[Q.2] Answer the following questions

**Based on company database**

(a) Explain what would be retrieved by the following query

SELECT DISTINCT PNUMBER FROM PROJECT

WHERE PNUMBER IN (SELECT PNUMBER FROM PROJECT, DEPARTMENT, EMPLOYEE

WHERE DNUM = DNUMBER AND MGRSSN = SSN AND LNAME = 'Smith')

OR

PNUMBER IN

(SELECT PNO FROM WORKS\_ON, EMPLOYEE

WHERE ESSN = SSN AND LNAME = 'Smith');

**It returns all project numbers of the projects, which involve Smith as a manager who controls the project or Smith as an employee who participates in the project. The operator ‘WHERE IN’ returns values that matches values in the list. Let’s run the part of the statement:**

SELECT DISTINCT PNUMBER FROM PROJECT

WHERE PNUMBER IN (SELECT PNUMBER FROM PROJECT, DEPARTMENT, EMPLOYEE

WHERE DNUM = DNUMBER AND MGRSSN = SSN AND LNAME = 'Smith');

**Upon running the statement, we see that it will not return anything because Smith is not a manager of any department who controls any project. Running the whole statement would yield the product numbers of all the projects that Smith participates in which are 1 and 2. So the ‘OR’ operator must have decided that Smith is not a manager which is the first inner query so it decides to check the second inner query which turns to be true so we get all the project numbers that Smith participates in.**

(b) Explain what would be retrieved by the following query:

SELECT DISTINCT PNUMBER FROM PROJECT, DEPARTMENT, EMPLOYEE, WORKS\_ON

WHERE (DNUM= DNUMBER AND MGRSSN=SSN AND LNAME='smith')

OR (PNUMBER = PNO AND ESSN = SSN AND LNAME='smith');

**It returns all project numbers of the projects, which involve Smith as a manager who controls the project or Smith as an employee who participates in the project.**

(c) Write a SQL statement to retrieve a list of all project names of the projects, which involve an employee whose last name is 'Smith' as a worker on the project, or as a manager of the department that controls the project.

**SELECT DISTINCT pname FROM project, department, employee, works\_on**

**WHERE (pno = pnumber AND ssn = essn AND lname = "smith")**

**OR (dnumber = dnum AND ssn = mgrssn AND lname = "smith");**

(d) Write a SQL to retrieve a list of employees and the projects they are working on, ordered by the employee's department, and within each department ordered alphabetically by employee last name.

**SELECT fname, lname, pname, dnumber, dname FROM employee, department, project, works\_on**

**WHERE (pnumber = pno AND ssn = essn AND dnumber = dnum)**

**ORDER BY dnumber, lname;**

(e) Write a SQL statement to retrieve the project number, project name, and the number of employees who work on that project for each project.

**SELECT COUNT(\*), pnumber, pname FROM employee, department, project, works\_on**

**WHERE (pnumber = pno AND dnumber = dnum AND ssn = essn)**

**GROUP BY pnumber;**

(f) Explain a view, and explain the view created by the following SQL statement:

CREATE VIEW WORKS\_ON

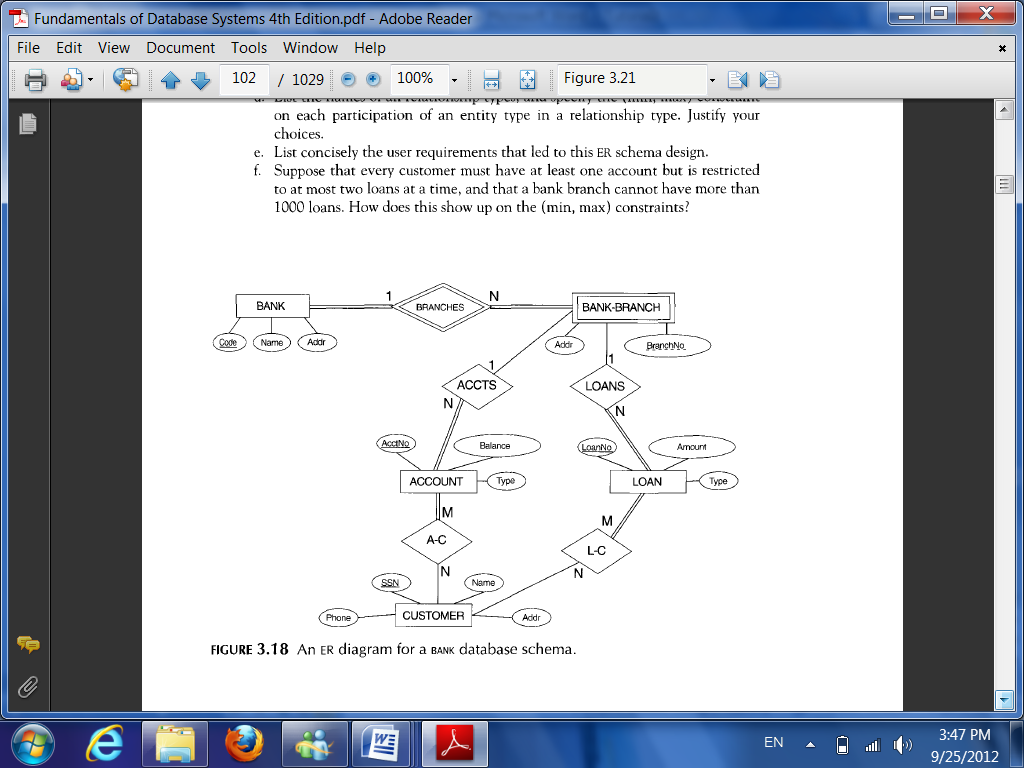
AS SELECT FNAME, LNAME, PNAME, HOURS

FROM EMPLOYEE, PROJECT, WORKS\_ON

WHERE SSN = ESSN AND PNO = PNUMBER;

**A view is a virtual table created on the result of a SQL query. A view contains columns and rows just like a table in a database. The above-mentioned SQL query returns the name, project name and project hours of all the employees who participated in a project and creates a new virtual table ‘WORKS\_ON’ and fills the fields with the result acquired by the query.**

[Q.6] Answer the following questions for a simple library application. Consider the ER diagram shown in the following Figure for part of a BANK database. Each bank can have multiple branches, and each branch can have multiple accounts and loans.



1. List the regular entity types in the ER diagram.

* **BANK**
* **ACCOUNT**
* **LOAN**
* **CUSTOMER**

(b) Is there a weak entity type? If so, give its name, its partial key, and its identifying relationship.

**Yes, there is a weak entity type. Its name is BANK-BRANCH. Its partial key is BranchNo. Its identifying relationship is BRANCHES.**

(c) What constraints do the partial key and the identifying relationship of the weak entity type specify in this diagram?

**Each BANK-BRANCH should belong to a BANK and each BANK must at least have one BANK-BRANCH hence, BranchNo. and bank Code uniquely identifies a branch.**

(d) List the names of all relationship types, and specify the (min,max) constraint on each

participation of an entity type in a relationship type.

**Relationship Types: BRANCHES, ACCTS, A-C, LOANS, L-C.**

N

1

branches

Bank

Bank branch

(1,N)

(1,1)

1

N

ACCTS

Accounts

Bank branch

(0,1)

(0,N)

M

N

A-C

Customer

Account

(1,1)

(0,N)

N

1

loans

Loan

Bank branch

(1,N)

(1,N)

N

M

(0,N)

(1,N)

L-C

Customer

Loan

(e) Suppose that every customer must have at least one account but is restricted to at most two loans at a time, and that a bank branch cannot have more than 1000 loans. How does this show up on the (min, max) constraints?

