

JOBSHEET

PRAKTIKUM BASIS DATA LANJUT

Jurusan Teknologi Informasi
POLITEKNIK NEGERI MALANG



WEEK 5

SQL SERVER - SUBQUERY, GROUPING, AND AGGREGATING



Information Technology Department, Malang State Polytechnic

Jobsheet 5: Subquery, Grouping, and Aggregating Advanced Database Course

Supervisor: Advanced Database Teaching Team

September 2024

Topics

1. Aggregation functions
2. Group By and Having
3. Sub-queries

Objective

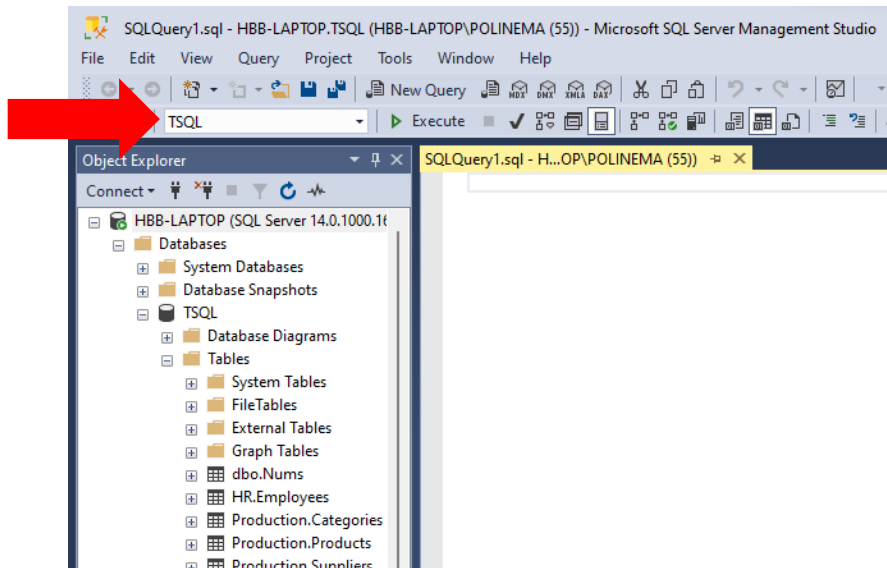
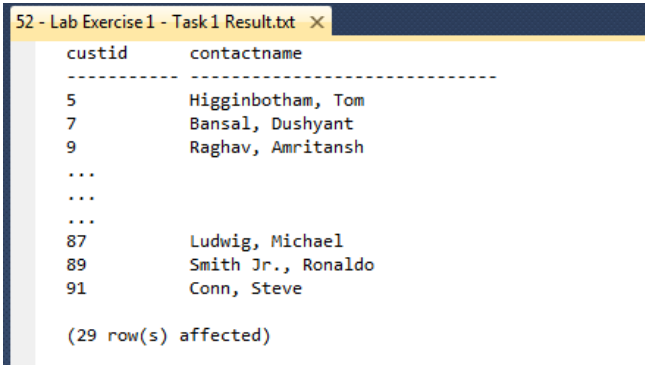
Students are expected to be able to:


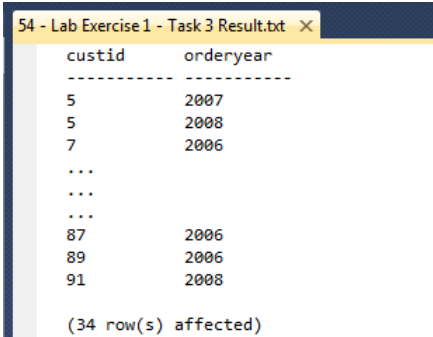
1. Implementing aggregation functions.
2. Performing queries with group by and having.
3. Creating sub-queries.

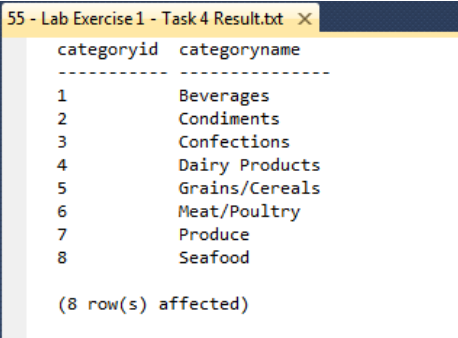
General Instructions

1. Follow the steps in the practical sections in the order given.
2. Answer all questions marked **[Question-X]** that are found in certain steps in each part of the practicum.
3. In each step of the practicum, there is an explanation that will help you answer the questions in instruction number 3, so read and do all the practicum parts in this jobsheet.
4. Write the answers to the questions in the instructions number 3 in a report that is done using a word processing application (Word, OpenOffice, or other similar). Export as a **PDF file** with the following name format:
 - **BDL_Class_03_YourFullName** .pdf
 - Collect the PDF files as a practical report to the supervising lecturer.
 - In addition to the file name, also include your identity on the first page of the report.

