1. **试叙述事务的4个性质，并解释每一个性质由DBMS的哪个子系统实现？每一个性质对DBS有什么益处？ 第四章**

1）事务的**原子性**是指一个事务对DB的所有操作，是一个不可分割的工作单位。原子性是由DBMS的**事务管理子系统**实现的。事务的原子性保证了DBS的完整性。

2）事务的**一致性**是指数据不会因事务的执行而遭受破坏。事务的一致性是由DBMS的**完整性子系统**实现的。事务的一致性保证DBS的完整性。

3）事务的**隔离性**是指事务的并发执行与这些事务单独执行时结果一样。事务的隔离性是由

DBMS的**并发控制子系统**实现的。隔离性使并发执行的事务不必关心其他事务，如同在单用户环境下执行一样。

4）事务的**持久性**是指事务对 DB的更新应永久地反映在DB中。持久性是由 DBMS的**恢复管理子系统**实现的。持久性能保证 DB具有可恢复性。

1. **事务的 COMMIT语句和 ROLLBACK语句的功能是什么？ 第四章**

答：COMMIT语句表示事务执行成功地结束（提交），此时告诉系统DB要进入一个新的正确状态，该事务对DB的所有更新都已交付实施（写入磁盘）。ROLLBACK语句表示事务执行不成功地结束（应该“回退”）此时告诉系统已发生错误，DB可能处在不正确的状态，该事务对DB的所有更新必须被撤消，DB应恢复该事务到初始状态。

1. **日志文件中记载了哪些内容？ 第七章**

答：日志文件中记载了事务开始标记、事务结束标记以及事务对DB的插入、删除和修改的每一次操作前后的值。 即ATL活动事务表、CTL提交事务表、前像文件、后像文件。

1. **并发操作会产生几种不一致情况，用什么方法避免各种不一致的情况? 第七章**

答：并发操作可能会产生丢失修改, 不能重复读和读”脏”数据这三种不一致情况，采用封锁机制来进行并发控制, 可避各种不一致情况。

一级封锁协议可以避免丢失修改，二级封锁协议可以避免丢失修改和读“脏”数据，三级封锁协议可以避免丢失修改、不能重复读和读“脏”数据。

1. **X封锁与S封锁有什么区别？ 第七章**

答： X 锁 S 锁

|  |  |
| --- | --- |
| 只允许一个事务独锁数据 | 允许多个事务并发S锁某一数据 |
| 获准X锁的事务可以修改数据 | 获准S锁的事务只能读取数据，但不能修改数据 |
| 事务的并发度低 | 事务的并发度高，但增加了死锁的可能性 |
| X锁必须保留到事务终点 | 根据需要，可随时解除S锁 |
| 解决“丢失更新”问题 | 解决“读不一致性”问题 |

1. **为什么要设立日志文件? 第七章**

答：设立日志文件的目的是为了记录对DB中数据的每一次更新操作，从而DBMS可以根据日志文件进行事务故障的恢复和系统故障的恢复，并结合后备副本进去(介质)故障的恢复。

1. **怎样进行事务故障的恢复? 第七章**

答：事务故障是指事务在运行至正常终止点前被中止，这时恢复子系统应利用日志文件撤消（UNDO）此事务已对数据库进行的修改。事务故障的恢复是由系统自动完成的，对用户是透明的。系统的恢复步骤是：

**⑴ 反向扫描**文件日志（即从最后向前扫描日志文件），查找该事务的更新操作。

**⑵** 对该事务的更新操作执行**逆操作**，即将日志记录中“更新前的值”写入数据库。如果记录中是插入操作，则相当于做删除操作（因此时“更新前的值”为空）；若记录中是删除操作，则做插入操作；若是修改操作，则相当于用修改前值代替修改后值。

**⑶** 继续**反向扫描**日志文件，查找该事务的其他更新操作，并做同样处理。

**⑷** 如此处理下去，直至读到此事务的开始标记，事务故障恢复就完成了。

1. **怎样进行系统故障的恢复? 第七章**

答：系统故障造成数据库不一致状态的原因有两个，一是未完成事务对数据库的更新可能已写入数据库，二是已提交事务对数据库的更新可能还留在缓冲区没来得及写入数据库。因此恢复操作就是要撤消故障发生时未完成的事务undo，重做已完成的事务redo。

