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*

SAFRIZAL RAHMAN_SIB_2G_19

Topics

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

Objective

Students are expected to be able to:

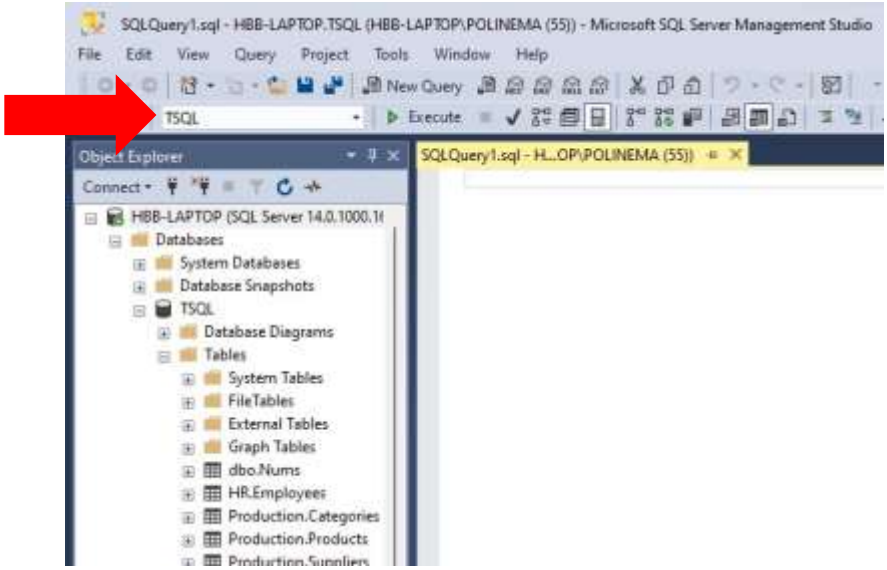
1. Implementing aggregation functions.
2. Performing queries with group by and having.
3. Creating sub-queries.

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	Scenario : A company's sales department wants to create additional *up-sell opportunities* 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. 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”.
	
2	[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! SELECT c.custid, c.contactname FROM Sales.Customers c JOIN Sales.Orders o ON c.custid = o.custid WHERE o.empid = 5 GROUP BY c.custid, c.contactname;

3	Compare the results in stage 2 with the following image. If it is the same then the T-SQL you wrote is correct. 52 - Lab Exercise 1 - Task 1 Result.txt custid contactname ---------- ---------- 5 Higginbotham, Tom 7 Bansal, Dushyant 9 Raghav, Amritansh ... ... 87 Ludwig, Michael 89 Smith Jr., Ronaldo 91 Conn, Steve (29 row(s) affected)

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! SELECT c.custid, c.contactname FROM Sales.Customers c JOIN Sales.Orders o ON c.custid = o.custid WHERE o.empid = 5 GROUP BY c.custid, c.contactname;

SQLQuery1.sql - M...IZAL RAHMAN (53))*

```sql
SELECT c.custid, c.contactname
FROM Sales.Customers c
JOIN Sales.Orders o ON c.custid = o.custid
WHERE o.empid = 5
GROUP BY c.custid, c.contactname;
```

100 %

Results | Messages

	custid	contactname
1	5	Higginbotham, Tom
2	7	Bansal, Dushyant
3	9	Raghav, Amritansh
4	14	Jelitto, Jacek
5	21	Russo, Giuseppe
6	24	San Juan, Patricia
7	25	Carlson, Jason
8	30	Shabalin, Rostislav
9	34	Cohen, Shy
10	41	Litton, Tim
11	46	Dressler, Marlies
12	47	Lupu, Cornel
13	50	Mace, Donald
14	52	Dupont-Roc, Patrice
15	58	Fakhouri, Fadi
16	60	Uppal, Sunil
17	62	Misiec, Anna
18	63	Veronesi, Giorgio
19	65	Moore, Michael
20	66	Voss, Florian
21	67	Garden, Euan
22	71	Navarro, Tomás

5

[Question 3] Is there an error message in the answer to question 2? What is the error message? Why does this message occur?

Column 'Sales.Customers.city' is invalid in the select list because it is not contained in either an aggregate function or the GROUP BY clause.

6	[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. 53 - Lab Exercise 1 - Task 2 Result.txt custid contactname city --- --- --- 5 Higginbotham, Tom Luleå 7 Bansal, Dushyant Strasbourg 9 Raghav, Amritansh Marseille ... ... 87 Ludwig, Michael Oulu 89 Smith Jr., Ronaldo Seattle 91 Conn, Steve Warszawa (29 row(s) affected) SELECT c.custid, c.contactname, c.city FROM Sales.Customers c JOIN Sales.Orders o ON c.custid = o.custid WHERE o.empid = 5 GROUP BY c.custid, c.contactname, c.city;

SQLQuery1.sql - M...IZAL RAHMAN (53))*

```sql
SELECT c.custid, c.contactname, c.city
FROM Sales.Customers c
JOIN Sales.Orders o ON c.custid = o.custid
WHERE o.empid = 5
GROUP BY c.custid, c.contactname, c.city;
```

100 %

Results | Messages

	custid	contactname	city
1	5	Higginbotham, Tom	Luleå
2	7	Bansal, Dushyant	Strasbourg
3	9	Raghav, Amritansh	Marseille
4	14	Jelitto, Jacek	Bern
5	21	Russo, Giuseppe	Sao Paulo
6	24	San Juan, Patricia	Bräcke
7	25	Carlson, Jason	München
8	30	Shabalin, Rostislav	Sevilla
9	34	Cohen, Shy	Rio de Janeiro
10	41	Litton, Tim	Toulouse
11	46	Dressler, Marlies	Barquisimeto
12	47	Lupu, Cornel	I. de Margarita
13	50	Mace, Donald	Bruxelles
14	52	Dupont-Roc, Patrice	Leipzig
15	58	Fakhouri, Fadi	México D.F.
16	60	Uppal, Sunil	Lisboa
17	62	Misiec, Anna	Sao Paulo
18	63	Veronesi, Giorgio	Cunewalde
19	65	Moore, Michael	Albuquerque
20	66	Voss, Florian	Reggio Emilia
21	67	Garden, Euan	Rio de Janeiro
22	71	Navarro, Tomás	Boise

Query executed successfully.

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!

```sql
SELECT c.custid, YEAR(o.orderdate) AS orderyear
FROM Sales.Customers c
JOIN Sales.Orders o ON c.custid = o.custid
WHERE o.empid = 5
GROUP BY c.custid, YEAR(o.orderdate);
```

SQLQuery1.sql - M...IZAL RAHMAN (53))*

```sql
SELECT c.custid, YEAR(o.orderdate) AS orderyear
FROM Sales.Customers c
JOIN Sales.Orders o ON c.custid = o.custid
WHERE o.empid = 5
GROUP BY c.custid, YEAR(o.orderdate);
```

100 %

Results | Messages

	custid	orderyear
1	5	2007
2	5	2008
3	7	2006
4	9	2007
5	14	2006
6	21	2007
7	24	2006
8	25	2007
9	30	2008
10	34	2008
11	41	2006
12	46	2008
13	47	2008
14	50	2007
15	52	2007
16	58	2007
17	60	2006
18	60	2007
19	62	2006
20	63	2007
21	65	2007
22	66	2008

Query executed successfully.

8	Compare the results in question 5 with the following image. If they are the same, then the TSQL you wrote is correct. 54 - Lab Exercise 1 - Task 3 Result.txt custid orderyear ---------- ---------- 5 2007 5 2008 7 2006 ... ... 87 2006 89 2006 91 2008 (34 row(s) affected)

SQLQuery1.sql - M...IZAL RAHMAN (53))*

```
SELECT c.custid, YEAR(o.orderdate) AS orderyear
FROM Sales.Customers c
JOIN Sales.Orders o ON c.custid = o.custid
WHERE o.empid = 5
GROUP BY c.custid, YEAR(o.orderdate);
```

