

1. 表、视图、查询表的区别。

答：视图：视图是从一个或几个基本表(或视图)导出的表。它与基本表不同，是一个虚表。视图只存定义，每次查看视图就相当于执行一次语句。

关系：基本表是实际存在的表，它是实际存储数据的逻辑表示；查询表是查询结果对应的表；视图表是由基本表或其他视图表导出的表，是虚表，不对应实际存储的数据。

可以理解为表最稳定，一般就放在那里；有时候不想给别人看整张表，就弄一个视图，但也相对稳定一点；查询最不稳定，需要什么再临时查，非常灵活多变。

2. 事务的 ACID 特征

答：原子性，一致性，隔离性，持久性

原子性：事务是不可分割的，要么执行全部内容，要么根本不执行

一致性：在没有其他事务并发执行的情况下，保持数据的一致性

隔离性：每个事务都感觉不到系统中有其他事务在并发地执行

持久性：一个事务成功完成之后，它对数据库的改变必须是永久的，即使出现系统故障

3. 事务的状态：

答：活动的、部分提交的、失败的、中止的、提交的

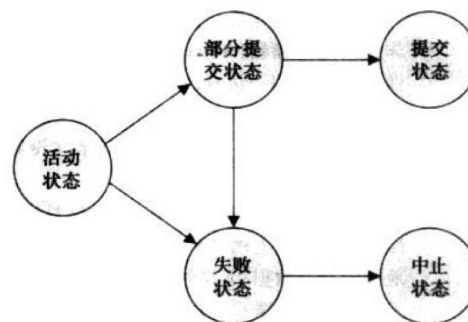


图 14-1 事务状态图

4. 冲突等价：如果调度 S 可以经过一系列非冲突指令交换转换成 S' ，我们称 S 和 S' 是冲突等价的

冲突可串行化：如果调度 S 可以经过一系列非冲突指令交换转换成串行调度 S' ，那么我们就说 S 是冲突可串行化的

5. 两阶段锁协议的作用以及内容。

答：作用：用来保证冲突可串行化

内容：分为两个阶段

- 第一阶段为增长阶段，只能对事务进行加锁操作，不能进行释放锁的操作
- 第二阶段为缩减阶段，只能对事务进行释放锁的操作，不能进行加锁操作

6. 严格两阶段锁和强两阶段锁

答：严格两阶段锁：在两阶段锁的基础上加上在当前事务提交或回滚之前，不能够释放排他锁。

强两阶段锁，在两阶段锁的基础上加上在当前事务提交或回滚之前，不释放任何锁

7. 死锁处理方法

答：(1) 死锁预防协议，不让系统进入死锁状态

(2) 死锁检测+死锁恢复，允许系统进入死锁状态

8. 死锁预防：

答：使用抢占和事务回滚

分类为：

Wait-die(非抢占),时间戳小才能等
Wound-wait(抢占),时间戳大才能等

9. 死锁恢复:

答: 解除死锁最通常的做法是回滚一个或多个事务。

首先根据“最小代价”选择牺牲者, 对选择的牺牲者进行彻底或部分回滚, 为了避免饿死, 我们需要规定, 每个事务被当作牺牲者的次数上限。

10. 强实体集和弱实体集的区别和联系

区别:

编号	强实体	弱实体
1	强实体总是有一个主键。	弱实体具有部分鉴别器键。
2	强实体不依赖于任何其他实体。	弱实体依赖强实体。
3	强实体由单个矩形表示。	弱实体由双矩形表示。
4	两个强实体的关系由单个菱形表示。	一强一弱实体之间的关系用双菱形表示。
5	强大的实体要么完全参与, 要么不参与。	弱实体总是完全参与。

联系: 都是具有相同性质的实体的集合

11. 实体: 是现实世界中可区别于所有其他对象的一个“事物”或“对象”

