資工三 109590004 呂育瑋

Q1(a).

```
UPDATE COURSE
SET CreditHour = 3
WHERE CourseName = 'Database Systems'
    AND Department = 'EECS';
```

Q1(b).

```
DELETE FROM STUDENT
WHERE `Name` = 'Edward'
    AND StudentNumber = '001';
```

Q1(c).

```
SELECT CourseName
FROM COURSE
WHERE CourseNumber IN (
    SELECT CourseNumber
    FROM SECTION
    WHERE Instructor = 'Liu'
        AND `Year` IN (2020, 2022)
);
```

Q1(d).

```
SELECT CourseNumber, Semester, `Year`, COUNT(*) AS Students
FROM SECTION AS SEC
INNER JOIN GRADE_REPORT AS GR USING(SectionNumber)
WHERE Instructor = 'Liu'
GROUP BY CourseNumber, Semester, `Year`;
```

Q1(e).

```
SELECT PREREQUISITE.PrerequisiteCourseNumber, PRECOURSE.CourseName AS
PrerequisiteCourseName
FROM COURSE
INNER JOIN PREREQUISITE USING(CourseNumber)
INNER JOIN COURSE AS PRECOURSE
    ON PREREQUISITE.PrerequisiteCourseNumber = PRECOURSE.CourseNumber
WHERE COURSE.CourseName = 'Web Programming'
    AND COURSE.Department = 'EECS';
```

Q1(f).

```
SELECT `Name`, COURSE.CourseNumber, CourseName, CreditHour, Semester, `Year`, Grade
FROM STUDENT
INNER JOIN GRADE_REPORT USING(StudentNumber)
INNER JOIN SECTION USING(SectionNumber)
INNER JOIN COURSE USING(CourseNumber)
WHERE Class = 3;
```

Q1(g).

```
SELECT `Name`
FROM STUDENT
WHERE StudentNumber IN (
    SELECT StudentNumber
    FROM GRADE_REPORT
    WHERE
        Grade >= 80
);
```

Q1(h).

```
SELECT `Name`, Major
FROM STUDENT AS ST
WHERE NOT EXISTS (
    SELECT StudentNumber
    FROM GRADE_REPORT AS GR
    WHERE ST.StudentNumber = GR.StudentNumber
        AND Grade < 60
);
```

Q1(i).

```
SELECT `Name`, Major
FROM STUDENT AS ST
WHERE EXISTS (
    SELECT StudentNumber
    FROM GRADE_REPORT AS GR
    WHERE ST.StudentNumber = GR.StudentNumber
        AND Grade < 60
)
ORDER BY ST.StudentNumber ASC;
```

Q1(j).

```
SELECT `Name`, AVG(Grade) as AverageGrade
FROM GRADE_REPORT AS GR
INNER JOIN STUDENT AS ST USING(StudentNumber)
INNER JOIN SECTION AS SEC USING(SectionNumber)
WHERE `Year` = 2022
GROUP BY ST.StudentNumber HAVING AverageGrade > 80;
```

Q1(k).

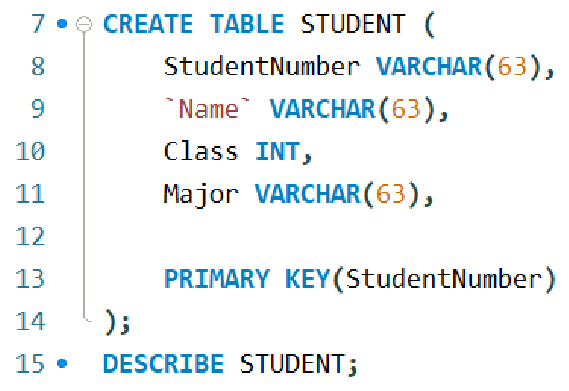
```
SELECT Major, COUNT(*)
FROM (
    SELECT Major, AVG(Grade) AS AverageGrade
    FROM GRADE_REPORT AS GR
    INNER JOIN STUDENT AS ST USING(StudentNumber)
    GROUP BY Major, ST.StudentNumber HAVING AverageGrade < 60
) AVGTABLE
GROUP BY Major;
```

Q1(l).

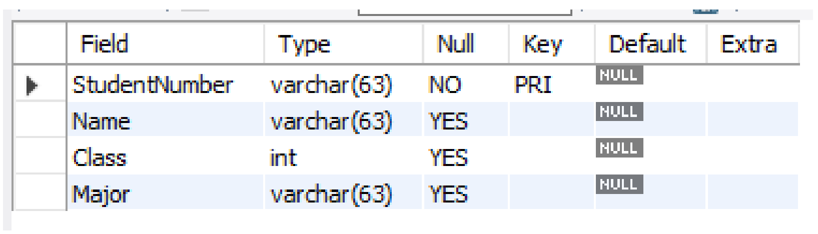
```
CREATE VIEW STUDENT_INFO_VIEW
AS
    SELECT StudentNumber, `Name`, CourseName, Semester, `Year`, Grade
    FROM STUDENT
    INNER JOIN GRADE_REPORT USING(StudentNumber)
    INNER JOIN SECTION USING(SectionNumber)
    INNER JOIN COURSE USING(CourseNumber);
```

Q2(a).

CREATE TABLE STUDENT



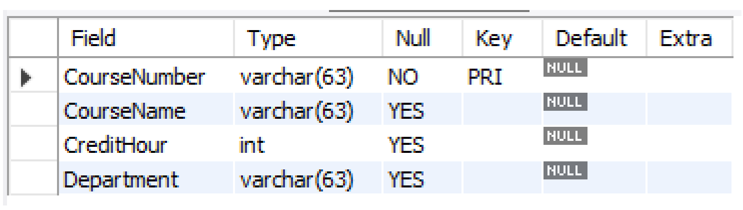
```sql
CREATE TABLE STUDENT (
    StudentNumber VARCHAR(63),
    `Name` VARCHAR(63),
    Class INT,
    Major VARCHAR(63),

    PRIMARY KEY(StudentNumber)
);
DESCRIBE STUDENT;
```



	Field	Type	Null	Key	Default	Extra
▶	StudentNumber	varchar(63)	NO	PRI	NULL	
	Name	varchar(63)	YES		NULL	
	Class	int	YES		NULL	
	Major	varchar(63)	YES		NULL	

