

CONTENTS

1

事务



2

并发控制



3

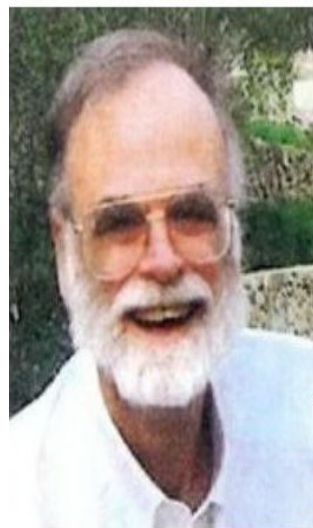
恢复与备份



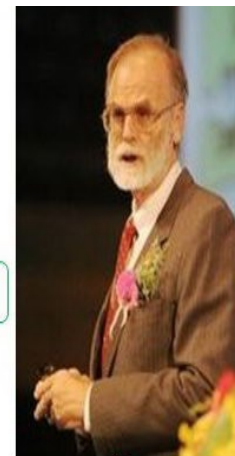
2007 年 1 月 28 日，数据库界的第三位图灵奖得主 James Gray（昵称 Jim Gray），要去做一件善事 -- 百善孝为先 -- 陪伴他的母亲。然后，然后他们就失联了。带着在数据库事务处理和系统实现方面的功勋，走了！



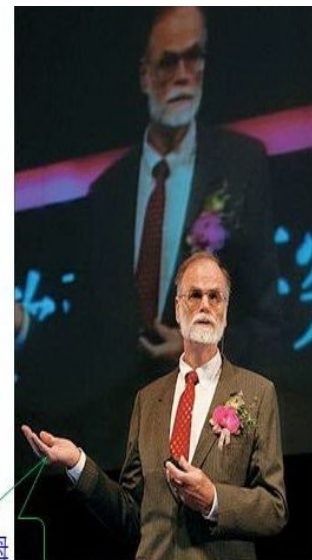
青椒 吉姆



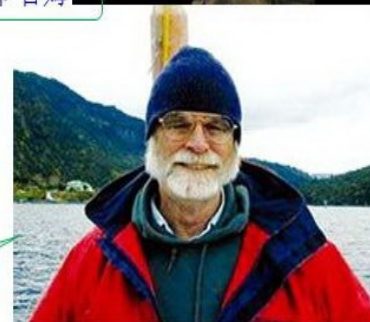
益友吉姆



良师 吉姆



大师 吉姆



航海发烧友 吉姆

■现实应用中，数据库的操作与操作之间往往具有一定的语义和关联性。数据库应用希望将这些有关联的操作当作一个逻辑工作单元看待，要么都执行，要么都不执行。

■[例 1] 飞机订票系统有两个表 Sale 和 Flight，分别记录各售票点 (agentNo) 的售票数及全部航班的剩余票数：

Sale(agentNo, flightNo, date, saledNumber)

Flight(flightNo, date, remainNumber)

买票，转账...

■现有 A0010 售票点 (agentNo) 欲出售 F005 航班 (flightNo) 2018 年 8 月 8 日 (date) 机票 2 张。

问题背景

■ 可编制如下程序：

查询 F005 航班 2018 年 8 月 8 日剩余票数 A; (1)

if (A<2)

拒绝操作，并通知票源不足 (agentNo) 的售票数及全部航班的剩余票数：

else

更新 A0010 售票点的售票数;
更新 F005 航班的剩余票数;

[例 1] 飞机订票系统有两个表 Sale 和 Flight，分别记录各售票点

Sale(agentNo, flightNo, date, soldNumber)

Flight(flightNo, date, remainNumber)

买票,转账...

如何用 SQL 语句分别实现语句 (1) 和语句 (2) ?

■ 语句 (1) 可用 SQL 语句表示为：

SELECT remainNumber

FROM Flight

WHERE flightNo='F005' AND date='2018-08-08'

- 语句 (2) 是在当 F005 航班 2018 年 8 月 8 日的剩余票数大于请求票数时更新 Sale 和 Flight 表。该更新包括两个 update 操作:

```
UPDATE Sale
```

```
SET saledNumber=saledNumber+2
```

```
WHERE agentNo='A0010' AND flightNo='F005'
```

```
AND date='2018-08-08'
```

```
UPDATE Flight
```

```
SET remainNumber=remainNumber-2
```

```
WHERE flightNo='F005' AND date='2018-08-08'
```

如果第一个 UPDATE 语句执行成功，而第二个 UPDATE 语句执行失败，会发生什么问题呢？？？

问题背景

- 假设 F005 航班共有 200 个座位，2018 年 8 月 8 日 (A0010 售出 20 张)，余票 2 张。
- 当第 1 个 UPDATE 语句执行成功时，发生故障重新提供服务时，如果又有 1 人购买了 8 日机票 2 张，由于 F005 的剩余票数未更新，又售了 2 张。
- 结果多卖了 2 张票！

Sale(agentNo, flightNo, date, saledNumber)			
agentNo	flightNo	date	saledNumber
A0010	F005	2018/8/8	20
..
Flight(flightNo, date, remainNumber)			
flightNo	date	remainNumber	
F005	2018/8/8	2	
...	

