**Name: Dikshya Kafle**

**Student Number: 2018380039**

**Experiment 3**

Data Integrity and Security

**Goal**
1. To practice how to define the data integrity.
2. To practice how to create users
3. To practice how to grant/revoke privileges of databases and tables.

**Content**
1. Add the following constraint and index with GUI (for student database).
   **(1) Not null：add the not null constraint to S(SBIRTH).**

   To add the not null constraint to Classroom (capacity), we go to university->Tables->Right click on Classroom table -> Design. Then we just   tick mark on the "Not null" corresponding to the **"SBIRTH"** column.

| Column Name | Datatype | PK | NN | UQ | B | UN | ZF | AI | G | Default/Expression |
|---|---|---|---|---|---|---|---|---|---|---|
| SNAME | VARCHAR(20) | ☑ | ☑ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | |
| SGENDER | VARCHAR(20) | ☐ | ☑ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | |
| SBIRTH | DATE | ☐ | ☑ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | |
| SDEPT | VARCHAR(5) | ☐ | ☑ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | |

   **(2) Primary key：set the SNO as the primary key.**

| Table Name: | sc | | | | | | | | | Schema: |
|---|---|---|---|---|---|---|---|---|---|---|

| Column Name | Datatype | PK | NN | UQ | B | UN | ZF | AI | G | |
|---|---|---|---|---|---|---|---|---|---|---|
| SNO | VARCHAR(5) | ☑ | ☑ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | |
| CNO | INT | ☐ | ☑ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | |
| GRADE | DECIMAL(2,0) | ☐ | ☑ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | |
| | | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | |

   **(3) Unique constraint：add unique constraint for the primary key of SNAME, the constraint name is uk_Sname.**
   Go to "Indexe Name" in the Design window of the table Instructor. Click "Add". In the grid under General, click Type and choose Unique Key from the drop-down list box to the right of the property. Choose the "sname" column, then set the name of the constraint as" uk_sname" as shown below:
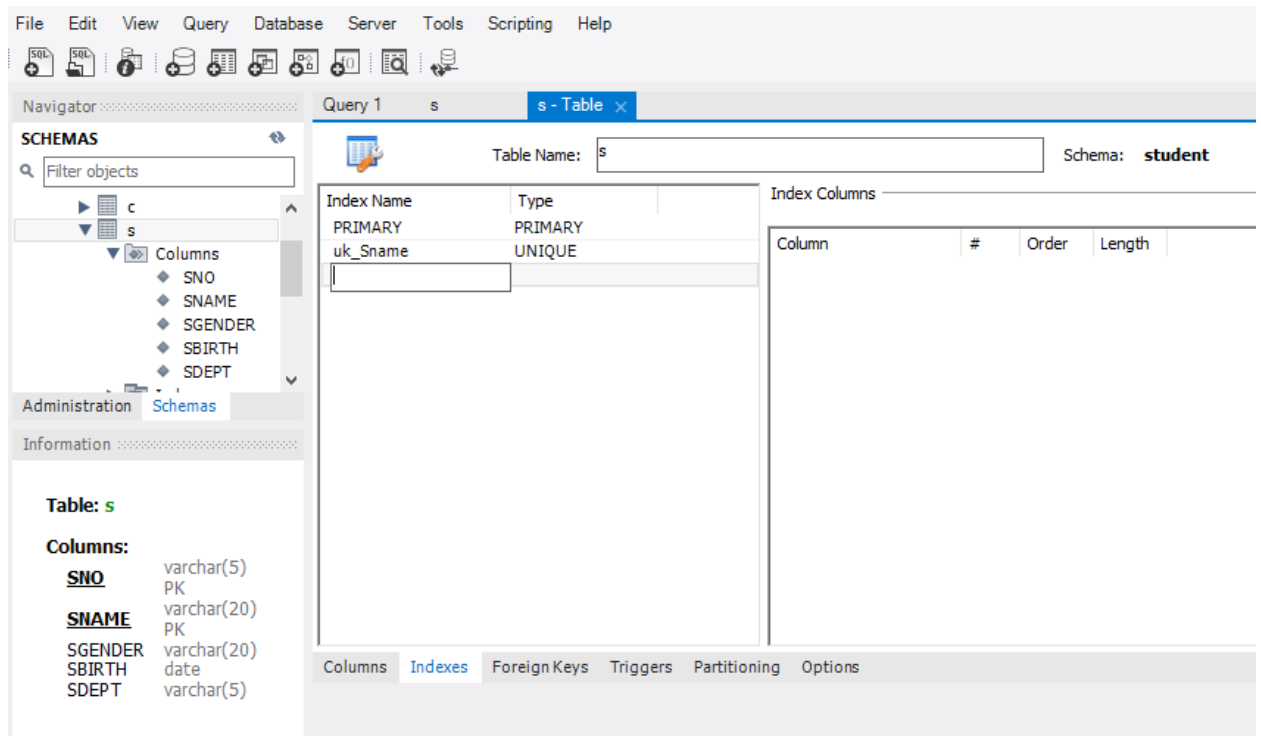
**(4) Default constraint：add the default value to S(SGENDER), the default value is "male".**
On the "Design" window of the s table, we click on the SGENDER attribute. A column properties window shows up at the bottom. We set the "Default /Expression " property to male as shown below:
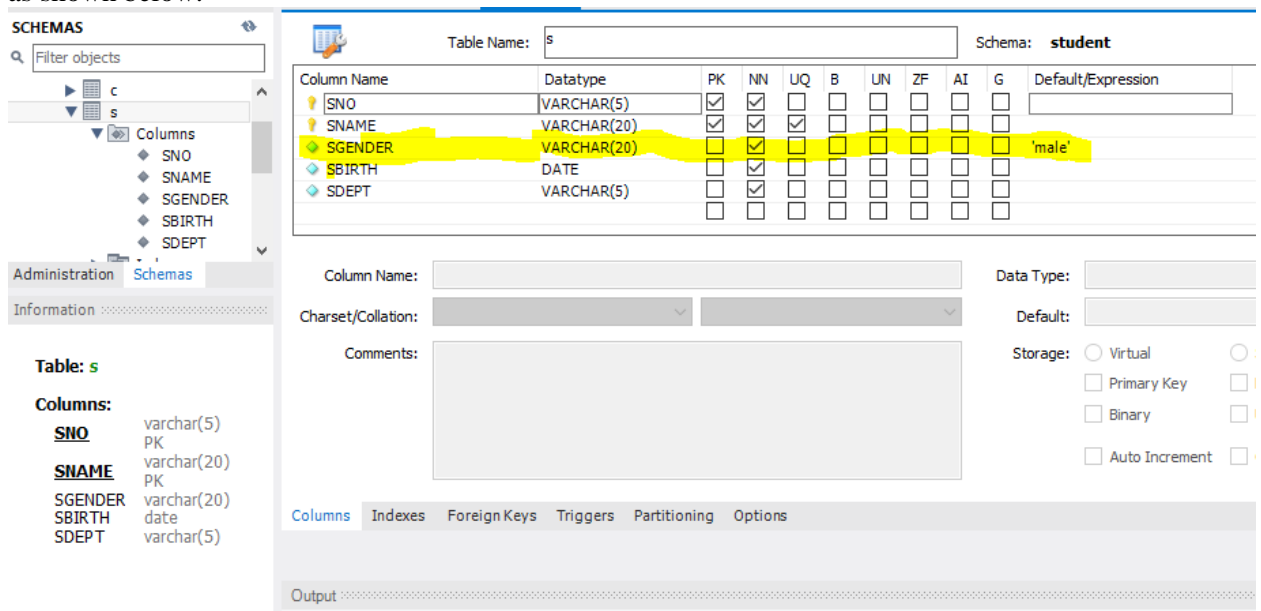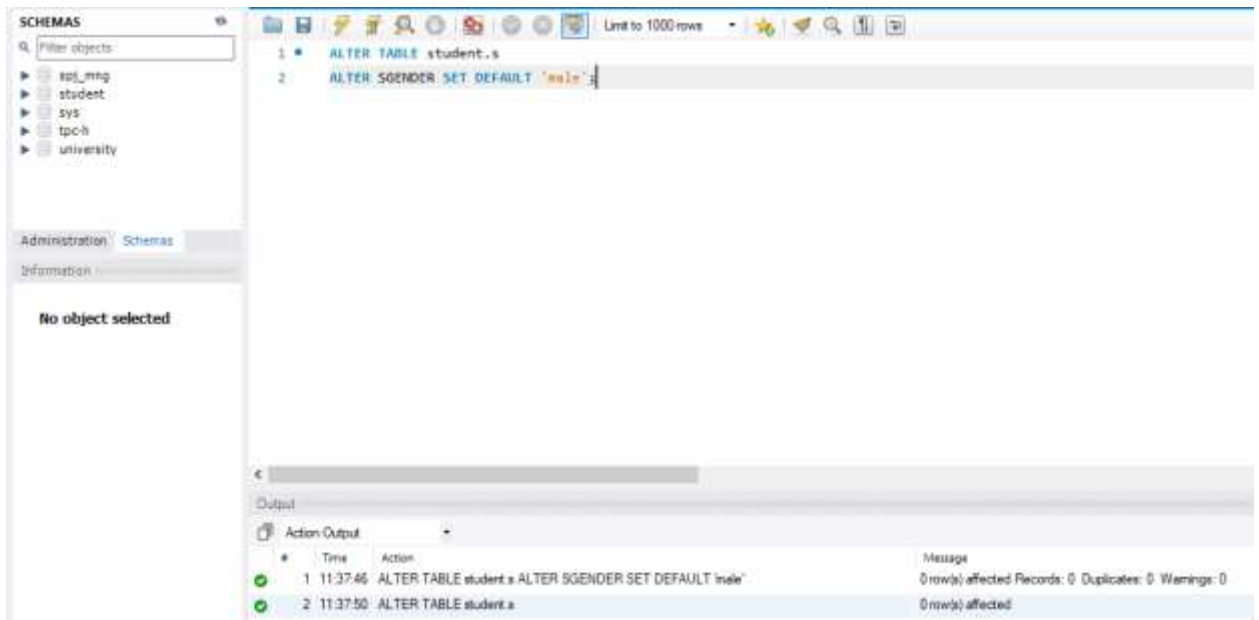
2. Add forgein key constraints to table **SC** in the database **Student** with GUI, set SNO(foreign key name is: sc_fk_sid) as a foreign key referencing table S, and set CNO as another foreign key referencing table , name it with sc_fk_ cno. Try and validate different strategies in violation of the foreign key constraints :