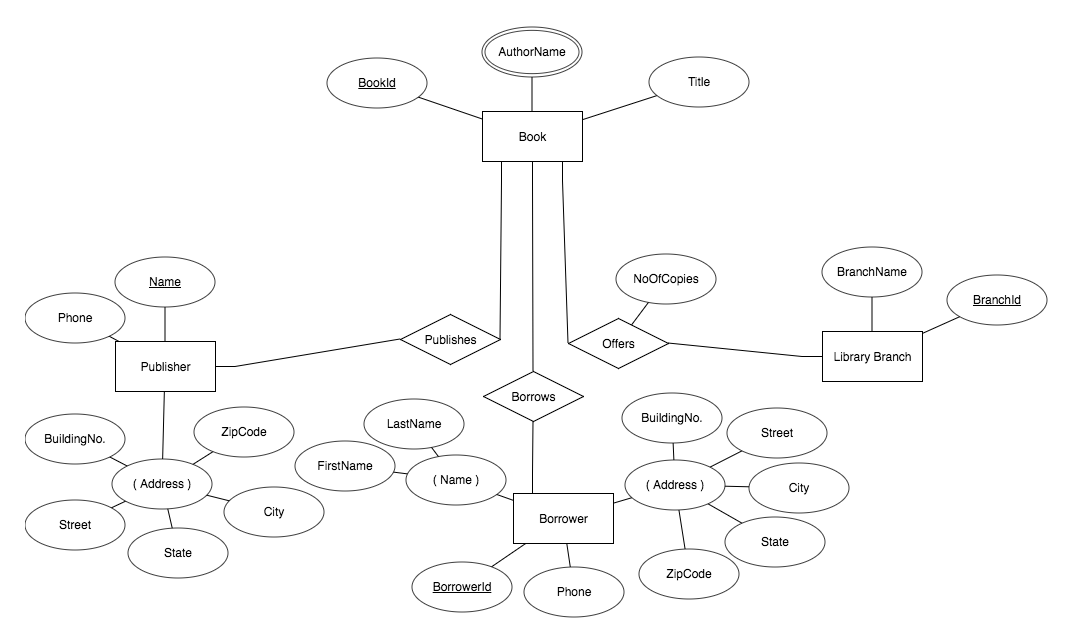
**Every customer must have at least one account: (1,N)**

**Every customer is restricted to at most two loans at a time: (0,2)**

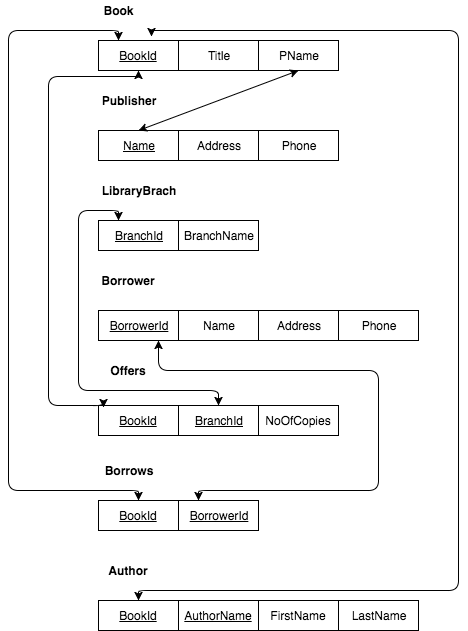
**A bank branch cannot have more than 1000 loans: (0,1000)**

[Q.4] Answer the following questions for a simple library application. • The data requirements of the library application are summarized as follows: BOOK entity is identified by BookId, it has title and multiple author names. PUBLISHER entity consists of Name, Address, and Phone attributes. Name is the key for the PUBLISHER. LIBRARY\_BRANCH entity has BranchId as a key and Branchname attribute additionally. BORROWER entity has BrowerId as key and additionally has name, address, phone attributes. Each LIBRARY\_BRANCH has one or more copies of the same book. In such a case, noOfCopies attribute needs to be maintained by the relationship. A book is published by only one publisher. A book can be loaned to a borrower at a specific library branch.

(a) Draw an ER diagram for the conceptual schema of the library database application. Note any unspecified requirements, and make appropriate assumptions to make the specification complete.