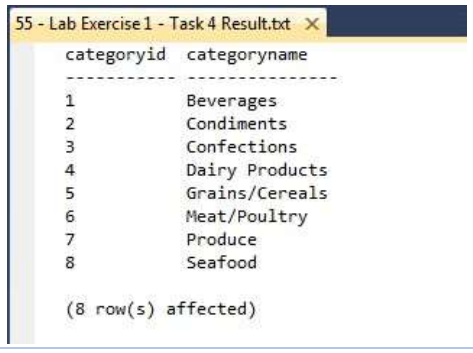
100 %

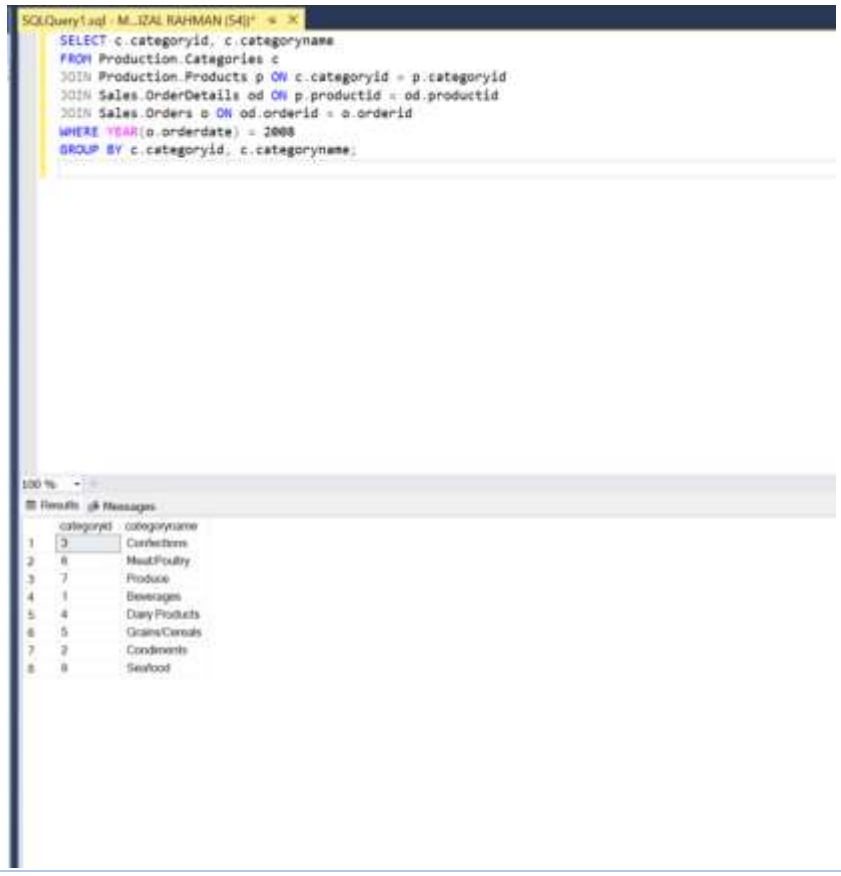
Results | Messages

	custid	orderyear
1	5	2007
2	5	2008
3	7	2006
4	9	2007
5	14	2006
6	21	2007
7	24	2006
8	25	2007
9	30	2008
10	34	2008
11	41	2006
12	46	2008
13	47	2008
14	50	2007
15	52	2007
16	58	2007
17	60	2006
18	60	2007
19	62	2006
20	63	2007
21	65	2007
22	66	2008

Query executed successfully.

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 TSQL you wrote is correct.
	

	
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	Scenario : 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. To carry out the experiment in this practical part 2, make sure the database is connected to "TSQL".
2	[Question-7] Write a SELECT statement that will return the orderid,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!

SQLQuery1.sql - M...IZAL RAHMAN (54)*

```sql
SELECT c.categoryid, c.categoryname
FROM Production.Categories c
JOIN Production.Products p ON c.categoryid = p.categoryid
JOIN Sales.OrderDetails od ON p.productid = od.productid
JOIN Sales.Orders o ON od.orderid = o.orderid
WHERE YEAR(o.orderdate) = 2008
GROUP BY c.categoryid, c.categoryname;
```

100 %

Results | Messages

	categoryid	categoryname
1	3	Confections
2	6	Meat/Poultry
3	7	Produce
4	1	Beverages
5	4	Dairy Products
6	5	Grains/Cereals
7	2	Condiments
8	8	Seafood

SQLQuery1.sql - M...IZAL RAHMAN (54))*

```sql
SELECT o.orderid, o.orderdate, SUM(od.qty * od.unitprice) AS salesamount
FROM Sales.Orders o
JOIN Sales.OrderDetails od ON o.orderid = od.orderid
GROUP BY o.orderid, o.orderdate
ORDER BY salesamount DESC;
```

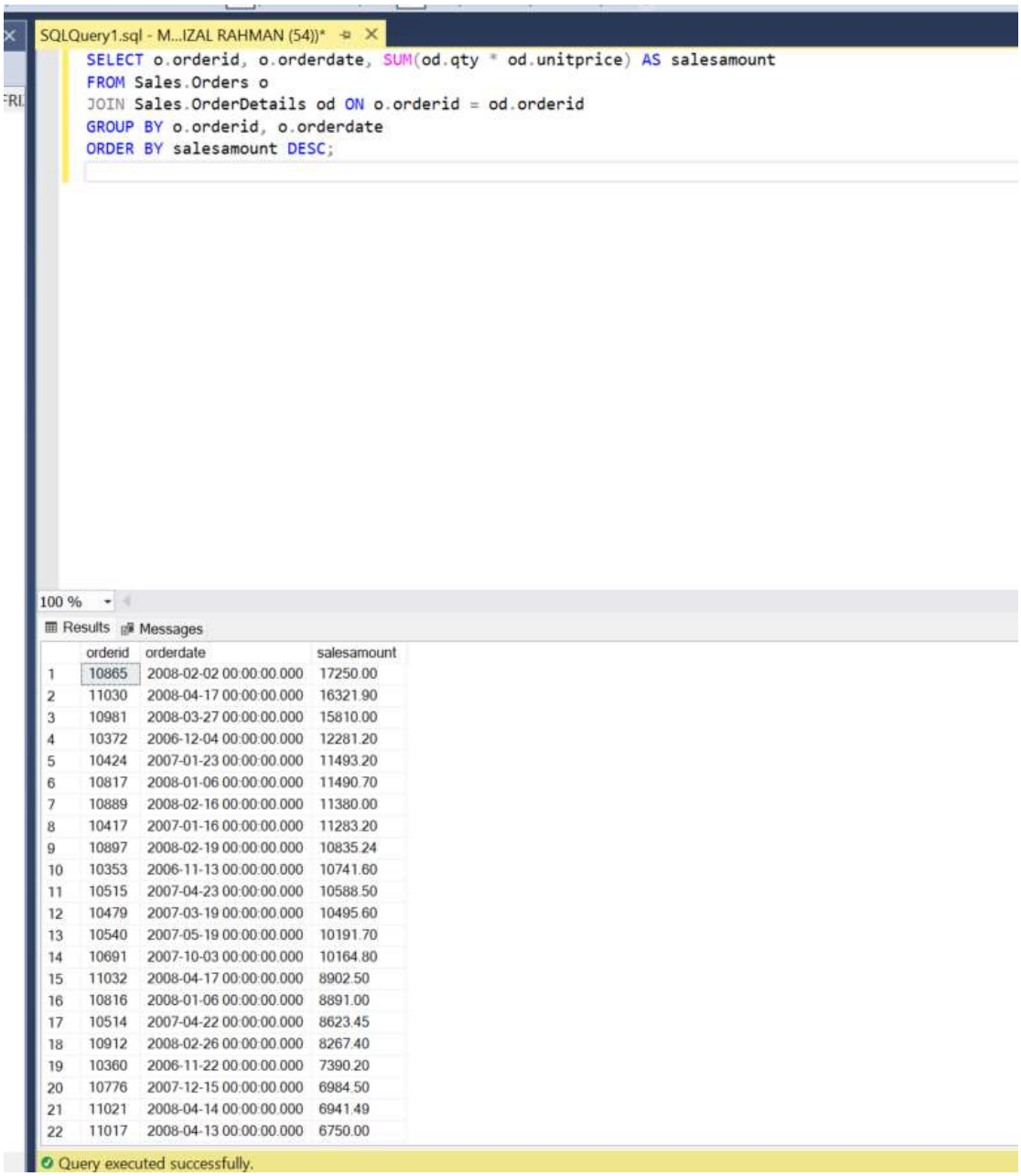
100 %

Results | Messages

	orderid	orderdate	salesamount
1	10865	2008-02-02 00:00:00.000	17250.00
2	11030	2008-04-17 00:00:00.000	16321.90
3	10981	2008-03-27 00:00:00.000	15810.00
4	10372	2006-12-04 00:00:00.000	12281.20
5	10424	2007-01-23 00:00:00.000	11493.20
6	10817	2008-01-06 00:00:00.000	11490.70
7	10889	2008-02-16 00:00:00.000	11380.00
8	10417	2007-01-16 00:00:00.000	11283.20
9	10897	2008-02-19 00:00:00.000	10835.24
10	10353	2006-11-13 00:00:00.000	10741.60
11	10515	2007-04-23 00:00:00.000	10588.50
12	10479	2007-03-19 00:00:00.000	10495.60
13	10540	2007-05-19 00:00:00.000	10191.70
14	10691	2007-10-03 00:00:00.000	10164.80
15	11032	2008-04-17 00:00:00.000	8902.50
16	10816	2008-01-06 00:00:00.000	8891.00
17	10514	2007-04-22 00:00:00.000	8623.45
18	10912	2008-02-26 00:00:00.000	8267.40
19	10360	2006-11-22 00:00:00.000	7390.20
20	10776	2007-12-15 00:00:00.000	6984.50
21	11021	2008-04-14 00:00:00.000	6941.49
22	11017	2008-04-13 00:00:00.000	6750.00

Query executed successfully.