    NO Action/restrict/cascade/set to null
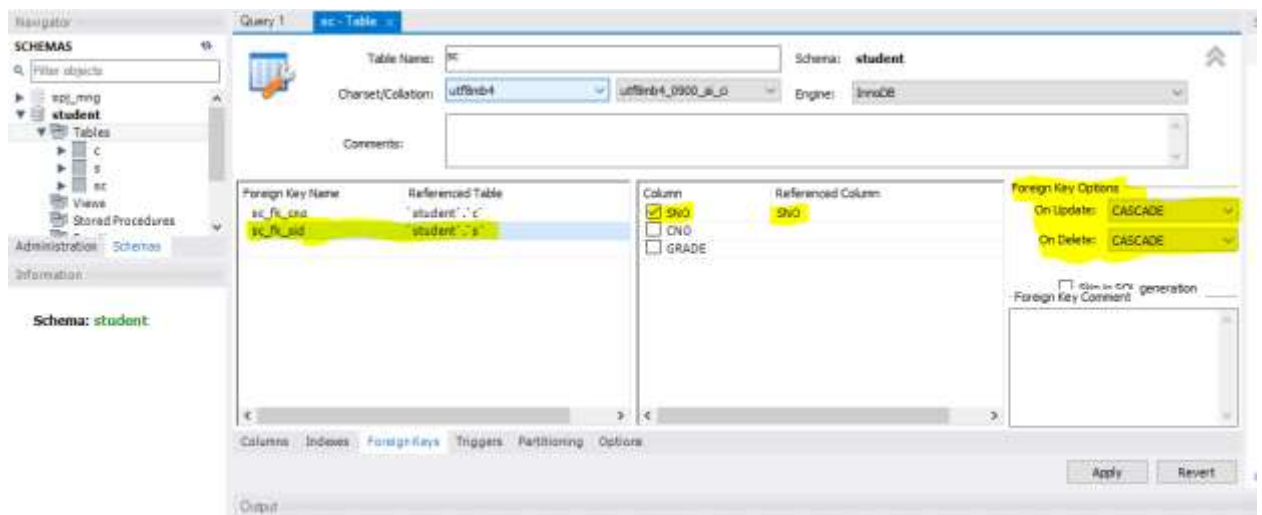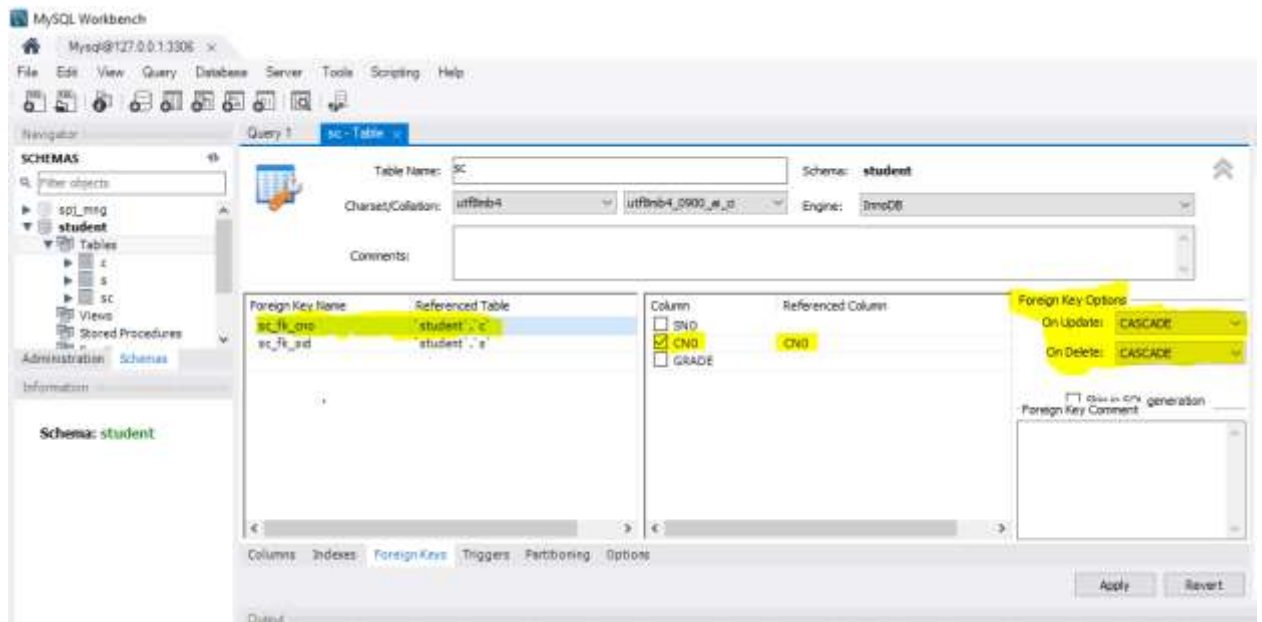
3. **Drop the three tables in database Student, and create some tables through SQL statements with the following constraints.**
   By clicking the schema we get to tables we are able to choose each table at once as shown below in the picture and by choosing drop table we drop each table.
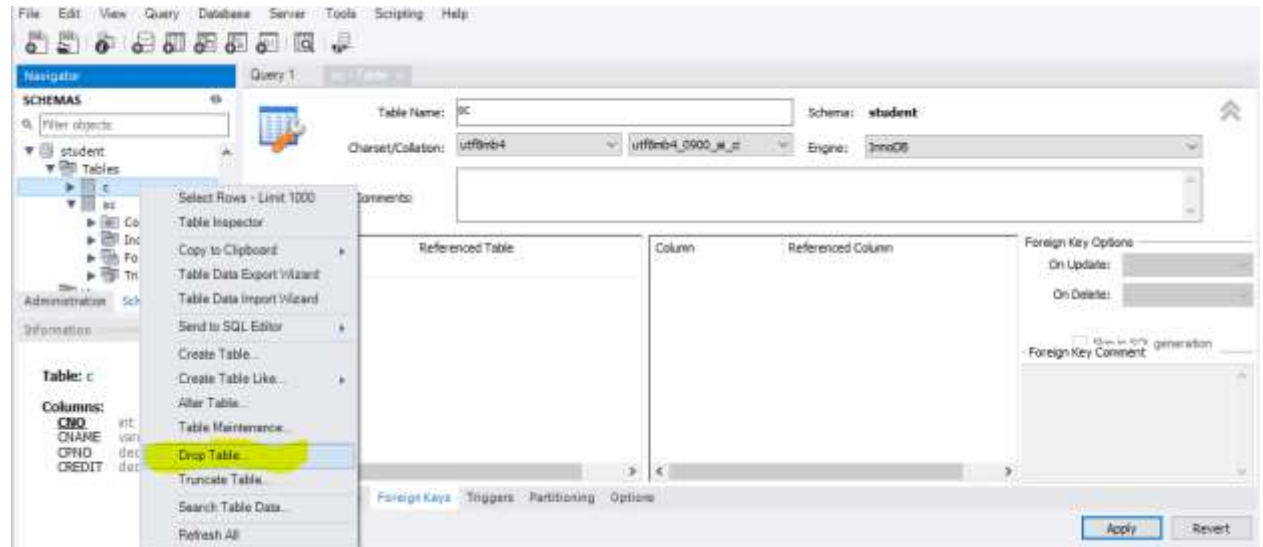


