資工三 110590029 陳思群

1.

(a)

```
UPDATE COURSE
SET CreditHour = 2
WHERE CourseName = 'Object-Oriented Programming' AND Department = 'EECS';
```

(b)

```
DELETE FROM STUDENT
WHERE Name = 'David' AND StudentNumber = '005';
```

(c)

```
SELECT DISTINCT C.CourseName
FROM COURSE C
JOIN SECTION S ON C.CourseNumber = S.CourseNumber
WHERE S.Instructor = 'John' AND S.Year IN (2022, 2023);
```

(d)

```
SELECT S.CourseNumber, S.Semester, S.Year, COUNT(G.StudentNumber) AS NumberOfStudents
FROM SECTION S
JOIN GRADE_REPORT G ON S.SectionNumber = G.SectionNumber
WHERE S.Instructor = 'Eric'
GROUP BY S.SectionNumber, S.CourseNumber, S.Semester, S.Year;
```

(e)

```
SELECT P.PrerequisiteCourseNumber, C.CourseName
FROM PREREQUISITE P
JOIN COURSE C ON P.PrerequisiteCourseNumber = C.CourseNumber
WHERE C.CourseName = 'Database Systems' AND C.Department = 'EECS';
```

(f)

```
SELECT S.Name, C.CourseNumber, C.CourseName, C.CreditHour, SECT.Semester, SECT.Year, G.Grade
FROM STUDENT S
JOIN GRADE_REPORT G ON S.StudentNumber = G.StudentNumber
JOIN SECTION SECT ON G.SectionNumber = SECT.SectionNumber
JOIN COURSE C ON SECT.CourseNumber = C.CourseNumber
WHERE S.Class = 3 AND S.Major = 'EECS';
```

(g)

```
SELECT DISTINCT S.Name
FROM STUDENT S
WHERE NOT EXISTS (
    SELECT *
    FROM GRADE_REPORT G
    WHERE S.StudentNumber = G.StudentNumber AND G.Grade < 80
);
```

(h)

```
SELECT S.Name, S.Major
FROM STUDENT S
WHERE NOT EXISTS (
    SELECT *
    FROM GRADE_REPORT G
    WHERE S.StudentNumber = G.StudentNumber AND G.Grade < 60
);
```

(i)

```
SELECT DISTINCT S.StudentNumber, S.Name, S.Major
FROM STUDENT S
JOIN GRADE_REPORT G ON S.StudentNumber = G.StudentNumber
WHERE G.Grade < 60
ORDER BY S.StudentNumber;
```

(j)

```
SELECT S.Name, AVG(G.Grade) AS AverageGrade
FROM STUDENT S
JOIN GRADE_REPORT G ON S.StudentNumber = G.StudentNumber
JOIN SECTION SECT ON G.SectionNumber = SECT.SectionNumber
WHERE SECT.Year = 2023
GROUP BY S.StudentNumber, S.Name
HAVING AVG(G.Grade) > 80.0;
```

資工三 110590029 陳思群

(k)

```
SELECT S.Major, COUNT(*) AS NumberOfStudents
FROM STUDENT S
JOIN GRADE_REPORT G ON S.StudentNumber = G.StudentNumber
GROUP BY S.Major
HAVING AVG(G.Grade) < 60.0;
```

(l)

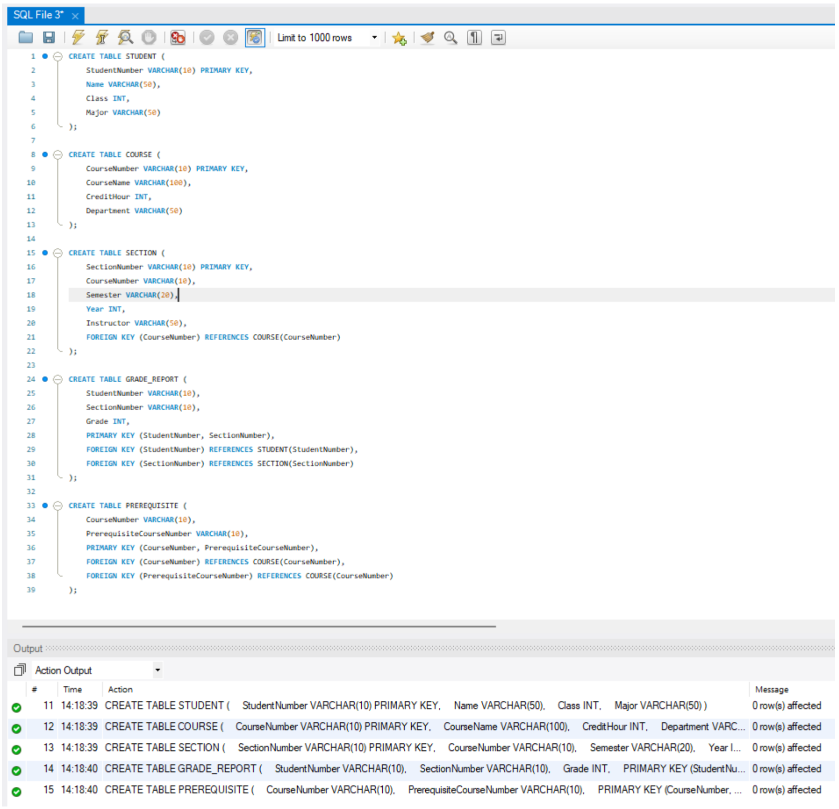
```
CREATE VIEW StudentCourseView AS
SELECT
    S.StudentNumber,
    S.Name AS StudentName,
    C.CourseName,
    SECT.Semester,
    SECT.Year,
    G.Grade
FROM STUDENT S
JOIN GRADE_REPORT G ON S.StudentNumber = G.StudentNumber
JOIN SECTION SECT ON G.SectionNumber = SECT.SectionNumber
JOIN COURSE C ON SECT.CourseNumber = C.CourseNumber;
```

```
SELECT * FROM StudentCourseView
```

資工三 110590029 陳思群

2.

