Xinyu Liu

B00783546

Xn427668@dal.ca

1) Design a relational database schema for a database application of your choice. (45% - break down is below)

a) State and describe your requirements, i.e. business rules for the application you choose. You may explore a similar existing system to come up with a list of requirements for your database and its front-end application. You may also use one of the databases you designed in the previous assignments as your starting point. (5%)

Requirement

a. A student has last name (Lname), first name (Fname), social insurance number (Ssn), student number (Bnum), major code (Major_code) and minor code (Minor_code). Both Ssn and student number have unique values for each student.

b. Each course has a course number (Cnum) and name (Cname), and a course section (Csec) as well as offering department (Cdept) . The course number is unique for each course.

c. Each department has its name (Dname) and code (Dcode) and department location (Dlocation). Both name and code are unique for each department.

d. Each Grade refers to a student (Ssn), a particular course (Cnum) and according grade.

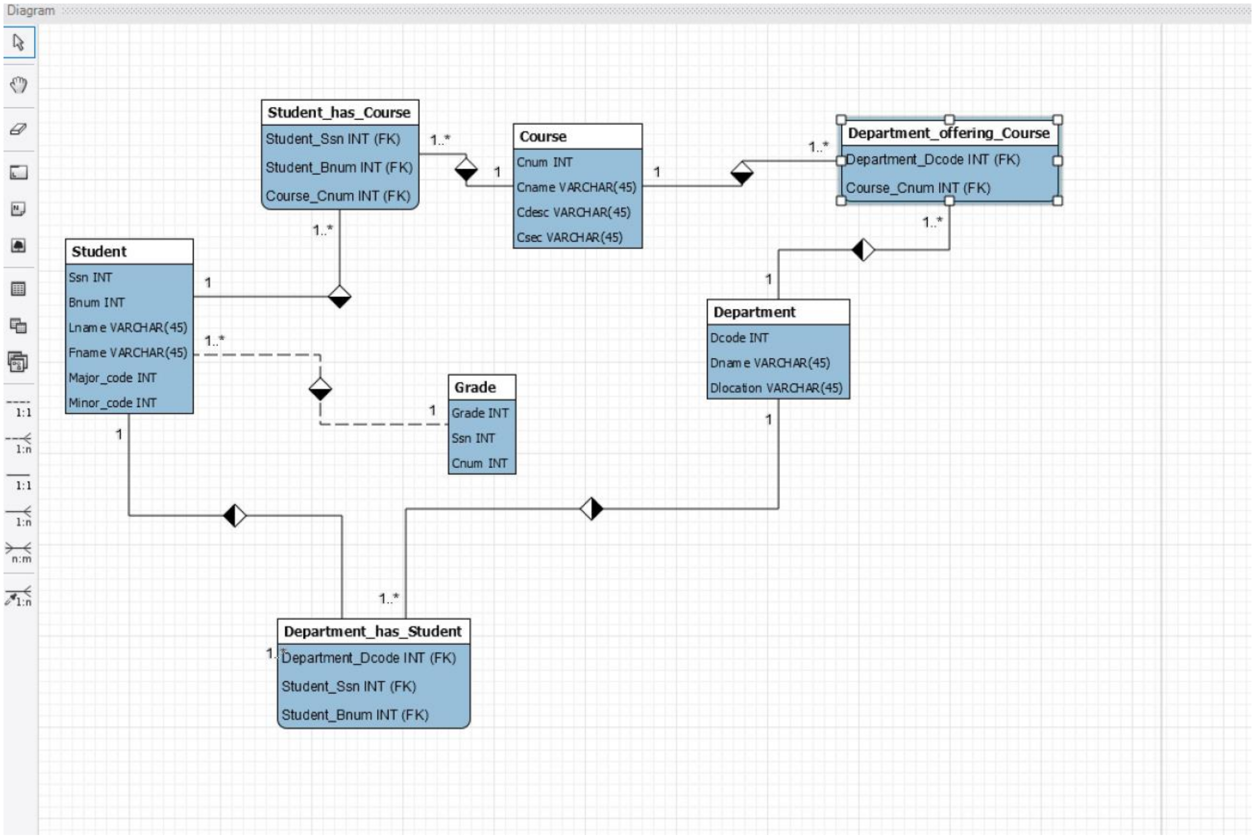
Rules: Students could choose more than one course and a course is taken by more than one student.

Department has many students and a student may in two departments due to minor and double majors.

A student has many grades but one grade with specific Ssn could only refer to one student.

b)

Design and draw your Entity – Relationship Diagrams using MySql Workbench / Microsoft Word / yEd / or any other drawing tool of your choice. (5%)



c) Design and declare your relational data model using SQL (10% - break down is below)

(1) Minimum 3 relations (tables) (2%)

(2) Minimum 3 attributes per table (2%)

(3) Minimum 10 records per table (6%)

There are four tables: Student, Department, Grade and Course

Student: {Ssn}->{Bnum,Fname,Lname,Major_code,Minor_code}

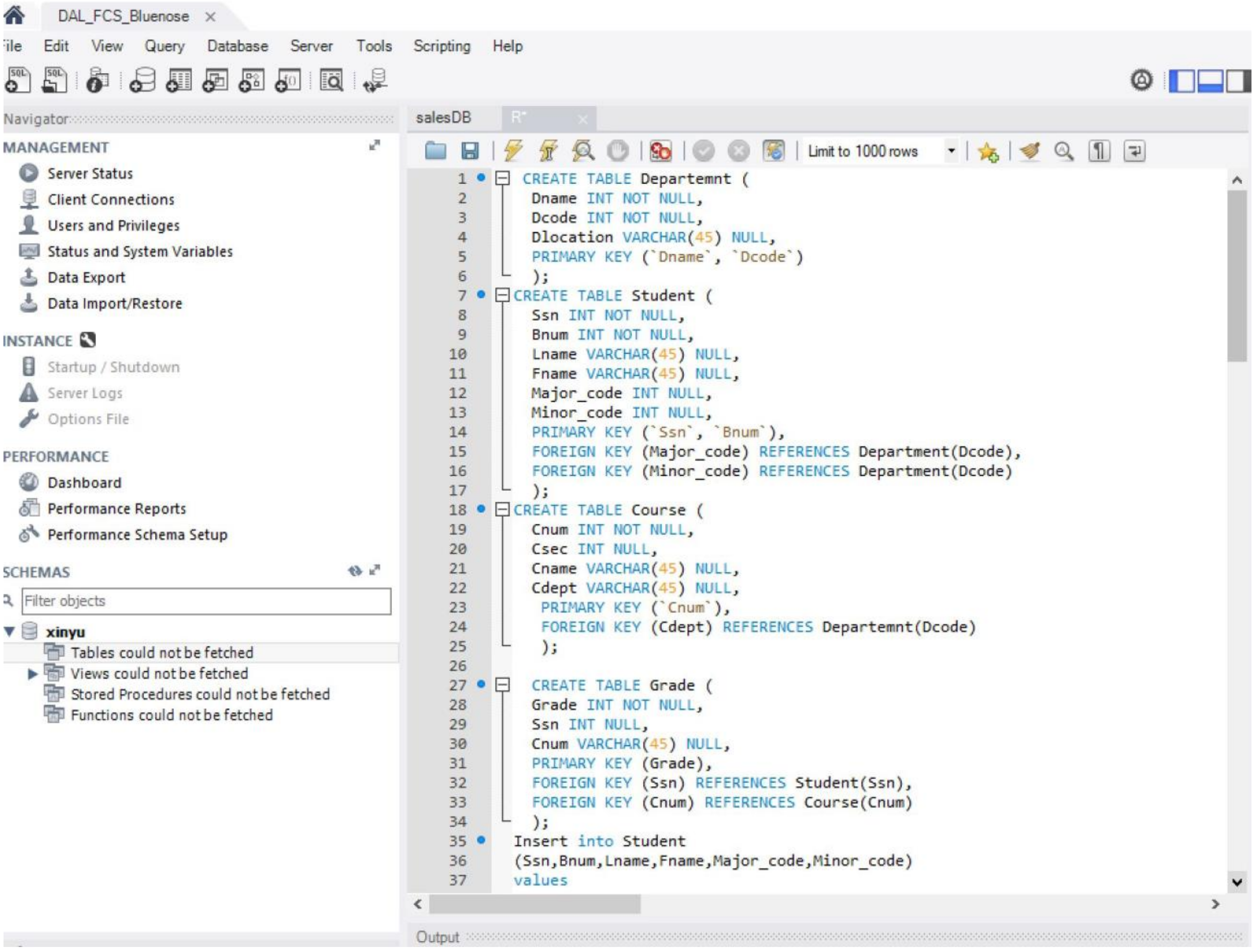
{Bnum}->{Ssn,Fname,Lname,Major_code,Minor_code}