✧ Table **S**: same to the constraints set in question 1(1).
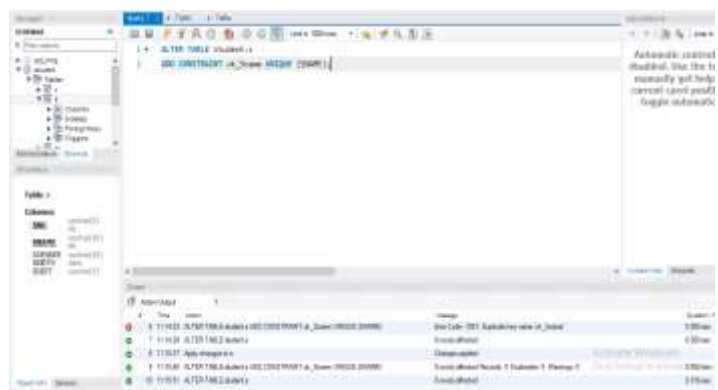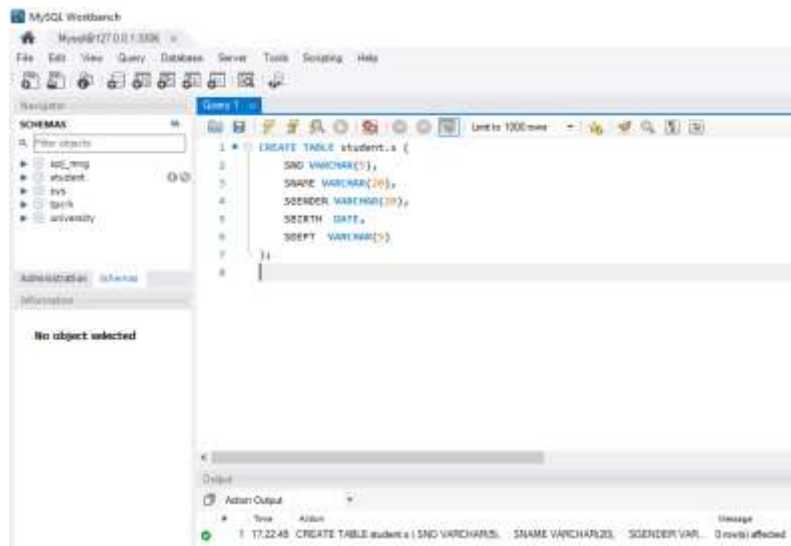
✧ Table *C*: set CPNO as a foreign key，referencing table C with the attribute CNO

```sql
1  CREATE TABLE table_name (
2      CNO INT PRIMARY KEY,
3      CNAME VARCHAR(20),
4      CPNO INT UNIQUE KEY,
5      CREDIT INT
6  );
7
```

```sql
1  ALTER TABLE student.c
2  ADD CONSTRAINT FK_CPNO
3  FOREIGN KEY (CPNO) REFERENCES c(CNO);
```

