实验——安全性和完整性

实验环境

SQL Server + SSMS + 交互式SQL使用的图书馆数据库。

> LibraryDB:
>
> - Book(book_id, title, isbn, publish_year, publisher_id, category, price, stock)
> - Author(author_id, name, birth_date, nationality)
> - Member(member_id, name, email, phone, join_date, membership_type, status)
> - Publisher(publisher_id, name, address, contact_email)
> - Librarian(librarian_id, name, email, hire_date, role)
> - BorrowRecord(borrow_id, member_id, book_id, librarian_id, borrow_date, return_date, due_date, status)
> - BookAuthor(book_id, author_id)

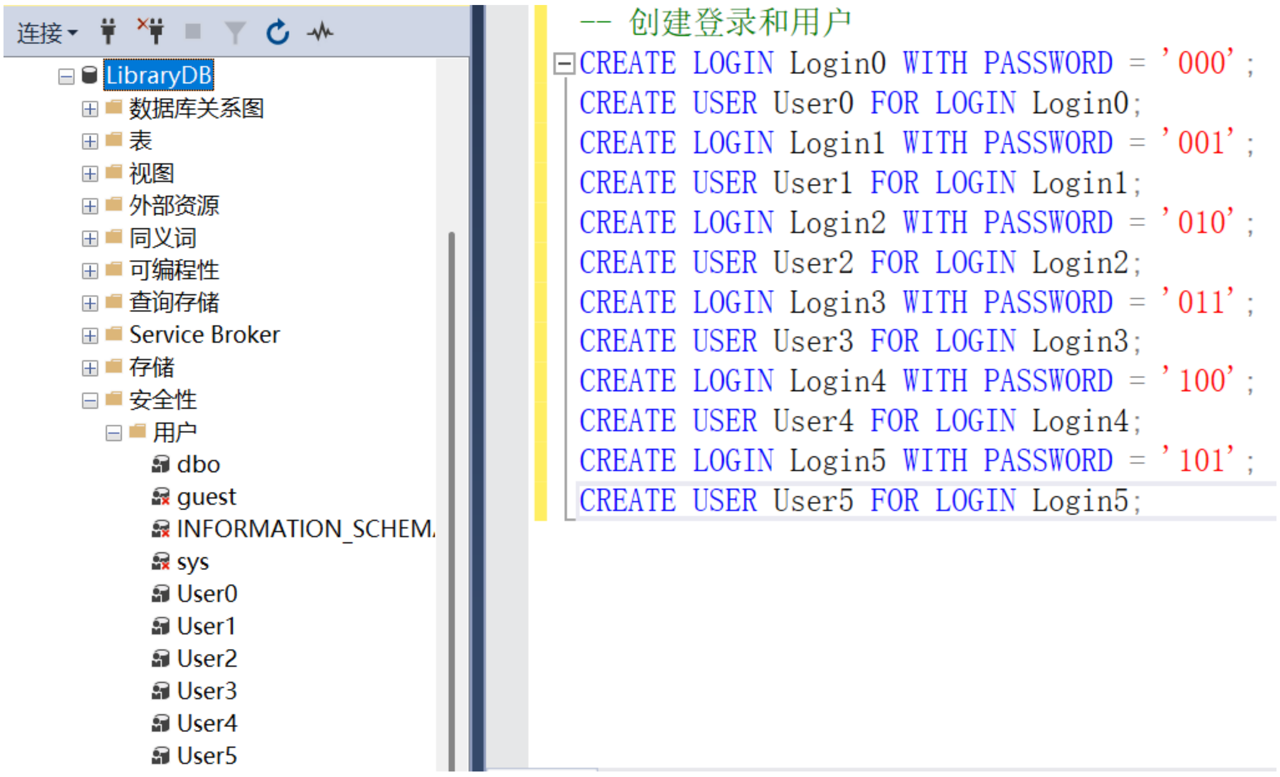
实验过程

一、授权与回收

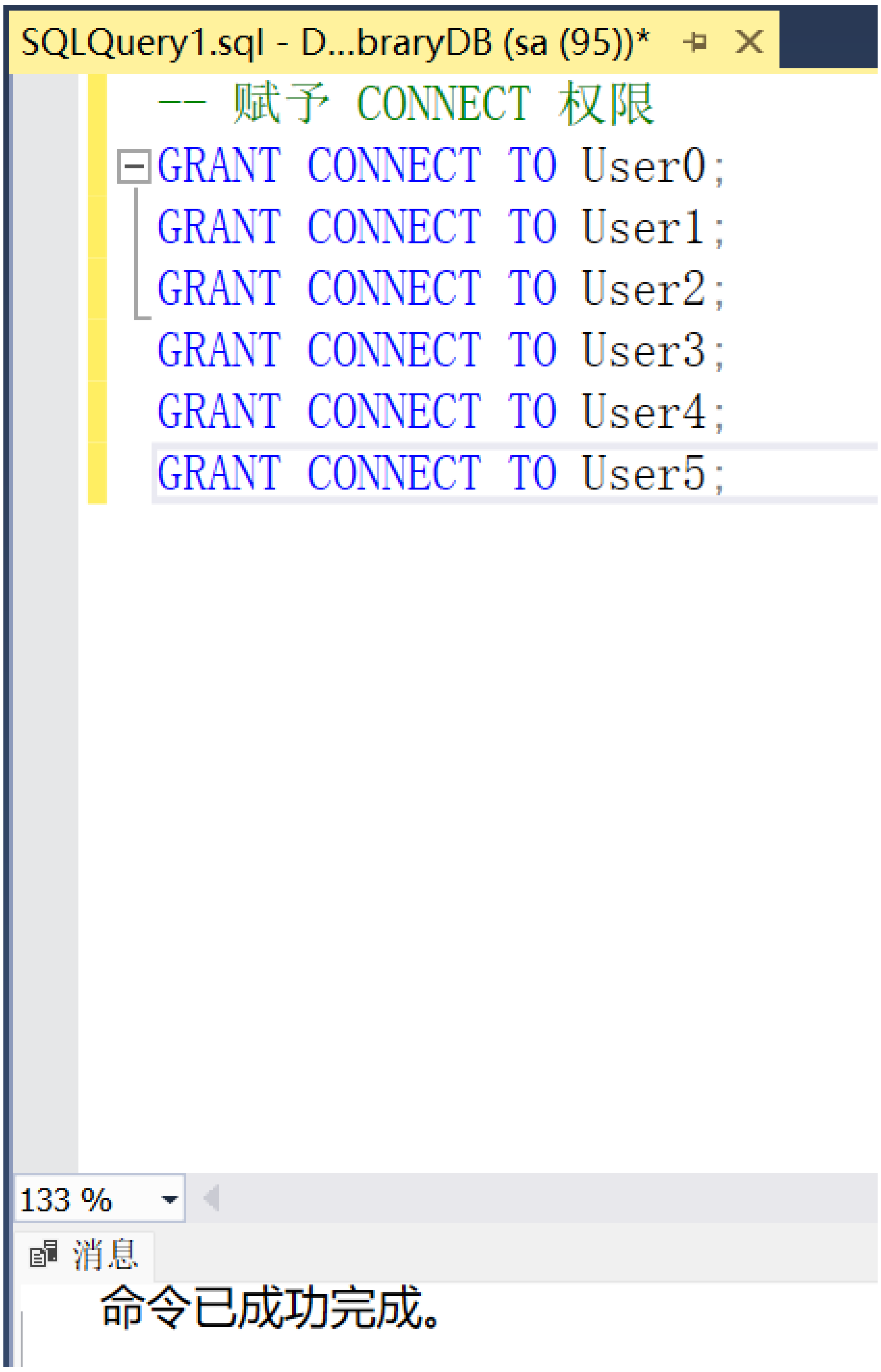
1. 在数据库中由DBA创建若干用戶，权限全部选择为 CONNECT，即SQL Server中的db_accessadmin角色；
   - 语句：

```sql
-- 创建登录和用户
CREATE LOGIN Login0 WITH PASSWORD = '000';
CREATE USER User0 FOR LOGIN Login0;
CREATE LOGIN Login1 WITH PASSWORD = '001';
CREATE USER User1 FOR LOGIN Login1;
CREATE LOGIN Login2 WITH PASSWORD = '010';
CREATE USER User2 FOR LOGIN Login2;
CREATE LOGIN Login3 WITH PASSWORD = '011';
CREATE USER User3 FOR LOGIN Login3;
CREATE LOGIN Login4 WITH PASSWORD = '100';
CREATE USER User4 FOR LOGIN Login4;
CREATE LOGIN Login5 WITH PASSWORD = '101';
CREATE USER User5 FOR LOGIN Login5;

-- 赋予 CONNECT 权限
GRANT CONNECT TO User0;
GRANT CONNECT TO User1;
GRANT CONNECT TO User2;
GRANT CONNECT TO User3;
GRANT CONNECT TO User4;
GRANT CONNECT TO User5;
```

- 结果：



```sql
-- 创建登录和用户
CREATE LOGIN Login0 WITH PASSWORD = '000';
CREATE USER User0 FOR LOGIN Login0;
CREATE LOGIN Login1 WITH PASSWORD = '001';
CREATE USER User1 FOR LOGIN Login1;
CREATE LOGIN Login2 WITH PASSWORD = '010';
CREATE USER User2 FOR LOGIN Login2;
CREATE LOGIN Login3 WITH PASSWORD = '011';
CREATE USER User3 FOR LOGIN Login3;
CREATE LOGIN Login4 WITH PASSWORD = '100';
CREATE USER User4 FOR LOGIN Login4;
CREATE LOGIN Login5 WITH PASSWORD = '101';
CREATE USER User5 FOR LOGIN Login5;
```



