浙江大學
ZHEJIANG UNIVERSITY

实验 4　SQL 安全性控制

2024-2025 春夏学期 数据库系统

课程实验报告

姓名　王浩雄

学号　3230106032

年级　2023 级

专业　混合班（计算机科学与技术）

班级　混合 2303 班

2025 年 3 月 25 日

实验 4　SQL 安全性控制

1　实验综述

1.1　实验目的

1. 熟悉通过 SQL 进行数据安全性控制的方法。

1.2　实验内容

1. 建立表，考察表的生成者拥有该表的哪些权限。
2. 使用 SQL 的 GRANT 和 REVOKE 命令对其他用户进行授权和权力回收，考察相应的作用。
3. 建立视图，并把该视图的查询权限授予其他用户，考察通过视图进行权限控制的作用。
4. 完成实验报告。

2　实验环境

- 操作系统：
  Windows 11 Pro 24H2（64 位操作系统，基于 x64 的处理器）
- DBMS 版本：
  SQL Server Developer（64-bit）v16.0.1135.2
  SQL Server Management Studio v20.2.30.0

3　考察表的默认权限

3.1　表的定义

使用下述 SQL 代码，建立大学（University）数据库所涉及的一些表。

1. 部门表——departments

```sql
CREATE TABLE departments (
department_id INT NOT NULL,
department_name NVARCHAR(50) NOT NULL,
PRIMARY KEY (department_id),
);
```

2. 学生表——students

```sql
CREATE TABLE students (
student_id INT NOT NULL,
student_name NVARCHAR(50) NOT NULL,
gender NVARCHAR(10) CHECK (gender IN ('Male', 'Female', 'Other')
    ),
birth_date DATE,
phone_number NVARCHAR(15),
department_id INT,
PRIMARY KEY (student_id),
FOREIGN KEY (department_id) REFERENCES departments(department_id
    ) ON DELETE SET NULL ON UPDATE CASCADE
);
```

3.2　权限查询

在表的生成者用户下，使用下述 SQL 代码，查询用户对表 Students 拥有的权限。

```
SELECT * FROM fn_my_permissions('Students', 'OBJECT');
```

运行结果：

	entity_name	subentity_name	permission_name
1	Students		SELECT
2	Students		UPDATE
3	Students		UNMASK
4	Students		REFERENCES
5	Students		INSERT
6	Students		DELETE
7	Students		EXECUTE
8	Students		RECEIVE
9	Students		VIEW CHANGE TRACKING
10	Students		VIEW DEFINITION
11	Students		ALTER
12	Students		TAKE OWNERSHIP
13	Students		CONTROL
14	Students	student_id	SELECT
15	Students	student_name	SELECT
16	Students	gender	SELECT
17	Students	birth_date	SELECT
18	Students	phone_number	SELECT
19	Students	department_id	SELECT
20	Students	student_id	UPDATE
21	Students	student_name	UPDATE
22	Students	gender	UPDATE
23	Students	birth_date	UPDATE
24	Students	phone_number	UPDATE
25	Students	department_id	UPDATE
26	Students	student_id	REFERENCES
27	Students	student_name	REFERENCES
28	Students	gender	REFERENCES
29	Students	birth_date	REFERENCES
30	Students	phone_number	REFERENCES
31	Students	department_id	REFERENCES

由上述运行结果可知，表的生成者默认拥有对所创建表的完全控制权限，具体包括以下内容：

权限名称	描述
SELECT	允许用户查询表中的数据。
INSERT	允许用户向表中插入新数据。
UPDATE	允许用户更新表中的数据。
DELETE	允许用户删除表中的数据。
ALTER	允许用户修改表的结构（如添加、删除或修改列）。
REFERENCES	允许用户创建外键约束，引用该表。
CONTROL	完全控制权限，包括所有权和权限管理。拥有该权限的用户可以管理表的所有权限。
TAKE OWNERSHIP	允许用户获取表的所有权。

4　考察权限的授予和回收

4.1　用户配置

在 SQL Server 的用户管理系统中，存在“登录名”和“数据库用户”两个不同的概念。其中，登录名是服务器级别的对象，用于控制用户是否可以连接到 SQL Server 实例；数据库用户是数据库级别的对象，与登录名关联后，用户可以在特定数据库中执行操作。登录名和数据库用户通常是一对一的关系，但一个登录名可以在不同数据库中关联不同的用户名。

