

Database System Lab 3

学号：3170100186

姓名：王俊

实验目的

- Familiar with the method of data integrity control through SQL

实验平台：

- MySQL（全平台）

网上教程 <http://www.runoob.com/mysql/mysql-tutorial.html>

实验内容

- 1) Define several tables, including the definition of primary key, foreign key, and check.
- 2) Insert data into the table and examine how the primary key controls entity integrity.
- 3) Delete the rows in the referenced table, and examine how the on delete clause in the foreign key controls the referential integrity.
- 4) Modify the primary key of the row in the referenced table, and examine how the on update clause in the foreign key controls the referential integrity.
- 5) Modify or insert data in the table, and examine how the check clause controls the integrity of the check
- 6) Define an assertion and examine how assertions control data integrity

by modifying the data in the table

7) Define a trigger and examine how the trigger works by modifying the data in the table

8) Complete the experiment report

实验指导

(根据实验说明中的实验步骤，将每个步骤中输入的 sql 语句和输出结果截图)

Integrity constraints ensure that the integrity of the data will not be destroyed when the database is modified. The integrity constraints include the following aspects:

- Constraints on a single relationship (not null, unique, check, default)

Example:

```
create table student (  
    sno char(12) primary key,  
    name varchar(20) not null,  
    city varchar(29) default 'Hangzhou',  
    gender char(1),  
    check ( gender in ('m', 'f') )  
);
```

(mysql does not support check, so we use enum)

- referential integrity:

By using a foreign key, a field in one table refers to another table, and you can specify how the corresponding row of the reference table should react when the data in the referenced table changes. You can use on delete

[action] or on update [action] to specify the action performed by the corresponding row in the referenced table when the row in the referenced table is deleted or updated, respectively. actions include:

- **no action:** When a row is deleted or updated, if an existing row in another table references the key of the row, an error is generated and rolled back. (When no on delete or on update is specified, the default is no action);
- **Cascas:** When a row is deleted or updated, if an existing row in another table references the key of that row, the corresponding row in other tables will be deleted or updated together.

There are other actions such as set null and set default. When the corresponding row of the referenced table is deleted, the foreign key can be set to the specified value, which will not be described in detail here.

Example of use:

```
create table exam (  
    sno char(12) unique,  
    score int,  
    foreign key (sno) references student  
        on delete cascade  
);
```

- **assertion**

The assertion expresses the constraints on the database. The constraints on the single relationship and referential integrity are special forms of assertion. You can express more and more complex constraints by creating

new assertions. The format for creating assertion is:

`create assertion <assertion_name> check <predicate>;`

(But MySQL does not support assertion, just for understanding)

- **trigger**

trigger specifies the command to be executed when the database is modified. In SQL Server, it is divided into after trigger and instead of trigger. The after trigger specifies the action to be performed after the update / insert / delete operation; the instead of trigger does not perform the original update / insert / delete operation when the corresponding action is triggered, and instead performs the user-specified action. MySQL is slightly different, divided into after triggers and before triggers. Students are advised to consult the relevant materials first to understand the trigger principle and writing method before performing this experiment.

实验步骤

(根据实验说明中的实验步骤，将每个步骤中输入的 sql 语句和输出结果截图)

1) On the basis of the table created in Experiment 2, add the following constraints: (Add the constraint condition using the statement `ALTER TABLE <table_name> ADD CONSTRAINT <constraint_name> <constraint>`):

Actual Code:

Since when I built the table, I have already took this into consideration and have no need to change it now:

```
create table book
(
    bno int NOT NULL,
    title char(100) NOT NULL,
    author char(100) NOT NULL,
    year year(4) NOT NULL,
    press char(100) NOT NULL,
    price Double(10,2) NOT NULL,
    total int NOT NULL,
    stock int NOT NULL,
    PRIMARY KEY(bno)
);
```

```
create table card
(
    cno int NOT NULL,
    name char(25) NOT NULL,
    department char(100) NOT NULL,
    type enum('S','T') NOT NULL,
    PRIMARY KEY(cno)
);
```

```
create table borrow
(
    cno int NOT NULL,
    bno int NOT NULL,
    borrow_date date,
    return_date date,
    FOREIGN KEY (bno) REFERENCES book(bno) on delete cascade,
    FOREIGN KEY (cno) REFERENCES card(cno) on update cascade
);
```

a) The book table uses bno as the primary key

```
ALTER TABLE book ADD CONSTRAINT book_no PRIMARY KEY(bno);
```

