**ACTIVITY RETRIEVING DATA**

* Obtain all information on the Students not attending course 1
* Obtain the first name, surname and Date of Birth for the student with the email address: val.bolger@example.com
* Obtain a list of the modules which have the subject Economics
* Obtain a list of class numbers and their dates which are scheduled before 21st September 2020

select \* from uni.student

where courseid != 1

order by CourseID

;

select forenames, surname, dateofbirth

from uni.student

where emailaddress = 'val.bolger@example.com'

;

select \* from uni.module

where subject = 'Economics'

;

select ClassID, CDate from uni.schedule

where cdate < '2020-09-21'

order by CDate

;

**ACTIVITY INSERTING DATA**

* Insert a record for a new course named Deep-Space Radar Telemetry
* write an insert to insert records for the following modules:
  + String Theory
  + Exotic Matter
  + Harnessing the Einstein-Rosen Bridge
  + Supercollision and miniature Black Holes

(these modules are worth 20 credits each, at level 6 and are taught on the Physics Course)

* Using the information from the previous example and the LecturerID of 6, create a class for each new module.

insert into uni.course(coursename)

values('Deep-Space Radar Telemetry')

;

insert into

uni.module(modulename,subject,level,courseid,credits)

values

('String Theory','Physics',6,6,20),

('Exotic Matter','Physics',6,6,20),

('Harnessing the Einstein-Rosen Bridge','Physics',6,6,20),

('Supercollision and minature Black Holes','Physics',6,6,20)

;

insert into

uni.class(ClassID, lecturerid, moduleid)

values

(95,6,105),

(96, 6,106),

(97, 6,107),

(98, 6,108)

;

;

--there are technically two possible correct answers to this question

**insert into uni.student(forenames, surname, dateofbirth, courseid)**

**values**

**('John','Jones','10/06/2000',2),**

**('John','Jones','10/06/2000',2),**

**('Catherine','Donovan','01/10/2002',5),**

**('Claire','Farron','12/02/2001',5),**

**('Claire','Farron','12/02/2001',5),**

**('Claire','Farron','12/02/2001',5)**

**;**

**--second answer**

**insert into uni.student(forenames, surname, dateofbirth, courseid)**

**values**

**('John','Jones','10/06/2000',2),**

**('Catherine','Donovan','01/10/2002',5),**

**('Claire','Farron','12/02/2001',5)**

**ACTIVITY CREATING CALCULATIONS**

* Count how students are enrolled overall
* Calculate the sum of full time fees for every full-time course
* Identify the cost of the least and most expensive course
* Calculate the average cost of all part time courses
* Calculate the fee of each full time course after applying (subtracting) the scholarship discount
* Extension:
* Select only the course number of the cheapest full-time course
* Find cost of the most expensive course after applying the scholarship discount
* Count the number of applications for History courses made between 01/03/2020 and 30/08/2020

select count(\*) from uni.student

;

select sum(fulltimefee) from uni.fees

;

select min(fulltimefee) from uni.fees

;

select max(fulltimefee) from uni.fees

;

select avg(parttimefee) from uni.fees

;

select courseid, fulltimefee-scholarshipdiscount

from uni.fees

;

select courseid from uni.fees where fulltimefee = (select min(fulltimefee) from uni.fees)

;

select courseid, fulltimefee-scholarshipdiscount

from uni.fees

order by fulltimefee-scholarshipdiscount desc

;

select \* from uni.application

where courseappliedfor = 11

and dateofapplication

between '2020-03-01' and '2020-08-30'

**ACTIVITY WORKING WITH LISTS**

* Obtain all the course information for courses with the CourseIDs of 1,3,5 and 7
* Obtain a list of all modules taught on courses which have a Full Time Fee greater than 9000
* Obtain a list of classes for modules taught on courses which have a Full Time Fee greater than 9000
* Extension: Find a list of studentIDs for the latest class on the most expensive course HINT: You may need an AND in your WHERE clause to solve this

select \* from uni.course

where courseid in (1,3,5,7)

;

select \* from modules

where courseid in

(select courseid

from uni.fees

where fulltimefee > 9000)

;

select \* from uni.class where moduleid in

(select moduleid from uni.module

where courseid in

(select courseid

from uni.fees

where fulltimefee > 9000))

;

--extension (this is a real stretch task)

select studentID from uni.registration where classid in

(select classid from uni.schedule where cdate =

(select max(cdate) from uni.schedule where classid in

(select classid from uni.class

where moduleid in

(select moduleid from uni.module

where courseid in

(select courseid from uni.fees

where fulltimefee =

(select max(fulltimefee) from uni.fees)))))

and classid in

(select classid from uni.class

where moduleid in

(select moduleid from uni.module

where courseid in

(select courseid from uni.fees

where fulltimefee =

(select max(fulltimefee) from uni.fees))))

)

;

**ACTIVITY : INNER JOIN**

* Obtain a list of Students and the name of the Courses they are studying
* Obtain a list of course names, full time fees and part time fees for each course
* Obtain a list of classIDs for the Economics Course and the modules they relate to

select student.\*, course.coursename

from uni.student inner join uni.course

on student.courseid = course.courseid

;

select course.coursename, fees.fulltimefee, fees.parttimefee

from uni.course inner join uni.fees

on course.courseid = fees.courseid

order by course.coursename

;

select \*

from uni.course

inner join uni.module on course.courseid = module.courseid

inner join uni.class on class.moduleid = module.moduleid

where course.courseid = 1

;