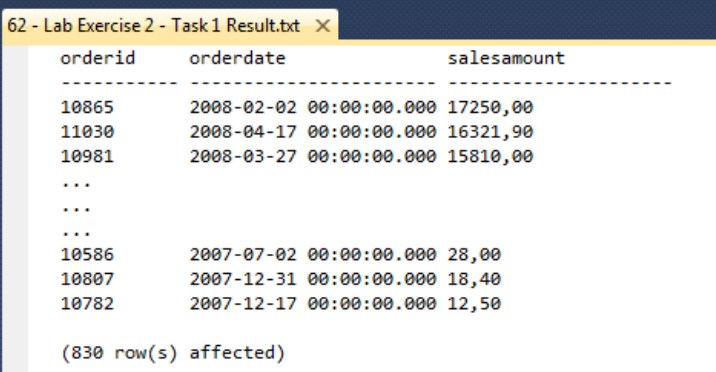
Lab – Part 1: Writing Queries Using the GROUP BY Clause

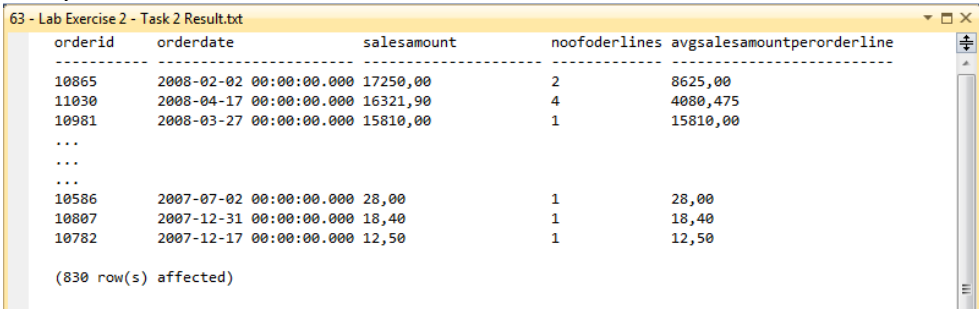
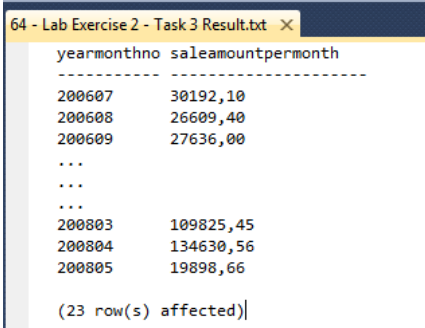
Step	Information
1	<p>Scenario :</p> <p>A company's sales department wants to create additional <i>up-sell opportunities</i> from customers. To do this, employees need to analyze various customer groups and product categories based on several business rules. Given this scenario, a T-SQL statement using the SELECT clause is needed to retrieve the required rows from the Sales.Customers table.</p> <p>To do the experiment in this practicum part 1, first log in to SQL Server Management Studio (SSMS). Then make sure the database is connected to "TSQL". Next, you can open a new worksheet by clicking "New Query".</p>
	
2	<p>[Question-1] Write a T-SQL SELECT that will display the group of customers who made a purchase. The SELECT clause must include the custid column from the Sales.Orders table and the contactname column from the Sales.Customers table. Group the two columns, and filter only orders from sales employees who have empid equal to 5!</p>
3	<p>Compare the results in stage 2 with the following image. If it is the same then the T-SQL you wrote is correct.</p> 

4	[Question-2] Copy the T-SQL answer to question-1. Then modify it to display additional information from the city column from the Sales.Customers table in the SELECT clause!
5	[Question 3] Is there an error message in the answer to question 2? What is the error message? Why does this message occur?
6	<p>[Question-4] Correct the error that occurs in the answer to question-2! If the execution result is the same as the following image, then the T-SQL created is correct.</p> 
7	[Question-5] Write a SELECT statement that will display a group of rows based on the custid column and will be calculated by the orderyear column representing the year of the order based on the orderdate column of the Sales.Orders table. Then filter the results to include only orders from sales employees whose empid is equal to 5!
8	<p>Compare the results in question 5 with the following image. If they are the same, then the T-SQL you wrote is correct.</p> 
9	[Question-6] Write a SELECT statement that will return rows grouped by the categoryname column in the Production.Categories table. Then filter the results to only product categories that were ordered in 2008!
10	Compare the results in question 6 with the following image. If they are the same, then the T-SQL you wrote is correct.