Because I have already created one, it would come out with a warning:

```
mysql> ALTER TABLE book ADD CONSTRAINT book_no PRIMARY KEY(bno);  
ERROR 1068 (42000): Multiple primary key defined
```

b) The card table uses cno as the primary key, and the type field can only take values in 'T' and 'S' (the enum field type can be used instead of the check statement)

```
ALTER TABLE card ADD CONSTRAINT card_no PRIMARY KEY(cno);  
mysql> ALTER TABLE card ADD CONSTRAINT card_no PRIMARY KEY(cno);  
ERROR 1068 (42000): Multiple primary key defined
```

Because I have already created one, it would come out with a warning:

```
ALTER TABLE card modify column type enum('S','T') NOT NULL;  
mysql> ALTER TABLE card modify column type enum('S','T') NOT NULL;  
Query OK, 0 rows affected (0.01 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

c) In the borrow table, bno is used as the foreign key to refer to the book table. When the record in the book table is deleted, the corresponding record in the borrow table is also deleted (cascade delete); cno is used as the foreign key to refer to the card table. When the record is updated, the corresponding record in the borrow table is also updated (cascade update).

Code:

```
ALTER TABLE borrow ADD CONSTRAINT book_no FOREIGN KEY(bno) REFERENCES book(bno) on delete cascade;  
ALTER TABLE borrow ADD CONSTRAINT card_no FOREIGN KEY(cno) REFERENCES card(cno) on update cascade;
```

```
mysql> ALTER TABLE borrow ADD CONSTRAINT book_no FOREIGN KEY(bno) REFERENCES book(bno) on delete cascade;  
Query OK, 4 rows affected (0.05 sec)  
Records: 4 Duplicates: 0 Warnings: 0
```

```
mysql> ALTER TABLE borrow ADD CONSTRAINT card_no FOREIGN KEY(cno) REFERENCES card(cno) on update cascade;  
Query OK, 4 rows affected (0.04 sec)  
Records: 4 Duplicates: 0 Warnings: 0
```


2) Insert the following line into the book table to see the system prompt:

3	Solid State Physics	Ashcroft	1976	Thomson Learning	98	2	2
---	---------------------------	----------	------	---------------------	----	---	---

```
INSERT INTO book (bno,title,author,YEAR,press,price,total,stock)
VALUES (3, 'Solid State Physics', 'Ashcroft', '1976','Thomson Learning', 98.00, 2, 2);
```

```
mysql> INSERT INTO book (bno,title,author,YEAR,press,price,total,stock)
-> VALUES (3, 'Solid State Physics', 'Ashcroft', '1976', 'Thomson Learning', 98.00, 2, 2);
ERROR 1062 (23000): Duplicate entry '3' for key 'PRIMARY'
```

There are two duplicate primary keys

3) Insert the following line into the card table to see the system prompt:

5	Andrew	Math	K
---	--------	------	---

Original:

```
mysql> select * from card;
+----+-----+-----+-----+
| cno | name  | department | type |
+----+-----+-----+-----+
| 1   | Jack  | CS         | T    |
| 2   | Lily  | Art        | S    |
| 3   | John  | CS         | S    |
| 4   | Lucy  | Physics    | T    |
+----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Insert:

```
INSERT INTO card (cno,name,department,type)
VALUES (5,'Andrew','Math','K');
```

```
mysql> INSERT INTO card (cno,name,department,type)
-> VALUES (5,'Andrew','Math','K');
ERROR 1265 (01000): Data truncated for column 'type' at row 1
```

Data type does not match

4) Can I delete Lily's library card in the card table? why?

No, because it's the main table, as a foreign key to another table.

So you need to delete the data in borrow first, and then it is possible for you to delete it. Otherwise, you cannot delete it because there has a foreign key constraint.

```
delete from card where name = 'Lily';
```

```
mysql> delete from card where name = 'Lily';
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails
RAINT `borrow_ibfk_1` FOREIGN KEY (`cno`) REFERENCES `card` (`cno`))
```

5) In the book table, change the bno of the Computer Network book to 0004. Check the system prompt. Why is this happening?

