# COS60009 – Database Management System for the Big Data age

## Individual Assignment -1

Name: Arun Ragavendhar Arunachalam Palaniyappan

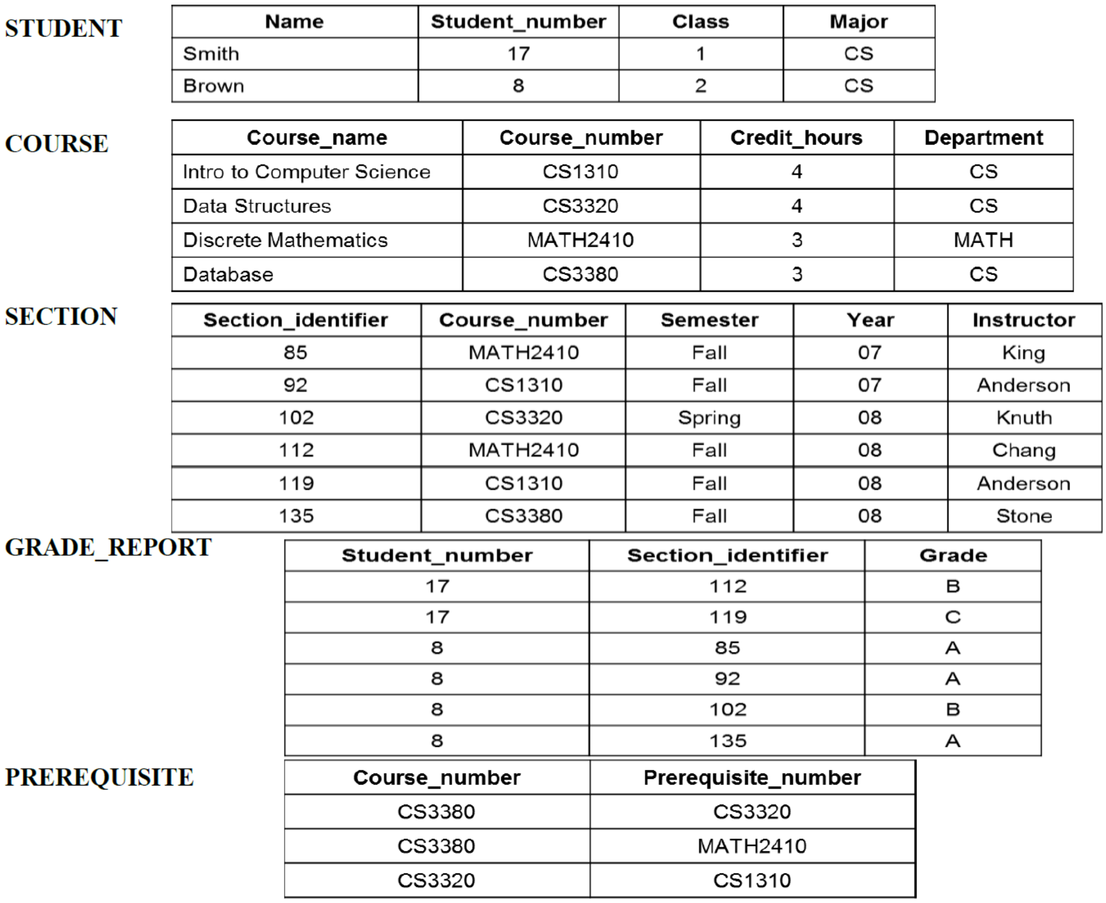
Student ID: 104837257

[Arun Ragavendhar]

[104837257]

Task A -

Specify the following queries in SQL on the database schema of Figure 1.2.
a. Retrieve the name of each course along with the name of the instructor who taught that course during the fall of 08.
b. For each section taught by Professor Anderson, retrieve the course number, semester, year, and number of students who took the section.
c. For each student who completed more than two courses, retrieve the name, student number of the student and the number of courses completed by that student.



