Assignment 2 Instructions:

Write SQL DDL and DML statements to answer the following two questions. Your answers need to have both of the SQL command and a screenshot of the corresponding table. For example, you write a CREATE TABLE command to create the STUDENT table. Then your screenshot is the result of

SELECT * FROM STUDENT;

STUDENT (StudentID, StudentName)

StudentID	StudentName
38214	Letersky
54907	Altvater
66324	Aiken
70542	Marra
...	

QUALIFIED (FacultyID, CourseID, DateQualified)

FacultyID	CourseID	DateQualified
2143	ISM 3112	9/2011
2143	ISM 3113	9/2011
3467	ISM 4212	9/2018
3467	ISM 4930	9/2019
4756	ISM 3113	9/2014
4756	ISM 3112	9/2014
...		

FACULTY (FacultyID, FacultyName)

FacultyID	FacultyName
2143	Birkin
3467	Berndt
4756	Collins
...	

SECTION (SectionNo, Semester, CourseID)

SectionNo	Semester	CourseID
2712	I-2021	ISM 3113
2713	I-2021	ISM 3113
2714	II-2021	ISM 4212
2715	II-2021	ISM 4930
...		

COURSE (CourseID, CourseName)

CourseID	CourseName
ISM 3113	Syst Analysis
ISM 3112	Syst Design
ISM 4212	Database
ISM 4930	Networking
...	

REGISTRATION (StudentID, SectionNo)

StudentID	SectionNo
38214	2714
54907	2714
54907	2715
66324	2713
...	

1. This problem is based on the class scheduling 3NF relations along with some sample data shown in the above Figure. Not shown in this figure are data for an ASSIGNMENT relation, which represents a many-to-many relationship between faculty and sections. Not that values of the SectionNo column do not repeat across semesters.

1.1. (7 relations, 3 points each, 21 points total) Write SQL DDL to create the relations shown, including the ASSIGNMENT relation, in MySQL (shorten, abbreviate, or change any data names, as needed for your SQL commands). Assume the following attribute data types:

StudentID (integer, primary key)

StudentName (25 characters)

FacultyID (integer, primary key)

FacultyName (25 characters)

CourseID (8 characters, primary key)

CourseName (15 characters)

DateQualitfied (date)

SectionNo (integer, primary key)

Semester (7 characters)

1.2. (6 tables, 3 points each, 18 points total) Write SQL DML statements to insert the sample data (shown in the Figure) into the corresponding tables. (Only one screenshot for each of the end tables)

1.3. (6 points) Use SQL to define the following view:

StudentName	Semester	Course
Letersky	II-2021	ISM 4212
Altvater	II-2021	ISM 4212
Altvater	II-2021	ISM 4930
Aiken	I-2021	ISM 3113

1.4. (3 points each, 15 points total) Write SQL commands for the following:

a. Write an INSERT command to add a student with a student ID of 65798 and last name Lopez to the Student table.

b. Now write a command that will remove Lopez from the Student table.

c. Create an SQL command that will modify the name of course ISM 4212 from “Database” to “Introduction to Relational Databases”.

d. Change the name of the second column in the Student table from “StudentName” to “StudentLastName”.

e. Add a check constraint on the Qualified table so that the DateQualified field can only take date values greater than or equal to “2011-09-01”. (No screenshots required)

TUTOR (TutorID, CertDate, Status)

TutorID	CertDate	Status
100	1/05/2021	Active
101	1/05/2021	Temp Stop
102	1/05/2021	Dropped
103	5/22/2021	Active
104	5/22/2021	Active
105	5/22/2021	Temp Stop
106	5/22/2021	Active

STUDENT (StudentID, Group, Read)

StudentID	Group	Read
3000	3	2.3
3001	2	5.6
3002	3	1.3
3003	1	3.3
3004	2	2.7
3005	4	4.8
3006	3	7.8
3007	4	1.5

MATCH HISTORY (MatchID, TutorID, StudentID, StartDate, EndDate)

