## **习题6**

6.1 什么是事务？它与一般的程序有什么不同？为什么一般程序不提ACID特性？

答：事务代表了用户要完成的一项工作任务，是将一组数据库操作打包起来形成一个逻辑独立的工作单元，这个工作单元不可分割，其中包含的数据库操作要么全部都发生，要么全部都不发生。

程序和事务是两个不同的概念，程序虽然也包含一组操作，但这组操作没有蕴含不可分割的原子性概念，程序执行过程中可能会出错或发生异常。

程序一般不提ACID特性，原因在于程序通常处理的是内存数据结构，其数据一致性体现在程序代码逻辑中。如果需要保证非易失性数据的ACID特性，则程序中可以包含一个或多个事务，通过事务来处理ACID特性。

6.2什么是日志文件？为什么要使用日志文件？ 登记日志文件时为什么必须先写日志文件，后写数据库？日志文件能否和数据库存储在一起，为什么？

答：日志文件是用来记录对数据库每一次更新活动的文件。使用日志文件，在系统发生故障时根据日志文件中的记录可以将数据库恢复到数据库事务执行之前的状态，也可以重新执行一个事务，保证数据库处于一个一致的状态。

登记日志文件时必须先写日志文件，后写数据库。原因是事务执行过程中可能发生故障数据库一致性遭到破坏，此时需要通过日志文件进行恢复。

日志文件必须和数据库存储在不同的磁盘上，避免磁盘损坏导致数据库和日志文件都遭到破坏，数据库无法恢复。

6.3 数据库运行过程中常见的故障有哪几类？试述对各类故障的恢复策略。

答：数据库运行过程中常见的故障有：事务故障、系统故障、介质故障。

事务故障是在事务执行过程中发生的故障，事务故障一定在事务提交前发生，这时应撤消（Undo）该事务对数据库一切更新，是由系统自动完成的，对用户透明。

系统故障是指造成系统停止运转并要求系统重新启动的事件。系统故障会使内存数据丢失，这样会使已提交的事务对数据库的更新还留在工作区而未写入数据库，所以，对所有已提交的事务需要重做，而对未提交的事务必须撤消所有对数据库的更新。恢复是由系统在重新启动时自动完成的，不须用户的干预。

介质故障是指外存故障，如磁盘损坏、磁头碰撞、瞬时强磁场干扰等，这类故障将使数据库受到破环。发生介质故障后，磁盘上的数据都可能被破坏。这时，恢复的方法是重装数据库，然后重做已经完成的事务。

6.4 什么是检查点？设置检查点有什么作用？设置检查点时系统将做什么动作？

答：检查点是一个数据库事件，DBMS按一定的间隔在日志文件中设置一个检查点。

采用检查点技术，在系统故障恢复时，记入重做队列的事务，只是从最近一个检查点之后到发生故障时已提交的事务，这样可以大大减少重做的工作量。

检查点发生时要暂停事务执行，将上一个检查点之后已提交事务留在内存工作区所有更新的数据写入数据库，并在日志文件中写入一个检查点记录。

6.5 给予下述问题的简要回答。

1. 什么是并发？

答：多个事务在时间上交叉执行并发地存取数据库，这种执行方式称为并发存取。

1. 并发操作会引起什么问题？

答：丢失修改（更新）、不可重复读、读“脏”数据。

1. 什么是丢失更新？

答：当两个事务T1和T2先后对同一数据对象A进行修改并写入数据库，后写入的结果会覆盖掉先写入的结果，导致先写入的事务修改结果丢失了，即丢失修改问题。

1. 什么是读脏数据？

答：读“脏”数据，简称脏读，是指一个事务读取了另一个未提交的事务中的数据。

1. 什么是读值不可复现？

答：一个事务没有执行任何更新操作，但同一个查询操作执行两次或多次，结果不一致，就说明产生了不可重复读的现象，即读值不可复现。

1. 什么样的并发操作是正确的？

答：当一个事务集的并发调度与它的某一串行调度是等价的，则称该并发调度是可串行化的。可串行化调度的结果保持数据库的一致性，是正确的。

1. 串行调度和可串行化调度有什么区别？

答：一个调度中各个事务没有交叉执行而是串行执行，则该调度称为串行调度。当一个事务集的并发调度与其某一串行调度等价，则称该并发调度是可串行化调度。

6.6 有两个事务：

T1：lock（A）， lock（B）， unlock（A）， unlock（B）

T2：lock（B）， unlock（B）， lock（A）， unlock（A）

它们有多少种合法调度？其中有多少种是可串行化的？

答：本题中，合法调度要求保持事务内部操作的先后次序，且满足加锁约束：事务在数据对象R上加锁则其他事务不允许在R上加锁。根据要求可以枚举出11种合法调度，其中包含两种串行调度，其余9种并发调度都不是可串行化调度。

6.7 什么是两段锁协议？什么是严格的2PL协议？如何实现2PL协议？

答：两段锁协议（简称2PL 协议），是一种能够实现并发调度可串行化的封锁协议。两段锁协议规定：（1）在任何数据对象进行读、写操作之前，事务要获得对数据对象的加锁；（2）在释放任意一个锁之后，事务不再允许获得任何其他加锁。