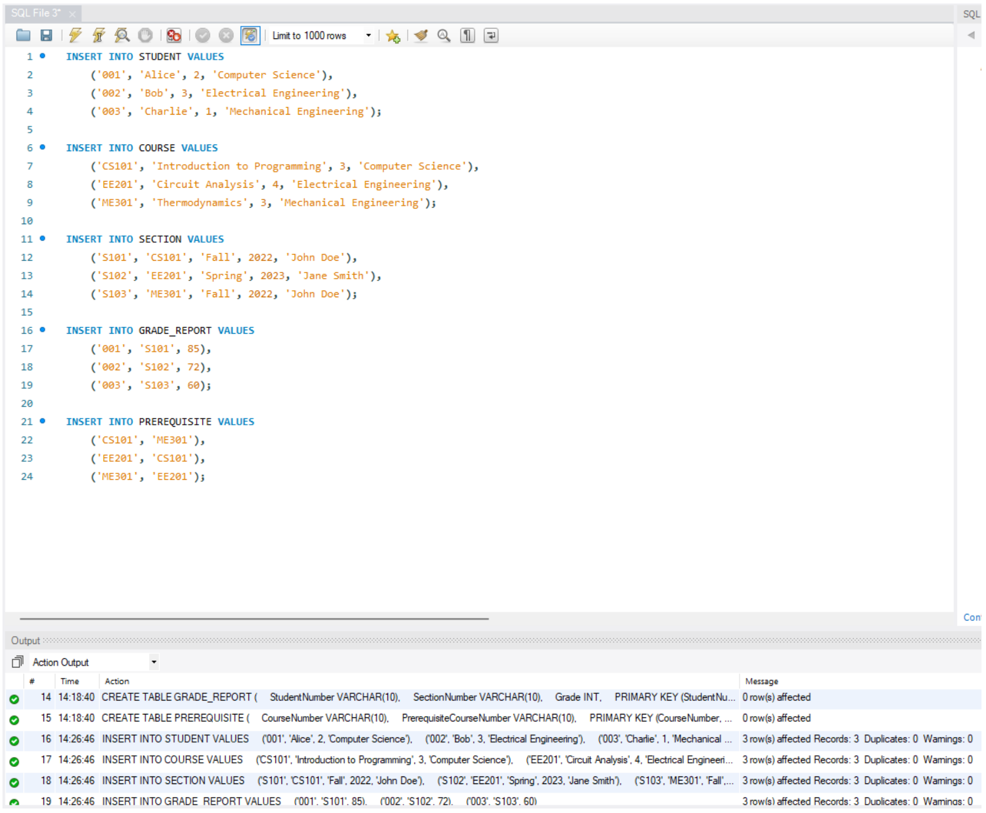
(a)



```sql
CREATE TABLE STUDENT (
    StudentNumber VARCHAR(10) PRIMARY KEY,
    Name VARCHAR(50),
    Class INT,
    Major VARCHAR(50)
);

CREATE TABLE COURSE (
    CourseNumber VARCHAR(10) PRIMARY KEY,
    CourseName VARCHAR(100),
    CreditHour INT,
    Department VARCHAR(50)
);

CREATE TABLE SECTION (
    SectionNumber VARCHAR(10) PRIMARY KEY,
    CourseNumber VARCHAR(10),
    Semester VARCHAR(20),
    Year INT,
    Instructor VARCHAR(50),
    FOREIGN KEY (CourseNumber) REFERENCES COURSE(CourseNumber)
);

CREATE TABLE GRADE_REPORT (
    StudentNumber VARCHAR(10),
    SectionNumber VARCHAR(10),
    Grade INT,
    PRIMARY KEY (StudentNumber, SectionNumber),
    FOREIGN KEY (StudentNumber) REFERENCES STUDENT(StudentNumber),
    FOREIGN KEY (SectionNumber) REFERENCES SECTION(SectionNumber)
);

CREATE TABLE PREREQUISITE (
    CourseNumber VARCHAR(10),
    PrerequisiteCourseNumber VARCHAR(10),
    PRIMARY KEY (CourseNumber, PrerequisiteCourseNumber),
    FOREIGN KEY (CourseNumber) REFERENCES COURSE(CourseNumber),
    FOREIGN KEY (PrerequisiteCourseNumber) REFERENCES COURSE(CourseNumber)
);
```

```
 #   Time      Action                                                                                                                      Message
 11  14:18:39  CREATE TABLE STUDENT (    StudentNumber VARCHAR(10) PRIMARY KEY,    Name VARCHAR(50),    Class INT,    Major VARCHAR(50) )     0 row(s) affected
 12  14:18:39  CREATE TABLE COURSE (    CourseNumber VARCHAR(10) PRIMARY KEY,    CourseName VARCHAR(100),    CreditHour INT,    Department VARC...  0 row(s) affected
 13  14:18:39  CREATE TABLE SECTION (    SectionNumber VARCHAR(10) PRIMARY KEY,    CourseNumber VARCHAR(10),    Semester VARCHAR(20),    Year I...  0 row(s) affected
 14  14:18:40  CREATE TABLE GRADE_REPORT (    StudentNumber VARCHAR(10),    SectionNumber VARCHAR(10),    Grade INT,    PRIMARY KEY (StudentNu...  0 row(s) affected
 15  14:18:40  CREATE TABLE PREREQUISITE (    CourseNumber VARCHAR(10),    PrerequisiteCourseNumber VARCHAR(10),    PRIMARY KEY (CourseNumber, ...  0 row(s) affected
```

(b)



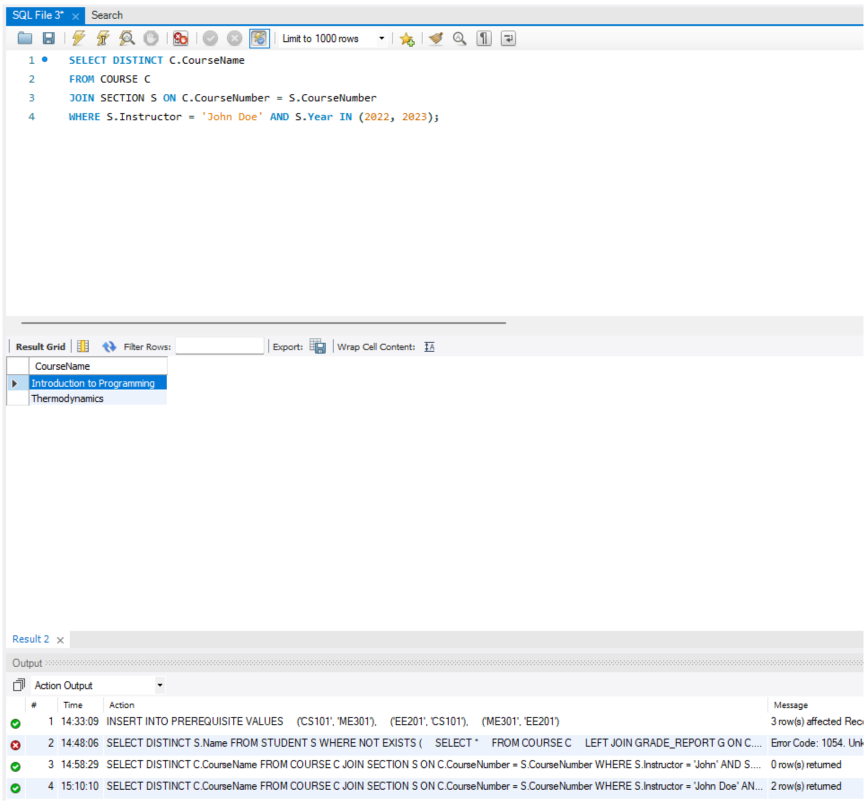
```sql
INSERT INTO STUDENT VALUES
    ('001', 'Alice', 2, 'Computer Science'),
    ('002', 'Bob', 3, 'Electrical Engineering'),
    ('003', 'Charlie', 1, 'Mechanical Engineering');

INSERT INTO COURSE VALUES
    ('CS101', 'Introduction to Programming', 3, 'Computer Science'),
    ('EE201', 'Circuit Analysis', 4, 'Electrical Engineering'),
    ('ME301', 'Thermodynamics', 3, 'Mechanical Engineering');

INSERT INTO SECTION VALUES
    ('S101', 'CS101', 'Fall', 2022, 'John Doe'),
    ('S102', 'EE201', 'Spring', 2023, 'Jane Smith'),
    ('S103', 'ME301', 'Fall', 2022, 'John Doe');

INSERT INTO GRADE_REPORT VALUES
    ('001', 'S101', 85),
    ('002', 'S102', 72),
    ('003', 'S103', 60);

INSERT INTO PREREQUISITE VALUES
    ('CS101', 'ME301'),
    ('EE201', 'CS101'),
    ('ME301', 'EE201');
```

```
 #   Time      Action                                                                                                                                  Message
 14  14:18:40  CREATE TABLE GRADE_REPORT (    StudentNumber VARCHAR(10),    SectionNumber VARCHAR(10),    Grade INT,    PRIMARY KEY (StudentNu...      0 row(s) affected
 15  14:18:40  CREATE TABLE PREREQUISITE (    CourseNumber VARCHAR(10),    PrerequisiteCourseNumber VARCHAR(10),    PRIMARY KEY (CourseNumber, ...     0 row(s) affected
 16  14:26:46  INSERT INTO STUDENT VALUES    ('001', 'Alice', 2, 'Computer Science'),    ('002', 'Bob', 3, 'Electrical Engineering'),    ('003', 'Charlie', 1, 'Mechanical ...   3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0
 17  14:26:46  INSERT INTO COURSE VALUES    ('CS101', 'Introduction to Programming', 3, 'Computer Science'),    ('EE201', 'Circuit Analysis', 4, 'Electrical Engineeri...   3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0
 18  14:26:46  INSERT INTO SECTION VALUES    ('S101', 'CS101', 'Fall', 2022, 'John Doe'),    ('S102', 'EE201', 'Spring', 2023, 'Jane Smith'),    ('S103', 'ME301', 'Fall',...   3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0
 19  14:26:46  INSERT INTO GRADE_REPORT VALUES    ('001', 'S101', 85),    ('002', 'S102', 72),    ('003', 'S103', 60)   3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0
```