**STUDENT**

| Name | Student_number | Class | Major |
|---|---|---|---|
| Smith | 17 | 1 | CS |
| Brown | 8 | 2 | CS |

**COURSE**

| Course_name | Course_number | Credit_hours | Department |
|---|---|---|---|
| Intro to Computer Science | CS1310 | 4 | CS |
| Data Structures | CS3320 | 4 | CS |
| Discrete Mathematics | MATH2410 | 3 | MATH |
| Database | CS3380 | 3 | CS |

**SECTION**

| Section_identifier | Course_number | Semester | Year | Instructor |
|---|---|---|---|---|
| 85 | MATH2410 | Fall | 07 | King |
| 92 | CS1310 | Fall | 07 | Anderson |
| 102 | CS3320 | Spring | 08 | Knuth |
| 112 | MATH2410 | Fall | 08 | Chang |
| 119 | CS1310 | Fall | 08 | Anderson |
| 135 | CS3380 | Fall | 08 | Stone |

**GRADE_REPORT**

| Student_number | Section_identifier | Grade |
|---|---|---|
| 17 | 112 | B |
| 17 | 119 | C |
| 8 | 85 | A |
| 8 | 92 | A |
| 8 | 102 | B |
| 8 | 135 | A |

**PREREQUISITE**

| Course_number | Prerequisite_number |
|---|---|
| CS3380 | CS3320 |
| CS3380 | MATH2410 |
| CS3320 | CS1310 |

**Figure 1.2 Example of a simple database**

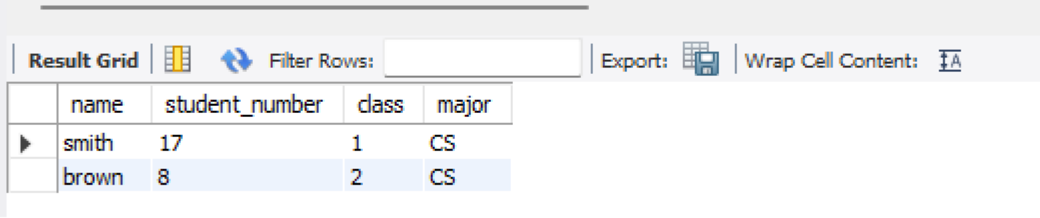
I understand that it was not an assignment requirement to submit the execution of the queries in SQL, but I have presented it out of own interest as well as for hands on practice.

[Arun Ragavendhar]

[104837257]

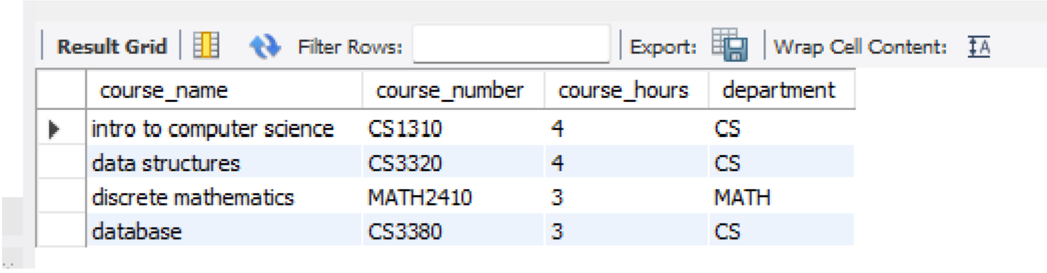
First, I created the respective table in my local host of my SQL workbench and then proceeded to work on the questions:

```
create database assignment1;
create table student(name varchar(30),student_number int ,class int,major varchar(20));
insert into student values('smith',17,1,'CS');
insert into student values('brown',8,2,'CS');
select* from student;
```



