ss1

create table **faculty** (facultyID int not null primary key, facultyName char(3))

insert into faculty values (1,'FOE')

insert into faculty values (2,'FCI')

insert into faculty values (3,'FCM')

insert into faculty values (4,'FOM')

select \* from faculty

ss2

create table **lecturer** (lecturerID int not null generated always as identity (start with 1 increment by 1) primary key, lecturerName varchar(30), facultyID int, foreign key (facultyID) references faculty(facultyID))

insert into lecturer (lecturername, facultyid) values ('Ian',1)

insert into lecturer (lecturername, facultyid) values ('Tan',1)

insert into lecturer (lecturername, facultyid) values ('Ah Lam',1)

insert into lecturer (lecturername, facultyid) values ('Ahmad',1)

insert into lecturer (lecturername, facultyid) values ('Kannan',2)

insert into lecturer (lecturername, facultyid) values ('Razak',2)

insert into lecturer (lecturername, facultyid) values ('Chandra',2)

insert into lecturer (lecturername, facultyid) values ('Bhavata',2)

insert into lecturer (lecturername, facultyid) values ('Jun Ming',3)

insert into lecturer (lecturername, facultyid) values ('Patrick',3)

insert into lecturer (lecturername, facultyid) values ('Amelia',3)

insert into lecturer (lecturername, facultyid) values ('Haikal',3)

insert into lecturer (lecturername, facultyid) values ('Lai',4)

insert into lecturer (lecturername, facultyid) values ('Michael',4)

insert into lecturer (lecturername, facultyid) values ('Shukri',4)

insert into lecturer (lecturername, facultyid) values ('Divagar',4)

ss3

select \* from lecturer

ss4

create table **student**(studentID int not null generated always as identity (start with 1 increment by 1) primary key, studentName varchar(20), studentDOB date, gpa decimal(5,2) with default 0.00, coursepassed int with default 0,facultyID int,foreign key (facultyID) references faculty(facultyID))

insert into student (studentname, studentdob, facultyid) values ('Hakim','1996-08-08',1)

insert into student (studentname, studentdob, facultyid) values ('Luqman','1996-03-21',1)

insert into student (studentname, studentdob, facultyid) values ('Diva','1997-09-09',1)

insert into student (studentname, studentdob, facultyid) values ('Kirthi','1995-11-11',1)

insert into student (studentname, studentdob, facultyid) values ('Jun','1995-03-02',1)

insert into student (studentname, studentdob, facultyid) values ('Chong','1996-04-04',2)

insert into student (studentname, studentdob, facultyid) values ('Lee','1998-01-01',2)

insert into student (studentname, studentdob, facultyid) values ('Wei','1996-02-02',2)

insert into student (studentname, studentdob, facultyid) values ('Vasan','1997-03-02',2)

insert into student (studentname, studentdob, facultyid) values ('Alif','1992-05-05',2)

insert into student (studentname, studentdob, facultyid) values ('Amira','1997-07-09',3)

insert into student (studentname, studentdob, facultyid) values ('Mau Jun','1998-03-08',3)

insert into student (studentname, studentdob, facultyid) values ('Firdaus','1994-12-12',3)

insert into student (studentname, studentdob, facultyid) values ('Chao','1995-11-11',3)

insert into student (studentname, studentdob, facultyid) values ('Shamba','1997-10-10',3)

insert into student (studentname, studentdob, facultyid) values ('Tharun','1995-09-09',4)

insert into student (studentname, studentdob, facultyid) values ('Farah','1996-06-06',4)

insert into student (studentname, studentdob, facultyid) values ('Lisa','1994-04-04',4)

insert into student (studentname, studentdob, facultyid) values ('Chloe','1996-06-04',4)

insert into student (studentname, studentdob, facultyid) values ('Brendan','1998-06-08',4)

ss5

select \* from student

ss6

create table **course** (courseid int not null generated always as identity (start with 1 increment by 1) primary key, coursename varchar(40), courseprice decimal(6,2), lecturerid int, foreign key (lecturerid) references lecturer(lecturerid))

insert into course (coursename, courseprice, lecturerid) values ('Calculus',699.99,1)

insert into course (coursename, courseprice, lecturerid) values ('Physics',699.99,2)

insert into course (coursename, courseprice, lecturerid) values ('Engineering Maths',549.99,3)

insert into course (coursename, courseprice, lecturerid) values ('Algebra',449.49,4)

insert into course (coursename, courseprice, lecturerid) values ('Programming Concept',499.99,5)

insert into course (coursename, courseprice, lecturerid) values ('Programming Fundamentals',549.99,6)

insert into course (coursename, courseprice, lecturerid) values ('Algorithms',399.99,7)

insert into course (coursename, courseprice, lecturerid) values ('Software Development',749.99,8)

insert into course (coursename, courseprice, lecturerid) values ('3D Animation',999.99,9)

insert into course (coursename, courseprice, lecturerid) values ('3D Illustration',849.99,10)

insert into course (coursename, courseprice, lecturerid) values ('Photography',949.49,11)

insert into course (coursename, courseprice, lecturerid) values ('Storyboard Creation',349.49,12)

insert into course (coursename, courseprice, lecturerid) values ('Economics',700.00,13)

insert into course (coursename, courseprice, lecturerid) values ('Accounting',1299.00,14)

insert into course (coursename, courseprice, lecturerid) values ('Management',599.49,15)