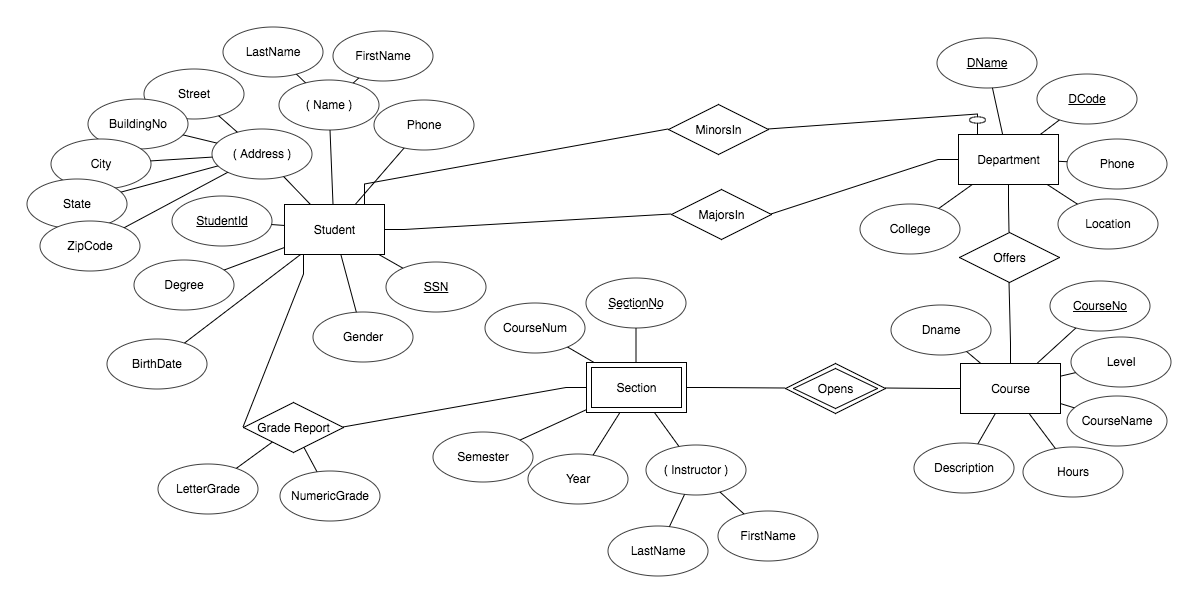


(b) Map the conceptual schema to logical database schema.

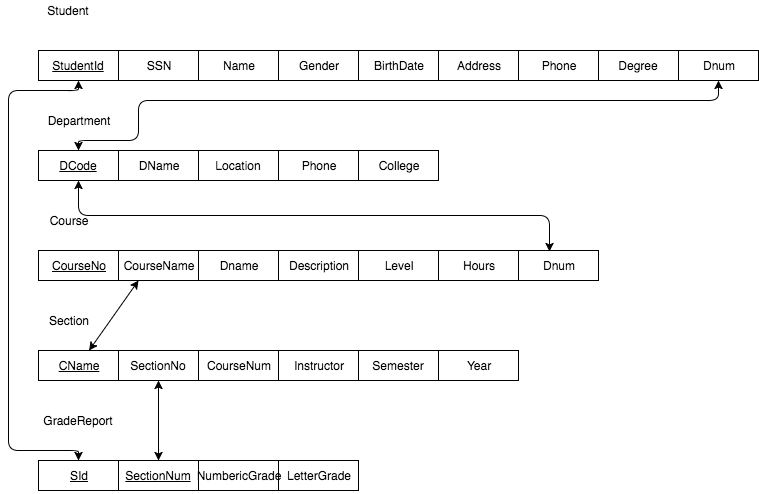


[Q.3] Consider the following set of requirements for a simpleUNIVERSITY database that is used to keep track of students' transcripts. • The university keeps track of each student's name, student number, social security number, current address and phone, permanent address and phone, birthdate, sex, class (freshman, sophomore, ..., graduate), major department, minor department (if any), and degree program 3 (B.A., B.S., ..., Ph.D.). Some user applications need to refer to the city, state, and zip of the student's permanent address, and to the student's last name. Both social security number and student number have unique values for each student. • Each department is described by a name, department code, office number, office phone, and college. Both name and code have unique values for each department. • Each course has a course name, description, course number, number of semester hours, level, and offering department. The value of course number is unique for each course. • Each section has an instructor, semester, year, course, and section number. The section number distinguishes different sections of the same course that are taught during the same semester/year; its values are 1, 2, 3, ..., up to the number of sections taught during each semester. • A grade report has a student, section, letter grade, and numeric grade (0, 1, 2, 3, 4 for F, D, C, B, A, respectively).

