



数据库系统实验报告

作业名称: SQL 数据完整性

姓 名: 汪珉凯

学 号: 3220100975

电子邮箱: 3220100975@zju.edu.cn

联系电话: 18157421318

指导老师: 孙建伶

2024 年 3 月 23 日

SQL 数据完整性

一、实验目的

熟悉 SQL 数据完整性控制的方法。

二、实验环境

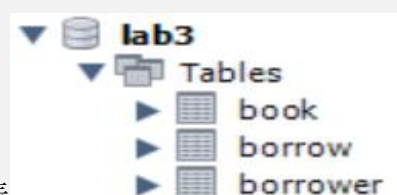
MySQL.

三、实验流程

1.创建新的数据库 lab3，然后创建 3 张表格，定义具体如下：

```
1  USE lab3;
2
3  ● ○ CREATE TABLE book(
4      id INT,
5      publish_year INT,
6      title VARCHAR (50),
7      author VARCHAR (50),
8      publisher VARCHAR (50),
9
10     PRIMARY KEY (id),
11     CHECK (id > 0),
12     CHECK (publish_year > 1900 )
13 );
14
15 ● ○ CREATE TABLE borrower(
16     mem_number INT,
17     borrower_name VARCHAR(50),
18
19     PRIMARY KEY (mem_number)
20 );
21
22 ● ○ CREATE TABLE borrow(
23     id INT,
24     mem_number INT,
25     borrow_date DATE,
26
27     PRIMARY KEY (id, mem_number, borrow_date),
28     FOREIGN KEY (id) REFERENCES book (id),
29     FOREIGN KEY (mem_number) REFERENCES borrower (mem_number) ,
30     CHECK (id > 0)
31 );
```

三张表格分别是：



1) 书目表 2) 读者表 3) 借阅表

2.考察 primary key 对数据完整性控制的影响:

先向表格中正常插入一些数据, 具体如下所示:

```
1 • INSERT INTO book (id, publish_year, title, author, publisher)
2   VALUES
3     (1, 1998, 'Harry Potter ', 'J.K. Rowling', 'Bloomsbury'),
4     (2, 2005, 'The Da Vinci Code', 'Dan Brown', 'Doubleday'),
5     (3, 1984, '1984', 'George Orwell', 'Secker & Warburg'),
6     (4, 1960, 'To Kill a Mockingbird', 'Harper Lee', 'J. B. Lippincott & Co. ');
7
8 • INSERT INTO borrower (mem_number, borrower_name)
9   VALUES
10    (1001, 'John Smith'),
11    (1002, 'Emily Johnson'),
12    (1003, 'Michael Williams');
13
14 • INSERT INTO borrow (id, mem_number, borrow_date)
15   VALUES
16    (1, 1001, '2022-03-15'),
17    (2, 1002, '2022-03-17'),
18    (3, 1003, '2022-03-19');
```

再尝试向表中插入数据以考验 primary key 对数据完整性的约束:

```
1 • INSERT INTO book VALUES (1,2008,'Twilight','Stephanie Meyer','Bloomsbury');
```

尝试向 book 表中插入一条已有 id (主键) 的数据, 出现如下报错:

```
✖ 8 00:26:12 INSERT INTO book VALUES (1,2008,'Twilight','Stephanie Meyer','Bloomsbury')
```

Error Code: 1062. Duplicate entry '1' for key 'book.PRIMARY'

00:26:12 INSERT INTO book VALUES

(1,2008,'Twilight','Stephanie Meyer','Bloomsbury');

Error Code: 1062. Duplicate entry '1' for key 'PRIMARY' 0.000 sec

3.删除被引用的行, 考察 foreign key 对数据完整性的影响:

执行以下代码:

```
1 • delete from book where title = 'Harry Potter ' ;
```

但是由上述插入的数据可知: 《哈利波特》这本书的 id 是 1, 刚好作为 borrow 表的外键被引用。代码执行后得到如下结果:

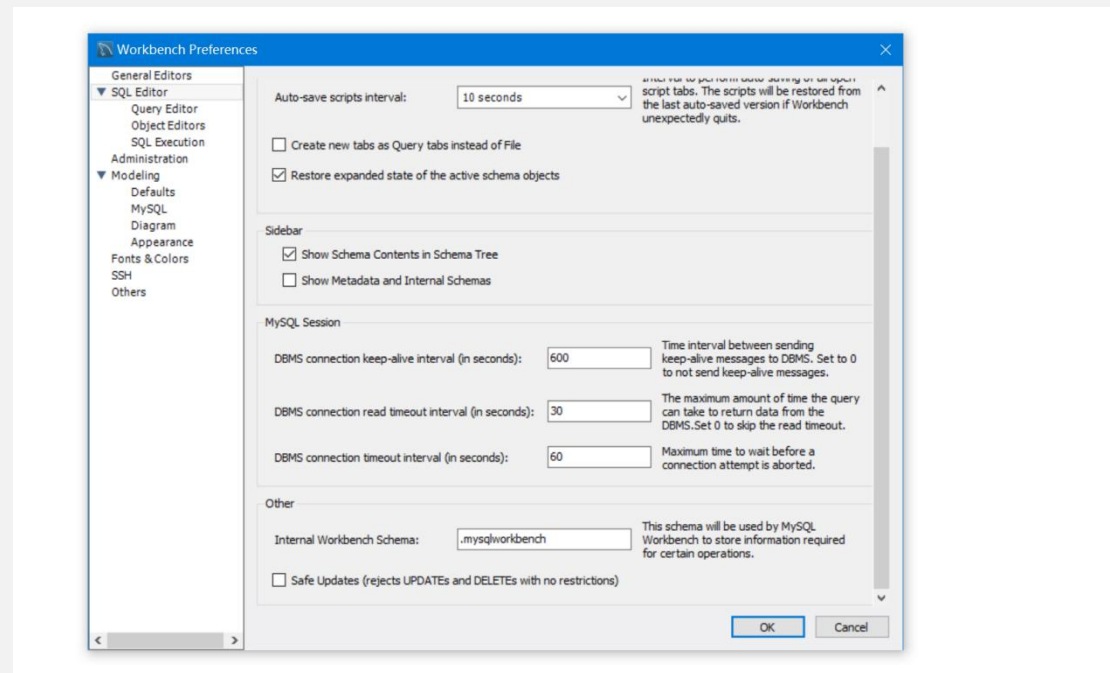
#	Time	Action
✖ 1	12:53:59	delete from book where title = 'Harry Potter '

Message

Error Code: 1175. You are using safe update mode and you tried to update a table without a WHERE that uses a KEY column. To disable safe mo

12:53:59 delete from book where title = 'Harry Potter ' Error Code: 1175. You are using safe update mode and you tried to update a table without a WHERE that uses a KEY column. To disable safe mode, toggle the option in Preferences -> SQL Editor and reconnect. 0.000 sec

接下来关闭 safe mode，重联数据库。



重新执行以上代码，结果如下：



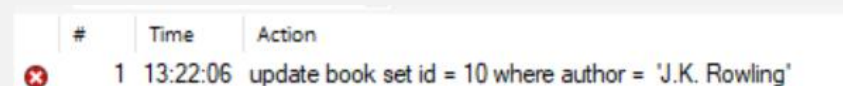
13:13:16 delete from book where title = 'Harry Potter ' Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails ('lab3`.`borrow`, CONSTRAINT 'borrow_ibfk_1' FOREIGN KEY ('id') REFERENCES 'book' ('id')) 0.016 sec

可以看到由于被删除的数据有作为外键被引用，删除操作失败。

4.修改 primary key 的值，考察 foreign key 中 on update 子句如何控制参照完整性：执行以下代码：

```
1 • update book set id = 10 where author = 'J.K. Rowling' ;
```

然而，被修改的这条信息被作为外键引用，因此会得到如下结果：



Message
Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (lab3`.`borrow`, CONSTRAINT `borrow_ibfk_1` FOREIGN KEY ... 0.000 sec

13:22:06 update book set id = 10 where author = 'J.K. Rowling' Error Code:
1451. Cannot delete or update a parent row: a foreign key constraint fails
(`lab3`.`borrow`, CONSTRAINT `borrow_ibfk_1` FOREIGN KEY (`id`)
REFERENCES `book` (`id`)) 0.000 sec

可以看到由于被修改的数据有作为外键被引用，修改操作失败。

5.修改或插入表中数据，考察 check 子句如何控制校验完整性。

执行以下代码：

```
1 •   INSERT INTO book VALUES (8,1320,'Divine Comedy','Dante Alighieri','none');
```

得到如下结果：

#	Time	Action
1	13:39:44	INSERT INTO book VALUES (8,1320,'Divine Comedy','Dante Alighieri','none')

Message
Error Code: 3819. Check constraint 'book_chk_2' is violated.

