**ACTIVITY BETWEEN TWO NULLS**

* **Obtain a list of applications where the CourseID is unknown**
* **Obtain a list of students where their CourseID is not unknown**
* **Obtain a list of students whom were born in the month of June 2002**
* **Obtain a list of applications where CourseID is unknown and the applications where made between 01/04/2020 and 31/07/2020**

select \* from uni.application

where courseappliedfor is null

;

select \* from uni.student

where courseid is not null

;

select \* from uni.student

where dateofbirth between '2002-06-01' and '2020-06-30'

;

select \* from uni.application

where courseappliedfor is null

and dateofapplication between '2020-04-01' and '2020-07-31'

**ACTIVITY GROUPING FUNCTIONS**

* Obtain the number of modules which are assigned to each course
* Retrieve Information on the number of successful applications per course
* Find the average MembershipFee of Student Clubs by the ID of the Staff member (Lecturer) supervising it
* Find the Sum total of Joining Fees for all active clubs by Staff Member supervising them

select courseid, count(\*)

from uni.module

group by courseid

;

select courseappliedfor, count(\*)

from uni.application

group by courseappliedfor

;

select supervisingstaff, avg(membershipfee) from uni.club

group by supervisingstaff

;

select supervisingstaff, sum(joiningfee) from uni.club

where active = 1

group by supervisingstaff

;

**ACTIVITY: ADVANCED JOINS**

* Obtain a list of all modules and the names of any courses they may be taught on (include modules without courses)
* Obtain a list of students along with any related application numbers if they have them
* Obtain the Class ID, Class Date and Feedback score of the latest class scheduled for each Class ID

select course.coursename, module.modulename

from uni.module left join uni.course

on course.courseid = module.courseid

;

select student.\*, application.applicationid

from uni.student left join uni.application

on student.studentid = application.studentid

;

select schedule.classid, a.mxdate, schedule.feedbackscore

from uni.schedule inner join

(select classid, max(cdate) mxdate

from uni.schedule

group by classid) a

on a.classid = schedule.classid and schedule.cdate = a.mxdate