Course: {Cnum}->{Csec,Cname,Cdept}

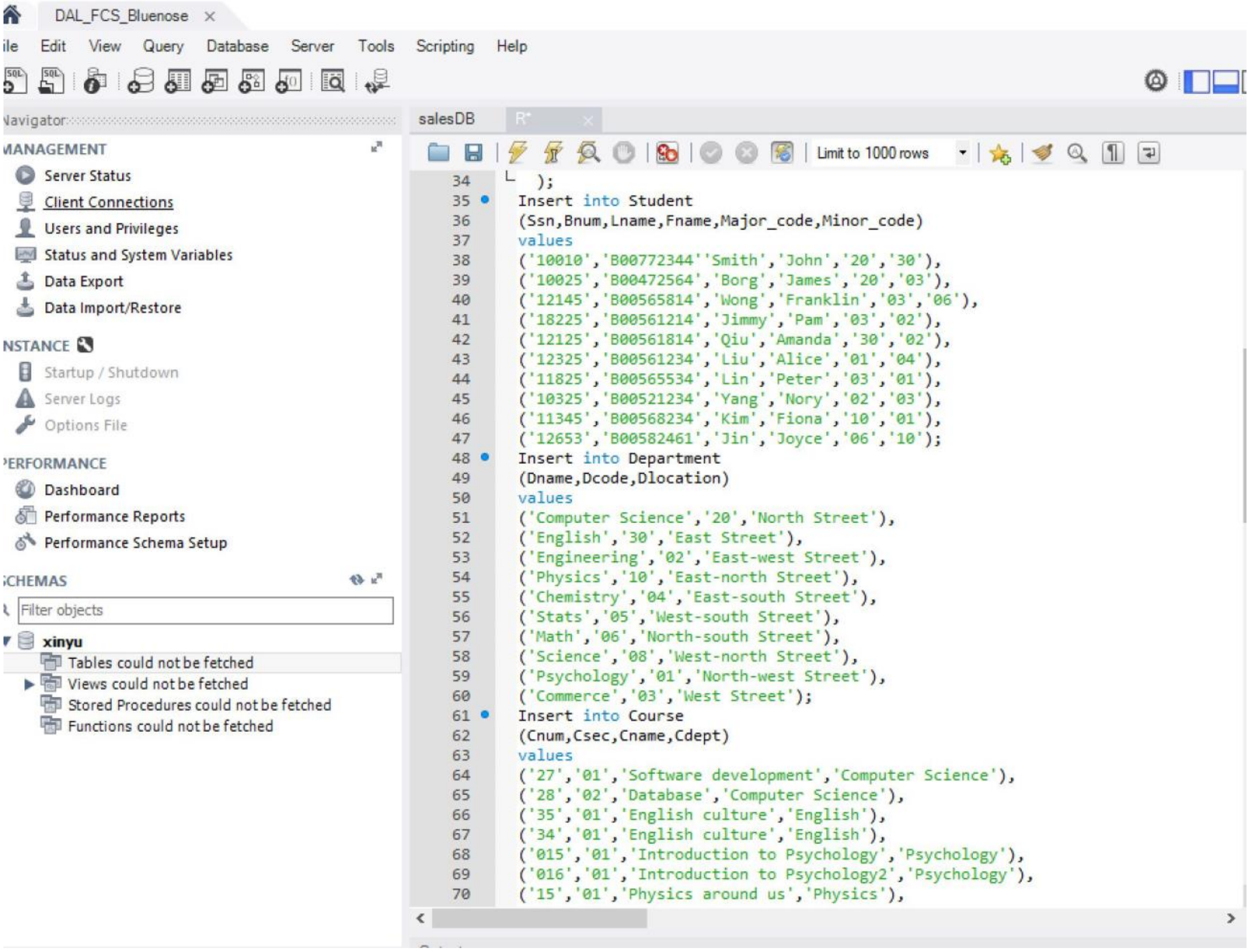
Department: {Dcode}->{Dname,Dlocation}

{Dname}->{Dcode,Dlocation}

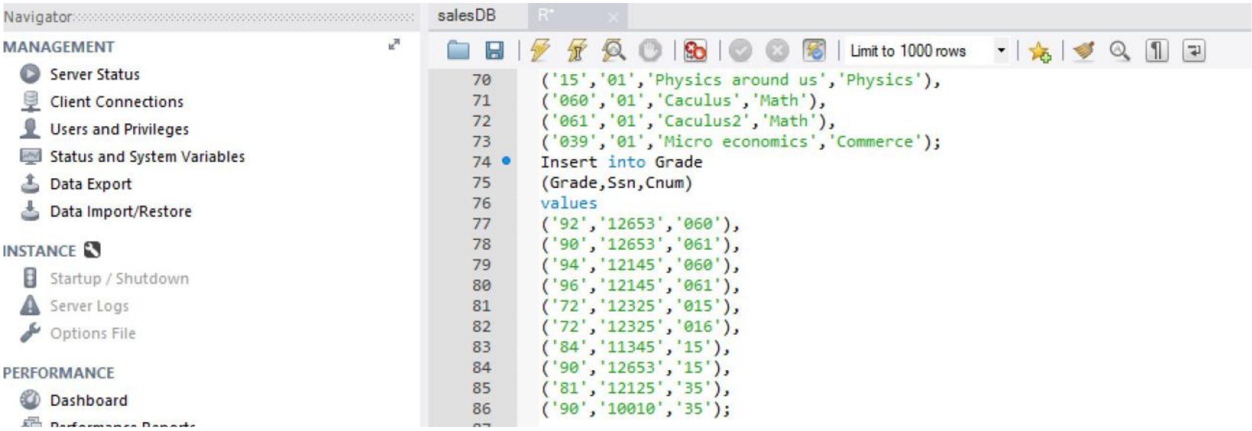
Grade: {Ssn}->{Grade,Cnum}



```
 CREATE TABLE Departemnt (
  Dname INT NOT NULL,
  Dcode INT NOT NULL,
  Dlocation VARCHAR(45) NULL,
  PRIMARY KEY (`Dname`, `Dcode`)
  );
CREATE TABLE Student (
  Ssn INT NOT NULL,
  Bnum INT NOT NULL,
  Lname VARCHAR(45) NULL,
  Fname VARCHAR(45) NULL,
  Major_code INT NULL,
  Minor_code INT NULL,
  PRIMARY KEY (`Ssn`, `Bnum`),
  FOREIGN KEY (Major_code) REFERENCES Department(Dcode),
  FOREIGN KEY (Minor_code) REFERENCES Department(Dcode)
  );
CREATE TABLE Course (
  Cnum INT NOT NULL,
  Csec INT NULL,
  Cname VARCHAR(45) NULL,
  Cdept VARCHAR(45) NULL,
   PRIMARY KEY (`Cnum`),
   FOREIGN KEY (Cdept) REFERENCES Departemnt(Dcode)
   );

  CREATE TABLE Grade (
  Grade INT NOT NULL,
  Ssn INT NULL,
  Cnum VARCHAR(45) NULL,
  PRIMARY KEY (Grade),
  FOREIGN KEY (Ssn) REFERENCES Student(Ssn),
  FOREIGN KEY (Cnum) REFERENCES Course(Cnum)
  );
Insert into Student
(Ssn,Bnum,Lname,Fname,Major_code,Minor_code)
values
```



```sql
 );
Insert into Student
(Ssn,Bnum,Lname,Fname,Major_code,Minor_code)
values
('10010','B00772344''Smith','John','20','30'),
('10025','B00472564','Borg','James','20','03'),
('12145','B00565814','Wong','Franklin','03','06'),
('18225','B00561214','Jimmy','Pam','03','02'),
('12125','B00561814','Qiu','Amanda','30','02'),
('12325','B00561234','Liu','Alice','01','04'),
('11825','B00565534','Lin','Peter','03','01'),
('10325','B00521234','Yang','Nory','02','03'),
('11345','B00568234','Kim','Fiona','10','01'),
('12653','B00582461','Jin','Joyce','06','10');
Insert into Department
(Dname,Dcode,Dlocation)
values
('Computer Science','20','North Street'),
('English','30','East Street'),
('Engineering','02','East-west Street'),
('Physics','10','East-north Street'),
('Chemistry','04','East-south Street'),
('Stats','05','West-south Street'),
('Math','06','North-south Street'),
('Science','08','West-north Street'),
('Psychology','01','North-west Street'),
('Commerce','03','West Street');
Insert into Course
(Cnum,Csec,Cname,Cdept)
values
('27','01','Software development','Computer Science'),
('28','02','Database','Computer Science'),
('35','01','English culture','English'),
('34','01','English culture','English'),
('015','01','Introduction to Psychology','Psychology'),
('016','01','Introduction to Psychology2','Psychology'),
('15','01','Physics around us','Physics'),
```



```sql
('15','01','Physics around us','Physics'),
('060','01','Caculus','Math'),
('061','01','Caculus2','Math'),
('039','01','Micro economics','Commerce');
Insert into Grade
(Grade,Ssn,Cnum)
values
('92','12653','060'),
('90','12653','061'),
('94','12145','060'),
('96','12145','061'),
('72','12325','015'),
('72','12325','016'),
('84','11345','15'),
('90','12653','15'),
('81','12125','35'),
('90','10010','35');
```

d) Normalize your database design to the level of 3NF: Use either a top-down or a bottom-up approach (10%)

Functional dependencies:

FD1: {Ssn} -> {Bnum, Lname,Fname, Major_code, Minor_code }

FD2: {Bnum} -> {Ssn, Lname,Fname, Major_code, Minor_code }

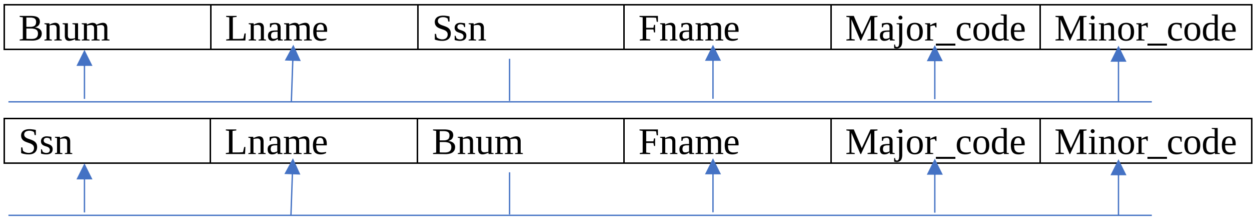
FD3: {Dname} -> {Dcode, Dlocation}

FD4: {Dcode} -> {Dname, Dlocation}

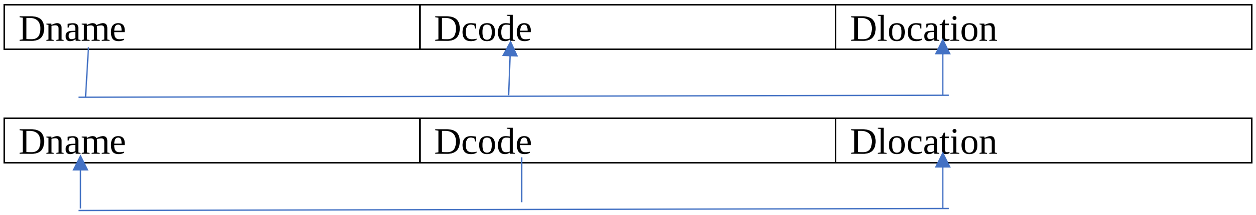
FD5: {Cnum} -> {Cname, Csec,Cdept}

FD6: {Ssn}->{Cnum, Grade}

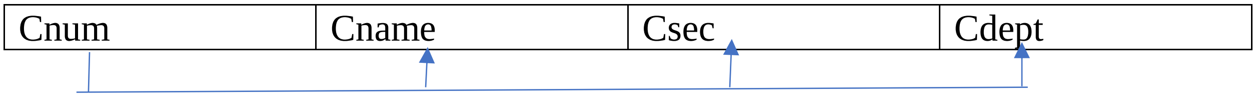
Student:



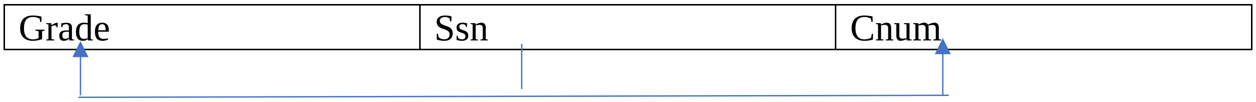
Departemnt:



Course:

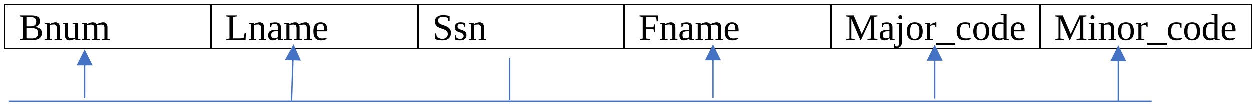


Grade:



According to 1NF, the only attribute values permitted are single atomic values. Consider R is in 1NF. Then find out partial dependencies because it disallows partial dependencies in 2NF. Ssn is working as a primary key to identify attributes in Student relation and Bnum is working as a primary key to identify attributes in Student relation. In case of avoiding partial dependencies and reducing redundancy, we assume that Bnum over Ssn as primary key of Student.

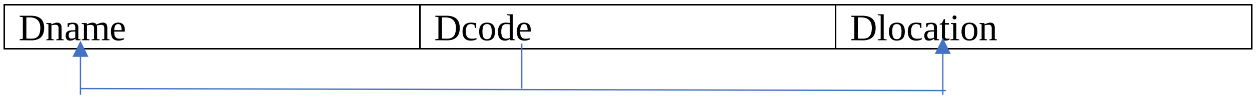
Hence,



Bnum	Lname	Ssn	Fname	Major_code	Minor_code

Similarly, for avoiding partial dependencies and redundancy, assume Decode over Dname as primary key of Department.

Hence,



Dname	Dcode	Dlocation

Next, we find out whether there exist any transitive dependencies or not. There is no transitive dependency in the above relations.

Hence, the above relations are in third normal form.

So, we have the following relations and for avoiding confusion, we change the attribute names of Grade that Ssn as GSsn, Cnum as GCnum and Grade.GSsn=Student.Ssn, Grade.GCnum=Course.Cnum.

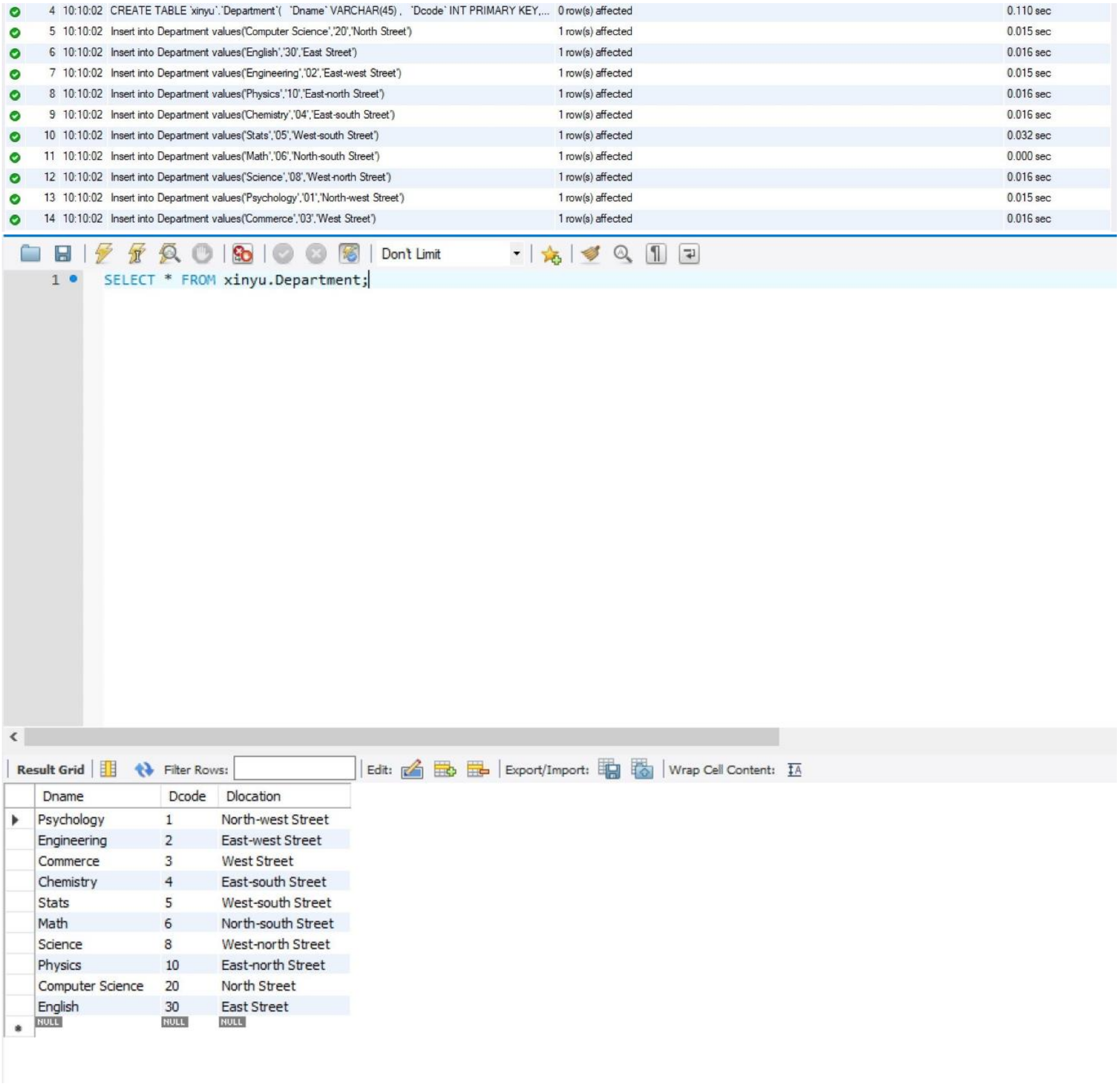
Student: {Ssn}->{Bnum,Fname,Lname,Major_code,Minor_code}