MatchID	TutorID	StudentID	StartDate	EndDate
1	100	3000	1/10/2021	
2	101	3001	1/15/2021	5/15/2021
3	102	3002	2/10/2021	3/01/2021
4	106	3003	5/28/2021	
5	103	3004	6/01/2021	6/15/2021
6	104	3005	6/01/2021	6/28/2021
7	104	3006	6/01/2021	

2. The above figure shows a database that tracks an adult literacy program. Tutors complete a certification class offered by the agency. Students complete an assessment interview that results in a report for the tutor and a recorded Read score. When matched with a student, a tutor meets with the student for one to four hours per week. Some students work with the same tutor for years, some for less than a month. Other students change tutors if their learning style does not match the tutor's tutoring style. Many tutors are retired and are available to tutor only part of the year. Tutor status is recorded as Active, Temp Stop, or Dropped.

2.1. (4 points each, 12 points total) Write SQL commands to create the 3 tables. Make necessary assumptions for the column data types.

2.2. (4 points each, 12 points total) Write SQL DML statements to insert the sample data (shown in the Figure) into the corresponding tables. (Only one screenshot for each of the end tables)

2.3. (4 points each, 16 points total) Write SQL commands for the following:

a. Modify both of the foreign keys on the MATCH HISTORY table are ON DELETE RESTRICT and ON UPDATE CASCADE. (No screenshots required)

b. Rename the table MATCH HISTORY to MATCH_HIST. (No screenshots required)

c. Assume we want the fields certDate in the TUTOR table and Read in the STUDENT table to be NOT NULL. Modify your tables to meet the requirements. (No screenshots required)

d. Create an index on the field Group in the STUDENT table. (No screenshots required)

Jared McMullen
GSE S520-01-2218

Homework 2

Question 1:

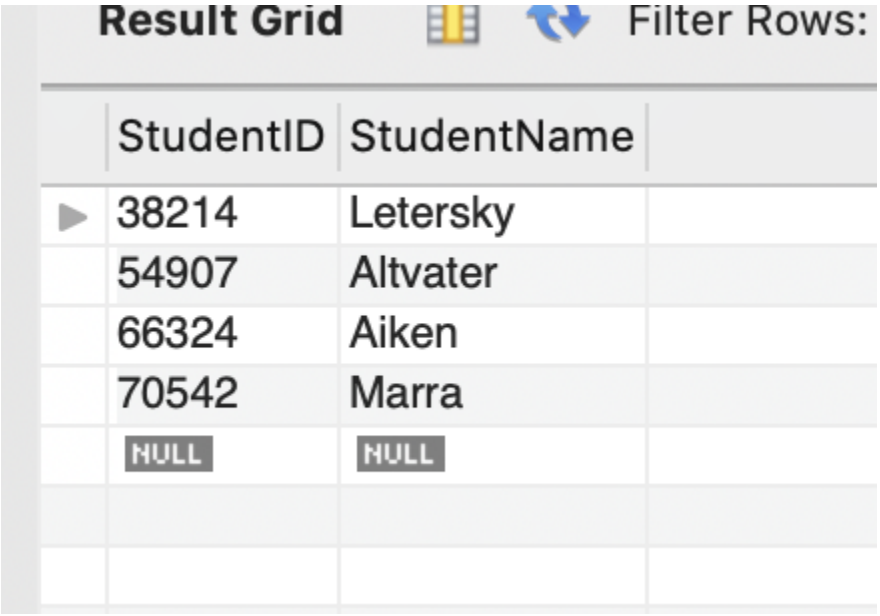
1.1)

```
CREATE DATABASE assignment2final1;
USE assignment2final1;

CREATE TABLE student
(
StudentID INT,
StudentName VARCHAR(25),
PRIMARY KEY (StudentID)
);


CREATE TABLE faculty
(
FacultyID INT,
FacultyName VARCHAR(25),
PRIMARY KEY (FacultyID)
);

CREATE TABLE course
(
CourseID CHAR(8),
CourseName VARCHAR(15),
PRIMARY KEY (CourseID)
);

CREATE TABLE qualified
(
FacultyID INT,
CourseID CHAR(8),
DateQualified DATE,
PRIMARY KEY (FacultyID, CourseID),
FOREIGN KEY (CourseID) REFERENCES course (CourseID)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
```

```
FOREIGN KEY (FacultyID) REFERENCES faculty (FacultyID)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);

CREATE TABLE section
(
SectionNo INT,
Semester VARCHAR(7),
CourseID CHAR(8),
PRIMARY KEY (SectionNo),
FOREIGN KEY (CourseID) REFERENCES course (CourseID)
        ON UPDATE CASCADE
   ON DELETE CASCADE
);

CREATE TABLE registration
(
StudentID INT,
SectionNo INT,
PRIMARY KEY (StudentID, SectionNo),
FOREIGN KEY (StudentID) REFERENCES Student (StudentID)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
FOREIGN KEY (SectionNo) REFERENCES Section (SectionNo)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);


CREATE TABLE assignment
(
SectionNo INT,
FacultyID INT,
PRIMARY KEY (SectionNo, FacultyID),
FOREIGN KEY (SectionNo) REFERENCES SECTION (SectionNo)
        ON UPDATE CASCADE
   ON DELETE CASCADE,
FOREIGN KEY (FacultyID) REFERENCES FACULTY (facultyID)
        ON UPDATE CASCADE
   ON DELETE CASCADE
);
```