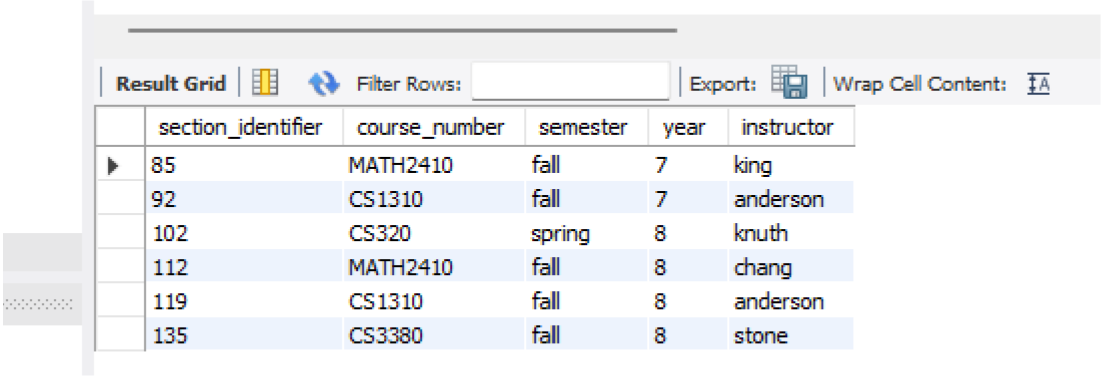
| name | student_number | class | major |
|---|---|---|---|
| smith | 17 | 1 | CS |
| brown | 8 | 2 | CS |

```
create table course(course_name varchar(30),course_number varchar(20),course_hours
int,department varchar(20));
insert into course(course_name,course_number,course_hours,department)
values ('intro to computer science','CS1310',4,'CS'),('data structures','CS3320',4,'CS'),('discrete
mathematics','MATH2410',3,'MATH'),('database','CS3380',3,'CS');
select* from course;
```



| course_name | course_number | course_hours | department |
|---|---|---|---|
| intro to computer science | CS1310 | 4 | CS |
| data structures | CS3320 | 4 | CS |
| discrete mathematics | MATH2410 | 3 | MATH |
| database | CS3380 | 3 | CS |

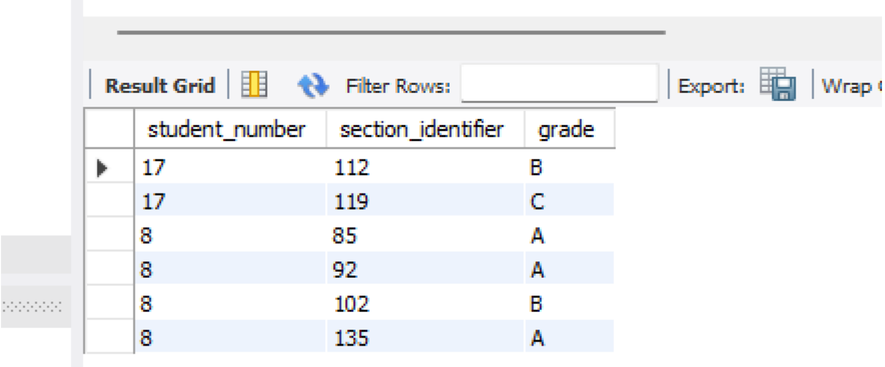
```
create table section(section_identifier int,course_number varchar(30),semester varchar(30),year
int,instructor varchar(30));

insert into section(section_identifier,course_number,semester,year,instructor)
 values
(85,'MATH2410','fall',7,'king'),(92,'CS1310','fall',7,'anderson'),(102,'CS320','spring',8,'knuth'),(11
2,'MATH2410','fall',8,'chang'),(119,'CS1310','fall',8,'anderson'),(135,'CS3380','fall',8,'stone');
select* from section;
```



| | section_identifier | course_number | semester | year | instructor |
|---|---|---|---|---|---|
| ▶ | 85 | MATH2410 | fall | 7 | king |
| | 92 | CS1310 | fall | 7 | anderson |
| | 102 | CS320 | spring | 8 | knuth |
| | 112 | MATH2410 | fall | 8 | chang |
| | 119 | CS1310 | fall | 8 | anderson |
| | 135 | CS3380 | fall | 8 | stone |