实体集: 是相同类型即具有相同性质的一个实体的集合

联系: 是指多个实体之间的相互关联

联系集: 相同类型联系的集合

联系集的度: 参与联系集的实体集的数目

12. 模式: 数据库的逻辑结构, 它分为物理模式和逻辑模式

实例: 特定时刻存储在数据库中的信息的集合被称作数据库的一个实例

13. 数据库模式: 数据库的逻辑设计

数据库实例: 给定时刻数据库中数据的一个快照

14. 数据库的三级抽象

答: 视图层、逻辑层、物理层

15. 文件处理系统中存储组织信息的主要弊端:

答: 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

16. 数据管理系统 (DBMS) 的组成: 一个相互关联的数据的集合+一组用以访问这些数据的程序

数据管理系统的主要目标: 提供一种方便、高效的存取数据库信息的途径

17. 数据模型: 一个描述数据、数据联系、数据语义以及一致性约束的概念工具的集合。

18. 数据库语言分类

答: 数据库操纵语言和数据库定义语言

数据库操纵语言分为过程化 DML 和声明式 DML

19. 什么是数据库索引？

答：一种用来排序的数据结构，用来加快查询和更新数据库表中的数据

20. 函数、过程、触发器

答：它们都是存储在数据库当中的一段代码。

差别是：

函数需要显式调用，有返回值

过程无返回值

触发器需要有出发时间，系统自动调用，无返回值

21. 范式分类

答：第一范式：强调的是列的原子性，即列不能够再分成其他几列

第二范式：首先是 1NF，另外包含两部分内容，一是表必须有主键；二是没有包含在主键中的类必须完全依赖于主键，不能只依赖主键的一部分；

第三范式：首先是 2NF，另外非主键必须直接依赖于主键，不能存在传递依赖。即不能存在：非主键列 A 依赖于非主键 B，非主键 B 依赖于主键的情况。

22. 在学生表中，将学号(ID)和姓名(name)的组合设计为该表的主键是否合理？为什么？

答：不合理。因为这种情况下，允许有多个姓名不同的学生对应相同学号的情况。

23. 如果张三想通过汇款的方式转给李四 200 元。张三的账户已经减掉 200 美元后系统发生故障，并没有在李四账户中增加 200 美元，请问数据库出现了什么样的状态？事务现在是一种什么状态？这个问题该怎么解决？数据库通过什么手段实现该操作？

请问事务现在是一种什么状态？这个问题该怎么解决？

答：数据库处于不一致的状态，事务处于失败的状态。应该进行事务回滚，即将张三账户中减掉的 200 美元再加回来。利用日志实现。

24. 视图可更新的条件

答：（1）创建的视图基于一个表，也就是单表视图

（2）视图要包含主键

（3）视图中不包含聚集函数

25. SQL 中的日期和时间类型

答：date:年、月、日

Time:时、分、秒

Timestamp:年、月、日、时、分、秒

Interval:一段时间

26. 简述用户自定义的类型和域之间的差别。

答：（1）类型是强类型检查，无法定义约束

（2）域是弱类型检查，可以定义约束

27. 断言和 check 约束的区别：

答：check 只能在单个关系中进行约束，断言可以在多个表之间进行约束

28. 数据库中常用的权限：

（1）对于某一个表中数据的权限：

- 读操作(read)
- 修改操作(update)