If we want to update this field successfully, and the corresponding record of the borrow table can also be updated automatically, how should we define the borrow table?

Check data in book:

```
mysql> select * from book;
```

bno	title	author	year	press	price	total	stock
1	Database System Concepts	Abraham	2011	McGrawHill	99.00	5	4
2	Modern Operating Systems	Andrew	2011	Pearson	75.00	3	1
3	Computer Network	Tanenbaum	2000	Pearson	58.00	4	3

```
#修改bno的名字;
update book set bno = 0004 where title = 'Computer Network';
```



```
mysql> update book set bno = 0004 where title = 'Computer Network';
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint f
RAINT `book_no` FOREIGN KEY (`bno`) REFERENCES `book` (`bno`) ON DELETE CASCADE)
mysql>
```

The reason: it lacks the constraint about on update cascade, you need to add it.

In order to do that, first, you need to drop the constraint, and add a new constraint;

Code:

```
ALTER TABLE borrow drop Foreign Key book_no;

ALTER TABLE borrow ADD CONSTRAINT book_no FOREIGN KEY(bno)
REFERENCES book(bno) on delete cascade on update cascade;
```

```
mysql> ALTER TABLE borrow drop Foreign Key book_no;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> ALTER TABLE borrow ADD CONSTRAINT book_no FOREIGN KEY(bno) REFERENCES book(bno) on delete cascade on update cascade;
Query OK, 4 rows affected (0.02 sec)
Records: 4 Duplicates: 0 Warnings: 0
```

Code:

```
mysql> update book set bno = 0004 where title = 'Computer Network';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

```
#修改bno的名字;
update book set bno = 0004 where title = 'Computer Network';
```

After changing:

```
mysql> select * from book;
```

bno	title	author	year	press	price	total
1	Database System Concepts	Abraham	2011	McGrawHi11	99.00	5
2	Modern Operating Systems	Andrew	2011	Pearson	75.00	3
4	Computer Network	Tanenbaum	2000	Pearson	58.00	4

```
3 rows in set (0.00 sec)
```

```
mysql> select * from borrow;
```

cno	bno	borrow_date	return_date
1	1	2018-01-01	2018-01-14
1	2	2018-01-06	2018-01-10
2	2	2018-02-03	2018-02-08
3	4	2018-02-05	2018-03-01

```
4 rows in set (0.01 sec)
```

6) Create a trigger so that when inserting a record into the borrow table, first check whether the book in the book table is in stock. If the inventory is not 0, then successfully lend it and reduce the book inventory in the book table by 1 ; If the inventory is 0, the lending operation is rejected (an exception can be thrown using the SIGNAL SQLSTATE statement), and the inventory is unchanged.

(Hint: you can use after trigger or before trigger, it is recommended to search the relevant information on the Internet first, familiar with the principle and use of trigger).

After setting the trigger, first try to borrow a book with sufficient inventory, and then check whether the book inventory has changed, and whether a record has been inserted into the borrow table. Then, try to borrow a book with 0 inventory, and then check The above information. In both cases, screenshots were taken and written in the experimental report.

After trigger code:

```
DELIMITER $$
create trigger borrow_afterinsert AFTER INSERT on borrow for each row
begin
    declare msg char(200);
    if (new.bno not in (select bno from book)) then
        set msg = 'book does not exist';
        SIGNAL SQLSTATE 'HY000' SET MESSAGE_TEXT = msg;
    elseif (new.cno not in (select cno from card)) then
        set msg = 'card number does not exist';
        SIGNAL SQLSTATE 'HY000' SET MESSAGE_TEXT = msg;
    elseif (new.bno in (select bno from book where stock=0)) then
        set msg = 'Sorry,stock is empty';
        SIGNAL SQLSTATE 'HY000' SET MESSAGE_TEXT = msg;
    else
        update book set stock = stock -1 where bno = new.bno;
    end if;
end$$

DELIMITER ;
```

Test:

```
mysql> DELIMITER $$
mysql> create trigger borrow_afterinsert AFTER INSERT on borrow for each row
-> begin
->     declare msg char(200);
->     if (new.bno not in (select bno from book)) then
->         set msg = 'book does not exist';
->         SIGNAL SQLSTATE 'HY000' SET MESSAGE_TEXT = msg;
->     elseif (new.cno not in (select cno from card)) then
->         set msg = 'card number does not exist';
->         SIGNAL SQLSTATE 'HY000' SET MESSAGE_TEXT = msg;
->     elseif (new.bno in (select bno from book where stock=0)) then
->         set msg = 'Sorry,stock is empty';
->         SIGNAL SQLSTATE 'HY000' SET MESSAGE_TEXT = msg;
->     else
->         update book set stock = stock -1 where bno = new.bno;
->     end if;
-> end$$
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
```

● First situation:

Check the book:

```
mysql> select * from book;
```

bno	title	author	year	press	price	total	stock
1	Database System Concepts	Abraham	2011	McGrawHill	99.00	5	4
2	Modern Operating Systems	Andrew	2011	Pearson	75.00	3	1
4	Computer Network	Tanenbaum	2000	Pearson	58.00	4	3

insert

```
INSERT INTO borrow (cno,bno,borrow_date,return_date)
VALUES (3,2,'2019-3-28','2019-3-29');
```

```
mysql> INSERT INTO borrow (cno,bno,borrow_date,return_date)
-> VALUES (3,2,'2019-3-28','2019-3-29');
Query OK, 1 row affected (0.01 sec)
```

Check in book and borrow;

```
mysql> select * from book;
```

bno	title	author	year	press	price	total	stock
1	Database System Concepts	Abraham	2011	McGrawHill	99.00	5	4
2	Modern Operating Systems	Andrew	2011	Pearson	75.00	3	0
4	Computer Network	Tanenbaum	2000	Pearson	58.00	4	3

3 rows in set (0.00 sec)

```
mysql> select * from borrow;
```

cno	bno	borrow_date	return_date
1	1	2018-01-01	2018-01-14
1	2	2018-01-06	2018-01-10
2	2	2018-02-03	2018-02-08
3	2	2019-03-28	2019-03-29
3	4	2018-02-05	2018-03-01

- **Second situation:**

The stock of bno 2 now is 0!!!

Second insert:

```
INSERT INTO borrow (cno,bno,borrow_date,return_date)
VALUES (5,2,'2019-3-28','2019-3-29');
```

Since the book is empty we will get a warning;

```
mysql> INSERT INTO borrow (cno,bno,borrow_date,return_date)
-> VALUES (5,2,'2019-3-28','2019-3-29');
ERROR 1644 (HY000): Sorry, stock is empty
```

Check in borrow:

The record does not exist:

```
mysql> select * from borrow;
```

cno	bno	borrow_date	return_date
1	1	2018-01-01	2018-01-14
1	2	2018-01-06	2018-01-10
2	2	2018-02-03	2018-02-08
3	2	2019-03-28	2019-03-29
3	4	2018-02-05	2018-03-01

```
5 rows in set (0.00 sec)
```

实验感想

1. Can't add primary key repeatedly

Original:

```
create table book
(
    bno int NOT NULL,
    title char(100) NOT NULL,
    author char(100) NOT NULL,
    year year(4) NOT NULL,
    press char(100) NOT NULL,
    price Double(10,2) NOT NULL,
    total int NOT NULL,
    stock int NOT NULL,
    PRIMARY KEY(bno)
);
```

```
mysql> ALTER TABLE book ADD CONSTRAINT book_no PRIMARY KEY(bno);
ERROR 1068 (42000): Multiple primary key defined
```

2. After changing the foreign key, it has another error:

Cannot add or update a child row: a foreign key constraint fails

(`no3170100186`.`borrow`, CONSTRAINT `borrow_ibfk_2` FOREIGN KEY

`(`bno`) REFERENCES `book` (`bno`))`

```
mysql> update book set bno = 0004 where title = 'Computer Network';  
ERROR 1452 (23000): Cannot add or update a child row: a foreign key  
T `borrow_ibfk_2` FOREIGN KEY (`bno`) REFERENCES `book` (`bno`))
```

Solution:

Drop the table and rebuild one;

Because at first when i created the table, it has an original foreign key which automatically name itself an index. And you need to drop it and create a new one