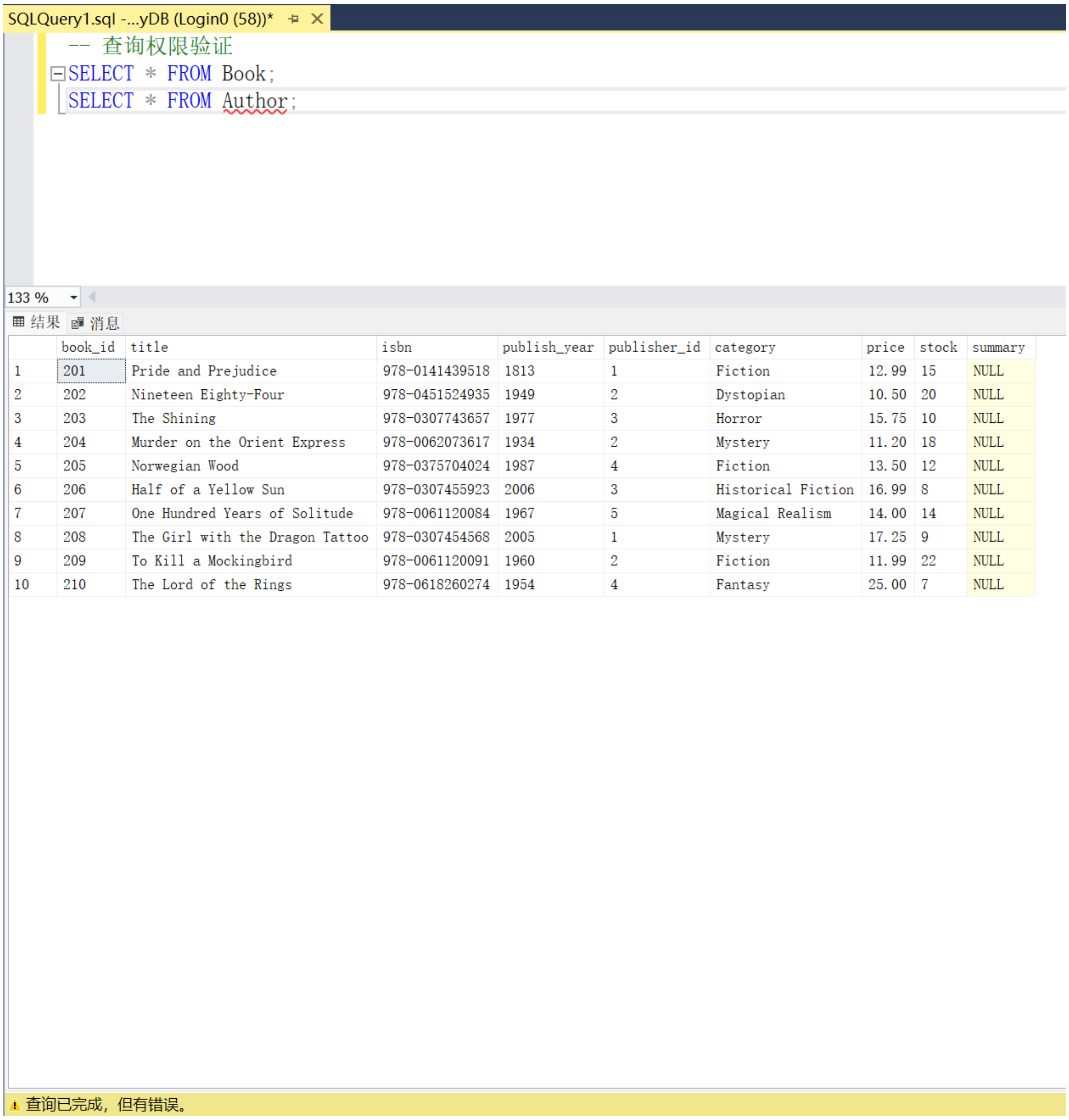
```
-- 赋予 CONNECT 权限
GRANT CONNECT TO User0;
GRANT CONNECT TO User1;
GRANT CONNECT TO User2;
GRANT CONNECT TO User3;
GRANT CONNECT TO User4;
GRANT CONNECT TO User5;
```

```
命令已成功完成。
```

2. 把查询 `Book` 和修改 `booki_id` 的权限授给用戶User0;
   - 语句：

```sql
-- 赋予User0可查Book和改变book_id的权限
GRANT UPDATE(price), SELECT
ON Book
TO User0;

-- 查询权限验证
SELECT * FROM Book;
SELECT * FROM Author;

-- 修改权限验证
UPDATE Book
SET price=20
WHERE title='Pride and Prejudice';

UPDATE Book
SET publish_year=2000
WHERE title='Pride and Prejudice';
```

- 结果（登录User0执行）：



SQLQuery1.sql -...yDB (Login0 (58))*

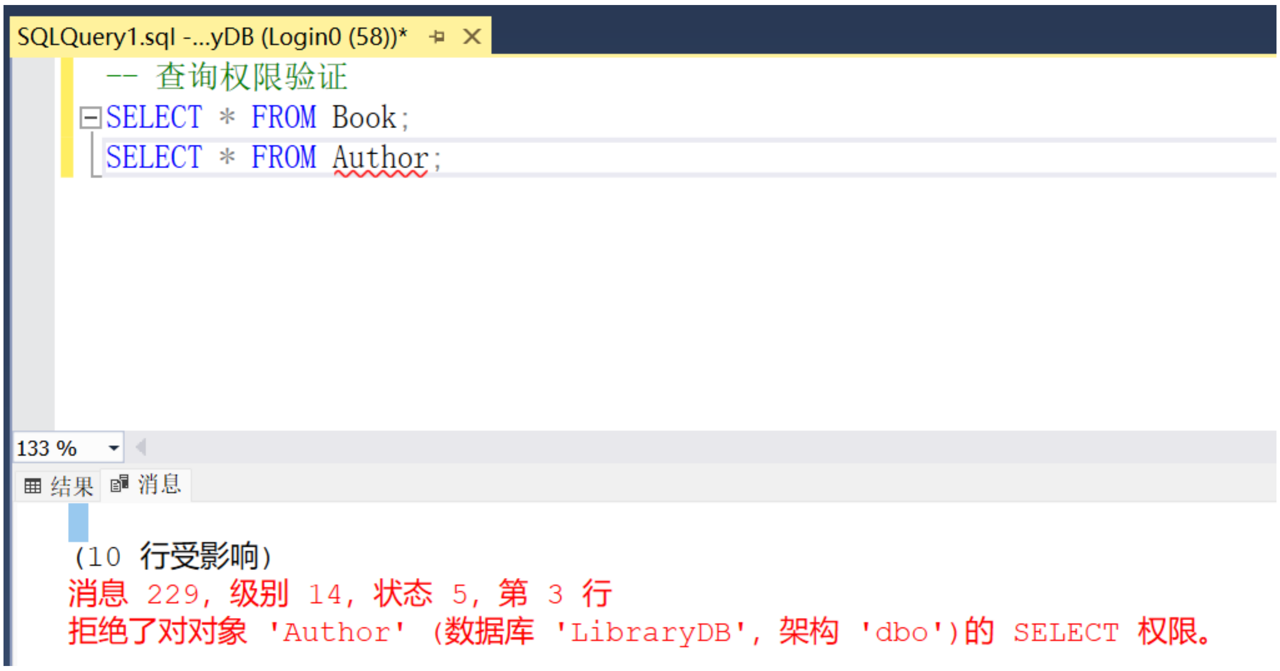
```sql
-- 查询权限验证
SELECT * FROM Book;
SELECT * FROM Author;
```

133 %

结果 消息

	book_id	title	isbn	publish_year	publisher_id	category	price	stock	summary
1	201	Pride and Prejudice	978-0141439518	1813	1	Fiction	12.99	15	NULL
2	202	Nineteen Eighty-Four	978-0451524935	1949	2	Dystopian	10.50	20	NULL
3	203	The Shining	978-0307743657	1977	3	Horror	15.75	10	NULL
4	204	Murder on the Orient Express	978-0062073617	1934	2	Mystery	11.20	18	NULL
5	205	Norwegian Wood	978-0375704024	1987	4	Fiction	13.50	12	NULL
6	206	Half of a Yellow Sun	978-0307455923	2006	3	Historical Fiction	16.99	8	NULL
7	207	One Hundred Years of Solitude	978-0061120084	1967	5	Magical Realism	14.00	14	NULL
8	208	The Girl with the Dragon Tattoo	978-0307454568	2005	1	Mystery	17.25	9	NULL
9	209	To Kill a Mockingbird	978-0061120091	1960	2	Fiction	11.99	22	NULL
10	210	The Lord of the Rings	978-0618260274	1954	4	Fantasy	25.00	7	NULL

查询已完成，但有错误。



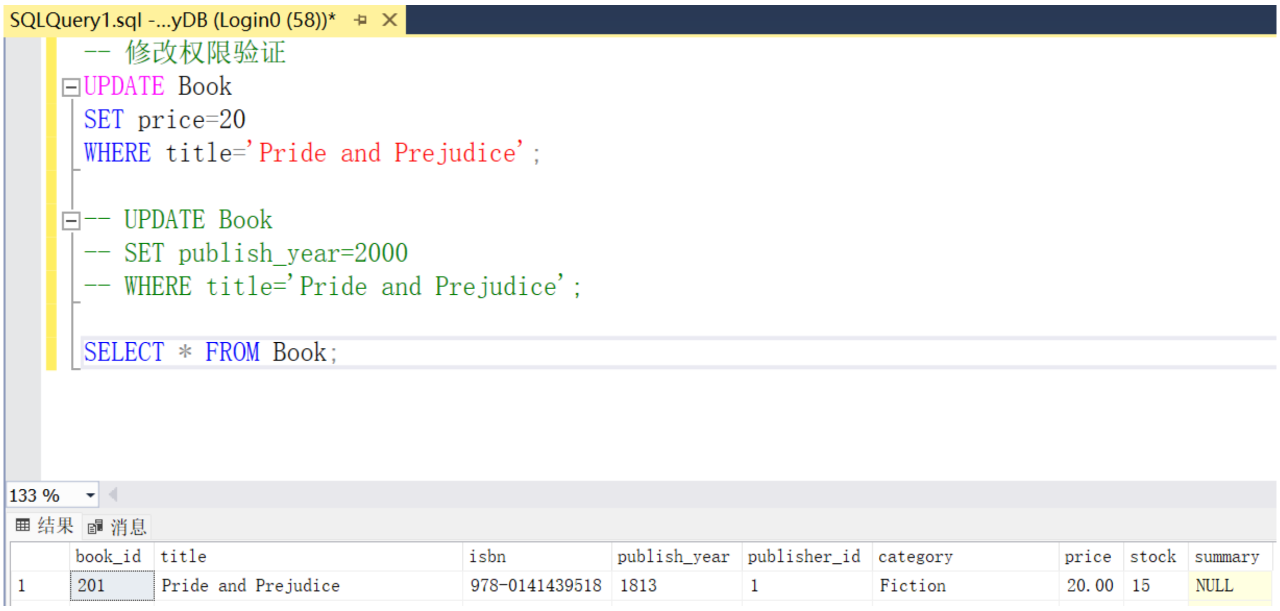
SQLQuery1.sql -...yDB (Login0 (58))*

```sql
-- 查询权限验证
SELECT * FROM Book;
SELECT * FROM Author;
```

133 %

结果 消息

```
(10 行受影响)
消息 229，级别 14，状态 5，第 3 行
拒绝了对对象 'Author' (数据库 'LibraryDB'，架构 'dbo')的 SELECT 权限。
```



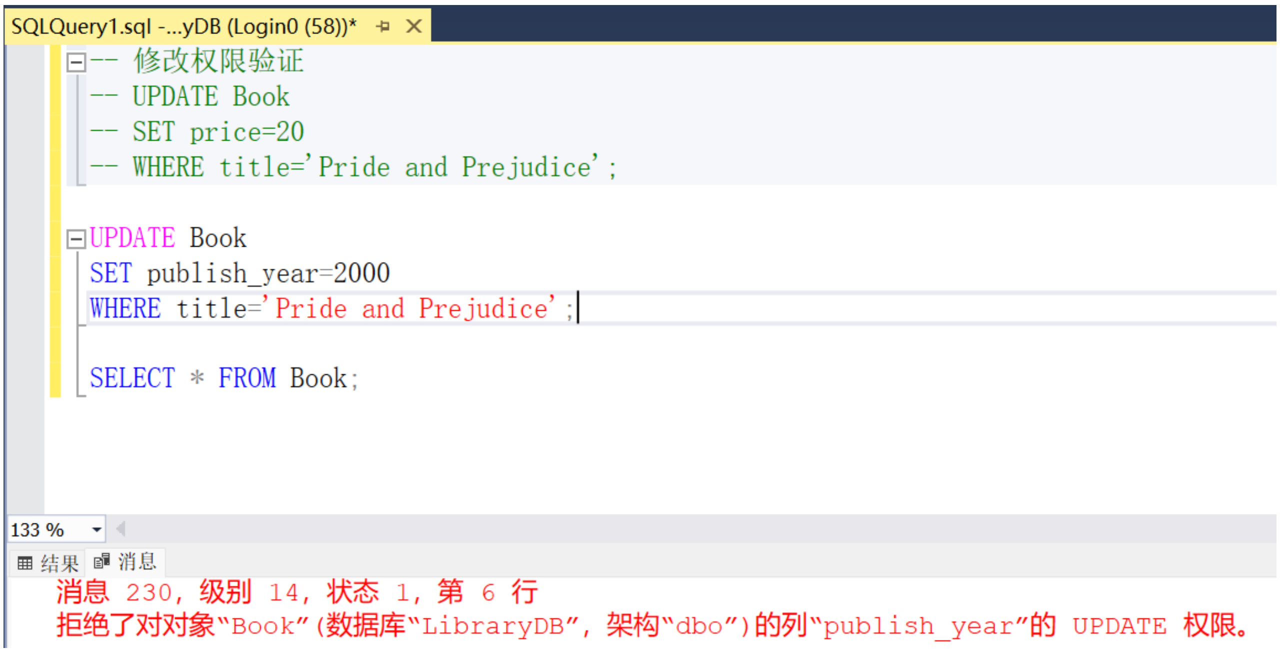
SQLQuery1.sql -...yDB (Login0 (58))*

```
-- 修改权限验证
UPDATE Book
SET price=20
WHERE title='Pride and Prejudice';

-- UPDATE Book
-- SET publish_year=2000
-- WHERE title='Pride and Prejudice';

SELECT * FROM Book;
```