insert into course (coursename, courseprice, lecturerid) values ('Financial Engineering',800.99,16)

insert into course (coursename, courseprice, lecturerid) values ('Professional Development',299.99,4)

insert into course (coursename, courseprice, lecturerid) values ('Entrepreneurship',249.49,8)

insert into course (coursename, courseprice, lecturerid) values ('Soft Skill Development',250.00,12)

insert into course (coursename, courseprice, lecturerid) values ('Speaking and Writing in English',349.49,16)

ss7

select \* from course

ss8

create table **enroll** (enrollid int not null generated always as identity (start with 1 increment by 1) primary key, studentid int, courseid int, foreign key (studentid) references student(studentid), foreign key (courseid) references course(courseid))

insert into enroll (studentid, courseid) values (1,1)

insert into enroll (studentid, courseid) values (1,3)

insert into enroll (studentid, courseid) values (2,2)

insert into enroll (studentid, courseid) values (2,4)

insert into enroll (studentid, courseid) values (3,3)

insert into enroll (studentid, courseid) values (3,2)

insert into enroll (studentid, courseid) values (4,4)

insert into enroll (studentid, courseid) values (4,17)

insert into enroll (studentid, courseid) values (5,1)

insert into enroll (studentid, courseid) values (5,18)

insert into enroll (studentid, courseid) values (6,5)

insert into enroll (studentid, courseid) values (6,7)

insert into enroll (studentid, courseid) values (7,6)

insert into enroll (studentid, courseid) values (7,8)

insert into enroll (studentid, courseid) values (8,7)

insert into enroll (studentid, courseid) values (8,6)

insert into enroll (studentid, courseid) values (9,8)

insert into enroll (studentid, courseid) values (9,19)

insert into enroll (studentid, courseid) values (10,5)

insert into enroll (studentid, courseid) values (10,20)

insert into enroll (studentid, courseid) values (11,9)

insert into enroll (studentid, courseid) values (11,12)

insert into enroll (studentid, courseid) values (12,10)

insert into enroll (studentid, courseid) values (12,11)

insert into enroll (studentid, courseid) values (13,11)

insert into enroll (studentid, courseid) values (13,10)

insert into enroll (studentid, courseid) values (14,12)

insert into enroll (studentid, courseid) values (14,17)

insert into enroll (studentid, courseid) values (15,9)

insert into enroll (studentid, courseid) values (15,18)

insert into enroll (studentid, courseid) values (16,13)

insert into enroll (studentid, courseid) values (16,15)

insert into enroll (studentid, courseid) values (17,14)

insert into enroll (studentid, courseid) values (17,16)

insert into enroll (studentid, courseid) values (18,15)

insert into enroll (studentid, courseid) values (18,13)

insert into enroll (studentid, courseid) values (19,16)

insert into enroll (studentid, courseid) values (19,19)

insert into enroll (studentid, courseid) values (20,13)

insert into enroll (studentid, courseid) values (20,20)

ss9

select \* from enrol

ss10

create table **passed\_course** (passid int not null generated always as identity (start with 1 increment by 1) primary key, studentid int, courseid int, gpa decimal(5,2), foreign key (studentid) references student(studentid), foreign key (courseid) references course(courseid))

ss11 (trigger)

**create trigger updatestudent** after insert on passed\_course referencing new as n for each row mode db2sql update student set gpa = gpa + n.gpa, coursepassed = coursepassed + 1 where n.studentid = student.studentid

ss12

insert into passed\_course (studentid, courseid, gpa) values (1,2,3.50)

insert into passed\_course (studentid, courseid, gpa) values (3,4,2.60)

insert into passed\_course (studentid, courseid, gpa) values (4,3,3.80)

insert into passed\_course (studentid, courseid, gpa) values (6,8,3.20)

insert into passed\_course (studentid, courseid, gpa) values (10,6,2.90)

insert into passed\_course (studentid, courseid, gpa) values (13,12,2.70)

insert into passed\_course (studentid, courseid, gpa) values (13,17,3.90)

insert into passed\_course (studentid, courseid, gpa) values (17,13,4.00)

insert into passed\_course (studentid, courseid, gpa) values (17,18,3.20)

insert into passed\_course (studentid, courseid, gpa) values (20,14,2.90)

ss13

select \* from passed\_course

ss14

select \* from student

to show that the trigger werks

ss15 (i. aggregate)

select sum(courseprice) as totalfee from course where courseid in (select courseid from enroll where studentid = 1)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ss16 (ii. GROUP BY & HAVING)

select facultyname, count(studentid) as totalstudents from student, facultywhere student.facultyid = faculty.facultyid **group by** facultyname **having** count(studentid) > 0

ss17 (iv. stored procedure)

create procedure cgpa() begin declare gpapercourse varchar(150); declare c cursor with return for s; set gpapercourse = 'select studentid, gpa/coursepassed as cgpa from student where studentid in (select studentid from student where coursepassed >0 and gpa >0)'; prepare s from gpapercourse; open c; end

call cgpa()

ss18 (view)

create **view activestudent** as select studentid, studentname,gpa/coursepassed as cgpa, facultyname from student, faculty where coursepassed <> 0 and student.facultyid = faculty.facultyid

ss19

select \* from activestudent

ss20 (sub/nested queries)

select studentid, studentname from student where studentid in (select studentid from passed\_course) order by gpa/coursepassed desc limit 5

this will show the top 5 highest cgpa students