

Database Programming with SQL 10-4: Correlated Subqueries Practice Activities

# Objectives

* Identify when correlated subqueries are needed
* Construct correlated subqueries
* Construct named subqueries using the WITH clause

# Try It / Solve It

1. Explain the main difference between correlated and non-correlated subqueries?

**As explained in Multiple-Row Subqueries Practice > Problem 3:**

**Correlated subquery is executed multiple times once for each intermediate result row from outer query And for each row of the intermediate results inner correlated subquery is executed, if the inner correlated subquery is satisfied, that row becomes part of final results from the whole query. A correlated subquery will get a candidate row from an outer query, execute the inner query using candidate row value, and use values from the inner query to qualify or disqualify the candidate row.**

**Non-correlated subquery: result/s of inner query are calculated first and reused throughout the execution of outer query.**

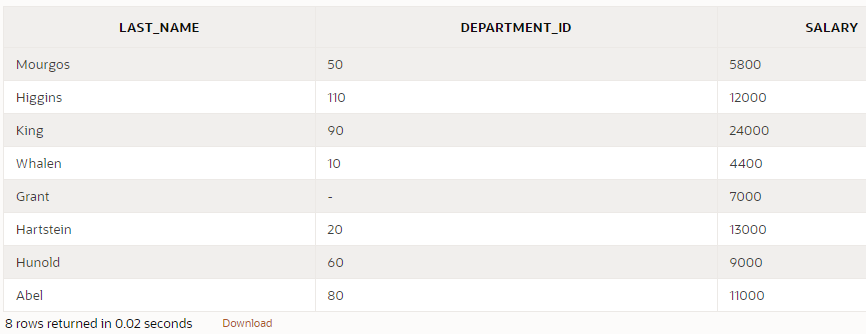
1. Write a query that lists the highest earners for each department. Include the last\_name, department\_id, and the salary for each employee.

**WITH highs AS ( SELECT NVL(department\_id,-1), MAX(salary) FROM employees GROUP BY NVL(department\_id,-1))**

**SELECT last\_name, department\_id, salary**

**FROM employees**

**WHERE (NVL(department\_id,-1), salary) IN (SELECT \* FROM highs );**

****

1. Examine the following select statement and finish it so that it will return the last\_name, department\_id, and salary of employees who have at least one person reporting to them. So we are effectively looking for managers only. In the partially written SELECT statement, the WHERE clause will work as it is. It is simply testing for the existence of a row in the subquery.

SELECT (enter columns here)

FROM (enter table name here) outer WHERE 'x' IN (SELECT 'x'

FROM (enter table name here) inner

WHERE inner(enter column name here) = inner(enter column name here) Finish off the statement by sorting the rows on the department\_id column.

**SELECT outer.last\_name, outer.department\_id, outer.salary**

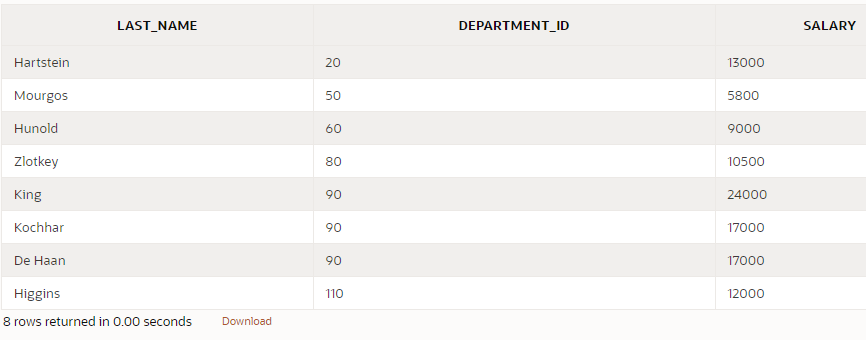
**FROM employees outer**

**WHERE outer.employee\_id  IN (SELECT DISTINCT inner.manager\_id**

**FROM employees  inner**

**WHERE inner.manager\_id = outer.employee\_id)**

**ORDER BY outer.department\_id;**



1. Using a WITH clause, write a SELECT statement to list the job\_title of those jobs whose maximum salary is more than half the maximum salary of the entire company. Name your subquery MAX\_CALC\_SAL. Name the columns in the result JOB\_TITLE and JOB\_TOTAL, and sort the result on JOB\_TOTAL in descending order.

Hint: Examine the jobs table. You will need to join JOBS and EMPLOYEES to display the job\_title.

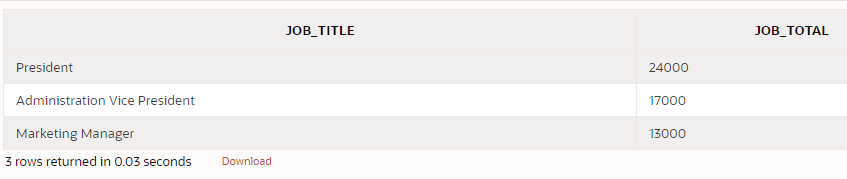
**WITH max\_calc\_sal AS (SELECT jobs.job\_id , jobs.job\_title, MAX(NVL(employees.salary,0)) AS job\_actual\_max FROM employees RIGHT OUTER JOIN jobs ON employees.job\_id = jobs.job\_id GROUP BY jobs.job\_id,jobs.job\_title)**

**SELECT job\_title, job\_actual\_max AS job\_total**

**FROM max\_calc\_sal**

**WHERE job\_actual\_max > (SELECT MAX(job\_actual\_max)/2 FROM max\_calc\_sal)**

**ORDER BY job\_total DESC;**



Copyright © 2020, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.