Q1 ANSWER:

--Creating the table as given in the HW2 description with an additional CHECK constraint that resolves the first issue.

CREATE TABLE ProjectRoomBookings(

roomNum INTEGER NOT NULL,

startTime INTEGER NOT NULL,

endTime INTEGER NOT NULL,

groupName VARCHAR(10) NOT NULL,

CONSTRAINT endTime CHECK(endTime>startTime),

PRIMARY KEY (roomNum, startTime)

)

--Second issue resolution.

The second issue can be resolved using the SQL Functions where in the code can be written such that for every insertion into the ProjectRoomBookings table, we have to check the if the existing roomNum is same as the new roomNum. If yes, we check if the existing groupName is same as the new GroupName. If yes, we may not have to check anything else, but, if no, then we have to check if the existing endTime is less than the new startTime. If yes, then we have to restrict such an entry into the table as it overlaps with the existing group in a room.

Q2 ANSWER:

--Creating table Enrollment as given in the HW2 description.

CREATE TABLE ENROLLMENT(S\_id INTEGER, ClassName VARCHAR(50), Grade VARCHAR(1));

--Inserting VALUES INTO the Enrollment table.

INSERT INTO ENROLLMENT VALUES(123,'Synthesis algorithms','A');

INSERT INTO ENROLLMENT VALUES(123,'EDM synthesis','B');

INSERT INTO ENROLLMENT VALUES(123,'MIDI controllers','B');

INSERT INTO ENROLLMENT VALUES(662,'SOUND MIXING','B');

INSERT INTO ENROLLMENT VALUES(662,'EDM synthesis','A');

INSERT INTO ENROLLMENT VALUES(662,'Electronic Music Fundamentals','A');

INSERT INTO ENROLLMENT VALUES(662,'MIDI controllers','B');

INSERT INTO ENROLLMENT VALUES(345,'MIDI controllers','A');

INSERT INTO ENROLLMENT VALUES(345,'Electronic Music Fundamentals','B');

INSERT INTO ENROLLMENT VALUES(345,'EDM synthesis','A');

INSERT INTO ENROLLMENT VALUES(555,'EDM synthesis','B');

INSERT INTO ENROLLMENT VALUES(555,'Electronic Music Fundamentals','B');

INSERT INTO ENROLLMENT VALUES(213,'Electronic Music Fundamentals','A');

--Main Query.

SELECT ClassName, COUNT(S\_id) AS Total

FROM Enrollment

GROUP BY ClassName

ORDER BY Total DESC;

------------------------------------------------------------------------------------------------------------

Q3 ANSWER:

--Creating the Project table as given the homework description.

CREATE TABLE PROJECT(P\_id VARCHAR(5), Step INTEGER, Status VARCHAR(1), PRIMARY KEY(P\_id, Step));

--Inserting the VALUES INTO the Project table.

INSERT INTO PROJECT VALUES('P333',3,'W');

INSERT INTO PROJECT VALUES('P333',2,'W');

INSERT INTO PROJECT VALUES('P333',1,'W');

INSERT INTO PROJECT VALUES('P333',0,'W');

INSERT INTO PROJECT VALUES('P201',0,'C');

INSERT INTO PROJECT VALUES('P201',1,'C');

INSERT INTO PROJECT VALUES('P100',0,'C');

INSERT INTO PROJECT VALUES('P100',1,'W');

INSERT INTO PROJECT VALUES('P100',2,'W');

INSERT INTO PROJECT VALUES('P444',0,'C');

INSERT INTO PROJECT VALUES('P444',1,'W');

--Using the below query the P444 gets updated and the it won't be retrieved using the main query.

--UPDATE PROJECT SET Status = 'C' WHERE P\_id = 'P444' AND Step = 1;

--Main Query

SELECT P\_id

FROM PROJECT

WHERE Status = 'C'

AND P\_id IN (SELECT P\_id FROM PROJECT WHERE Step = 0)

GROUP BY P\_id

HAVING COUNT(P\_id) = 1;

------------------------------------------------------------------------------------------------------------

Q4 ANSWER:

--Creating table for Instructor as given in the HW2 description.

