

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、完整性约束：

数据库完整性(Database Integrity)是指数据库中数据的正确性、有效性和相容性。数据库完整性由各种各样的完整性约束来保证，因此可以说数据库完整性设计就是数据库完整性约束的设计。

- 1、 实体完整性：要求每个关系模式有且仅有一个主码，每个主码的值必须唯一，而且不能为空。
- 2、 域完整性：数据库表中的列必须满足某种特定的数据类型或约束。其中约束又包括取值范围、精度等规定。表中的CHECK、FOREIGN KEY 约束和DEFAULT、 NOT NULL 定义都属于域完整性的范畴。
- 3、 参照完整性：参照的完整性要求关系中不允许引用不存在的实体。当更新、删除、插入一个表中的数据时，通过参照引用相互关联的另一个表中的数据，来检查对表的数据操作是否正确。

3. DBMS

数据库管理系统(Database Management System)是一种操纵和管理数据库的大型软件，用于建立、使用和维护数据库，简称DBMS。它对数据库进行统一的管理和控制，以保证数据库的安全性和完整性。用户通过DBMS访问数据库中的数据，数据库管理员也通过dbms进行数据库的维护工作。它可使多个应用程序和用户用不同的方法在同时或不同时刻去建立，修改和询问数据库。大部分DBMS提供数据定义语言DDL (Data Definition Language) 和数据操作语言DML (Data Manipulation Language)，供用户定义数据库的模式结构与权限约束，实现对数据的追加、删除等操作。

4. 什么是数据独立性？数据库系统如何实现数据独立性？

答：数据独立性是指应用程序和数据之间相互独立、互不影响，及数据结构的修改不会引起应用程序的修改。数据独立性包括物理独立性和逻辑独立性。

物理独立性是指数据库物理结构的变化时不必修改现有的应用程序。

逻辑独立性是指数据库逻辑结构变化时不需要改变应用程序。

数据独立性是由DBMS的三级模式和二级映像来实现的。

数据库系统通常采用外模式、模式和内模式三级结构，数据库管理系统在这三级模式之间提供了外模式/模式和模式/内模式两层映像。当整个系统要求改变模式时，由DBMS对各个外模式/模式映像作相应的修改，使外模式保持不变，从而使基于外模式的应用程序保持不变，从而保证了数据的逻辑独立性。当数据的存储结构改变时，由DBMS对模式/内模式映像进行修改，可以使模式保持不变，从而使应用程序也不必改变，保证了数据的物理独立性。

