**COS60009 – Database Management System for the Big Data age**

**Individual Assignment -1**

Name: Arun Ragavendhar Arunachalam Palaniyappan

Student ID: 104837257

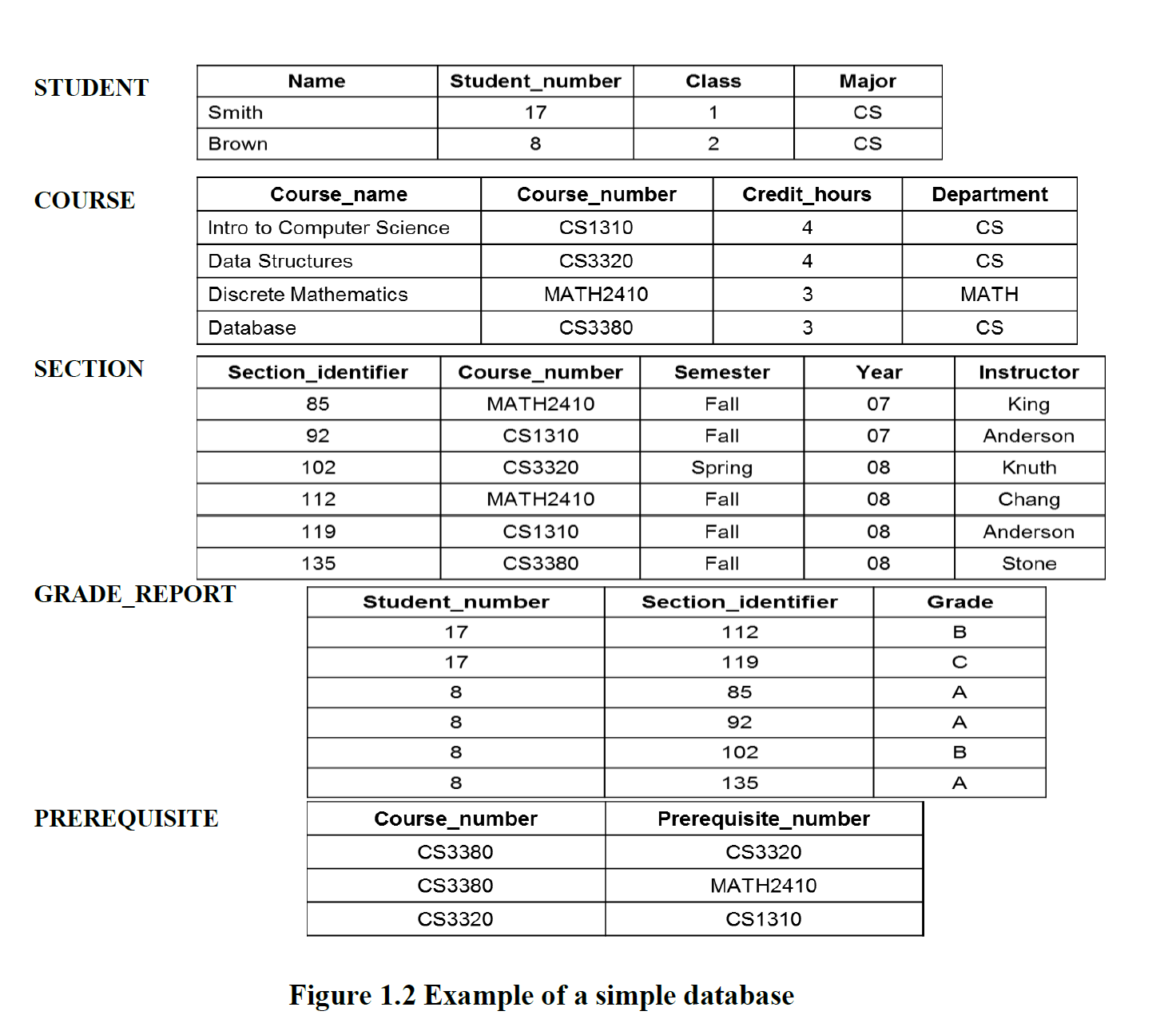
Task A -

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.