CREATE TABLE COURSE

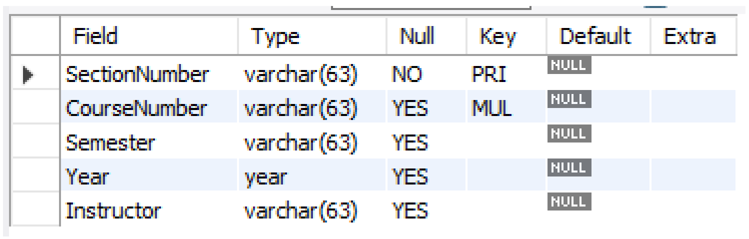
```sql
CREATE TABLE COURSE (
    CourseNumber VARCHAR(63),
    CourseName VARCHAR(63),
    CreditHour INT,
    Department VARCHAR(63),

    PRIMARY KEY(CourseNumber)
);
DESCRIBE COURSE;
```



	Field	Type	Null	Key	Default	Extra
▶	CourseNumber	varchar(63)	NO	PRI	NULL	
	CourseName	varchar(63)	YES		NULL	
	CreditHour	int	YES		NULL	
	Department	varchar(63)	YES		NULL	

CREATE TABLE SECTION

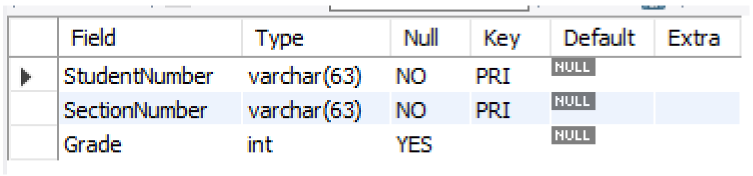
```sql
CREATE TABLE SECTION (
    SectionNumber VARCHAR(63),
    CourseNumber VARCHAR(63),
    Semester VARCHAR(63),
    `Year` YEAR,
    Instructor VARCHAR(63),

    PRIMARY KEY(SectionNumber),
    FOREIGN KEY(CourseNumber) REFERENCES COURSE(CourseNumber) ON DELETE SET NULL
);
DESCRIBE SECTION;
```



Field	Type	Null	Key	Default	Extra
SectionNumber	varchar(63)	NO	PRI	NULL	
CourseNumber	varchar(63)	YES	MUL	NULL	
Semester	varchar(63)	YES		NULL	
Year	year	YES		NULL	
Instructor	varchar(63)	YES		NULL	

CREATE TABLE GRADE_REPORT

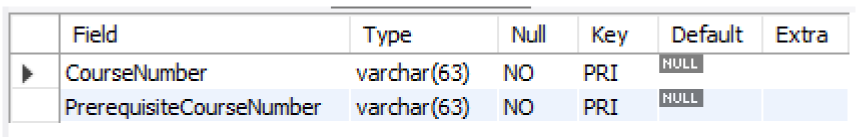
```sql
CREATE TABLE GRADE_REPORT (
    StudentNumber VARCHAR(63),
    SectionNumber VARCHAR(63),
    Grade INT,

    PRIMARY KEY(StudentNumber, SectionNumber),
    FOREIGN KEY(StudentNumber) REFERENCES STUDENT(StudentNumber) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY(SectionNumber) REFERENCES SECTION(SectionNumber) ON DELETE CASCADE ON UPDATE CASCADE
);
DESCRIBE GRADE_REPORT;
```



Field	Type	Null	Key	Default	Extra
StudentNumber	varchar(63)	NO	PRI	NULL	
SectionNumber	varchar(63)	NO	PRI	NULL	
Grade	int	YES		NULL	

CREATE TABLE PREREQUISITE

```sql
CREATE TABLE PREREQUISITE (
    CourseNumber VARCHAR(63),
    PrerequisiteCourseNumber VARCHAR(63),

    PRIMARY KEY(CourseNumber, PrerequisiteCourseNumber),
    FOREIGN KEY(CourseNumber) REFERENCES COURSE(CourseNumber) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY(PrerequisiteCourseNumber) REFERENCES COURSE(CourseNumber) ON DELETE CASCADE ON UPDATE CASCADE
);
DESCRIBE PREREQUISITE;
```

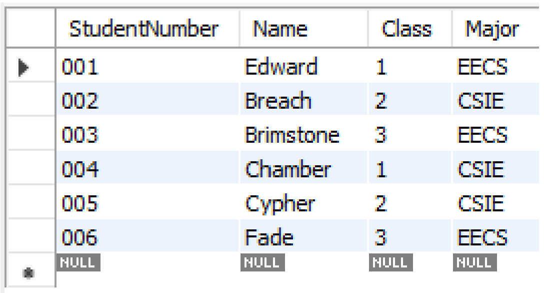


Field	Type	Null	Key	Default	Extra
CourseNumber	varchar(63)	NO	PRI	NULL	
PrerequisiteCourseNumber	varchar(63)	NO	PRI	NULL	

Q2(b).

INSERT INTO STUDENT

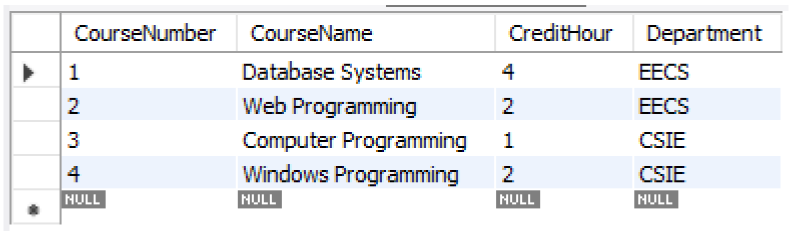
```
INSERT INTO STUDENT
VALUES
    ('001', 'Edward', 1, 'EECS'),
    ('002', 'Breach', 2, 'CSIE'),
    ('003', 'Brimstone', 3, 'EECS'),
    ('004', 'Chamber', 1, 'CSIE'),
    ('005', 'Cypher', 2, 'CSIE'),
    ('006', 'Fade', 3, 'EECS');
SELECT * FROM STUDENT;
```



StudentNumber	Name	Class	Major
001	Edward	1	EECS
002	Breach	2	CSIE
003	Brimstone	3	EECS
004	Chamber	1	CSIE
005	Cypher	2	CSIE
006	Fade	3	EECS
NULL	NULL	NULL	NULL