	 <pre> categoryid categoryname ----- 1 Beverages 2 Condiments 3 Confections 4 Dairy Products 5 Grains/Cereals 6 Meat/Poultry 7 Produce 8 Seafood (8 row(s) affected) </pre>
11	Conclusion : After completing this section of the practicum, students will be able to use the GROUP BY clause in T-SQL statements.

Lab – Part 2: Writing Queries Using Aggregation Functions

Step	Information
1	<p>Scenario :</p> <p>The marketing department wants to launch a new campaign, so employees need to gain better insight into customers' buying behavior. Therefore, a different sales report should be created based on the average annual sales amount per customer.</p> <p>To carry out the experiment in this practical part 2, make sure the database is connected to “TSQL”.</p>
2	<p>[Question-7] Write a SELECT statement that will return theorderid,orderdate columns from the Sales.Orders table and the total sales amount per orderid (Hint: Multiply the qty and unitprice columns from the Sales.OderDetails table). Use the alias salesamount for the calculated column. Then sort the result by the total sales amount in descending order!</p>
3	<p>Compare the results in question 7 with the following image. If they are the same, then the T-SQL you wrote is correct.</p>  <pre> orderid orderdate salesamount ----- 10865 2008-02-02 00:00:00.000 17250,00 11030 2008-04-17 00:00:00.000 16321,90 10981 2008-03-27 00:00:00.000 15810,00 10586 2007-07-02 00:00:00.000 28,00 10807 2007-12-31 00:00:00.000 18,40 10782 2007-12-17 00:00:00.000 12,50 (830 row(s) affected) </pre>
4	<p>[Question-8] Copy the T-SQL statement in the answer to question-7 and modify it by inserting the number of order lines for each order and the average sales amount per orderid according to the order. Use the alias names nooforderlines and avgsalesamountperorderlines respectively!</p>

5	<p>Compare the results in question 8 with the following image. If they are the same, then the T-SQL you wrote is correct.</p> 
6	<p>[Question-9] Write a SELECT statement to get the total sales amount for each month! The use of the SELECT clause should include the calculation of the yearmonthno column (notation YYYYMM) based on the orderdate column in the Sales.Orders table and the total sales amount (Multiplication of the qty column with the unitprice from the Sales.OrderDetails table) which is given the alias saleamountpermonth. The order of the results is based on the calculation of the yearmonthno column.</p>
7	<p>Compare the results in question 9 with the following image. If they are the same, then the T-SQL you wrote is correct.</p> 
8	<p>[Question-10] Write a SELECT statement that will retrieve all customers (including those without orders) and the sales amount, maximum order amount per row, and number of orders! The SELECT clause must include the custid and contactname columns from the Sales.Customers table and 4 (four) columns calculated based on the following aggregation functions:</p> <ol style="list-style-type: none"> 1) totalsalesamount, is an alias for the total sales amount per order 2) maxsalesamountperorderline, is an alias for the maximum sales amount per order line 3) numberofrows, is an alias for number of rows (use * in the COUNT function) 4) numberoforderlines, is an alias for the number of order lines (use the orderid column in the COUNT function) <p>Sort the results by the totalsalesamount column.</p>
9	<p>Compare the results in question 10 with the following image. If they are the same, then the T-SQL you wrote is correct.</p>



65 - Lab Exercise 2 - Task 4 Result.txt

custid	contactname	totalsalesamount	maxsalesamountperorderline	numberofrows	numberoforderlines
22	Bueno, Janaina Burdan, Neville	NULL	NULL	1	0
57	Tollefsen, Bjørn	NULL	NULL	1	0
13	Benito, Almudena	100,80	80,00	2	2
...					
...					
20	Kane, John	113236,68	6360,00	102	102
71	Navarro, Tomás	115673,39	7427,40	116	116
63	Veronesi, Giorgio	117483,39	15810,00	86	86