133 %

结果 消息

	book_id	title	isbn	publish_year	publisher_id	category	price	stock	summary
1	201	Pride and Prejudice	978-0141439518	1813	1	Fiction	20.00	15	NULL



SQLQuery1.sql -...yDB (Login0 (58))*

```
-- 修改权限验证
-- UPDATE Book
-- SET price=20
-- WHERE title='Pride and Prejudice';

UPDATE Book
SET publish_year=2000
WHERE title='Pride and Prejudice';

SELECT * FROM Book;
```

133 %

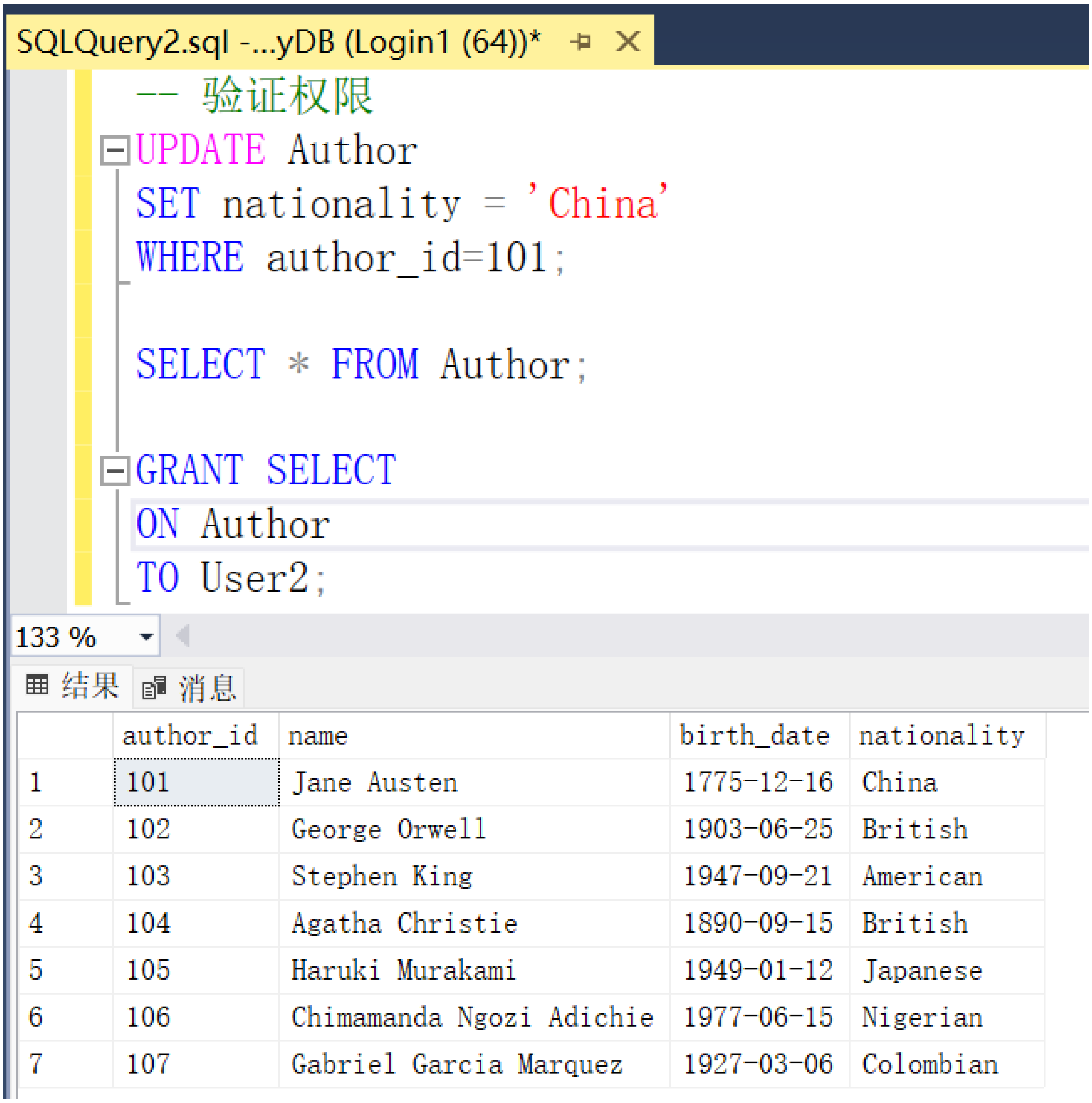
结果 消息

```
消息 230，级别 14，状态 1，第 6 行
拒绝了对对象"Book"(数据库"LibraryDB"，架构"dbo")的列"publish_year"的 UPDATE 权限。
```

3. 把对 `Author` 表的查询和修改`nationality`的权限授予用戶User1，并允许其再将此权限授予其他用戶；
   - 语句：

```sql
-- 赋予User0可查Book和改变book_id的权限
GRANT UPDATE(nationality), SELECT
ON Author
TO User1 WITH GRANT OPTION;

-- 验证权限(User1)
UPDATE Author
SET nationality = 'China'
WHERE author_id=101;

SELECT * FROM Author;

GRANT SELECT
ON Author
TO User2;

-- 验证权限(User2)
SELECT * FROM Author;
```

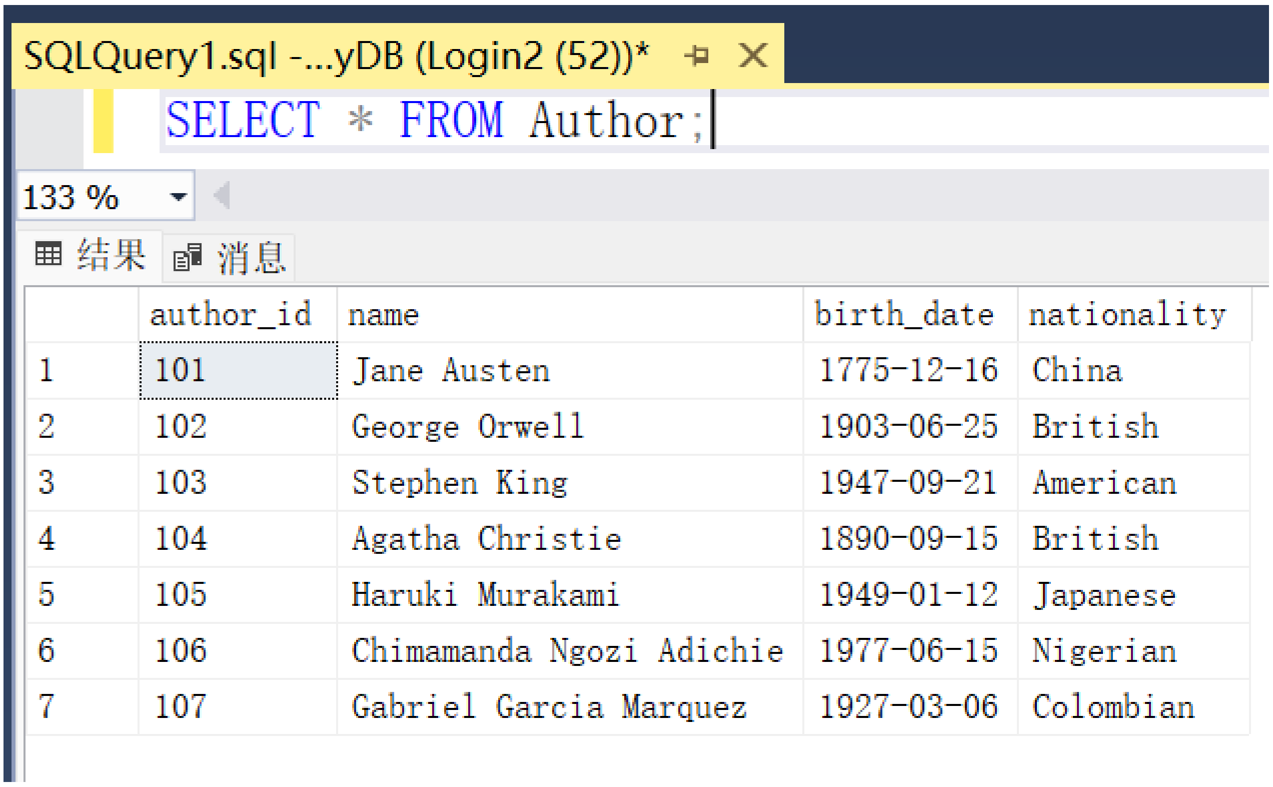
- 结果（登录User1执行）：



```sql
-- 验证权限
UPDATE Author
SET nationality = 'China'
WHERE author_id=101;

SELECT * FROM Author;

GRANT SELECT
ON Author
TO User2;
```

	author_id	name	birth_date	nationality
1	101	Jane Austen	1775-12-16	China
2	102	George Orwell	1903-06-25	British
3	103	Stephen King	1947-09-21	American
4	104	Agatha Christie	1890-09-15	British
5	105	Haruki Murakami	1949-01-12	Japanese
6	106	Chimamanda Ngozi Adichie	1977-06-15	Nigerian
7	107	Gabriel Garcia Marquez	1927-03-06	Colombian