3	Compare the results in question 7 with the following image. If they are the same, then the TSQL you wrote is correct. 62 - Lab Exercise 2 - Task 1 Result.txt 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)

<table>
<tr>
<td></td>
<td>


	orderid	orderdate	salesamount
1	10865	2008-02-02 00:00:00.000	17250.00
2	11030	2008-04-17 00:00:00.000	16321.90
3	10981	2008-03-27 00:00:00.000	15810.00
4	10372	2006-12-04 00:00:00.000	12281.20
5	10424	2007-01-23 00:00:00.000	11493.20
6	10817	2008-01-06 00:00:00.000	11490.70
7	10889	2008-02-16 00:00:00.000	11380.00
8	10417	2007-01-16 00:00:00.000	11283.20
9	10897	2008-02-19 00:00:00.000	10835.24
10	10353	2006-11-13 00:00:00.000	10741.60
11	10515	2007-04-23 00:00:00.000	10588.50
12	10479	2007-03-19 00:00:00.000	10495.60
13	10540	2007-05-19 00:00:00.000	10191.70
14	10691	2007-10-03 00:00:00.000	10164.80
15	11032	2008-04-17 00:00:00.000	8902.50
16	10816	2008-01-06 00:00:00.000	8891.00
17	10514	2007-04-22 00:00:00.000	8623.45
18	10912	2008-02-26 00:00:00.000	8267.40
19	10360	2006-11-22 00:00:00.000	7390.20
20	10776	2007-12-15 00:00:00.000	6984.50
21	11021	2008-04-14 00:00:00.000	6941.49
22	11017	2008-04-13 00:00:00.000	6750.00
</td>
</tr>
<tr>
<td>4</td>
<td>[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!</td>
</tr>
</table>

SQLQuery1.sql - M...IZAL RAHMAN (54))*

```sql
SELECT o.orderid, o.orderdate, SUM(od.qty * od.unitprice) AS salesamount
FROM Sales.Orders o
JOIN Sales.OrderDetails od ON o.orderid = od.orderid
GROUP BY o.orderid, o.orderdate
ORDER BY salesamount DESC;
```

100 %

Results | Messages

	orderid	orderdate	salesamount
1	10865	2008-02-02 00:00:00.000	17250.00
2	11030	2008-04-17 00:00:00.000	16321.90
3	10981	2008-03-27 00:00:00.000	15810.00
4	10372	2006-12-04 00:00:00.000	12281.20
5	10424	2007-01-23 00:00:00.000	11493.20
6	10817	2008-01-06 00:00:00.000	11490.70
7	10889	2008-02-16 00:00:00.000	11380.00
8	10417	2007-01-16 00:00:00.000	11283.20
9	10897	2008-02-19 00:00:00.000	10835.24
10	10353	2006-11-13 00:00:00.000	10741.60
11	10515	2007-04-23 00:00:00.000	10588.50
12	10479	2007-03-19 00:00:00.000	10495.60
13	10540	2007-05-19 00:00:00.000	10191.70
14	10691	2007-10-03 00:00:00.000	10164.80
15	11032	2008-04-17 00:00:00.000	8902.50
16	10816	2008-01-06 00:00:00.000	8891.00
17	10514	2007-04-22 00:00:00.000	8623.45
18	10912	2008-02-26 00:00:00.000	8267.40
19	10360	2006-11-22 00:00:00.000	7390.20
20	10776	2007-12-15 00:00:00.000	6984.50
21	11021	2008-04-14 00:00:00.000	6941.49
22	11017	2008-04-13 00:00:00.000	6750.00

Query executed successfully.

5	Compare the results in question 8 with the following image. If they are the same, then the TSQL you wrote is correct. 63 - Lab Exercise 2 - Task 2 Result.txt orderid orderdate salesamount noofoderlines avgsalesamountperorderline ---------- ---------- ---------- ---------- ---------- 10865 2008-02-02 00:00:00.000 17250,00 2 8625,00 11030 2008-04-17 00:00:00.000 16321,90 4 4080,475 10981 2008-03-27 00:00:00.000 15810,00 1 15810,00 ... ... 10586 2007-07-02 00:00:00.000 28,00 1 28,00 10807 2007-12-31 00:00:00.000 18,40 1 18,40 10782 2007-12-17 00:00:00.000 12,50 1 12,50 (830 row(s) affected)
6	[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.OrderDetils table) which is given the alias saleamountpermonth. The order of the results is based on the calculation of the

yearmonthno column.

SQLQuery1.sql - M...IZAL RAHMAN (54))*

```sql
SELECT
    CONVERT(VARCHAR(6), o.orderdate, 112) AS yearmonthno,
    SUM(od.qty * od.unitprice) AS saleamountpermonth
FROM
    Sales.Orders o
JOIN
    Sales.OrderDetails od ON o.orderid = od.orderid
GROUP BY
    CONVERT(VARCHAR(6), o.orderdate, 112)
ORDER BY
    yearmonthno;
```

100 %

Results | Messages

	yearmonthno	saleamountpermonth
1	200607	30192.10
2	200608	26609.40
3	200609	27636.00
4	200610	41203.60
5	200611	49704.00
6	200612	50953.40
7	200701	66692.80
8	200702	41207.20
9	200703	39979.90
10	200704	55699.39
11	200705	56823.70
12	200706	39088.00
13	200707	55464.93
14	200708	49981.69
15	200709	59733.02
16	200710	70328.50
17	200711	45913.36
18	200712	77476.26

7	Compare the results in question 9 with the following image. If they are the same, then the TSQL you wrote is correct. 64 - Lab Exercise 2 - Task 3 Result.txt yearmonthno saleamountpermonth ---------- ---------- 200607 30192,10 200608 26609,40 200609 27636,00 ... ... 200803 109825,45 200804 134630,56 200805 19898,66 (23 row(s) affected)

SQLQuery1.sql - M...IZAL RAHMAN (54))*

```sql
SELECT
    CONVERT(VARCHAR(6), o.orderdate, 112) AS yearmonthno,
    SUM(od.qty * od.unitprice) AS saleamountpermonth
FROM
    Sales.Orders o
JOIN
    Sales.OrderDetails od ON o.orderid = od.orderid
GROUP BY
    CONVERT(VARCHAR(6), o.orderdate, 112)
ORDER BY
    yearmonthno;
```

100 %

Results | Messages

	yearmonthno	saleamountpermonth
1	200607	30192.10
2	200608	26609.40
3	200609	27636.00
4	200610	41203.60
5	200611	49704.00
6	200612	50953.40
7	200701	66692.80
8	200702	41207.20
9	200703	39979.90
10	200704	55699.39
11	200705	56823.70
12	200706	39088.00
13	200707	55464.93
14	200708	49981.69
15	200709	59733.02
16	200710	70328.50
17	200711	45913.36
18	200712	77476.26

8	[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: 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) Sort the results by the totalsalesamount column.

SQLQuery1.sql - M...IZAL RAHMAN (54))*

```sql
SELECT
    c.custid,
    c.contactname,
    ISNULL(SUM(od.qty * od.unitprice), 0) AS totalsalesamount,
    ISNULL(MAX(od.qty * od.unitprice), 0) AS maxsalesamountperorderline,
    COUNT(*) AS numberofrows,
    COUNT(o.orderid) AS numberoforderlines
FROM
    Sales.Customers c
LEFT JOIN
    Sales.Orders o ON c.custid = o.custid
LEFT JOIN
    Sales.OrderDetails od ON o.orderid = od.orderid
GROUP BY
    c.custid,
    c.contactname
ORDER BY
    totalsalesamount DESC;
```

100 %

Results | Messages

	custid	contactname	totalsalesamount	maxsalesamountperorderline	numberofrows	numberoforderlines
1	63	Veronesi, Giorgio	117483.39	15810.00	86	86
2	71	Navarro, Tomás	115673.39	7427.40	116	116
3	20	Kane, John	113236.68	6360.00	102	102
4	37	Cr?ciun, Ovidiu V.	57317.39	9903.20	55	55
5	65	Moore, Michael	52245.90	10540.00	71	71
6	34	Cohen, Shy	34101.15	15810.00	32	32
7	24	San Juan, Patricia	32555.55	6189.50	45	45
8	51	Taylor, Maurice	32203.90	10329.20	32	32
9	39	Song, Lolan	31745.75	7905.00	39	39
10	62	Misiec, Anna	30226.10	8432.00	40	40
11	89	Smith Jr., Ronaldo	29073.45	6587.50	40	40
12	25	Carlson, Jason	28722.71	3080.00	48	48
13	5	Higginbotham, Tom	26968.15	3952.50	52	52
14	59	Meston, Tosh	26259.95	10540.00	23	23
15	76	Gulbis, Katrin	24704.40	2750.00	39	39
16	9	Raghav, Amritansh	23850.95	1500.00	44	44
17	35	Langohr, Kris	23611.58	2640.00	45	45
18	10	Bassols, Pilar Colome	22607.70	2958.00	35	35
19	44	Louverdis, George	21282.02	2228.22	39	39
20	68	Myrcha, Jacek	20033.20	4456.44	30	30
21	32	Krishnan, Venky	19711.13	7905.00	22	22
22	7	Bansal, Dushyant	19088.00	3465.00	26	26

9	Compare the results in question 10 with the following image. If they are the same, then the TSQL you wrote is correct.

65 - Lab Exercise 2 - Task 4 Result.txt

```
custid      contactname                    totalsalesamount      maxsalesamountperorderline numberofrows numberoforderlines
----------- ------------------------------ --------------------- -------------------------- ------------ ------------------
22          Bueno, Janaina Burdan, Neville NULL                  NULL                       1            0
57          Tollevsen, 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)
```