資工三 110590029 陳思群

(c)

1.(c)

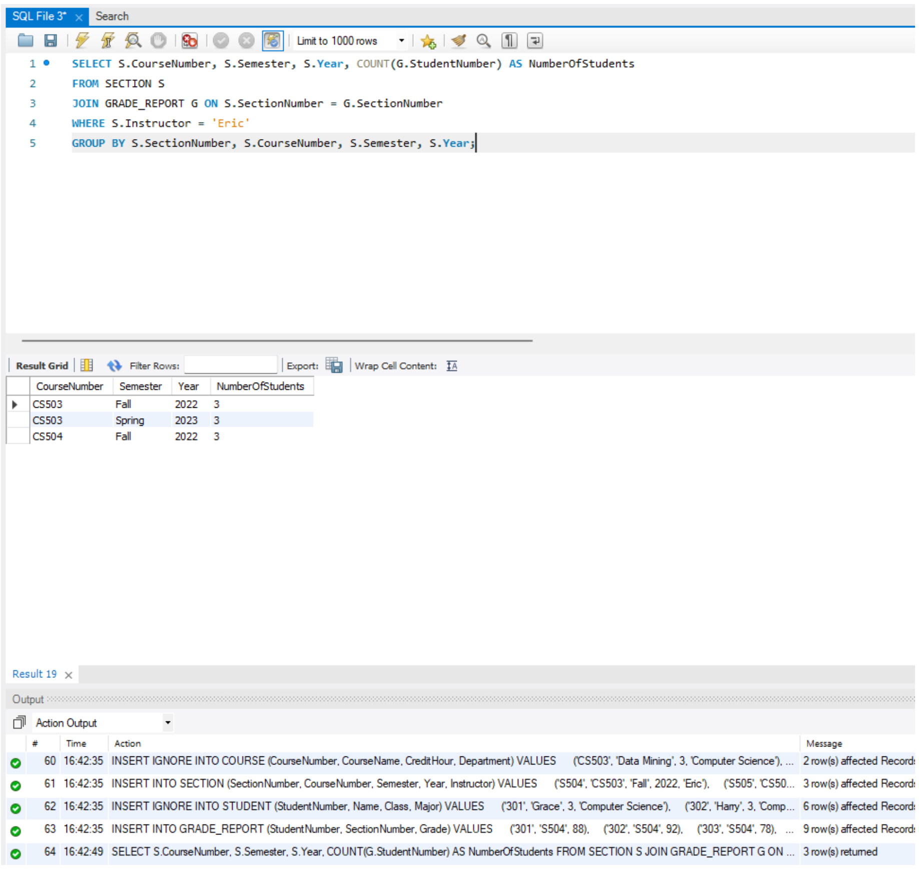


```sql
SELECT DISTINCT C.CourseName
FROM COURSE C
JOIN SECTION S ON C.CourseNumber = S.CourseNumber
WHERE S.Instructor = 'John Doe' AND S.Year IN (2022, 2023);
```

CourseName
Introduction to Programming
Thermodynamics

```
#   Time      Action                                                                                                            Message
1   14:33:09  INSERT INTO PREREQUISITE VALUES    ('CS101', 'ME301'),    ('EE201', 'CS101'),    ('ME301', 'EE201')               3 row(s) affected Rec
2   14:48:06  SELECT DISTINCT S.Name FROM STUDENT S WHERE NOT EXISTS (    SELECT *    FROM COURSE C    LEFT JOIN GRADE_REPORT G ON C....  Error Code: 1054. Unk
3   14:58:29  SELECT DISTINCT C.CourseName FROM COURSE C JOIN SECTION S ON C.CourseNumber = S.CourseNumber WHERE S.Instructor = 'John' AND S....  0 row(s) returned
4   15:10:10  SELECT DISTINCT C.CourseName FROM COURSE C JOIN SECTION S ON C.CourseNumber = S.CourseNumber WHERE S.Instructor = 'John Doe' AN...  2 row(s) returned
```

1.(d)



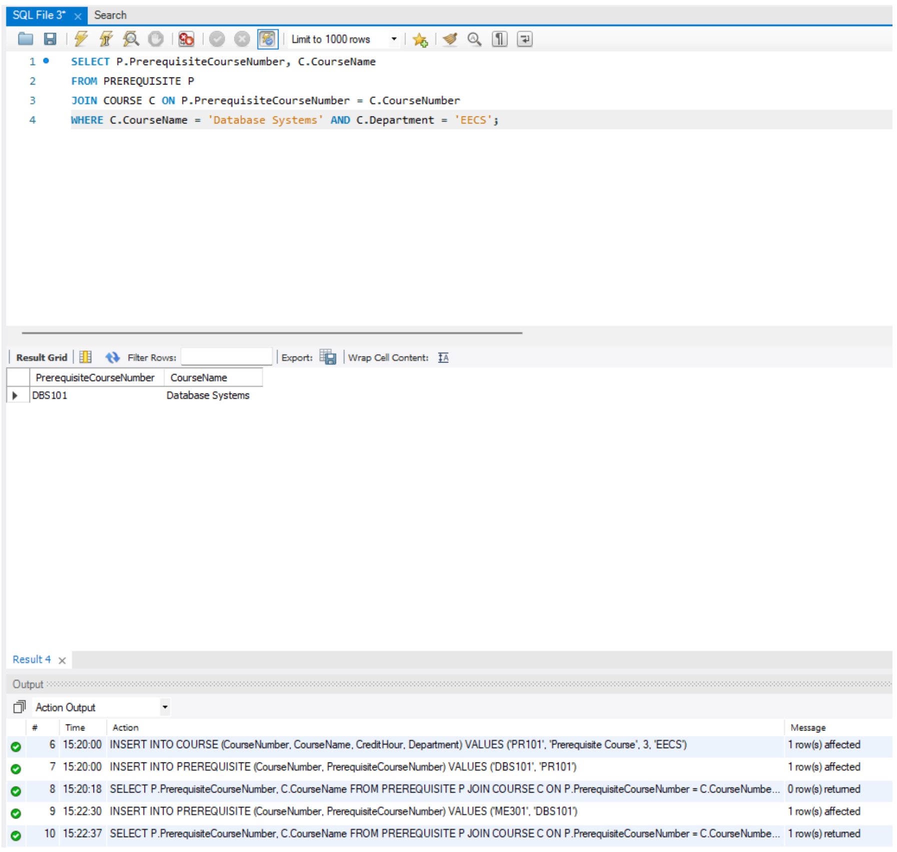
```sql
SELECT S.CourseNumber, S.Semester, S.Year, COUNT(G.StudentNumber) AS NumberOfStudents
FROM SECTION S
JOIN GRADE_REPORT G ON S.SectionNumber = G.SectionNumber
WHERE S.Instructor = 'Eric'
GROUP BY S.SectionNumber, S.CourseNumber, S.Semester, S.Year;
```