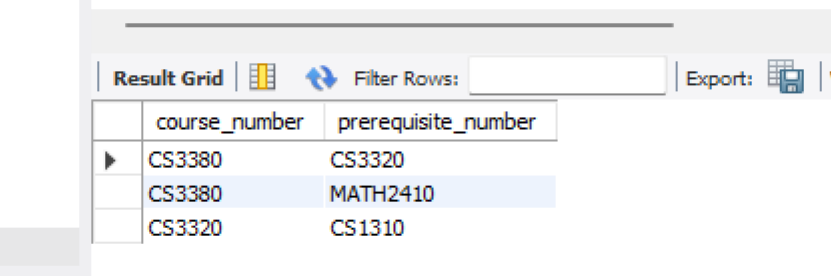
**create table grade_report(student_number int,section_identifier int, grade varchar(30));**

**insert into grade_report(student_number,section_identifier,grade)**
**values (17,112,'B'),(17,119,'C'),(8,85,'A'),(8,92,'A'),(8,102,'B'),(8,135,'A');**
**select* from grade_report;**



| | student_number | section_identifier | grade |
|---|---|---|---|
| ▶ | 17 | 112 | B |
| | 17 | 119 | C |
| | 8 | 85 | A |
| | 8 | 92 | A |
| | 8 | 102 | B |
| | 8 | 135 | A |

**create table prerequisite(course_number varchar(30),prerequisite_number varchar(30));**
**insert into prerequisite(course_number,prerequisite_number)**
**values ('CS3380','CS3320'),('CS3380','MATH2410'),('CS3320','CS1310');**
**select* from prerequisite;**



| | course_number | prerequisite_number |
|---|---|---|
| ▶ | CS3380 | CS3320 |
| | CS3380 | MATH2410 |
| | CS3320 | CS1310 |

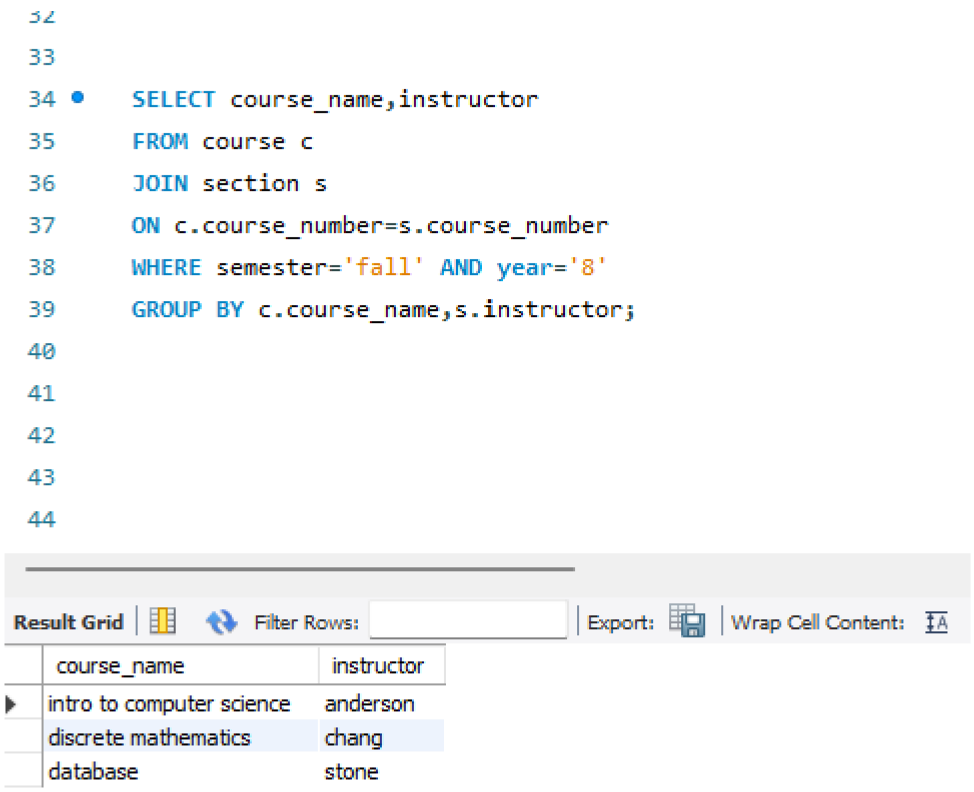
[Arun Ragavendhar]

[104837257]

1). Retrieve the name of each course along with the name of the instructor who taught that course during the fall of 08.