**⑴ 正向扫描**日志文件（即从头扫描日志文件），找出在故障发生前已经提交的事务（这些事务既有BEGIN TRANSACTION记录，也有COMMIT记录），将其事务标识记入重做(REDO)队列。同时找出故障发生时尚未完成的事务（这些事务只有BEGIN TRANSACTION记录，无相应的COMMIT记录），将其事务标识记入撤消(UNDO)队列。

**⑵** 对撤消队列中的各个事务进行撤消(UNDO)处理。进行UNDO处理的方法是，**反向扫描**日志文件，对每个UNDO事务的更新操作执行**逆操作**，即将日志记录中“更新前的值”写入数据库。

**⑶** 对重做队列中的各个事务进行重做(REDO)处理。进行 REDO处理的方法是： 正向扫描日志文件，对每个 REDO事务重新执行日志文件登记的操作。即将日志记录中“更新后的值”写入数据库。

1. **怎样进行介质故障的恢复? 第七章**

答：发生介质故障后，磁盘上的物理数据和日志文件被破坏，恢复方法是重装数据库，然后重做已完成的事务。具体地说就是：

**⑴** 装入最新的数据库**后备副本**（离故障发生时刻最近的**转储副本**），使数据库恢复到最近一次转储时的一致性状态。

对于动态转储的数据库副本，还须同时装入转储开始时刻的日志文件副本，利用恢复系统故障的方法（即REDO+UNDO），才能将数据库恢复到一致性状态。

**⑵** 装入相应的日志文件副本（转储结束时刻的日志文件副本），**重做已完成的事务**。即：

首先扫描日志文件，找出故障发生时已提交的事务的标识，将其记入重做队列，然后**正向扫描**日志文件，对重做队列中所有事务进行重做处理。即将日志记录中“更新后的值”写入DB。

1. **试述文件系统与数据库系统的区别和联系。 第一章**

答：**区别：文件系统**面向某一应用程序，共享性差，冗余度大，数据独立性差，记录内有结构，整体无结构，由应用程序自己控制。**数据库系统**面向现实世界，共享性高，冗余度小，具有较高的物理独立性和一定的逻辑独立性，整体结构化，用数据模型描述，由数据库管理系统提供数据的安全性、完整性、并发控制和恢复能力。

**联系：**文件系统与数据库系统都是计算机系统中管理数据中管理数据的软件。

1. **试述查询优化的一般步骤。 第六章**

1）把查询转换成某种内部表示；2）把语法树转换成标准（优化）形式；

3）降低层的存取路径；4）生成查询计划，选择代价最小的。

1. **数据库中为什么要有恢复子系统？它的功能是什么？ 第七章**

答：因为计算机系统中硬件的故障、软件的错误、操作员的失误以及恶意的破坏是不可避免的，这些故障轻则造成运行事务非正常中断，影响数据库中数据的正确性，重则破坏数据库，使数据库中全部或部分数据丢失。

**功能：**把数据库从错误状态恢复到某一已知的正确状态（亦称为一致状态或完整状态），这就是数据库的恢复。

1. **数据库转储的意义是什么？常用的有几种方法？ 第七章**

答：**数据转储**是数据库恢复中采用的基本技术，所谓转储即DBA定期将整个数据库复制到磁带或另一个磁盘上保存起来的过程。这些备用的数据文本称为**后备副本**。当数据库遭到破坏后可以将后备副本重新载入，将数据库恢复到转储时的状态。