CourseNumber	Semester	Year	NumberOfStudents
CS503	Fall	2022	3
CS503	Spring	2023	3
CS504	Fall	2022	3

```
#   Time      Action                                                                                                                 Message
60  16:42:35  INSERT IGNORE INTO COURSE (CourseNumber, CourseName, CreditHour, Department) VALUES    ('CS503', 'Data Mining', 3, 'Computer Science'), ...  2 row(s) affected Record
61  16:42:35  INSERT INTO SECTION (SectionNumber, CourseNumber, Semester, Year, Instructor) VALUES    ('S504', 'CS503', 'Fall', 2022, 'Eric'),    ('S505', 'CS50...  3 row(s) affected Record
62  16:42:35  INSERT IGNORE INTO STUDENT (StudentNumber, Name, Class, Major) VALUES    ('301', 'Grace', 3, 'Computer Science'),    ('302', 'Harry', 3, 'Comp...  6 row(s) affected Record
63  16:42:35  INSERT INTO GRADE_REPORT (StudentNumber, SectionNumber, Grade) VALUES    ('301', 'S504', 88),    ('302', 'S504', 92),    ('303', 'S504', 78),  ...  9 row(s) affected Record
64  16:42:49  SELECT S.CourseNumber, S.Semester, S.Year, COUNT(G.StudentNumber) AS NumberOfStudents FROM SECTION S JOIN GRADE_REPORT G ON ...  3 row(s) returned
```

資工三 110590029 陳思群

1.(e)



SQL File 3* × Search

Limit to 1000 rows

```sql
SELECT P.PrerequisiteCourseNumber, C.CourseName
FROM PREREQUISITE P
JOIN COURSE C ON P.PrerequisiteCourseNumber = C.CourseNumber
WHERE C.CourseName = 'Database Systems' AND C.Department = 'EECS';
```

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

PrerequisiteCourseNumber	CourseName
DBS101	Database Systems

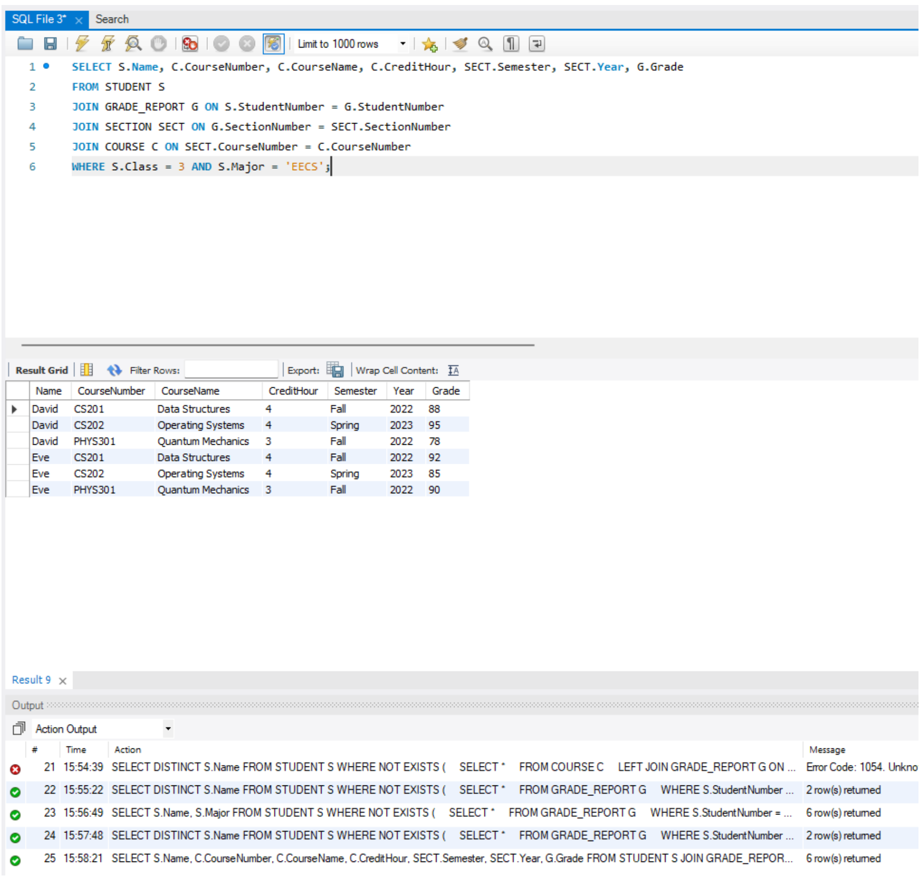
Result 4 ×

Output

Action Output

```
#   Time      Action                                                                                                                  Message
6   15:20:00  INSERT INTO COURSE (CourseNumber, CourseName, CreditHour, Department) VALUES ('PR101', 'Prerequisite Course', 3, 'EECS')  1 row(s) affected
7   15:20:00  INSERT INTO PREREQUISITE (CourseNumber, PrerequisiteCourseNumber) VALUES ('DBS101', 'PR101')                              1 row(s) affected
8   15:20:18  SELECT P.PrerequisiteCourseNumber, C.CourseName FROM PREREQUISITE P JOIN COURSE C ON P.PrerequisiteCourseNumber = C.CourseNumbe...  0 row(s) returned
9   15:22:30  INSERT INTO PREREQUISITE (CourseNumber, PrerequisiteCourseNumber) VALUES ('ME301', 'DBS101')                              1 row(s) affected
10  15:22:37  SELECT P.PrerequisiteCourseNumber, C.CourseName FROM PREREQUISITE P JOIN COURSE C ON P.PrerequisiteCourseNumber = C.CourseNumbe...  1 row(s) returned
```

1.(f)



SQL File 3* × Search

Limit to 1000 rows

```sql
SELECT S.Name, C.CourseNumber, C.CourseName, C.CreditHour, SECT.Semester, SECT.Year, G.Grade
FROM STUDENT S
JOIN GRADE_REPORT G ON S.StudentNumber = G.StudentNumber
JOIN SECTION SECT ON G.SectionNumber = SECT.SectionNumber
JOIN COURSE C ON SECT.CourseNumber = C.CourseNumber
WHERE S.Class = 3 AND S.Major = 'EECS';
```

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

Name	CourseNumber	CourseName	CreditHour	Semester	Year	Grade
David	CS201	Data Structures	4	Fall	2022	88
David	CS202	Operating Systems	4	Spring	2023	95
David	PHYS301	Quantum Mechanics	3	Fall	2022	78
Eve	CS201	Data Structures	4	Fall	2022	92
Eve	CS202	Operating Systems	4	Spring	2023	85
Eve	PHYS301	Quantum Mechanics	3	Fall	2022	90