下图为User2执行结果：



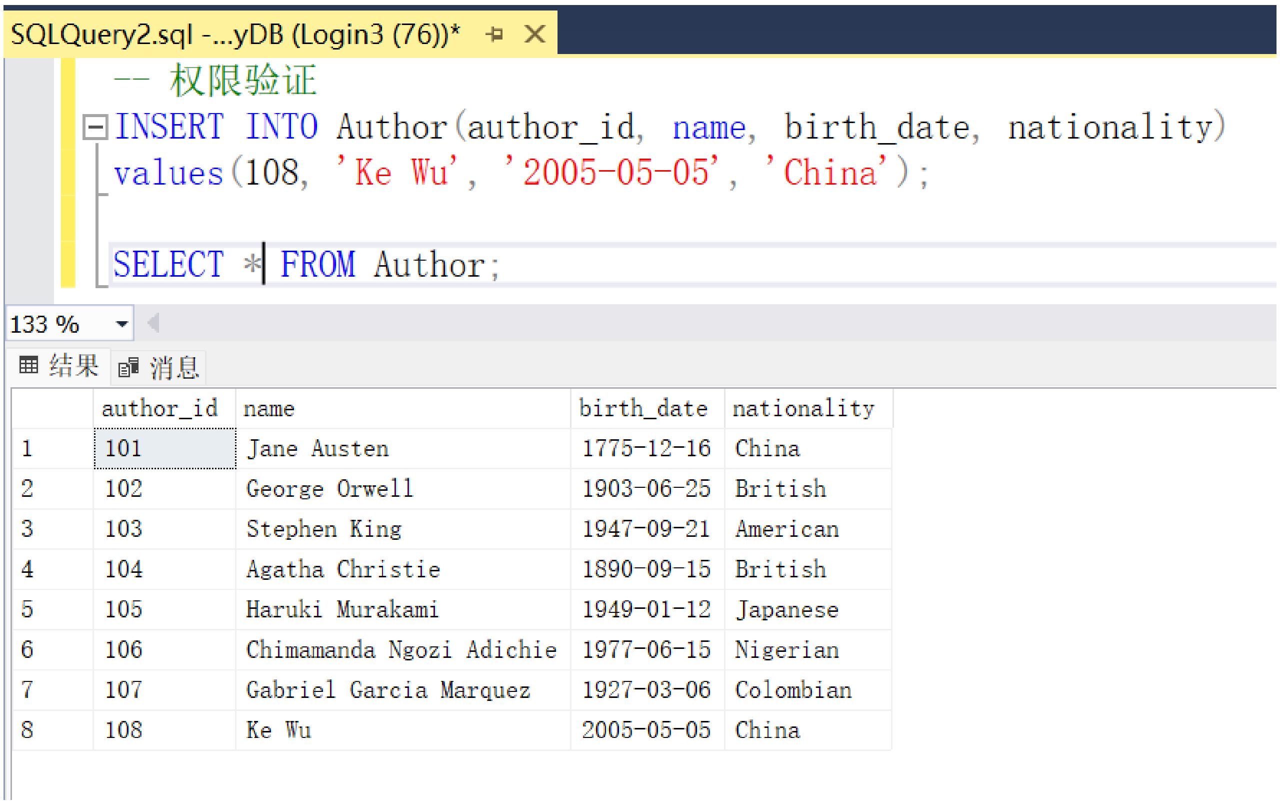
```
SELECT * FROM Author;
```

	author_id	name	birth_date	nationality
1	101	Jane Austen	1775-12-16	China
2	102	George Orwell	1903-06-25	British
3	103	Stephen King	1947-09-21	American
4	104	Agatha Christie	1890-09-15	British
5	105	Haruki Murakami	1949-01-12	Japanese
6	106	Chimamanda Ngozi Adichie	1977-06-15	Nigerian
7	107	Gabriel Garcia Marquez	1927-03-06	Colombian

4. 把对 `Author` 表的全部权限授予给用户User3;
   - 语句：

```sql
-- 赋予User3所有权限
GRANT ALL PRIVILEGES
ON Author
TO User3;

-- 权限验证
INSERT INTO Author(author_id, name, birth_date, nationality)
values(108, 'Ke Wu', '2005-05-05', 'China');

SELECT * FROM Author;
```

- 结果（登录User3执行）：



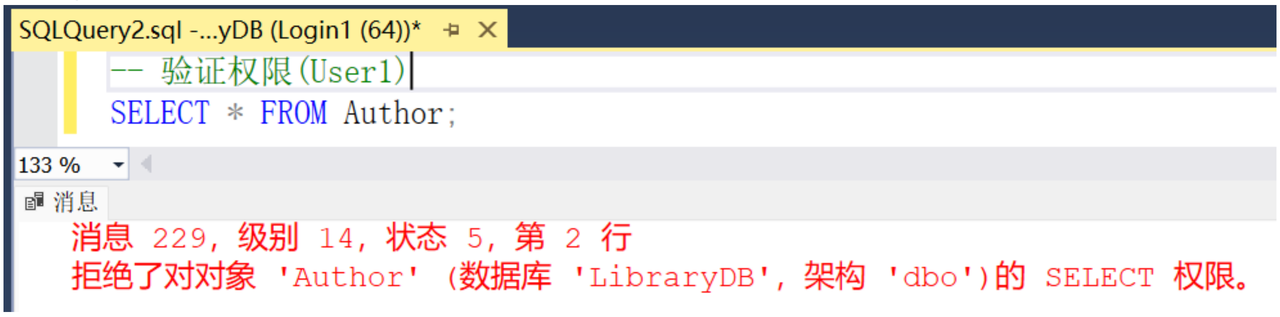
```
-- 权限验证
INSERT INTO Author(author_id, name, birth_date, nationality)
values(108, 'Ke Wu', '2005-05-05', 'China');

SELECT * FROM Author;
```

	author_id	name	birth_date	nationality
1	101	Jane Austen	1775-12-16	China
2	102	George Orwell	1903-06-25	British
3	103	Stephen King	1947-09-21	American
4	104	Agatha Christie	1890-09-15	British
5	105	Haruki Murakami	1949-01-12	Japanese
6	106	Chimamanda Ngozi Adichie	1977-06-15	Nigerian
7	107	Gabriel Garcia Marquez	1927-03-06	Colombian
8	108	Ke Wu	2005-05-05	China

5. 收回User1查询 `Author` 表的权限以及他赋予的权限；
   - 语句：

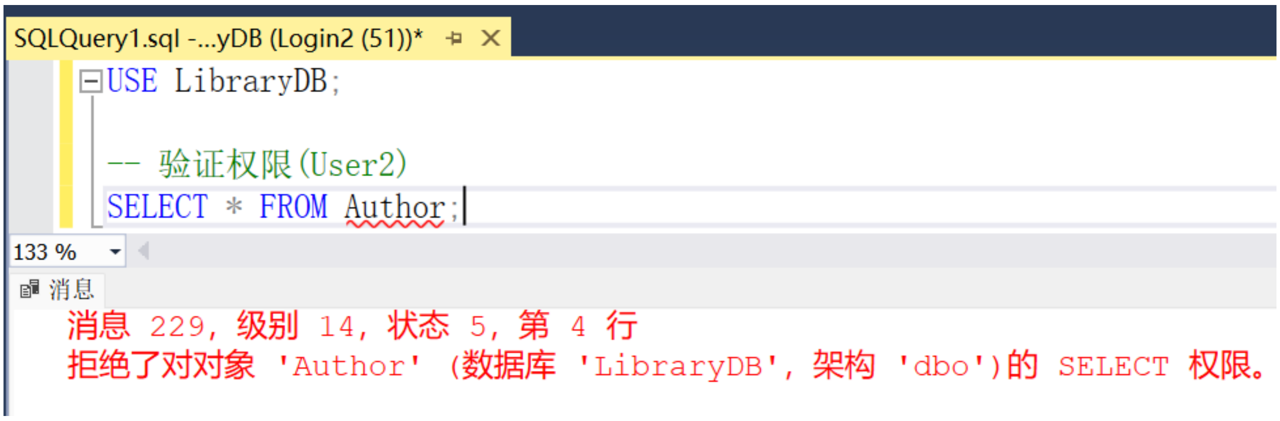
```
-- 收回权限
REVOKE SELECT
ON Author
FROM User1 CASCADE;

-- 验证权限(User1)
SELECT * FROM Author;

-- 验证权限(User2)
SELECT * FROM Author;
```

- 结果（登录User1）：



```
-- 验证权限(User1)
SELECT * FROM Author;
```

```
消息 229，级别 14，状态 5，第 2 行
拒绝了对对象 'Author' (数据库 'LibraryDB'，架构 'dbo')的 SELECT 权限。
```



SQLQuery1.sql -...yDB (Login2 (51))*

```
USE LibraryDB;

-- 验证权限(User2)
SELECT * FROM Author;
```

133 %

消息

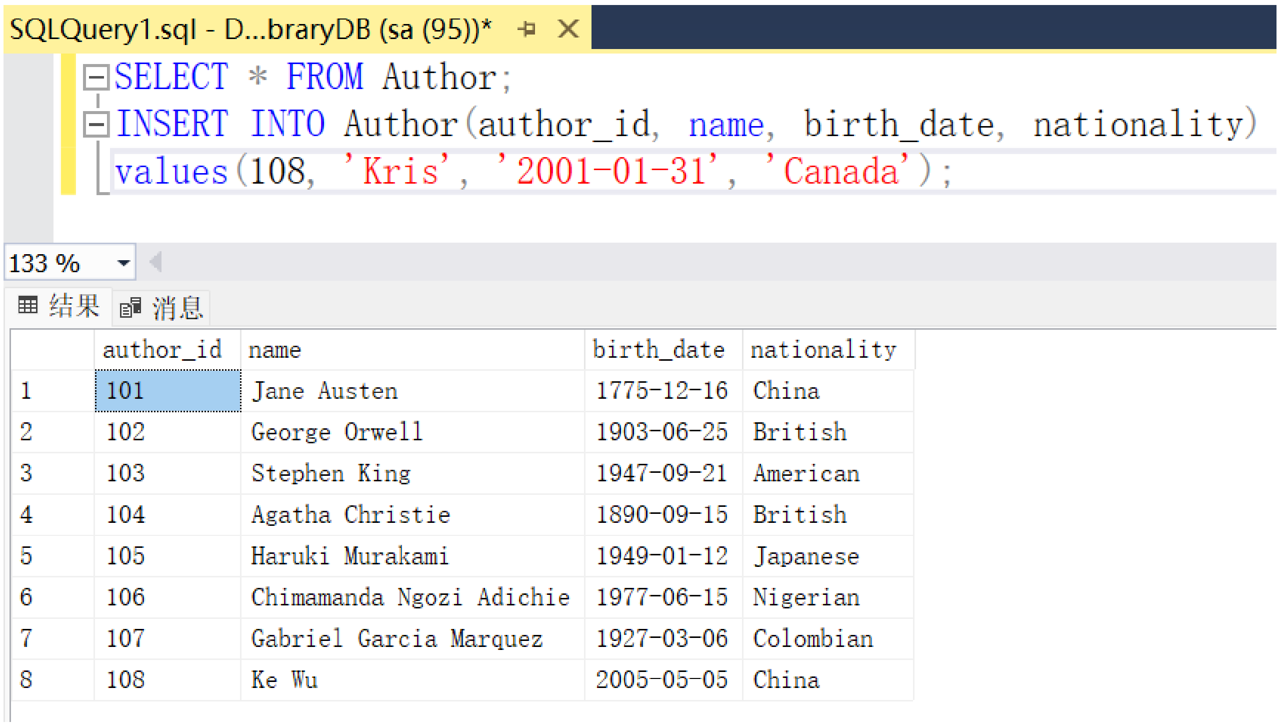
```
消息 229，级别 14，状态 5，第 4 行
拒绝了对对象 'Author' (数据库 'LibraryDB'，架构 'dbo')的 SELECT 权限。
```

二、完整性控制

1. 实体完整性：已定义 `author_id` 为 `Author` 主键；
   - 语句：

```sql
SELECT * FROM Author;

INSERT INTO Author(author_id, name, birth_date, nationality)
VALUES(108, 'Kris', '2001-01-31', 'Canada');
```