Course: {Cnum}->{Csec,Cname,Cdept}

Department: {Dcode}->{Dname,Dlocation}

Grade: {GSsn}->{Grade,GCnum}

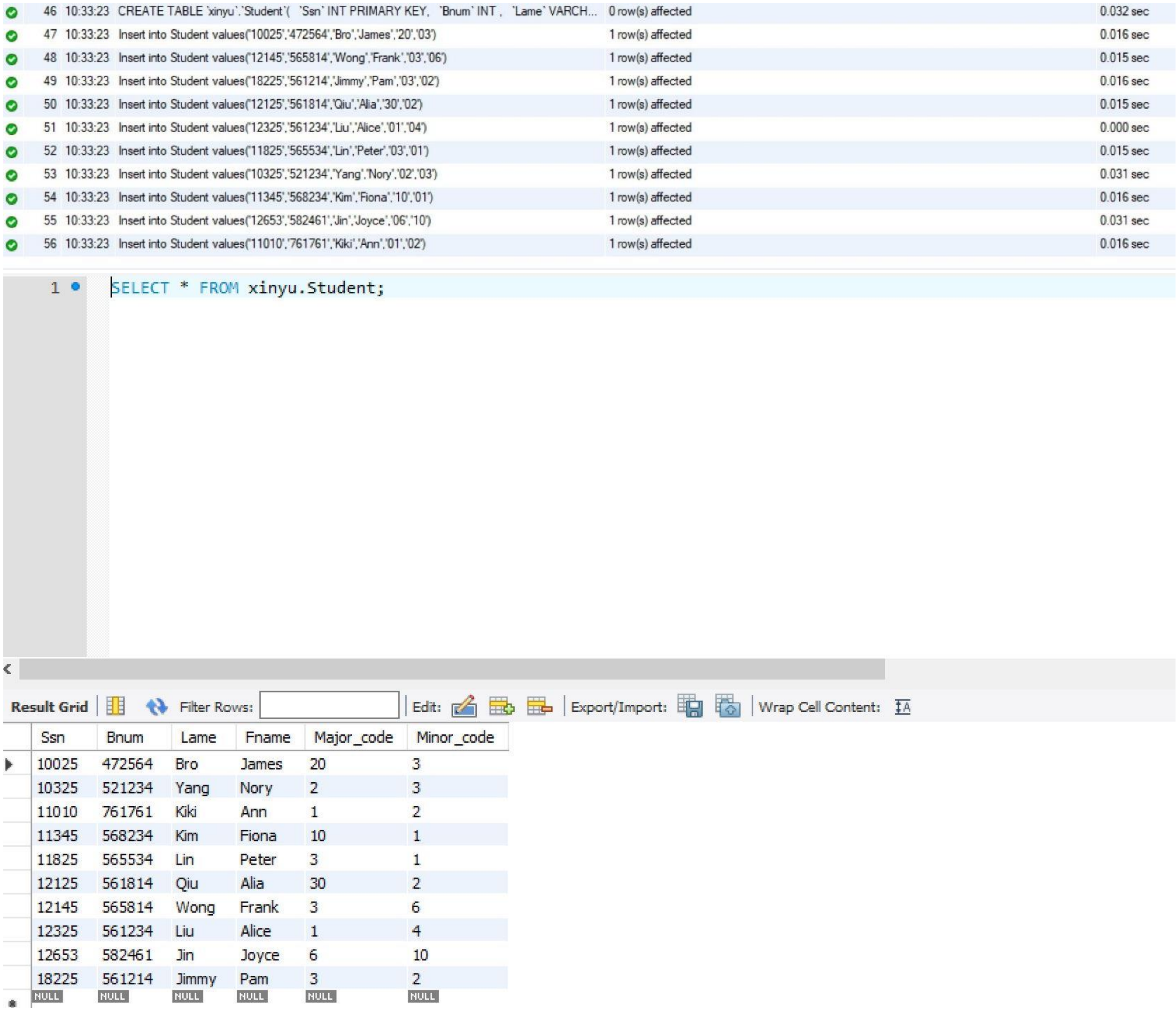
And we have following records in tables:



```
 4  10:10:02  CREATE TABLE `xinyu`.`Department`(  `Dname` VARCHAR(45) ,  `Dcode` INT PRIMARY KEY,...  0 row(s) affected  0.110 sec
 5  10:10:02  Insert into Department values('Computer Science','20','North Street')   1 row(s) affected  0.015 sec
 6  10:10:02  Insert into Department values('English','30','East Street')             1 row(s) affected  0.016 sec
 7  10:10:02  Insert into Department values('Engineering','02','East-west Street')    1 row(s) affected  0.015 sec
 8  10:10:02  Insert into Department values('Physics','10','East-north Street')       1 row(s) affected  0.016 sec
 9  10:10:02  Insert into Department values('Chemistry','04','East-south Street')     1 row(s) affected  0.016 sec
10  10:10:02  Insert into Department values('Stats','05','West-south Street')         1 row(s) affected  0.032 sec
11  10:10:02  Insert into Department values('Math','06','North-south Street')         1 row(s) affected  0.000 sec
12  10:10:02  Insert into Department values('Science','08','West-north Street')       1 row(s) affected  0.016 sec
13  10:10:02  Insert into Department values('Psychology','01','North-west Street')    1 row(s) affected  0.015 sec
14  10:10:02  Insert into Department values('Commerce','03','West Street')            1 row(s) affected  0.016 sec
```

```
SELECT * FROM xinyu.Department;
```

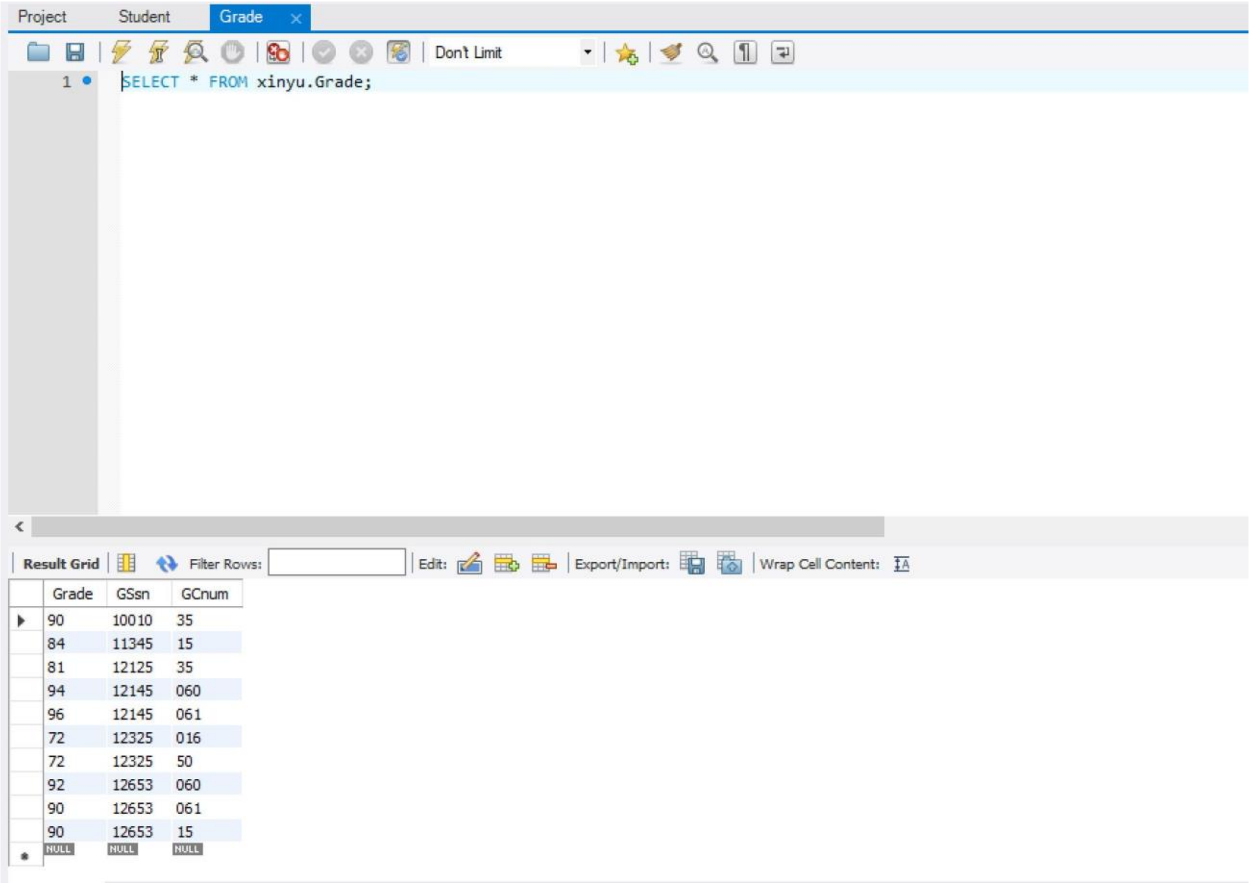
Dname	Dcode	Dlocation
Psychology	1	North-west Street
Engineering	2	East-west Street
Commerce	3	West Street
Chemistry	4	East-south Street
Stats	5	West-south Street
Math	6	North-south Street
Science	8	West-north Street
Physics	10	East-north Street
Computer Science	20	North Street
English	30	East Street
NULL	NULL	NULL



```
46  10:33:23  CREATE TABLE `xinyu`.`Student`(  `Ssn` INT PRIMARY KEY,  `Bnum` INT ,  `Lame` VARCH...  0 row(s) affected  0.032 sec
47  10:33:23  Insert into Student values('10025','472564','Bro','James','20','03')  1 row(s) affected  0.016 sec
48  10:33:23  Insert into Student values('12145','565814','Wong','Frank','03','06')  1 row(s) affected  0.015 sec
49  10:33:23  Insert into Student values('18225','561214','Jimmy','Pam','03','02')  1 row(s) affected  0.016 sec
50  10:33:23  Insert into Student values('12125','561814','Qiu','Alia','30','02')  1 row(s) affected  0.015 sec
51  10:33:23  Insert into Student values('12325','561234','Liu','Alice','01','04')  1 row(s) affected  0.000 sec
52  10:33:23  Insert into Student values('11825','565534','Lin','Peter','03','01')  1 row(s) affected  0.015 sec
53  10:33:23  Insert into Student values('10325','521234','Yang','Nory','02','03')  1 row(s) affected  0.031 sec
54  10:33:23  Insert into Student values('11345','568234','Kim','Fiona','10','01')  1 row(s) affected  0.016 sec
55  10:33:23  Insert into Student values('12653','582461','Jin','Joyce','06','10')  1 row(s) affected  0.031 sec
56  10:33:23  Insert into Student values('11010','761761','Kiki','Ann','01','02')  1 row(s) affected  0.016 sec
```

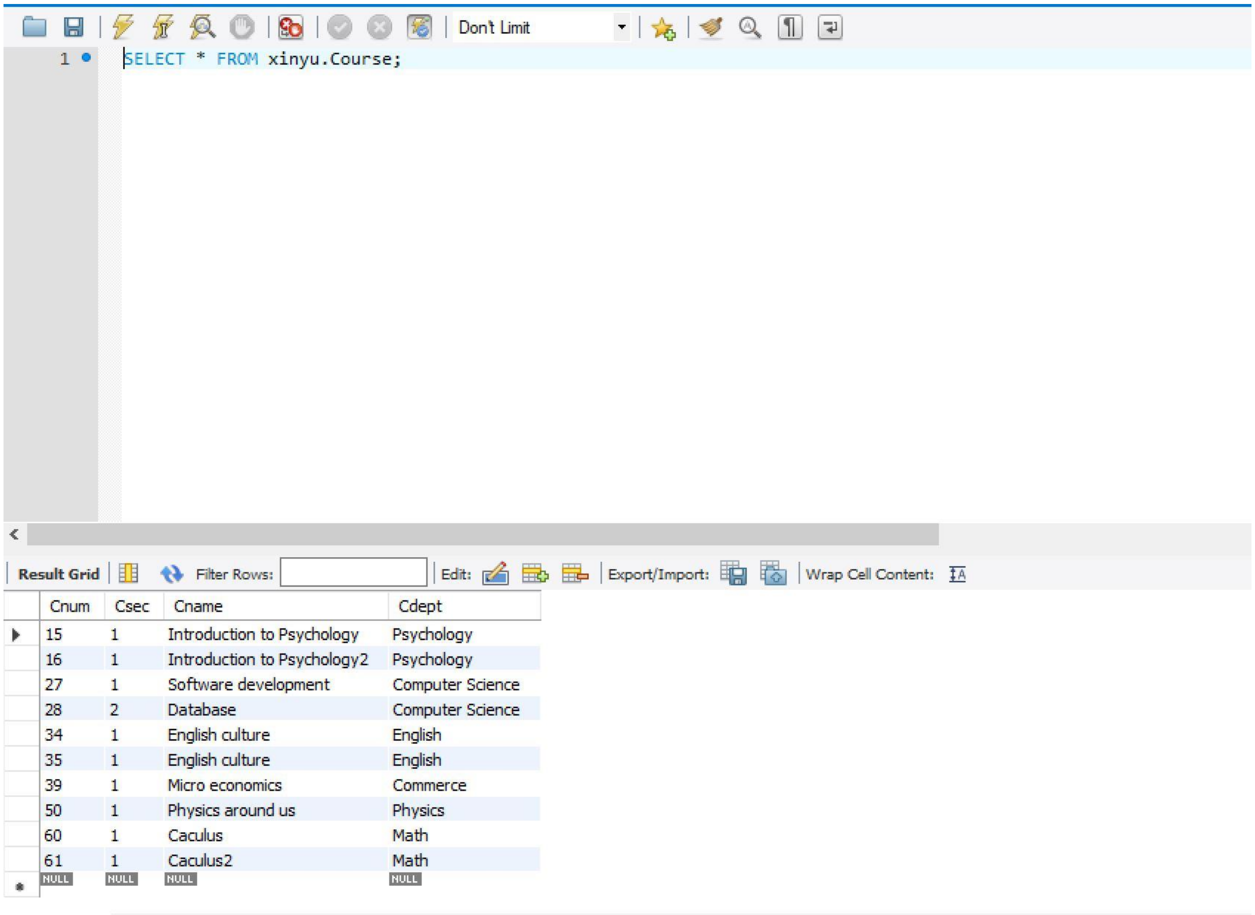
```
SELECT * FROM xinyu.Student;
```

Ssn	Bnum	Lame	Fname	Major_code	Minor_code
10025	472564	Bro	James	20	3
10325	521234	Yang	Nory	2	3
11010	761761	Kiki	Ann	1	2
11345	568234	Kim	Fiona	10	1
11825	565534	Lin	Peter	3	1
12125	561814	Qiu	Alia	30	2
12145	565814	Wong	Frank	3	6
12325	561234	Liu	Alice	1	4
12653	582461	Jin	Joyce	6	10
18225	561214	Jimmy	Pam	3	2
NULL	NULL	NULL	NULL	NULL	NULL



```
SELECT * FROM xinyu.Grade;
```

