CIS 182 – SQL Fundamentals – Winter 2024

W6 Exercises: Subqueries

(For the due date, please refer to this lab’s posting on Canvas)

Exercises

1. Write a SELECT statement that returns the same result set as this SELECT statement, but don’t use a join. Instead, use a subquery in a WHERE clause that uses the IN keyword.

|  |
| --- |
| SELECT DISTINCT LastName, FirstName  FROM Instructors i JOIN Courses c  ON i.InstructorID = c.InstructorID  ORDER BY LastName, FirstName |

Please paste a screenshot of your SQL code and result in the boxes below.

|  |
| --- |
| *Code* |
| *Result* |

1. Write a SELECT statement that answers this question: Which instructors have an annual salary that’s greater than the average annual salary for all instructors?

Return the LastName, FirstName, and AnnualSalary columns for each Instructor.

Sort the result set by the AnnualSalary column in descending sequence.

Please paste a screenshot of your SQL code and result in the boxes below.

|  |
| --- |
| *Code* |
| *Result* |

1. Write a SELECT statement that returns the LastName and FirstName columns from the Instructors table.

Return one row for each instructor that doesn’t have any courses in the Courses table. To do that, use a subquery introduced with the NOT EXISTS operator.

Sort the result set by LastName and then by FirstName.

Please paste a screenshot of your SQL code and result in the boxes below.

|  |
| --- |
| *Code* |
| *Result* |

1. Write a SELECT statement that returns the LastName and FirstName columns from the Students table, along with a count of the number of courses each student is taking from the StudentCourses table.

Return one row for each student who is taking more than one class. To do that, use a subquery with the IN class that groups the student course by StudentID.

Group and sort the result set by the LastName and then by the FirstName.

Please paste a screenshot of your SQL code and result in the boxes below.

|  |
| --- |
| *Code* |
| *Result* |

1. Write a SELECT statement that returns the LastName, FirstName, and AnnualSalary columns of each instructor that has a unique annual salary. In other words, don’t include instructors that have the same annual salary as another instructor.

Sort the results by LastName and then by FirstName.

Please paste a screenshot of your SQL code and result in the boxes below.

|  |
| --- |
| *Code* |
| *Result* |

1. Write a SELECT statement that returns one row for each course with these columns:

* The CourseID column from the Courses table
* The most recent enrollment date for that course from the Students table

Change the SELECT statement to a CTE. Then, write a SELECT statement that returns one row per course that shows the CourseDescription for the course and the LastName, FirstName, and EnrollmentDate for the student with the most recent enrollment data.

Please paste a screenshot of your SQL code and result in the boxes below.

|  |
| --- |
| *Code* |
| *Result* |

1. Write a SELECT statement that returns one row for each student that has courses with these columns:

* The StudentID column from the Students table
* The sum of the course units in the Courses table

Include only those students who are taking more than 9 units (fulltime).

Change the SELECT statement to a CTE. Then, write a SELECT statement that uses this CTE to return the student ID, sum of course units, and the tuition. (The tuition is equal to the FullTimeCost column, plus the PerUnitCost column multiplied by the number of units.)

To do that, you can use a cross join to add the columns from the Tuition table to the query. This works because there’s only one row in the Tuition table.

Please paste a screenshot of your SQL code and result in the boxes below.

|  |
| --- |
| *Code* |
| *Result* |