5. 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

6. 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。

7. 弱实体集：

弱实体集：如果一个实体集的所有属性都不足以形成主码，则称这样的实体集为弱实体集。每个弱实体集必须与另一个称作标识或主实体集的实体集关联才能有意义。

8. 正则覆盖：

满足下列条件的函数依赖集F称为正则覆盖，记作 F_c ：

- 1) F_c 与 F 等价
- 2) F_c 中任何函数依赖都不含无关属性
- 3) F_c 中函数依赖的左半部都是唯一的

9. 多值依赖与函数依赖：

设X,Y是关系R的两个属性集合，当任何时刻R中的任意两个元组中的X属性值相同时，则它们的Y属性值也相同，则称X函数决定Y，或Y函数依赖于X。

函数依赖规定某些元组不能出现在关系中，也称为相等产生依赖；

多值依赖要求某种形式的其它元组必须在关系中，称为元组产生依赖。

$X \rightarrow Y$ 的有效性仅决定于X、Y 属性集上的值；

$X \twoheadrightarrow Y$ 的有效性与属性集范围有关

10、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。

11、索引：

索引是一个单独的、物理的数据库结构，它是某个表中一系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

12、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。

13、试述 ER 模型、层次模型、网状模型、关系模型和面向对象模型的主要特点。

答：

ER 模型直接表示实体类型及实体间联系，与计算机系统无关，充分反映用户的需求，用户容易理解。

层次模型的数据结构为树结构，记录之间联系通过指针实现，查询较快，但DML 属于过程化的，操作复杂。

网状模型的数据结构为有向图，记录之间联系通过指针实现，查询较快，并且容易实现M:N 联系，但DML 属于过程化的语言，编程较复杂。

关系模型的数据结构为二维表格，容易为初学者理解。记录之间联系通过关键码实现。DML 属于非过程化语言，编程较简单。

面向对象模型能完整描述现实世界的数据结构，具有丰富的表达能力，能表达嵌套、递归的数据结构。但涉及的知识面较广，用户较难理解

14、什么是函数依赖？

答：设 $R(U)$ 是属性集 U 上的关系模式， $X, Y \subseteq U$ ， r 是 $R(U)$ 上的任意一个关系，如果成立 对 $\forall t$ ， $s \in r$ ，若 $t[X] = s[X]$ ，则 $t[Y] = s[Y]$ 那么称“ X 函数决定 Y ”，或“ Y 函数依赖于 X ”，记作 $X \rightarrow Y$

15、在嵌入式SQL中，什么情况下的DML语句不必涉及到游标操作？

答：INSERT、DELETE和UPDATE语句；
对于SELECT语句，如果已知查询结果肯定是单值时。

16、 简述数据库管理员的职责；

答：

模式定义：DBA通过用DDL书写的一系列定义来创造最初的数据库模式。

存储结构及存取方式定义。

模式及物理组织的修改：由DBA对模式及物理组织进行修改，以反映组织的需求变化，或者为提高性能选择不同的物理组织；

数据访问授权：通过授予不同的权限，DBA可以规定不同的用户各自可以访问的数据库的部分；

日常维护：定期备份数据库、确保正常运转时所需的空余磁盘空间、监视数据库的运行。

17、什么是多值依赖？

关系模式 $R(U)$ ， X 、 Y 、 Z 是关系模式上的属性集， $Z=U-X-Y$ ，对于 $R(U)$ 的任一关系 r ，若存在元组 t_1, t_2 ，使得 $t_1[X] = t_2[X]$ ，那么就必然存在元组 t_3, t_4 ，使得： $t_3[X] = t_4[X] = t_1[X] = t_2[X]$ ， $t_3[Y] = t_1[Y]$ ， $t_4[Y] = t_2[Y]$ ， $t_3[Z] = t_2[Z]$ ， $t_4[Z] = t_1[Z]$ ，则称 Y 多值依赖于 X ，记作 $X \twoheadrightarrow Y$

18、 死锁的发生是坏事还是好事？试说明理由。如何解除死锁状态？

答：在DBS 运行时，死锁状态是我们不希望发生的，因此死锁的发生本身是一件坏事。但是坏事可以转换为好事。如果我们不让死锁发生，让事务任意并发做下去，那么有可能破坏DB 中数据，或用户读了错误的数据。从这个意义上讲，死锁的发生是一件好事，能防止错误的发生。

在发生死锁后，系统的死锁处理机制和恢复程序就能起作用，抽取某个事务作为牺牲品，把它撤消，做ROLLBACK 操作，使系统有可能摆脱死锁状态，继续运行下去。

19、什么是“饿死”问题？如何解决？

有可能存在一个事务序列，其中每个事务都申请对某数据项加S 锁，且每个事务在授权加锁后一小段时内释放封锁，此时若另有一个事务 T_1 欲在该数据项上加X 锁， 则将永远轮不上封锁的机会。这种现象称为“饿死”（starvation）。

可以用下列方式授权加锁来避免事务饿死。

当事务 T_2 中请对数据项 Q 加S 锁时，授权加锁的条件是：

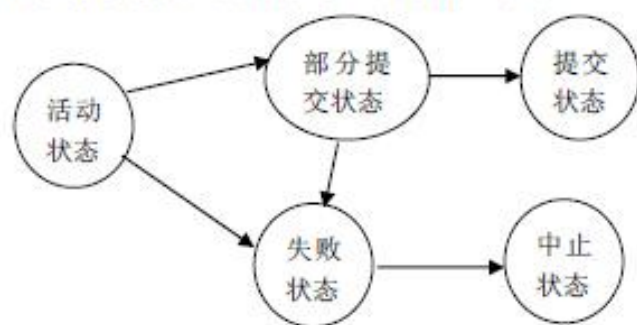
- (1)不存在在数据项 Q 上持有X 锁的其他事务；
- (2)不存在等待对数据项 Q 加锁且先于 T_2 申请加锁的事务。