INSERT INTO COURSE

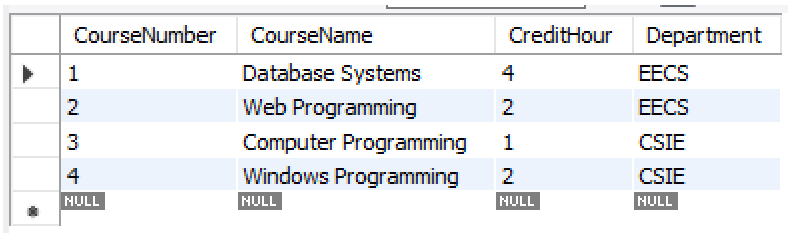
```
INSERT INTO COURSE
VALUES
    ('1', 'Database Systems', 4, 'EECS'),
    ('2', 'Web Programming', 2, 'EECS'),
    ('3', 'Computer Programming', 1, 'CSIE'),
    ('4', 'Windows Programming', 2, 'CSIE');
SELECT * FROM COURSE;
```



CourseNumber	CourseName	CreditHour	Department
1	Database Systems	4	EECS
2	Web Programming	2	EECS
3	Computer Programming	1	CSIE
4	Windows Programming	2	CSIE
NULL	NULL	NULL	NULL

INSERT INTO SECTION

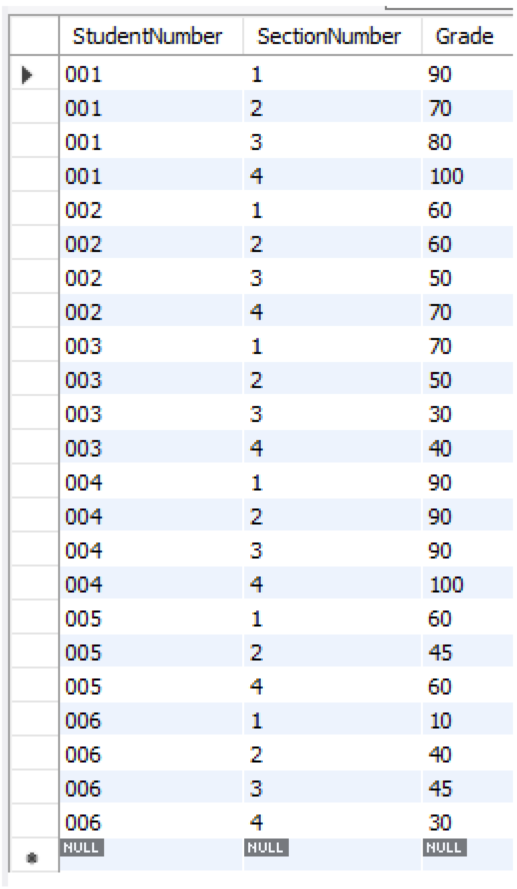
```
INSERT INTO SECTION
VALUES
    ('1', '1', 'Fall', 2020, 'Liu'),
    ('2', '4', 'Spring', 2021, 'Chen'),
    ('3', '2', 'Spring', 2022, 'Liu'),
    ('4', '3', 'Fall', 2022, 'Chen');
SELECT * FROM SECTION;
```



CourseNumber	CourseName	CreditHour	Department
1	Database Systems	4	EECS
2	Web Programming	2	EECS
3	Computer Programming	1	CSIE
4	Windows Programming	2	CSIE
NULL	NULL	NULL	NULL

INSERT INTO GRADE_REPORT

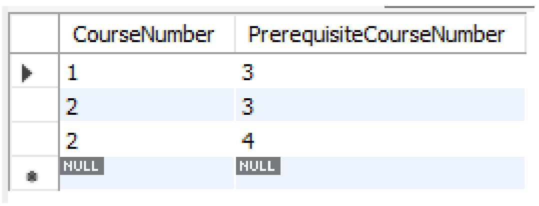
```sql
INSERT INTO GRADE_REPORT
VALUES
    ('001', '1', 90),
    ('001', '2', 70),
    ('001', '3', 80),
    ('001', '4', 100),
    ('002', '1', 60),
    ('002', '2', 60),
    ('002', '3', 50),
    ('002', '4', 70),
    ('003', '1', 70),
    ('003', '2', 50),
    ('003', '3', 30),
    ('003', '4', 40),
    ('004', '1', 90),
    ('004', '2', 90),
    ('004', '3', 90),
    ('004', '4', 100),
    ('005', '1', 60),
    ('005', '2', 45),
    ('005', '4', 60),
    ('006', '1', 10),
    ('006', '2', 40),
    ('006', '3', 45),
    ('006', '4', 30);
SELECT * FROM GRADE_REPORT;
```



	StudentNumber	SectionNumber	Grade
▶	001	1	90
	001	2	70
	001	3	80
	001	4	100
	002	1	60
	002	2	60
	002	3	50
	002	4	70
	003	1	70
	003	2	50
	003	3	30
	003	4	40
	004	1	90
	004	2	90
	004	3	90
	004	4	100
	005	1	60
	005	2	45
	005	4	60
	006	1	10
	006	2	40
	006	3	45
	006	4	30
*	NULL	NULL	NULL

INSERT INTO PREREQUISITE

```sql
INSERT INTO PREREQUISITE
VALUES
    ('2', '3'),
    ('2', '4'),
    ('1', '3');
SELECT * FROM PREREQUISITE;
```

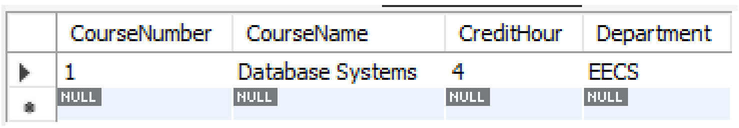


	CourseNumber	PrerequisiteCourseNumber
▶	1	3
	2	3
	2	4
*	NULL	NULL

Q2(c).

