**STORED PROCEDURE**

DELIMITER //

DROP PROCEDURE IF EXISTS courseinfo;//

DELIMITER $$

CREATE PROCEDURE courseinfo(IN code CHAR(8), IN quarter CHAR(2), IN c\_year CHAR(4))

BEGIN

SELECT UoSName, Enrollment, MaxEnrollment, Name

FROM uosoffering inner join unitofstudy using(UoSCode)

inner join faculty on InstructorId = Id

WHERE UoSCode = code

and Semester = quarter

and Year = c\_year;

END$$

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

DELIMITER //

DROP PROCEDURE IF EXISTS withdrawStudent;//

DELIMITER $$

CREATE PROCEDURE withdrawStudent(IN sID INT, IN cID CHAR(8), IN quarter CHAR(2), IN c\_year CHAR(4))

BEGIN

DELETE FROM transcript

WHERE StudId = sID

and UoSCode = cID

and Semester = quarter

and Year = c\_year

and Grade is NULL;

UPDATE uosoffering

SET Enrollment = Enrollment - 1

WHERE UoSCode = cID

and Semester = quarter

and Year = c\_year;

END$$

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

DELIMITER //

DROP PROCEDURE IF EXISTS getAvailableCourses;//

DELIMITER $$

CREATE PROCEDURE getAvailableCourses(IN sid INT, IN quarter CHAR(2), IN c\_year CHAR(4))

BEGIN

SELECT UoSCode, UoSName, Enrollment, MaxEnrollment

FROM uosoffering left outer join unitofstudy using(UoSCode)

WHERE Year = c\_year and Semester = quarter and UoSCode NOT IN

(SELECT UoSCode

FROM transcript

WHERE StudId = sid and Semester = quarter and Year = c\_year);

END$$

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

DELIMITER //

DROP PROCEDURE IF EXISTS prerequisitesCheck;//

DELIMITER $$

CREATE PROCEDURE prerequisitesCheck(IN cid CHAR(8), IN sid INT)

BEGIN

SELECT PrereqUoSCode, UoSName

FROM requires left outer join unitofstudy using(UoSCode)

WHERE UoSCode = cid and prereqUoSCode NOT IN

(SELECT UoSCode

FROM transcript

WHERE StudId = sid and (Grade = “CR” or Grade = “D” or Grade = “P”));

END$$

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

DELIMITER //

DROP PROCEDURE IF EXISTS addCourse;//

DELIMITER $$

CREATE PROCEDURE addCourse(IN cid CHAR(8), IN sid INT, IN quarter CHAR(2), IN c\_year CHAR(4))

BEGIN

INSERT INTO transcript

VALUES (sid, cid, quarter, c\_year, NULL);

UPDATE uosoffering

SET enrollment = enrollment + 1

WHERE UoScode = cid AND Semester = quarter AND year = c\_year;

END$$

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

**TRIGGER**

DELIMITER //

drop trigger if exists enrollment\_update\_constraint//

create trigger enrollment\_update\_constraint

after update on uosoffering

for each row

begin

declare enrollmt int;

declare maxenrollmnt int;

set @message = "";

set enrollmt = (select Enrollment as enrollmt from uosoffering where uoscode = new.uoscode and semester =new.semester and year = new.year);

set maxenrollmnt = (select maxenrollment as maxenrollmnt from uosoffering where uoscode = new.uoscode and semester =new.semester and year = new.year);

if(enrollmt < (maxenrollmnt/2)) then

set @message = "Enrolled value for this course is 50% below max enrolled value";

end if;

end; //

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