

《数据库系统实验》

实验报告

题目	实验十二
姓名	卜海涛
学号	22336016
班级	22 级计算机科学与技术（人工智能与大数据）

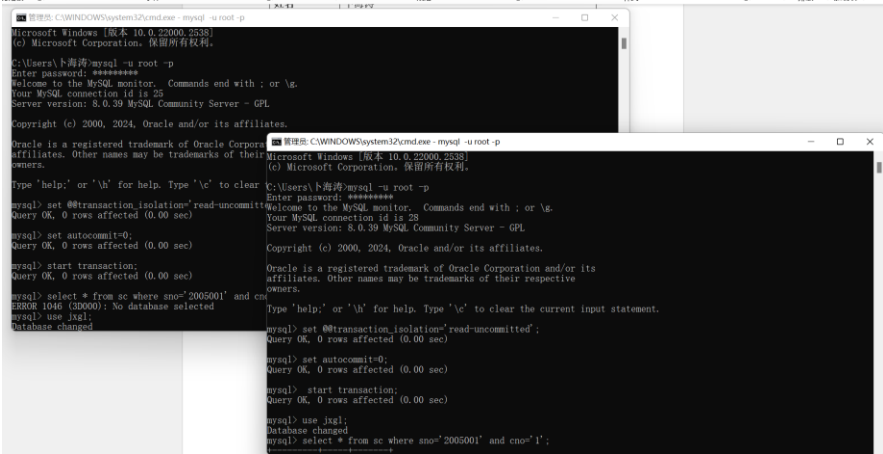
一. 实验环境:

操作系统: Windows64 位

图形界面: MySQL Workbench

二. 实验内容与完成情况:

1. 同时打开两个 cmd 窗口连接 mysql 数据库, 每个窗口执行一个事务



```
管理控制台 C:\WINDOWS\system32\cmd.exe - mysql -u root -p
Microsoft Windows [版本 10.0.22000.2538]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\卜海涛>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 25
Server version: 8.0.39 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> set @@transaction_isolation='read-uncommitted';
Query OK, 0 rows affected (0.00 sec)

mysql> set autocommit=0;
Query OK, 0 rows affected (0.00 sec)

mysql> start transaction;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from sc where sno='2005001' and cno='1';
ERROR 1046 (3D000): No database selected
mysql> use jxgl;
Database changed
mysql> select * from sc where sno='2005001' and cno='1';
+-----+-----+
| sno   | cno   |
+-----+-----+
| 2005001 | 1     |
+-----+-----+

管理控制台 C:\WINDOWS\system32\cmd.exe - mysql -u root -p
Microsoft Windows [版本 10.0.22000.2538]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\卜海涛>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 25
Server version: 8.0.39 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> set @@transaction_isolation='read-uncommitted';
Query OK, 0 rows affected (0.00 sec)

mysql> set autocommit=0;
Query OK, 0 rows affected (0.00 sec)

mysql> start transaction;
Query OK, 0 rows affected (0.00 sec)

mysql> use jxgl;
Database changed
mysql> select * from sc where sno='2005001' and cno='1';
+-----+-----+
| sno   | cno   |
+-----+-----+
| 2005001 | 1     |
+-----+-----+
```

2. 复现表 12-5

执行事务 1

代码:

```
set @@transaction_isolation='read-uncommitted';
set autocommit=0;
start transaction;
use jxgl;
select * from sc where sno='2005001' and cno='1';
```

结果:

```
mysql> set @@transaction_isolation='read-uncommitted';
Query OK, 0 rows affected (0.00 sec)

mysql> set autocommit=0;
Query OK, 0 rows affected (0.00 sec)

mysql> start transaction;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from sc where sno='2005001' and cno='1';
ERROR 1046 (3D000): No database selected
mysql> use jxgl;
Database changed
mysql> select * from sc where sno='2005001' and cno='1';
+-----+-----+-----+
| sno   | cno  | grade |
+-----+-----+-----+
| 2005001 | 1    | 87    |
+-----+-----+-----+
1 row in set (0.00 sec)
```

执行事务 2

代码:

```
set @@transaction_isolation='read-uncommitted';
set autocommit=0;
start transaction;
use jxgl;
select * from sc where sno='2005001' and cno='1';
```

结果:

```
mysql> set @@transaction_isolation='read-uncommitted';
Query OK, 0 rows affected (0.00 sec)

mysql> set autocommit=0;
Query OK, 0 rows affected (0.00 sec)

mysql> start transaction;
Query OK, 0 rows affected (0.00 sec)

mysql> use jxgl;
Database changed
mysql> select * from sc where sno='2005001' and cno='1';
+-----+-----+-----+
| sno   | cno  | grade |
+-----+-----+-----+
| 2005001 | 1    | 87    |
+-----+-----+-----+
1 row in set (0.01 sec)
```

执行事务 1

代码:

```
update sc set grade=grade+5 where sno='2005001' and cno='1';
select * from sc where sno='2005001' and cno='1';
```