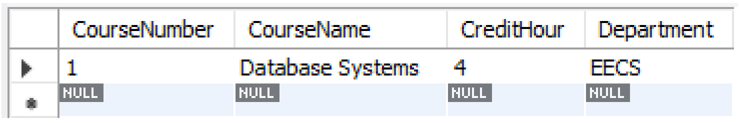
1(a)

```sql
SELECT *
FROM COURSE
WHERE CourseName = 'Database Systems'
    AND Department = 'EECS';
```



CourseNumber	CourseName	CreditHour	Department
1	Database Systems	4	EECS
NULL	NULL	NULL	NULL

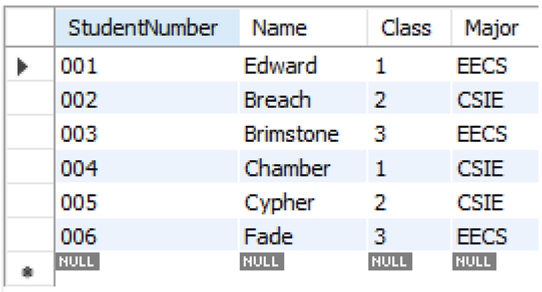
```sql
UPDATE COURSE
SET CreditHour = 3
WHERE CourseName = 'Database Systems'
    AND Department = 'EECS';

SELECT *
FROM COURSE
WHERE CourseName = 'Database Systems'
    AND Department = 'EECS';
```



CourseNumber	CourseName	CreditHour	Department
1	Database Systems	4	EECS
NULL	NULL	NULL	NULL

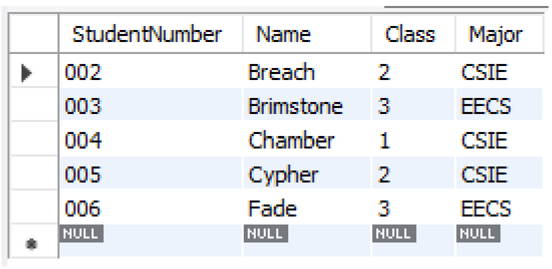
1(b)

```sql
SELECT * from STUDENT;
```



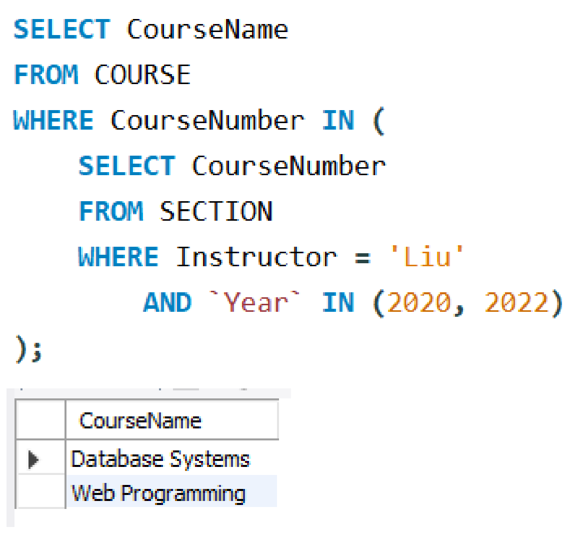
StudentNumber	Name	Class	Major
001	Edward	1	EECS
002	Breach	2	CSIE
003	Brimstone	3	EECS
004	Chamber	1	CSIE
005	Cypher	2	CSIE
006	Fade	3	EECS
NULL	NULL	NULL	NULL

```sql
DELETE FROM STUDENT
WHERE `Name` = 'Edward'
    AND StudentNumber = '001';

SELECT * from STUDENT;
```



StudentNumber	Name	Class	Major
002	Breach	2	CSIE
003	Brimstone	3	EECS
004	Chamber	1	CSIE
005	Cypher	2	CSIE
006	Fade	3	EECS
NULL	NULL	NULL	NULL

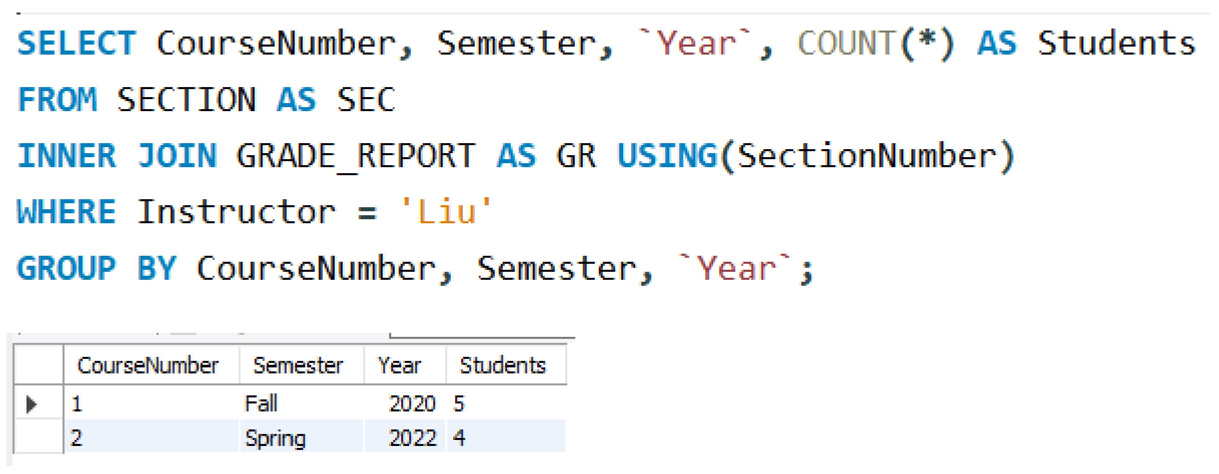
1(c)



```sql
SELECT CourseName
FROM COURSE
WHERE CourseNumber IN (
    SELECT CourseNumber
    FROM SECTION
    WHERE Instructor = 'Liu'
        AND `Year` IN (2020, 2022)
);
```

CourseName
Database Systems
Web Programming

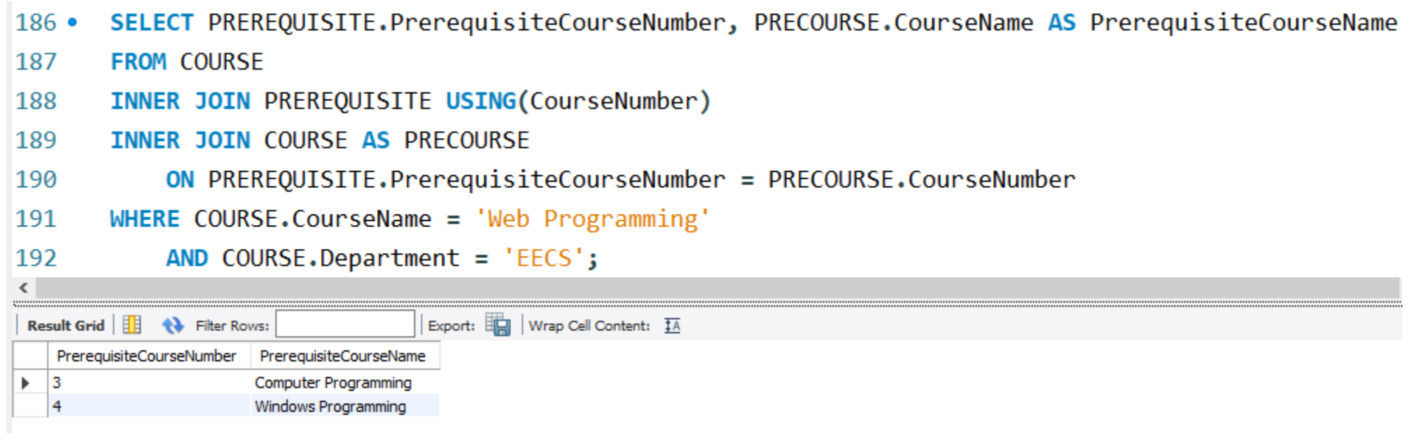
1(d)



```sql
SELECT CourseNumber, Semester, `Year`, COUNT(*) AS Students
FROM SECTION AS SEC
INNER JOIN GRADE_REPORT AS GR USING(SectionNumber)
WHERE Instructor = 'Liu'
GROUP BY CourseNumber, Semester, `Year`;
```