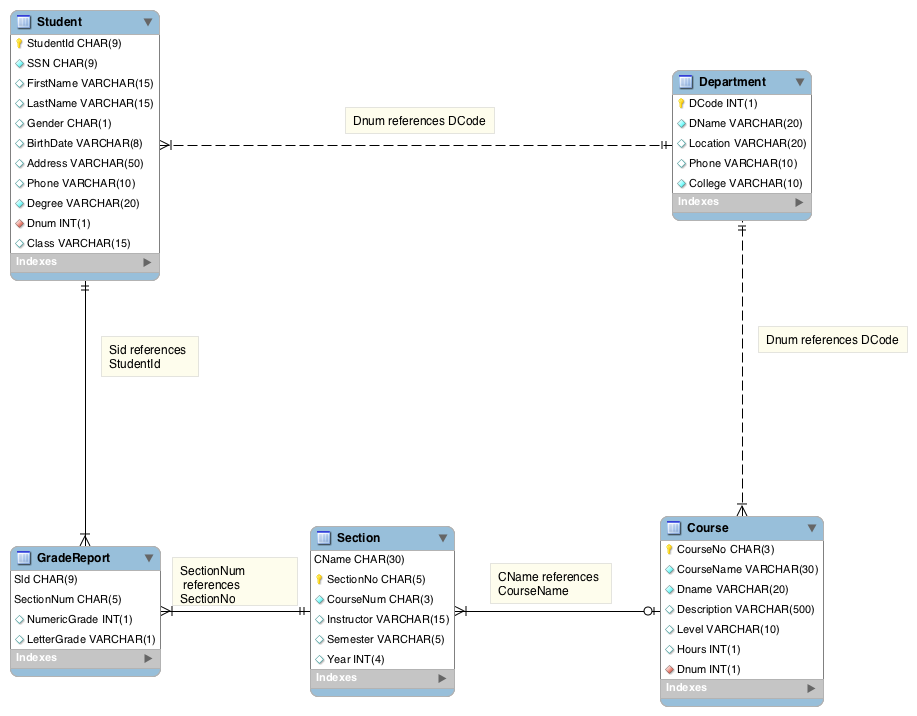
(a) Design the conceptual database schema using the ER diagram for this application. a. Specify key attributes of each entity type and structural constraints on each relationship type. Note any unspecified requirements, and make appropriate assumptions to make the specification complete.



(b) Map the conceptual schema to logical database schema.



(c) Using MySQL workbench, draw ER diagram of the derived logical database to generate the SQL script to create simpleUniversity database. The relationship types of the MySQL workbench ER diagram must be created based on referential constraints.



(d) Write SQL statements, using insert statement, to populate the simpleUniversity database. Each table must be populated with at least 4 tuples.

**Student**

INSERT INTO Student VALUES(111111111,890112290,"Larry", "James", "M", "09111994", "8901 Queens Blvd.", 7189990188, "Psychology", 1, "Freshman");

INSERT INTO Student VALUES(222222222,718900110,"Hulk", "Jamon", "M", "08011995", "9118 Parson Blvd.", 3476770177, "Computer Science", 2, "Senior");

INSERT INTO Student VALUES(333333333, 819901321, "Harris", "Rodman", "M", "01301997", "8999 Harris Blvd.", 9178910132, "Computer Science", 2, "Sophomore" );

INSERT INTO Student VALUES(444444444, 801290001,"James", "Williams", "M", "10091995", "1027 Katty Blvd.", 6461210177, "Chemical Engineering", 3, "Junior");

**Department**

INSERT INTO Department VALUES(1, "Psychology", "Harris Hall", "7187710188", "St. Davis");

INSERT INTO Department VALUES(2, "Computer Science", "David Hall", "7187710187", "St. Davis");

INSERT INTO Department VALUES(3, "Chemical Engineering", "Saint Hall", "7187710186", "St. Davis");

INSERT INTO Department VALUES(4, "Humanities", "Psychic Hall", "7187710180", "St. Davis");

**Course**

INSERT INTO Course VALUES(231, "Intro to Psychology", "Psychology", "Human mind", "Easy", 2, 1);

INSERT INTO Course VALUES(333, "Intro to networking", "Computer Science", "Networking learning", "Medium", 3, 2);

