Cmis563Lab8; updated 08/05/2021

# CMIS563 Lab 8

# Writing Sub-queries (27 points @ 3 points)

This lab reinforces Chapter 8 learning objectives as follows:

* Learn the formal subquery definition and write a subquery.
* Learn the subquery restrictions.
* Use the IN operator when writing a subquery.
* Nest subqueries at multiple levels.
* Use comparison operators when writing a subquery.
* Use the ALL and ANY keywords when writing a subquery.
* Write a correlated subquery including the use of the EXISTS operator.
* Use the ORDER BY clause when writing a subquery.

In answering the SQL exercises and questions, **submit a SPOOL file** of **each command that you execute** and any messages that Oracle generates while executing your SQL commands. Also **list the output** for any result table that is generated by SELECT statements.

**Make sure your spool file is in the .doc format.** If not, rename the extension of your spool file to .doc (in the unix server using WinSCP) or copy the contents of your spool file into a Word .doc file before submitting your assignment on Blackboard.

***Note:*** You should create meaningful column names for the result tables produced by your queries.

**IMPORTANT: Write all queries as subqueries – no JOINs unless stated otherwise.**

1. The *ProjectAssignment table* stores data about the hours that employees are working on specific projects. A senior project manager needs a listing of employee names (first and last concatenated) who have not worked on projects 1, 2, 7 or 8. Use a subquery approach and sort the rows of the result table by employee last name.

2. Management would like a report on all employees with a salary GREATER than the MINIMUM salary of the employees in Department 3. Use a subquery approach and include the employee first name, last name (concatenated), department number, and salary. Sort the rows by departmentnumber.

3. Management is concerned that some employees are not putting in sufficient work hours on assigned projects 1, 2, and 3. List the names of employees (last and first concatenated) for those employees who worked on one of these three projects, but worked fewer hours than the average number of hours worked on these three projects combined. This is a nested subquery. Order the report by last name.

4. The previous report has piqued the project manager’s curiosity. He would now like a report that lists all employees who have worked fewer hours than the average for all projects combined. Computer the average hours worked on a project in a subquery. In the report, list the employee’s first and last name.

5. The Chief Nurse would like a list of all room numbers that have a bed with a description that has Surgical anywhere in the field and are available. The result field should list the room

6. The project manager needs to know all available employees who can work on a new project. He wants a report that lists the employee last name, first name, and department name. But, he only wants the employees who are in the departments that are currently working on projects 3 or 5. This query will require both a JOIN (in the outer query) and a subquery.

7. The head of Pediatrics would like a list of all current patients in Pediatrics beds. Pediatric beds will always have PED somewhere in the roomnumber field. The hospital may add more pediatric beds in the future. Create the list by using a subquery. Format patient names with one heading.

8. Provide the treatment number, patientId, and employeeID of all treatments with a service Category description of ‘Surgery’ and have a standardCharge greater than $5,000. This is a nested subquery. Order the report by treatment number.

9. The Hospital Chief has requested a list of employees whose salary is less than all employee salaries in Department 2. The result should not include any employee who has a wagerate rather than a salary (i.e. salary should not be null). Use a subquery approach and the ALL function.

End of Lab.