CourseNumber	Semester	Year	Students
1	Fall	2020	5
2	Spring	2022	4

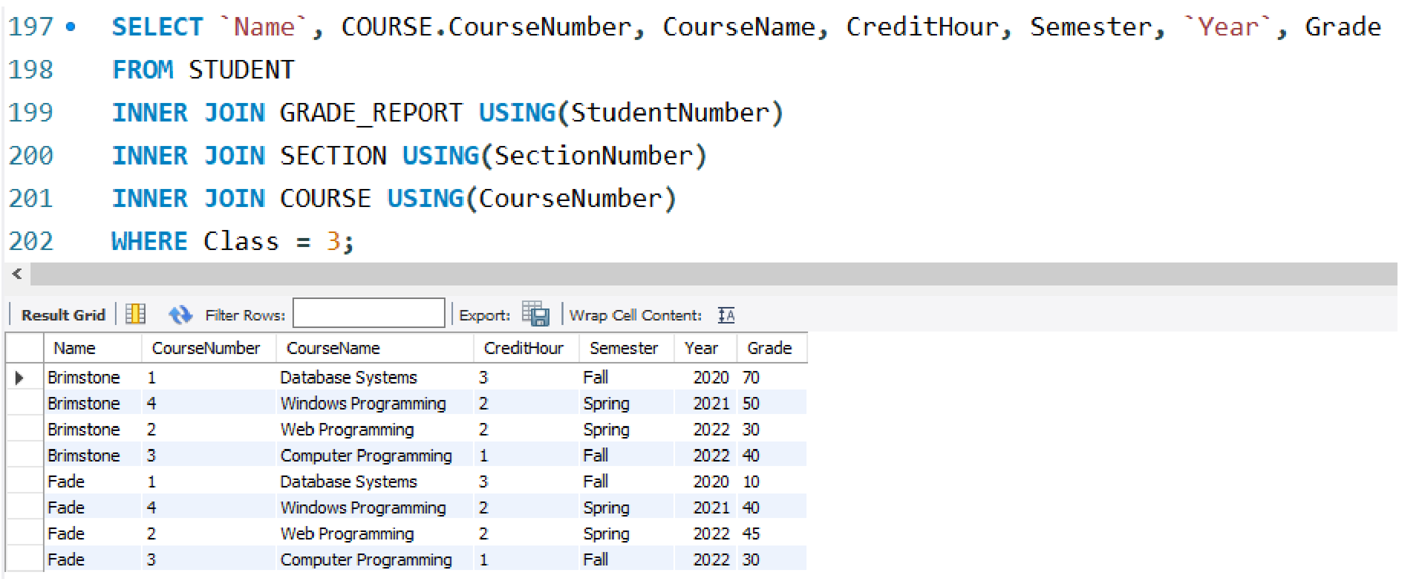
1(e)



```sql
SELECT PREREQUISITE.PrerequisiteCourseNumber, PRECOURSE.CourseName AS PrerequisiteCourseName
FROM COURSE
INNER JOIN PREREQUISITE USING(CourseNumber)
INNER JOIN COURSE AS PRECOURSE
    ON PREREQUISITE.PrerequisiteCourseNumber = PRECOURSE.CourseNumber
WHERE COURSE.CourseName = 'Web Programming'
    AND COURSE.Department = 'EECS';
```

PrerequisiteCourseNumber	PrerequisiteCourseName
3	Computer Programming
4	Windows Programming

1(f)