Result 9 ×

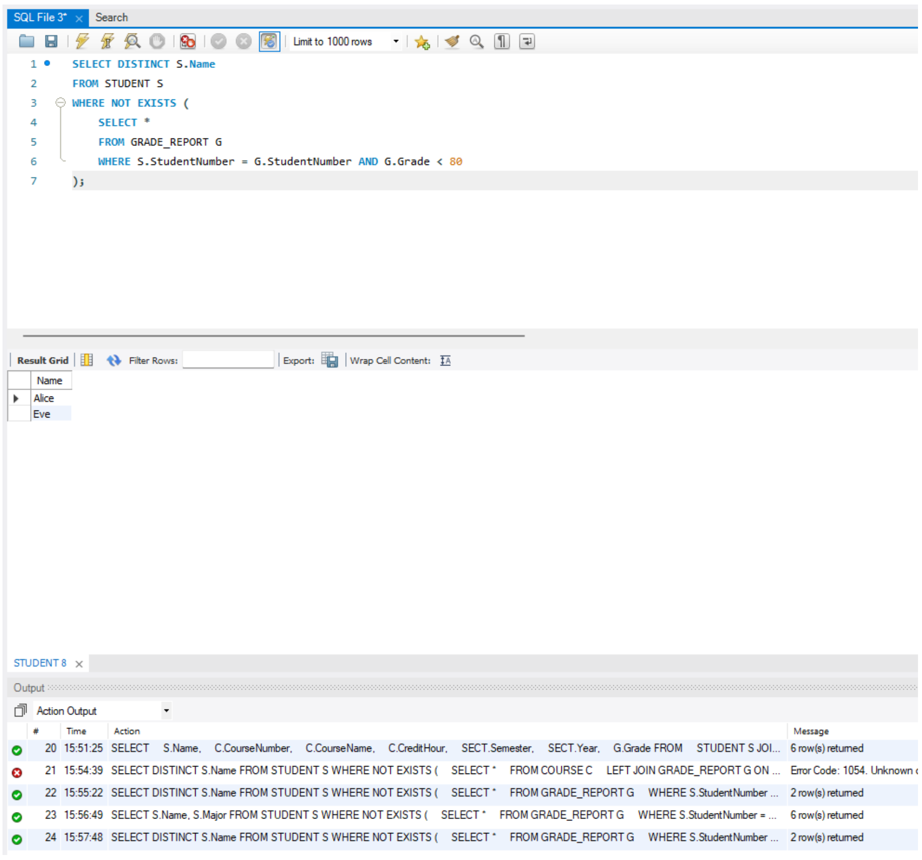
Output

Action Output

```
#   Time      Action                                                                                                          Message
21  15:54:39  SELECT DISTINCT S.Name FROM STUDENT S WHERE NOT EXISTS (    SELECT *    FROM COURSE C    LEFT JOIN GRADE_REPORT G ON ...  Error Code: 1054. Unkno
22  15:55:22  SELECT DISTINCT S.Name FROM STUDENT S WHERE NOT EXISTS (    SELECT *    FROM GRADE_REPORT G    WHERE S.StudentNumber ...  2 row(s) returned
23  15:56:49  SELECT S.Name, S.Major FROM STUDENT S WHERE NOT EXISTS (    SELECT *    FROM GRADE_REPORT G    WHERE S.StudentNumber = ...  6 row(s) returned
24  15:57:48  SELECT DISTINCT S.Name FROM STUDENT S WHERE NOT EXISTS (    SELECT *    FROM GRADE_REPORT G    WHERE S.StudentNumber ...  2 row(s) returned
25  15:58:21  SELECT S.Name, C.CourseNumber, C.CourseName, C.CreditHour, SECT.Semester, SECT.Year, G.Grade FROM STUDENT S JOIN GRADE_REPOR...  6 row(s) returned
```

資工三 110590029 陳思群

1.(g)



```sql
SELECT DISTINCT S.Name
FROM STUDENT S
WHERE NOT EXISTS (
    SELECT *
    FROM GRADE_REPORT G
    WHERE S.StudentNumber = G.StudentNumber AND G.Grade < 80
);
```

Name
Alice
Eve

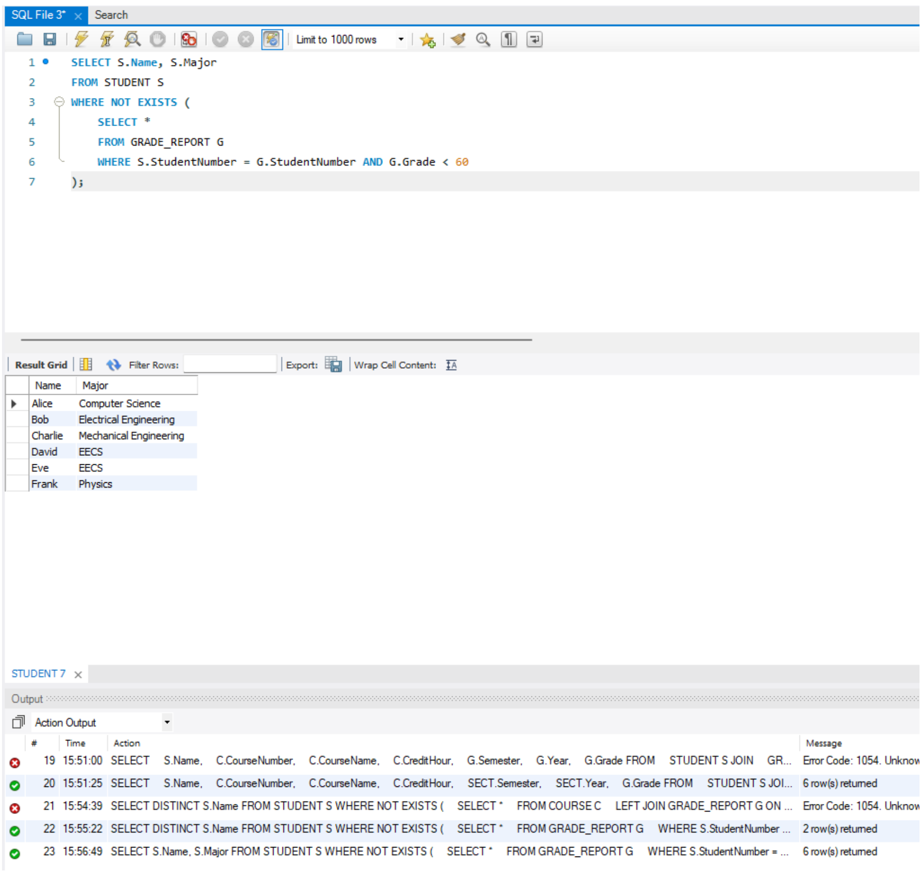
STUDENT 8

Output

Action Output

```
#   Time      Action                                                                                                                                      Message
20  15:51:25  SELECT   S.Name,  C.CourseNumber,  C.CourseName,  C.CreditHour,  SECT.Semester,  SECT.Year,  G.Grade FROM  STUDENT S JOI...                  6 row(s) returned
21  15:54:39  SELECT DISTINCT S.Name FROM STUDENT S WHERE NOT EXISTS (  SELECT *  FROM COURSE C  LEFT JOIN GRADE_REPORT G ON ...                         Error Code: 1054. Unknown c
22  15:55:22  SELECT DISTINCT S.Name FROM STUDENT S WHERE NOT EXISTS (  SELECT *  FROM GRADE_REPORT G  WHERE S.StudentNumber ...                          2 row(s) returned
23  15:56:49  SELECT S.Name, S.Major FROM STUDENT S WHERE NOT EXISTS (  SELECT *  FROM GRADE_REPORT G  WHERE S.StudentNumber = ...                        6 row(s) returned
24  15:57:48  SELECT DISTINCT S.Name FROM STUDENT S WHERE NOT EXISTS (  SELECT *  FROM GRADE_REPORT G  WHERE S.StudentNumber ...                          2 row(s) returned
```

1.(h)



```sql
SELECT S.Name, S.Major
FROM STUDENT S
WHERE NOT EXISTS (
    SELECT *
    FROM GRADE_REPORT G
    WHERE S.StudentNumber = G.StudentNumber AND G.Grade < 60
);
```

Name	Major
Alice	Computer Science
Bob	Electrical Engineering
Charlie	Mechanical Engineering
David	EECS
Eve	EECS
Frank	Physics