   - 结果： 下图为目前 `Author` 表内容：



SQLQuery1.sql - D...braryDB (sa (95))*

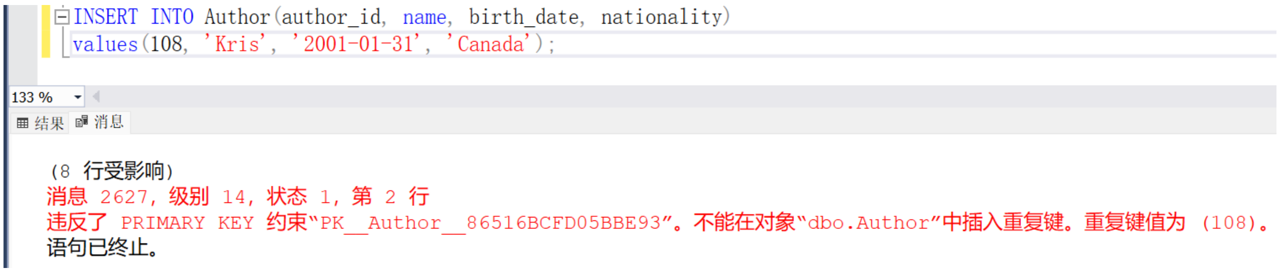
```
SELECT * FROM Author;
INSERT INTO Author(author_id, name, birth_date, nationality)
values(108, 'Kris', '2001-01-31', 'Canada');
```

133 %

结果 消息

	author_id	name	birth_date	nationality
1	101	Jane Austen	1775-12-16	China
2	102	George Orwell	1903-06-25	British
3	103	Stephen King	1947-09-21	American
4	104	Agatha Christie	1890-09-15	British
5	105	Haruki Murakami	1949-01-12	Japanese
6	106	Chimamanda Ngozi Adichie	1977-06-15	Nigerian
7	107	Gabriel Garcia Marquez	1927-03-06	Colombian
8	108	Ke Wu	2005-05-05	China

下图为尝试添加重复主键结果：



```
INSERT INTO Author(author_id, name, birth_date, nationality)
values(108, 'Kris', '2001-01-31', 'Canada');
```

133 %

结果 消息

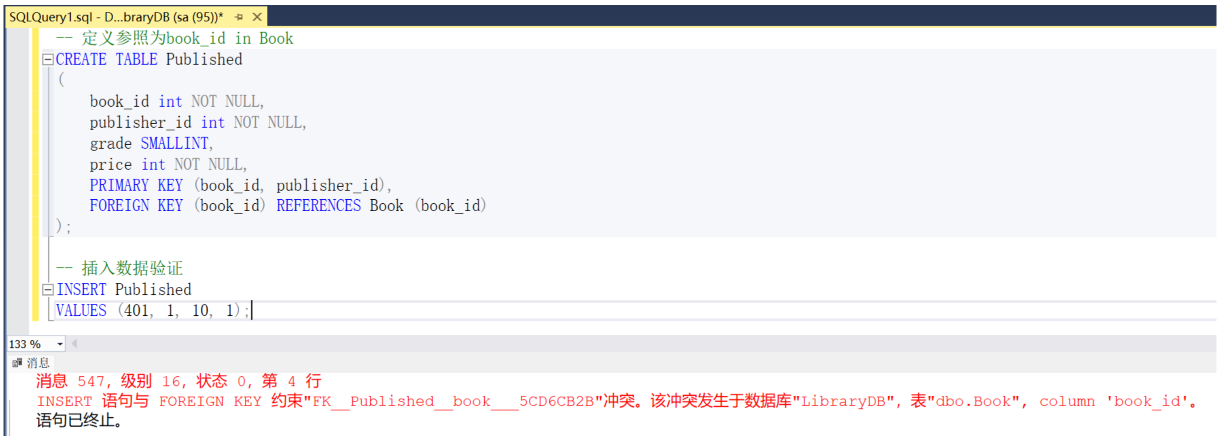
```
(8 行受影响)
消息 2627，级别 14，状态 1，第 2 行
违反了 PRIMARY KEY 约束"PK__Author__86516BCFD05BBE93"。不能在对象"dbo.Author"中插入重复键。重复键值为 (108)。
语句已终止。
```

2. 参照完整性：由于前面的定义都已完成，此处重新建立一个联系 `Published(book_id, publisher_id);`

1. 定义 `Published` 中的参照完整性，`Published` 的 `book_id` 参照 `Book` 表的主码 `book_id`：
   - 语句：

```sql
-- 定义参照为book_id in Book
CREATE TABLE Published
(
    book_id int NOT NULL,
    publisher_id int NOT NULL,
    grade SMALLINT,
    price int NOT NULL,
    PRIMARY KEY (book_id, publisher_id),
    FOREIGN KEY (book_id) REFERENCES Book (book_id)
);

-- 插入数据验证
INSERT Published
VALUES (401, 1, 10, 1);
```

   - 结果：



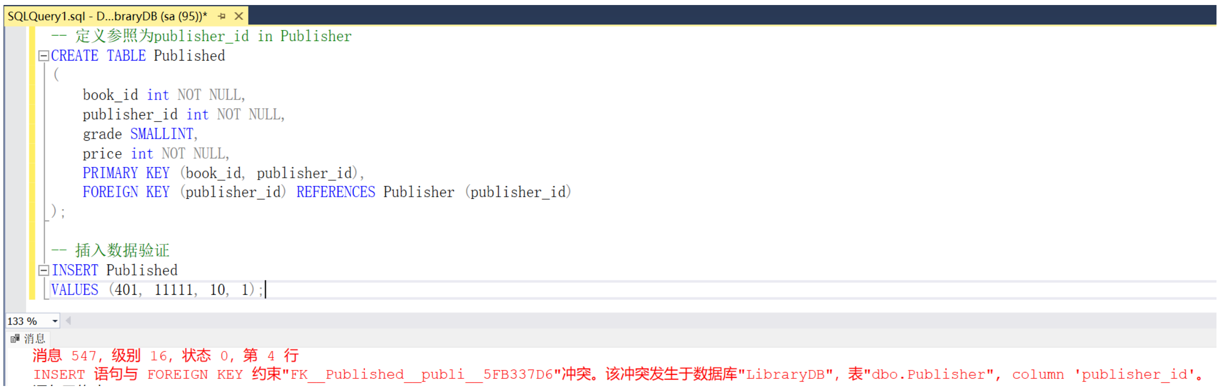
```
-- 定义参照为book_id in Book
CREATE TABLE Published
(
    book_id int NOT NULL,
    publisher_id int NOT NULL,
    grade SMALLINT,
    price int NOT NULL,
    PRIMARY KEY (book_id, publisher_id),
    FOREIGN KEY (book_id) REFERENCES Book (book_id)
);

-- 插入数据验证
INSERT Published
VALUES (401, 1, 10, 1);
```

```
消息 547，级别 16，状态 0，第 4 行
INSERT 语句与 FOREIGN KEY 约束"FK__Published__book___5CD6CB2B"冲突。该冲突发生于数据库"LibraryDB"，表"dbo.Book", column 'book_id'。
语句已终止。
```

2. 定义 `Published` 中的参照完整性，`Published` 的 `publisher_id` 参照 `Publisher` 表的主码 `publisher_id`：
   - 语句：

```sql
-- 定义参照为publisher_id in Publisher
CREATE TABLE Published
(
    book_id int NOT NULL,
    publisher_id int NOT NULL,
    grade SMALLINT,
    price int NOT NULL,
    PRIMARY KEY (book_id, publisher_id),
    FOREIGN KEY (publisher_id) REFERENCES Publisher (publisher_id)
);

-- 插入数据验证
INSERT Published
VALUES (401, 11111, 10, 1);
```

- 结果：



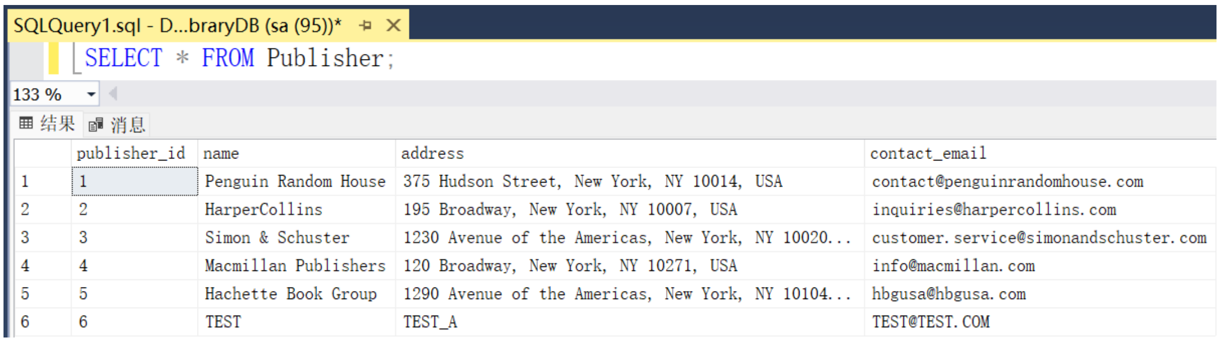
```
-- 定义参照为publisher_id in Publisher
CREATE TABLE Published
(
    book_id int NOT NULL,
    publisher_id int NOT NULL,
    grade SMALLINT,
    price int NOT NULL,
    PRIMARY KEY (book_id, publisher_id),
    FOREIGN KEY (publisher_id) REFERENCES Publisher (publisher_id)
);

-- 插入数据验证
INSERT Published
VALUES (401, 11111, 10, 1);
```

```
消息 547，级别 16，状态 0，第 4 行
INSERT 语句与 FOREIGN KEY 约束"FK__Published__publi__5FB337D6"冲突。该冲突发生于数据库"LibraryDB"，表"dbo.Publisher", column 'publisher_id'。
```

3. 设置了联级删除 Published 表中相应的元组，即在删除 Publisher 表中某个元组时会分带删除 Published 表中相应的元组；
   - 语句：

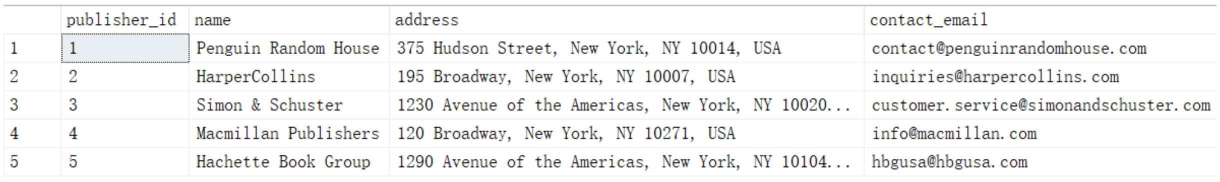
```sql
-- 定义参照为publisher_id in Publisher
CREATE TABLE Published
(
    book_id int NOT NULL,
    publisher_id int NOT NULL,
    grade SMALLINT,
    price int NOT NULL,
    PRIMARY KEY (book_id, publisher_id),
    FOREIGN KEY (publisher_id) REFERENCES Publisher (publisher_id) ON
DELETE CASCADE
);

-- 验证
INSERT Published
VALUES (401, 6, 10, 1);

SELECT * FROM Published;

DELETE Publisher WHERE publisher_id=6;

SELECT * FROM Published;
```