出现上述问题的原因是什么？

出现故障后，系统重新提供服务时数据库状态与现实世界状态出现了不一致。对于机票系统来说，一航班的剩余票数加上已售出票数应等于该航班全部座位数。而重新提供服务时，F005 的已售票数与剩余票数之和为 202(不等于 200!)，导致多买了 2 张。

- 微信钱包，若你要发放红包或者购物需要花费 100 元，从银行转账到微信钱包。具体流程是：
- 0： 查询您账户余额是否够？ 如够则
- 1： 您银行卡账户减去 100 （ update ）
- 2： 您微信钱包增加 100 （ update ）

更新时如果 0 ， 1 步骤成功， 2 没有成功。则会出现什么？

- 为解决上述问题，数据库管理系统引入了事务概念，它将这些有内在联系的操作当作一个逻辑单元看待，并采取相应策略保证一个逻辑单元内的全部操作要么都执行成功，要么都不执行。
- 对数据库用户而言，只需将具有完整逻辑意义的一组操作正确地定义在一个事务之内即可。

事务的概念是什么？

- 对于用户而言，事务是具有完整逻辑意义的数据库操作序列的集合。
- 对于数据库管理系统而言，事务则是一个读写操作序列。这些操作是一个不可分割的逻辑工作单元，要么都做，要么都不做。
- 事务是数据库管理系统中竞争资源、并发控制和恢复的基本单元。它是由数据库操作语言（如 SQL）或高级编程语言（如 Java、C、C++）提供的事务开始语句、事务结束语句以及由它们包含的全部数据库操作语句组成。

■事务结束的两种类型:

- 事务提交 (commit) : 将成功完成事务的**执行结果 (即更新)**永久化, 并释放事务占有的全部资源。
- 事务回滚 (rollback) : 中止当前事务、**撤销其对数据库所做的更新**, 并释放事务占有的全部资源。

DBMS 很多, 事务模式有所区别, 如 SQL Server 事务模式是这样的

。

■SQL Server 数据库提供了三种类型的事务模式：

- **显式事务**是指用户使用 Transact-SQL 事务语句所定义的事务，其事务语句包括：
 - 事务开始： **BEGIN TRANSACTION**
 - 事务提交： **COMMIT TRANSACTION** ， **COMMIT WORK**
 - 事务回滚： **ROLLBACK TRANSACTION** ， **ROLLBACK WORK**
- **隐式事务**是指事务提交或回滚后，系统自动开始新的事务。该类事务不需要采用 **BEGIN TRANSACTION** 语句标识事务的开始。
- **自动定义事务**： 当一个语句成功执行后，它被自动提交，而当执行过程中出错时，则被自动回滚。

- [例 2] 利用 SQL Server 提供的显式事务模式定义例 1 中的数据
库更新事务。

```
BEGIN TRANSACTION
```

```
UPDATE Sale
```

```
SET saledNumber=saledNumber+2
```

```
WHERE agentNo='A0010' AND flightNo='F005'
```

```
AND date='2008-08-08'
```

```
UPDATE Flight
```

```
SET remainNumber=remainNumber-2
```

```
WHERE flightNo='F005' AND date='2008-08-08'
```

```
COMMIT TRANSACTION
```

- 为了保证事务并发执行或发生故障时数据库的一致性（完整性），事务应具有以下 ACID 特性：
 - 原子性 (atomicity) 。事务的所有操作要么全部都被执行，要么都不被执行。
 - 一致性 (consistency) 。一个单独执行的事务应保证其执行结果的一致性，即总是将数据库从一个一致性状态转化到另一个一致性状态。
 - 隔离性 (isolation) 。当多个事务并发执行时，一个事务的执行不能影响另一个事务，即并发执行的各个事务不能互相干扰。
 - 持久性 (durability) 。一个事务成功提交后，它对数据库的改变必须是永久的，即使随后系统出现故障也不会受到影响。

DBMS 保证事务特性措施

- 原子性也称为故障原子性或 (故障) 可靠性
 - 由 DBMS 通过撤销未完成事务对数据库的影响来实现。
- 一致性是指单个事务的一致性，也称为并发原子性或正确性
 - 由编写该事务代码的应用程序负责，但有时也可利用 DBMS 提供的数据库完整性约束 (如触发器) 的自动检查功能来保证。
- 隔离性也称为执行原子性或可串行化，可以看作是多个事务并发执行时的一致性或正确性要求
 - 由 DBMS 的并发控制模块保证。
- 持久性也称为恢复原子性或恢复可靠性
 - 它是利用已记录在稳固存储介质 (如磁盘阵列) 中的恢复信息 (如日志、备份等) 来实现丢失数据 (如因中断而丢失的存放在主存中但还未保存到磁盘数据库中去的数据等) 的恢复。
 - 它是由 DBMS 的恢复管理模块保证。

事务并发执行

■ 数据库管理系统允许多个事务并发执行:

● 优点

- 增加系统吞吐量 (throughput) 。吞吐量是指单位时间系统完成事务的数量。当一事务需等待磁盘 I/O 时, CPU 可去处理其它正在等待 CPU 的事务。这样, 可减少 CPU 和磁盘空闲时间, 增加给定时间内完成事务的数量。
- 减少平均响应时间 (average response time) 。事务响应时间是指事务从提交给系统到最后完成所需要的时间。事务的执行时间有长有短, 如果按事务到达的顺序依次执行, 则短事务就可能会由于等待长事务导致完成时间的延长。如果允许并发执行, 短事务可以较早地完成。因此, 并发执行可减少事务的平均响应时间。

● 缺点

- 若不对事务的并发执行加以控制, 则可能破坏数据库的一致性。

本节主要讲述了事务的概念，四大特征和由事务并发执行带来的问题等。