## SQL Query:

```
SELECT      course_name,instructor
FROM        course c
JOIN        section s
ON          c.course_number=s.course_number
WHERE       semester='fall' AND year='8'
GROUP BY    c.course_name,s.instructor;
```



```
SELECT course_name,instructor
FROM course c
JOIN section s
ON c.course_number=s.course_number
WHERE semester='fall' AND year='8'
GROUP BY c.course_name,s.instructor;
```

| course_name | instructor |
|---|---|
| intro to computer science | anderson |
| discrete mathematics | chang |
| database | stone |

b) For each section taught by Professor Anderson, retrieve the course number, semester, year, and number of students who took the section.

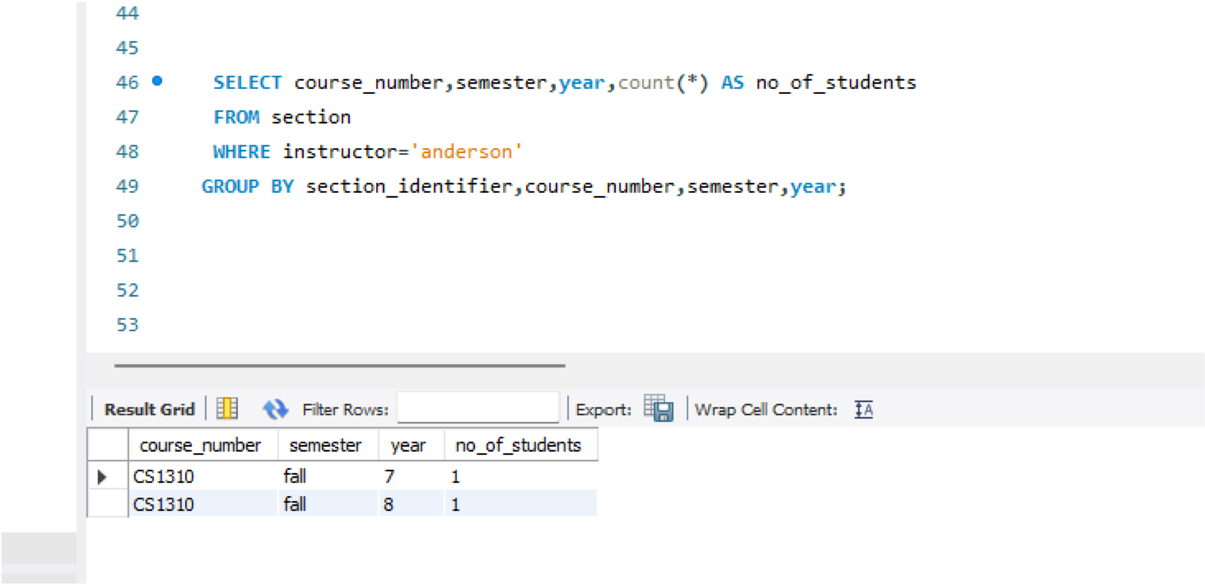
**SQL Query:**

**SELECT** course_number,semester,year,count(*) **AS** no_of_students

**FROM** section

**WHERE** instructor='anderson'

**GROUP BY** section_identifier,course_number,semester,year;



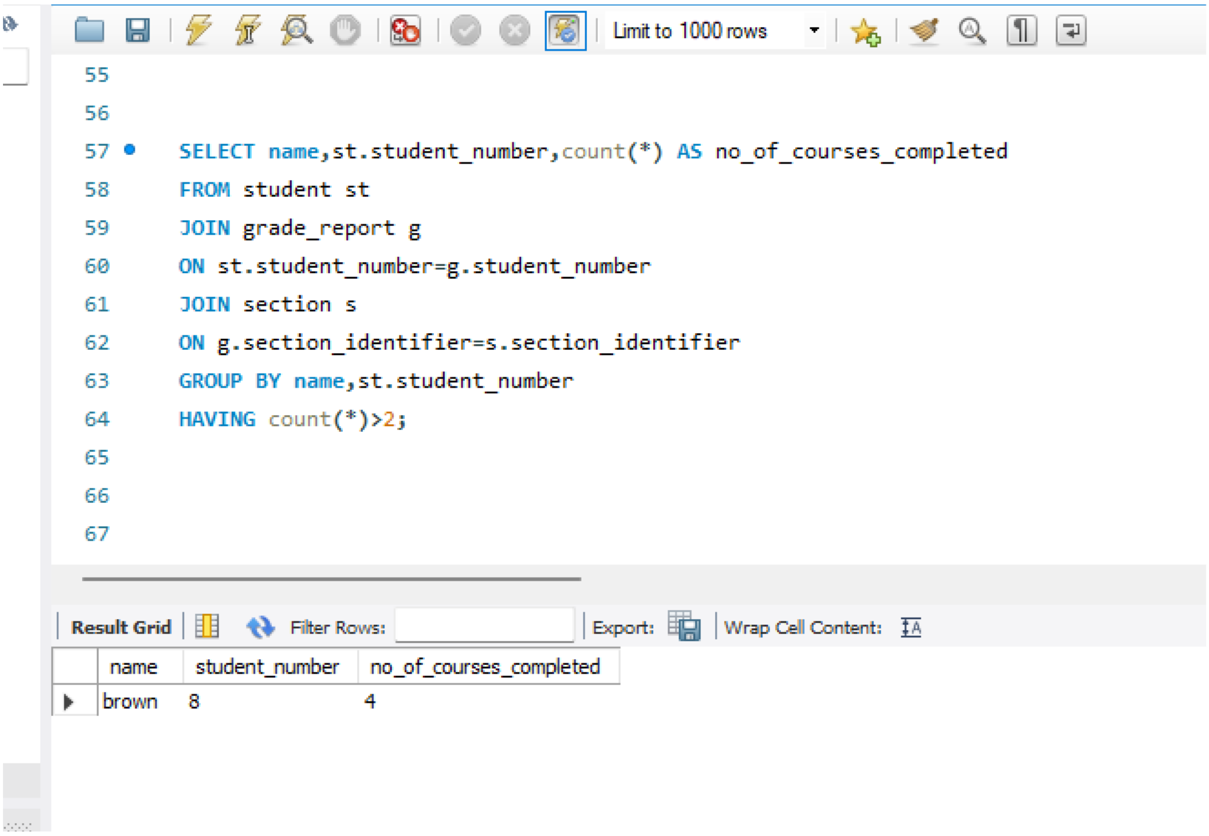
```
 SELECT course_number,semester,year,count(*) AS no_of_students
 FROM section
 WHERE instructor='anderson'
GROUP BY section_identifier,course_number,semester,year;
```

| course_number | semester | year | no_of_students |
|---|---|---|---|
| CS1310 | fall | 7 | 1 |
| CS1310 | fall | 8 | 1 |

c) For each student who completed more than two courses, retrieve the name, student number of the student and the number of courses completed by that student.

SQL Query:

```
SELECT      name,st.student_number,count(*) AS no_of_courses_completed
FROM        student st
JOIN        grade_report g
ON          st.student_number=g.student_number
JOIN        section s
ON          g.section_identifier=s.section_identifier
GROUP BY    name,st.student_number
HAVING      count(*)>2;
```