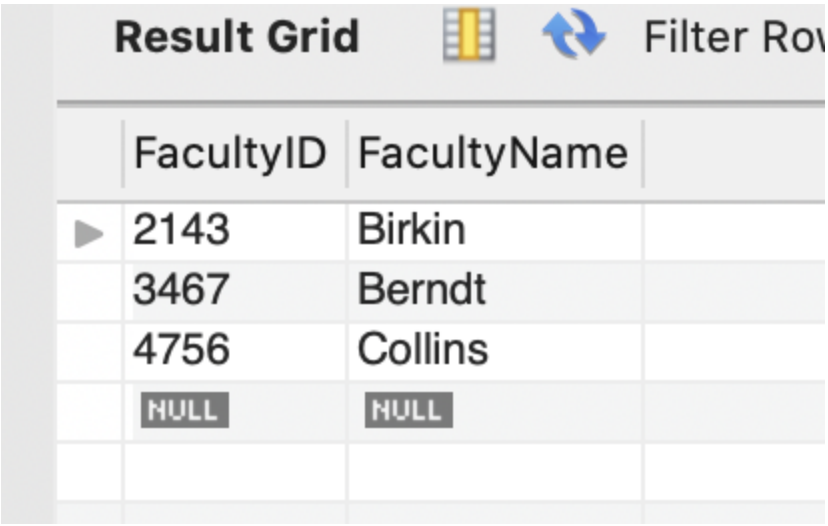
1.2)

```
INSERT INTO student

VALUES

("38214", "Letersky"),

("54907", "Altvater"),

("66324", "Aiken"),

("70542", "Marra");

SELECT * from student;
```



StudentID	StudentName
38214	Letersky
54907	Altvater
66324	Aiken
70542	Marra
NULL	NULL

```
INSERT INTO faculty

VALUES

("2143", "Birkin"),

("3467", "Berndt"),

("4756", "Collins");

SELECT * FROM faculty;
```