Grade	GSsn	GCnum
90	10010	35
84	11345	15
81	12125	35
94	12145	060
96	12145	061
72	12325	016
72	12325	50
92	12653	060
90	12653	061
90	12653	15
NULL	NULL	NULL



```sql
SELECT * FROM xinyu.Course;
```

Cnum	Csec	Cname	Cdept
15	1	Introduction to Psychology	Psychology
16	1	Introduction to Psychology2	Psychology
27	1	Software development	Computer Science
28	2	Database	Computer Science
34	1	English culture	English
35	1	English culture	English
39	1	Micro economics	Commerce
50	1	Physics around us	Physics
60	1	Caculus	Math
61	1	Caculus2	Math
NULL	NULL	NULL	NULL

e) The logical model of your database application should include: (15% - breakdown is below)

(1) Minimum one INSERT, one DELETE, one UPDATE query (3%)

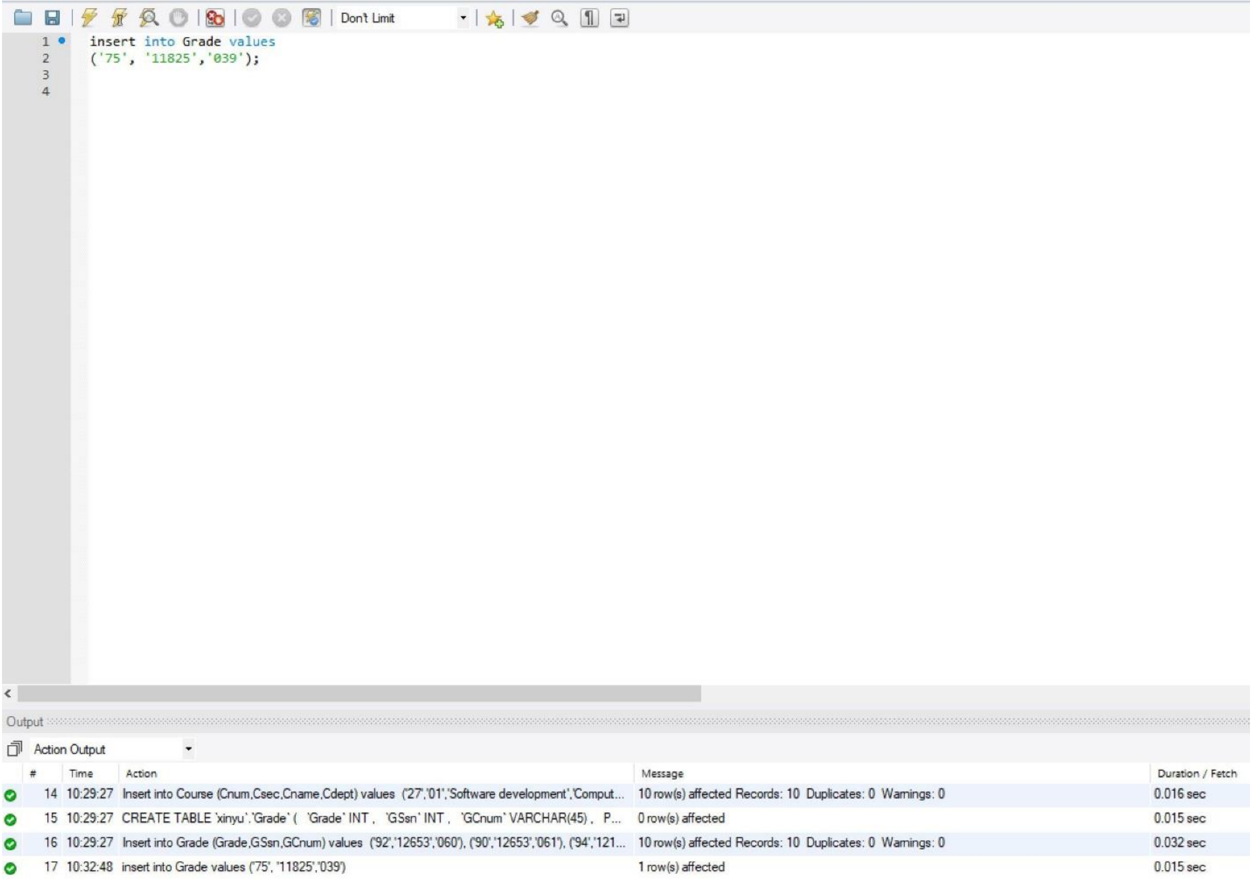
(2) Minimum three SELECT queries (3%)

(3) Minimum one JOIN, one GROUP BY, one VIEW query (3%)

(4) Minimum one Trigger (3%)

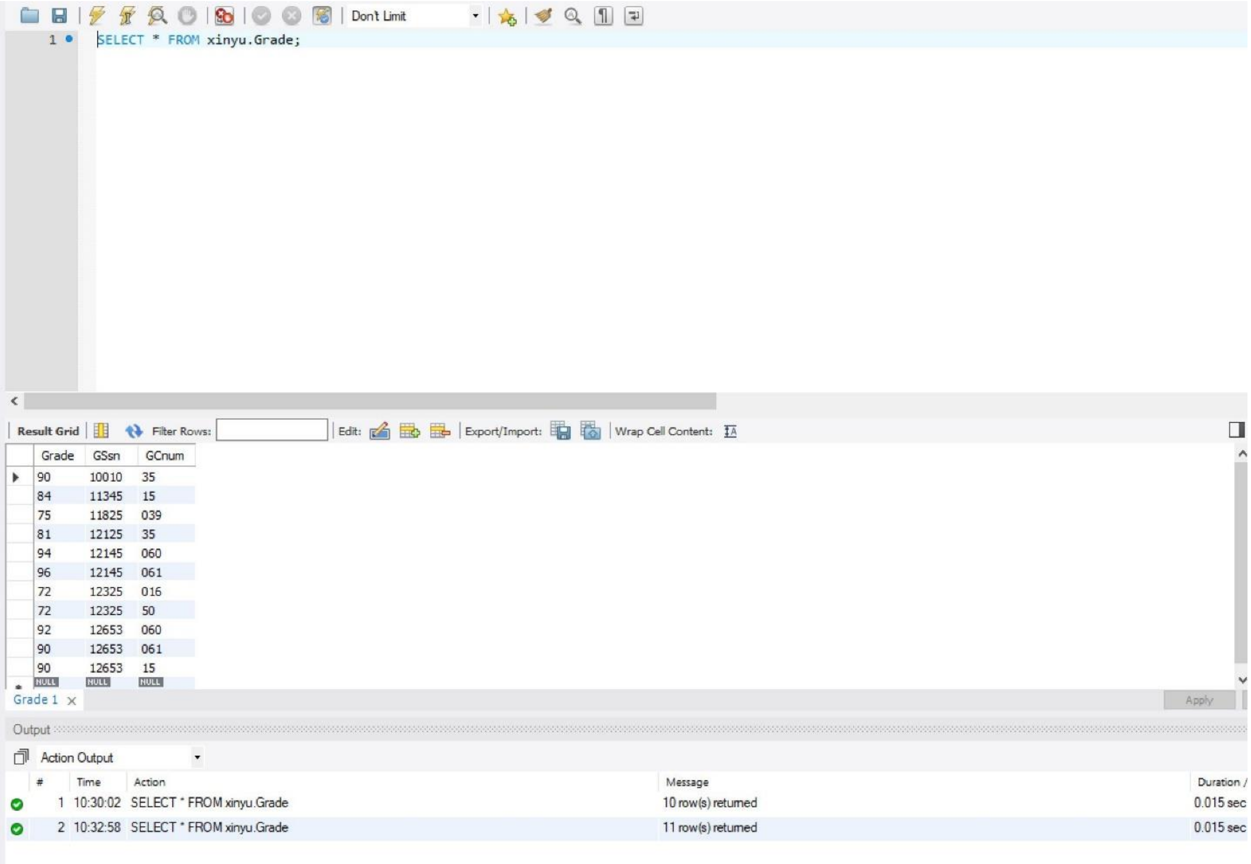
(5) Minimum one Stored Procedure (3%)

Insert and the result of insert:



```
insert into Grade values
('75', '11825','039');
```

	#	Time	Action	Message	Duration / Fetch
✓	14	10:29:27	Insert into Course (Cnum,Csec,Cname,Cdept) values ('27','01','Software development','Comput...	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.016 sec
✓	15	10:29:27	CREATE TABLE `xinyu`.`Grade` (`Grade` INT , `GSsn` INT , `GCnum` VARCHAR(45) , P...	0 row(s) affected	0.015 sec
✓	16	10:29:27	Insert into Grade (Grade,GSsn,GCnum) values ('92','12653','060'), ('90','12653','061'), ('94','121...	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.032 sec
✓	17	10:32:48	insert into Grade values ('75', '11825','039')	1 row(s) affected	0.015 sec

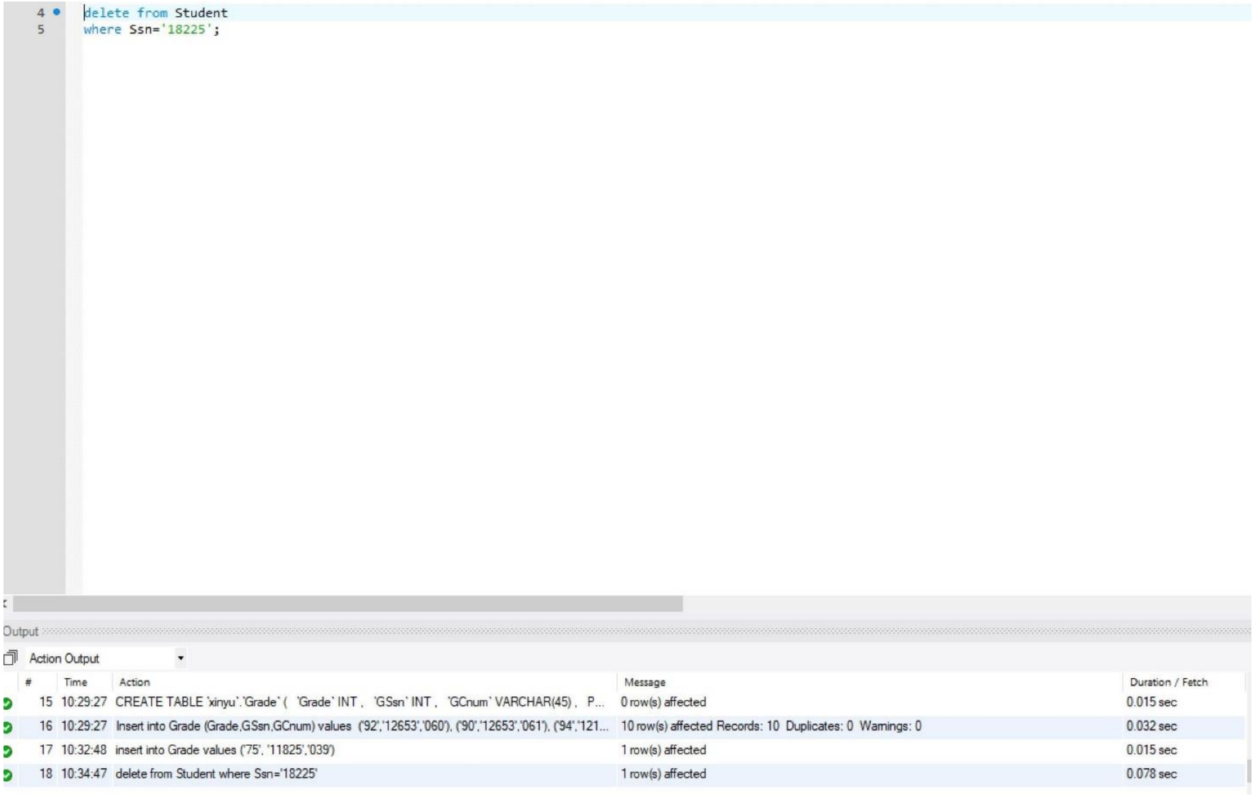


```
SELECT * FROM xinyu.Grade;
```

Grade	GSsn	GCnum
90	10010	35
84	11345	15
75	11825	039
81	12125	35
94	12145	060
96	12145	061
72	12325	016
72	12325	50
92	12653	060
90	12653	061
90	12653	15
NULL	NULL	NULL

```
#  Time      Action                       Message               Duration /
1  10:30:02  SELECT * FROM xinyu.Grade    10 row(s) returned    0.015 sec
2  10:32:58  SELECT * FROM xinyu.Grade    11 row(s) returned    0.015 sec
```

Delete:



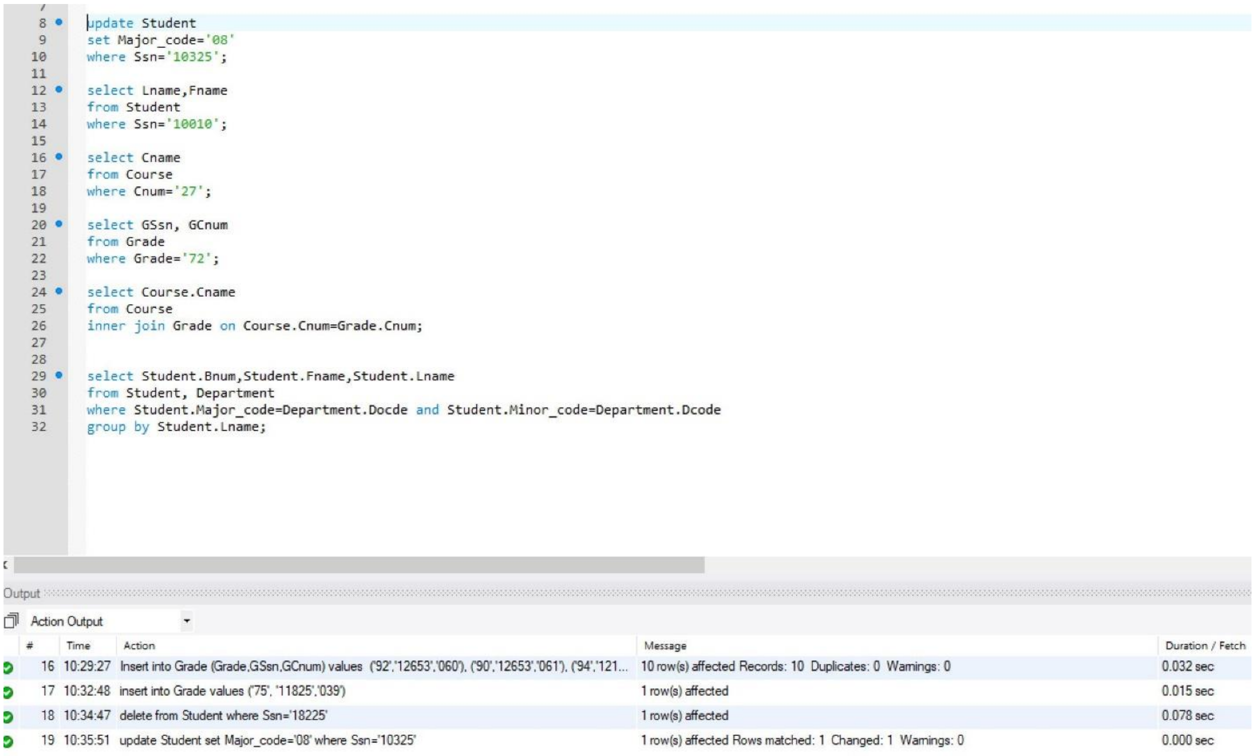
```sql
delete from Student
where Ssn='18225';
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
15	10:29:27	CREATE TABLE `xinyu`.`Grade` (`Grade` INT , `GSsn` INT , `GCnum` VARCHAR(45) , P...	0 row(s) affected	0.015 sec
16	10:29:27	Insert into Grade (Grade,GSsn,GCnum) values ('92','12653','060'), ('90','12653','061'), ('94','121...	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.032 sec
17	10:32:48	insert into Grade values ('75', '11825','039')	1 row(s) affected	0.015 sec
18	10:34:47	delete from Student where Ssn='18225'	1 row(s) affected	0.078 sec

Update:



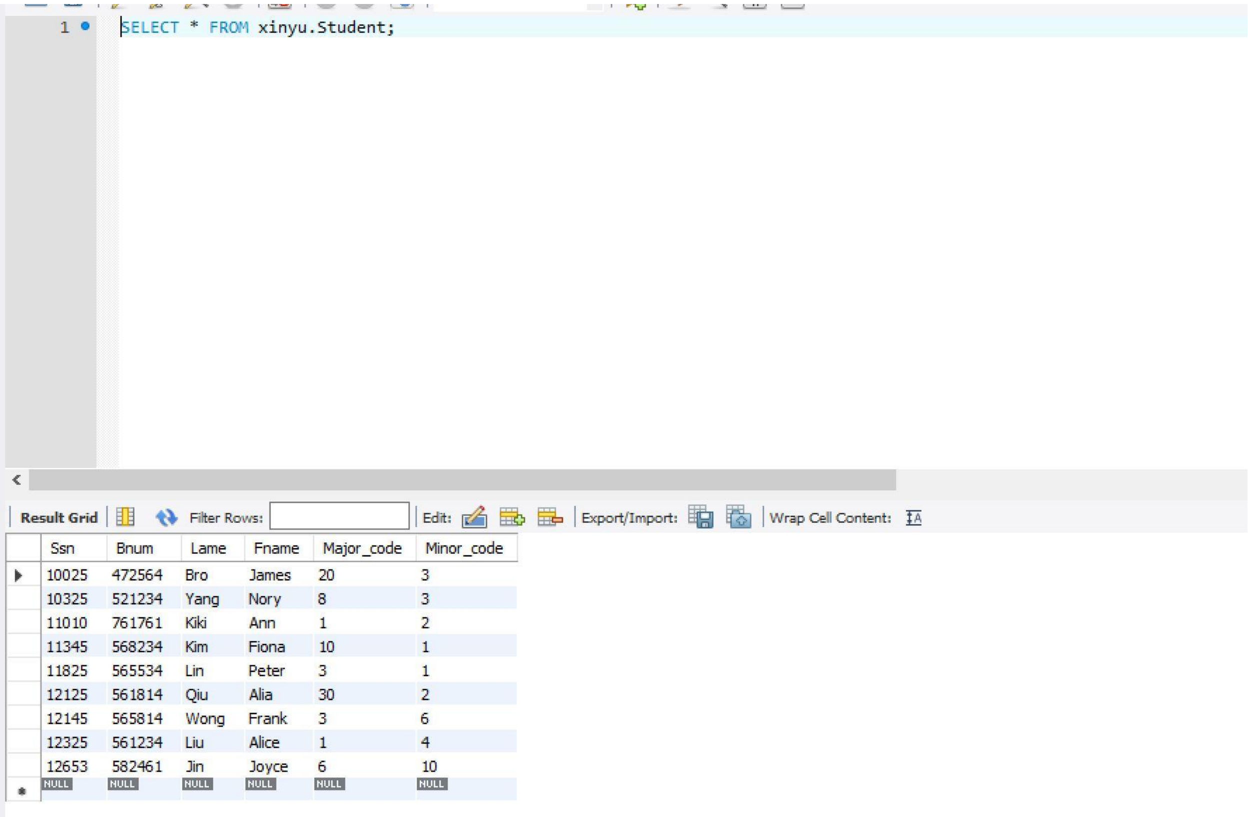
```sql
update Student
set Major_code='08'
where Ssn='10325';

select Lname,Fname
from Student
where Ssn='10010';

select Cname
from Course
where Cnum='27';

select GSsn, GCnum
from Grade
where Grade='72';

select Course.Cname
from Course
inner join Grade on Course.Cnum=Grade.Cnum;


select Student.Bnum,Student.Fname,Student.Lname
from Student, Department
where Student.Major_code=Department.Docde and Student.Minor_code=Department.Dcode
group by Student.Lname;
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
16	10:29:27	Insert into Grade (Grade,GSsn,GCnum) values ('92','12653','060'), ('90','12653','061'), ('94','121...	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.032 sec
17	10:32:48	insert into Grade values ('75', '11825','039')	1 row(s) affected	0.015 sec
18	10:34:47	delete from Student where Ssn='18225'	1 row(s) affected	0.078 sec
19	10:35:51	update Student set Major_code='08' where Ssn='10325'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec

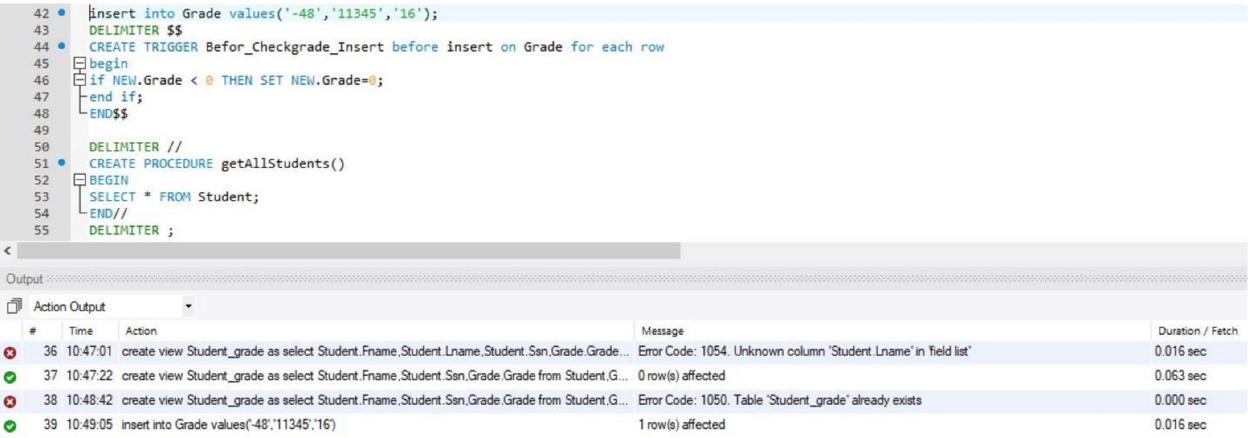
Result of delete and update:



```
SELECT * FROM xinyu.Student;
```

Ssn	Bnum	Lame	Fname	Major_code	Minor_code
10025	472564	Bro	James	20	3
10325	521234	Yang	Nory	8	3
11010	761761	Kiki	Ann	1	2
11345	568234	Kim	Fiona	10	1
11825	565534	Lin	Peter	3	1
12125	561814	Qiu	Alia	30	2
12145	565814	Wong	Frank	3	6
12325	561234	Liu	Alice	1	4
12653	582461	Jin	Joyce	6	10
NULL	NULL	NULL	NULL	NULL	NULL

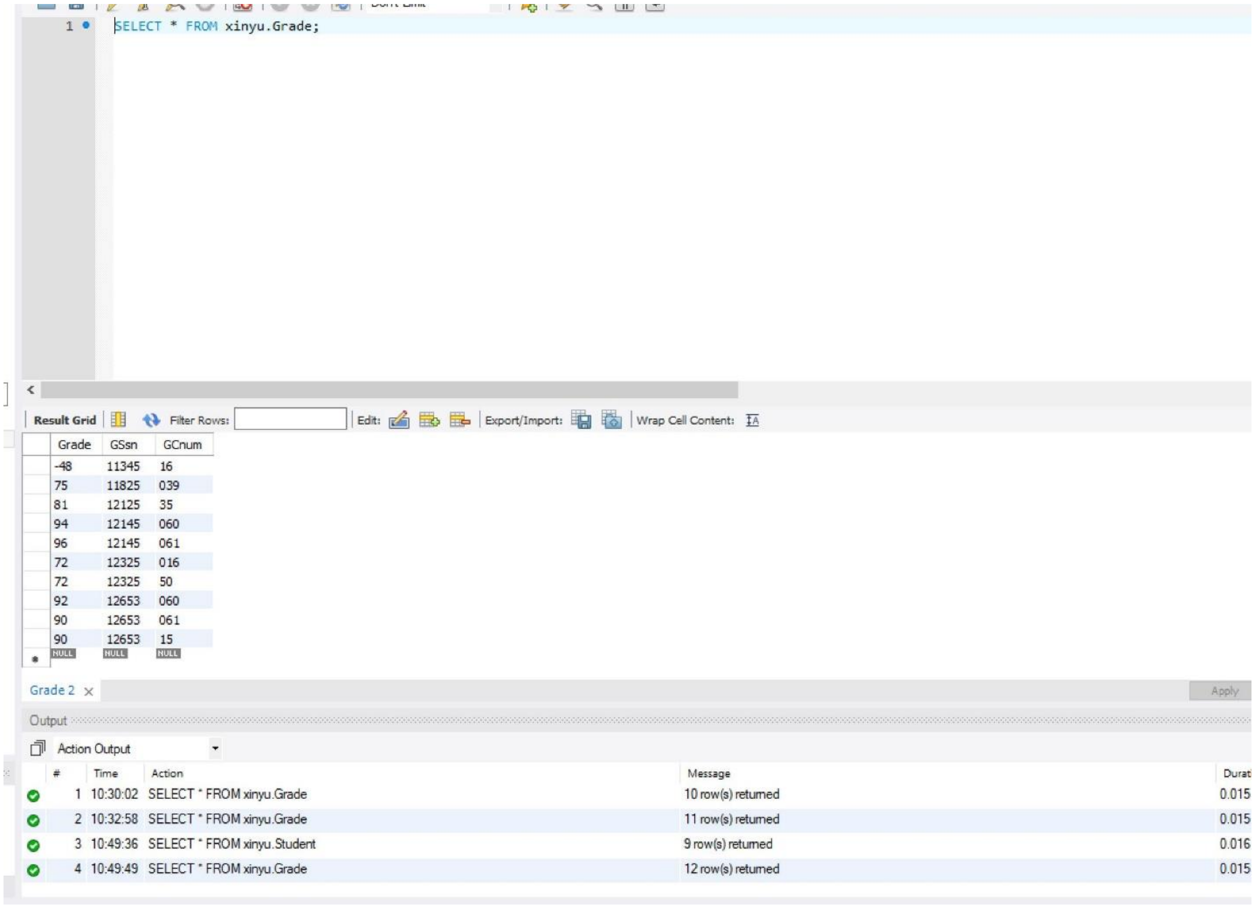
Insert:



```
insert into Grade values('-48','11345','16');
DELIMITER $$
CREATE TRIGGER Befor_Checkgrade_Insert before insert on Grade for each row
begin
if NEW.Grade < 0 THEN SET NEW.Grade=0;
end if;
END$$

DELIMITER //
CREATE PROCEDURE getAllStudents()
BEGIN
SELECT * FROM Student;
END//
DELIMITER ;
```