为执行本次实验，首先需要创建一个新的登录名“user001”，然后将该登录名与 University 数据库的用户“dbuser001”进行绑定。完成上述操作后，使用登录名“user001”连接到 SQL Server 实例后，即可以使用数据库用户“dbuser001”的权限对 University 数据库进行操作。

运行结果：

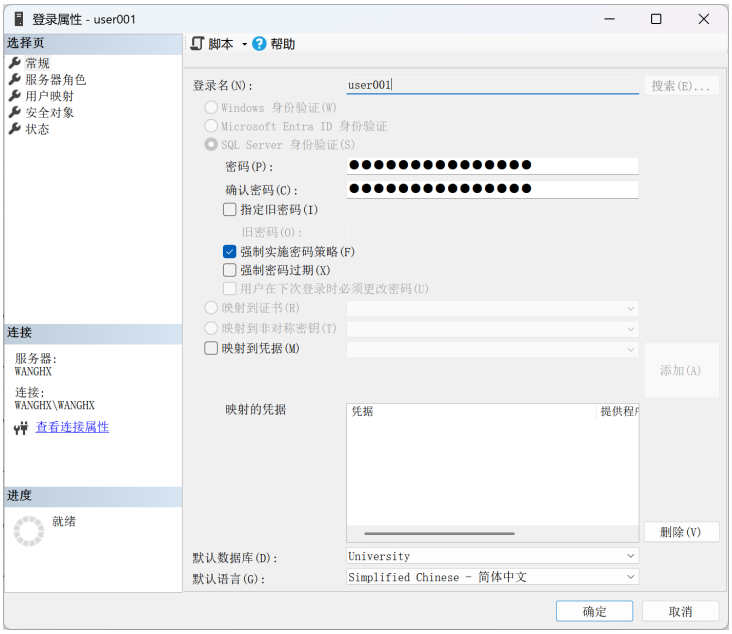


图 1: 登录名的设置页面

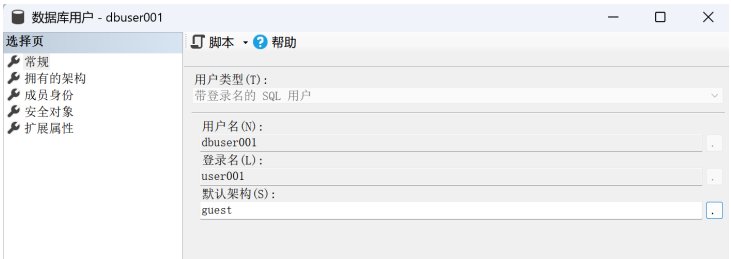
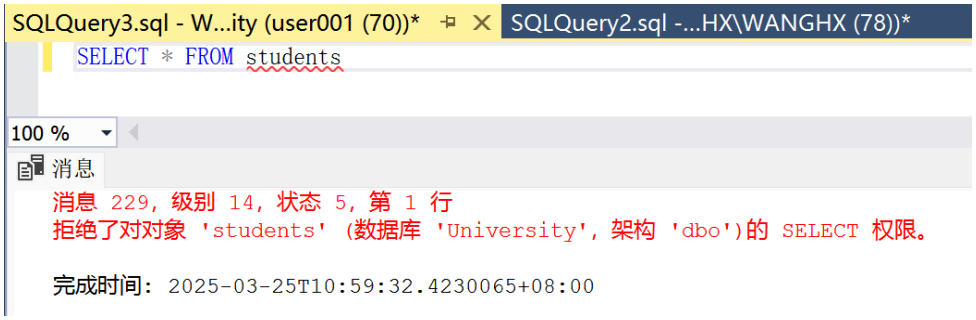
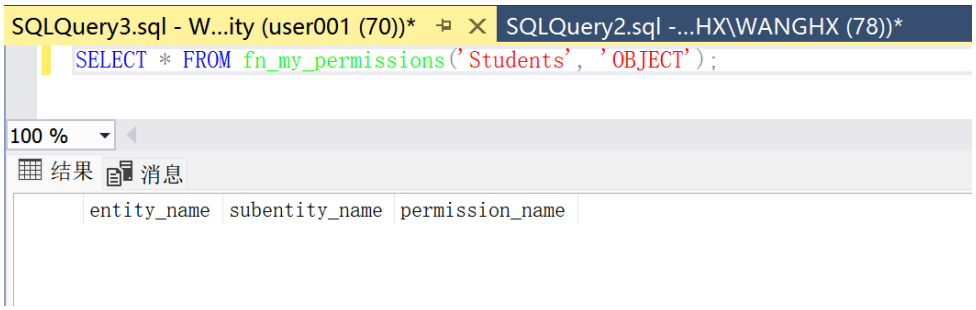