转储可分为**静态转储和动态转储。**静态转储是在系统中无运行事务时进行的转储操作。

动态转储是指转储期间允许对数据库进行存取或修改，即转储和用户事务可以并发执行。

转储还可分为**海量转储和增量转储**两种方式。

1. **什么是日志文件？为什么要设立日志文件？ 第七章**

答：日志文件是用来记录事务对数据库的更新操作的文件。

目的是进行事物故障恢复；进行系统故障恢复；协助后备副本进行介质故障恢复。

1. **试述使用检查点(CP)方法进行恢复的步骤。 第七章 7.6**

**1）**从重新开始文件中找到最后一个检查点记录在日志文件中的地址，由该地址在日志文件中找到最后一个检查点记录。

**2）**由该检查点记录得到检查点建立时刻所有正在执行的事务清单ACL。

这里建立两个事务队列：

①UNDO-LIST：需要执行 undo 操作的事务集合；

②REDO-LIST需要执行 redo 操作的事务集合。

把ACL暂时放入UNDO队列， REDO队列暂时为空。

**3）**检查点开始**正向扫描**日志文件：

①如果有新开始的事务Ti，把Ti暂时放入UNDO队列；

②如有提交的事务Tj，把Tj从UNDO队列移到REDO队列，直到日志文件结束；

1. 对 UNDO队列中每个事务执行UNDO操作，对REDO队列中每个事务执行REDO操作。
2. **什么是数据库镜像？它有什么用途？ 第七章 7.1**

答：根据DBA的要求，自动把整个数据库或其中的关键数据复制到另一个磁盘上。每当主数据库更新时，DBMS自动把更新后的数据恢复过去，即DBMS自动保证镜像数据与主数据的一致性。

**用途：** 1）一旦出现介质故障，可由镜像磁盘继续提供使用，同时DBMS自动利用镜像磁盘数据进行数据库恢复，不需要关闭系统和重装数据库副本。

1. 在没有出现故障时，数据库镜像还可以并发操作。提高数据库的可用性。
2. **试述两段锁协议的概念。**

答：所谓两段锁协议是指所有事务必须分两个阶段对数据项加锁和解锁。

1）在对任何数据进行读、写操作之前，首先要申请并获得对该数据的封锁；

2）在释放一个封锁之后，事务不再申请和获得对该数据的封锁。

所谓“两段”锁的含义是，事务分为两个阶段，第一阶段是获得封锁，也称为扩展阶段。在这个阶段，事务可以申请获得任何数据项上的任何类型的锁，但是不释放任何锁。第二阶段是释放封锁，也称为收缩阶段。在这个阶段，事务可以释放任何数据项上的任何类型的锁，但是不能再申请任何锁。

1. **数据冗余的优缺点。 第一章+第七章**

数据冗余会造成存储浪费，造成数据不一致，带来版本控制问题。

数据冗余的优点有：对数据库恢复，数据冗余是必须的。

1. **检查点 第七章**

1）Checkpoint用于数据库发生系统失效时，避免大量的无效redo。

2）可以减少，具体与取CP的周期有关。

3）取后备副本后可以清空。

1. **第七章**

1）并发事务的正确性准则是可串行性。

2）S和U锁相容，可以提高并发度；U和U锁相容，造成死锁。

3）从降低管理锁代价角度，（S,X）锁更好

1. **试述视图的作用？ 第三章+第七章**

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。

1. **登记日志文件时必须遵循两条原则： 第七章**
2. 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。
3. **数据库管理系统（DBMS）的功能： 第一章+第四章**

DBMS是位于操作系统与用户之间的一个数据管理软件，它主要功能包括以下几个方面：

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。

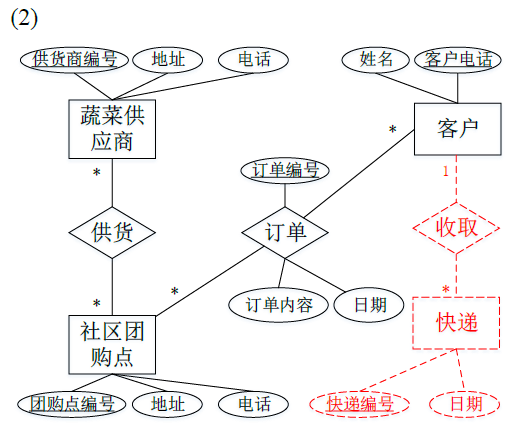
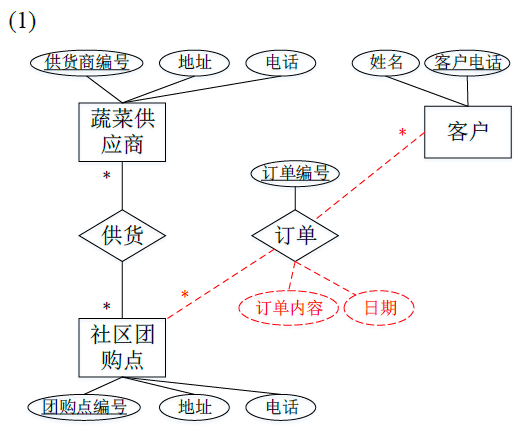
3）数据库的运行管理：这是DBMS 运行时的核心部分，它包括并发控制，安全性检查，完整性约束条件的检查和执行，发生故障后的恢复等。

