

## 概念部分：

### 1.试说明数据、数据库、数据库管理系统和数据库系统的概念以及它们之间的关系。

答：(1) 数据(Data)：描述事物的符号记录称为数据。数据的种类有数字、文字、图形、图像、声音、正文等。数据与其语义是不可分的。

(2) 数据库(Database, 简称 DB)：数据库是长期储存在计算机内的、有组织的、可共享的数据集合。数据库中的数据按一定的数据模型组织、描述和储存，具有较小的冗余度、较高的数据独立性和易扩展性，并可为各种用户共享。

(3) 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，一般由数据库、数据库管理系统(及其开发工具)、应用系统、数据库管理员构成。

(4) 数据库管理系统 ( Database Management System, 简称 DBMS )：数据库管理系统是位于用户与操作系统之间的一层数据管理软件，用于科学地组织和存储数据、高效地获取和维护数据。DBMS 的主要功能包括数据库的建立和维护功能、数据定义功能、数据组织存储和管理功能、数据操作功能、事务的管理和运行功能。

**它们之间的联系：**数据库系统包括数据库、数据库管理系统、应用系统、数据库管理员，所以数据库系统是个大的概念。数据库是长期存储在计算机内的有组织、可共享的大量的数据集合，数据库管理系统是由管理员操作管理数据库的查询、更新、删除等操作的，数据库应用系统是用来操作数据库的。

### 2.比较用文件管理和用数据库管理数据的主要区别。

答：数据库系统与文件系统相比实际上是在应用程序和存储数据的数据库之间增加了一个系统软件，即数据库管理系统，使得以前在应用程序中由开发人员实现的很多繁琐的操作和功能，都可以由这个系统软件完成，这样应用程序不再需要关心数据的存储方式，而且数据的存储方式的变化也不再影响应用程序。而在文件系统中，应用程序和数据的存储是紧密相关的，数据的存储方式的任何变化都会影响到应用程序，因此不利于应用程序的维护。

答：文件系统把数据组织成相互独立的数据文件，实现了记录内的结构性，但整体无结构；而数据库系统实现整体数据的结构化，这是数据库的主要特征之一，也是数据库系统与文件系统的本质区别。

在文件系统中，数据冗余度大，浪费存储空间，容易造成数据的不一致；数据库系统中，数据是面向整个系统，数据可以被多个用户、多个应用共享使用，减少了数据冗余。

文件系统中的文件是为某一特定应用服务的，当要修改数据的逻辑结构时，必须修改应用程序，修改文件结构的定义，数据和程序之间缺乏独立性；数据库系统中，通过 DBMS 的两级映象实现了数据的物理独立性和逻辑独立性，把数据的定义从程序中分离出去，减少了应用程序的维护和修改。

文件系统和数据库系统均可以长期保存数据，由数据管理软件管理数据，数据库系统是在文件系统基础上发展而来。

### 3.数据库系统由哪几部分组成，每一部分在数据库系统中的作用大致是什么？

答：数据库系统由三个主要部分组成，即数据库、数据库管理系统和应用程序。数据库是数据的汇集，它以一定的组织形式存于存储介质上；数据库管理系统是管理数据库的系统软件，它可以实现数据库系统的各种功能；应用程序指以数据库数据为核心的应用程序。

## 数据模式部分：

#### **4.解释数据模型的概念，为什么要将数据模型分成两个层次？**

答：数据模型是对现实世界数据特征的抽象。数据模型一般要满足三个条件：第一是数据模型要能够比较真实地模拟现实世界；第二是数据模型要容易被人们理解；第三是数据模型要能够很方便地在计算机上实现。由于用一种模型同时很好地满足这三方面的要求在目前是比较困难的，因此在数据库系统中就可以针对不同的使用对象和应用目的，采用不同的数据模型。根据模型应用的不同目的，将这些模型分为概念层数据模型和组织层数据模型两大类，以方便对信息的描述。

#### **5.数据库系统包含哪三级模式？试分别说明每一级模式的作用？**

答：数据库系统包含的三级模式为：内模式、模式和外模式。外模式是对现实系统中用户感兴趣的整体数据结构的局部描述，用于满足不同数据库用户需求的数据视图，是数据库用户能够看见和使用的局部数据的逻辑结构和特征的描述，是对数据库整体数据结构的子集或局部重构。模式是数据库中全体数据的逻辑结构和特征的描述，是所有用户的公共数据视图。内模式是对整个数据库的底层表示，它描述了数据的存储结构。

#### **6.数据库系统的两级映像的功能是什么？它带来了哪些功能？**

答：数据库系统的两级映像是外模式 / 模式映像和模式 / 内模式映像。模式 / 内模式映像定义了概念视图和存储的数据库的对应关系，它说明了概念层的记录和字段在内部层次怎样表示。如果数据库的存储结构改变了，那么，必须对模式 / 内模式映像进行必要的调整，使模式能够保持不变。外模式 / 模式映像定义了特定的外部视图和概念视图之间的对应关系，当概念模式的结构可发生改变时，也可以通过调整外模式 / 模式映像间的映像关系，使外模式可以保持不变。