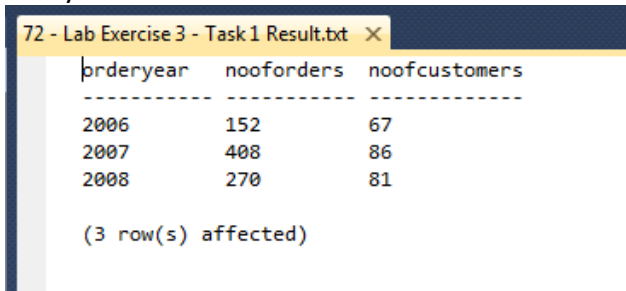
Warning: Null value is eliminated by an aggregate or other SET operation.

(91 row(s) affected)

10

Conclusion : After carrying out this part of the practicum, students will know how to use the aggregation function.

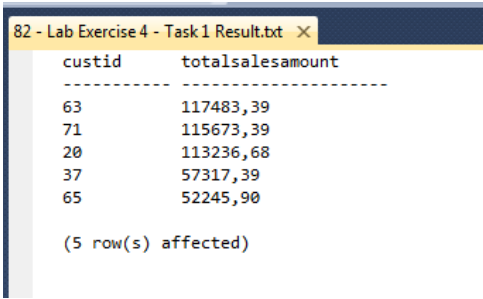
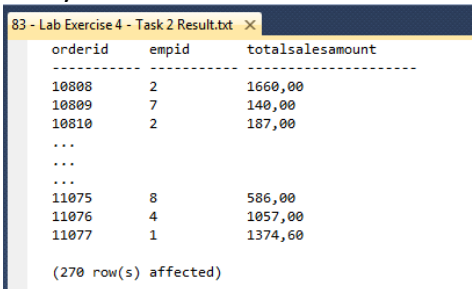
Lab – Part 3: Writing Queries Using Distinct Aggregation Functions

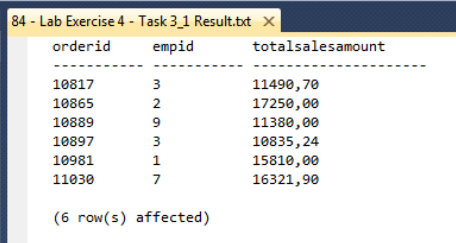
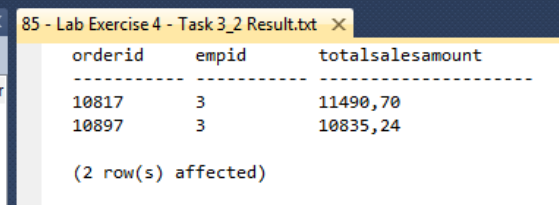
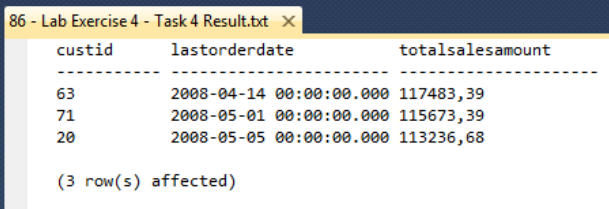
Step	Information
1	<p>Scenario :</p> <p>The marketing department would like to have some additional reports showing the number of customers who had orders within a certain time period and the number of customers by first letter and <i>contact name</i>.</p> <p>To carry out the experiment in this practical part 3, make sure the database is connected to “TSQL”.</p>
2	<p>[Question-11] Based on the T-SQL execution results below, why is the number of orders (nooforders) the same as the number of customers (noofcustomers)?</p> <pre>SELECT YEAR (order date) AS orderyear , COUNT (orderid) AS nooforders , COUNT (custid) AS noofcustomers FROM Sales . Orders GROUP BY YEAR (orderdate);</pre>
3	<p>[Question-12] Fix the T-SQL in question-12 to show the correct number of customers who placed orders each year!</p>
4	<p>Compare the results in question 12 with the following image. If they are the same, then the T-SQL you wrote is correct.</p> 
5	<p>[Question-13] Write a SELECT statement to retrieve the number of customers based on the first letter of the value in the contactname column of the Sales.Customers table. Add a column that shows the number of orders placed by each customer group. Use the aliases firstletter, noofcustomers and nooforders respectively. Sort the results by the firstletter column!</p>
6	<p>Compare the results in question 13 with the following image. If they are the same, then the T-SQL you wrote is correct.</p>