STUDENT 7

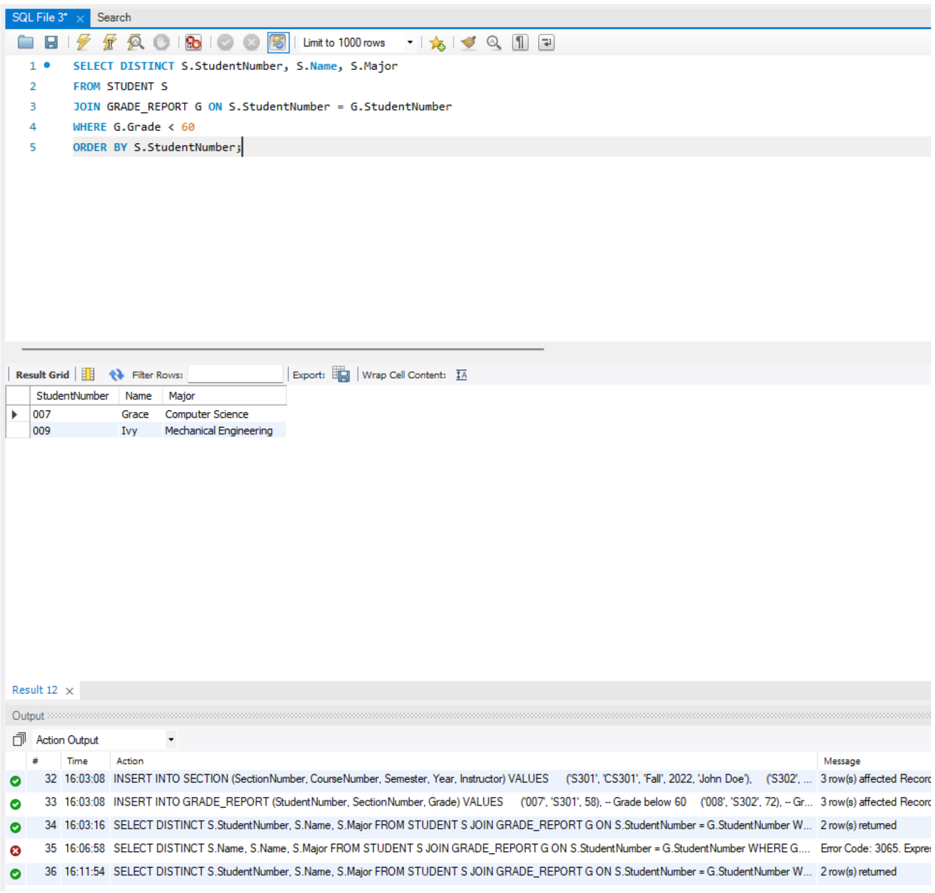
Output

Action Output

```
#   Time      Action                                                                                                                                      Message
19  15:51:00  SELECT   S.Name,  C.CourseNumber,  C.CourseName,  C.CreditHour,  G.Semester,  G.Year,  G.Grade FROM  STUDENT S JOIN  GR...                   Error Code: 1054. Unknow
20  15:51:25  SELECT   S.Name,  C.CourseNumber,  C.CourseName,  C.CreditHour,  SECT.Semester,  SECT.Year,  G.Grade FROM  STUDENT S JOI...                  6 row(s) returned
21  15:54:39  SELECT DISTINCT S.Name FROM STUDENT S WHERE NOT EXISTS (  SELECT *  FROM COURSE C  LEFT JOIN GRADE_REPORT G ON ...                         Error Code: 1054. Unknow
22  15:55:22  SELECT DISTINCT S.Name FROM STUDENT S WHERE NOT EXISTS (  SELECT *  FROM GRADE_REPORT G  WHERE S.StudentNumber ...                          2 row(s) returned
23  15:56:49  SELECT S.Name, S.Major FROM STUDENT S WHERE NOT EXISTS (  SELECT *  FROM GRADE_REPORT G  WHERE S.StudentNumber = ...                        6 row(s) returned
```

資工三 110590029 陳思群

1.(i)



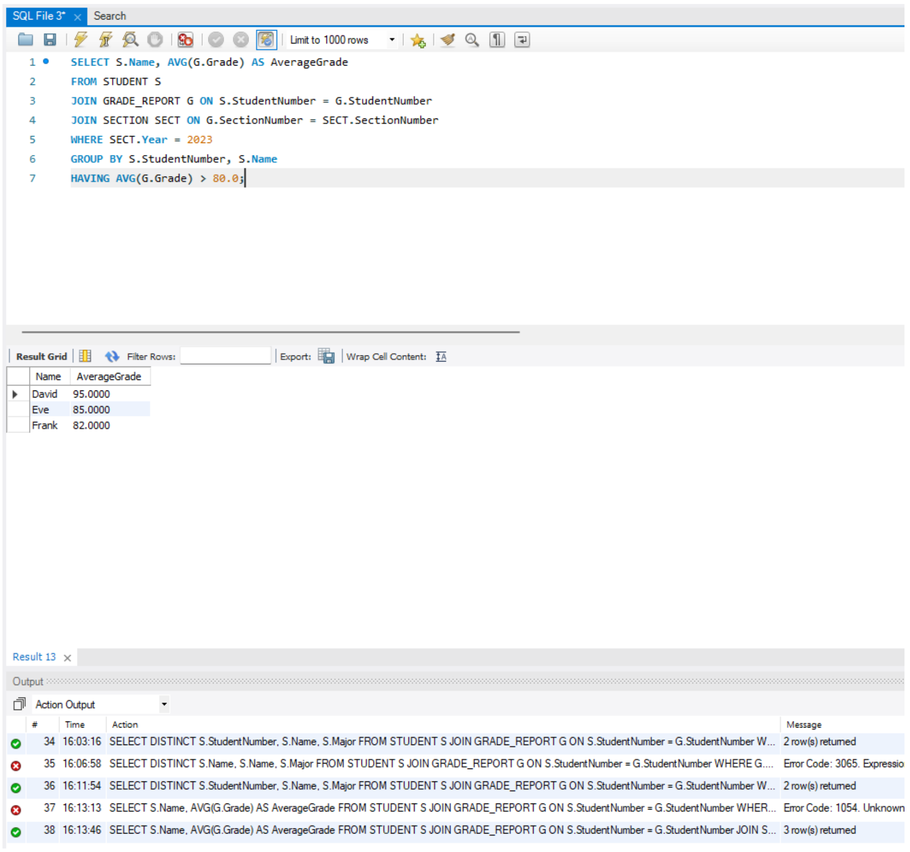
```sql
SELECT DISTINCT S.StudentNumber, S.Name, S.Major
FROM STUDENT S
JOIN GRADE_REPORT G ON S.StudentNumber = G.StudentNumber
WHERE G.Grade < 60
ORDER BY S.StudentNumber;
```

StudentNumber	Name	Major
007	Grace	Computer Science
009	Ivy	Mechanical Engineering

Result 12

```
#   Time      Action                                                                                                                      Message
32  16:03:08  INSERT INTO SECTION (SectionNumber, CourseNumber, Semester, Year, Instructor) VALUES    ('S301', 'CS301', 'Fall', 2022, 'John Doe'),    ('S302', ...  3 row(s) affected Record
33  16:03:08  INSERT INTO GRADE_REPORT (StudentNumber, SectionNumber, Grade) VALUES    ('007', 'S301', 58), -- Grade below 60    ('008', 'S302', 72), -- Gr...  3 row(s) affected Record
34  16:03:16  SELECT DISTINCT S.StudentNumber, S.Name, S.Major FROM STUDENT S JOIN GRADE_REPORT G ON S.StudentNumber = G.StudentNumber W...  2 row(s) returned
35  16:06:58  SELECT DISTINCT S.Name, S.Name, S.Major FROM STUDENT S JOIN GRADE_REPORT G ON S.StudentNumber = G.StudentNumber WHERE G....  Error Code: 3065. Expres
36  16:11:54  SELECT DISTINCT S.StudentNumber, S.Name, S.Major FROM STUDENT S JOIN GRADE_REPORT G ON S.StudentNumber = G.StudentNumber W...  2 row(s) returned
```