结果:

```
mysql> update sc set grade=grade+5 where sno='2005001' and cno='1';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from sc where sno='2005001' and cno='1';
+-----+-----+-----+
| sno   | cno  | grade |
+-----+-----+-----+
| 2005001 | 1    | 92    |
+-----+-----+-----+
1 row in set (0.00 sec)
```

此时 sc 表在事务 1 中被更新，但是还未提交到数据库

执行事务 2

代码：

```
select * from sc where sno='2005001' and cno='1';
```

结果：

```
mysql> select * from sc where sno='2005001' and cno='1';
+-----+-----+-----+
| sno   | cno  | grade |
+-----+-----+-----+
| 2005001 | 1    | 92    |
+-----+-----+-----+
1 row in set (0.00 sec)
```

事务 2 读取了事务 1 还未提交的更新结果

由于两个事务采用的隔离级别为 **read-uncommitted**（未提交读），当一个事务的更新结果还未提交时，另一个事务仍然能够从缓存中读到其更新的结果，从而产生了脏读，这违反了数据库的隔离性

执行事务 1

代码：

```
rollback;
```

结果：

```
mysql> rollback;
Query OK, 0 rows affected (0.01 sec)
```

执行事务 2

代码：

```
select * from sc where sno='2005001' and cno='1';
commit;
```

结果：

```
mysql> select * from sc where sno='2005001' and cno='1';
+-----+-----+-----+
| sno   | cno  | grade |
+-----+-----+-----+
| 2005001 | 1    | 87    |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> commit;
Query OK, 0 rows affected (0.00 sec)
```

同理，由于事务 1 的隔离级别为 read-uncommitted，因此事务 2 能读到其 rollback 后的数据

3. 复现表 12-6

执行事务 1

代码：

```
set @@transaction_isolation='read-uncommitted';
set autocommit=0;
start transaction;
use jxgl;
select * from sc where sno='2005001' and cno='1' for update;
```

结果：

```
mysql> set @@transaction_isolation='read-uncommitted';
Query OK, 0 rows affected (0.00 sec)

mysql> set autocommit=0;
Query OK, 0 rows affected (0.00 sec)

mysql> start transaction;
Query OK, 0 rows affected (0.00 sec)

mysql> use jxgl;
Database changed
mysql> select * from sc where sno='2005001' and cno='1' for update;
+-----+-----+-----+
| sno   | cno  | grade |
+-----+-----+-----+
| 2005001 | 1    | 87    |
+-----+-----+-----+
1 row in set (0.00 sec)
```

事务 1 通过 for update 设置了排他锁，上锁之后事务 2 既不可读也不可写，需要等待事务 1 释放锁才能继续执行

执行事务 2

代码：

```
set @@transaction_isolation='read-uncommitted';
set autocommit=0;
start transaction;
use jxgl;
select * from sc where sno='2005001' and cno='1' lock in share mode;
```

结果:

```
mysql> set @@transaction_isolation='read-uncommitted';
Query OK, 0 rows affected (0.00 sec)

mysql> set autocommit=0;
Query OK, 0 rows affected (0.00 sec)

mysql> start transaction;
Query OK, 0 rows affected (0.00 sec)

mysql> use jxgl;
Database changed
mysql> select * from sc where sno='2005001' and cno='1' lock in share mode;
ERROR 1205 (HY000): Lock wait timeout exceeded; try restarting transaction
```

由于事务 1 使用了排他锁, 事务 2 需等待事务 1 释放锁后才能执行查询操作, 而事务 2 在查询时使用了共享锁, 即在查询过程中不允许其他事务的写操作, 共享锁在等待一段时间后未获取结果就报错

执行事务 1

代码:

```
update sc set grade=grade+5 where sno='2005001' and cno='1';
select * from sc where sno='2005001' and cno='1';
rollback;
```

结果:

```
mysql> update sc set grade=grade+5 where sno='2005001' and cno='1';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from sc where sno='2005001' and cno='1';
+-----+-----+-----+
| sno   | cno  | grade |
+-----+-----+-----+
| 2005001 | 1    | 92    |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> rollback;
Query OK, 0 rows affected (0.00 sec)
```

执行事务 2

代码:

```
select * from sc where sno='2005001' and cno='1' lock in share mode;
commit;
```

结果:

```
+----+-----+-----+
| sno | cno | grade |
+----+-----+-----+
| 2005001 | 1 | 87 |
+----+-----+-----+
1 row in set (4.38 sec)

mysql> select * from sc where sno='2005001' and cno='1' lock in share mode;
+----+-----+-----+
| sno | cno | grade |
+----+-----+-----+
| 2005001 | 1 | 87 |
+----+-----+-----+
1 row in set (0.00 sec)

mysql> commit;
Query OK, 0 rows affected (0.00 sec)
```

事务 1 回滚之后排他锁才被释放，事务 2 获得锁后继续执行未完成的查询操作，这样事务 2 就不会读到事务 1 修改但还未提交的数据，避免了脏读的发生