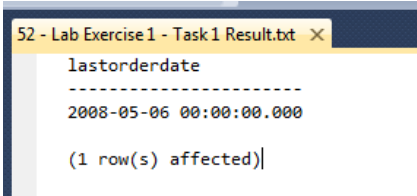
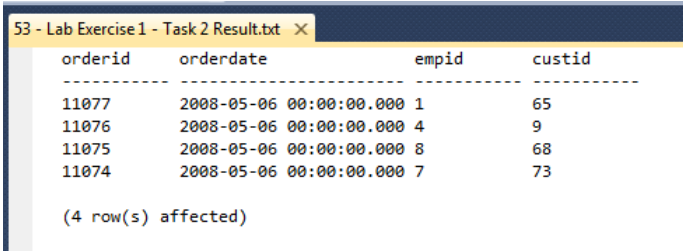
	<div>73 - Lab Exercise 3 - Task 2 Result.txt ✕</div> <pre> firstletter noofcustomers nooforders ----- A 2 19 B 6 37 C 6 72 V 3 43 W 4 29 Y 1 3 Warning: Null value is eliminated by an aggregate or other SET operation. (23 row(s) affected) </pre>
7	<p>[Question-14] Copy the T-SQL in the answer to question-6 then modify it by including information about each product category: number of sales, number of orders, and average sales amount per order. Use the respective alias names, nooforders, and avgsalesamountperorder.</p>
8	<p>Compare the results in question 13 with the following image. If they are the same, then the T-SQL you wrote is correct.</p> <div>74 - Lab Exercise 3 - Task 3 Result.txt ✕</div> <pre> categoryid categoryname totalsalesamount nooforders avgsalesamountperorder ----- 1 Beverages 122223,75 128 954,873 2 Condiments 34557,45 62 557,3782 3 Confections 58359,73 89 655,7273 4 Dairy Products 82803,90 90 920,0433 5 Grains/Cereals 30422,25 55 553,1318 6 Meat/Poultry 60275,57 43 1401,7574 7 Produce 32415,85 42 771,8059 8 Seafood 48712,84 101 482,3053 (8 row(s) affected) </pre>
9	<p>Conclusion : After carrying out this section of the practicum, students can apply the DISTINCT aggregation function.</p>

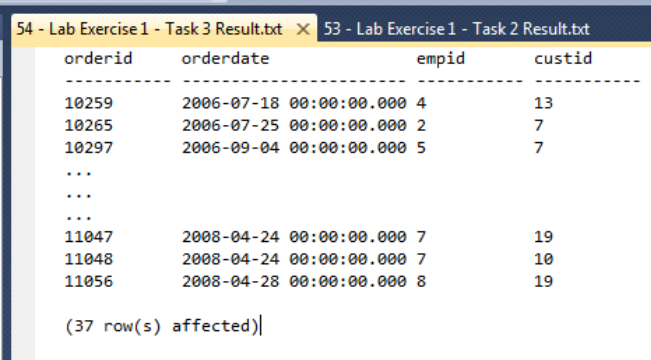
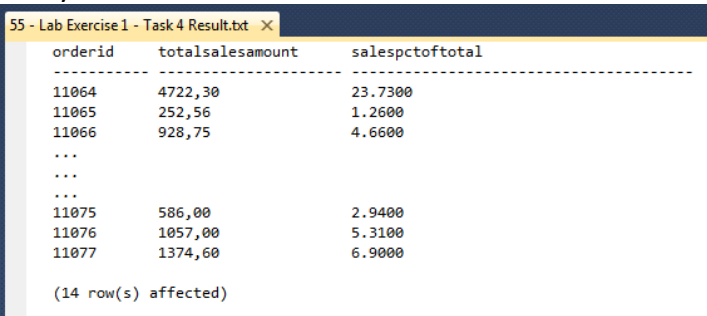
Lab – Part 4: Writing a Query That Performs Group Filtering With the HAVING Clause