   - 结果：Publisher 表数据对比：



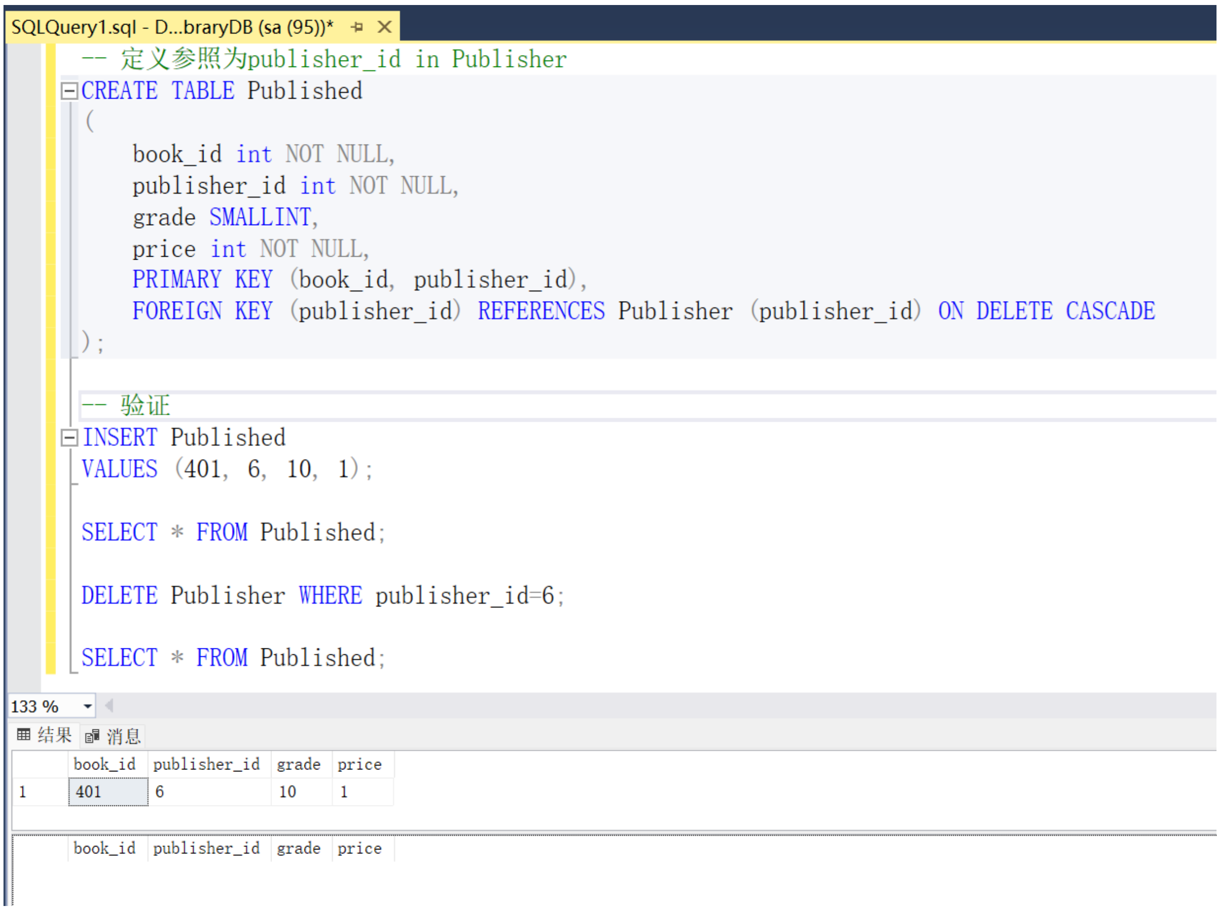
```
SELECT * FROM Publisher;
```

	publisher_id	name	address	contact_email
1	1	Penguin Random House	375 Hudson Street, New York, NY 10014, USA	contact@penguinrandomhouse.com
2	2	HarperCollins	195 Broadway, New York, NY 10007, USA	inquiries@harpercollins.com
3	3	Simon & Schuster	1230 Avenue of the Americas, New York, NY 10020...	customer.service@simonandschuster.com
4	4	Macmillan Publishers	120 Broadway, New York, NY 10271, USA	info@macmillan.com
5	5	Hachette Book Group	1290 Avenue of the Americas, New York, NY 10104...	hbgusa@hbgusa.com
6	6	TEST	TEST_A	TEST@TEST.COM



	publisher_id	name	address	contact_email
1	1	Penguin Random House	375 Hudson Street, New York, NY 10014, USA	contact@penguinrandomhouse.com
2	2	HarperCollins	195 Broadway, New York, NY 10007, USA	inquiries@harpercollins.com
3	3	Simon & Schuster	1230 Avenue of the Americas, New York, NY 10020...	customer.service@simonandschuster.com
4	4	Macmillan Publishers	120 Broadway, New York, NY 10271, USA	info@macmillan.com
5	5	Hachette Book Group	1290 Avenue of the Americas, New York, NY 10104...	hbgusa@hbgusa.com

`Published` 表数据对比：



SQLQuery1.sql - D...braryDB (sa (95))*

```sql
-- 定义参照为publisher_id in Publisher
CREATE TABLE Published
(
    book_id int NOT NULL,
    publisher_id int NOT NULL,
    grade SMALLINT,
    price int NOT NULL,
    PRIMARY KEY (book_id, publisher_id),
    FOREIGN KEY (publisher_id) REFERENCES Publisher (publisher_id) ON DELETE CASCADE
);

-- 验证
INSERT Published
VALUES (401, 6, 10, 1);

SELECT * FROM Published;

DELETE Publisher WHERE publisher_id=6;

SELECT * FROM Published;
```

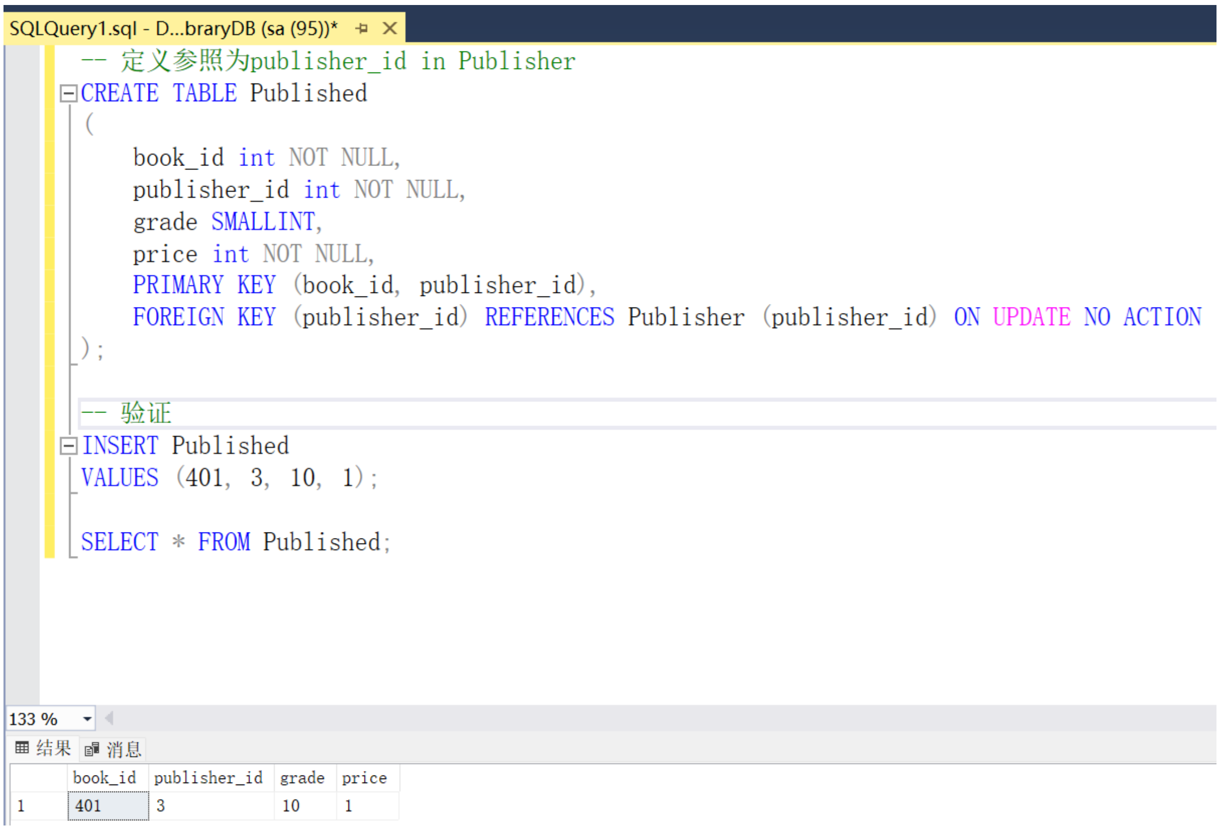
	book_id	publisher_id	grade	price
1	401	6	10	1

book_id	publisher_id	grade	price

4. 设置了当修改被参照表 `Publisher` 中的元组时，DBMS会拒绝修改。
   - 语句：

```sql
-- 定义参照为publisher_id in Publisher
CREATE TABLE Published
(
    book_id int NOT NULL,
    publisher_id int NOT NULL,
    grade SMALLINT,
    price int NOT NULL,
    PRIMARY KEY (book_id, publisher_id),
    FOREIGN KEY (publisher_id) REFERENCES Publisher (publisher_id) ON
DELETE CASCADE
);

-- 验证
INSERT Published
VALUES (401, 6, 10, 1);

SELECT * FROM Published;

UPDATE Publisher
SET name='Joker'
WHERE publisher_id=3;
```

- 结果： 创建结果：

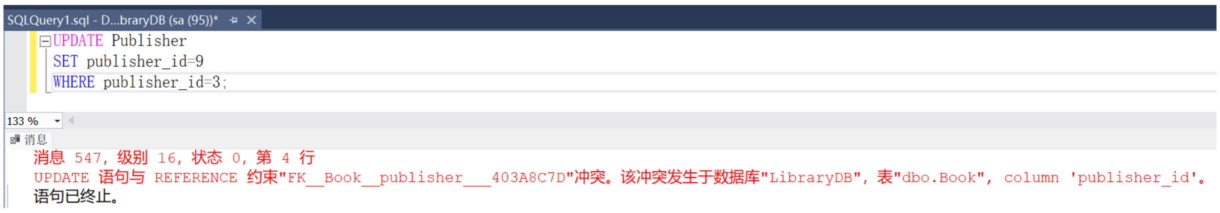


```sql
-- 定义参照为publisher_id in Publisher
CREATE TABLE Published
(
    book_id int NOT NULL,
    publisher_id int NOT NULL,
    grade SMALLINT,
    price int NOT NULL,
    PRIMARY KEY (book_id, publisher_id),
    FOREIGN KEY (publisher_id) REFERENCES Publisher (publisher_id) ON UPDATE NO ACTION
);

-- 验证
INSERT Published
VALUES (401, 3, 10, 1);

SELECT * FROM Published;
```

```
   book_id  publisher_id  grade  price
1  401      3             10     1
```

修改报错，由于前面的 Book 表设计时也使用了该方式定义，因此最终结果显示先显示了 Book 引发

的报错，之所以新建表视为了说明语句书写方法：



SQLQuery1.sql - D...braryDB (sa (95))*

```
UPDATE Publisher
SET publisher_id=9
WHERE publisher_id=3;
```