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.

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

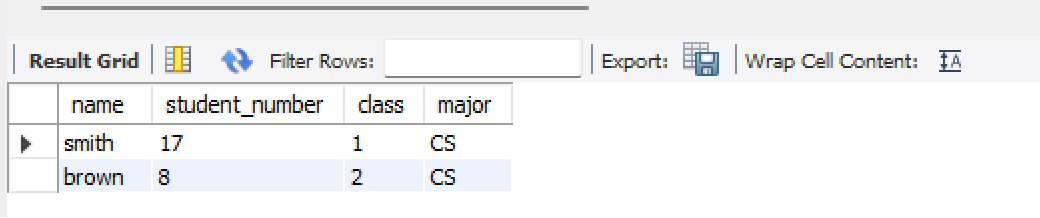
**create database assignment1;**

**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;**

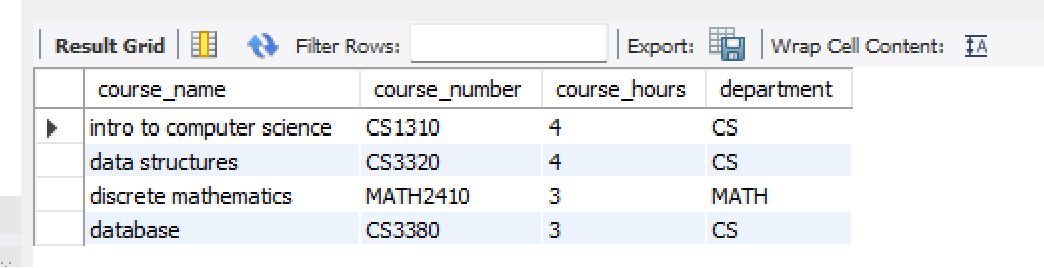
****

**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;**

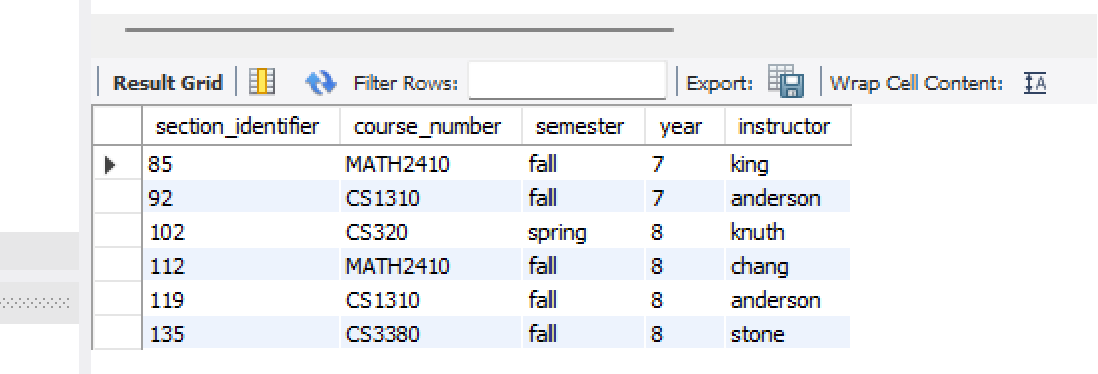


**create table section(section\_identifier int,course\_number varchar(30),semester varchar(30),year int,instructor varchar(30));**

**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'),(112,'MATH2410','fall',8,'chang'),(119,'CS1310','fall',8,'anderson'),(135,'CS3380','fall',8,'stone');**