#### **7.什么叫数据与程序的物理独立性？什么叫数据与程序的逻辑独立性？**

答：数据与程序的物理独立性是指当数据库的存储结构改变，将模式/内模式映象作相应改变，使模式保持不变，从而应用程序不必改变。数据与程序的逻辑独立性是指当模式改变时，将外模式/模式的映象作相应改变，使外模式保持不变，从而应用程序不用修改。

#### **8.数据库三级模式划分的优点是什么？它能带来哪些数据独立性？**

答：数据库的三级模式的划分实际上将用户、逻辑数据库与物理数据库进行了划分，使彼此之间的相互干扰减到最少。这三个模式的划分实际上带来了两个数据独立性，即物理独立性和逻辑独立性。这使得底层的修改和变化尽量不影响到上层。

### **视图部分：**

#### **一、试说明使用视图的好处。**

答： 1、简化数据查询语句 2、使用户能从多角度看待同一数据  
3、提高了数据的安全性 4、提供了一定程度的逻辑独立性

#### **二、使用视图可以加快数据的查询速度，这句话对吗？为什么？**

答：不对。因为数据库管理系统在对视图进行查询时，首先检查要查询的视图是否存在，如果存在，则从数据字典中提取视图的定义，把视图的定义语句对视图语句结合起来，转换成等价的对基本表的查询，然后再执行转换后的查询，所以使用视图不可以加快数据的查询速度。

### **事务部分：**

#### **4. 登记日志文件时为什么必须先写日志文件，后写数据库？**

答：把数据的修改写到数据库和把对数据的修改操作写到日志文件是两个不同的操作，在两个操作之间可能会发生故障如果先写了数据库修改，而在日志文件中没有登记这个修改，在恢复的时候就无法恢复这个修改了；如果先写日志，但没有修改数据库，按日志恢复时只是多执行一次撤消操作，并不影响数据库的正确性，为了安全，一定要先写日志文件。

**8. 试叙述事务的4个性质，并结合每一个性质由DBMS的哪个子系统实现？每一个性质对DBS有什么益处？**

答：1) 事务的原子性是指一个事务对DB的所以操作，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。事务的一致性保证数据库的完整性。

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。

**9.在数据库中为什么要有并发控制机制？**

答：数据库系统一个明显的特点是多个用户共享数据库资源，尤其是多个用户可以同时存取相同数据。在这样的系统中，在同一时刻同时运行的事务可达数百个。若对多用户的并发操作不加控制，就会造成数据库存、取的错误，破坏数据的一致性和完整性。所以数据库中要有并发控制机制。

**14. 怎样进行系统故障的恢复？**

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，重做已完成的事务。

(1) 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。

(2) 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。

(3) 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

**15. 怎样进行介质故障的恢复？**

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：

(1) 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。

(2) 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，对重做队列中的所有事务进行重做处理。即将日志记录中“更新后的值”写入数

数据库。

### 17. 怎样进行事务故障的恢复

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：

- (1) 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。
- (2) 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。这样，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
- (3) 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。
- (4) 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。

### 24. 试述使用检查点方法进行恢复的步骤。

答：

- 1) 从重新开始文件中找到追后一个检查点记录在日志文件中的地址，由该地址在体制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 ACTIVE-LIST。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 ACTIVE-LIST 暂时放入 UNDO-LIST 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  $T_i$ ，把  $T_i$  暂时放入 UNDO-LIST 队列；
  - ②如有提交的事务  $T_j$ ，把  $T_j$  从 UNDO-LIST 队列移到 REDO-LIST 队列，直到日志文件结束；
- 4) 对 UNDO-LIST 中的每个事务执行 UNDO 操作，对 REDO-LIST 中的每个事务执行 REDO 操作。

### 25. 什么是数据库镜像？它有什么用途？

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。2) 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。

### 28. 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。

### 29. 数据模型的优化包含哪些方法？

- 答：
- 1、确定各属性间的函数依赖关系。根据需要分析阶段得出的语义，分别写出每个关系模式的各属性之间的函数依赖以及不同关系模式中各属性之间的数据依赖关系。
  - 2、对各个关系模式之间的数据依赖进行极小化处理，消除冗余的联系。
  - 3、判断每个关系模式的范式，根据实际需要确定最合适的范式。
  - 4、根据需要分析阶段得到的处理要求，分析这些模式对于这样的应用环境是否合适，确定

是否要对某些模式进行分解或合并。

4. 假设物理块的有效大小  $B=492$  字节，块的指针为 6 字节，Sailors 表的 sid 属性 2 字节，sname 占 4 字节，rating 占 1 字节，age 占 1 字节，master 占 2 字节。每个记录除属性外，还需增加一个字节作为删除标记。

(1) 若 Sailors 表的 rating 属性上建有 B+树簇集索引，求 B+树的秩  $k$ ? (6 分)

索引集： $2k * 1 + (2k+1) * 6 \leq 492$

顺序集： $2k * 1 + 2 * 6 + 2k * 6 \leq 492$

(2) 若 Sailors 表的 sid 属性上建有 B+树的主索引，求 B+树的秩  $k$ ? (6 分)

索引集： $2k * 2 + (2k+1) * 6 \leq 492$

顺序集： $2k * 2 + 2 * 6 + 2k * (6+2) \leq 492$