Step	Information
1	<p>Scenario :</p> <p>The report on customer behavior analysis that was created in the previous experiment has met the needs of the sales and marketing department. Now the department needs the report filtered based on the total sales amount and the number of orders. So this section of the scenario will discuss how to filter the results of the previous experiment based on the aggregation function and learn the use of the WHERE and HAVING clauses.</p> <p>To carry out the experiment in this practical part 4, make sure the database is connected to “TSQL”.</p>
2	<p>[Question-15] Write a T-SQL command with SELECT clause to retrieve the top 5 customers with total sales more than \$10,000. Display the custid column from the order table and calculate the column containing the sales amount based on the qty and unitprice columns from the Sales.OrderDetails table. Use the alias totalsalesamount.</p>
3	<p>Compare the results in question 15 with the following image. If they are the same, then the T-SQL you wrote is correct.</p> 
4	<p>[Question-16] Write a T-SQL command with a SELECT clause to retrieve the empid, orderid columns and the column that represents the calculation of total sales (total sales amount) based on the Sales.Orders and Sales.OrderDetails tables. Filter the results into a group of data rows only for orders in the year 2008!</p>
5	<p>Compare the results in question 16 with the following image. If they are the same, then the T-SQL you wrote is correct.</p> 
6	<p>[Question-17] Copy the T-SQL command from question-16 and modify it to add a filter that only retrieves rows that have a sales amount greater than \$10,000!</p>

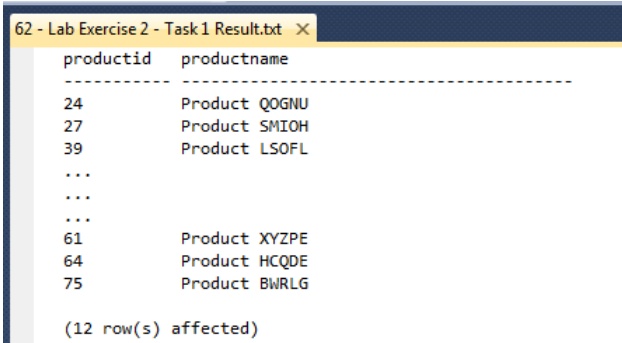
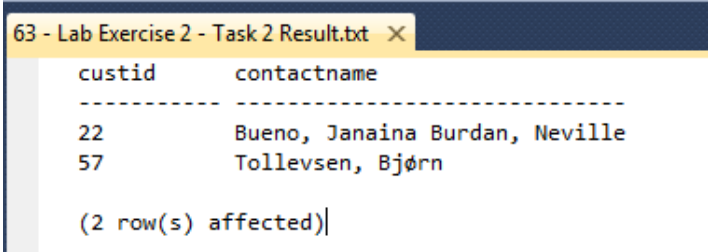
7	<p>Compare the results in question 17 with the following image. If they are the same, then the T-SQL you wrote is correct.</p> 
8	<p>[Question-18] Copy the T-SQL command for the answer to question-17 and modify it to add a filter that only displays employees with empid equal to 3(three)!</p>
9	<p>Compare the results in question 18 with the following image. If they are the same, then the T-SQL you wrote is correct.</p> 
10	<p>[Question-19] Write a T-SQL command with SELECT clause to retrieve all customers who have more than 25 orders, and add information about the last order date and sales amount. Display the custid column from the Sales.Orders table and two calculation columns (lastorderdate based on the orderdate column and totalsalesamount based on the qty and unitprice columns from the Sales.OrderDetails table!</p>
11	<p>Compare the results in question 19 with the following image. If they are the same, then the T-SQL you wrote is correct.</p> 
12	<p>Conclusion : After doing this section of the practicum, students can use the HAVING clause.</p>

Lab – Part 5: Writing Queries Using Self-Contained Sub-queries

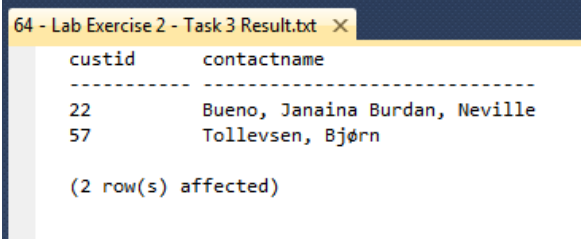
Step	Information
1	<p>Scenario :</p> <p>The sales department needs some advanced reports to analyze sales orders. This requires a SELECT statement that uses a self-contained sub-query.</p> <p>To carry out the experiment in this practical part 5, make sure the database is connected to "TSQL".</p>
2	<p>[Question-20] Write a SELECT statement to display the maximum orderdate from the Sales.Orders table.</p>
3	<p>Compare the results in question 20 with the following image. If they are the same, then the T-SQL you wrote is correct.</p> 
4	<p>[Question-21] Write a SELECT statement to display the orderid, orderdate, empid, and custid columns from the Sales.Orders table. Then filter the results by including only orders that match the latest order time (Use the query in the answer to question-20 as a self-contained sub-query)!</p>
5	<p>Compare the results in question 21 with the following image. If they are the same, then the T-SQL you wrote is correct.</p> 
6	<p>[Question-22] Execute the T-SQL below, then modify it with a customer filter based on contact names starting with the letter B!</p> <pre> SELECT orderid , orderdate , empid , custid FROM Sales . Orders WHERE custodian = (SELECT custid FROM Sales . Customers WHERE contactname LIKE N'I%'); </pre>
7	<p>[Question-23] Is there an error in the execution result of question-22? Why?</p>