INSERT INTO course

VALUES

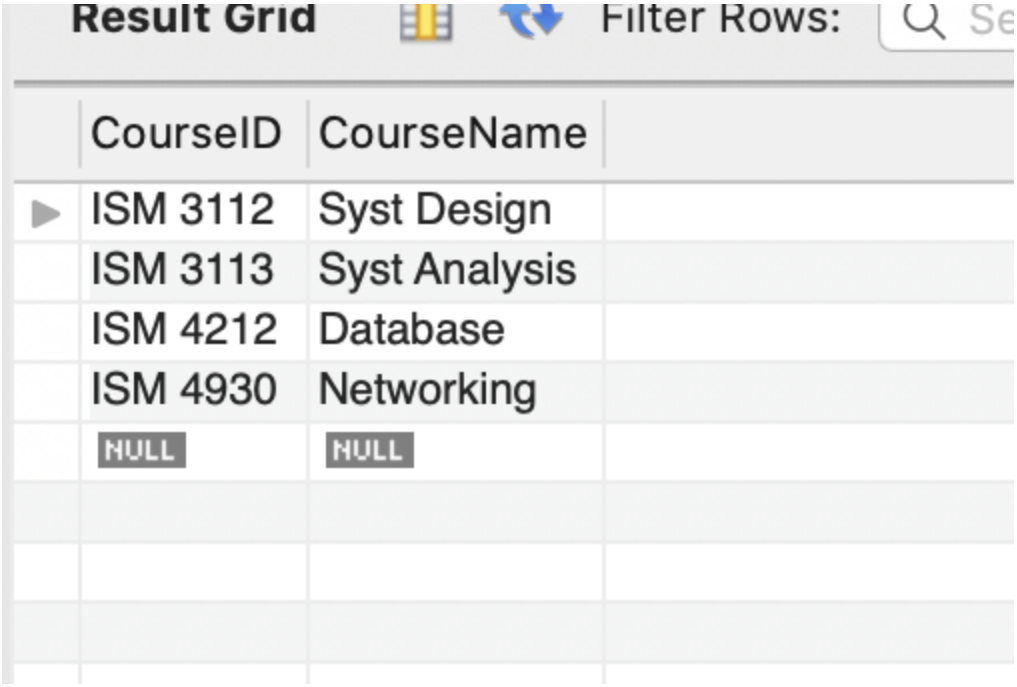
("ISM 3113", "Syst Analysis"),

("ISM 3112", "Syst Design"),

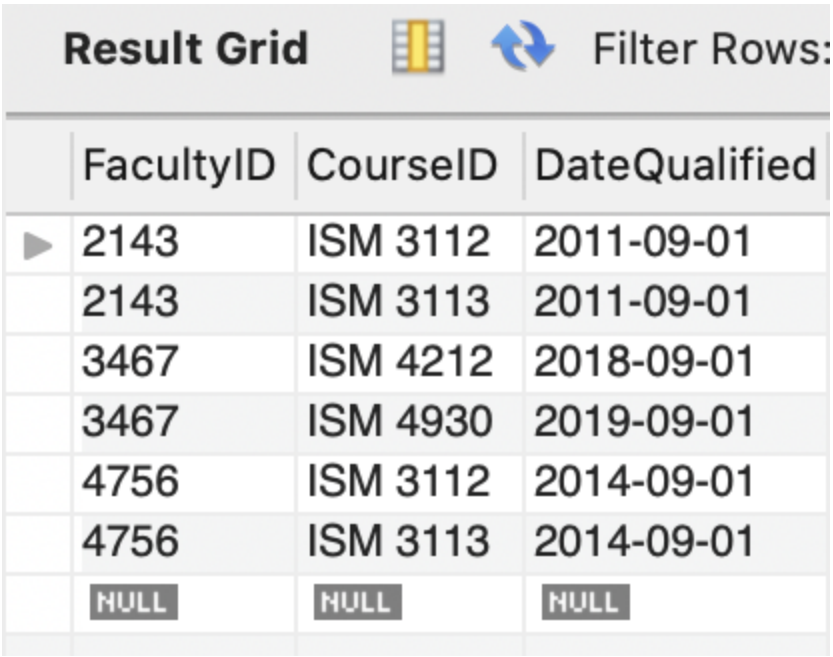
("ISM 4212", "Database"),

("ISM 4930", "Networking");

SELECT * FROM course;




```
INSERT INTO qualified

VALUES

("2143", "ISM 3112", "2011-09-01"),

("2143", "ISM 3113", "2011-09-01"),

("3467", "ISM 4212", "2018-09-01"),

("3467", "ISM 4930", "2019-09-01"),

("4756", "ISM 3113", "2014-09-01"),

("4756", "ISM 3112", "2014-09-01");


SELECT * FROM qualified;
```



Result Grid Filter Rows:

