SQL WORKBOOK (Subqueries)

A subquery in SQL is a query nested within another query. It's a powerful feature that allows you to retrieve data from multiple tables or perform complex calculations. Subqueries can be used in various parts of a SQL statement, such as the SELECT, FROM, WHERE, or HAVING clauses.

Here's a basic example of a subquery in the WHERE clause:

```
SELECT column1, column2
FROM table1
WHERE column1 IN (SELECT column1 FROM table2 WHERE condition);
```

In this example:

- The outer query selects columns from table1.
- The subquery(SELECT column1 FROM table2 WHERE condition) gets data from table2
- The outer query filters rows where column1 matches any value returned by the subquery.

Subqueries can also be used in other parts of a SQL statement. For instance, you can use a subquery in the SELECT clause to perform calculations:

```
SELECT column1, (SELECT COUNT(*) FROM table2 WHERE condition) AS count
FROM table1;
```

Here, the subquery calculates the count of rows in table2 satisfying a condition, and the result is returned as a new column in the outer query.

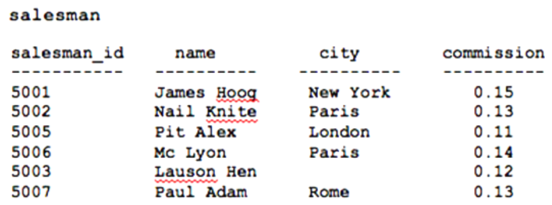
Subqueries can be **correlated** or **non-correlated**.
Correlated subqueries depend on values from the outer query, whereas non-correlated subqueries can run independently of the outer query.

```
-- Non-correlated subquery
SELECT column1
FROM table1
WHERE column2 = (SELECT MAX(column2) FROM table2);

-- Correlated subquery
SELECT column1
FROM table1 AS t1
WHERE column2 = (SELECT MAX(column2) FROM table2 WHERE t1.id = table2.id);
```

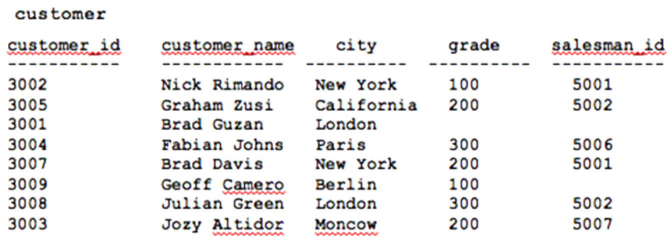
In the non-correlated subquery, the subquery can run independently. In the correlated subquery, the condition in the subquery depends on values from the outer query (t1.id = table2.id).

Below are the queries that you would need to perform in this task and the dataset you would use.



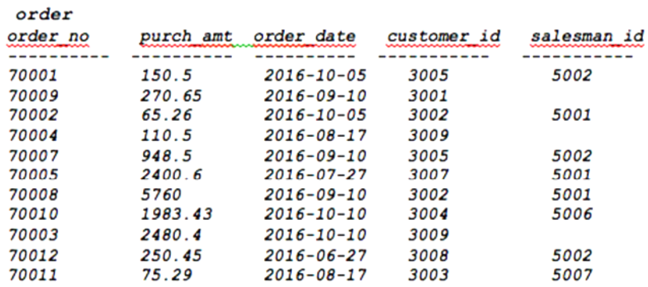
salesman

salesman_id	name	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5003	Lauson Hen		0.12
5007	Paul Adam	Rome	0.13



customer

customer_id	customer_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3005	Graham Zusi	California	200	5002
3001	Brad Guzan	London		
3004	Fabian Johns	Paris	300	5006
3007	Brad Davis	New York	200	5001
3009	Geoff Camero	Berlin	100	
3008	Julian Green	London	300	5002
3003	Jozy Altidor	Moncow	200	5007



order

order_no	purch_amt	order_date	customer_id	salesman_id
70001	150.5	2016-10-05	3005	5002
70009	270.65	2016-09-10	3001	
70002	65.26	2016-10-05	3002	5001
70004	110.5	2016-08-17	3009	
70007	948.5	2016-09-10	3005	5002
70005	2400.6	2016-07-27	3007	5001
70008	5760	2016-09-10	3002	5001
70010	1983.43	2016-10-10	3004	5006
70003	2480.4	2016-10-10	3009	
70012	250.45	2016-06-27	3008	5002
70011	75.29	2016-08-17	3003	5007

SUBQUERIES (NESTED QUERIES)

1. Display all the orders which values are greater than the average order value for 10th October 2012.
2. Find all orders attributed to salesmen in Paris.
3. Extract the data from the orders table for the salesman who earned the maximum commission.
4. Find the name and ids of all salesmen who had more than one customer.
5. Write a query to find all the salesmen who worked for only one customer.
6. Display all the orders that had amounts that were greater than at least one of the orders from September 10th 2012.
7. display only those customers whose grade are, in fact, higher than every customer in New York.