8	[Question-24] Correct the answer to question-23 so that the result is not an error!
9	<p>Compare the results in question 24 with the following image. If they are the same, then the T-SQL you wrote is correct.</p> 
10	<p>[Question-25] Write a SELECT statement to retrieve the orderid column from the Sales.Orders table and also the calculation result column:</p> <ol style="list-style-type: none"> 1) totalsalesamount (based on qty and unitprice columns from Sales.OrderDetails table) 2) salespctoftotal (percentage of total sales amount of each order divided by the total sales amount for all orders in a certain period) <p>Filter the results only for orders made in May 2008.</p>
11	<p>Compare the results in question 25 with the following image. If they are the same, then the T-SQL you wrote is correct.</p> 
12	<p>Conclusion : After completing this section of the practicum, students can use self-contained sub-queries in T-SQL statements.</p>

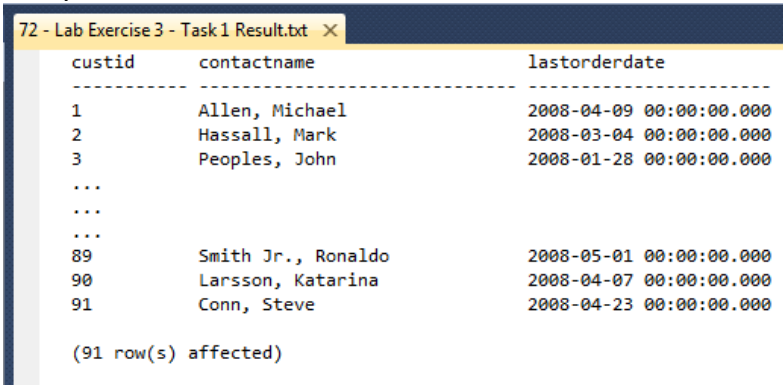
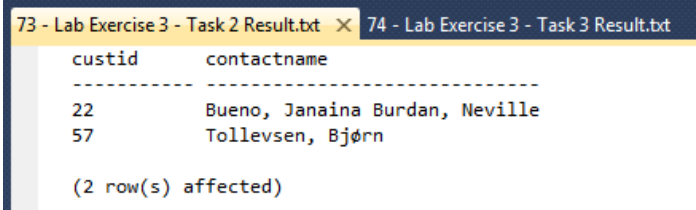
Lab – Part 6: Writing Queries That Use Scalar And Multi-Valued Sub-Query