♦ Table **SC**: set the foreign key constraints same to question2, and set the valid range of attribute GRADE with [0,100]. In addition, add one attribute ID to table SC, and set it as a primary key, and it can increase automatically. When a new tuple is inserted to the table, its(ID) value will increase by +1.
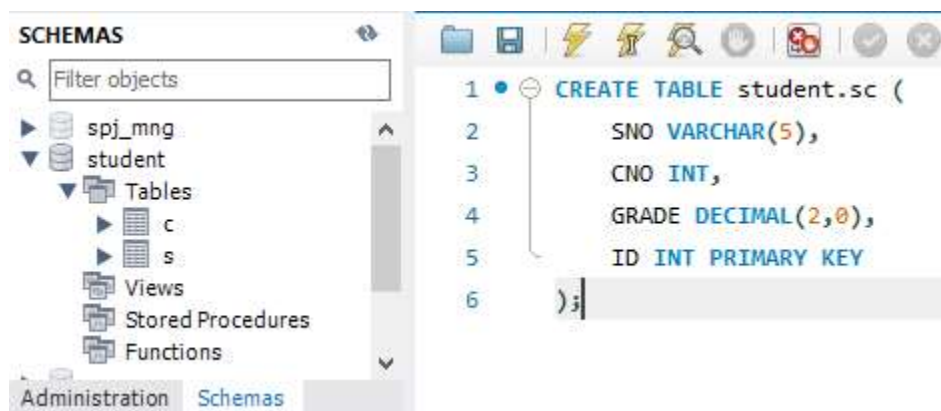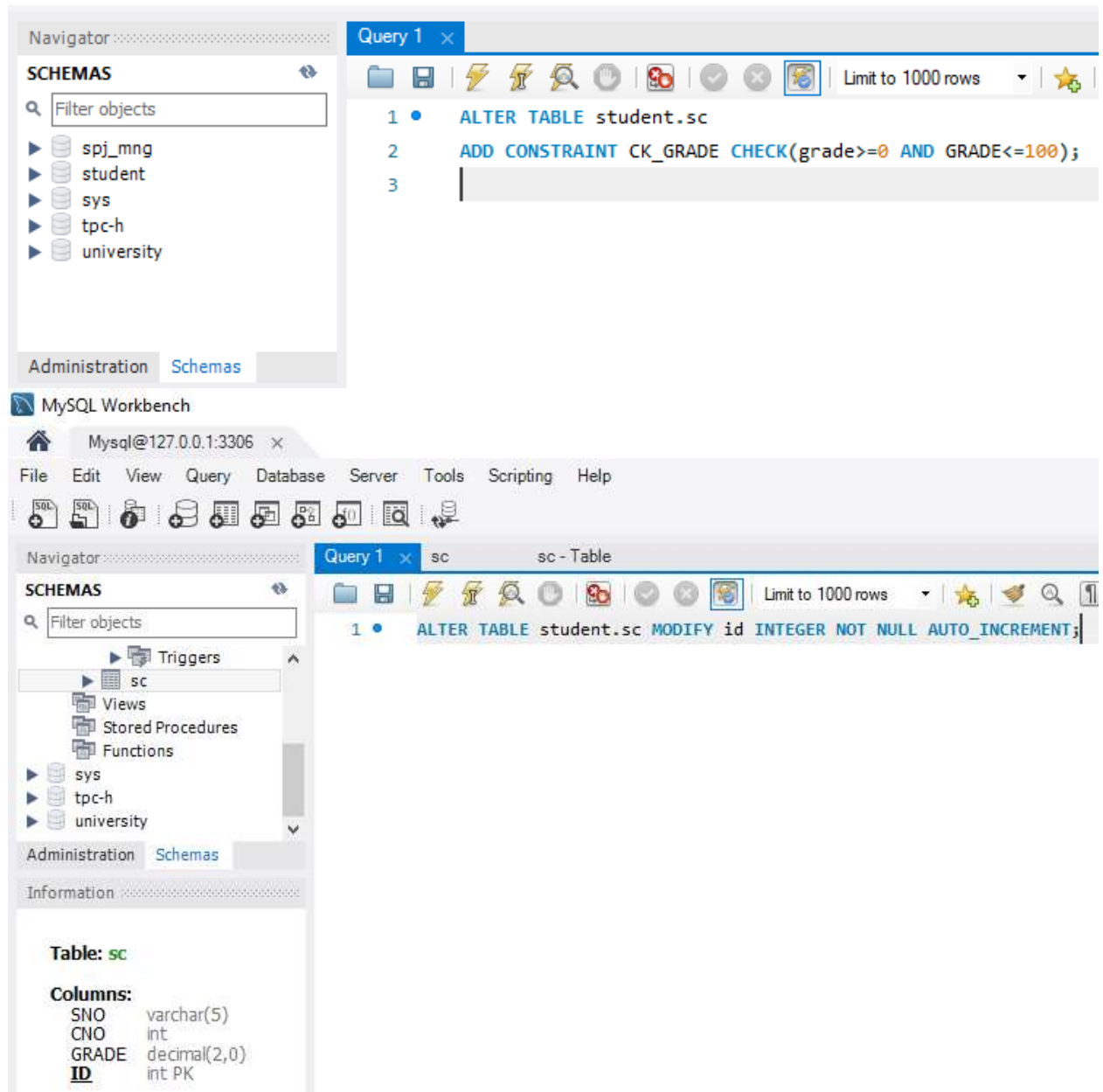
**4. Add or remove the following integrity constraints with SQL language.**

(1) Add constraint: SGENDER's value in table S can only be "male" or "female".

(2) Delete the SGENDER value constraint created in table S.



(3) Remove the foreign key constraints from the SC table.