**select\* from section;**

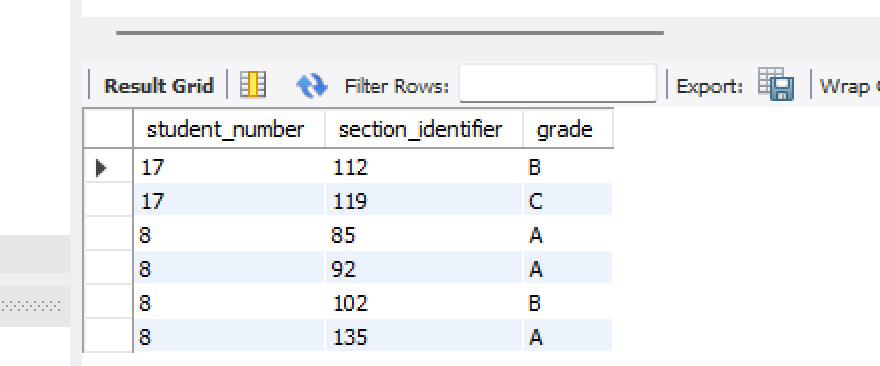


**create table grade\_report(student\_number int,section\_identifier int, grade varchar(30));**

**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;**

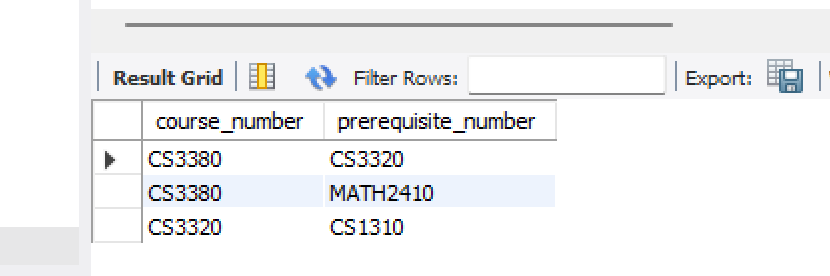


**create table prerequisite(course\_number varchar(30),prerequisite\_number varchar(30));**

**insert into prerequisite(course\_number,prerequisite\_number)**