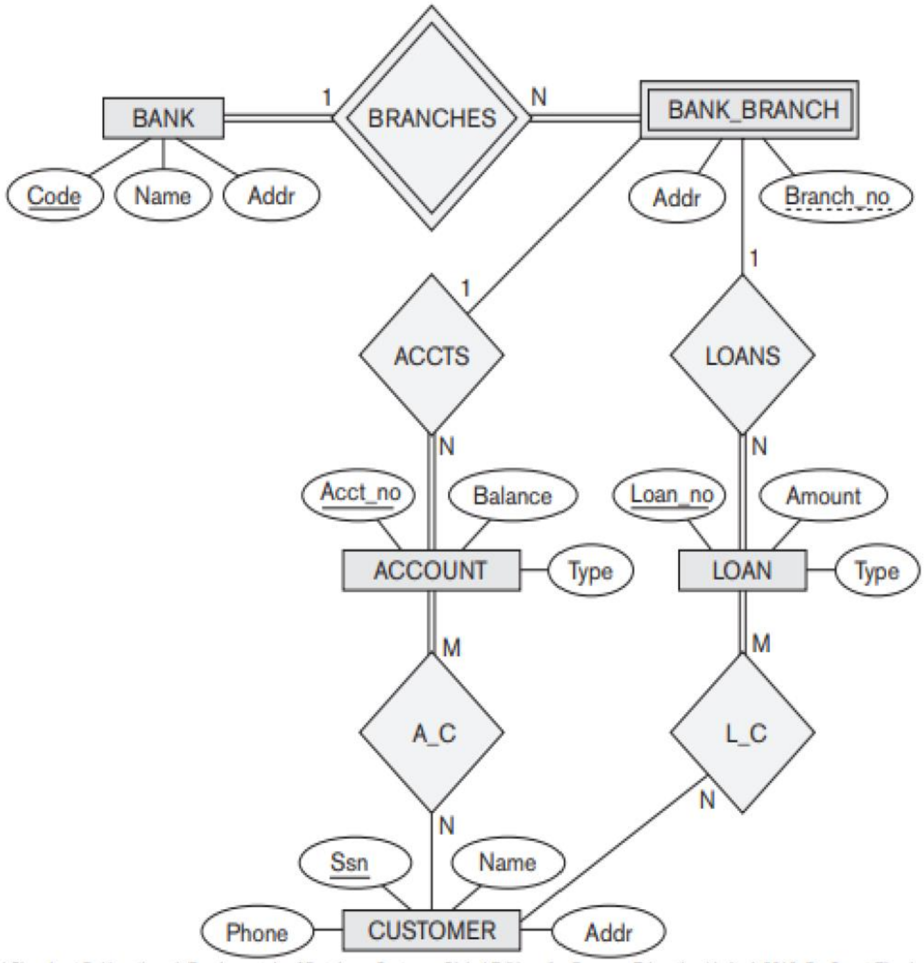
```
SELECT name,st.student_number,count(*) AS no_of_courses_completed
FROM student st
JOIN grade_report g
ON st.student_number=g.student_number
JOIN section s
ON g.section_identifier=s.section_identifier
GROUP BY name,st.student_number
HAVING count(*)>2;
```

| name | student_number | no_of_courses_completed |
|---|---|---|
| brown | 8 | 4 |

b) Map the BANK ER schema of shown in Figure 3.22 into a relational schema. Specify all primary keys and foreign keys.

**Figure 3.22**

An ER diagram for a BANK database schema.



**BANK ( Code , Name , Addr )**

Primary key - Code

**BANK_BRANCH ( Code, Branch_no , Addr )**

weak entity, hence have included attribute ‘code’ from BANK to reference it and make it stronger

