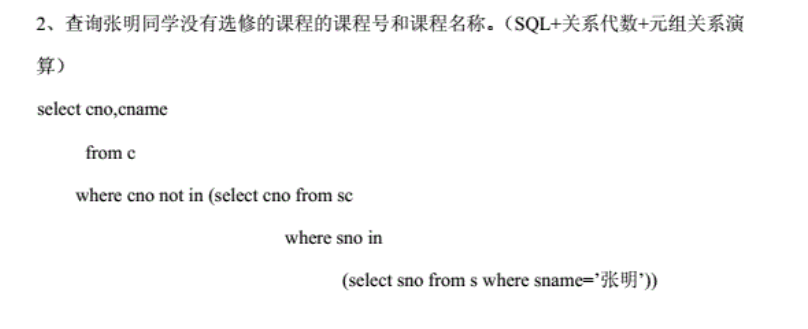
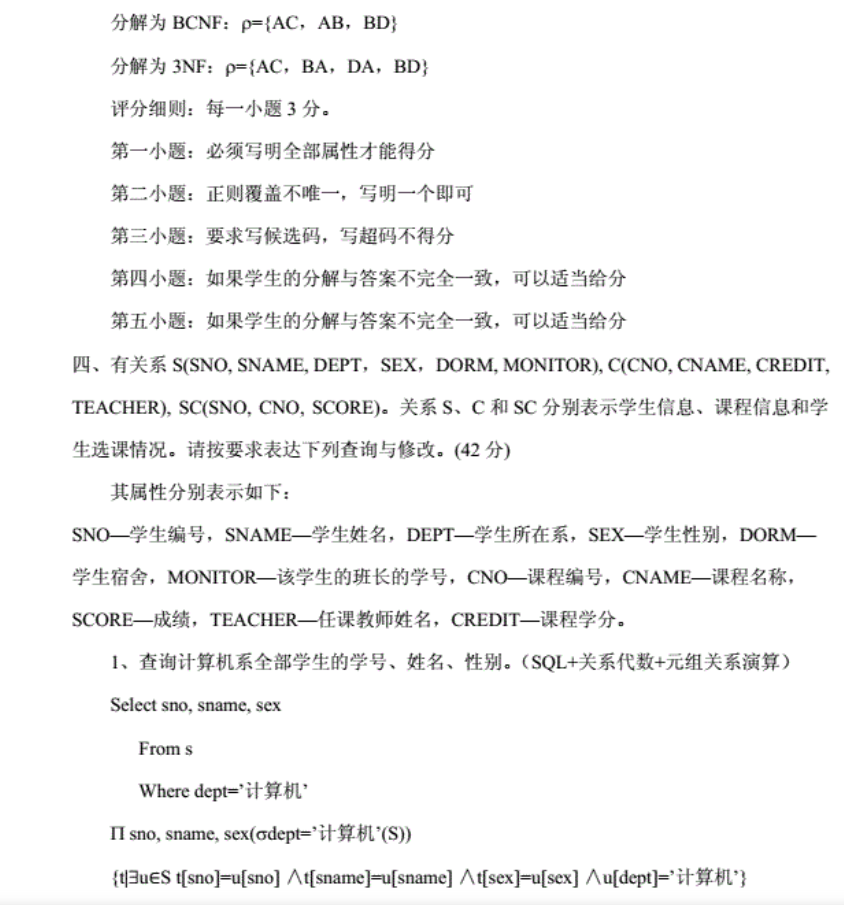
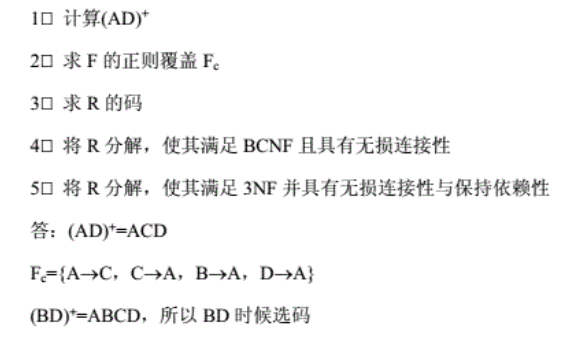
**说明串行调度和可串行化调度的区别。**

串行调度是多个事务按照一定的次序依次执行；而可串行化调度是指一个并发调度的结果与某一串行调度执行的结果等价时的并发调度。串行调度在某一时刻只有一个事务在执行，而可串行调度在某一时刻有多个事务同时被处理。

数据库答题例卷，2017







**DDL DML**

DDL:数据定义语言，Data-Deifinition Language，提供定义关系模式，删除关系以及修改关系模式的命令。

DML:数据操纵语言，Data-Manipulation Language，提供从数据库中查询信息，以及在数据库中插入元组、删除元组、修改元组的能力。

**数据库系统vs 文件系统**

数据的冗余和不一致

数据访问困难

数据孤立

完整性问题

原子性问题

并发访问异常

安全性问题