Apellidos: González Mahagamage

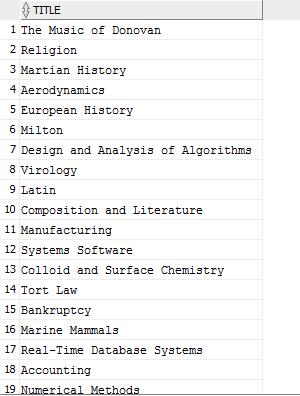
Nombre: Iván

UO: 239795

**RI - EJERCICIO CONSULTAS SQL**

1. Basic SQL
   1. Find the names of courses in Computer science department which have 3 credits  
      SELECT DISTINCT c.TITLE

FROM COURSE c  
WHERE c.CREDITS = 3;

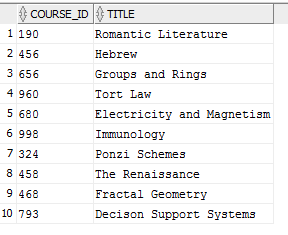


* 1. For the student with ID 12345 (or any other value), find all course\_id and title of all courses registered for by the student and then show the total number of credits for such courses (taken by that student). Don't display the tot\_creds value from the student table, you should use SQL aggregation on courses taken by the student.   
     (He usado el ID = ‘1000’ para tener algún resultado)  
     SELECT c.COURSE\_ID, c.TITLE

FROM COURSE c, STUDENT s

WHERE c.DEPT\_NAME = s.DEPT\_NAME

AND s.ID = '1000';



SELECT SUM(c.CREDITS) AS Creditos\_totales

FROM COURSE c, STUDENT s

WHERE c.DEPT\_NAME = s.DEPT\_NAME

AND s.ID = '1000'

GROUP BY s.ID;



* 1. Display the IDs and names of all instructors who have never taught a couse (Notesad1) Oracle uses the keyword minus in place of except; (2) interpret "taught" as "taught or is scheduled to teach")  
     SELECT i.ID

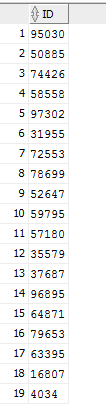
FROM INSTRUCTOR i

WHERE i.ID NOT IN (

SELECT t.ID

FROM TEACHES t

);

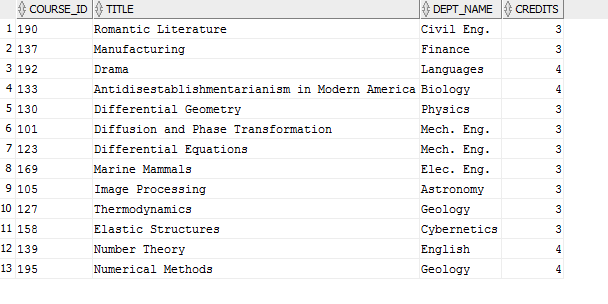


1. Intermediate SQL
   1. Find all courses whose identifier starts with the string "CS-1"  
      He tenido que cambiar el string porque con la VERSIÓN” LARGA” DE LA BASE DE DATOS no hay ninguno que empieza por “CS-1”.

SELECT DISTINCT \*

FROM COURSE c

WHERE c.COURSE\_ID LIKE 'CS-1%';



* 1. Some of you may have noticed that the tot\_creds value for students did not match the credits from courses they have taken. Write and execute query to update tot\_creds based on the credits passed, to bring the database back to consistency. (This query is provided in the book/slides.)  
     UPDATE STUDENT s set s.TOT\_CRED =

(SELECT SUM(c.credits) AS TOTAL

FROM STUDENT s2, TAKES t, COURSE c

WHERE s.ID = t.ID

AND t.COURSE\_ID = c.COURSE\_ID

AND s.id = s2.ID);

1. Advanced SQL
   1. Grades are mapped to a grade point as follows: A:10, B:8, C:6, D:4 and F:0. Create a table to store these mappings, and write a query to find the average grade of each student, using this table. Make sure students who have not got a non-null grade in any course are displayed with an average of null.  
      create table points(

letra VARCHAR(2) NOT NULL,

numero NUMERIC(2) NOT NULL,

CONSTRAINT points\_pk PRIMARY KEY (letra, numero)

);

insert into points values('A', 10);

insert into points values('A-', 9);

insert into points values('B+', 8.5);

insert into points values('B', 8);

insert into points values('B-', 7);

insert into points values('C+', 6.5);

insert into points values('C', 6);

insert into points values('C-', 5);

insert into points values('D', 4);

insert into points values('F', 0);

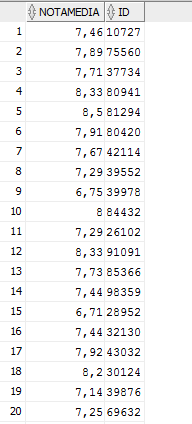
SELECT ROUND(AVG(p.NUMERO),2) AS notaMedia, t.ID

FROM takes t, points p

WHERE t.GRADE = p.LETRA

AND t.Grade != 'F'

GROUP BY t.ID;



* 1. Find all rooms that have been assigned to more than one section at the same time. Display the rooms along with the assigned sections; I suggest you use a with clause or a view to simplify this query.  
       
     CREATE VIEW vistaRepaso AS

SELECT DISTINCT c.ROOM\_NUMBER

FROM CLASSROOM c, SECTION s1, SECTION s2, TIME\_SLOT t1, TIME\_SLOT t2

WHERE c.BUILDING = s1.BUILDING

AND c.ROOM\_NUMBER = s1.ROOM\_NUMBER

AND c.BUILDING = s2.BUILDING

AND c.ROOM\_NUMBER = s2.ROOM\_NUMBER

AND s1.COURSE\_ID != s2.COURSE\_ID

AND s1.TIME\_SLOT\_ID = s2.TIME\_SLOT\_ID;

