

Name: Dikshya Kafle

Student Number: 2018380039

Experiment 2: Manipulate the Data in Table

### **Goal**

1. Master all kinds of data operation about basic table in GUI.
2. Familiar with SQL statements for data insertion, modification and deletion of basic tables.
3. Master the SQL statement of data query.
4. Master the basic knowledge of SQL query performance analysis.
5. Understand TPC-H benchmark database.

### **Content**

1. Use SQL statement to insert all the tuples into the database SPJ\_MNG and university which have been list in the previous experiment.
2. Modificaion the data of tables with SQL statement.
  - (1) Modify one tuple in the table of student
  - (2) Delete one tuple from table of student.
3. In the database of SPJ\_MNG, use SQL statement to do the following update operations:
  - (1) Change the color of all red parts to blue.
  - (2) Part P6 supplied by S5 for J4 is replaced by S3, please make necessary modification.
  - (3) Delete S2 record from supplier table and delete corresponding record from supply table.
  - (4) Please insert (S2, J6, p4200) into the supply table SPJ.Finish the following queries about the database university with SQL statement.
4. Use three different ways (SQL statement) to find the student ID and name of all students who take "Database System Concept", and then analyze and compare the performance of each query process.
5. For university database, complete the following data query with SQL statement
  - (1) Query the total score of credits obtained by each student , and output the student ID, name and credit obtained in the order from high to low.
  - (2) Query the name of the student: the student has taken all courses and one of the courses has a grade of better than B .
6. Use at least three different SQL statements to query the university database: query the student ID and name of the course named "database", and then design the experiment by ourselves, compare and analyze the efficiency of the three kinds of query with data, and analyze the reasons.

**Use SQL statement to insert all the tuples into the database SPJ MNG and university which have been list in the previous experiment.**

### **S table:**

MySQL Workbench

mysql@127.0.0.1:3306 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

spj\_mng

Tables

j

p

s

spj

Views

Stored Procedures

Functions

sys

Tables

Administration Schemas

Information

Table: s

Columns:

SNO varchar(45) PK

SNAME varchar(45)

STATUS varchar(45)

CITY varchar(45)

Query 1 Administration - Data Export new\_schema - Schema spj\_mng.s s - Table

Limit to 1000 rows

```

1 • SELECT * FROM spj_mng.s;
2 • INSERT INTO `s` (SNO,SNAME,STATUS,CITY) VALUES ('SN1','精益','20','天津');
3 • INSERT INTO `s` (SNO,SNAME,STATUS,CITY) VALUES ('SN2','盛锡','10','北京');
4 • INSERT INTO `s` (SNO,SNAME,STATUS,CITY) VALUES ('SN3','东方红','30','北京');
5 • INSERT INTO `s` (SNO,SNAME,STATUS,CITY) VALUES ('SN4','丰泰盛','20','天津');
6 • INSERT INTO `s` (SNO,SNAME,STATUS,CITY) VALUES ('SN5','为民','30','上海');
7

```

Result Grid

SNO	SNAME	STATUS	CITY
SN1	精益	20	天津
SN2	精益	10	北京
SN3	精益	30	北京
SN4	精益	20	天津
SN5	精益	30	上海
NULL	NULL	NULL	NULL

s 3 x

Apply

### **Spj table:**

MySQL Workbench

mysql@127.0.0.1:3306 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

Triggers

s

spj

Views

Stored Procedures

Functions

sys

Tables

Views

Stored Procedures

Functions

Administration Schemas

Information

Table: spj

Columns:

SNO varchar(100)

PNO varchar(100)

JNO varchar(100)

QTY int

spj - Table

Limit to 1000 rows

```

15 • INSERT INTO `spj` (SNO,PNO,JNO,QTY) VALUES ('S4','P6','J3','100');
16 • INSERT INTO `spj` (SNO,PNO,JNO,QTY) VALUES ('S4','P6','J4','300');
17 • INSERT INTO `spj` (SNO,PNO,JNO,QTY) VALUES ('S5','P2','J4','100');
18 • INSERT INTO `spj` (SNO,PNO,JNO,QTY) VALUES ('S5','P3','J1','200');
19 • INSERT INTO `spj` (SNO,PNO,JNO,QTY) VALUES ('S5','P6','J2','200');
20 • INSERT INTO `spj` (SNO,PNO,JNO,QTY) VALUES ('S5','P6','J4','500');