20、什么是无关属性？

答：对于函数依赖集 F 及 F 中函数依赖 $\alpha \rightarrow \beta$ ，1) 属性 A 在 α 中是无关的，如果 $A \in \alpha$ ，并且 $F \vdash (F - \{\alpha \rightarrow \beta\}) \cup \{(\alpha - A) \rightarrow \beta\}$ ；2) 属性 A 在 β 中是无关的，如果 $A \in \beta$ ，并且 $(F - \{\alpha \rightarrow \beta\}) \cup \{\alpha \rightarrow (\beta - A)\} \vdash F$ 。

21、

请画出事务状态图，各种状态之间有什么联系？



22、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。

23、DBS 中有哪些类型的故障？哪些故障破坏了数据库？哪些故障未破坏数据库，但使其中某些数据变得不正确？

DBS 中DB 的故障主要有三类：事务故障、系统故障和介质故障。前两类故障未破坏DB，但使其中某些数据变得不正确，此时只要利用日志撤消或重做事务。介质故障将破坏DB，此时只能把DB 备份拷贝到新的磁盘，再利用日志重做事务对DB 的修改。

24、为什么稳定性存储器是不可能实现的？在DBS 中采用什么方法追求这个目标？

由于存储器受制于本身的故障，会导致信息的丢失，因此稳定性存储器是不可能实现的。但可以通过对非易失性存储器进行技术处理来达到稳定性存储器的目标。主要是通过数据备份和数据银行形式实现。

25、与传统的关系模型相比，对象关系模型有哪些扩充？

1 在定义语言上有三个扩充：

数据类型的扩充（引入复合类型）；

在类型一级和表一级实现继承性；

使用“引用类型”。

2 在查询语言方面，用户需记住属性值是单值还是多值。在多值时，需定义新的元组变量。

26、可以接受的分解必须是无损分解吗？

1 泛关系存在问题，因此进行模式分解；

2 如果分解有损，即分解后的模式不能有效表示泛关系原来含有的全部信息，那么分解没有价值，有损分解是不可实际接受的

27、模式规范化的基本思路是什么？

1 从泛关系出发

2 通过模式分解得到好的(规范的)模式

3 模式分解: $R \rightarrow \{R_1, R_2, \dots\}$

评价模式分解的两个重要指标是:

1 无损连接分解/有损连接分解;

2 是否保持依赖

28、大模式的缺点有哪些?

(1) 数据冗余大

(2) 容易出现数据不一致

(3) 不能表示某些信息: 没有学生的院系

(4) 插入异常

(5) 删除异常

之所以出现上述问题, 要思考问题的本质, 是因为

(1) 两件事不应使用一个表来表示

(2) 应该将这个模式分解为两个模式

(3) 按照一定的策略, 无序分解不可接受

该大则大, 该小则小, 设计的结果应尽可能真实的反映现实。

29、视图

比较:

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

视图数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:

* 简单性。看到的就是需要的。视图不仅可以简化用户对数据的理解, 也可以简化他们的操作。那些被经常使用的查询可以被定义为视图, 从而使得用户不必为以后的操作每次指定全部的条件。

* 安全性。通过视图用户只能查询和修改他们所能见到的数据。但不能授权到数据库特定行和特定的列上。通过视图, 用户可以被限制在数据的不同子集上:

视图的定义: 计算机数据库中的视图是一个虚拟表, 其内容由查询定义。同真实的表一样, 视图包含一系列带有名称的列和行数据。但是, 视图并不在数据库中以存储的数据值集形式存在。而是在引用视图时动态生成。

30、举例说明参照完整性对数据有什么要求。

如果关系 R_2 的外部码Fk 与关系 R_1 的主码Pk 相对应, 则 R_2 中的每一个元组的Fk 值或者等于 R_1 中某个元组的Pk 值, 或者为空值。