SQLQuery1.sql - M...IZAL RAHMAN (54))*

```sql
SELECT
    c.custid,
    c.contactname,
    ISNULL(SUM(od.qty * od.unitprice), 0) AS totalsalesamount,
    ISNULL(MAX(od.qty * od.unitprice), 0) AS maxsalesamountperorderline,
    COUNT(*) AS numberofrows,
    COUNT(o.orderid) AS numberoforderlines
FROM
    Sales.Customers c
LEFT JOIN
    Sales.Orders o ON c.custid = o.custid
LEFT JOIN
    Sales.OrderDetails od ON o.orderid = od.orderid
GROUP BY
    c.custid,
    c.contactname
ORDER BY
    totalsalesamount DESC;
```

100 %

Results | Messages

	custid	contactname	totalsalesamount	maxsalesamountperorderline	numberofrows	numberoforderlines
1	63	Veronesi, Giorgio	117483.39	15810.00	86	86
2	71	Navarro, Tomás	115673.39	7427.40	116	116
3	20	Kane, John	113236.68	6360.00	102	102
4	37	Cr?ciun, Ovidiu V.	57317.39	9903.20	55	55
5	65	Moore, Michael	52245.90	10540.00	71	71
6	34	Cohen, Shy	34101.15	15810.00	32	32
7	24	San Juan, Patricia	32555.55	6189.50	45	45
8	51	Taylor, Maurice	32203.90	10329.20	32	32
9	39	Song, Lolan	31745.75	7905.00	39	39
10	62	Misiec, Anna	30226.10	8432.00	40	40
11	89	Smith Jr., Ronaldo	29073.45	6587.50	40	40
12	25	Carlson, Jason	28722.71	3080.00	48	48
13	5	Higginbotham, Tom	26968.15	3952.50	52	52
14	59	Meston, Tosh	26259.95	10540.00	23	23
15	76	Gulbis, Katrin	24704.40	2750.00	39	39
16	9	Raghav, Amritansh	23850.95	1500.00	44	44
17	35	Langohr, Kris	23611.58	2640.00	45	45
18	10	Bassols, Pilar Colome	22607.70	2958.00	35	35
19	44	Louverdis, George	21282.02	2228.22	39	39
20	68	Myrcha, Jacek	20033.20	4456.44	30	30
21	32	Krishnan, Venky	19711.13	7905.00	22	22
22	7	Bansal, Dushyant	19088.00	3465.00	26	26

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	Scenario : 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 *contact name.* To carry out the experiment in this practical part 3, make sure the database is connected to "TSQL".

2	[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)?
	`SELECT`
	`YEAR ( order date ) AS orderyear ,`
	`COUNT ( orderid ) AS nooforders ,`
	`COUNT ( custid ) AS noofcustomers`
	`FROM Sales . Orders`

	GROUP BY YEAR (orderdate); QLQuery1.sql - M...IZAL RAHMAN (54))* SELECT YEAR (order date) AS orderyear , COUNT (orderid) AS nooforders , COUNT (custid) AS noofcustomers FROM Sales . Orders GROUP BY YEAR (orderdate); 0 % Messages Msg 156, Level 15, State 1, Line 2 Incorrect syntax near the keyword 'order'. Completion time: 2024-09-27T18:18:34.8486959+07:00 The reason the number of orders (nooforders) is the same as the number of customers (noofcustomers) is because the COUNT function is counting the number of rows in the Sales.Orders table for each year. Since each order has a custid, the COUNT(custid) is effectively counting the same rows as COUNT(orderid). This does not account for unique customers, just the total number of orders.
3	[Question-12] Fix the T-SQL in question-12 to show the correct number of customers who placed orders each year!

Lab –

<table>
<tr><td></td><td>
SQLQuery1.sql - M...IZAL RAHMAN (54))*

```sql
SELECT
    YEAR(orderdate) AS orderyear,
    COUNT(orderid) AS nooforders,
    COUNT(DISTINCT custid) AS noofcustomers
FROM
    Sales.Orders
GROUP BY
    YEAR(orderdate);
```

100 %

Results | Messages

	orderyear	nooforders	noofcustomers
1	2006	152	67
2	2007	408	86
3	2008	270	81
</td></tr>
<tr><td>4</td><td>
Compare the results in question 12 with the following image. If they are the same, then the TSQL you wrote is correct.

72 - Lab Exercise 3 - Task 1 Result.txt

```
orderyear   nooforders  noofcustomers
----------- ----------- -------------
2006        152         67
2007        408         86
2008        270         81

(3 row(s) affected)
```
</td></tr>
<tr><td>5</td><td>
[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,
</td></tr>
</table>

noofcustomers and nooforders respectively. Sort the results by the firstletter column!

SQLQuery1.sql - M...IZAL RAHMAN (54))*

```sql
SELECT
    LEFT(contactname, 1) AS firstletter,
    COUNT(DISTINCT custid) AS noofcustomers,
    COUNT(o.orderid) AS nooforders
FROM
    Sales.Customers c
LEFT JOIN
    Sales.Orders o ON c.custid = o.custid
GROUP BY
    LEFT(contactname, 1)
ORDER BY
    firstletter;
```

100 %

Results | Messages

	orderyear	nooforders	noofcustomers
1	2006	152	67
2	2007	408	86
3	2008	270	81

6 Compare the results in question 13 with the following image. If they are the same, then the TSQL you wrote is correct.

73 - Lab Exercise 3 - Task 2 Result.txt

```
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)
```

7 [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

Lab –

order. Use the respective alias names, nooforders, and avgsalesamountperorder.

SQLQuery1.sql - M...IZAL RAHMAN (54))*

```sql
SELECT
    c.categoryid,
    c.categoryname,
    SUM(od.qty * od.unitprice) AS totalsalesamount,
    COUNT(DISTINCT o.orderid) AS nooforders,
    AVG(od.qty * od.unitprice) AS avgsalesamountperorder
FROM
    Production.Categories c
JOIN
    Production.Products p ON c.categoryid = p.categoryid
JOIN
    Sales.OrderDetails od ON p.productid = od.productid
JOIN
    Sales.Orders o ON od.orderid = o.orderid
WHERE
    YEAR(o.orderdate) = 2008
GROUP BY
    c.categoryid, c.categoryname
ORDER BY
    totalsalesamount DESC;
```

100 %

Results | Messages

	categoryid	categoryname	totalsalesamount	nooforders	avgsalesamountperorder
1	1	Beverages	122223.75	128	814.825
2	4	Dairy Products	82803.90	90	803.9213
3	6	Meat/Poultry	60275.57	43	1310.3384
4	3	Confections	58359.73	89	561.1512
5	8	Seafood	48712.84	101	434.936
6	2	Condiments	34557.45	62	486.7246
7	7	Produce	32415.85	42	753.8569
8	5	Grains/Cereals	30422.25	55	490.6814

8

Compare the results in question 13 with the following image. If they are the same, then the TSQL you wrote is correct.

74 - Lab Exercise 3 - Task 3 Result.txt

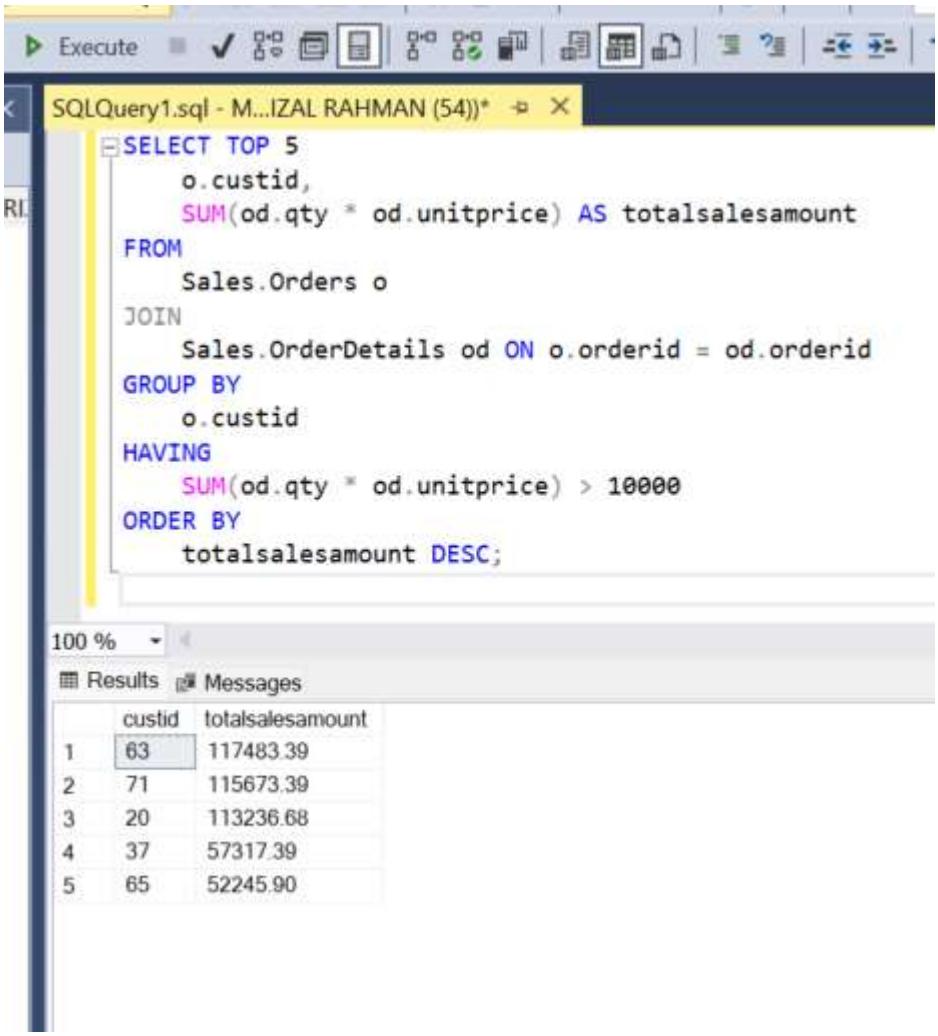
```
categoryid  categoryname    totalsalesamount      nooforders  avgsalesamountperorder
----------- --------------- --------------------- ----------- ----------------------
1           Beverages       122223,75             128         954,873
2           Condiments      34557,45              62          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)
```