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。

1. **简述等值连接与自然连接的区别和联系。 第二章**

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。



某医院病房管理系统中，包括四个实体型，分别为：

科室：科名，科地址，科电话

病房：病房号，病房地址

医生：工作证号，姓名，职称，年龄

病人：病历号，姓名，性别

且存在如下语义约束：

① 一个科室多个病房、 多个医生，一个病房只能属于一个科室，一个医生只属于一个科室；

② 一个医生可负责多个病人的诊治，一个病人的主管医生只有一个；

③ 一个病房可入住多个病人，一个病人只能入住在一个病房。

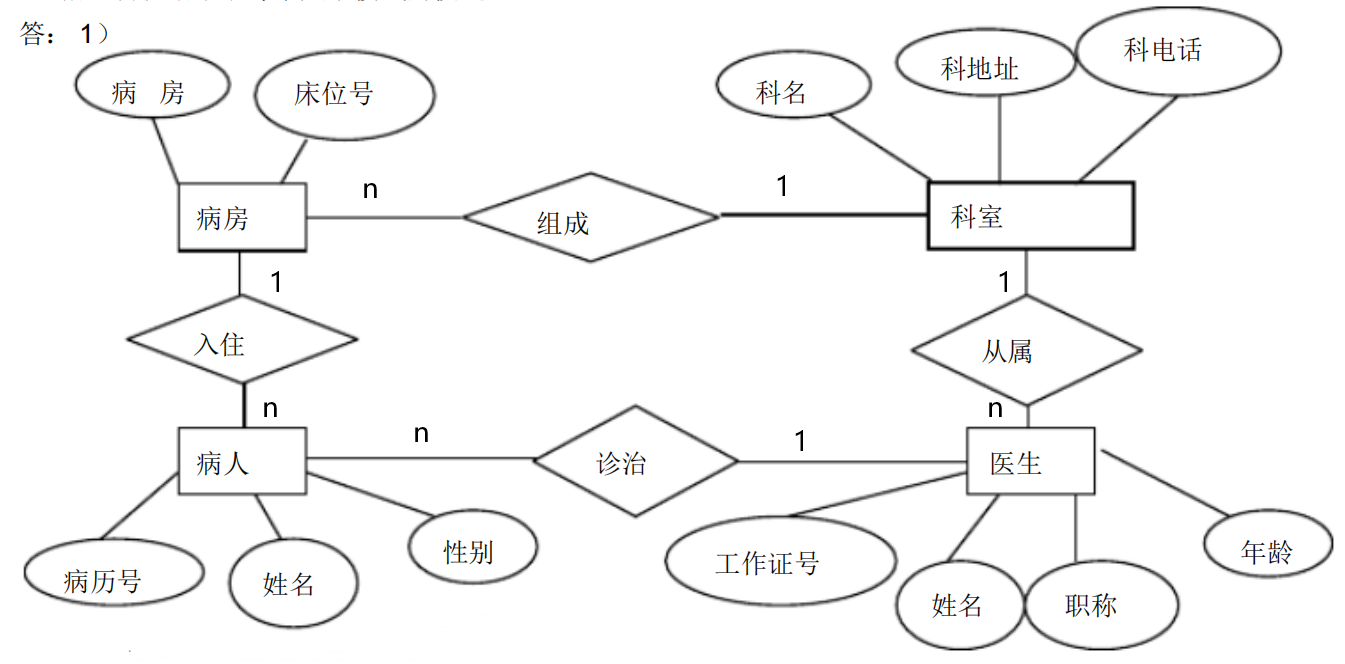
注意：不同科室可能有相同的病房号。

完成如下设计：

1）画出该医院病房管理系统的E－R 图；（5分）

2）将该 E-R 图转换为关系模型；（5 分）（要求： 1: 1 和 1: n 的联系进行合并）

3）指出转换结果中每个关系模式的主码和外码。（5 分）



2）转化后的关系模式如下： **科室**（科名，科地址，科电话） **病房**（病房号，病房地址，科名）

**医生**（工作证号，姓名，职称，年龄，科名） **病人**（病历号，姓名，性别，主管医生，病房号，科名）

3）每个关系模式的主码、外码如下： **科室**：主码是科名； **病房**：主码是科名十病房号，外码是科名；

**医生**：主码是工作证号，外码是科名； **病人**：主码是病历号，外码是科名十病房号。（主管医生工作证号）