(4) Add a new column TNAME (indicating the name of the student's tutor) in the student table S, and require that the tutor name must be all uppercase or lowercase letters, and the length cannot be less than 8 characters (hint: char can be used_ Length() function and regular expression).
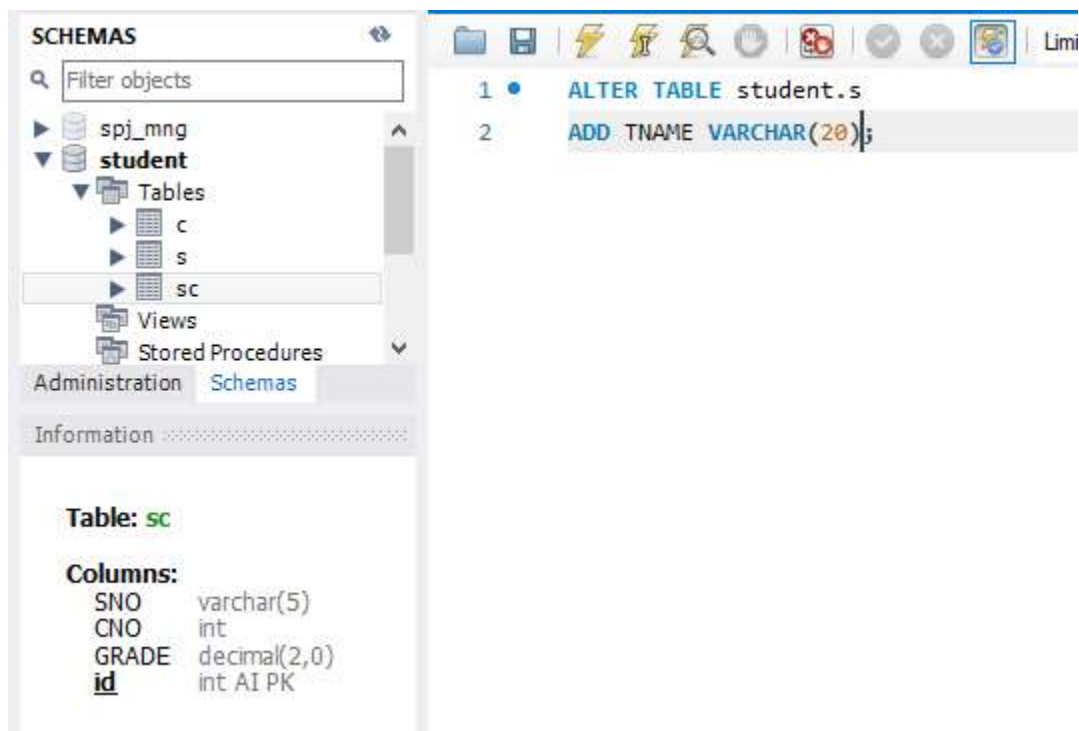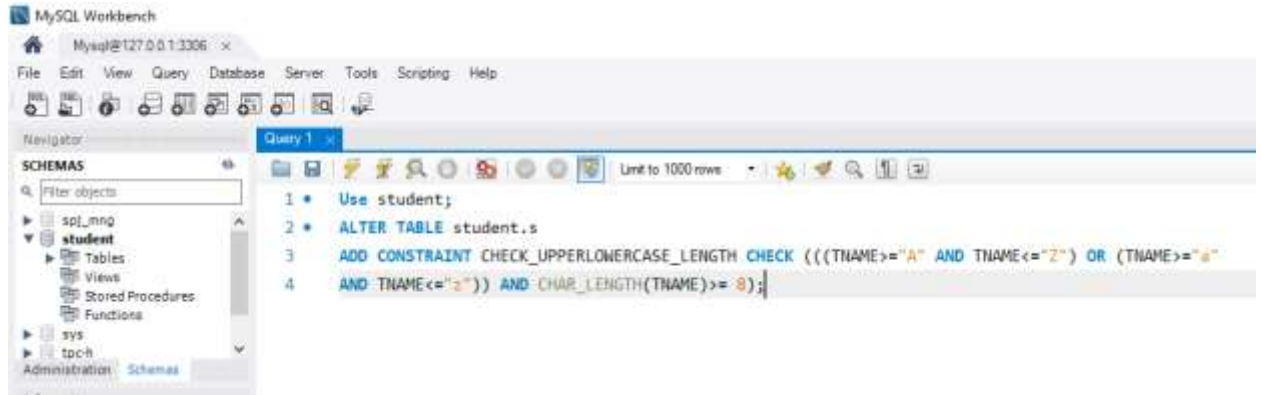
Try to finish the following requirements by connecting to the local database server

5．Create and authorize new users in GUI.
  (1) Create two users who can access the current student database: Wang Ming and Li Yong.
     To create the new user we go to administration-user and privileges.
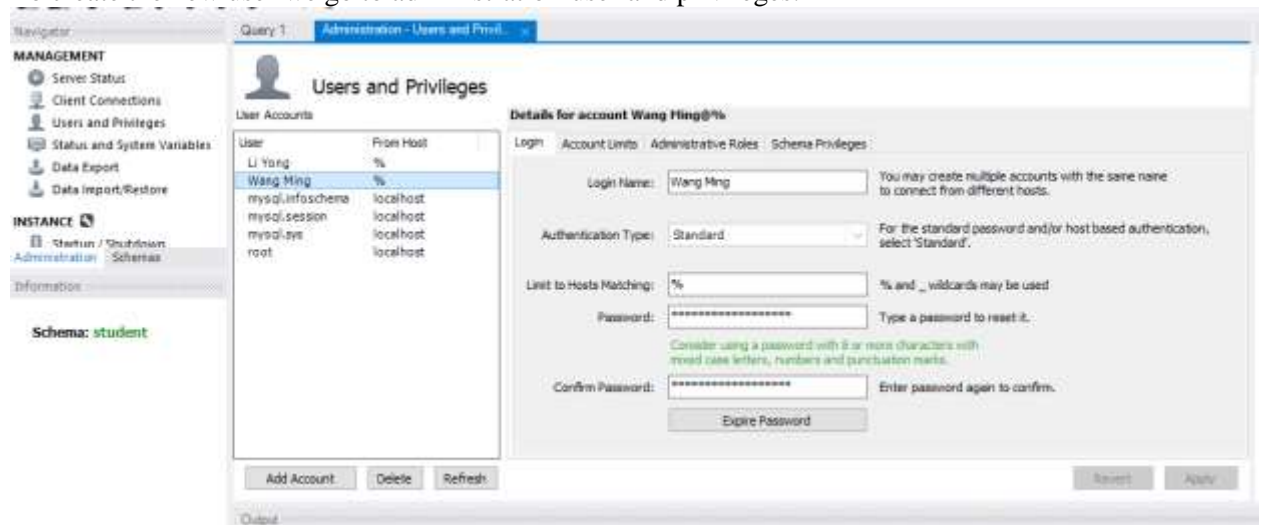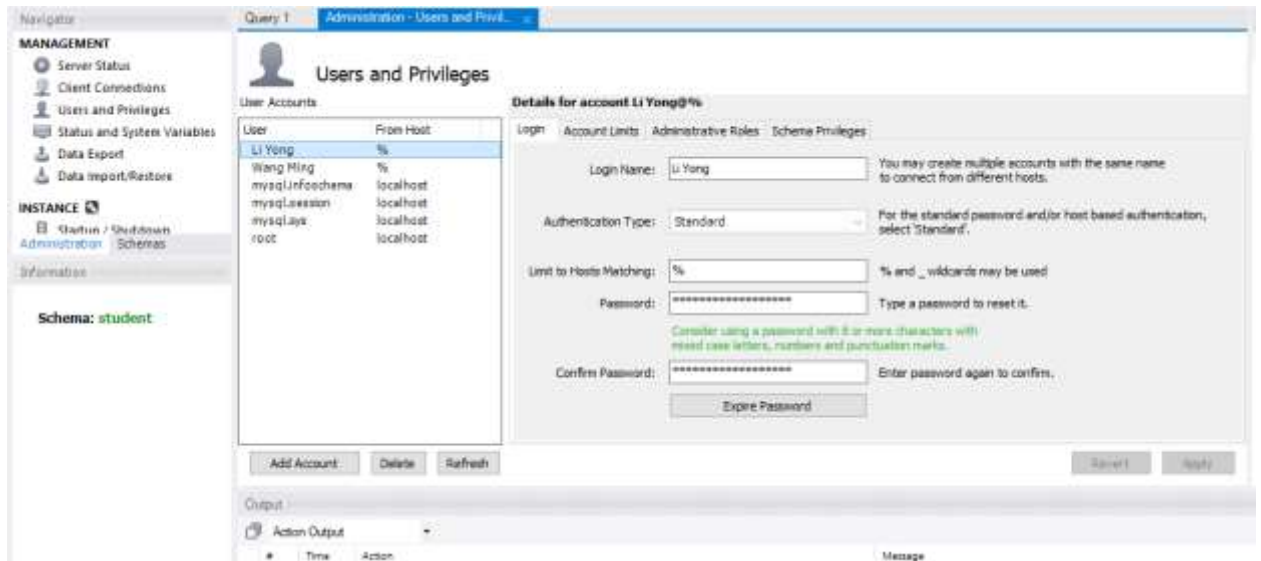
(2) Complete the following authorizing:

① User Wang Ming has the priviliges to select and insert all tables.

② User Li Yong has the priviliges of select, insert, delete, update and create on the database.

**Users and Privileges**

**User Accounts**

| User | From Host |
|------|-----------|
| Wang Ming | % |
| mysql.infoschema | localhost |
| mysql.session | localhost |
| mysql.sys | localhost |
| newuser | % |
| root | localhost |

**Details for account newuser@%**

Login | Account Limits | Administrative Roles | Schema Privileges

Login Name: Li Yong
You may create multiple accounts with the same name to connect from different hosts.

Authentication Type: Standard
For the standard password and/or host based authentication, select 'Standard'.

Limit to Hosts Matching: %
% and _ wildcards may be used

Password: *************
Type a password to reset it.

Strong password.

Confirm Password: *************
Enter password again to confirm.

Expire Password

Add Account | Delete | Refresh

Revert | Apply

Output

---

**New Schema Privilege Definition**

Select the Schema for which the user 'newuser' will have the privileges you want to define.

**Schema**

○ All Schema (%)
This rule will apply to any schema name.

○ Schemas matching pattern:
This rule will apply to schemas that match the given name or pattern. You may use _ and % as wildcards in a pattern.
Escape these characters with \ in case you want their literal value.

● Selected schema: student
Select a specific schema name for the rule to apply to.

---

**Users and Privileges**

MANAGEMENT
- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE
- Startup / Shutdown
- Administration  Schemas

Information

**Schema: student**

**User Accounts**

| User | From Host |
|------|-----------|
| Wang Ming | % |
| mysql.infoschema | localhost |
| mysql.session | localhost |
| mysql.sys | localhost |
| newuser | % |
| root | localhost |

**Details for account newuser@%**

Login | Account Limits | Administrative Roles | Schema Privileges

| Schema | Privileges |
|--------|------------|
| student | CREATE, DELETE, INSERT, SELECT, UPDATE |

Schema and Host fields may use % and _ wildcards.
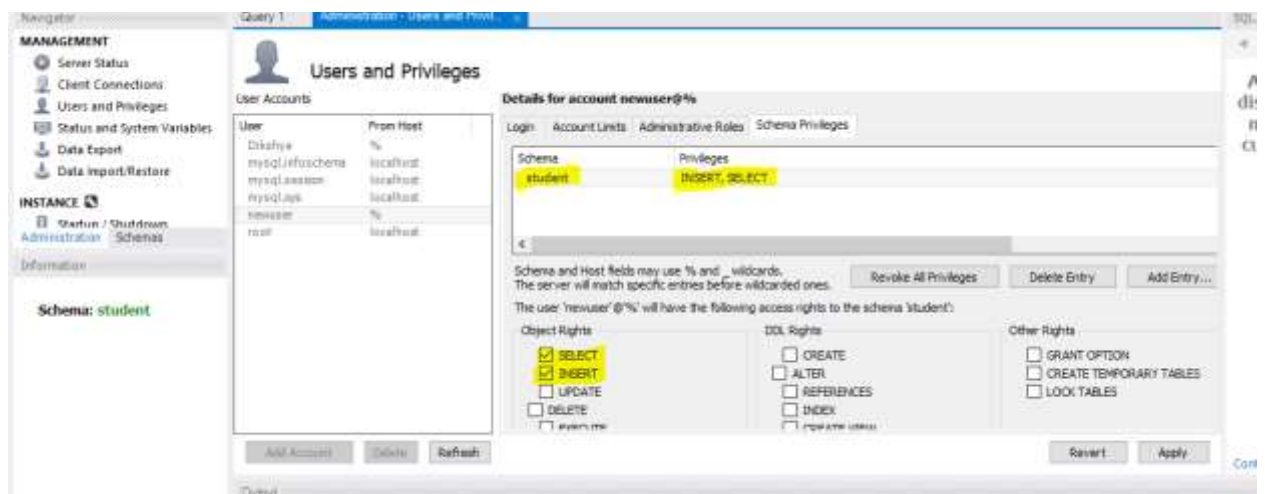The server will match specific entries before wildcarded ones.

The user 'newuser'@'%' will have the following access rights to the schema 'student':

Revoke All Privileges | Delete Entry | Add Entry...

**Object Rights**
- ☑ SELECT
- ☑ INSERT
- ☑ UPDATE
- ☑ DELETE
- ☐ EXECUTE

**DDL Rights**
- ☑ CREATE
- ☐ ALTER
- ☐ REFERENCES
- ☐ INDEX
- ☐ CREATE VIEW

**Other Rights**
- ☐ GRANT OPTION
- ☐ CREATE TEMPORARY TABLES
- ☐ LOCK TABLES

Revert | Apply

2. Use SQL statement to authorize and withdraw permissions and verify permissions.

Requirements: create relevant users and specified database tables, complete authorization and authority verification for each question, and then withdraw the authority and verify the authority.