INSERT INTO Course VALUES(731, "Science of Petroleum", "Chemical Engineering", "Petroleum", "Hard", 3, 3);

INSERT INTO Course VALUES(115, "Intro to Species", "Humanities", "Introduction", "Easy", 2, 4);

INSERT INTO Course VALUES(410, "iOS programming", "Computer Science", "xcode, ios programming", "Hard", 4, 2);

**Section**

INSERT INTO Section VALUES("Intro to Psychology", "55151", 231,"Davis", "Fall", 2015);

INSERT INTO Section VALUES("Intro to Psychology", "55152", 231,"Letherman", "Fall", 2015);

INSERT INTO Section VALUES("intro to Psychology", "55153", 231,"James", "Fall", 2015);

INSERT INTO Section VALUES("Intro to Networking", "55677", 333,"Yu", "Fall", 2015);

INSERT INTO Section VALUES("Intro to Networking", "55678", 333,"Hernandez", "Fall", 2015);

INSERT INTO Section VALUES("Science of Petroleum", "55890", 731,"Betham", "Fall", 2015);

INSERT INTO Section VALUES("Intro to Species", "55655", 115,"Begam", "Fall", 2015);

**GradeReport**

INSERT INTO GradeReport VALUES(111111111,55151,2,"B");

INSERT INTO GradeReport VALUES(222222222,55677,1,"A");

INSERT INTO GradeReport VALUES(222222222,55152,1,"A");

INSERT INTO GradeReport VALUES(333333333,55678,1,"A");

INSERT INTO GradeReport VALUES(444444444,55890,3,"C");

(e) Write a SQL statement to retrieve the names of all senior students majoring in CS.

**SELECT FirstName, LastName FROM Student, Department**

**WHERE (Dnum = DCode AND class = "Senior" AND Degree = “Computer Science”);**

(f) Retrieve the name and major departments of all students who do not have any grade A in any of their courses.

**SELECT FirstName, LastName, DName FROM Student, Department**

**WHERE NOT EXISTS (SELECT LetterGrade FROM GradeReport WHERE StudentId = SId AND LetterGrade = "A")**

**AND Dnum = DCode;**

(g) Retrieve the name and major departments of all students with at least one grade A in any of their courses.

**SELECT FirstName, LastName, DName FROM Student, Department**

**WHERE NOT EXISTS (SELECT LetterGrade FROM GradeReport WHERE StudentId = SId AND NOT(LetterGrade = "A"))**

**AND Dnum = DCode;**

**AHAMED ABBAS**

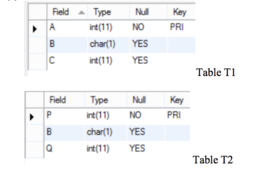
**Prof. Gwang**

**Database Systems**

**10/16/2017**

[Q.1] Answer the following questions

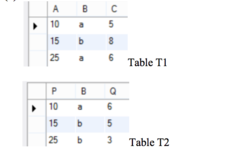
(a) Create table T1 and table T2 as described below.



**CREATE TABLE IF NOT EXISTS T1(A INT(11) NOT NULL, B CHAR(1) NULL, C INT(11) NULL, PRIMARY KEY(A));**

**CREATE TABLE IF NOT EXISTS T2(P INT(11) NOT NULL. B CHAR(1) NULL, Q INT(11) NULL, PRIMARY KEY(P));**

(b) Insert values into table T1 and table T2 as shown below.



**INSERT INTO T1 VALUES(10,”a”,”5”);**

**INSERT INTO T1 VALUES(15,”b”,”8”);**

**INSERT INTO T1 VALUES(25,”a”,”6”);**

**INSERT INTO T2 VALUES(10,”a”,”6”);**

**INSERT INTO T2 VALUES(15,”b”,”5”);**

**INSERT INTO T2 VALUES(25,”b”,”3”);**

(c) Show the retrieved table after the following query is executed.

select \* from T1, T2

where T1.A = T2.P;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | C | P | B | Q |
| 10 | a | 5 | 10 | a | 6 |
| 15 | b | 8 | 15 | b | 5 |
| 25 | a | 6 | 25 | b | 3 |

(d) Show the retrieved table content after the following query is executed.