	#	Time	Action	Message	Duration / Fetch
☒	36	10:47:01	create view Student_grade as select Student.Fname,Student.Lname,Student.Ssn,Grade.Grade...	Error Code: 1054. Unknown column 'Student.Lname' in 'field list'	0.016 sec
✓	37	10:47:22	create view Student_grade as select Student.Fname,Student.Ssn,Grade.Grade from Student,G...	0 row(s) affected	0.063 sec
☒	38	10:48:42	create view Student_grade as select Student.Fname,Student.Ssn,Grade.Grade from Student,G...	Error Code: 1050. Table 'Student_grade' already exists	0.000 sec
✓	39	10:49:05	insert into Grade values('-48','11345','16')	1 row(s) affected	0.016 sec

Result:

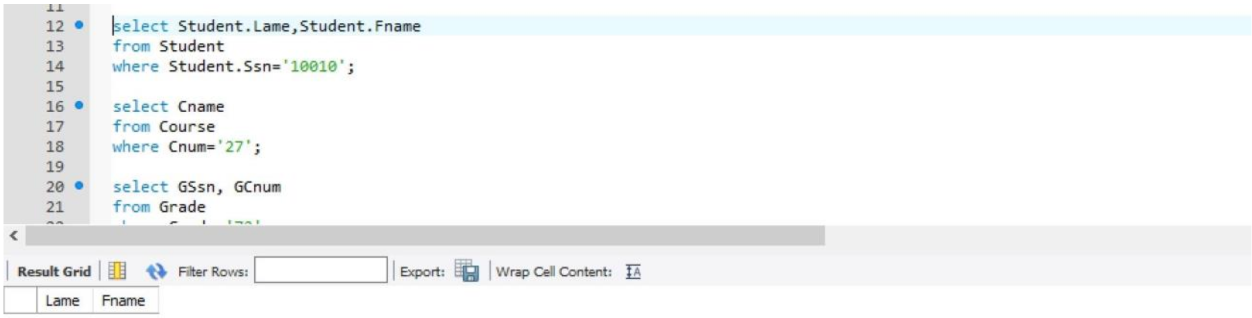


```
SELECT * FROM xinyu.Grade;
```

Grade	GSsn	GCnum
-48	11345	16
75	11825	039
81	12125	35
94	12145	060
96	12145	061
72	12325	016
72	12325	50
92	12653	060
90	12653	061
90	12653	15
NULL	NULL	NULL

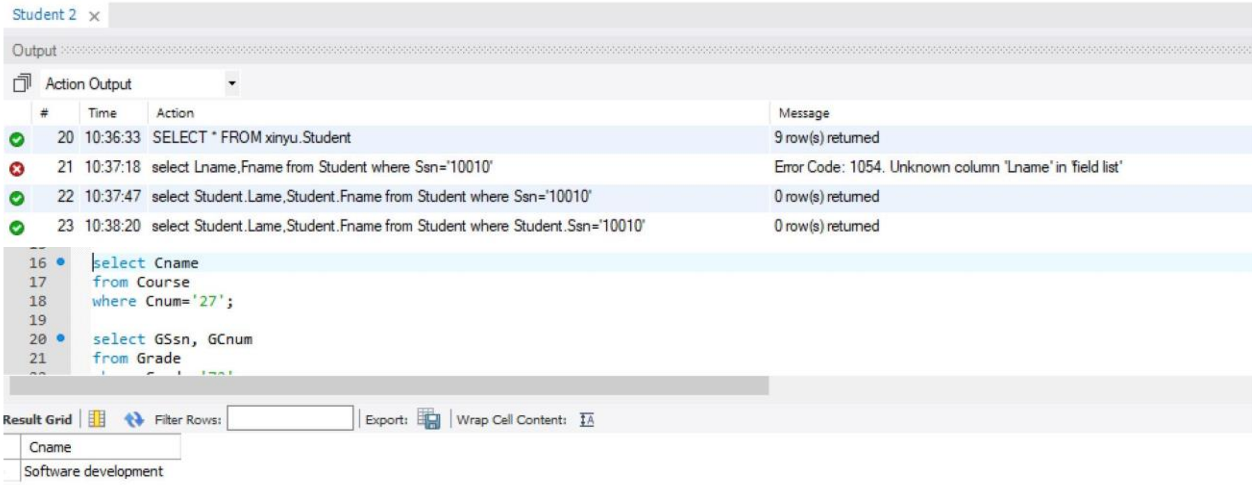
#	Time	Action	Message	Durat
1	10:30:02	SELECT * FROM xinyu.Grade	10 row(s) returned	0.015
2	10:32:58	SELECT * FROM xinyu.Grade	11 row(s) returned	0.015
3	10:49:36	SELECT * FROM xinyu.Student	9 row(s) returned	0.016
4	10:49:49	SELECT * FROM xinyu.Grade	12 row(s) returned	0.015

Select:



```
select Student.Lame,Student.Fname
from Student
where Student.Ssn='10010';

select Cname
from Course
where Cnum='27';

select GSsn, GCnum
from Grade
```

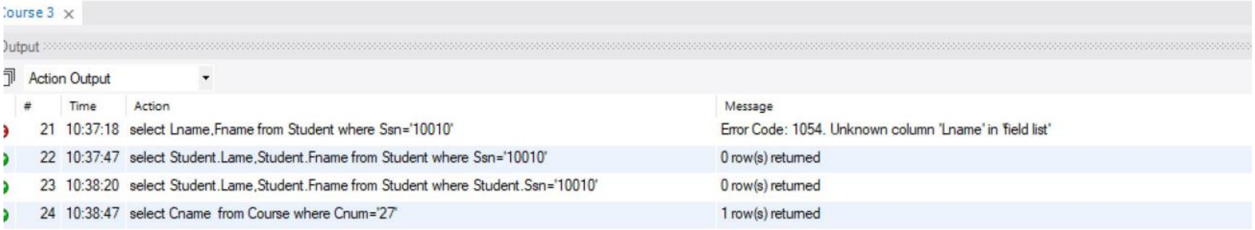
```
Lame  Fname
```



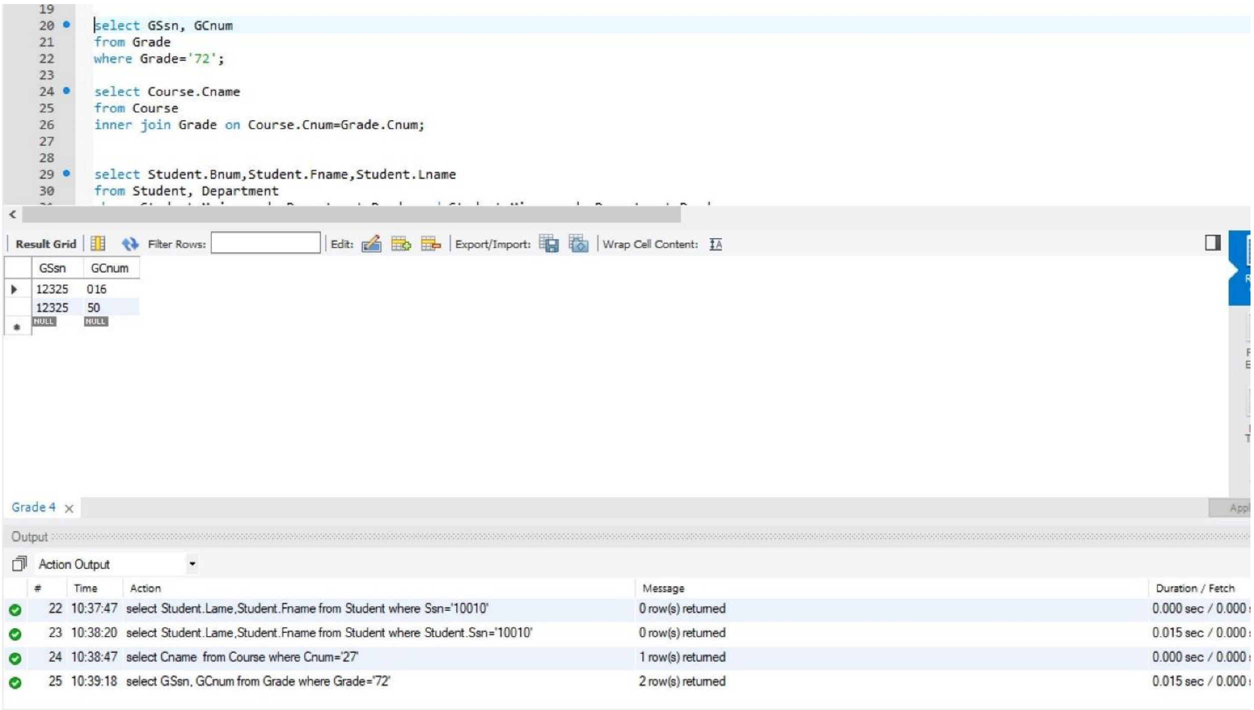
```
#   Time      Action                                                                  Message
20  10:36:33  SELECT * FROM xinyu.Student                                             9 row(s) returned
21  10:37:18  select Lname,Fname from Student where Ssn='10010'                       Error Code: 1054. Unknown column 'Lname' in 'field list'
22  10:37:47  select Student.Lame,Student.Fname from Student where Ssn='10010'        0 row(s) returned
23  10:38:20  select Student.Lame,Student.Fname from Student where Student.Ssn='10010'  0 row(s) returned
```

```
select Cname
from Course
where Cnum='27';

select GSsn, GCnum
from Grade
```

```
Cname
Software development
```



```
#   Time      Action                                                                  Message
21  10:37:18  select Lname,Fname from Student where Ssn='10010'                       Error Code: 1054. Unknown column 'Lname' in 'field list'
22  10:37:47  select Student.Lame,Student.Fname from Student where Ssn='10010'        0 row(s) returned
23  10:38:20  select Student.Lame,Student.Fname from Student where Student.Ssn='10010'  0 row(s) returned
24  10:38:47  select Cname  from Course where Cnum='27'                               1 row(s) returned
```

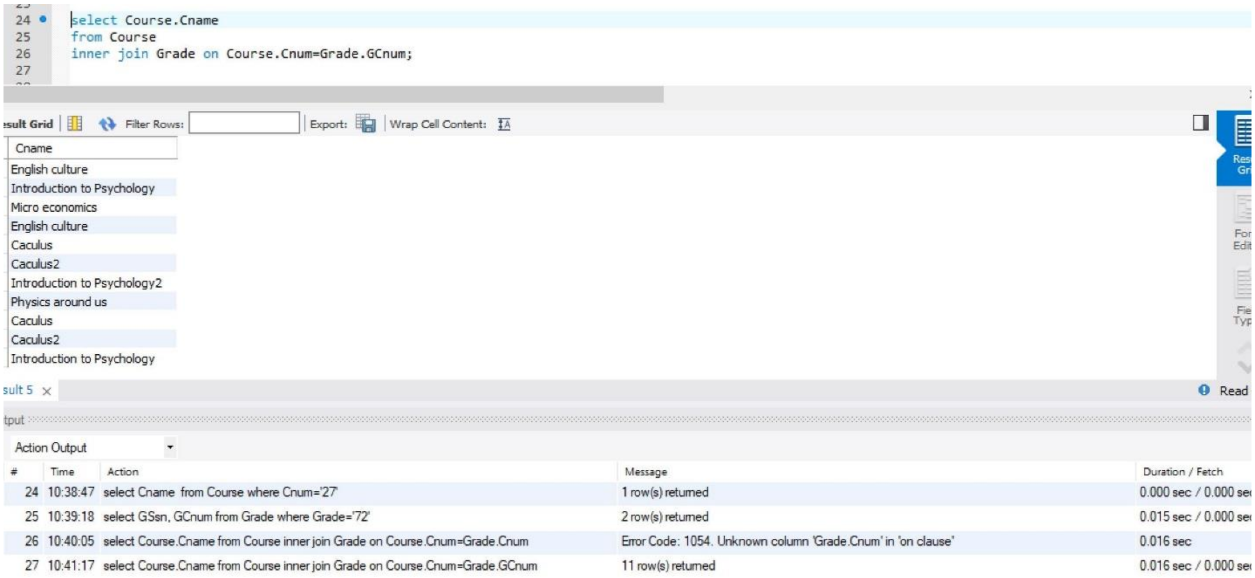


```
select GSsn, GCnum
from Grade
where Grade='72';

select Course.Cname
from Course
inner join Grade on Course.Cnum=Grade.Cnum;


select Student.Bnum,Student.Fname,Student.Lname
from Student, Department
```

GSsn	GCnum
12325	016
12325	50
NULL	NULL

```
#   Time      Action                                                                  Message            Duration / Fetch
22  10:37:47  select Student.Lame,Student.Fname from Student where Ssn='10010'          0 row(s) returned  0.000 sec / 0.000
23  10:38:20  select Student.Lame,Student.Fname from Student where Student.Ssn='10010'  0 row(s) returned  0.015 sec / 0.000
24  10:38:47  select Cname  from Course where Cnum='27'                                1 row(s) returned  0.000 sec / 0.000
25  10:39:18  select GSsn, GCnum from Grade where Grade='72'                           2 row(s) returned  0.015 sec / 0.000
```

Join:

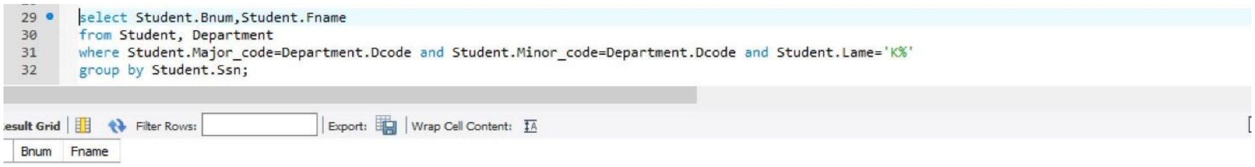


```
select Course.Cname
from Course
inner join Grade on Course.Cnum=Grade.GCnum;
```