FacultyID	CourseID	DateQualified
2143	ISM 3112	2011-09-01
2143	ISM 3113	2011-09-01
3467	ISM 4212	2018-09-01
3467	ISM 4930	2019-09-01
4756	ISM 3112	2014-09-01
4756	ISM 3113	2014-09-01
NULL	NULL	NULL

INSERT INTO section

VALUES

("2712", "I-2021", "ISM 3113"),

("2713", "I-2021", "ISM 3113"),

("2714", "II-2021", "ISM 4212"),

("2715", "II-2021", "ISM 4930");

SELECT * FROM section;

Result Grid Filter Rows:

SectionNo	Semester	CourseID
2712	I-2021	ISM 3113
2713	I-2021	ISM 3113
2714	II-2021	ISM 4212
2715	II-2021	ISM 4930
NULL	NULL	NULL

```
INSERT INTO registration
VALUES
("38214", "2714"),

("54907", "2714"),

("54907", "2715"),

("66324", "2713");

SELECT * FROM registration;
```

StudentID	SectionNo
66324	2713
38214	2714
54907	2714
54907	2715
NULL	NULL

1.3)

```
CREATE VIEW studentcourse AS
        SELECT s.StudentName, se.Semester, se.CourseID
    FROM student s JOIN section se JOIN registration r
    ON s.StudentID = r.StudentID
    AND r.SectionNo = se.SectionNo;

SELECT * FROM studentcourse;
```

1.4)

a)

```
INSERT INTO student (StudentID, StudentName) VALUES (65798 , "Lopez");
```

b)

```
DELETE FROM student
WHERE StudentID = 65798 AND StudentName = "Lopez";
```

c)

```
UPDATE course SET
CourseName = "Introduction to Relational Databases"
WHERE CourseID = "ISM 4212";
```

d)

```
ALTER TABLE student
        RENAME COLUMN StudentName TO StudentLastName;
```

e)

```
CREATE TABLE qualified
(
FacultyID INT,
CourseID CHAR(8),
DateQualified DATE,
PRIMARY KEY (FacultyID, CourseID),
FOREIGN KEY (CourseID) REFERENCES course (CourseID)
        ON UPDATE CASCADE
```

```
        ON DELETE CASCADE,
FOREIGN KEY (FacultyID) REFERENCES faculty (FacultyID)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
CONSTRAINT CHK_DateQualified CHECK (DateQualified >= "2011-09-01")
);
```

Question 2:

2.1)

```
CREATE DATABASE assignment2final2;
USE assignment2final2;

CREATE TABLE TUTOR
(
TutorID INT,
CertDate DATE,
Status1 varchar(9),
PRIMARY KEY (TutorID)
);

CREATE TABLE STUDENT
(
StudentID INT,
Group1 INT,
Read1 FLOAT,
PRIMARY KEY (StudentID)
);

CREATE TABLE MATCH_HISTORY
(
MatchID INT,
TutorID INT,
StudentId INT,
StartDate DATE,
EndDate DATE,
```

```
PRIMARY KEY (MatchID),
FOREIGN KEY (TutorID) REFERENCES TUTOR (TutorID)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
FOREIGN KEY (StudentID) REFERENCES STUDENT (StudentID)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);
```

2.2)

TutorID	CertDate	Status1
100	2021-01-05	Active
101	2021-01-05	Temp Stop
102	2021-01-05	Dropped
103	2021-05-22	Active
104	2021-05-22	Active
105	2021-05-22	Temp Stop
106	2021-05-22	Active
NULL	NULL	NULL

StudentID	Group1	Read1
3000	3	2.3
3001	2	5.6
3002	3	1.3
3003	1	3.3
3004	2	2.7
3005	4	4.8
3006	3	7.8
3007	4	1.5
NULL	NULL	NULL

MatchID	TutorID	StudentId	StartDate	EndDate
1	100	3000	2021-01-20	NULL
2	101	3001	2021-01-15	2021-05-15
3	102	3002	2021-02-10	2021-03-01
4	106	3003	2021-05-28	NULL
5	103	3004	2021-06-01	2021-06-15
6	104	3005	2021-06-01	2021-06-28
7	104	3006	2021-06-01	NULL
NULL	NULL	NULL	NULL	NULL

