1. Write the following SQL queries. (5 points each)
   1. Find the titles of courses in the Comp. Sci. department that have 3 credits.

SELECT DISTINCT title  
FROM course  
WHERE credits = 3

* 1. Find the IDs of all students who were taught by an instructor named Einstein; make sure there are no duplicates in the result.

SELECT DISTINCT student.id  
FROM student natural join takes, instructor natural join teaches  
WHERE (takes.course\_id, takes.sec\_id, takes.semester, takes.year) = (teaches.course\_id, teaches.sec\_id, teaches.semester, teaches.year)  
AND instructor.name = 'Einstein'

* 1. Find the highest salary of any instructor.

SELECT max(salary)  
FROM instructor

* 1. Find all instructors earning the highest salary (there may be more than one with the same salary).

SELECT name

FROM instructor

WHERE salary = (SELECT max(salary) from instructor)

* 1. Find the enrollment of each section that was offered in Fall 2009. Remember just a number is meaningless unless you include what it refers to.

SELECT course\_id, sec\_id, count(id) enrollment  
FROM takes  
WHERE (semester, year) = ('Fall', 2009)  
GROUP BY course\_id, sec\_id

* 1. Find the maximum enrollment, across all sections, in Fall 2009.

SELECT max(e.enrollment)  
FROM (SELECT count(id) enrollment  
 FROM takes  
 WHERE (semester, year) = ('Fall', 2009)  
 GROUP BY course\_id, sec\_id) e

* 1. Find the sections that had the maximum enrollment in Fall 2009. Make sure you use a WITH clause.

WITH enrollment as (SELECT course\_id, sec\_id, count(id) enrollment  
 FROM takes  
 WHERE (semester, year) = ('Fall', 2009)  
 GROUP BY course\_id, sec\_id  
 ORDER BY enrollment DESC  
 LIMIT 1)  
SELECT course\_id, sec\_id, enrollment  
FROM enrollment

1. Now run these queries on your university database and submit the screen shots of each answer. If the answer does not match what you think it should be, correct the query and try again. This is a way for you to learn from your mistakes. You can submit each one separately but make sure you label them appropriately. Otherwise paste them all into one Word document and submit that. Make sure I can see all rows and columns for the output. You might have to make the result window full screen. (2 points each)
   1. Find the titles of courses in the Comp. Sci. department that have 3 credits.   
      A screenshot of a cell phone

      Description automatically generated
   2. Find the IDs of all students who were taught by an instructor named Einstein; make sure there are no duplicates in the result.

A screenshot of a cell phone

Description automatically generated

* 1. Find the highest salary of any instructor.

A screenshot of a cell phone

Description automatically generated

* 1. Find all instructors earning the highest salary (there may be more than one with the same salary).

A screenshot of a cell phone

Description automatically generated

* 1. Find the enrollment of each section that was offered in Fall 2009. Remember just a number is meaningless unless you include what it refers to.

A screenshot of a cell phone

Description automatically generated

* 1. Find the maximum enrollment, across all sections, in Fall 2009.

A screenshot of a cell phone

Description automatically generated

* 1. Find the sections that had the maximum enrollment in Fall 2009. Make sure you use a WITH clause.

A screenshot of a cell phone

Description automatically generated

1. Write the statements to insert 3 records into the student relation. Test them on our database. If they do not work, tell me why. If they work, tell me what happened. (5 points each – 3 points for insertion statement and 2 points for explanation of what happened)
   1. Insert a student with your last name into the Finance department with an id of 00005 and 0 credits.

INSERT INTO student VALUES ('00005', 'Lesperance', 'Finance', 0)

This insertion was successful, as there were no duplicate keys already in the table, and the insertion did not break any constraints on the table.

* 1. Insert a student with my last name into the Comp. Sci. department with an id of 12345 and null credits.

INSERT INTO student VALUES ('12345', 'Schwartz', 'Comp. Sci.', null)

This was unsuccessful. This insertion failed because there already exists a student in that table with an id of 12345. Because the id attribute is the primary key of the table, there cannot be multiple students with the same id.

* 1. Insert a student with the last name of Zhang into the Comp. Sci. department with an id of 00300 and 100 total credits.

INSERT INTO student VALUES ('00300', 'Zhang', 'Comp. Sci.', 100)

This was successful, for the same reason that the first insertion was successful. There were no conflicting keys in the relation.

1. Show me a screen shot of the contents of the student relation. Make sure it shows all of the content and is large enough for me to read without a magnifying glass. (5 points)   
   A screenshot of a cell phone

   Description automatically generated
2. Write but do NOT execute the following queries. (3 points each)
   1. Delete the contents of the prereq table.

DELETE FROM prereq

* 1. Delete student 00300 from the student table.

DELETE FROM student WHERE id = ‘00300’

1. Find any department whose name contains all of the letters ‘psy ’in this order, but be careful because you want to include departments that might start with those letters and the first one would be capitalized. See if you can write it with only one WHERE predicate. If not, I will accept two. (10 points)

SELECT dept\_name  
FROM department  
WHERE LOWER(dept\_name) like '%psy%'

1. Find the names and major departments of all straight A students. This does not include students with A- as even one grade. We do not give A+. A course that has not yet had a grade assigned should be ignored as the grade does not affect the GPA. (10 points)

(SELECT DISTINCT name, dept\_name  
 FROM student natural join takes  
 WHERE grade = 'A')  
EXCEPT  
(SELECT DISTINCT name, dept\_name  
 FROM student natural join takes  
 WHERE grade != 'A'  
 AND grade is not null)

1. Run the statement from 7 in our database and explain the results. (5 points)

For this problem, I utilized sets in order to find the straight A students. First, I created a set that contained all of the students that ever received in A in any class. Next, I created a second set that contained all of the students that had a grade other than an A in any of their classes. I did not include any class that was not yet graded (which would be null) in this second set. Finally, I selected all of the tuples in the first set that did not exist in the second set (difference operation using the EXCEPT keyword).

A screenshot of a cell phone

Description automatically generated