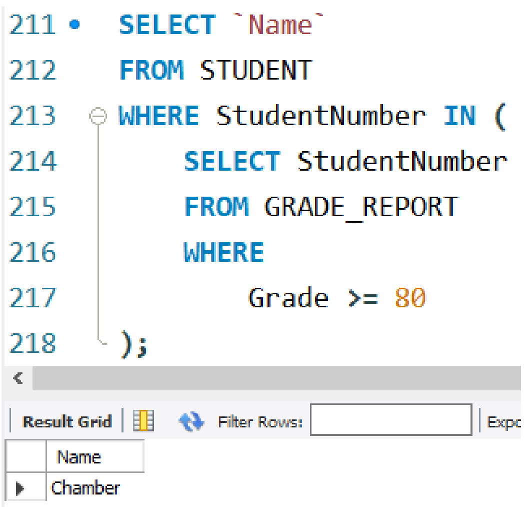


```sql
SELECT `Name`, COURSE.CourseNumber, CourseName, CreditHour, Semester, `Year`, Grade
FROM STUDENT
INNER JOIN GRADE_REPORT USING(StudentNumber)
INNER JOIN SECTION USING(SectionNumber)
INNER JOIN COURSE USING(CourseNumber)
WHERE Class = 3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Name	CourseNumber	CourseName	CreditHour	Semester	Year	Grade
Brimstone	1	Database Systems	3	Fall	2020	70
Brimstone	4	Windows Programming	2	Spring	2021	50
Brimstone	2	Web Programming	2	Spring	2022	30
Brimstone	3	Computer Programming	1	Fall	2022	40
Fade	1	Database Systems	3	Fall	2020	10
Fade	4	Windows Programming	2	Spring	2021	40
Fade	2	Web Programming	2	Spring	2022	45
Fade	3	Computer Programming	1	Fall	2022	30

1(g)

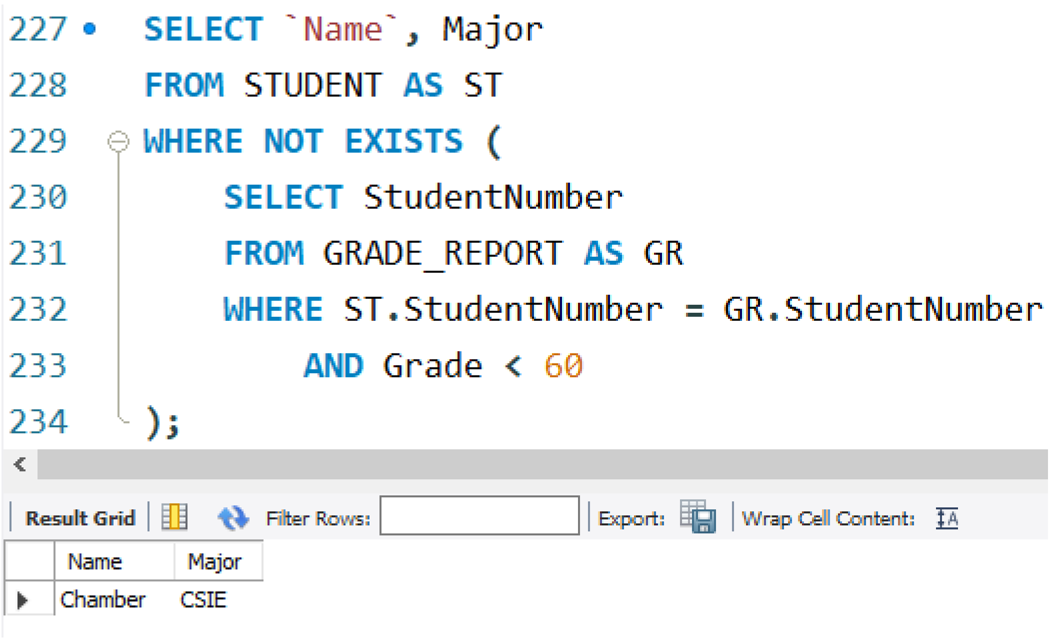


```sql
SELECT `Name`
FROM STUDENT
WHERE StudentNumber IN (
    SELECT StudentNumber
    FROM GRADE_REPORT
    WHERE
        Grade >= 80
);
```

Result Grid | Filter Rows: | Expo

Name
Chamber

1(h)

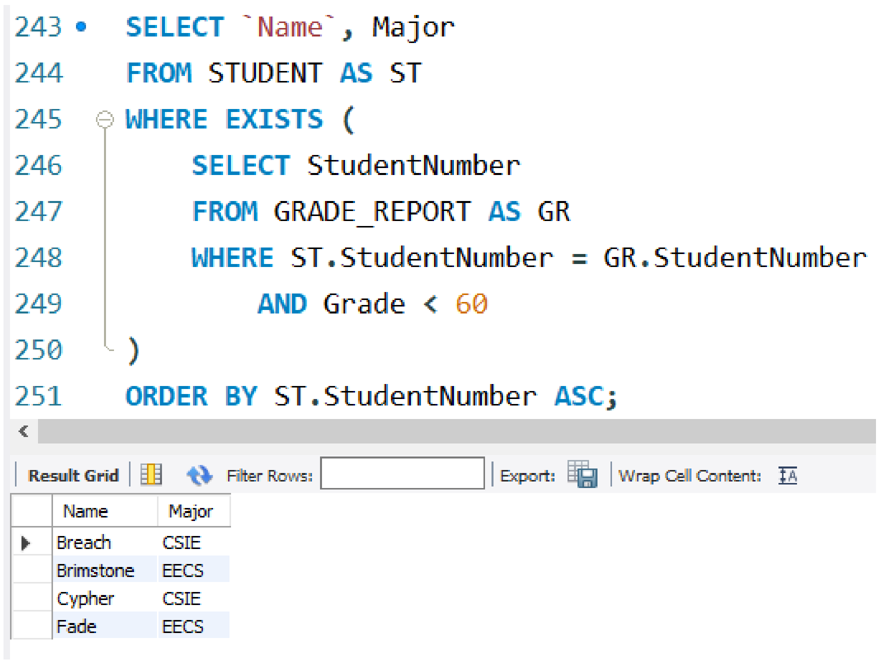


```sql
SELECT `Name`, Major
FROM STUDENT AS ST
WHERE NOT EXISTS (
    SELECT StudentNumber
    FROM GRADE_REPORT AS GR
    WHERE ST.StudentNumber = GR.StudentNumber
        AND Grade < 60
);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Name	Major
Chamber	CSIE

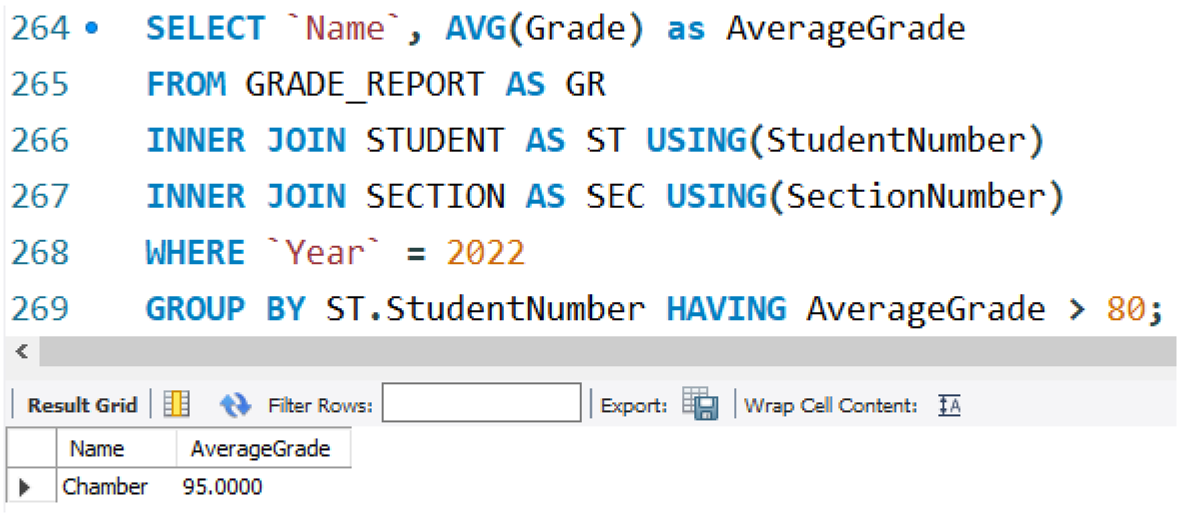
1(i)



```sql
SELECT `Name`, Major
FROM STUDENT AS ST
WHERE EXISTS (
    SELECT StudentNumber
    FROM GRADE_REPORT AS GR
    WHERE ST.StudentNumber = GR.StudentNumber
        AND Grade < 60
)
ORDER BY ST.StudentNumber ASC;
```

```
Name       Major
Breach     CSIE
Brimstone  EECS
Cypher     CSIE
Fade       EECS
```