```

Result Grid

SNO	PNO	JNO	QTY
S1	P1	J1	200
S1	P1	J3	100
S1	P1	J4	700
S1	P2	J2	100
S2	P3	J1	400
S2	P3	J2	200
S2	P3	J4	500
S2	P3	J5	400

spj 4 x

Read Only

Output

Action Output

#	Time	Action	Message
83	08:17:57	INSERT INTO `spj` (SNO,PNO,JNO,QTY) VALUES ('S4','P6','J3','100')	1 row(s) affected
84	08:17:57	INSERT INTO `spj` (SNO,PNO,JNO,QTY) VALUES ('S4','P6','J4','300')	1 row(s) affected
85	08:17:57	INSERT INTO `spj` (SNO,PNO,JNO,QTY) VALUES ('S5','P2','J4','100')	1 row(s) affected
86	08:17:57	INSERT INTO `spj` (SNO,PNO,JNO,QTY) VALUES ('S5','P3','J1','200')	1 row(s) affected
87	08:17:57	INSERT INTO `spj` (SNO,PNO,JNO,QTY) VALUES ('S5','P6','J2','200')	1 row(s) affected
88	08:17:57	INSERT INTO `spj` (SNO,PNO,JNO,QTY) VALUES ('S5','P6','J4','500')	1 row(s) affected

Object Info Session

Active Go to Se

## J table:

MySQL Workbench

Mysql@127.0.0.1:3306 x

File Edit View Query Database Server Tools Scripting Help

Navigator: table new\_table - Table j s j - Table j - Table j j - Table j - Table j j

SCHEMAS

Filter objects

spj\_mng

Tables

j

Columns

Indexes

Foreign Keys

Triggers

Administration Schemas

Information

Table: j

Columns:

JNO varchar(45) PK

JNAME varchar(45)

CITY varchar(45)

```
1 SELECT * FROM spj_mng.j;
2 INSERT INTO `j` (JNO,JNAME,CITY) VALUES ('J1','Rising','Guangzhou');
3 INSERT INTO `j` (JNO,JNAME,CITY) VALUES ('J2','Powercity','Beijing');
4 INSERT INTO `j` (JNO,JNAME,CITY) VALUES ('J3','Bechange','Hangzhou');
5 INSERT INTO `j` (JNO,JNAME,CITY) VALUES ('J4','Wetogether','Shanghai');
6 INSERT INTO `j` (JNO,JNAME,CITY) VALUES ('J5','Yesforchange','Shenzhen');
7 INSERT INTO `j` (JNO,JNAME,CITY) VALUES ('J6','gotthis','Xian');
```

JNO	JNAME	CITY
J1	Rising	Guangzhou
J2	Powercity	Beijing
J3	Bechange	Hangzhou
J4	Wetogether	Shanghai
J5	Yesforchange	Shenzhen
J6	gotthis	Xian
NULL	NULL	NULL

Result Grid

Filter Rows:

Output

Action Output

#	Time	Action	Message
38	22:03:41	INSERT INTO `j` (JNO,JNAME,CITY) VALUES ('J4','Wetogether','Shanghai')	1 row(s) affected
39	22:03:42	INSERT INTO `j` (JNO,JNAME,CITY) VALUES ('J5','Yesforchange','Shenzhen')	1 row(s) affected
40	22:03:42	INSERT INTO `j` (JNO,JNAME,CITY) VALUES ('J6','gotthis','Xian')	1 row(s) affected
41	22:05:39	INSERT INTO `j` (JNO,JNAME,CITY) VALUES ('J6','gotthis','Xian')	Error Code: 1062. Duplicate entry 'J6' for key 'j.PRIMARY'
42	22:05:46	SELECT * FROM spj_mng.j LIMIT 0, 1000	6 row(s) returned
43	22:05:46	INSERT INTO `j` (JNO,JNAME,CITY) VALUES ('J1','Rising','Guangzhou')	Error Code: 1062. Duplicate entry 'J1' for key 'j.PRIMARY'

## P table:

Result Grid

Filter Rows:

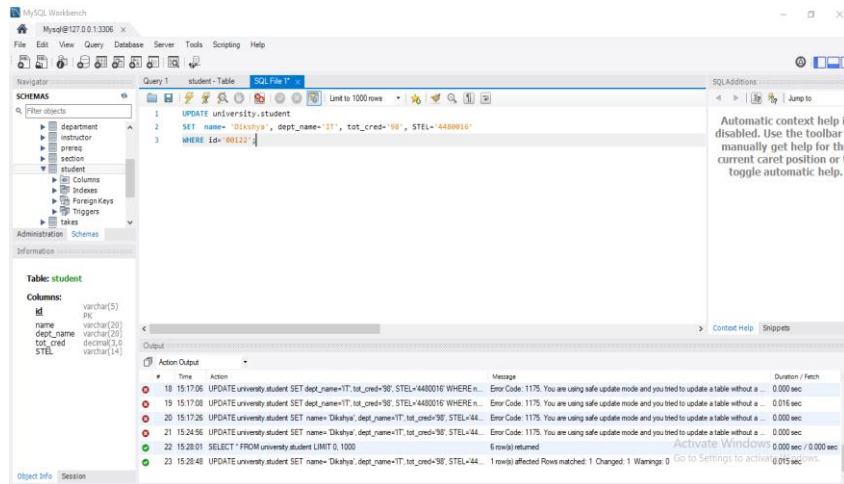
Output

Action Output

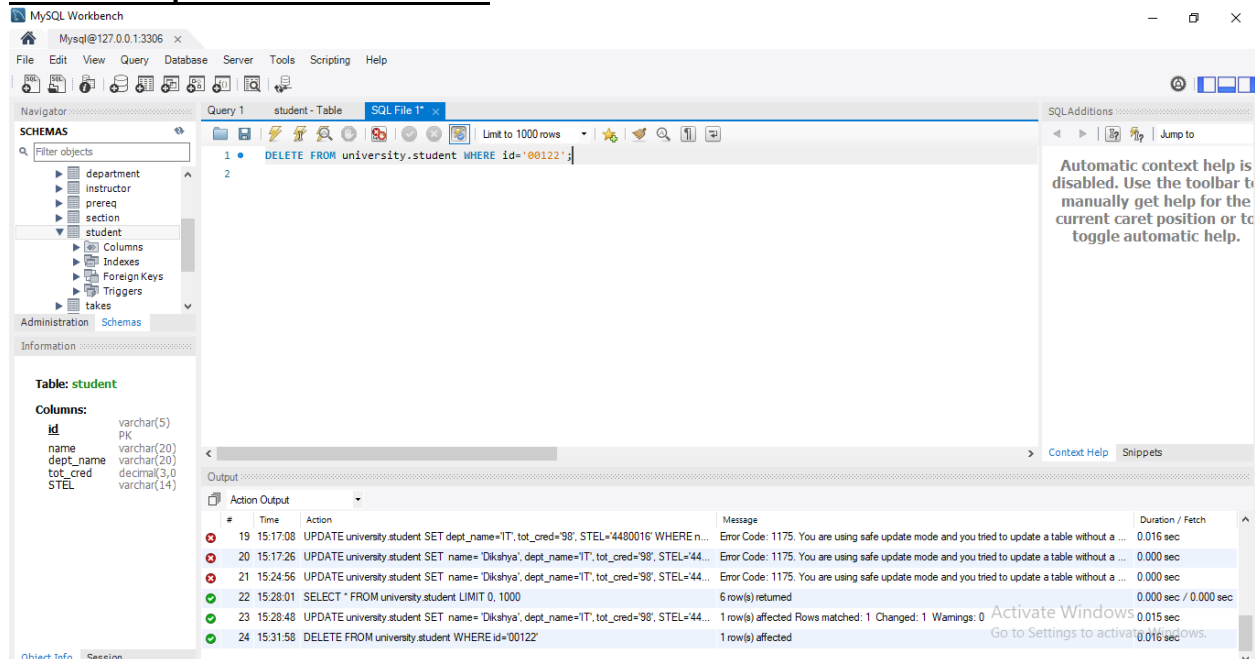
#	Time	Action	Message
1	08:30:14	SELECT * FROM spj_mng.p LIMIT 0, 1000	6 row(s) returned

## Modificaion the data of tables with SQL statement:

### Modify one tuple in the table of student:



## Delete one tuple from table of student:



**In the database of SPJ\_MNG, use SQL statement to do the following update operations:**

**Change the color of all red parts to blue:**

MySQL Workbench

unconnected x Mysql@127.0.0.1:3306 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

spj\_mng

Tables

j

p

s

spj

Views

Stored Procedures

Functions

sys

Administration Schemas

Information

Table: p

Columns:

PNO varchar(50) PK

PNAME varchar(45)

COLOR varchar(45)

WEIGHT varchar(45)

Query 1 x p - Table

Limit to 1000 rows

1 • UPDATE spj\_mng.p SET COLOR='blue' WHERE COLOR='red'

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	15:54:14	UPDATE spj_mng.spj SET SNO='S3' WHERE SNO='S5' AND PNO='P6' AND JNO='J4'	2 row(s) affected Rows matched: 2 Changed: 2 Warnings: 0	0.015 sec
2	15:56:58	UPDATE spj_mng.j SET COLOR='blue' WHERE COLOR='red'	Error Code: 1054. Unknown column 'COLOR' in 'where clause'	0.000 sec
3	15:57:36	UPDATE spj_mng.j SET COLOR='red' WHERE COLOR='blue'	Error Code: 1054. Unknown column 'COLOR' in 'where clause'	0.000 sec
4	15:58:10	UPDATE spj_mng.p SET COLOR='red' WHERE COLOR='blue'	5 row(s) affected Rows matched: 5 Changed: 5 Warnings: 0	0.015 sec
5	15:58:27	UPDATE spj_mng.p SET COLOR='blue' WHERE COLOR='red'	5 row(s) affected Rows matched: 5 Changed: 5 Warnings: 0	0.016 sec

Activate Windows  
Go to Settings to activate Windows.

MySQL Workbench

Mysql@127.0.0.1:3306 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

spj\_mng

Tables

j

p

s

spj

Columns

Indexes

Foreign Keys

Triggers

Administration Schemas

Information

Table: p

Columns:

PNO varchar(50) PK

PNAME varchar(45)

COLOR varchar(45)

WEIGHT varchar(45)

Query 1 x p - Table p

Limit to 1000 rows

1 • SELECT \* FROM spj\_mng.p;

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid

PNO	PNAME	COLOR	WEIGHT
P1	螺母	blue	12
P2	螺栓	green	17
P3	螺栓刀	blue	14
P4	螺栓刀	blue	14
P5	凸轮	blue	40
P6	齿轮	blue	30