把所有锁都放在事务结束时释放的两段锁协议称为严格的2PL协议。

在实现中，两段锁协议把事务的加锁过程分成扩展和收缩阶段，其中扩展阶段逐步增加锁，收缩阶段逐步释放锁。

6.8 什么叫活锁？如何防止活锁？

答：系统可能使某个事务永远处于等待状态，得不到封锁的机会，这种现象称为系统进入了活锁。采用先来先服务的策略可以防止活锁。

6.9 什么叫死锁？如何预防和处理死锁？

答：如果系统中有两个或两个以上的事务都处于等待状态，并且每个事务都在等待其中另一个事务解除封锁，它才能够继续执行下去，结果造成任何一个事务都无法继续执行，这种现象称为系统进入了死锁。对付死锁有两种方法：一是死锁预防；二是死锁检测，即发现死锁后解除，死锁检测比较常用。

6.10设有两个事务的一个调度S：

T1：Write（A）， T2：Write（B）， T1：Write（B）， T2：Write（A）

T1─时间标记为20， T2—时间标记为30。

请说明T1，T2在2PL协议下的执行过程，及其等效的串行执行次序。

答：在2PL协议下事务T1和T2的调度S会导致死锁，不存在等效的串行执行次序。

|  |  |  |
| --- | --- | --- |
|  | T1 | T2 |
| 20 | XLock（A）  Write（A） |  |
| 30 |  | XLock（B）  Write（B） |
|  | XLock（B）  等待…… |  |
|  | 等待…… | XLock（A）  等待…… |
|  | 等待…… | 等待…… |

6.11 什么是数据库的安全性和完整性？两者之间有什么联系和区别？

答：数据库的安全性是指保护数据库以防止不合法的使用所造成的数据泄露、更改或者破坏。数据库的完整性是指数据库中数据的正确性和相容性，即为了防止数据库中存在不符合语义的数据，防止错误信息的输入和输出。

安全性防范的是非法用户和非法操作，完整性防范的是不合语义的数据。

6.12 假设有如下两个关系模式：

Emp(Eno, Ename, Eage, Salary, Deptno)

Dept(Deptno, Dname, Phone, Loc)

现在有三个用户U1，U2和U3。使用SQL的授权语句实现下列要求。

1. U1只能读Emp关系中除了 alary以外的所有属性。

答：GRANT SELECT(Eno, Ename, Eage, Deptno) ON TABLE Emp TO U1;

1. U2可以读、增、删Dept关系，并可以修改此关系的Phone属性。

答：GRANT SELECT,INSERT,DELETE,UPDATE(Phone) ON TABLE Dept TO U2;

1. U3可以读、增Dept关系，并可将这些权限转授给其他用户。

答：GRANT SELECT,INSERT ON TABLE Dept TO U3 WITH GRANT OPTION;

1. 所有用户可以读Dept关系。

答：GRANT SELECT ON TABLE Dept TO PUBLIC;

6.13什么是角色？请针对6.12题目对Dept可以读、增、删权限建立一个角色，并将这些权限授予用户U1、U2和U3。

答：角色是命名的权限集合，可以为一组具有相同权限的用户创建一个角色，简化授权的过程。

CREATE ROLE R1;

GRANT SELECT, INSERT, DELETE ON TABLE Dept TO R1;

GRANT R1 TO U1, U2, U3;

6.14在6.12题目的Emp和Dept关系上建立一个视图：

SeniorEmp(Sname, Sage, Salary, Dname)

其中Sage>50。并授予U4 用户可以读该视图的权限。请用SQL语言实现上述要求。

答：CREATE VIEW SeniorEmp(Sname, Sage, Salary, Dname)

AS SELECT Sname, Eage as Sage, Salary, Dname FROM Emp, Dept

WHERE Emp. Deptno = Dept.Deptno AND Eage>50;

GRANT SELECT ON SeniorEmp TO U4;

6.15请用SQL语句创建题目6.12中Emp和Dept关系，并有以下约束。

1. Dept的主键是Deptno,Emp的主键是Eno。
2. Emp的外键是Deptno，被参考的关系是Dept。
3. Emp的 Eage取值在20~60之间。
4. Dept的Dname是惟一的并且非空。
5. Emp的Salary >1000。

答：

CREATE TABLE Emp (

Eno CHAR(10) PRIMARY KEY,

Ename CHAR(20),

Eage INT CHECK (Eage>=20 AND Eage<=60),

Salary INT CHECK (Salary>1000),

Deptno char(5) REFERENCES Dept(Deptno)

);

CREATE TABLE Dept(

Deptno CHAR(5) PRIMARY KEY,

Dname CHAR(50) UNIQUE NOT NULL,

Phone CHAR(12),

Loc CHAR(100)

);

6.16在题目6.12中定义的Emp关系中，试用触发器表示下列完整性约束。

1. 20≤Eage≤60;

答：

CREATE TRIGGER EmpTrig\_Eage

BEFORE INSERT, UPDATE ON Emp

REFERENCING

NEW AS *NewTuple*

FOR EACH ROW

WHEN (NewTuple.Eage <20 OR NewTuple. Eage >60)

BEGIN

raise\_error('年龄不在有效范围内', 16, 1 );

END;

1. 1000≤Salary≤10000;

答：

CREATE TRIGGER EmpTrig\_Salary

BEFORE INSERT, UPDATE ON Emp

REFERENCING

NEW AS *NewTuple*

FOR EACH ROW

WHEN (NewTuple.Salary <1000 OR NewTuple.Salary >10000)

BEGIN

raise\_error('工资不在有效范围内', 16, 1 );

END;

1. 当插入或者修改 一个职工记录的时候，如果工资低于1000元则自动改为1000元。

答：

CREATE TRIGGER EmpTrig\_IUSalary

AFTER INSERT, UPDATE ON Emp

REFERENCING

NEW AS *NewTuple*

FOR EACH ROW

WHEN (NewTuple.Salary <1000)

BEGIN

UPDATE Emp SET Salary=1000 WHERE Eno=NewTuple.Eno ;

END;