133 %

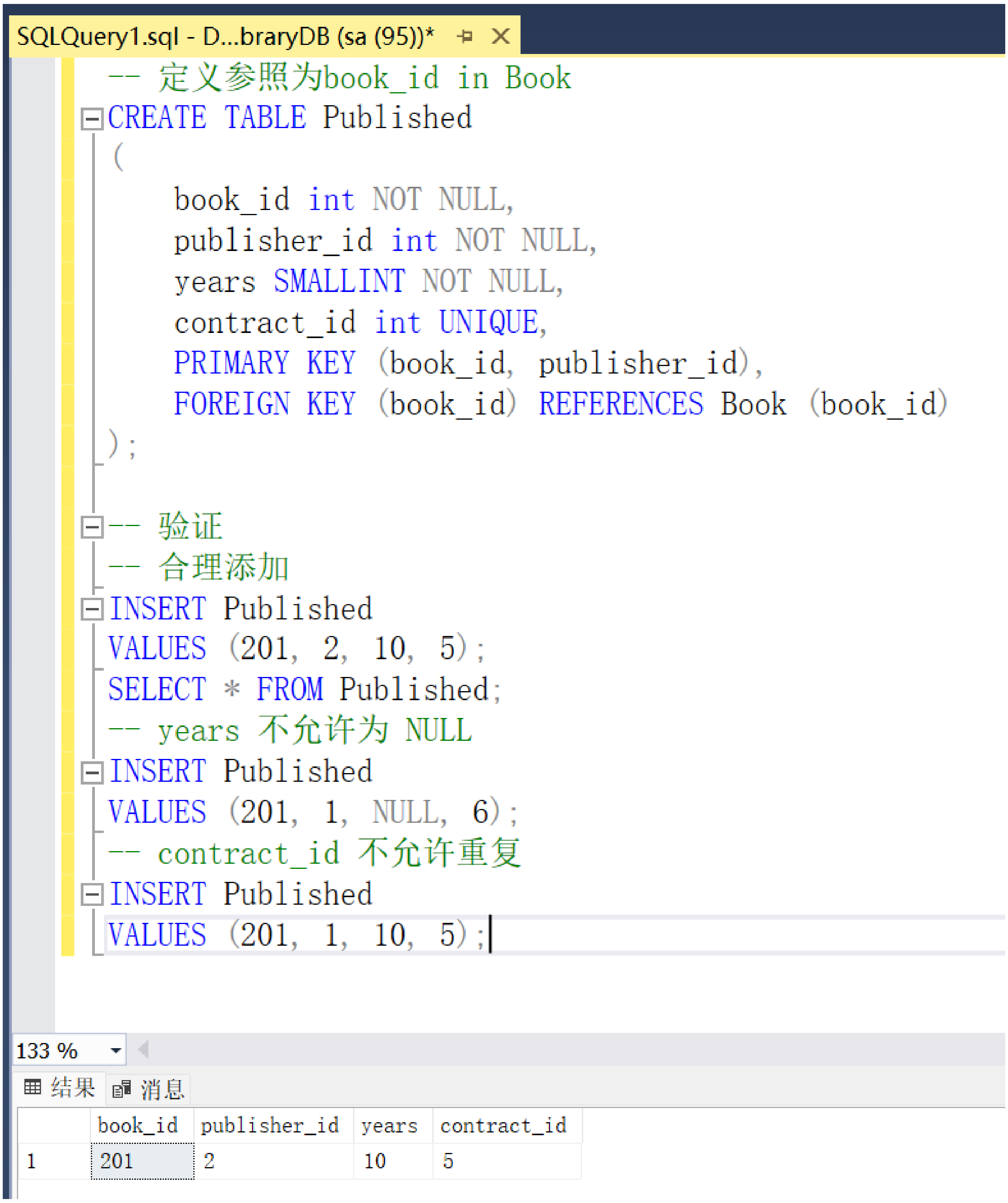
消息

```
消息 547，级别 16，状态 0，第 4 行
UPDATE 语句与 REFERENCE 约束"FK__Book__publisher___403A8C7D"冲突。该冲突发生于数据库"LibraryDB"，表"dbo.Book", column 'publisher_id'。
语句已终止。
```

3. 用户定义完整性：以 `Published(book_id, publisher_id, years, contract_id)` 为操作对象；
   - 语句：

```sql
-- 定义参照为book_id in Book
CREATE TABLE Published
(
    book_id int NOT NULL,
    publisher_id int NOT NULL,
    years SMALLINT NOT NULL,
    contract_id int UNIQUE,
    PRIMARY KEY (book_id, publisher_id),
    FOREIGN KEY (book_id) REFERENCES Book (book_id)
);

-- 验证
-- 合理添加
INSERT Published
VALUES (201, 2, 10, 5);

SELECT * FROM Published;

-- years 不允许为 NULL
INSERT Published
VALUES (201, 1, NULL, 6);

-- contract_id 不允许重复
INSERT Published
VALUES (201, 1, 10, 5);
```

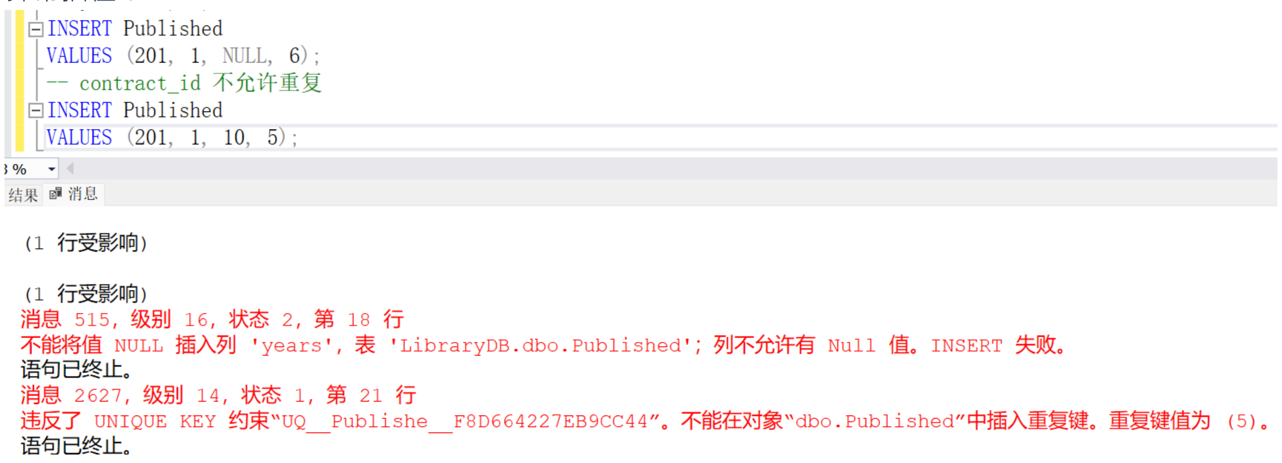
- 结果：合法操作：



```sql
-- 定义参照为book_id in Book
CREATE TABLE Published
(
    book_id int NOT NULL,
    publisher_id int NOT NULL,
    years SMALLINT NOT NULL,
    contract_id int UNIQUE,
    PRIMARY KEY (book_id, publisher_id),
    FOREIGN KEY (book_id) REFERENCES Book (book_id)
);

-- 验证
-- 合理添加
INSERT Published
VALUES (201, 2, 10, 5);
SELECT * FROM Published;
-- years 不允许为 NULL
INSERT Published
VALUES (201, 1, NULL, 6);
-- contract_id 不允许重复
INSERT Published
VALUES (201, 1, 10, 5);
```

	book_id	publisher_id	years	contract_id
1	201	2	10	5

异常插值：



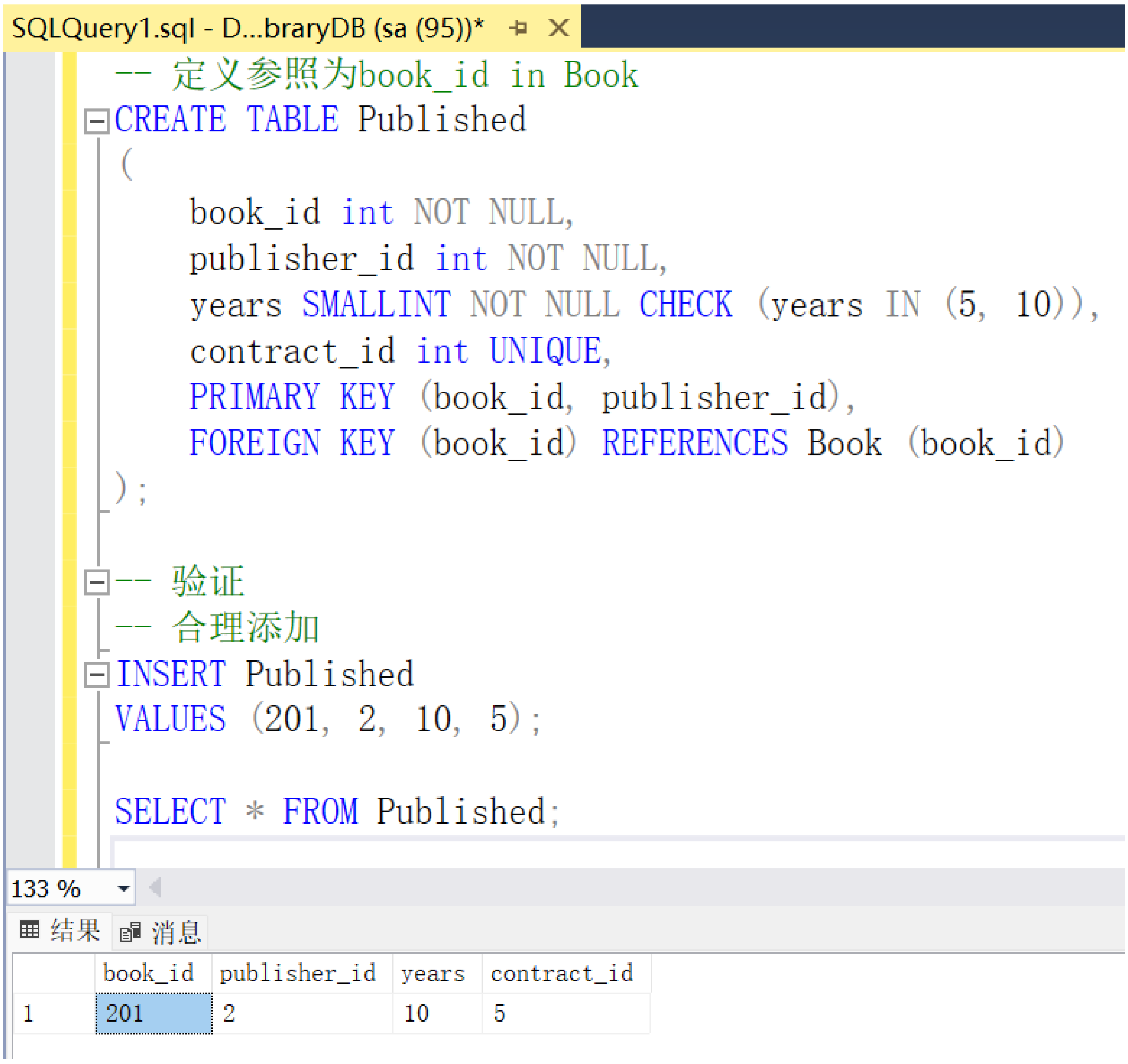
```
INSERT Published
VALUES (201, 1, NULL, 6);
-- contract_id 不允许重复
INSERT Published
VALUES (201, 1, 10, 5);
```

```
(1 行受影响)

(1 行受影响)
消息 515，级别 16，状态 2，第 18 行
不能将值 NULL 插入列 'years'，表 'LibraryDB.dbo.Published'；列不允许有 Null 值。INSERT 失败。
语句已终止。
消息 2627，级别 14，状态 1，第 21 行
违反了 UNIQUE KEY 约束"UQ__Publishe__F8D664227EB9CC44"。不能在对象"dbo.Published"中插入重复键。重复键值为 (5)。
语句已终止。
```