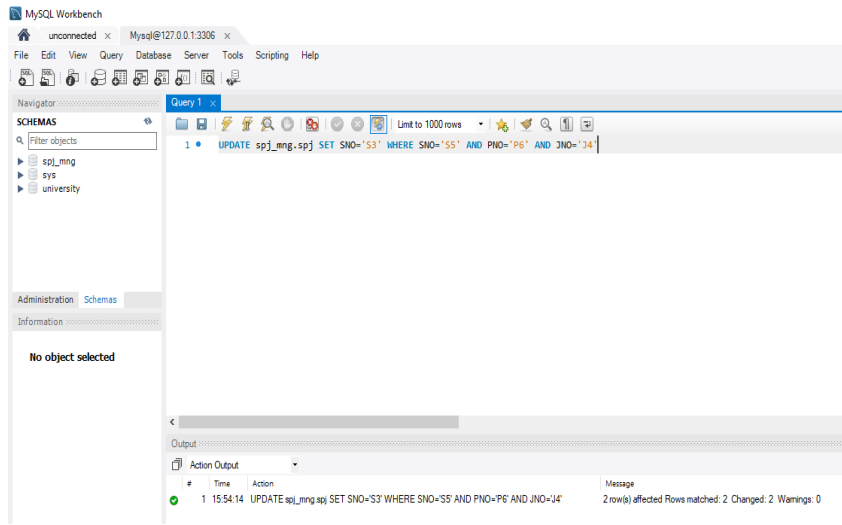
Output

Action Output

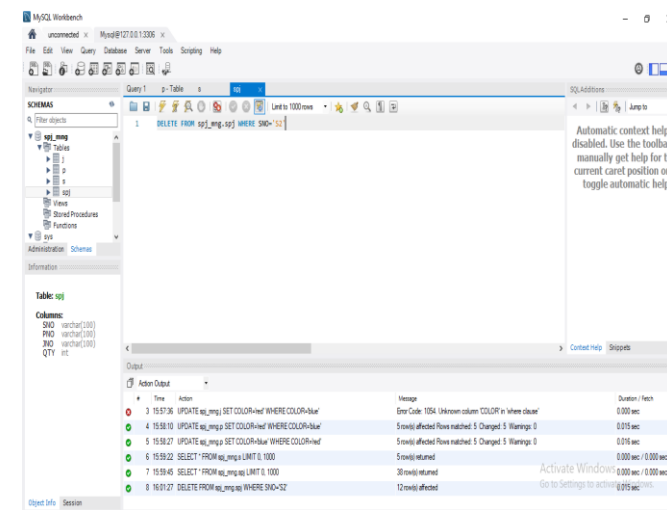
#	Time	Action	Message	Duration / Fetch
1	15:34:26	SELECT * FROM spj_mng.p LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
2	15:35:53	UPDATE spj_mng.p SET COLOR='blue' WHERE COLOR='red'	Error Code: 1175. You are using safe update mode and you tried to update a table without a W...	0.000 sec
3	15:36:13	UPDATE spj_mng.p SET COLOR='blue' WHERE COLOR='red'	Error Code: 1175. You are using safe update mode and you tried to update a table without a W...	0.000 sec
4	15:36:35	SELECT * FROM spj_mng.p LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
5	15:38:32	UPDATE spj_mng.p SET COLOR='blue' WHERE PNO='P1'OR PNO='P4' OR PNO='P6'	3 row(s) affected Rows matched: 3 Changed: 3 Warnings: 0	0.000 sec
6	15:38:41	SELECT * FROM spj_mng.p LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec

Activate Windows  
Go to Settings to activate Windows.

**Part P6 supplied by S5 for J4 is replaced by S3, please make necessary modification.**



**Delete S2 record from supplier table and delete corresponding record from supply table.**



**Please insert (S2, J6, p4200) into the supply table SPJ.Finish the following queries about the database university with SQL statement**

MySQL Workbench

unconnected x Mysql@127.0.0.1:3306 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMAS

Filter objects

spj\_mng

Tables

Views

Stored Procedures

Functions

sys

Schemas

Information

Table: spj

Columns:

SNO varchar(100)

PNO varchar(100)

JNO varchar(100)

QTY int

Query 1

1 INSERT INTO spj\_mng.spj (SNO, PNO, JNO, QTY) VALUES ('S2', 'P4', 'J6', 200)

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Output

Action Output

#	Time	Action	Message	Duration / Fetch
5	15:58:27	UPDATE spj_mng.p SET COLOR=blue WHERE COLOR=red	5 row(s) affected Rows matched: 5 Changed: 5 Warnings: 0	0.016 sec
6	15:59:22	SELECT * FROM spj_mng.s LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
7	15:59:45	SELECT * FROM spj_mng.sq LIMIT 0, 1000	38 row(s) returned	0.000 sec / 0.000 sec
8	16:01:27	DELETE FROM spj_mng.sq WHERE SNO=S2	12 row(s) affected	0.015 sec
9	16:02:07	SELECT * FROM spj_mng.sq LIMIT 0, 1000	26 row(s) returned	0.000 sec / 0.000 sec
10	16:05:17	INSERT INTO spj_mng.spj (SNO, PNO, JNO, QTY) VALUES (S2, P4, J6, 200)	1 row(s) affected	0.015 sec

Result Grid

Filter Rows:

	SNO	PNO	JNO	QTY
	S1	P1	J1	200
	S1	P1	J3	100
	S1	P1	J4	700
	S1	P2	J2	100
	S2	P4	J6	200
	S3	P1	J1	200
	S3	P3	J1	200

spj 1 x

**Use three different ways (SQL statement) to find the student ID and name of all students who take “Database System Concept”, and then analyze and compare the performance of each query process.**

MySQL Workbench

Mysql@127.0.0.1:3306 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMAS

Filter objects

student

Columns

Indexes

Foreign Keys

Triggers

takes

teaches

time\_slot

Views

Stored Procedures

Administration

Schemas

Information

Table: student

Query 1

student takes takes - Table course university.takes takes takes

1 SELECT \* FROM University.course

2 natural join university.takes where title =

3 (Select title from university.course where course\_id ='CS-347')

Result Grid

Filter Rows:

Export: Wrap Cell Contents

course_id	title	dept_name	credits	id	sec_id	semester	year	grade
CS-347	Database Management System	Comp.Sci	3	12345	1	Fall	2009	A