图 2: 数据库用户的设置页面

在对数据库用户 dbuser001 授予任何权限前，该用户的默认身份为 guest。切换至 dbuser001 用户，使用该用户尝试读取 Students 表，提示权限不足。



切换至 dbuser001 用户，查询该用户对表 Students 拥有的权限，结果为空白，表明身份仅为 guest 的数据库用户对任何表不具有任何权限。

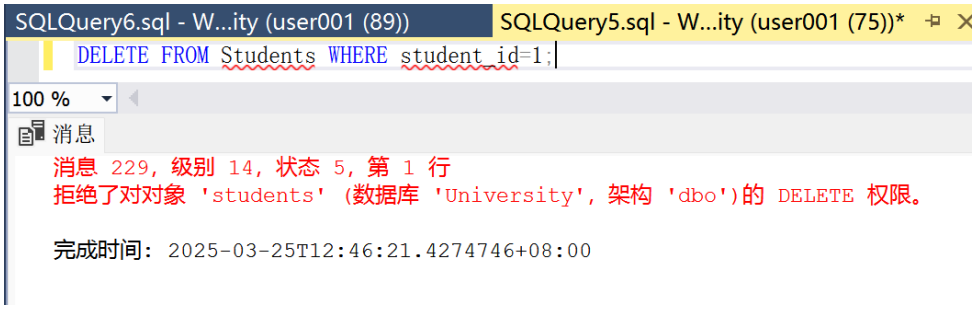


4.2　权限的授予

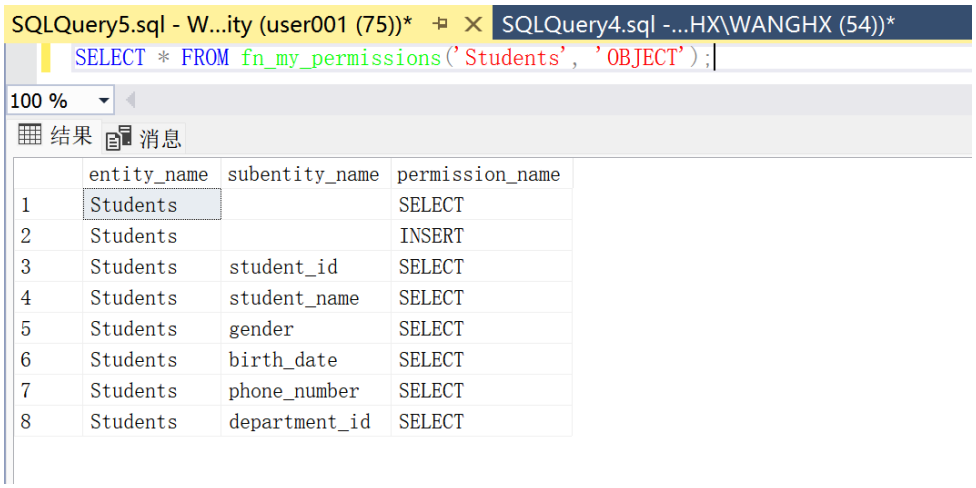
使用管理员用户 dbo 为用户 dbuser001 授予 Students 表的 SELECT 与 INSERT 权限，使用下述 SQL 代码：

```
1 GRANT SELECT,INSERT ON Students TO dbuser001;
```

切换至 dbuser001 用户，使用该用户尝试读取 Students 表，能够正常读取；尝试删除 Students 表的记录时，提示权限不足。



切换至 dbuser001 用户，查询该用户对表 Students 拥有的权限，结果为下图所示，表明 SELECT 与 INSERT 权限授予成功。



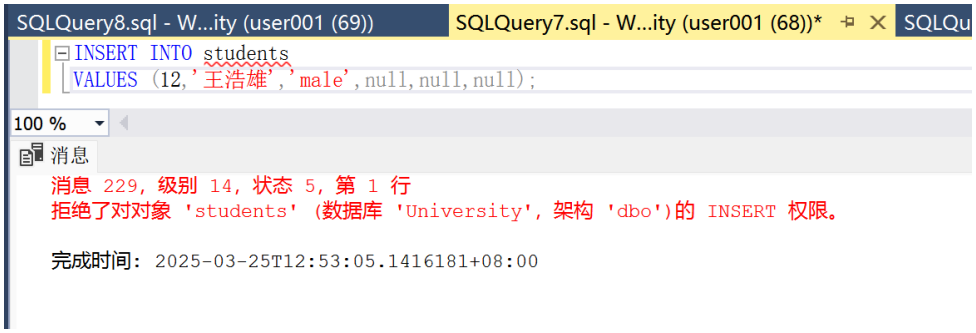
	entity_name	subentity_name	permission_name
1	Students		SELECT
2	Students		INSERT
3	Students	student_id	SELECT
4	Students	student_name	SELECT
5	Students	gender	SELECT
6	Students	birth_date	SELECT
7	Students	phone_number	SELECT
8	Students	department_id	SELECT