13:36:42 INSERT INTO book VALUES (8,1320,'Divine Comedy','Dante
Alighieri','none') Error Code: 3819. Check constraint 'book_chk_2' is violated.0.000
sec

可以看到，插入 book 表的数据不满足第二项 check 条件（即出版年份是在 20 世纪以后），因此插入失败。

6.定义一个 asseration, 并通过修改表中数据考察断言如何控制数据完整性。

```
1 ✖   CREATE asseration mem check (  
2       not exist (select * from book  
3       Where publish_year = 1960)  
4   );
```

然而可以看到：MySQL 语言并不支持 asseration 语法，因此只能用 trigger 代替。

7.定义一个 trigger, 并通过修改表中数据考察触发器如何起作用。

执行如下代码：

```

1  Delimiter $$
2  • Create trigger pub_year_check
3      After update on book
4      For each row
5  Begin
6      Update borrower set borrower_name = 'Adam'
7      where borrower.mem_number = 1001 ;
8  end; $$
9  Delimiter ;

```

该触发器的功能是：设置一个名为 pub_year_check 的触发器，当 book 发生更新时，将 borrower 表中 mem_number 是 1001 的那一个元组的 borrower_name 设置为 Adam。

得到结果如下：

#	Time	Action	Message
1	14:45:56	Create trigger pub_year_check After update on book For each row Begin Update borrower set borrower_name = 'Adam'...	0 row(s) affected

现在更新 book 的值，代码如下：

```
1 update book set author = 'Joanne Rowling ' where title = 'Harry Potter ';
```

得到结果如下：《哈利波特》的借阅者从 John Smith 变成了 Adam。

原始数据	触发器生效后																														
<table><tr><th></th><th>mem_number</th><th>borrower_name</th></tr><tr><td>▶</td><td>1001</td><td>John Smith</td></tr><tr><td></td><td>1002</td><td>Emilyb Johnson</td></tr><tr><td></td><td>1003</td><td>Michael Williams</td></tr><tr><td>●</td><td>NULL</td><td>NULL</td></tr></table>		mem_number	borrower_name	▶	1001	John Smith		1002	Emilyb Johnson		1003	Michael Williams	●	NULL	NULL	<table><tr><th></th><th>mem_number</th><th>borrower_name</th></tr><tr><td>▶</td><td>1001</td><td>Adam</td></tr><tr><td></td><td>1002</td><td>Emilyb Johnson</td></tr><tr><td></td><td>1003</td><td>Michael Williams</td></tr><tr><td>●</td><td>NULL</td><td>NULL</td></tr></table>		mem_number	borrower_name	▶	1001	Adam		1002	Emilyb Johnson		1003	Michael Williams	●	NULL	NULL
	mem_number	borrower_name																													
▶	1001	John Smith																													
	1002	Emilyb Johnson																													
	1003	Michael Williams																													
●	NULL	NULL																													
	mem_number	borrower_name																													
▶	1001	Adam																													
	1002	Emilyb Johnson																													
	1003	Michael Williams																													
●	NULL	NULL																													

四、遇到的问题及解决方法

遇到的问题：在使用触发器时，一开始始终无法成功更新 book 表格的内容，给出的报错理由是 foreign key 的限制，但我并没有修改 book 表格中被作为外键引用的相关数据。

解决方法：后来发现并不是修改 book 表本身会不满足 foreign key 的限制，而是触发器造成的更新操作会改变 borrower 表中，被作为外键引用的 mem_number 值发生了更新，因此违背了 foreign key 原则。所以只需要修改触发器被触发后的操作即可。

五、总结

本次实验我熟悉了 mysql 中各种数据完整性约束方式，其中触发器的使用由于在上课期间并没有学习，因此比较陌生，花了较多时间学习，但是现在已经对这些约束方式较为熟悉，且对 workbench 本身也有了更深的了解，希望再接再厉。