1.(j)



```sql
SELECT S.Name, AVG(G.Grade) AS AverageGrade
FROM STUDENT S
JOIN GRADE_REPORT G ON S.StudentNumber = G.StudentNumber
JOIN SECTION SECT ON G.SectionNumber = SECT.SectionNumber
WHERE SECT.Year = 2023
GROUP BY S.StudentNumber, S.Name
HAVING AVG(G.Grade) > 80.0;
```

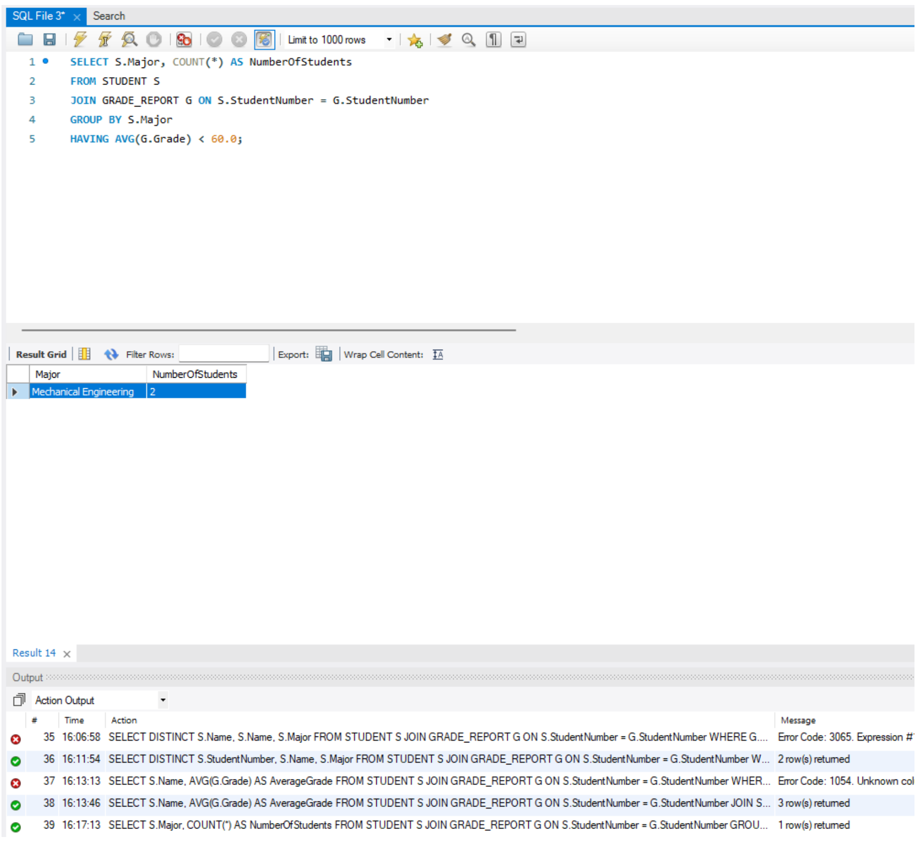
Name	AverageGrade
David	95.0000
Eve	85.0000
Frank	82.0000

Result 13

```
#   Time      Action                                                                                                                      Message
34  16:03:16  SELECT DISTINCT S.StudentNumber, S.Name, S.Major FROM STUDENT S JOIN GRADE_REPORT G ON S.StudentNumber = G.StudentNumber W...  2 row(s) returned
35  16:06:58  SELECT DISTINCT S.Name, S.Name, S.Major FROM STUDENT S JOIN GRADE_REPORT G ON S.StudentNumber = G.StudentNumber WHERE G....  Error Code: 3065. Expressio
36  16:11:54  SELECT DISTINCT S.StudentNumber, S.Name, S.Major FROM STUDENT S JOIN GRADE_REPORT G ON S.StudentNumber = G.StudentNumber W...  2 row(s) returned
37  16:13:13  SELECT S.Name, AVG(G.Grade) AS AverageGrade FROM STUDENT S JOIN GRADE_REPORT G ON S.StudentNumber = G.StudentNumber WHER...  Error Code: 1054. Unknown
38  16:13:46  SELECT S.Name, AVG(G.Grade) AS AverageGrade FROM STUDENT S JOIN GRADE_REPORT G ON S.StudentNumber = G.StudentNumber JOIN S...  3 row(s) returned
```

資工三 110590029 陳思群

1.(k)



```
SELECT S.Major, COUNT(*) AS NumberOfStudents
FROM STUDENT S
JOIN GRADE_REPORT G ON S.StudentNumber = G.StudentNumber
GROUP BY S.Major
HAVING AVG(G.Grade) < 60.0;
```

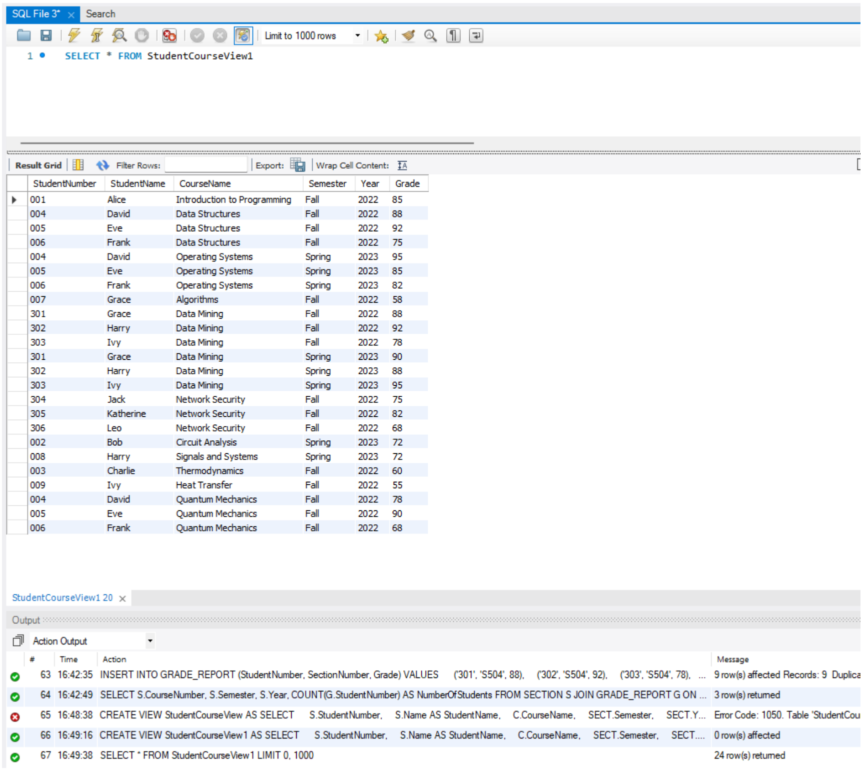
Major	NumberOfStudents
Mechanical Engineering	2

Result 14

```
#   Time      Action                                                                                                                        Message
35  16:06:58  SELECT DISTINCT S.Name, S.Name, S.Major FROM STUDENT S JOIN GRADE_REPORT G ON S.StudentNumber = G.StudentNumber WHERE G....    Error Code: 3065. Expression #
36  16:11:54  SELECT DISTINCT S.StudentNumber, S.Name, S.Major FROM STUDENT S JOIN GRADE_REPORT G ON S.StudentNumber = G.StudentNumber W...  2 row(s) returned
37  16:13:13  SELECT S.Name, AVG(G.Grade) AS AverageGrade FROM STUDENT S JOIN GRADE_REPORT G ON S.StudentNumber = G.StudentNumber WHER...     Error Code: 1054. Unknown col
38  16:13:46  SELECT S.Name, AVG(G.Grade) AS AverageGrade FROM STUDENT S JOIN GRADE_REPORT G ON S.StudentNumber = G.StudentNumber JOIN S...   3 row(s) returned
39  16:17:13  SELECT S.Major, COUNT(*) AS NumberOfStudents FROM STUDENT S JOIN GRADE_REPORT G ON S.StudentNumber = G.StudentNumber GROU...    1 row(s) returned
```