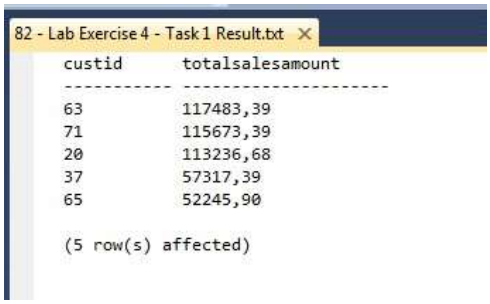
9	**Conclusion** : After carrying out this section of the practicum, students can apply the DISTINCT aggregation function.

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

Step	Information
1	Scenario : 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. To carry out the experiment in this practical part 4, make sure the database is connected to "TSQL".
2	[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.

Lab –

<table>
<tr><td></td><td>


</td></tr>
<tr><td>3</td><td>Compare the results in question 15 with the following image. If they are the same, then the TSQL you wrote is correct.

</td></tr>
<tr><td>4</td><td>[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</td></tr>
</table>

	only for orders in the year 2008!

Lab –

SQLQuery1.sql - M...IZAL RAHMAN (54))*

```sql
SELECT
    o.empid,
    o.orderid,
    SUM(od.qty * od.unitprice) AS totalsalesamount
FROM
    Sales.Orders o
JOIN
    Sales.OrderDetails od ON o.orderid = od.orderid
WHERE
    YEAR(o.orderdate) = 2008
GROUP BY
    o.empid, o.orderid;
```

100 %

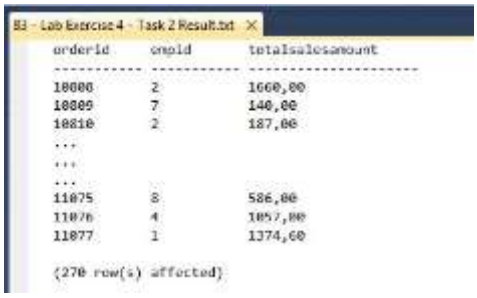
Results | Messages

	empid	orderid	totalsalesamount
1	2	10808	1660.00
2	7	10809	140.00
3	2	10810	187.00
4	8	10811	852.00
5	5	10812	1852.00
6	1	10813	648.00
7	3	10814	2070.00
8	2	10815	40.00
9	4	10816	8891.00
10	3	10817	11490.70
11	7	10818	833.00
12	2	10819	477.00
13	3	10820	1140.00
14	1	10821	678.00
15	6	10822	237.90
16	5	10823	3107.50
17	8	10824	250.80
18	1	10825	1030.76
19	6	10826	730.00
20	1	10827	843.00
21	9	10828	932.00
22	9	10829	1764.00
23	4	10830	1974.00
24	3	10831	2684.40
25	2	10832	568.95
26	6	10833	1007.70
27	1	10834	1508.12
28	1	10835	851.00
29	7	10836	4705.50
30	9	10837	1254.00
31	3	10838	2584.50
32	3	10839	919.50

Query executed successfully.

5

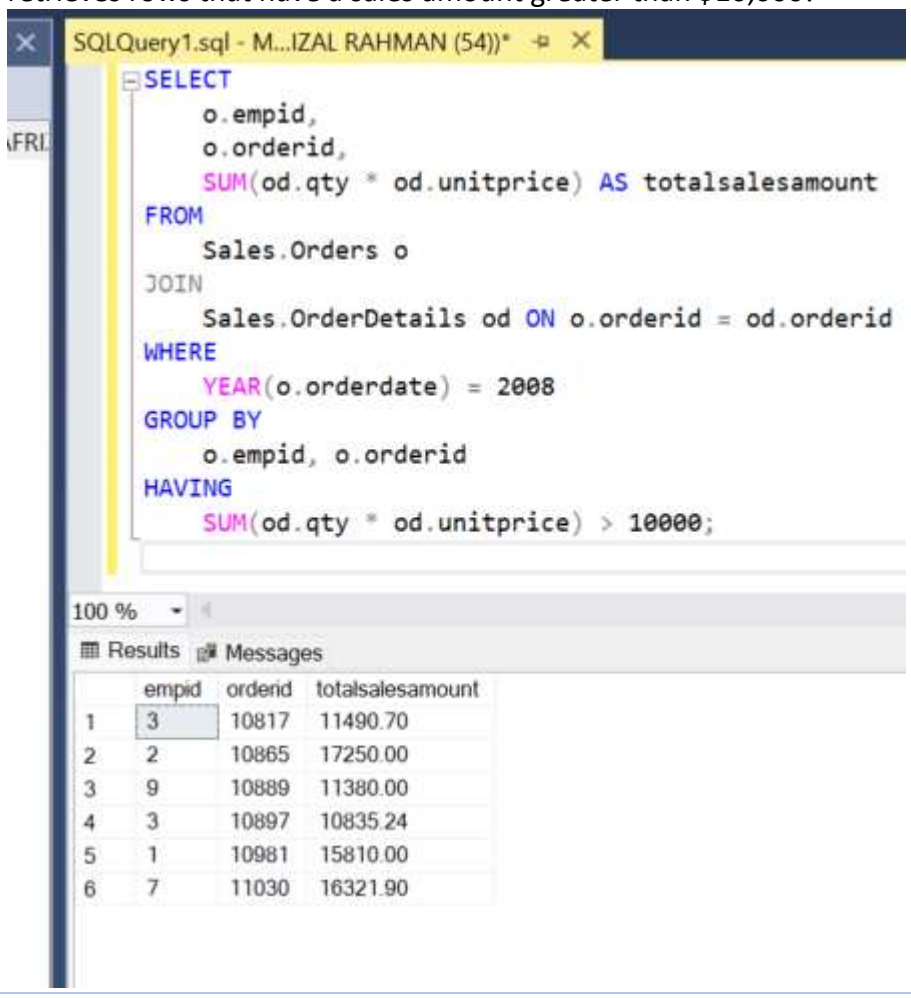
Compare the results in question 16 with the following image. If they are the same, then the TSQL you wrote is correct.



```
B3 - Lab Exercise 4 - Task 2 Result.txt
orderid     empid       totalsalesamount
----------- ----------- ---------------------
10800       2           1660,00
10809       7           140,00
10810       2           187,00
...
...
...
11075       8           586,00
11076       4           1057,00
11077       1           1374,60

(270 row(s) affected)
```

6

[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!



```
SELECT
    o.empid,
    o.orderid,
    SUM(od.qty * od.unitprice) AS totalsalesamount
FROM
    Sales.Orders o
JOIN
    Sales.OrderDetails od ON o.orderid = od.orderid
WHERE
    YEAR(o.orderdate) = 2008
GROUP BY
    o.empid, o.orderid
HAVING
    SUM(od.qty * od.unitprice) > 10000;
```

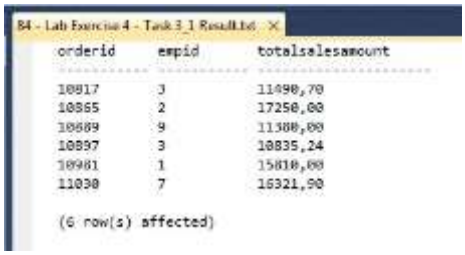
100 %

Results | Messages

	empid	orderid	totalsalesamount
1	3	10817	11490.70
2	2	10865	17250.00
3	9	10889	11380.00
4	3	10897	10835.24
5	1	10981	15810.00
6	7	11030	16321.90