**values ('CS3380','CS3320'),('CS3380','MATH2410'),('CS3320','CS1310');**

**select\* from prerequisite;**

****

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

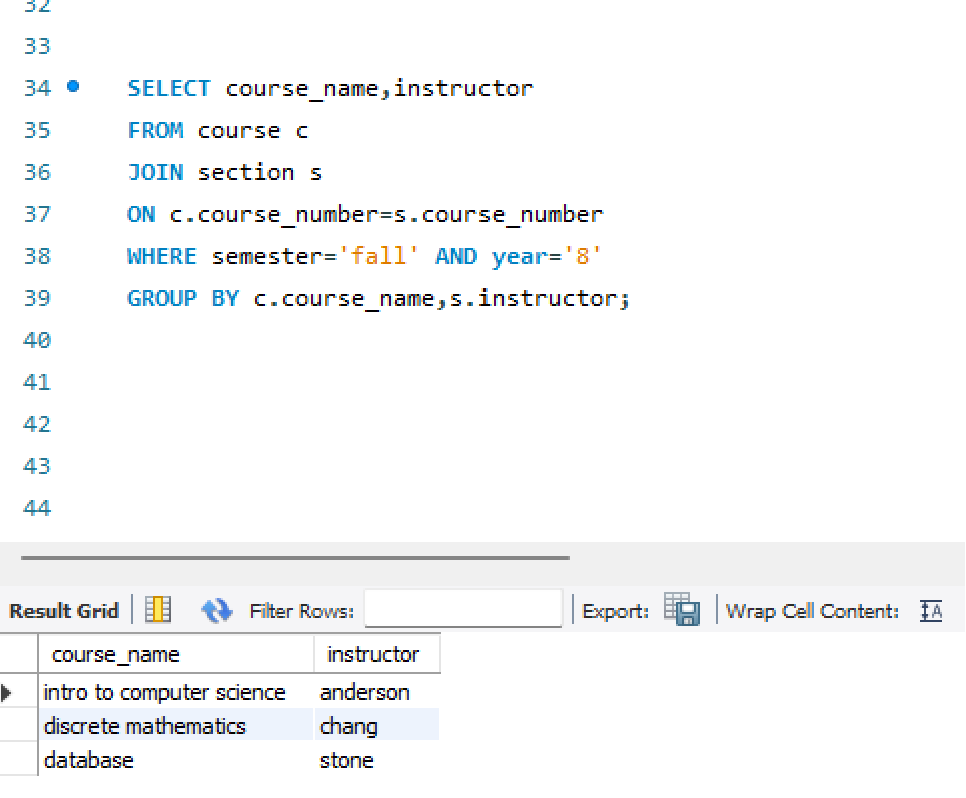
**FROM** course c

**JOIN** section s

**ON** c.course\_number=s.course\_number

**WHERE** semester='fall' AND year='8'

**GROUP BY** c.course\_name,s.instructor;



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