- 插入操作(insert)
 - 删除操作(delete)
- (2) 对于数据库当中的对象的权限
- Index: 可以创建和删除索引
 - Resources: 将常用的权限归结在一起形成一个角色
 - Alteration: 允许对关系当中的某一些属性进行修改
 - Drop: 删除关系
29. SQL 标准包括的权限:
- 答: select, insert, update, delete, all privileges
30. 为什么要在数据库中引入事务的概念?
- 答: 要在数据库并发控制下保持数据的一致性
31. 为什么要在数据库中设置多种不同粒度的锁?
- 答: 某些情况下需要把多个数据项聚为一组, 将它们作为一个同步单元, 此时如果还是每个数据上一个锁的话效率会大大降低。
32. 为什么要对数据库的调度进行可串行化判别, 其实际意义是什么?
- 答: 保证所执行的调度的效果与没有并发执行的调度效果一样, 可以确保数据的一致性
33. 数据库中, 实体的完整性是如何被保证的?
- 答: not null, unique, check, 参照完整性
34. 如何降低数据库中数据的冗余度?
- 答: 设计选择更小的模式
35. 关系代数中, 与等值连接相比, 自然连接的缺点是什么?
- 答: 自然连接是在笛卡尔积的基础上保留相同属性下取值相同的元组。
而等值连接可以在笛卡尔积的基础上自由选择比较的属性。
36. 数据库管理系统中 DDL 所能完成的操作包括哪些?
- 答: create drop alter
37. 关系数据库设计中, 至少应满足的规范化条件是什么?
- 答: 1NF
38. 判断分解后的关系模式是否合理的两个重要标志\
- 答: 无损分解和保持依赖
39. 基于多表的视图, 可以完成哪些操作, 不能完成哪些操作?
- 答: 可以查询, 不能更新
40. 实体之间的联系有哪几种?
- 答: 一对一 多对多 一对多 多对一
41. 简述 where 子句和 having 子句的区别。
- 答: where 对自然属性进行判断(group 之前)
Having 对中间结果进行判断(group 之后)
42. Group by 子句后的属性:
- 答: 出现在 select 中的属性(除聚集函数之外), 必定出现在 group by 中
43. 简述数据库中为何要进行并发控制?
- 答: 当多个事务在数据库中并发地执行时, 数据的一致性可能不再维持, 系统有必要控制各事务之间的相互作用。
并发控制机制存在的目的是为了获得高度的并发性, 同时保证所产生的调度是冲突可串行化的或视图可串行化可恢复的, 并且无级联的。

44. 举例说明数据库中死锁含义及其解决方法。

答：事务之间相互等待对方所占用的资源。

解决方法是两个死锁事务回滚其中的一个。

45. 并、交、差的关键词

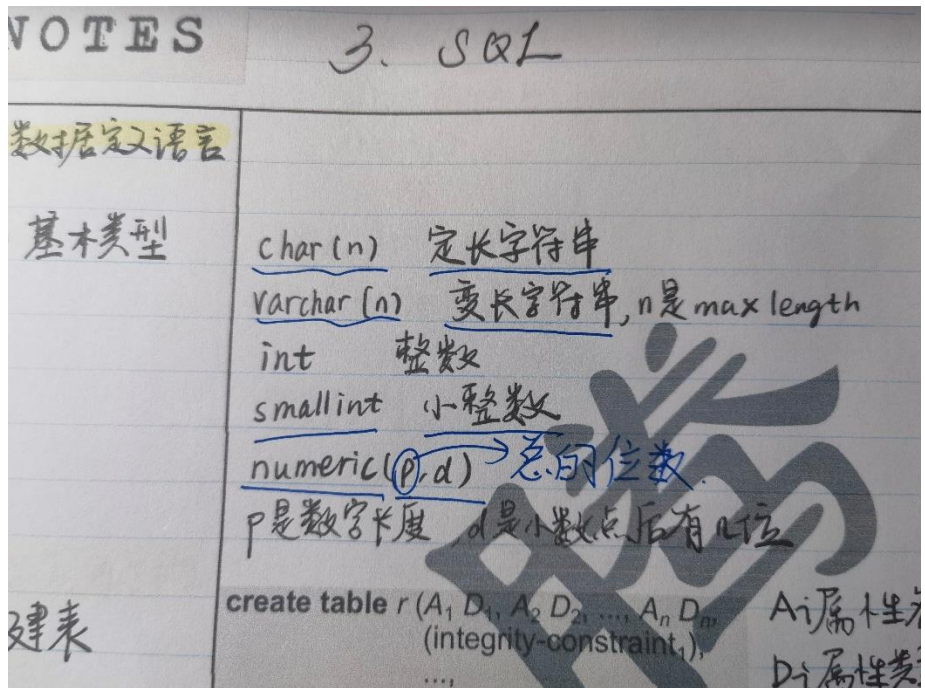
答：并：union

交：intersect

差：except

46. SQL 数据基本类型

答：



47. 大对象类型：

答：作用：存储 KB、MB、甚至 GB 级的属性

分类：字符数据大对象类型 blob

二进制数据大对象类型 clob