4. CHECK 短语：以 Published(book_id, publisher_id, years, contract_id) 为操作对象；
   - 语句：

```sql
-- 定义参照为book_id in Book
CREATE TABLE Published
(
    book_id int NOT NULL,
    publisher_id int NOT NULL,
    years SMALLINT NOT NULL CHECK (years IN (5, 10)),
    contract_id int UNIQUE,
    PRIMARY KEY (book_id, publisher_id),
    FOREIGN KEY (book_id) REFERENCES Book (book_id)
);

-- 验证
-- 合理添加
INSERT Published
VALUES (201, 2, 10, 5);

SELECT * FROM Published;

-- years 不允许为5和10以外的数
INSERT Published
VALUES (201, 1, 6, 6);
```

- 结果： 合法操作：



SQLQuery1.sql - D...braryDB (sa (95))*

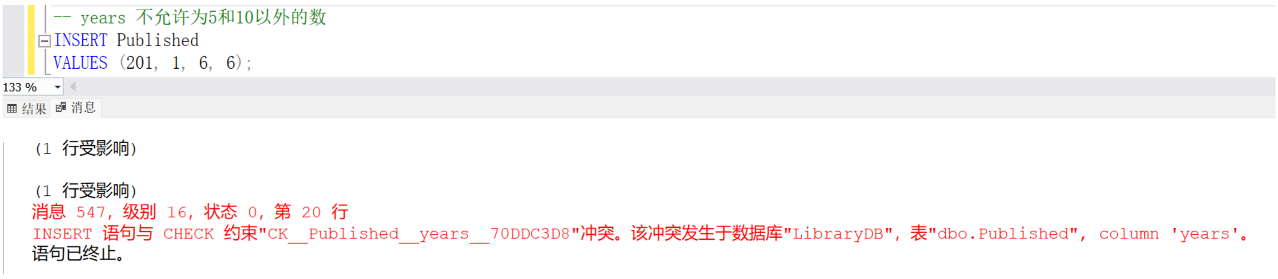
```sql
-- 定义参照为book_id in Book
CREATE TABLE Published
(
    book_id int NOT NULL,
    publisher_id int NOT NULL,
    years SMALLINT NOT NULL CHECK (years IN (5, 10)),
    contract_id int UNIQUE,
    PRIMARY KEY (book_id, publisher_id),
    FOREIGN KEY (book_id) REFERENCES Book (book_id)
);

-- 验证
-- 合理添加
INSERT Published
VALUES (201, 2, 10, 5);

SELECT * FROM Published;
```

133 %

结果 消息

	book_id	publisher_id	years	contract_id
1	201	2	10	5

异常插值：



```sql
-- years 不允许为5和10以外的数
INSERT Published
VALUES (201, 1, 6, 6);
```

133 %

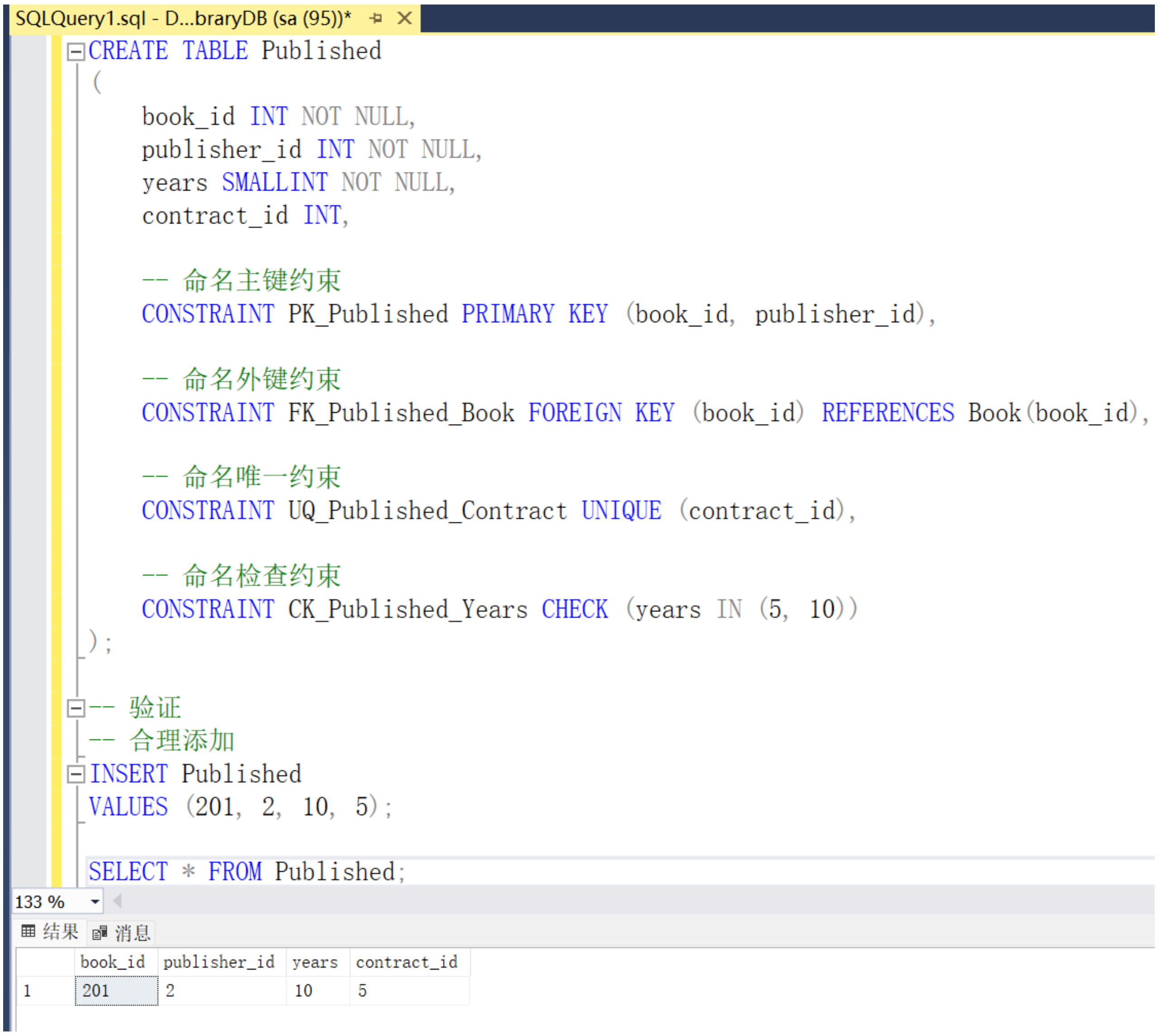
结果 消息

```
(1 行受影响)

(1 行受影响)
消息 547，级别 16，状态 0，第 20 行
INSERT 语句与 CHECK 约束"CK__Published__years__70DDC3D8"冲突。该冲突发生于数据库"LibraryDB"，表"dbo.Published", column 'years'。
语句已终止。
```

5. CONSTRAINT 子句：以 Published(book_id, publisher_id, years, contract_id) 为操作对象。
   - 语句：

```sql
CREATE TABLE Published
(
    book_id INT NOT NULL,
    publisher_id INT NOT NULL,
    years SMALLINT NOT NULL,
    contract_id INT,

    -- 命名主键约束
    CONSTRAINT PK_Published PRIMARY KEY (book_id, publisher_id),

    -- 命名外键约束
    CONSTRAINT FK_Published_Book FOREIGN KEY (book_id) REFERENCES
Book(book_id),

    -- 命名唯一约束
    CONSTRAINT UQ_Published_Contract UNIQUE (contract_id),

    -- 命名检查约束
    CONSTRAINT CK_Published_Years CHECK (years IN (5, 10))
);

-- 验证
-- 合理添加
INSERT Published
VALUES (201, 2, 10, 5);

SELECT * FROM Published;

-- years 不允许为5和10以外的数
INSERT Published
VALUES (201, 1, 6, 6);

-- contract_id 不允许重复
INSERT Published
VALUES (201, 1, 10, 5);
```

- 结果：合法操作：



SQLQuery1.sql - D...braryDB (sa (95))*

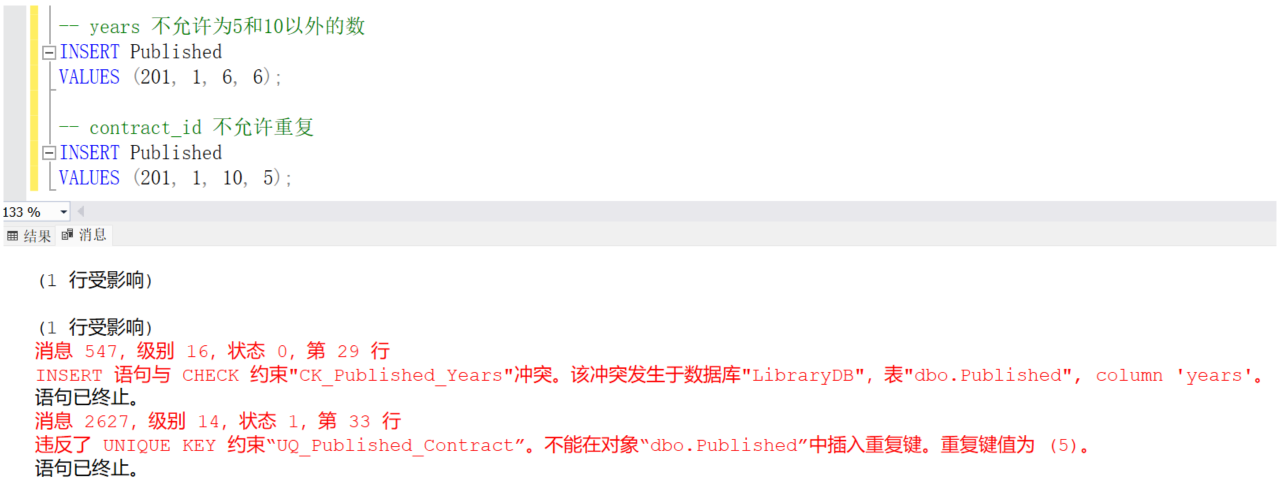
```sql
CREATE TABLE Published
(
    book_id INT NOT NULL,
    publisher_id INT NOT NULL,
    years SMALLINT NOT NULL,
    contract_id INT,

    -- 命名主键约束
    CONSTRAINT PK_Published PRIMARY KEY (book_id, publisher_id),

    -- 命名外键约束
    CONSTRAINT FK_Published_Book FOREIGN KEY (book_id) REFERENCES Book(book_id),

    -- 命名唯一约束
    CONSTRAINT UQ_Published_Contract UNIQUE (contract_id),

    -- 命名检查约束
    CONSTRAINT CK_Published_Years CHECK (years IN (5, 10))
);

-- 验证
-- 合理添加
INSERT Published
VALUES (201, 2, 10, 5);

SELECT * FROM Published;
```

133 %

结果 消息

	book_id	publisher_id	years	contract_id
1	201	2	10	5

异常插值：



```sql
-- years 不允许为5和10以外的数
INSERT Published
VALUES (201, 1, 6, 6);

-- contract_id 不允许重复
INSERT Published
VALUES (201, 1, 10, 5);
```

133 %

结果 消息

```
(1 行受影响)

(1 行受影响)
消息 547，级别 16，状态 0，第 29 行
INSERT 语句与 CHECK 约束"CK_Published_Years"冲突。该冲突发生于数据库"LibraryDB"，表"dbo.Published", column 'years'。
语句已终止。
消息 2627，级别 14，状态 1，第 33 行
违反了 UNIQUE KEY 约束"UQ_Published_Contract"。不能在对象"dbo.Published"中插入重复键。重复键值为 (5)。
语句已终止。
```