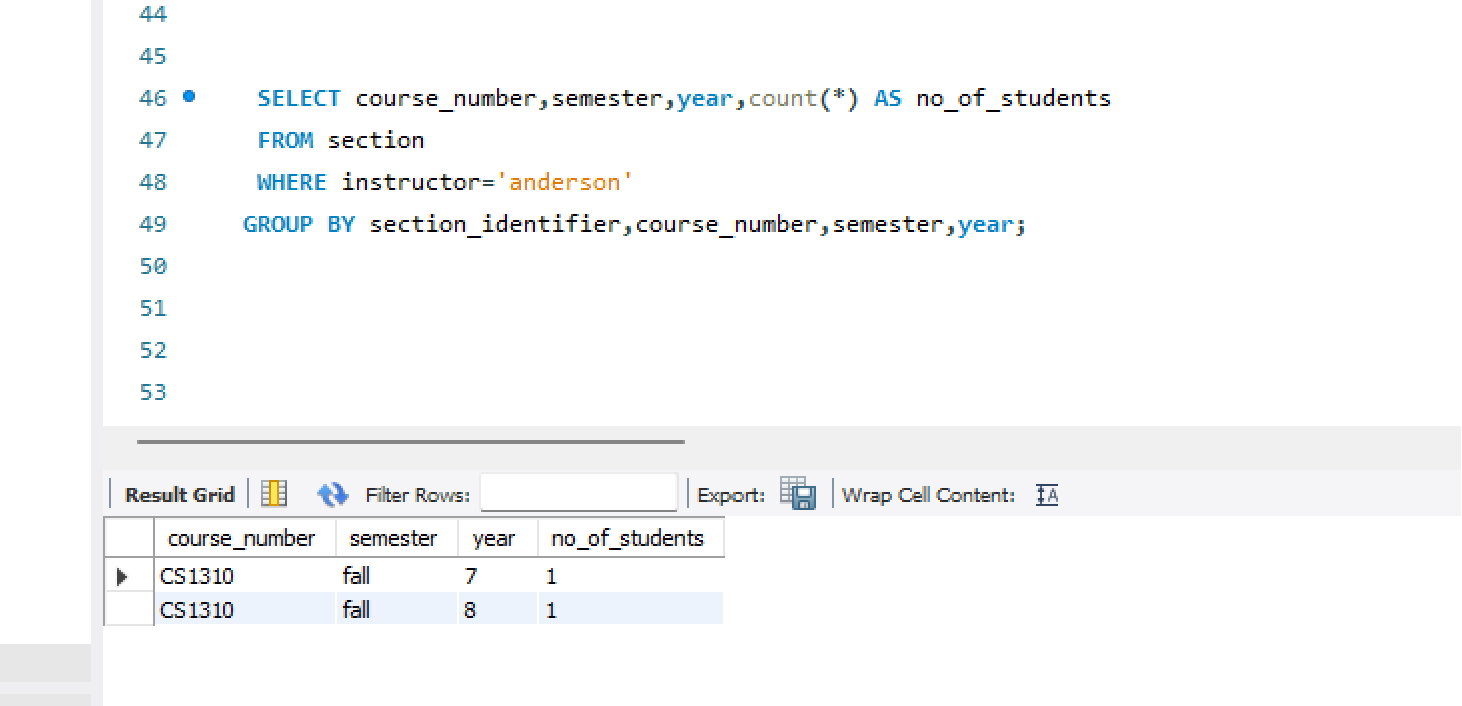
**SQL Query:**

**SELECT** course\_number,semester,year,count(\*) **AS** no\_of\_students

**FROM** section

**WHERE** instructor='anderson'

**GROUP BY**  section\_identifier,course\_number,semester,year;



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

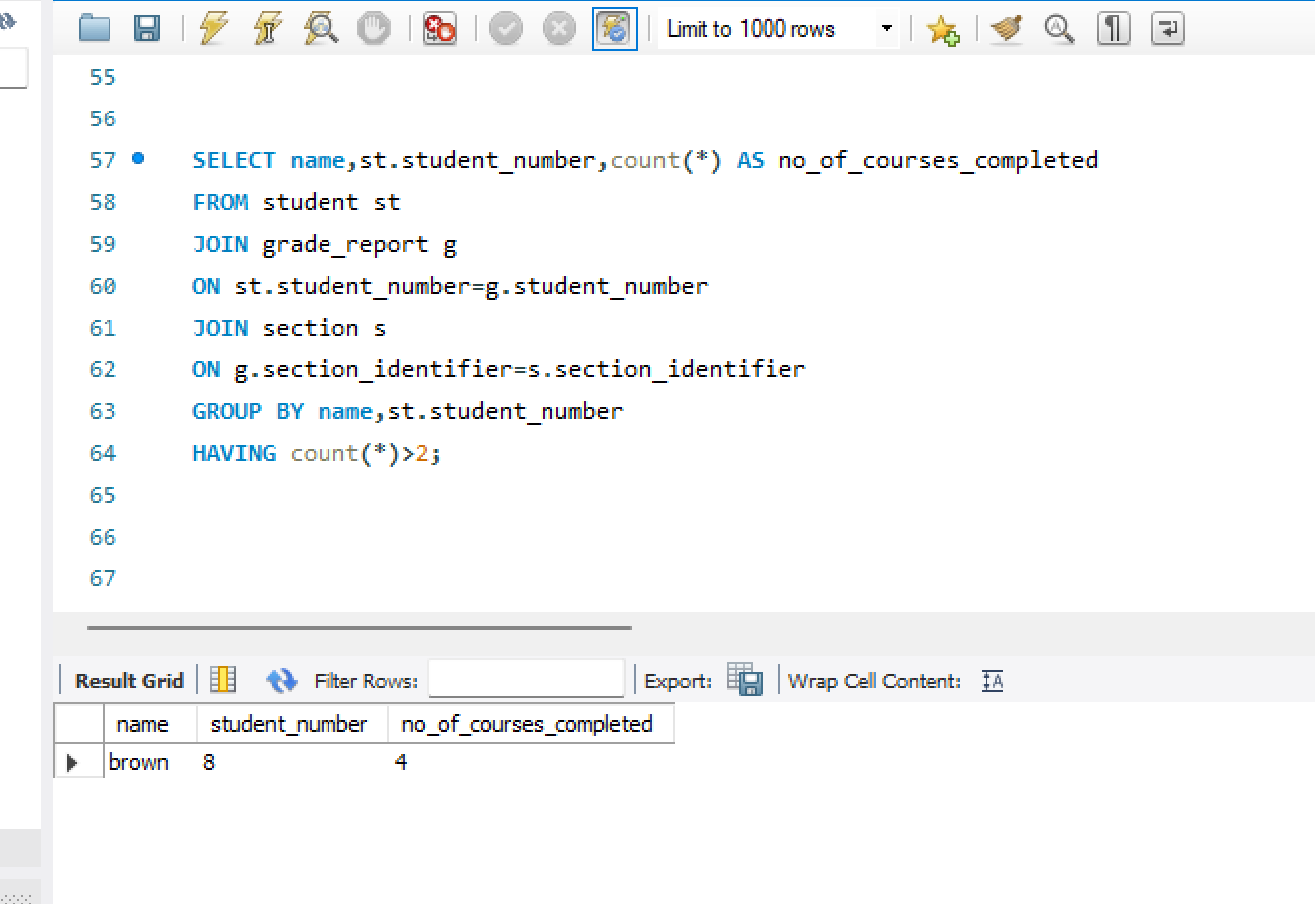
**ON** st.student\_number=g.student\_number