2.3)

a)

b)

```
ALTER TABLE MATCH_HISTORY
        RENAME TO MATCH_HIST;
```

c)

```
ALTER TABLE TUTOR
MODIFY certDate DATE NOT NULL;
ALTER TABLE STUDENT
MODIFY Read1 FLOAT NOT NULL;
```

d)

```
CREATE INDEX fgroup ON STUDENT (Group1);
SHOW INDEX FROM STUDENT;
```

```
CREATE TABLE TUTOR
(
TutorID INT,
CertDate DATE,
Status1 varchar(9),
PRIMARY KEY (TutorID)
);

SELECT * FROM TUTOR;


INSERT INTO TUTOR VALUES
(100, "2021-01-05", "Active"),

(101, "2021-01-05", "Temp Stop"),
```

(102, "2021-01-05", "Dropped"),

(103, "2021-05-22", "Active"),

(104, "2021-05-22", "Active"),

(105, "2021-05-22", "Temp Stop"),

(106, "2021-05-22", "Active");

```
CREATE TABLE STUDENT
(
StudentID INT,
Group1 INT,
Read1 FLOAT,
PRIMARY KEY (StudentID)
);

SELECT * FROM STUDENT;

INSERT INTO STUDENT VALUES
(3000, 3, 2.3),
(3001, 2, 5.6),
(3002, 3, 1.3),
(3003, 1, 3.3),
(3004, 2, 2.7),
(3005, 4, 4.8),
(3006, 3, 7.8),
(3007, 4, 1.5);

CREATE TABLE MATCH_HISTORY
(
MatchID INT,
TutorID INT,
StudentId INT,
```

```
StartDate DATE,
EndDate DATE,
PRIMARY KEY (MatchID),
FOREIGN KEY (TutorID) REFERENCES TUTOR (TutorID)
        ON UPDATE CASCADE
        ON DELETE RESTRICT,
FOREIGN KEY (StudentID) REFERENCES STUDENT (StudentID)
        ON UPDATE CASCADE
        ON DELETE RESTRICT
);

INSERT INTO MATCH_HISTORY VALUES
(1, 100, 3000, "2021-01-20", NULL ),
(2, 101, 3001, "2021-01-15", "2021-05-15"),
(3, 102, 3002, "2021-02-10", "2021-03-01"),
(4, 106, 3003, "2021-05-28", NULL),
(5, 103, 3004, "2021-06-01", "2021-06-15"),
(6, 104, 3005, "2021-06-01", "2021-06-28"),
(7, 104, 3006, "2021-06-01", NULL);

SELECT * FROM MATCH_HISTORY;


#2.3a

CREATE TABLE MATCH_HISTORY
(
MatchID INT,
TutorID INT,
StudentId INT,
StartDate DATE,
EndDate DATE,
PRIMARY KEY (MatchID),
FOREIGN KEY (TutorID) REFERENCES TUTOR (TutorID)
        ON UPDATE CASCADE
        ON DELETE RESTRICT,
FOREIGN KEY (StudentID) REFERENCES STUDENT (StudentID)
        ON UPDATE CASCADE
        ON DELETE RESTRICT
);
```

 # A

```
ALTER TABLE Match_History
        ADD CONSTRAINT TutorID
                FOREIGN KEY (TutorID) REFERENCES Tutor(TutorID)
                        ON DELETE RESTRICT
                        ON UPDATE CASCADE,
        ADD CONSTRAINT StudentID
                FOREIGN KEY (StudentID) REFERENCES Student(StudentID)
                        ON DELETE RESTRICT
                        ON UPDATE CASCADE;
```

Hi arash
Hi diek

#b

```
ALTER TABLE MATCH_HISTORY
        RENAME TO MATCH_HIST;
```

#C

```
ALTER TABLE TUTOR
MODIFY certDate DATE NOT NULL;

ALTER TABLE STUDENT
MODIFY Read1 FLOAT NOT NULL;
```

#D

```
CREATE INDEX fgroup ON STUDENT (Group1);
SHOW INDEX FROM STUDENT;
```