7

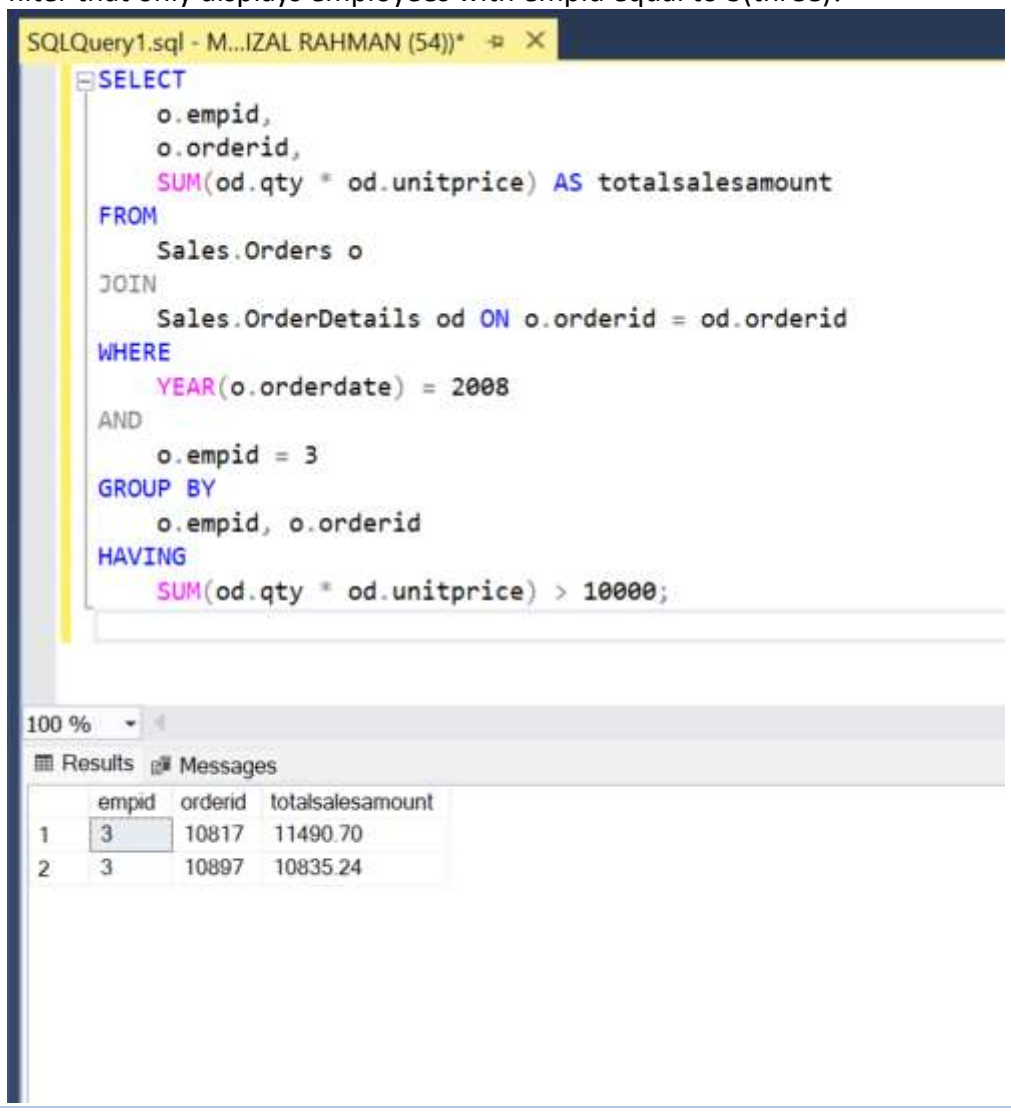
Compare the results in question 17 with the following image. If they are the same, then the TSQL you wrote is correct.



```
orderid     empid       totalsalesamount
----------- ----------- ---------------------
10817       3           11490,70
10865       2           17250,00
10889       9           11380,00
10897       3           10835,24
10981       1           15810,00
11030       7           16321,90

(6 row(s) affected)
```

8

[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)!



```
SELECT
    o.empid,
    o.orderid,
    SUM(od.qty * od.unitprice) AS totalsalesamount
FROM
    Sales.Orders o
JOIN
    Sales.OrderDetails od ON o.orderid = od.orderid
WHERE
    YEAR(o.orderdate) = 2008
AND
    o.empid = 3
GROUP BY
    o.empid, o.orderid
HAVING
    SUM(od.qty * od.unitprice) > 10000;
```

	empid	orderid	totalsalesamount
1	3	10817	11490.70
2	3	10897	10835.24

9	Compare the results in question 18 with the following image. If they are the same, then the TSQL you wrote is correct. 85 - Lab Exercise 4 - Task 3_2 Result.txt orderid empid totalsalesamount ---------- ---------- ---------- 10817 3 11490,70 10897 3 10835,24 (2 row(s) affected)
10	[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

Lab –

from the Sales.OrderDetails table!

SQLQuery1.sql - M...IZAL RAHMAN (54))*

```sql
SELECT
    o.custid,
    MAX(o.orderdate) AS lastorderdate,
    SUM(od.qty * od.unitprice) AS totalsalesamount
FROM
    Sales.Orders o
JOIN
    Sales.OrderDetails od ON o.orderid = od.orderid
GROUP BY
    o.custid
HAVING
    COUNT(o.orderid) > 25;
```

100 %

Results | Messages

	custid	lastorderdate	totalsalesamount
1	46	2008-05-05 00:00:00.000	17825.06
2	9	2008-05-06 00:00:00.000	23850.95
3	89	2008-05-01 00:00:00.000	29073.45
4	72	2008-02-04 00:00:00.000	17172.05
5	35	2008-04-28 00:00:00.000	23611.58
6	86	2008-04-23 00:00:00.000	10653.85
7	63	2008-04-14 00:00:00.000	117483.39
8	67	2008-04-29 00:00:00.000	12924.40
9	87	2008-04-15 00:00:00.000	16617.10
10	7	2008-01-12 00:00:00.000	19088.00
11	44	2008-05-05 00:00:00.000	21282.02
12	24	2008-04-27 00:00:00.000	32555.55
13	47	2008-04-21 00:00:00.000	17889.55
14	30	2008-04-21 00:00:00.000	11830.10
15	10	2008-04-24 00:00:00.000	22607.70
16	41	2008-04-27 00:00:00.000	10272.35
17	4	2008-04-10 00:00:00.000	13806.50
18	65	2008-05-06 00:00:00.000	52245.90
19	25	2008-04-09 00:00:00.000	28722.71
20	62	2008-05-04 00:00:00.000	30226.10
21	5	2008-03-04 00:00:00.000	26968.15

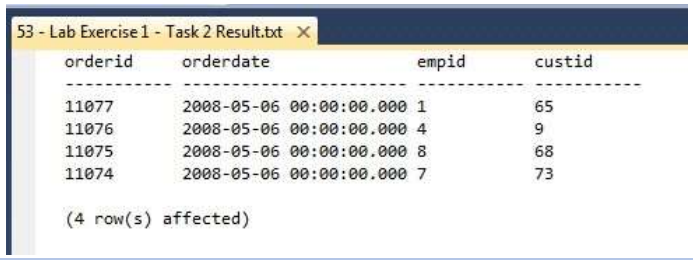
11	Compare the results in question 19 with the following image. If they are the same, then the TSQL you wrote is correct. 86 - Lab Exercise 4 - Task 4 Result.txt custid lastorderdate totalsalesamount ---------- ---------- ---------- 63 2008-04-14 00:00:00.000 117483,39 71 2008-05-01 00:00:00.000 115673,39 20 2008-05-05 00:00:00.000 113236,68 (3 row(s) affected)
12	**Conclusion** : After doing this section of the practicum, students can use the HAVING clause.

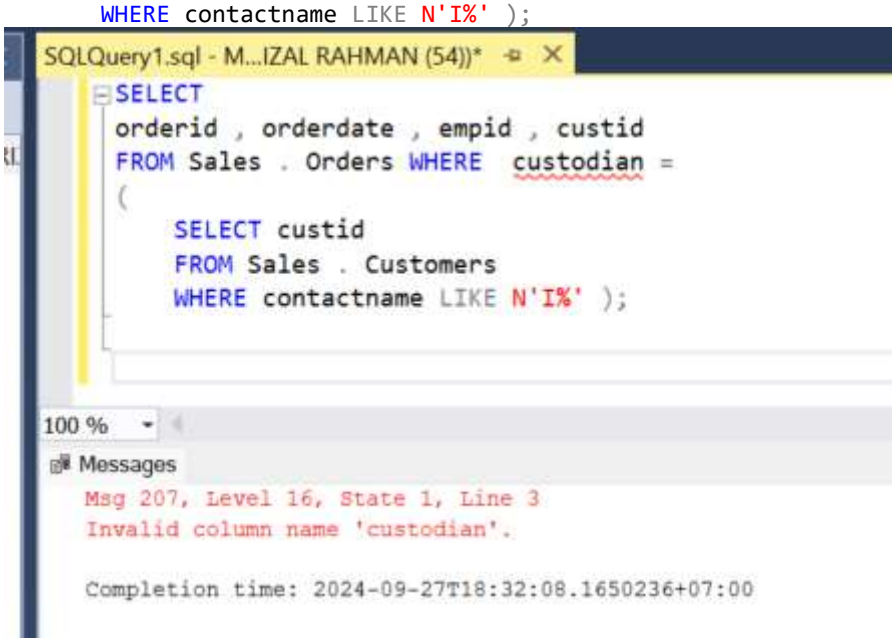
Part 5: Writing Queries Using Self-Contained Sub-queries

Step	Information
1	Scenario : The sales department needs some advanced reports to analyze sales orders. This requires a SELECT stat self-contained sub-query. To carry out the experiment in this practical part 5, make sure the database is connected to “TSQL”.