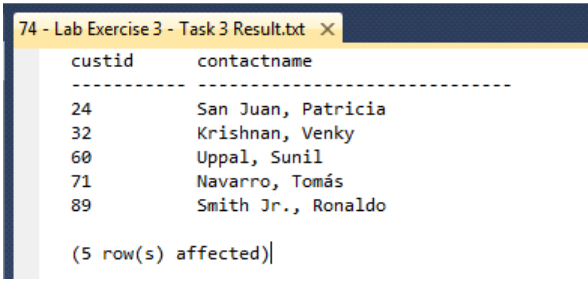
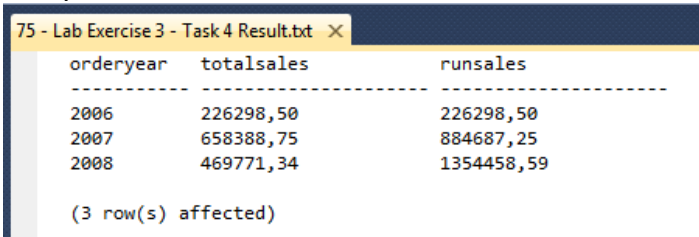
Step	Information
1	<p>Scenario :</p> <p>The marketing department wants to prepare materials for various product and customer groups based on historical sales information. This requires a SELECT statement using a Sub-Query in the WHERE clause.</p> <p>To carry out the experiment in this practical part 6, make sure the database is connected to “TSQL”.</p>
2	<p>[Question-26] Write a SELECT statement to retrieve the productid and productname columns from the Production.Products table. Then filter the results to display products that are sold in large quantities (more than 100 products) for a particular order row!</p>
3	<p>Compare the results in question 26 with the following image. If they are the same, then the T-SQL you wrote is correct.</p> 
4	<p>[Question-27] Write a SELECT statement to retrieve the custid and contactname columns from the Sales.Customers table. Then filter only for customers who do not have any orders!</p>
5	<p>Compare the results in question 27 with the following image. If they are the same, then the T-SQL you wrote is correct.</p> 
6	<p>[Problem-28] There is one additional row of data in the Sales.Orders table with T-SQL as follows:</p> <pre> INSERT INTO Sales . Orders (custid , empid , orderdate , requireddate , shippeddate , shipperid , freight , shipname , shipaddress , shipcity , shipregion , shippostalcode , shipcountry) VALUES (NULL , 1 , '20111231' , '20111231' , '20111231' , 1 , 0 , 'ShipOne' , 'ShipAddress' , 'ShipCity' , 'RA' , '1000' , 'USA'); </pre>



	Execute the command! Then copy the answer to question 27. How do the results of the two T-SQLs compare? Why?
7	[Question-29] Modify the answer to question-27 (different way with the same output), by deleting rows with unknown values in the custid column!
8	<p>Compare the results in question 29 with the following image. If they are the same, then the T-SQL you wrote is correct.</p>  <pre>64 - Lab Exercise 2 - Task 3 Result.txt X custid contactname ----- 22 Bueno, Janaina Burdan, Neville 57 Tollefsen, Bjørn (2 row(s) affected)</pre>
9	Conclusion : After completing the practicum and answering the questions in this section, students can use multi-results in T-SQL statements.

Lab – Part 7: Writing Queries That Use Correlated Sub-Query And EXISTS Predicate

Step	Information
1	<p>Scenario :</p> <p>The sales department wants to have some additional reports to display various analyses for customers. Since the sales department's request is complex, it requires the use of correlated Sub-Query.</p> <p>To carry out the experiment in this practical part 7, make sure the database is connected to "TSQL".</p>
2	<p>[Question-30] Write a SELECT statement to retrieve the custid and contactname columns from the Sales.Customers table. Add a lastorderdate column containing the last date from the Sales.Orders table for each customer (Use a correlated sub-query).</p>
3	<p>Compare the results in question 30 with the following image. If they are the same, then the T-SQL you wrote is correct.</p> 
4	<p>[Question-31] Write a SELECT statement to retrieve all customers who do not have orders in the Sales.Orders table. Use the EXISTS predicate to filter to include customers who do not have orders! (There is no need to explicitly check the custid column of the Sales.Orders table for not NULL status)</p>
5	<p>Compare the results in question 31 with the following image. If they are the same, then the T-SQL you wrote is correct.</p> 
6	<p>[Question-32] Write a SELECT statement to retrieve the custid and contactname columns from the Sales.Customers table. Then filter the results to only customers who placed orders on or after April 1, 2008, and placed orders with a high price tag above \$100!</p>
7	<p>Compare the results in question-32 with the following image. If they are the same, then the T-SQL you wrote is correct.</p>

	
8	<p>[Question-33] Write a SELECT statement that will retrieve information for each year as follows:</p> <ol style="list-style-type: none"> 1) Order year 2) Total sales amount 3) Total amount of sales sold over the years (every year returns the total amount of sales up to a certain year, for example at the beginning of 2006 returns the total amount of sales for the following year 2007) 4) The SELECT statement must have 3 columns: <ul style="list-style-type: none"> • orderyear, comes from the orderyear column of the Sales.Orders table • totalsales, comes from the qty and unitprice columns of the Sales.OrderDetails table • runsales, represents the number of sales currently taking place. This column uses a correlated sub-query
9	<p>Compare the results in question 33 with the following image. If they are the same, then the T-SQL you wrote is correct.</p> 
10	<p>Conclusion : After working on the practical work and questions in this section, you know how to use correlated Sub-Query in T-SQL.</p>

--- Have a great time doing it ---