CREATE TABLE INSTRUCTOR(I\_name VARCHAR(20), Course VARCHAR(50), Rate INTEGER, Strength INTEGER);

--Inserting values into the Instructor table.

INSERT INTO INSTRUCTOR VALUES('Dat','MIDI controllers',25,20);

INSERT INTO INSTRUCTOR VALUES('Dat','Electronic Music Fundamentals',25,20);

INSERT INTO INSTRUCTOR VALUES('Dat','EDM synthesis',25,15);

INSERT INTO INSTRUCTOR VALUES('Tad','MIDI controllers',23,15);

INSERT INTO INSTRUCTOR VALUES('Tad','Electronic Music Fundamentals',23,20);

INSERT INTO INSTRUCTOR VALUES('Tad','EDM synthesis',23,15);

INSERT INTO INSTRUCTOR VALUES('Thor','MIDI controllers',30,30);

INSERT INTO INSTRUCTOR VALUES('Thor','Electronic Music Fundamentals',30,20);

INSERT INTO INSTRUCTOR VALUES('Thor','EDM synthesis',30,15);

INSERT INTO INSTRUCTOR VALUES('Shawn','MIDI controllers',23,15);

INSERT INTO INSTRUCTOR VALUES('Shawn','Electronic Music Fundamentals',23,20);

INSERT INTO INSTRUCTOR VALUES('Shawn','EDM synthesis',23,15);

--Main Query.

SELECT MAX(Rate\*sum\_of\_class\_counts\*0.1) AS Bonus

FROM Instructor, (SELECT SUM(Strength) AS sum\_of\_class\_counts FROM INSTRUCTOR GROUP BY I\_name)

------------------------------------------------------------------------------------------------------------

Q5 ANSWER:

--Creating table for Instructors as given in HW2 description.

CREATE TABLE INSTRUCTORS(I\_name varchar(20), Course varchar(50));

--Creating an additional table for specific courses that we are looking for.

CREATE TABLE COURSE(CourseName varchar(50));

--Inserting values into Course Table.

INSERT INTO COURSE VALUES('Electronic Music Fundamentals');

INSERT INTO COURSE VALUES('MIDI controllers');

INSERT INTO COURSE VALUES('EDM synthesis');

--Inserting values into Instructors Table.

INSERT INTO INSTRUCTORS VALUES('Dat','Electronic Music Fundamentals');

INSERT INTO INSTRUCTORS VALUES('Dat','MIDI controllers');

INSERT INTO INSTRUCTORS VALUES('Dat','EDM synthesis');

INSERT INTO INSTRUCTORS VALUES('Aleph','MIDI controllers');

INSERT INTO INSTRUCTORS VALUES('Bit','EDM synthesis');

INSERT INTO INSTRUCTORS VALUES('CRC','Electronic Music Fundamentals');

INSERT INTO INSTRUCTORS VALUES('CRC','EDM synthesis');

INSERT INTO INSTRUCTORS VALUES('Emscr','EDM synthesis');

INSERT INTO INSTRUCTORS VALUES('Emscr','Electronic Music Fundamentals');

INSERT INTO INSTRUCTORS VALUES('Emscr','MIDI controllers');

INSERT INTO INSTRUCTORS VALUES('Emscr','abc');

INSERT INTO INSTRUCTORS VALUES('Tad','EDM synthesis');

INSERT INTO INSTRUCTORS VALUES('Tad','Electronic Music Fundamentals');

INSERT INTO INSTRUCTORS VALUES('Tad','abc');

INSERT INTO INSTRUCTORS VALUES('Tad','MIDI controllers');

INSERT INTO INSTRUCTORS VALUES('Kid','EDM synthesis');

INSERT INTO INSTRUCTORS VALUES('Kid','Electronic Music Fundamentals');

INSERT INTO INSTRUCTORS VALUES('Kid','abc');

INSERT INTO INSTRUCTORS VALUES('Kid','MIDI controllers');

--Main Query.

SELECT I\_name

FROM Instructors

WHERE COURSE IN (SELECT CourseName FROM COURSE)

AND COURSE <> ALL (SELECT COURSE FROM INSTRUCTORS where COURSE NOT IN (SELECT CourseName FROM COURSE))

GROUP BY I\_name HAVING COUNT(I\_name)>=3;