

Assignment 2

Question 1

```
CREATE TABLE Course (  
    courseNo VARCHAR(255) NOT NULL,  
    courseTitle VARCHAR(255) NOT NULL,  
    AKTS INT NOT NULL,  
    PRIMARY KEY(courseNo)  
);
```

```
CREATE TABLE Section (  
    sectionNo INT NOT NULL,  
    courseNo VARCHAR(255) NOT NULL,  
    year INT NOT NULL,  
    maxRegistration INT NOT NULL,  
    PRIMARY KEY (sectionNo, courseNo, year),  
    FOREIGN KEY (courseNo) REFERENCES Course(courseNo),  
    UNIQUE(courseNo)  
);
```

```
CREATE TABLE Registration (  
    sectionNo INT NOT NULL,  
    courseNo VARCHAR(255) NOT NULL,  
    year INT NOT NULL,  
    studentNo BIGINT NOT NULL,  
    grade DECIMAL(3,2),  
    PRIMARY KEY (sectionNo, courseNo, year, studentNo),  
    FOREIGN KEY (sectionNo, courseNo, year) REFERENCES Section(sectionNo, courseNo, year),  
    FOREIGN KEY (studentNo) REFERENCES Student(studentNo)  
);
```

```
CREATE TABLE Student (  
    studentNo BIGINT NOT NULL,  
    studentName VARCHAR(255) NOT NULL,  
    studentAddress VARCHAR(255) NOT NULL,  
    PRIMARY KEY(studentNo)  
);
```

Question 2

/* I believe we can add Semester table which holds which courses are on which semester such that CSE204 -> Spring Term and CSE101T -> Fall Term

*/

```
CREATE TABLE Semester (  
    courseNo VARCHAR(10),  
    termNo VARCHAR(10)  
    CHECK (termNo IN ('S', 'F')),  
    /* S represents Spring and F represents Fall */  
    PRIMARY KEY (courseNo, termNo),  
    FOREIGN KEY (courseNo) REFERENCES Course(courseNo)  
);
```

Question 3

```
SELECT *  
FROM Course;
```

Question 4

```
SELECT studentName  
FROM Student;
```

Question 5

```
INSERT INTO Student  
VALUES (20220808011, 'Emre Topcu', "Muratpasa/Antalya");
```

Question 6

```
INSERT INTO Registration (sectionNo, courseNo, year, studentNo, grade)  
VALUES (1, 'CSE204', 2024, 20220808011, NULL);
```

Question 7

```
SELECT *  
FROM Section  
WHERE year = 2024;
```

Question 8

```
SELECT DISTINCT st.studentName,st.studentNo  
FROM Registration re  
INNER JOIN Student st ON re.studentNo = st.studentNo  
WHERE re.courseNo LIKE 'CSE%' AND re.year = 2024;
```

Question 9

```
DELETE FROM Student st  
WHERE EXISTS (  
    SELECT *  
    FROM Registration re  
    WHERE re.studentNo = st.studentNo  
    AND re.year <= 2020  
);
```

Question 10

```
UPDATE Section  
SET maxRegistration = maxRegistration * 1.25;
```

Question 11

```
SELECT s.studentName, s.studentAddress, SUM(c.AKTS) AS total_AKTS  
FROM Student s  
JOIN Registration r ON s.studentNo = r.studentNo  
JOIN Course c ON r.courseNo = c.courseNo  
WHERE r.year = 2023  
  
GROUP BY s.studentNo, s.studentName, s.studentAddress
```

Question 12

```
SELECT se.studentName, re.grade
FROM Registration re
INNER JOIN Student se ON re.studentNo = se.studentNo
WHERE se.courseNo = 'CSE204' AND re.year = 2023
ORDER BY re.grade DESC;
```

Question 13

```
SELECT co.courseNo, co.courseTitle
FROM Course co
LEFT JOIN Registration re ON
co.courseNo = re.courseNo AND re.year = 2024
WHERE re.courseNo IS NULL;
```

Question 14

```
SELECT st.studentName, re.courseNo, co.courseTitle
FROM Registration re
INNER JOIN Student st ON re.studentNo = st.studentNo
INNER JOIN Course co ON re.courseNo = co.courseNo
GROUP BY st.studentName, re.courseNo, co.courseTitle;
```

Question 15

```
SELECT st.studentName, st.studentAddress, co.courseNo, co.courseTitle
FROM Registration re
INNER JOIN Student st ON re.studentNo = st.studentNo
INNER JOIN Course co ON re.courseNo = co.courseNo
WHERE re.year = 2024
GROUP BY st.studentNo, st.studentName, st.studentAddress
HAVING SUM(co.AKTS) > 60,
ORDER BY st.studentNo, re.year DESC;
```

Question 16

UPDATE Registration

SET grade = 100

WHERE studentNo = 20220808011 AND courseNo = 'CSE204' AND year = 2024;

Question 17

SELECT st.studentName, st.studentAddress

FROM Student st

WHERE st.studentNo IN (

SELECT re.studentNo

FROM Registration re

WHERE re.courseNo = 'CSE101' AND re.year = 2023

EXCEPT (

SELECT re.studentNo

FROM Registration re

WHERE re.courseNo = 'CSE101' AND re.year < 2023)

);

Question 18

CREATE TABLE RegistrationArchive LIKE Registration;

INSERT INTO RegistrationArchive

SELECT * FROM Registration

WHERE year < 2020;

DELETE FROM Registration

WHERE year < 2020;

Question 19

```
CREATE VIEW FullCourses2024CSE AS
SELECT co.courseNo, co.courseTitle, se.sectionNo
FROM Course co
INNER JOIN Section se ON co.courseNo = se.courseNo
INNER JOIN Registration re ON se.sectionNo = re.sectionNo AND se.courseNo = re.courseNo
WHERE co.courseNo LIKE 'CSE%' AND se.year = 2024
GROUP BY co.courseNo, co.courseTitle, se.sectionNo
HAVING COUNT(DISTINCT re.studentNo) = se.maxRegistration;
```

Question 20

```
CREATE VIEW StudentsInCSE204_2023 AS
SELECT st.studentName, st.studentNo
FROM Student st
INNER JOIN Registration re ON st.studentNo = re.studentNo
WHERE re.courseNo = 'CSE204' AND re.year = 2023;
```

Question 21

```
CREATE VIEW RepeatStudentsCSE204_2024 AS
SELECT st.studentName, st.studentNo
FROM Student st
INNER JOIN Registration R2024 ON st.studentNo = R2024.studentNo
AND R2024.courseNo = 'CSE204' AND R2024.year = 2024
INNER JOIN Registration RPrev
ON st.studentNo = RPrev.studentNo AND RPrev.courseNo = 'CSE204'
AND RPrev.year < 2024;
```

Question 22

GRANT SELECT

ON FullCourses2024CSE

TO Dean, Chair WITH GRANT OPTION;

GRANT SELECT

ON StudentsInCSE204_2023

TO Dean, Chair WITH GRANT OPTION;

GRANT SELECT

ON RepeatStudentsCSE

TO Dean, Chair WITH GRANT OPTION;

Question 23

GRANT ALL

PRIVILEGES ON Registration

TO Secretary;

REVOKE INSERT, UPDATE, DELETE

ON Registration

FROM Secretary;

GRANT SELECT, INSERT, UPDATE

ON Registration

TO Secretary;