Cname
English culture
Introduction to Psychology
Micro economics
English culture
Caculus
Caculus2
Introduction to Psychology2
Physics around us
Caculus
Caculus2
Introduction to Psychology

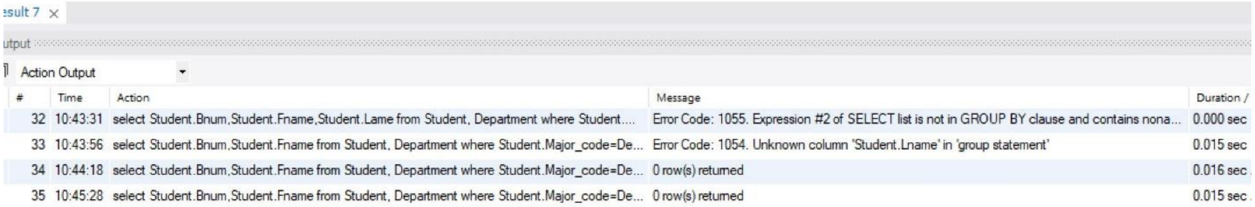
```
#   Time      Action                                                               Message                                                    Duration / Fetch
24  10:38:47  select Cname  from Course where Cnum='27'                             1 row(s) returned                                          0.000 sec / 0.000 sec
25  10:39:18  select GSsn, GCnum from Grade where Grade='72'                        2 row(s) returned                                          0.015 sec / 0.000 sec
26  10:40:05  select Course.Cname from Course inner join Grade on Course.Cnum=Grade.Cnum   Error Code: 1054. Unknown column 'Grade.Cnum' in 'on clause'   0.016 sec
27  10:41:17  select Course.Cname from Course inner join Grade on Course.Cnum=Grade.GCnum  11 row(s) returned                                         0.016 sec / 0.000 sec
```

Group by:



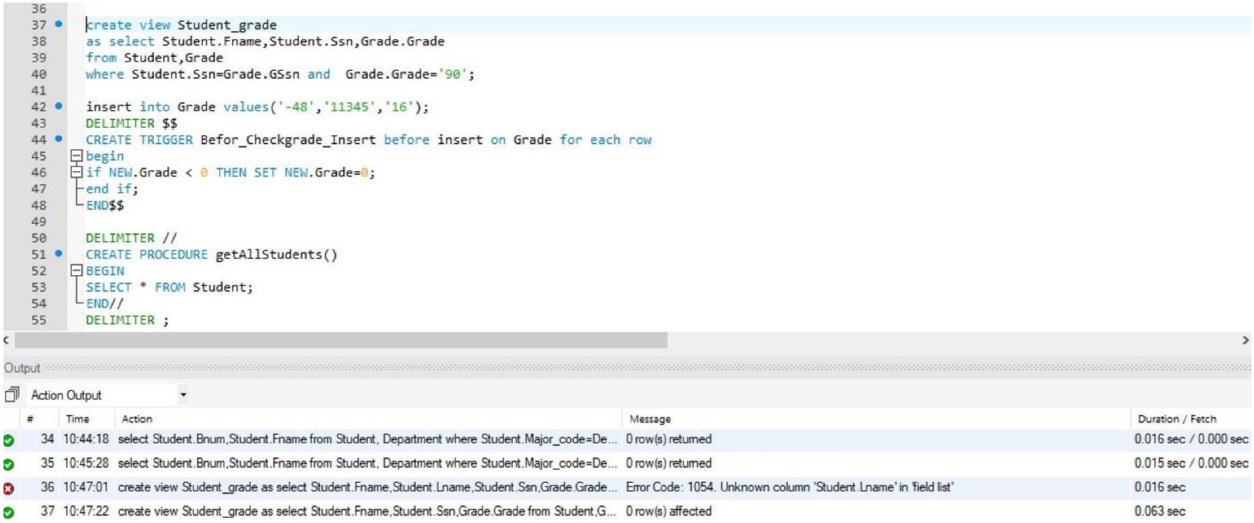
```sql
select Student.Bnum,Student.Fname
from Student, Department
where Student.Major_code=Department.Dcode and Student.Minor_code=Department.Dcode and Student.Lame='K%'
group by Student.Ssn;
```

Bnum	Fname



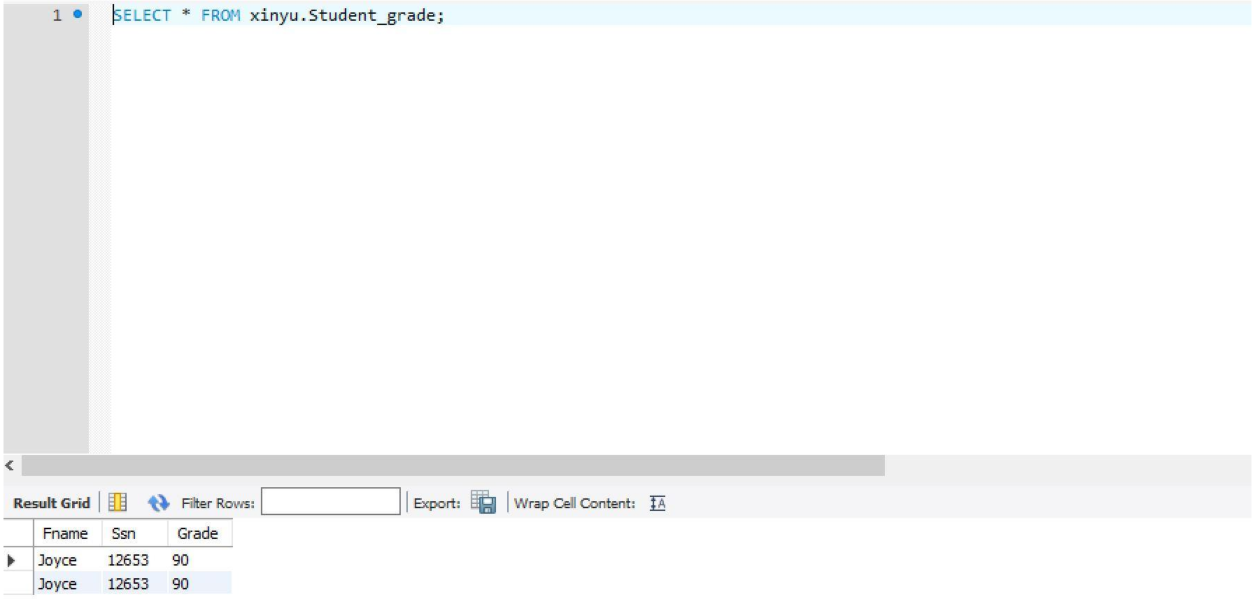
#	Time	Action	Message	Duration /
32	10:43:31	select Student.Bnum,Student.Fname,Student.Lame from Student, Department where Student....	Error Code: 1055. Expression #2 of SELECT list is not in GROUP BY clause and contains nona...	0.000 sec
33	10:43:56	select Student.Bnum,Student.Fname from Student, Department where Student.Major_code=De...	Error Code: 1054. Unknown column 'Student.Lname' in 'group statement'	0.015 sec
34	10:44:18	select Student.Bnum,Student.Fname from Student, Department where Student.Major_code=De...	0 row(s) returned	0.016 sec
35	10:45:28	select Student.Bnum,Student.Fname from Student, Department where Student.Major_code=De...	0 row(s) returned	0.015 sec

View:



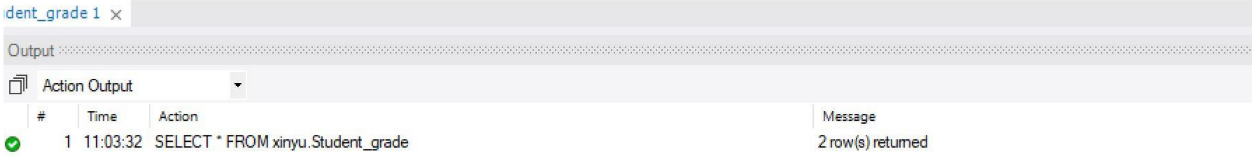
```sql

create view Student_grade
as select Student.Fname,Student.Ssn,Grade.Grade
from Student,Grade
where Student.Ssn=Grade.GSsn and  Grade.Grade='90';

insert into Grade values('-48','11345','16');
DELIMITER $$
CREATE TRIGGER Befor_Checkgrade_Insert before insert on Grade for each row
begin
if NEW.Grade < 0 THEN SET NEW.Grade=0;
end if;
END$$

DELIMITER //
CREATE PROCEDURE getAllStudents()
BEGIN
SELECT * FROM Student;
END//
DELIMITER ;
```

#	Time	Action	Message	Duration / Fetch
34	10:44:18	select Student.Bnum,Student.Fname from Student, Department where Student.Major_code=De...	0 row(s) returned	0.016 sec / 0.000 sec
35	10:45:28	select Student.Bnum,Student.Fname from Student, Department where Student.Major_code=De...	0 row(s) returned	0.015 sec / 0.000 sec
36	10:47:01	create view Student_grade as select Student.Fname,Student.Lname,Student.Ssn,Grade.Grade...	Error Code: 1054. Unknown column 'Student.Lname' in 'field list'	0.016 sec
37	10:47:22	create view Student_grade as select Student.Fname,Student.Ssn,Grade.Grade from Student,G...	0 row(s) affected	0.063 sec



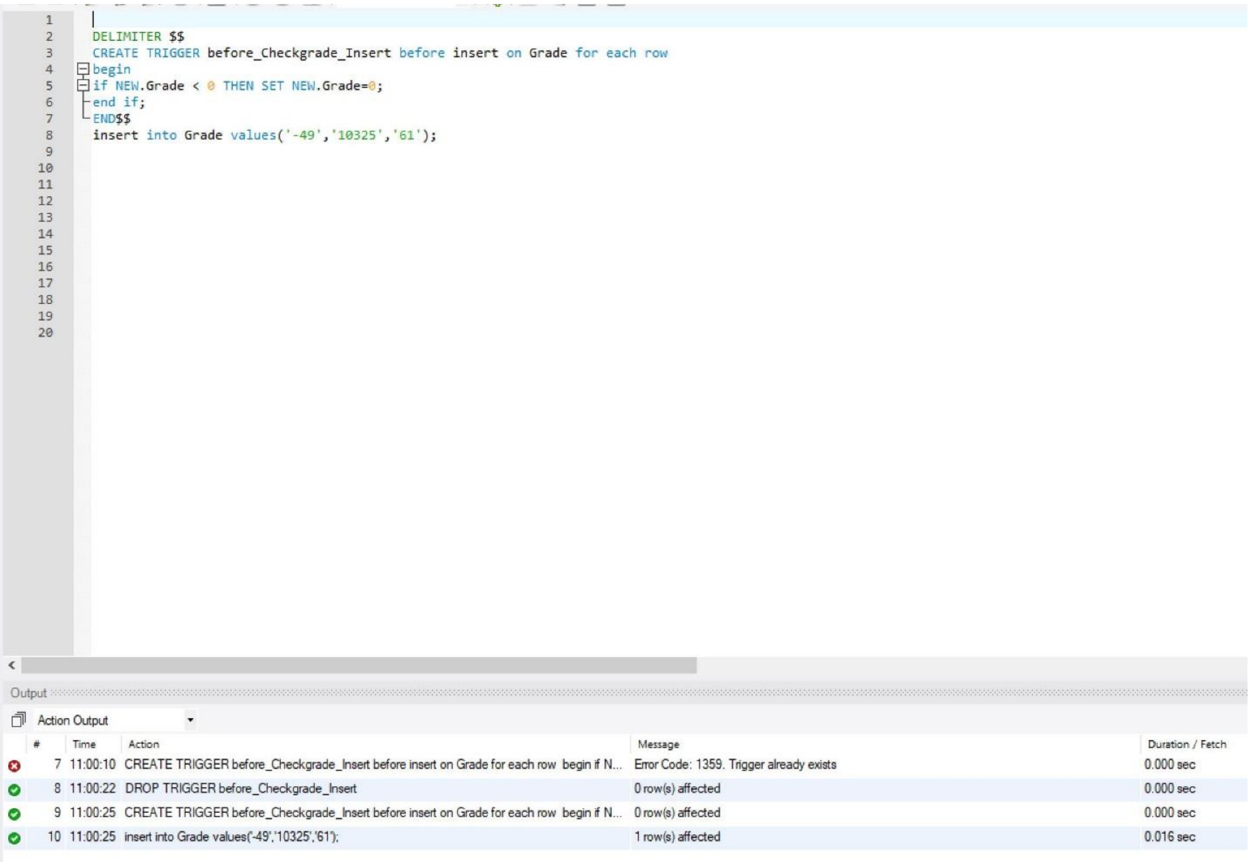
```sql
SELECT * FROM xinyu.Student_grade;
```

Fname	Ssn	Grade
Joyce	12653	90
Joyce	12653	90



#	Time	Action	Message
1	11:03:32	SELECT * FROM xinyu.Student_grade	2 row(s) returned

Trigger:

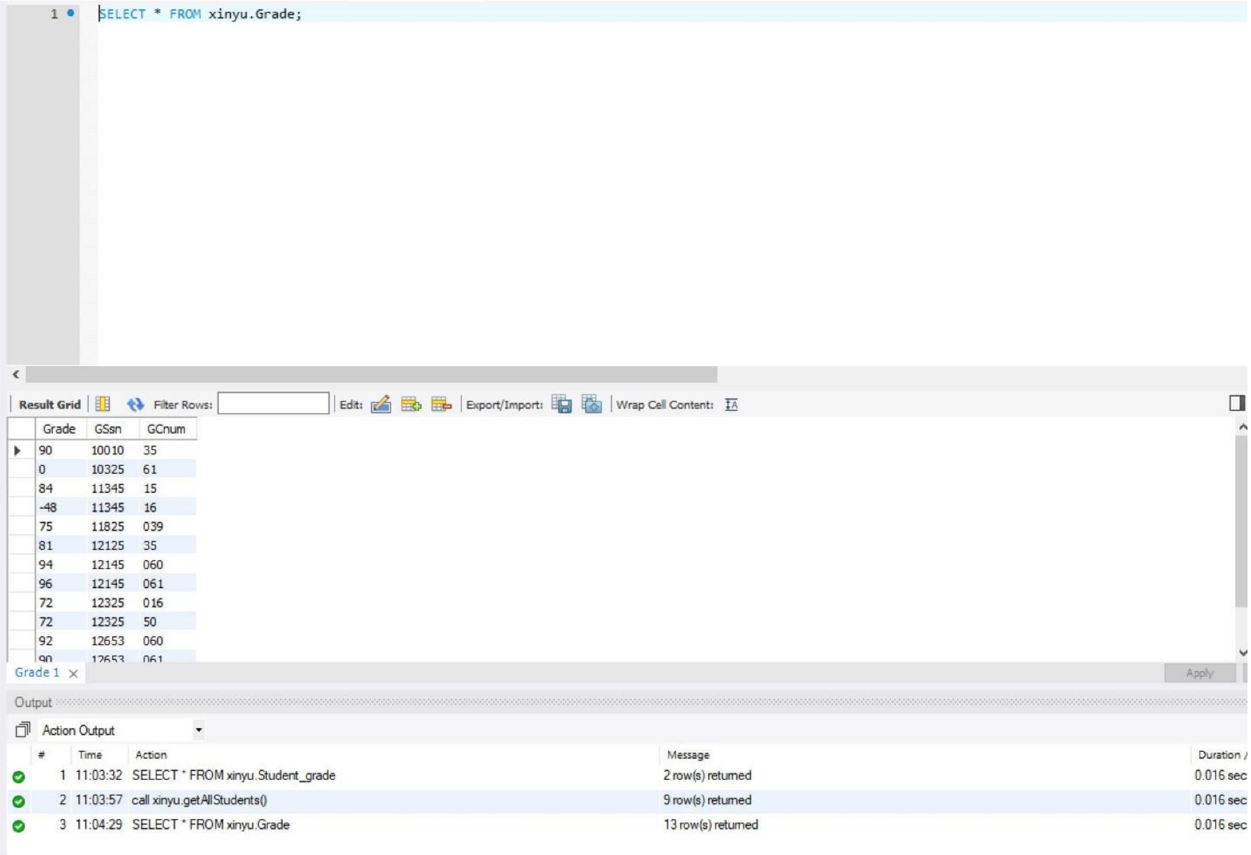


```sql

DELIMITER $$
CREATE TRIGGER before_Checkgrade_Insert before insert on Grade for each row
begin
if NEW.Grade < 0 THEN SET NEW.Grade=0;
end if;
END$$
insert into Grade values('-49','10325','61');
```

```
 #   Time      Action                                                                      Message                                  Duration / Fetch
 7   11:00:10  CREATE TRIGGER before_Checkgrade_Insert before insert on Grade for each row  begin if N...  Error Code: 1359. Trigger already exists  0.000 sec
 8   11:00:22  DROP TRIGGER before_Checkgrade_Insert                                       0 row(s) affected                        0.000 sec
 9   11:00:25  CREATE TRIGGER before_Checkgrade_Insert before insert on Grade for each row  begin if N...  0 row(s) affected                        0.000 sec
10   11:00:25  insert into Grade values('-49','10325','61');                               1 row(s) affected                        0.016 sec
```

If the insert value of grade<0, then set grade=0;

Result: set -49=0;

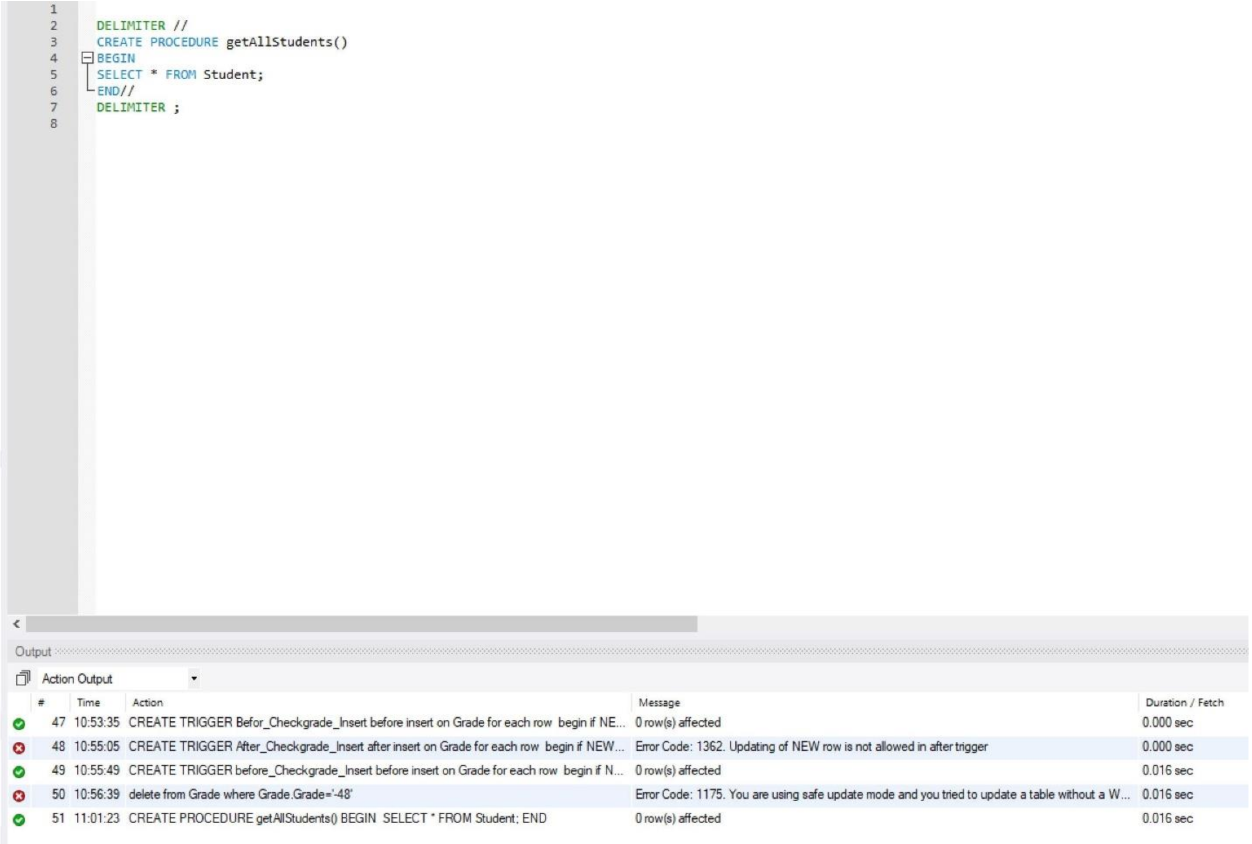


```
SELECT * FROM xinyu.Grade;
```

Grade	GSsn	GCnum
90	10010	35
0	10325	61
84	11345	15
-48	11345	16
75	11825	039
81	12125	35
94	12145	060
96	12145	061
72	12325	016
72	12325	50
92	12653	060
90	12653	061

```
#  Time      Action                               Message             Duration /
1  11:03:32  SELECT * FROM xinyu.Student_grade    2 row(s) returned   0.016 sec
2  11:03:57  call xinyu.getAllStudents()          9 row(s) returned   0.016 sec
3  11:04:29  SELECT * FROM xinyu.Grade            13 row(s) returned  0.016 sec
```

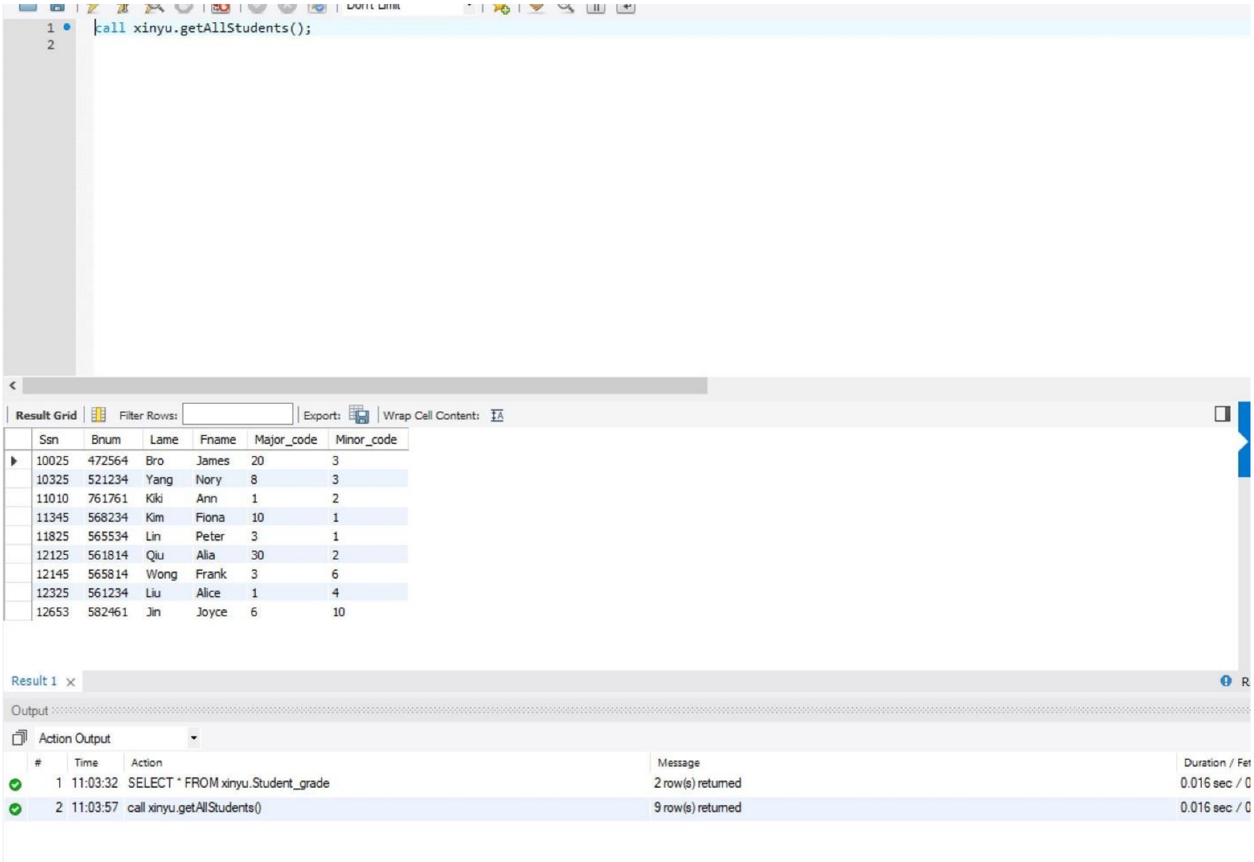
Procedure:



```sql
DELIMITER //
CREATE PROCEDURE getAllStudents()
BEGIN
SELECT * FROM Student;
END//
DELIMITER ;
```

#	Time	Action	Message	Duration / Fetch
✓ 47	10:53:35	CREATE TRIGGER Befor_Checkgrade_Insert before insert on Grade for each row begin if NE...	0 row(s) affected	0.000 sec
✗ 48	10:55:05	CREATE TRIGGER After_Checkgrade_Insert after insert on Grade for each row begin if NEW...	Error Code: 1362. Updating of NEW row is not allowed in after trigger	0.000 sec
✓ 49	10:55:49	CREATE TRIGGER before_Checkgrade_Insert before insert on Grade for each row begin if N...	0 row(s) affected	0.016 sec
✗ 50	10:56:39	delete from Grade where Grade.Grade='-48'	Error Code: 1175. You are using safe update mode and you tried to update a table without a W...	0.016 sec
✓ 51	11:01:23	CREATE PROCEDURE getAllStudents() BEGIN SELECT * FROM Student; END	0 row(s) affected	0.016 sec

Result of Procedure:



```
call xinyu.getAllStudents();
```

Ssn	Bnum	Lame	Fname	Major_code	Minor_code
10025	472564	Bro	James	20	3
10325	521234	Yang	Nory	8	3
11010	761761	Kiki	Ann	1	2
11345	568234	Kim	Fiona	10	1
11825	565534	Lin	Peter	3	1
12125	561814	Qiu	Alia	30	2
12145	565814	Wong	Frank	3	6
12325	561234	Liu	Alice	1	4
12653	582461	Jin	Joyce	6	10

```
#  Time      Action                                  Message             Duration / Fet
1  11:03:32  SELECT * FROM xinyu.Student_grade        2 row(s) returned   0.016 sec / 0
2  11:03:57  call xinyu.getAllStudents()              9 row(s) returned   0.016 sec / 0
```

2) Implement / Develop the physical model of your database application using MySql Workbench. Your MySQL code is the third tier (backend data tier) of your 3-Tier architecture. (25%)

See Project.file

3) The business logic forms the application tier, which is the second tier of your 3-Tier architecture. The application tier takes the information from the presentation tier and queries the data tier (backend). (20%)

See attached code

4) The first tier of your 3-Tier architecture is the presentation tier which enables the client/user to access the database. This user interface could be a form to fill in, or a field to choose etc. depending on your application and design. (10%)

See attached code