select \* from T1

natural join T2;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| B | A | C | P | Q |
| a | 10 | 5 | 10 | 6 |
| a | 25 | 6 | 10 | 6 |
| b | 15 | 8 | 15 | 5 |
| b | 15 | 8 | 25 | 3 |

(e) Show the retrieved table content after the following query is executed.

select \* from T2

right outer join T1

on T1.C = T2.Q;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P | B | Q | A | B | C |
| 10 | a | 6 | 25 | a | 6 |
| 15 | b | 5 | 10 | a | 5 |
| null | null | null | 15 | b | 8 |

(f) Create table named A with two columns x as int, y as varchar(5)

**CREATE TABLE IF NOT EXISTS A(x INT(), y VARCHAR(5));**

(g) Create table named B with two columns v as int, w as varchar(5)

**CREATE TABLE IF NOT EXISTS B(v INT(), w VARCHAR(5));**

(h) Write SQL statement to get union of two tables

**SELECT \* FROM A**

**UNION**

**SELECT \* FROM B;**

(i) MySQL does not provide difference (or minus) and intersection set operations, write SQL statements to get intersection and difference of two tables (i.e., A ∩B and A-B))

**Intersection:**

**SELECT \* FROM A**

**WHERE A.x IN (SELECT B.v FROM B);**

**Difference:**

**SELECT \* FROM A**

**WHERE NOT EXISTS (SELECT B.v FROM B WHERE A.x = B.v);**

[Q.5] Answer the following questions You are invited as a database architect to develop database schema for maintaining data for keeping track of the members of each Study Group organized for the CS students at Lehman College.

• Each member is uniquely identified by a member ID

o Each member in a study group must have an email address, first name, and last name 4

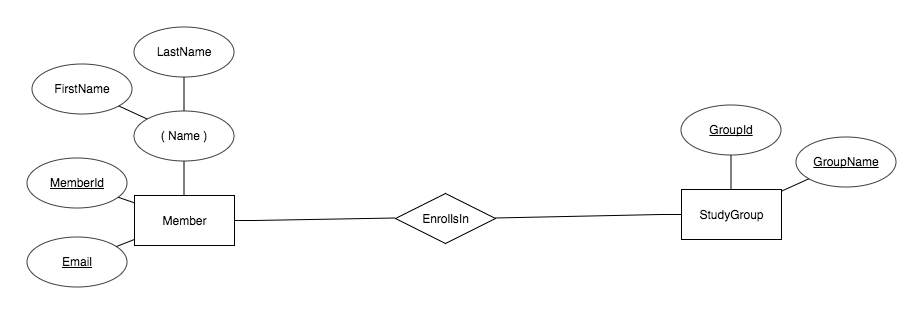
o Member’s email address is unique

o Each member can belong to any number of groups

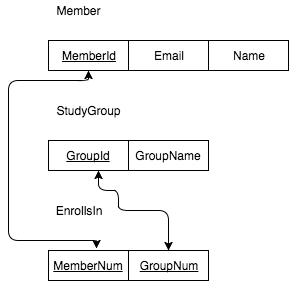
• Each group is identified by a unique group ID

o Each group must have a unique name

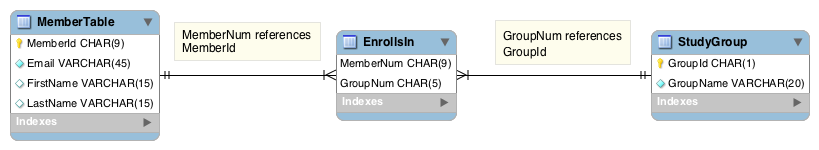
(a) Design a conceptual schema using EER diagram for storing data (members, groups) as explained.



(b) Derive the database schema based on EER to Relational Mapping.



(c) Name member relation as MemberTable, and write a SQL script to create the member relation (Note: make sure to enforce referential integrity constraints).



(d) Based on the database relations (tables) you derived from the EER diagram, write a SQL to retrieve member’s first name, last name, email address, and study group name identified by group id 2 (we assume there are five study groups).

**SELECT FirstName, LastName, Email, GroupName FROM MemberTable, StudyGroup, EnrollsIn**

**WHERE (MemberId = MemberNum AND GroupId = GroupNum AND GroupId = 2);**