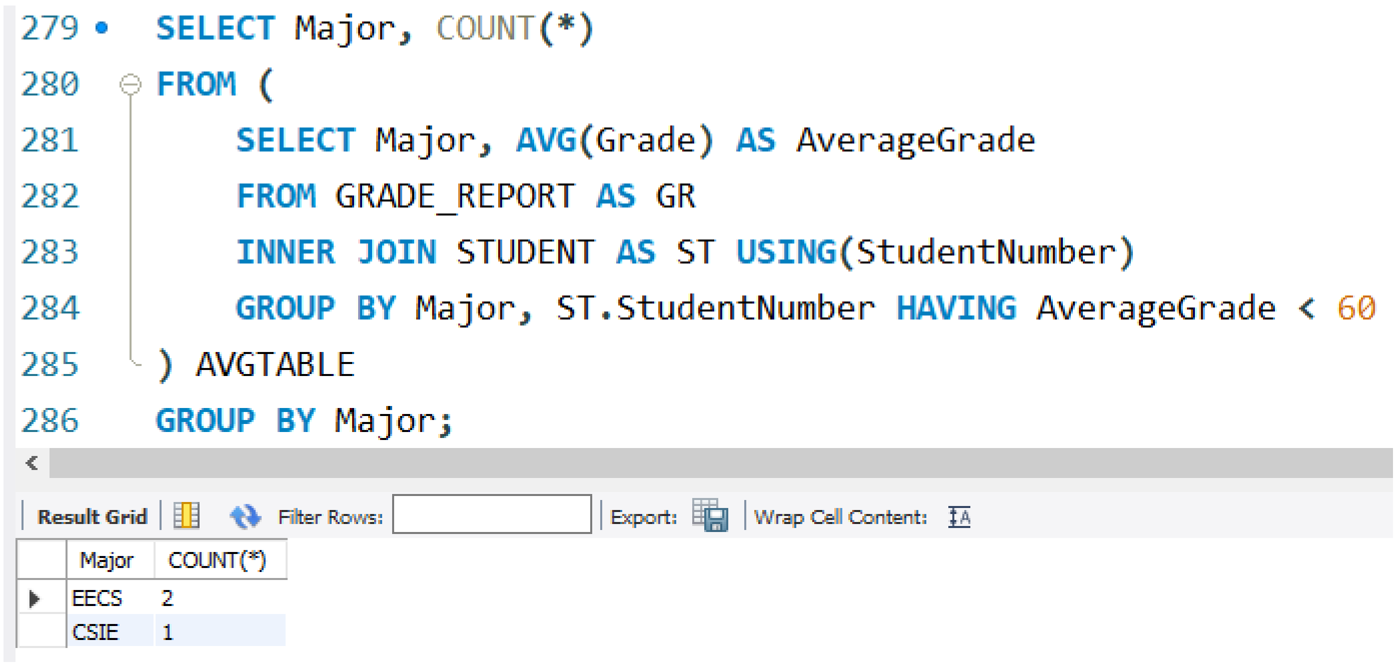
1(j)



```sql
SELECT `Name`, AVG(Grade) as AverageGrade
FROM GRADE_REPORT AS GR
INNER JOIN STUDENT AS ST USING(StudentNumber)
INNER JOIN SECTION AS SEC USING(SectionNumber)
WHERE `Year` = 2022
GROUP BY ST.StudentNumber HAVING AverageGrade > 80;
```

```
Name      AverageGrade
Chamber   95.0000
```

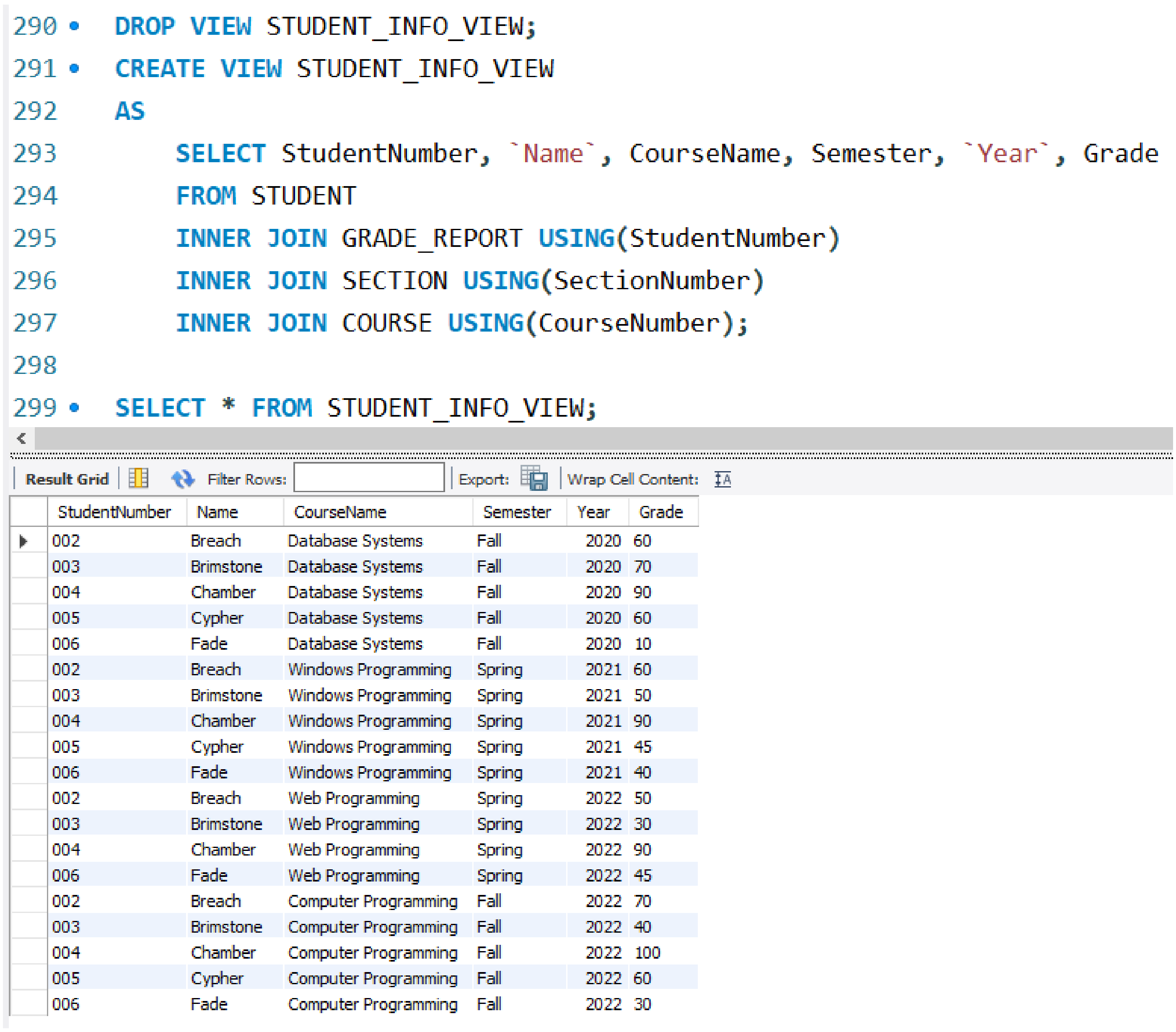
1(k)



```sql
SELECT Major, COUNT(*)
FROM (
    SELECT Major, AVG(Grade) AS AverageGrade
    FROM GRADE_REPORT AS GR
    INNER JOIN STUDENT AS ST USING(StudentNumber)
    GROUP BY Major, ST.StudentNumber HAVING AverageGrade < 60
) AVGTABLE
GROUP BY Major;
```

```
Major   COUNT(*)
EECS    2
CSIE    1
```