(1) User Wang Ming has query privilege on two tables.

```
1    create user wangming@localhost;
2 •  grant select on student.s to wangming@localhost;
3
4 •  create user wangming@localhost;
5 •  grant select on student.c to wangming@localhost;
```

(2) User Li Yong has insert and delete privileges on the two tables.
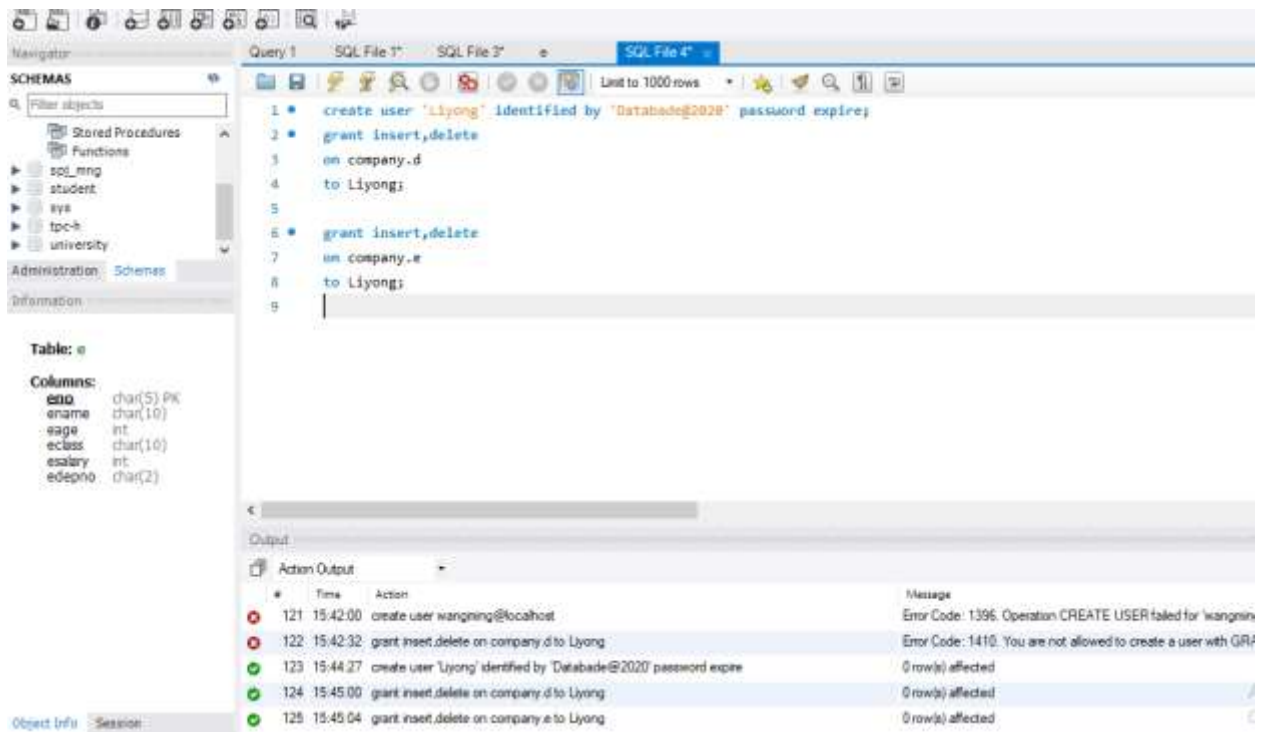
**(3)** Each instructor only has the right to query his own record.



**(4)** User Liu Xing has query privilege on employee table and update privilige on salary field.

```
15 ●    create user 'Liuxing' identified by 'Databade@2020' password expire;
16 ●    grant select,update(esalary)
17       on company.e
18       to Liuxing;
19       |
```

Output

Action Output ▾

| # | Time | Action | Message |
|---|---|---|---|
| ✗ | 126 15:46:06 | grant select on company.d to public | Error Code: 1410. You are not allowed to crea |
| ✓ | 127 15:47:06 | create user 'public' identified by 'Databade@2020' password expire | 0 row(s) affected |
| ✓ | 128 15:47:10 | grant select on company.d to public | 0 row(s) affected |
| ✓ | 129 15:49:04 | create user 'Liuxing' identified by 'Databade@2020' password expire | 0 row(s) affected |
| ✓ | 130 15:49:04 | grant select,update(esalary) on company.e to Liuxing | 0 row(s) affected |

(5) User Zhang Xin has the right to modify the structure of the two tables.

```
20 ●    create user 'Zhangxin' identified by 'Databade@2020' password expire;
21 ●    grant update
22       on company.d
23       to Zhangxin;
24 ●    grant update
25       on company.e
26       to Zhangxin;
27       |
```

Output

Action Output ▾

| # | Time | Action | Message |
|---|---|---|---|
| ✓ | 129 15:49:04 | create user 'Liuxing' identified by 'Databade@2020' password expire | 0 row(s) affected |
| ✓ | 130 15:49:04 | grant select,update(esalary) on company.e to Liuxing | 0 row(s) affected |
| ✓ | 131 15:53:08 | create user 'Zhangxin' identified by 'Databade@2020' password expire | 0 row(s) affected |
| ✓ | 132 15:53:08 | grant update on company.d to Zhangxin | 0 row(s) affected |
| ✓ | 133 15:53:08 | grant update on company.e to Zhangxin | 0 row(s) affected |

(6) User Zhou Ping has all permissions on the two tables, and has the authority to authorize other users.
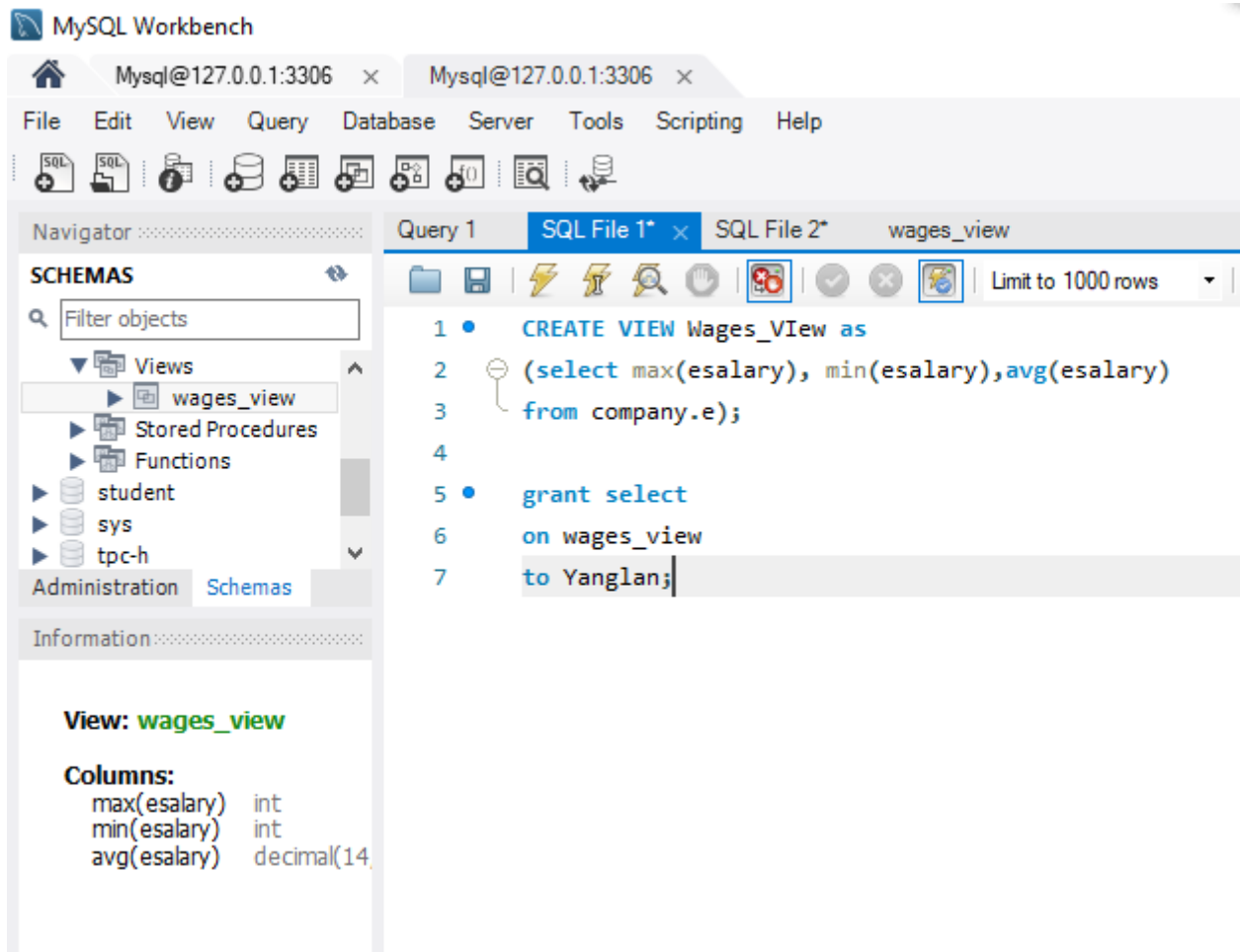
```
28 ●     create user 'Zhouping' identified by 'Databade@2020' password expire;
29 ●     grant all privileges
30       on worker
31       to Zhouping
32       with grant option;
33 ●     grant all privileges
34       on apartment
35       to Zhouping
36       with grant option;
37       |
```