1.(l)

```
CREATE VIEW StudentCourseView AS
SELECT
    S.StudentNumber,
    S.Name AS StudentName,
    C.CourseName,
    SECT.Semester,
    SECT.Year,
    G.Grade
FROM STUDENT S
JOIN GRADE_REPORT G ON S.StudentNumber = G.StudentNumber
JOIN SECTION SECT ON G.SectionNumber = SECT.SectionNumber
JOIN COURSE C ON SECT.CourseNumber = C.CourseNumber;
```



```
SELECT * FROM StudentCourseView1
```

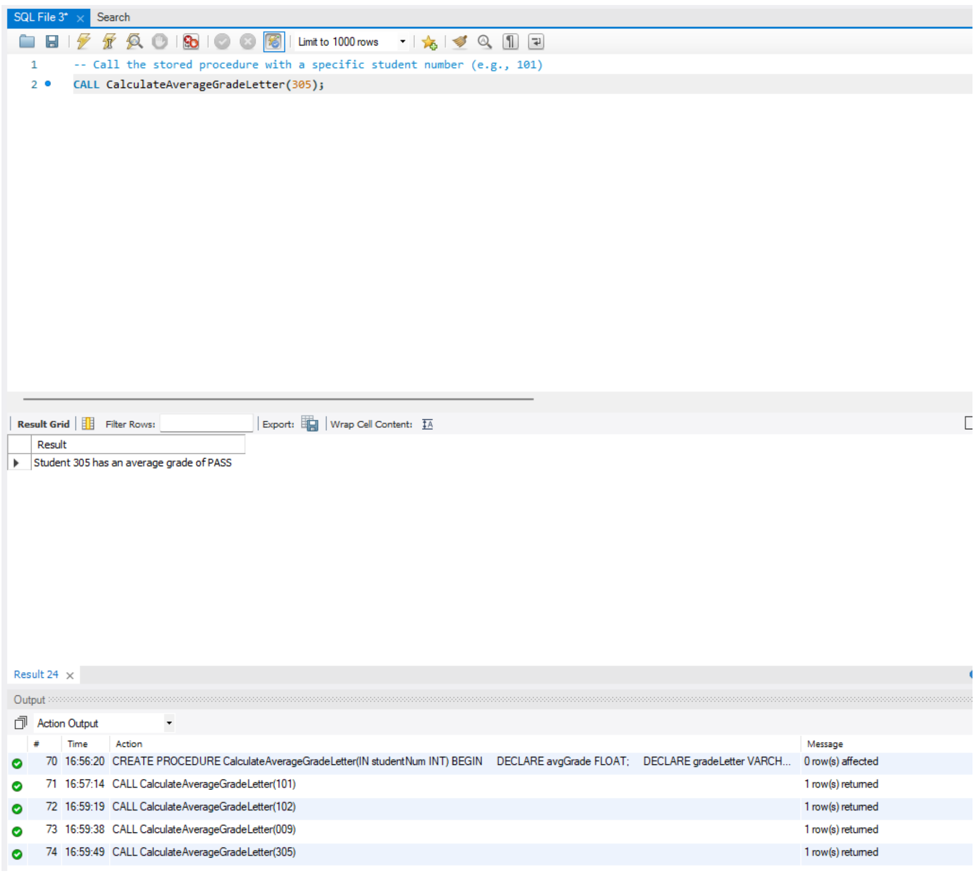
StudentNumber	StudentName	CourseName	Semester	Year	Grade
001	Alice	Introduction to Programming	Fall	2022	85
004	David	Data Structures	Fall	2022	88
005	Eve	Data Structures	Fall	2022	92
006	Frank	Data Structures	Fall	2022	75
004	David	Operating Systems	Spring	2023	95
005	Eve	Operating Systems	Spring	2023	85
006	Frank	Operating Systems	Spring	2023	82
007	Grace	Algorithms	Fall	2022	58
301	Grace	Data Mining	Fall	2022	88
302	Harry	Data Mining	Fall	2022	92
303	Ivy	Data Mining	Fall	2022	78
301	Grace	Data Mining	Spring	2023	90
302	Harry	Data Mining	Spring	2023	88
303	Ivy	Data Mining	Spring	2023	95
304	Jack	Network Security	Fall	2022	75
305	Katherine	Network Security	Fall	2022	82
306	Leo	Network Security	Fall	2022	68
002	Bob	Circuit Analysis	Spring	2023	72
008	Harry	Signals and Systems	Spring	2023	72
003	Charlie	Thermodynamics	Fall	2022	60
009	Ivy	Heat Transfer	Fall	2022	55
004	David	Quantum Mechanics	Fall	2022	78
005	Eve	Quantum Mechanics	Fall	2022	90
006	Frank	Quantum Mechanics	Fall	2022	68

StudentCourseView1 20

```
#   Time      Action                                                                                                                                 Message
63  16:42:35  INSERT INTO GRADE_REPORT (StudentNumber, SectionNumber, Grade) VALUES    ('301', 'S504', 88),    ('302', 'S504', 92),    ('303', 'S504', 78),  ...   9 row(s) affected Records: 9  Duplica
64  16:42:49  SELECT S.CourseNumber, S.Semester, S.Year, COUNT(G.StudentNumber) AS NumberOfStudents FROM SECTION S JOIN GRADE_REPORT G ON ...    3 row(s) returned
65  16:48:38  CREATE VIEW StudentCourseView AS SELECT    S.StudentNumber,    S.Name AS StudentName,    C.CourseName,    SECT.Semester,    SECT.Y...   Error Code: 1050. Table 'StudentCou
66  16:49:16  CREATE VIEW StudentCourseView1 AS SELECT    S.StudentNumber,    S.Name AS StudentName,    C.CourseName,    SECT.Semester,    SECT....  0 row(s) affected
67  16:49:38  SELECT * FROM StudentCourseView1 LIMIT 0, 1000                                                                                         24 row(s) returned
```

(d)

```
DELIMITER //

CREATE PROCEDURE CalculateAverageGradeLetter(IN studentNum INT)
BEGIN
    DECLARE avgGrade FLOAT;
    DECLARE gradeLetter VARCHAR(10);

    -- Calculate average grade for the student
    SELECT AVG(Grade) INTO avgGrade
    FROM GRADE_REPORT
    WHERE StudentNumber = studentNum;

    -- Determine the grade letter
    IF avgGrade >= 60 THEN
        SET gradeLetter = 'PASS';
    ELSE
        SET gradeLetter = 'FAIL';
    END IF;

    -- Print the result
    SELECT CONCAT('Student ', studentNum, ' has an average grade of ',
    gradeLetter) AS Result;

END //

DELIMITER ;
```



```
-- Call the stored procedure with a specific student number (e.g., 101)
CALL CalculateAverageGradeLetter(305);
```

Result
Student 305 has an average grade of PASS

	#	Time	Action	Message
✓	70	16:56:20	CREATE PROCEDURE CalculateAverageGradeLetter(IN studentNum INT) BEGIN DECLARE avgGrade FLOAT; DECLARE gradeLetter VARCH...	0 row(s) affected
✓	71	16:57:14	CALL CalculateAverageGradeLetter(101)	1 row(s) returned
✓	72	16:59:19	CALL CalculateAverageGradeLetter(102)	1 row(s) returned
✓	73	16:59:38	CALL CalculateAverageGradeLetter(009)	1 row(s) returned
✓	74	16:59:49	CALL CalculateAverageGradeLetter(305)	1 row(s) returned