4.3　权限的回收

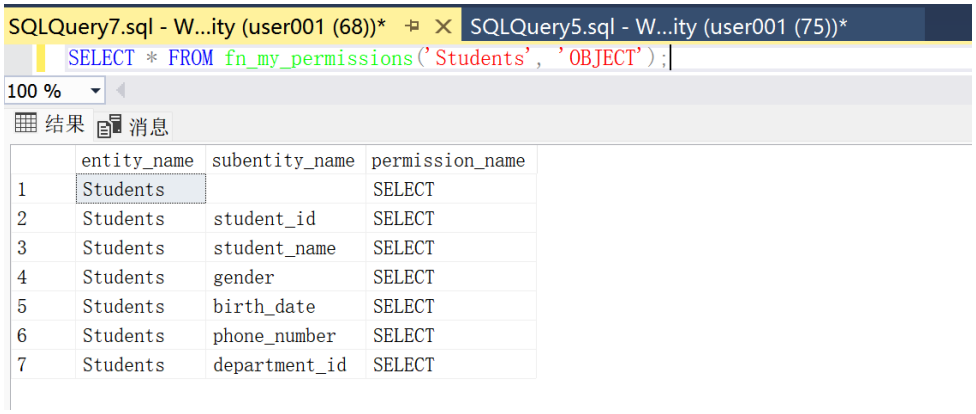
使用管理员用户 dbo 为用户 dbuser001 收回 Students 表的 INSERT 权限，使用下述 SQL 代码：

```
1 REVOKE INSERT ON Students FROM dbuser001;
```

切换至 dbuser001 用户，使用该用户尝试读取 Students 表，能够正常读取；尝试向 Students 表插入记录时，提示权限不足。



切换至 dbuser001 用户，查询该用户对表 Students 拥有的权限，结果为下图所示，表明 INSERT 权限收回成功。



	entity_name	subentity_name	permission_name
1	Students		SELECT
2	Students	student_id	SELECT
3	Students	student_name	SELECT
4	Students	gender	SELECT
5	Students	birth_date	SELECT
6	Students	phone_number	SELECT
7	Students	department_id	SELECT

5　通过视图进行权限控制

5.1　视图的定义

在数据库 University 中，我们已定义了两个表：departments 和 students。通过定义如下的视图，我们实现一个输入学生姓名查询学生所属部门名称的功能，同时实现对学生学号、部门代号的隐藏。

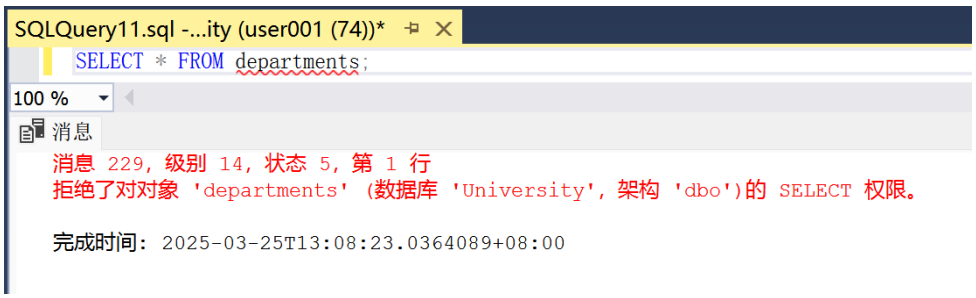
```
CREATE VIEW student_dept AS
SELECT student_name, department_name
FROM students, departments
WHERE students.department_id = departments.department_id;
```

5.2　通过视图进行权限授予

使用管理员用户 dbo 为用户 dbuser001 授予 student_dept 视图的 SELECT 权限，使用下述 SQL 代码：

```
GRANT SELECT ON student_dept TO dbuser001;
```

切换至 dbuser001 用户，使用该用户尝试读取 Students 表或 Departments 表，均提示权限不足。



切换至 dbuser001 用户，使用该用户尝试读取 student_dept 视图，能够正常读取，表明成功通过视图为该用户授予了有限的权限。