1(l)

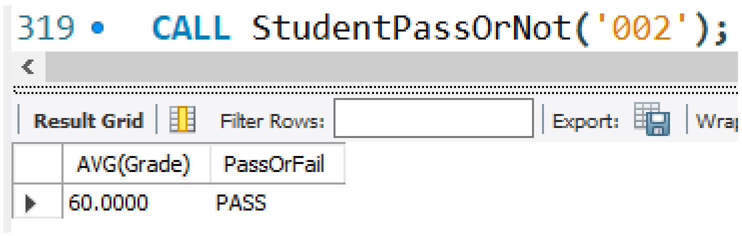


```
DROP VIEW STUDENT_INFO_VIEW;
CREATE VIEW STUDENT_INFO_VIEW
AS
    SELECT StudentNumber, `Name`, CourseName, Semester, `Year`, Grade
    FROM STUDENT
    INNER JOIN GRADE_REPORT USING(StudentNumber)
    INNER JOIN SECTION USING(SectionNumber)
    INNER JOIN COURSE USING(CourseNumber);

SELECT * FROM STUDENT_INFO_VIEW;
```

```
StudentNumber  Name       CourseName            Semester  Year  Grade
002            Breach     Database Systems      Fall      2020  60
003            Brimstone  Database Systems      Fall      2020  70
004            Chamber    Database Systems      Fall      2020  90
005            Cypher     Database Systems      Fall      2020  60
006            Fade       Database Systems      Fall      2020  10
002            Breach     Windows Programming   Spring    2021  60
003            Brimstone  Windows Programming   Spring    2021  50
004            Chamber    Windows Programming   Spring    2021  90
005            Cypher     Windows Programming   Spring    2021  45
006            Fade       Windows Programming   Spring    2021  40
002            Breach     Web Programming       Spring    2022  50
003            Brimstone  Web Programming       Spring    2022  30
004            Chamber    Web Programming       Spring    2022  90
006            Fade       Web Programming       Spring    2022  45
002            Breach     Computer Programming  Fall      2022  70
003            Brimstone  Computer Programming  Fall      2022  40
004            Chamber    Computer Programming  Fall      2022  100
005            Cypher     Computer Programming  Fall      2022  60
006            Fade       Computer Programming  Fall      2022  30
```

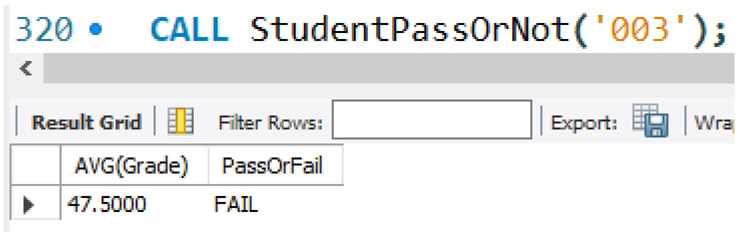
Q2(d).

```
DELIMITER $$
CREATE PROCEDURE StudentPassOrNot(IN `StudentNumber` VARCHAR(63))
BEGIN
    SELECT AVG(Grade),
    CASE
        WHEN AVG(Grade) >= 60 THEN "PASS"
        WHEN AVG(Grade) < 60 THEN "FAIL"
    END AS PassOrFail
    FROM STUDENT ST
    INNER JOIN GRADE_REPORT G USING(StudentNumber)
    WHERE `StudentNumber` = ST.StudentNumber;
END$$
DELIMITER ;
```



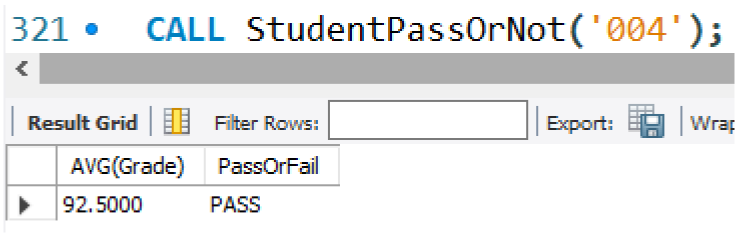
```sql
CALL StudentPassOrNot('002');
```

```
AVG(Grade)  PassOrFail
60.0000     PASS
```



```sql
CALL StudentPassOrNot('003');
```

```
AVG(Grade)  PassOrFail
47.5000     FAIL
```



```sql
CALL StudentPassOrNot('004');
```

```
AVG(Grade)  PassOrFail
92.5000     PASS
```