Lab –

2	[Question-20] Write a SELECT statement to display the maximum orderdate from the Sa Execute SQLQuery1.sql - M...IZAL RAHMAN (54))* SELECT MAX(orderdate) AS lastorderdate FROM Sales.Orders; 100 % Results Messages lastorderdate 1 2008-05-06 00:00:00.000
3	Compare the results in question 20 with the following image. If they are the same, then the TSQL you wro 52 - Lab Exercise 1 - Task 1 Result.txt lastorderdate ---------- 2008-05-06 00:00:00.000 (1 row(s) affected)
4	[Question-21] Write a SELECT statement to display the orderid, orderdate, empid, and custid columns fro Sales.Orders table. Then filter the results by including only orders that match the latest order time (Use th answer to question-20 as a self-contained sub-query)! SQLQuery1.sql - M...IZAL RAHMAN (54))* SELECT orderid, orderdate, empid, custid FROM Sales.Orders WHERE orderdate = (SELECT MAX(orderdate) FROM Sales.Orders); 100 % Results Messages lastorderdate 1 2008-05-06 00:00:00.000

5	Compare the results in question 21 with the following image. If they are the same, then the TSQL you wro 

6	[Question-22] Execute the T-SQL below, then modify it with a customer filter based on contact names sta letter B!
	`SELECT`
	`orderid , orderdate , empid , custid`
	`FROM Sales . Orders WHERE custodian =`
	`(`
	`SELECT custid`
	`FROM Sales . Customers`
	`WHERE contactname LIKE N'I%' );`
	 

7	[Question-23] Is there an error in the execution result of question-22? Why? Yes, there is an error because the sub-query in the original T-SQL command might return multiple rows, which is not allowed in a scalar sub-query context.
8	[Question-24] Correct the answer to question-23 so that the result is not an error!

Lab –

	The corrected T-SQL command (as shown in the modified version for Question-22) uses IN instead of = to handle multiple rows returned by the sub-query:

SQLQuery1.sql - M...IZAL RAHMAN (54))*

```
SELECT orderid, orderdate, empid, custid
FROM Sales.Orders
WHERE custid IN (
    SELECT custid
    FROM Sales.Customers
    WHERE contactname LIKE N'B%'
);
```

100 %

Results | Messages

	orderid	orderdate	empid	custid
1	10259	2006-07-18 00:00:00.000	4	13
2	10265	2006-07-25 00:00:00.000	2	7
3	10297	2006-09-04 00:00:00.000	5	7
4	10360	2006-11-22 00:00:00.000	4	7
5	10364	2006-11-26 00:00:00.000	1	19
6	10389	2006-12-20 00:00:00.000	4	10

9

Compare the results in question 24 with the following image. If they are the same, then the TSQL you wro

54 - Lab Exercise 1 - Task 3 Result.txt | 53 - Lab Exercise 1 - Task 2 Result.txt

```
orderid     orderdate               empid       custid
----------- ----------------------- ----------- -----------
10259       2006-07-18 00:00:00.000 4           13
10265       2006-07-25 00:00:00.000 2           7
10297       2006-09-04 00:00:00.000 5           7
...
...
...
11047       2008-04-24 00:00:00.000 7           19
11048       2008-04-24 00:00:00.000 7           10
11056       2008-04-28 00:00:00.000 8           19

(37 row(s) affected)
```

10

[Question-25] Write a SELECT statement to retrieve the orderid column from the Sales.Orders table and a result column:

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 f
certain period Filter the results only for orders made in

SQLQuery1.sql - M...IZAL RAHMAN (54))*

```
    SELECT
        o.orderid,
        SUM(od.qty * od.unitprice) AS totalsalesamount
    FROM
        Sales.Orders o
    JOIN
        Sales.OrderDetails od ON o.orderid = od.orderid
    WHERE
        YEAR(o.orderdate) = 2008 AND MONTH(o.orderdate) = 5
    GROUP BY
        o.orderid
)
SELECT
    orderid,
    totalsalesamount,
    (totalsalesamount / (SELECT SUM(totalsalesamount) FROM OrderTotals) * 100) AS salespctoftotal
FROM
    OrderTotals;
```

100 %

Results | Messages

	orderid	totalsalesamount	salespctoftotal
1	11064	4722.30	23.73
2	11065	252.56	1.26
3	11066	928.75	4.66
4	11067	86.85	0.43
5	11068	2384.80	11.98
6	11069	360.00	1.80
7	11070	1873.50	9.41
8	11071	510.00	2.56
9	11072	5218.00	26.22
10	11073	300.00	1.50
11	11074	244.30	1.22
12	11075	586.00	2.94
13	11076	1057.00	5.31
14	11077	1374.60	6.90

11 Compare the results in question 25 with the following image. If they are the same, then the TSQL you wro

55 - Lab Exercise 1 - Task 4 Result.txt

```
orderid     totalsalesamount      salespctoftotal
----------- --------------------- ---------------------------------------
11064       4722,30               23.7300
11065       252,56                1.2600
11066       928,75                4.6600
---
---
---
11075       586,00                2.9400
11076       1057,00               5.3100
11077       1374,60               6.9000

(14 row(s) affected)
```

12 **Conclusion** : After completing this section of the practicum, students can use self-contained subqueries in

Lab –

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

Step	Information
1	Scenario : 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. To carry out the experiment in this practical part 6, make sure the database is connected to “TSQL”.
2	[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!

```sql
SELECT
    p.productid,
    p.productname
FROM
    Production.Products p
WHERE
    p.productid IN (
        SELECT od.productid
        FROM Sales.OrderDetails od
        WHERE od.qty > 100
    );
```

100 %

Results | Messages

	productid	productname
1	24	Product QOGNU
2	27	Product SMIOH
3	39	Product LSOFL
4	41	Product TTEEX
5	45	Product AQOKR
6	51	Product APITJ
7	53	Product BKGEA
8	55	Product YYWRT
9	59	Product UKXRI
10	61	Product XYZPE
11	64	Product HCQDE

3

Compare the results in question 26 with the following image. If they are the same, then the TSQL you wrote is correct.

62 - Lab Exercise 2 - Task 1 Result.txt

```
productid   productname
----------- ----------------------------------------
24          Product QOGNU
27          Product SMIOH
39          Product LSOFL
...
...
...
61          Product XYZPE
64          Product HCQDE
75          Product BWRLG

(12 row(s) affected)
```

Lab –

4	[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!
5	Compare the results in question 27 with the following image. If they are the same, then the TSQL you wrote is correct.

SQLQuery1.sql - M...IZAL RAHMAN (54))*

```sql
SELECT
    c.custid,
    c.contactname
FROM
    Sales.Customers c
WHERE
    NOT EXISTS (
        SELECT 1
        FROM Sales.Orders o
        WHERE o.custid = c.custid
    );
```