**JOIN** section s

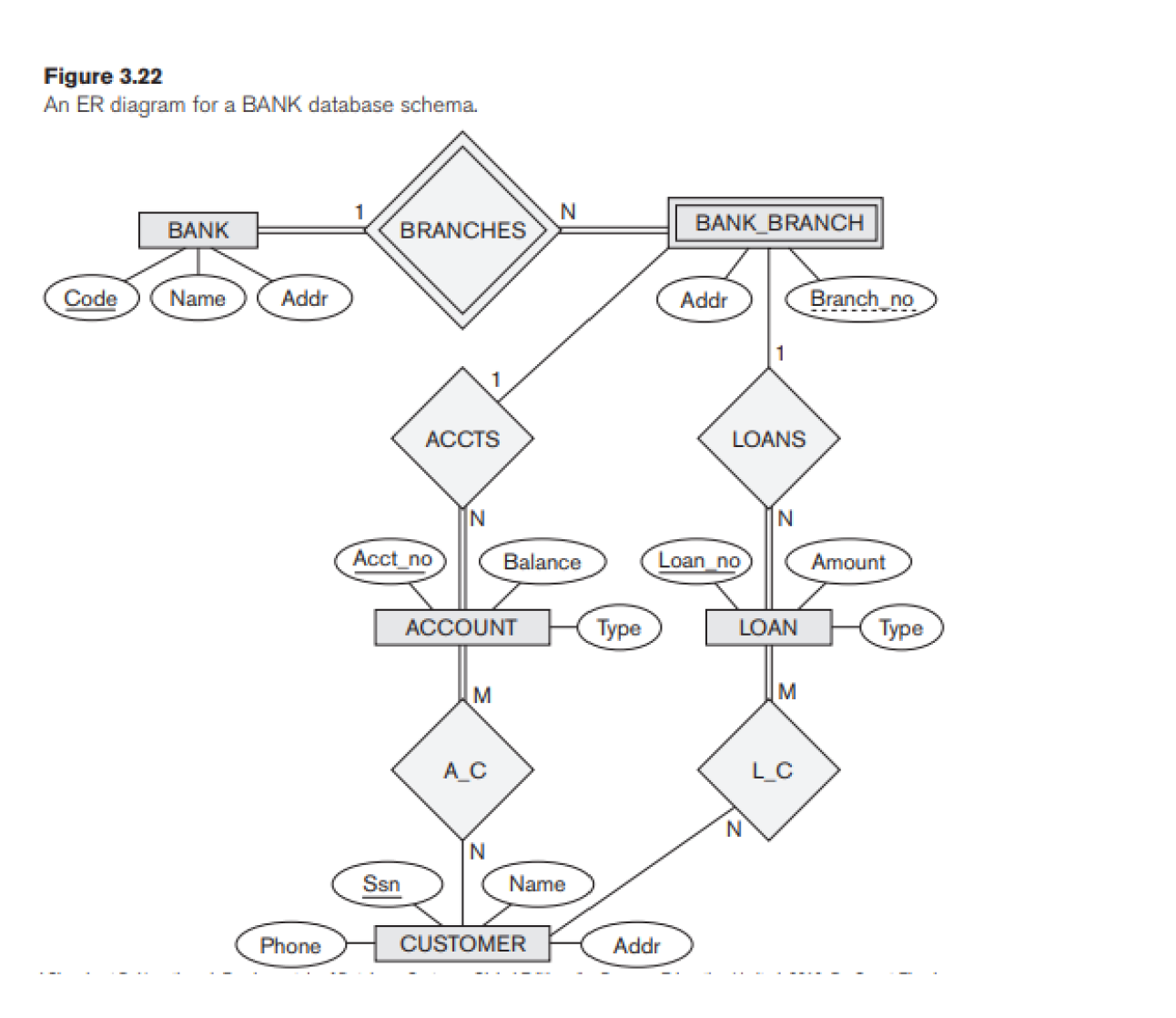
**ON** g.section\_identifier=s.section\_identifier