```
mysql> ALTER TABLE borrow drop Foreign Key book_ibfk_2;  
ERROR 1091 (42000): Can't DROP 'book_ibfk_2'; check that column/key exists  
mysql> Alter table borrow drop foreign key `borrow_ibfk_2`;  
Query OK, 0 rows affected (0.01 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

3. Spelling problem:

```
mysql> slect * from borrow;  
ERROR 1064 (42000): You have an error in your SQL synt  
for the right syntax to use near 'slect * from borrow
```

Use end if to Substitute endif:

```
    else  
        update book  
        endif;  
    end$$  
64 (42000): You have  
nd' at line 13
```

4. Many problems were found in the process of using the database:

```
mysql> insert into book  
-> (bno,title,author,year,press,price,total,stock)  
-> VALUES(1, ' Database System Concepts' , ' Abraham' , '  
ERROR 1064 (42000): You have an error in your SQL syntax; che  
for the right syntax to use near 'System Concepts' , ' Abrah
```


- Because there does not exist a use of '=' to judge the equal;

You need to change it into '=';

```
ERROR 1064 (42000): You have an error in your SQL syntax;
for the right syntax to use near '==0)) then
    set msg = '图书库存为0';
    SIGNAL SQLSTATE' at line 8
```

- Because the quotation marks is not in the right form and it will get to this:



And in this case, you need to input a quotation to get back to the original mode;

```
'> -
'> ;
'> ;
```

- The capitalization is the same in the code, but the table names and column names are different.

```
+-----+-----+-----+-----+
| Cno   | bno   | Borrow_date | Return_date |
+-----+-----+-----+-----+
```

4.when using add and drop like this:

```
alter table book add column category char(100);
alter table book drop column category char(100);
```

```
ERROR 1064 (42000): You have an error in your SQL syntax;
for the right syntax to use near 'char(100)' at line 1
```

Drop can not add the type of data in the end;

5.when use constraint like it :

```
return_date date,
CONSTRAINT borrow_ID PRIMARY KEY (cno,bno),
mysql> INSERT INTO borrow (cno,bno,borrow_date,return_date)
-> VALUES (1,2,'2019-3-28','2019-3-29');
ERROR 1062 (23000): Duplicate entry '1-2' for key 'PRIMARY'
```

Error would happen, because the primary key is wrong;

6.chinese can not be shown, you need to change it into english:

```

DELIMITER $$
create trigger borrow_afterinsert AFTER INSERT on borrow for each row
begin
  declare msg char(200);
  if (new.bno not in (select bno from book)) then
    set msg = '图书不存在';
    SIGNAL SQLSTATE 'HY000' SET MESSAGE_TEXT = msg;
  elseif (new.cno not in (select cno from card)) then
    set msg = '图书证不存在';
    SIGNAL SQLSTATE 'HY000' SET MESSAGE_TEXT = msg;
  elseif (new.bno in (select bno from book where stock=0)) then
    set msg = '图书库存为0';
    SIGNAL SQLSTATE 'HY000' SET MESSAGE_TEXT = msg;
  else
    update book set stock = stock -1 where bno = new.bno;
  end if;
end$$

DELIMITER ;

```

```

mysql> INSERT INTO borrow (cno,bno,borrow_date,return_date)
-> VALUES (5,2,'2019-3-28','2019-3-29');
ERROR 1366 (HY000): Incorrect string value: '\xCD\xBC\xCA\xE9\xBF\xE2...' for column 'msg' at row 1

```

After changing into english:

```

mysql> INSERT INTO borrow (cno,bno,borrow_date,return_date)
-> VALUES (5,2,'2019-3-28','2019-3-29');
ERROR 1644 (HY000): Sorry,stock is empty

```

7.The use of DELIMITER:

The delimiter command specifies the terminator of the mysql interpreter command line. The default is ";"

To put it plainly is to tell where the command ends, and you can execute this command.

But generally there are multiple semicolons in the stored procedure. We don't want to execute the command as soon as we encounter the semicolon, so we can use the delimiter command to specify other terminator instead of ";"

This terminator can be defined by yourself. Commonly used are "/" and "\$\$"

```
DELIMITER $$
```

You can use \$\$ to end it.

end\$\$

Pay attention to change it back.

DELIMITER ;