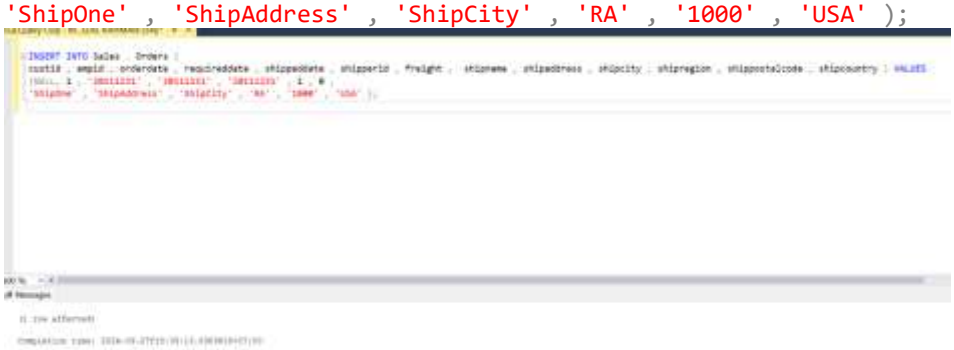
100 %

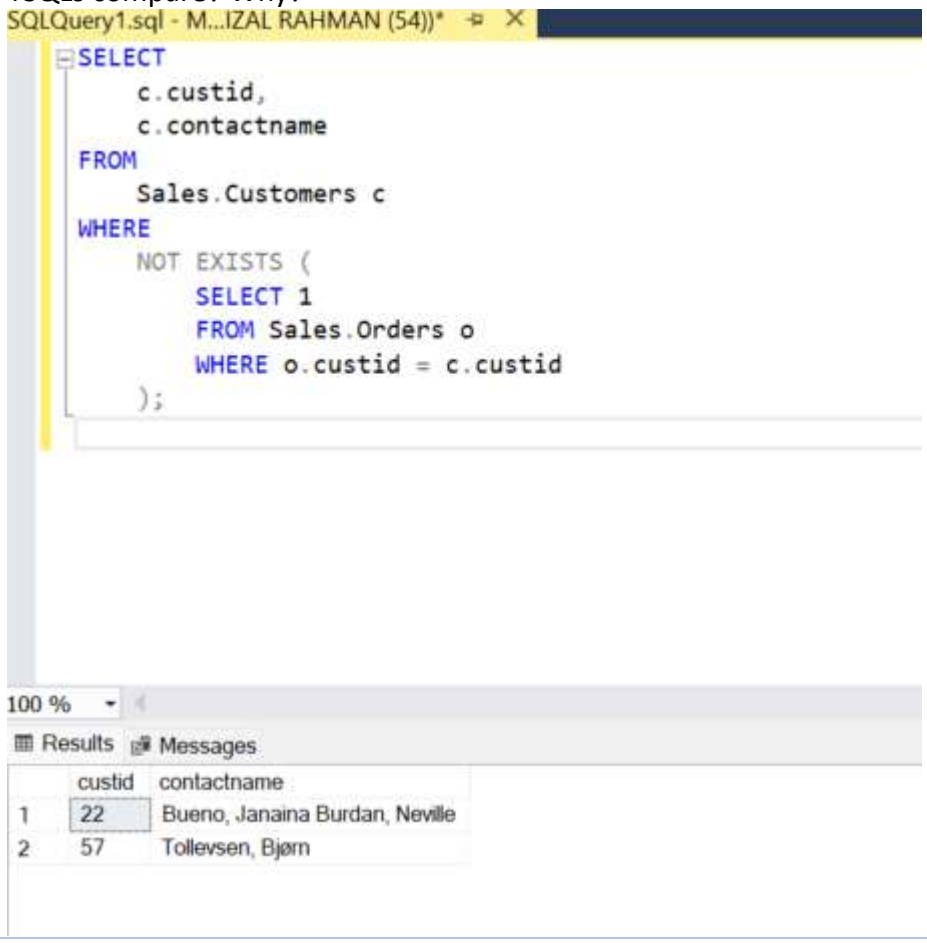
Results | Messages

	custid	contactname
1	22	Bueno, Janaina Burdan, Neville
2	57	Tollevsen, Bjørn

63 - Lab Exercise 2 - Task 2 Result.txt

```
custid      contactname
----------- ------------------------------
22          Bueno, Janaina Burdan, Neville
57          Tollevsen, Bjørn

(2 row(s) affected)
```

6	[Problem-28] There is one additional row of data in the Sales.Orders table with T-SQL as follows:
	
	`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' );`
	 

	Execute the command! Then copy the answer to question 27. How do the results of the two TSQLs compare? Why? 

Lab –

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! SQLQuery1.sql - M...IZAL RAHMAN (54))* `SELECT c.custid, c.contactname FROM Sales.Customers c WHERE c.custid NOT IN ( SELECT o.custid FROM Sales.Orders o WHERE o.custid IS NOT NULL );` 100 % Results Messages custid contactname 1 22 Bueno, Janaina Burdan, Neville 2 57 Tollevsen, Bjørn

8	Compare the results in question 29 with the following image. If they are the same, then the TSQL you wrote is correct.
	64 - Lab Exercise 2 - Task 3 Result.txt
	`custid contactname`
	`---------- ----------`
	`22 Bueno, Janaina Burdan, Neville`
	`57 Tollevsen, Bjørn`
	`(2 row(s) affected)`

9	**Conclusion** : After completing the practicum and answering the questions in this section, students can use multi-results in T-SQL statements.

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

Step	Information
1	Scenario : 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. To carry out the experiment in this practical part 7, make sure the database is connected to "TSQL".
2	[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

Lab –

	Sales.Orders table for each customer (Use a correlated sub-query).

SQLQuery1.sql - M...IZAL RAHMAN (54))*

```sql
SELECT
    c.custid,
    c.contactname,
    (SELECT MAX(o.orderdate)
     FROM Sales.Orders o
     WHERE o.custid = c.custid) AS lastorderdate
FROM
    Sales.Customers c;
```

100 %

Results | Messages

	custid	contactname	lastorderdate
1	1	Allen, Michael	2008-04-09 00:00:00.000
2	2	Hassall, Mark	2008-03-04 00:00:00.000
3	3	Peoples, John	2008-01-28 00:00:00.000
4	4	Arndt, Torsten	2008-04-10 00:00:00.000
5	5	Higginbotham, Tom	2008-03-04 00:00:00.000
6	6	Poland, Carole	2008-04-29 00:00:00.000
7	7	Bansal, Dushyant	2008-01-12 00:00:00.000
8	8	Ilyina, Julia	2008-03-24 00:00:00.000
9	9	Raghav, Amritansh	2008-05-06 00:00:00.000
10	10	Bassols, Pilar Colome	2008-04-24 00:00:00.000
11	11	Jaffe, David	2008-04-14 00:00:00.000
12	12	Ray, Mike	2008-04-28 00:00:00.000
13	13	Benito, Almudena	2006-07-18 00:00:00.000
14	14	Jelitto, Jacek	2008-04-22 00:00:00.000
15	15	Richardson, Shawn	2008-04-22 00:00:00.000
16	16	Birkby, Dana	2008-01-23 00:00:00.000
17	17	Jones, TiAnna	2008-05-04 00:00:00.000
18	18	Rizaldy, Arif	2008-02-16 00:00:00.000
19	19	Boseman, Randall	2008-04-28 00:00:00.000
20	20	Kane, John	2008-05-05 00:00:00.000
21	21	Russo, Giuseppe	2007-10-31 00:00:00.000
22	22	Bueno, Janaina Burdan, Neville	NULL
23	23	Khanna, Karan	2007-12-22 00:00:00.000
24	24	San Juan, Patricia	2008-04-27 00:00:00.000
25	25	Carlson, Jason	2008-04-09 00:00:00.000
26	26	Koch, Paul	2008-03-24 00:00:00.000
27	27	Schmöllerl, Martin	2008-04-30 00:00:00.000
28	28	Cavaglieri, Giorgio	2008-03-19 00:00:00.000

Lab –

3	Compare the results in question 30 with the following image. If they are the same, then the TSQL you wrote is correct. 72 - Lab Exercise 3 - Task 1 Result.txt custid contactname lastorderdate 1 Allen, Michael 2008-04-09 00:00:00.000 2 Hassall, Mark 2008-03-04 00:00:00.000 3 Peoples, John 2008-01-28 00:00:00.000 ... ... 89 Smith Jr., Ronaldo 2008-05-01 00:00:00.000 90 Larsson, Katarina 2008-04-07 00:00:00.000 91 Conn, Steve 2008-04-23 00:00:00.000 (91 row(s) affected)
4	[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) SQLQuery1.sql - M...IZAL RAHMAN (54))* SELECT c.custid, c.contactname FROM Sales.Customers c WHERE NOT EXISTS (SELECT 1 FROM Sales.Orders o WHERE o.custid = c.custid); 100 % Results Messages custid contactname 1 22 Bueno, Janaina Burdan, Neville 2 57 Tollevsen, Bjørn
5	Compare the results in question 31 with the following image. If they are the same, then the TSQL you wrote is correct. 73 - Lab Exercise 3 - Task 2 Result.txt 74 - Lab Exercise 3 - Task 3 Result.txt custid contactname 22 Bueno, Janaina Burdan, Neville 57 Tollevsen, Bjørn (2 row(s) affected)

6

[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!

SQLQuery1.sql - M...IZAL RAHMAN (54))*

```sql
SELECT
    c.custid,
    c.contactname
FROM
    Sales.Customers c
WHERE
    c.custid IN (
        SELECT o.custid
        FROM Sales.Orders o
        JOIN Sales.OrderDetails od ON o.orderid = od.orderid
        WHERE o.orderdate >= '2008-04-01'
        AND od.unitprice > 100
    );
```

100 %

Results | Messages

	custid	contactname
1	24	San Juan, Patricia
2	32	Krishnan, Venky
3	60	Uppal, Sunil
4	71	Navarro, Tomás
5	89	Smith Jr., Ronaldo

7

Compare the results in question-32 with the following image. If they are the same, then the TSQL you wrote is correct.

74 - Lab Exercise 3 - Task 3 Result.txt

```
custid      contactname
----------- ------------------------------
24          San Juan, Patricia
32          Krishnan, Venky
60          Uppal, Sunil
71          Navarro, Tomás
89          Smith Jr., Ronaldo

(5 row(s) affected)
```

Lab –

8	[Question-33] Write a SELECT statement that will retrieve information for each year as follows: 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: ● 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

SQLQuery1.sql - M...IZAL RAHMAN (54))*

```sql
WITH YearlySales AS (
    SELECT
        YEAR(o.orderdate) AS orderyear,
        SUM(od.qty * od.unitprice) AS totalsales
    FROM
        Sales.Orders o
    JOIN
        Sales.OrderDetails od ON o.orderid = od.orderid
    GROUP BY
        YEAR(o.orderdate)
)
SELECT
    orderyear,
    totalsales,
    (SELECT SUM(totalsales)
     FROM YearlySales ys2
     WHERE ys2.orderyear <= ys1.orderyear) AS runsales
FROM
    YearlySales ys1
ORDER BY
    orderyear;
```

100 %

Results | Messages

	orderyear	totalsales	runsales
1	2006	226298.50	226298.50
2	2007	658388.75	884687.25
3	2008	469771.34	1354458.59

Lab –

9	Compare the results in question 33 with the following image. If they are the same, then the TSQL you wrote is correct. 75 - Lab Exercise 3 - Task 4 Result.txt orderyear totalsales runsales ---------- ---------- ---------- 2006 226298,50 226298,50 2007 658388,75 884687,25 2008 469771,34 1354458,59 (3 row(s) affected)
10	**Conclusion** : After working on the practical work and questions in this section, you know how to use correlated Sub-Query in T-SQL.

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