Primary key – Code,Branch_no

Foreign key – Code - to reference BANK

**CUSTOMER ( Ssn , Name , Addr , Phone )**

Primary key – Ssn

**A_C ( Ssn , Acct_no)**

Primary key - Ssn,Acct_no

Foreign keys – Ssn – to reference CUSTOMER and Acct_no – to reference ACCOUNT

**ACCOUNT ( Acct_no , Type , Balance ,Code,Branch_no )**

Primary key - Acct_no

Foreign key – Code,Branch_no - to reference BANK_BRANCH

**L_C ( Ssn , Loan_no )**

Primary key - Ssn,Loan_no

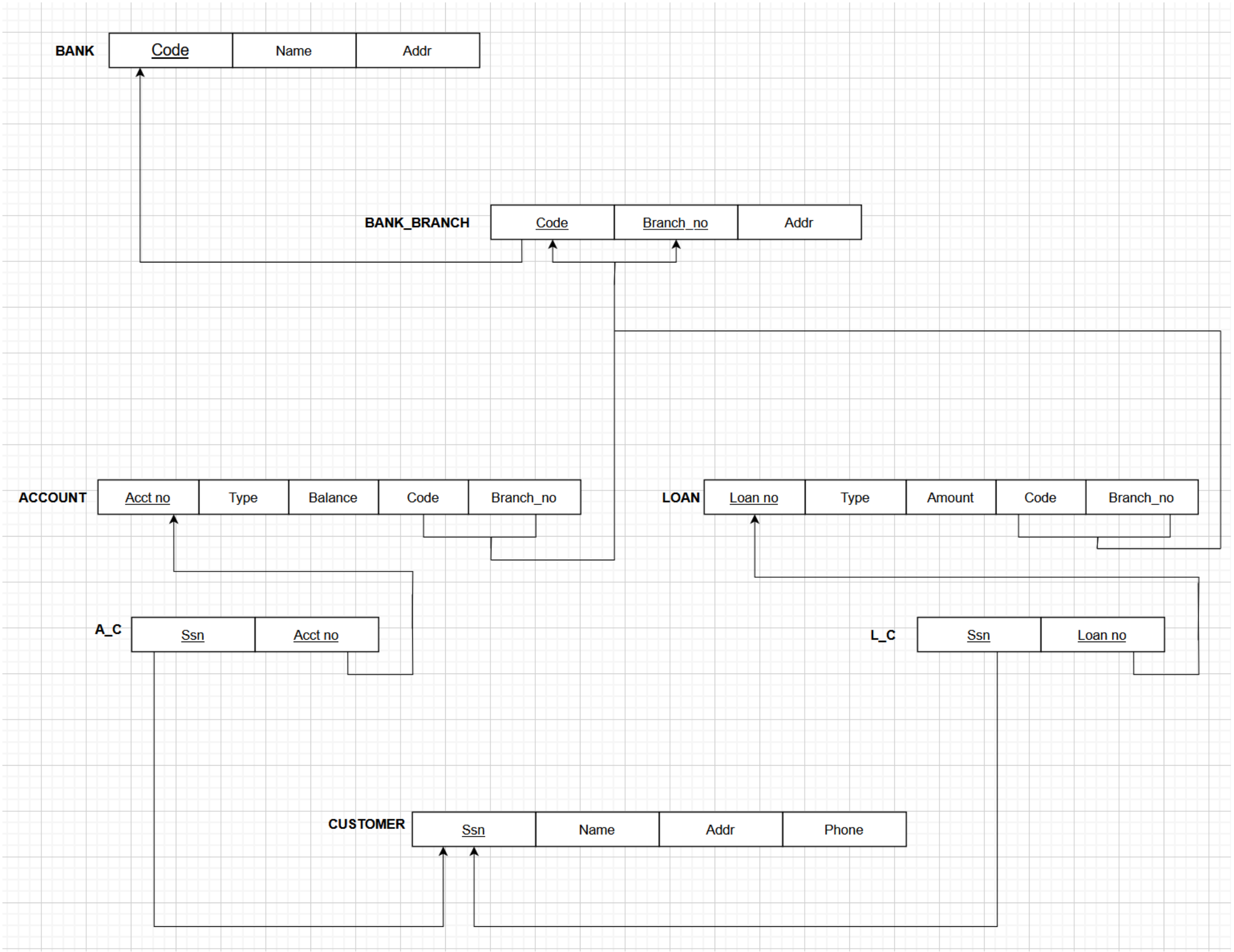
Foreign keys – Ssn – to reference CUSTOMER and Loan_no – to reference LOAN

**LOAN ( Loan_no , Type , Amount , Code, Branch_no )**

Primary key - Loan_no

Foreign key – Code,Branch_no – to reference BANK_BRANCH

This relational schema is then laid out as a diagram below:


BANK
Code
Name
Addr
BANK_BRANCH
Code
Branch_no
Addr
ACCOUNT
Acct no
Type
Balance
Code
Branch_no
LOAN
Loan no
Type
Amount
Code
Branch_no
A_C
Ssn
Acct no
L_C
Ssn
Loan no
CUSTOMER
Ssn
Name
Addr
Phone