**GROUP BY** name,st.student\_number

**HAVING** count(\*)>2;



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



**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

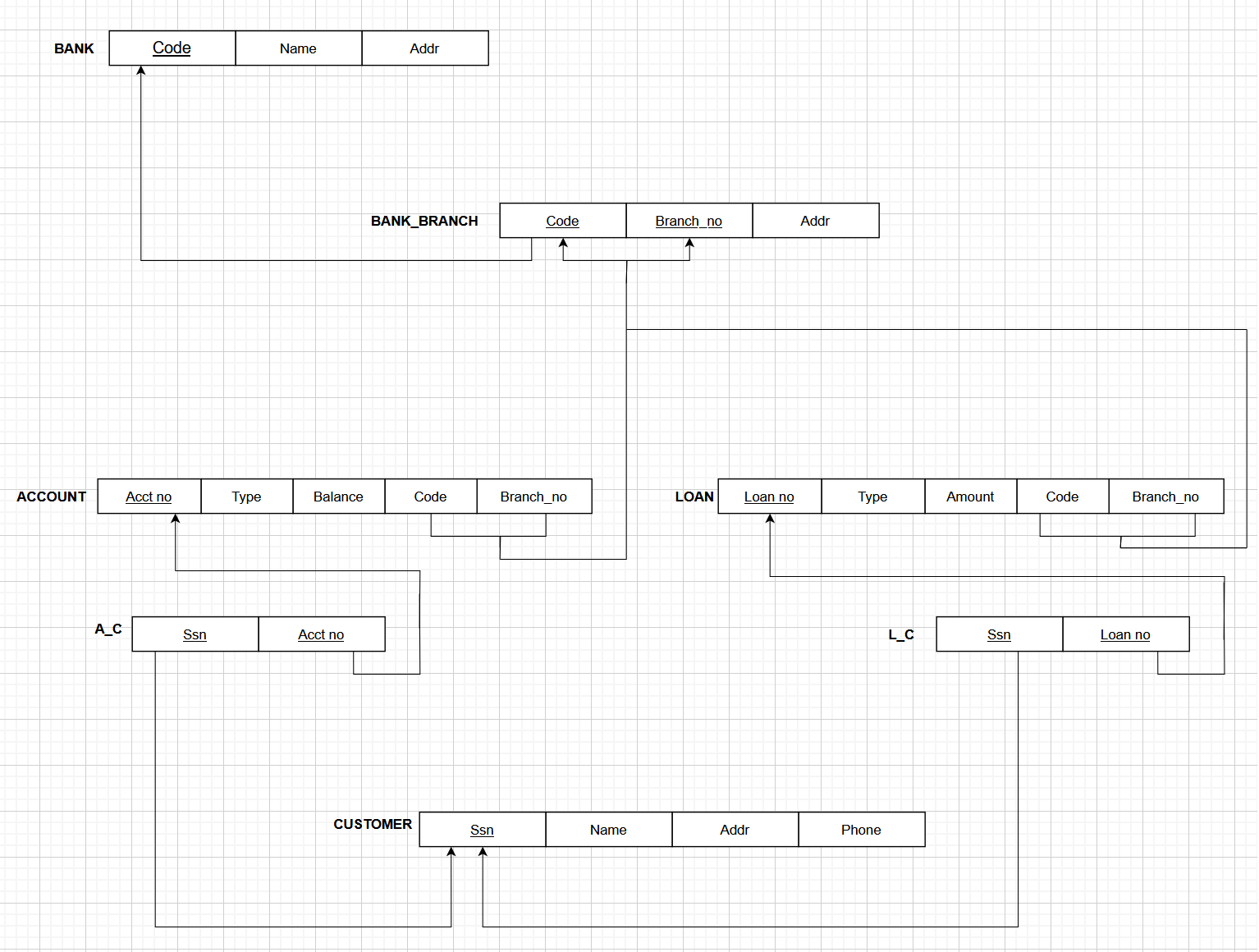
Foreign keys – Ssn – to reference CUSTOMER and Loan\_no – to reference LOAN

**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:

****