Output

Action Output ▾

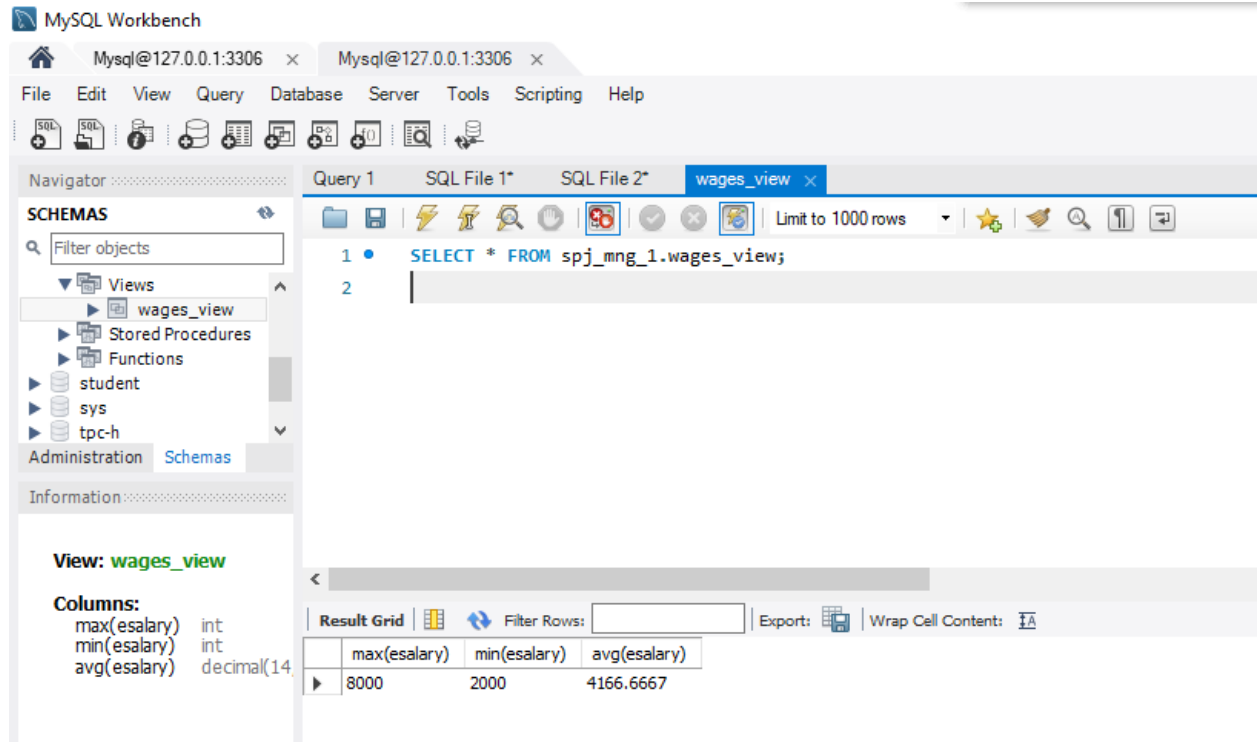| # | Time | Action | Message |
|---|------|--------|---------|
| ✓ | 132 15:53:08 | grant update on company.d to Zhangxin | 0 row(s) affected |
| ✓ | 133 15:53:08 | grant update on company.e to Zhangxin | 0 row(s) affected |
| ✓ | 134 15:54:55 | create user 'Zhouping' identified by 'Databade@2020' password expire | 0 row(s) affected |
| ✓ | 135 15:54:55 | grant all privileges on worker to Zhouping with grant option | 0 row(s) affected |
| ✓ | 136 15:54:55 | grant all privileges on apartment to Zhouping with grant option | 0 row(s) affected |

(7) User Yang Lan has the privilige to query the maximum wage, minimum wage and average wage from each department employee, but he cannot view the salary of each person.

Mysql@127.0.0.1:3306    ×        Mysql@127.0.0.1:3306    ×

File    Edit    View    Query    Database    Server    Tools    Scripting    Help

Navigator

SCHEMAS

Filter objects

▼ Views
  ▶ wages_view
  ▶ Stored Procedures
  ▶ Functions
▶ student
▶ sys
▶ tpc-h

Administration    Schemas

Information

**View: wages_view**

**Columns:**
max(esalary)    int
min(esalary)    int
avg(esalary)    decimal(14,

Query 1    SQL File 1*  ×    SQL File 2*    wages_view

Limit to 1000 rows

```
1   CREATE VIEW Wages_VIew as
2   (select max(esalary), min(esalary),avg(esalary)
3   from company.e);
4
5   grant select
6   on wages_view
7   to Yanglan;
```

**Problems:**

For the view and index experiment, in particular, we faced a lot of technical failures. A basic code typing mistake, for example, prompted me to redo the whole database again. Even some of the questions were really difficult.

**Solutions:**

We were able to overcome most of our mistakes with the very helpful guidance and assistance of Professor. In order to identify our remaining errors that led to fatal errors, we also used the internet and our textbook, which also provided us with some SQL codes that we have not studied.

**Summary:**

In the first experiment, we worked on data integrity. We also learned how to add and drop multiple constraints using the GUI and SQL, such as primary key, foreign key, not null, unique and default values constraints. We knew how we could explain the view and the index. Then we moved to create logins and users and try granting VIEW permission to the newly created user.