The screenshot shows the MySQL Workbench interface. The 'SCHEMAS' pane on the left displays the database structure, including tables like 'student', 'takes', 'teaches', and 'time\_slot'. The 'Query 1' editor contains the following SQL statement:

```
SELECT * FROM university.student as ss
NATURAL JOIN university.takes where course_id = 'CS-347'
```

The 'Result Grid' shows the output of the query, which is a single row for student ID 12345 (Shankar) who has taken course CS-347 with a grade of A.

id	name	dept_name	tot_cred	STEL	course_id	sec_id	semester	year	grade
12345	Shankar	Com. Sci.	32	STEL	CS-347	1	Fall	2009	A

**For university database, complete the following data query with SQL statement**

**Query the total score of credits obtained by each student , and output the student ID, name and credit obtained in the order from high to low.**

The screenshot shows the MySQL Workbench interface. The 'Query 1' editor contains the following SQL statement:

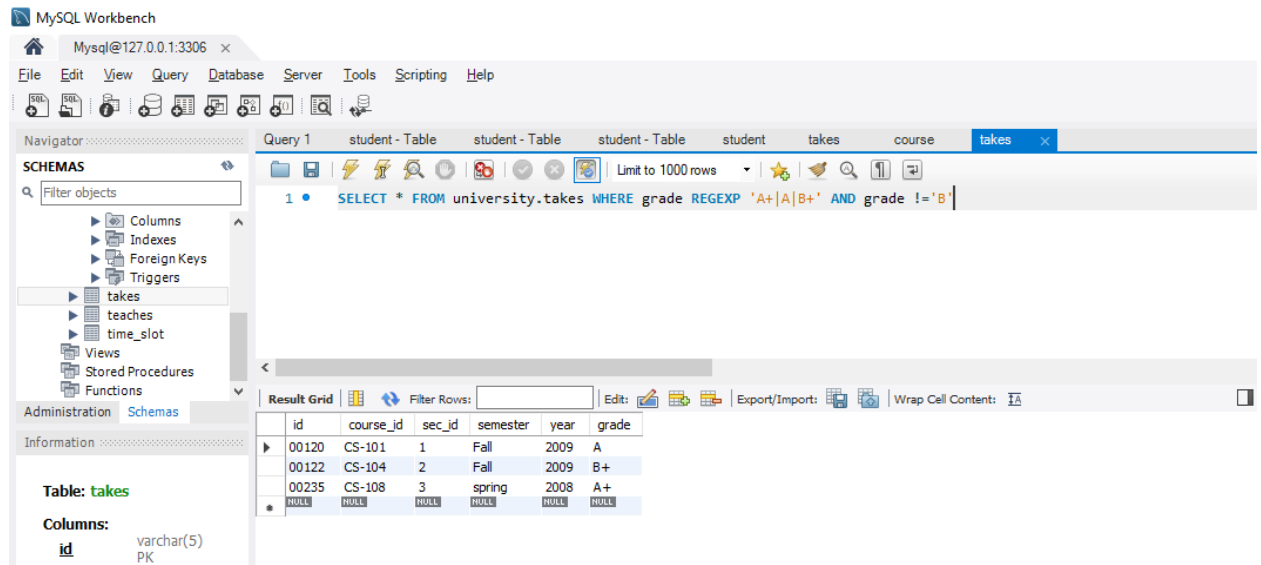
```
SELECT id,name,tot_cred FROM university.student ORDER BY tot_cred DESC
```

The 'Result Grid' shows the output of the query, which is a list of students ordered by their total credits in descending order.

id	name	tot_cred
00125	Zhang	110
00128	Zhang	102
19991	Zhang	80
00138	Zhang	46
12345	Shankar	32
00039	00039	00039

**Query the name of the student: the student has taken all courses and one of the courses has a grade of better than B**





### **Problem:**

Had some simple syntax error. Because of the complexity of the question had hard time figuring it out.

### **Solution:**

To solve these problems I looked for information in internet. In order to understand some questions and procedure I also asked the teacher to help me understand them. And provided instructions helped to solve some of my errors during the experiment.

### **Summary:**

From this experiment I have learned SQL statement to create database and table. I have learned how to update and delete methods of database and table. Have become familiar with SQL statements of data insertion, modification and deletion of basic tables. Learned all kinds of data operation about basic table in GUI. This experiment allowed us to understand SQL statement of data query. Therefore, I have been able to have basic knowledge of SQL query performances and analysis.