

JOBSHEET

PRAKTIKUM BASIS DATA LANJUT

**Jurusan Teknologi Informasi
POLITEKNIK NEGERI MALANG
2024**



Week 2

SQL SERVER- SELECT, JOIN SORTING DAN FILTERING DATA

Team Teaching:

Habibie Ed Dien, S.Kom., M.T.

Irsyad Arif Mashudi, S.Kom M.Kom

Vit Zuraida, S.Kom., M.Kom.

Rokhimatul Wakhidah, S.Pd., M.T.

Annisa Taufika Firdausi, ST., MT.

Elok Nur Hamdana, S.T., M.T



Information Technology Department, Malang State
Polytechnic

Jobsheet- 1 : Introduction to Transact-SQL and Statements

SELECT, Join, Sorting, and Filtering data

Advanced Database Course

Supervisor: Advanced Database Teaching Team *September 2024*

SAFRIZAL RAHMAN_19_SIB_2G

Topics

1. Introduction to T-SQL and *Query Select*
2. Querying Multiple Tables
3. Sorting and Filtering Data

Objective

Students are expected to be able to:

1. Understanding the basic differences between Transact-SQL (T-SQL) and ANSI SQL.
2. Understanding how to create *a database* from an existing SQL file
3. Understand how to execute part or all of a SQL *script* from an existing file.
4. Understanding the concept of using '*comments*' in T-SQL.
5. Understand the concept of using the SELECT statement to analyze existing tables in *a database*.
6. Understanding how to display data in a *unique / distinct manner* .
7. Understand how to use *ALIAS* for table names and column names.
8. Understand the concept of *CASE* expressions and how to use them.
9. Students understand how to query multiple tables in a SELECT clause using JOIN.
10. Students understand how to write INNER JOIN , OUTER JOIN , SELF-JOIN and CROSS JOIN queries
- .
11. Students understand how to do Data Sorting , Data Filtering with predicates , Data Filtering with TOP and OFFSET-FETCH
12. Students understand how to handle missing and unknown values in real data.

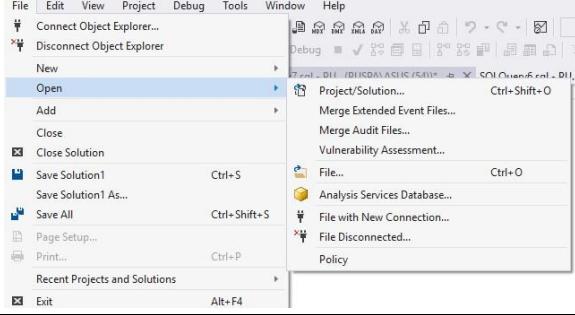
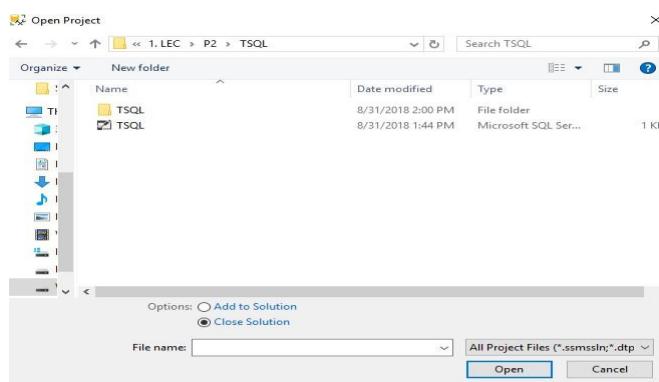
General Instructions

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.

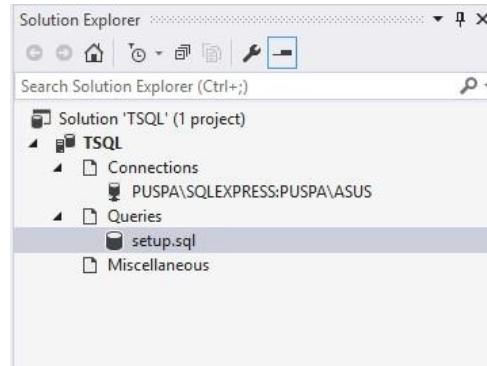
Practical Preparation: Creating a Database from Existing SQL

Step	Information
1	Create a TSQL database 
2	On the File menu , click Open and click Project/Solution . 
3	In the Open Project file.  window , open the given project



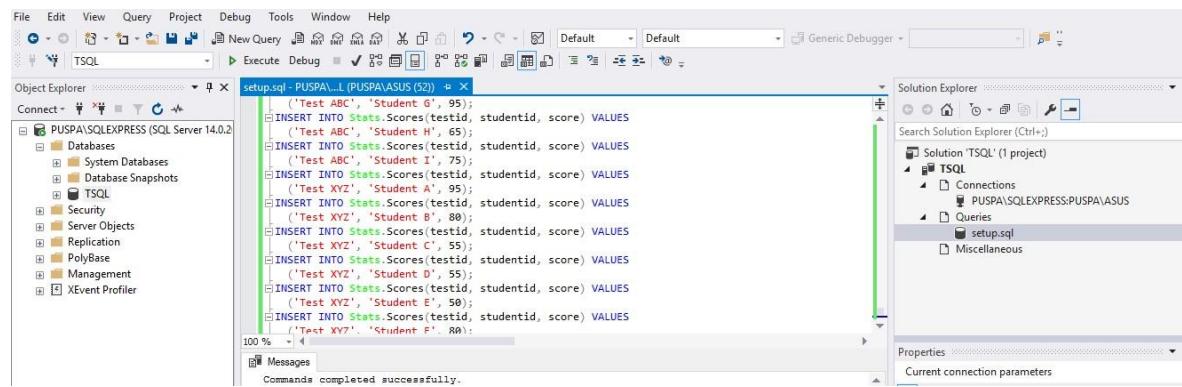
Next, the Solution Explorer window will display the following display. Then please open the “Setup” file. This file contains the sql script to create the tables needed for this practicum.

4



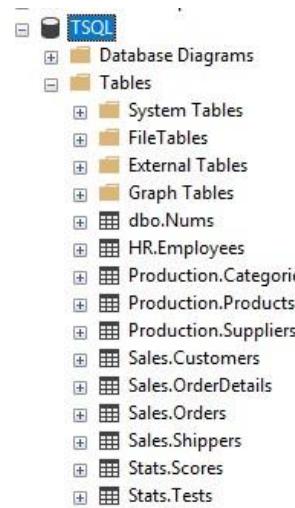
5

After the setup file is opened, a display like the image below will appear. Then click *Execute* and please wait until the process is complete.



6

After the process is successful, several tables will be formed, as shown in the image below.





7

For example, to check *records* in the Sales.Customers table, please execute the command below:

```
USE [TSQL]
GO

SELECT [custid]
      ,[companyname]
      ,[contactname]
      ,[contacttitle]
      ,[address]
      ,[city]
      ,[region]
      ,[postalcode]
      ,[country]
      ,[phone]
      ,[fax]
  FROM [Sales].[Customers]
GO
```

The results of the SQL command above are as follows

	custid	companyname	contactname	contacttitle	address	city	region
1	1	Customer NRZBB	Allen, Michael	Sales Representative	Obere Str. 0123	Berlin	NULL
2	2	Customer MLTDN	Hassall, Mark	Owner	Avda. de la Constitución 5678	México D.F.	NULL
3	3	Customer KBUDE	Peoples, John	Owner	Mataderos 7890	México D.F.	NULL
4	4	Customer HFBZG	Amdt, Torsten	Sales Representative	7890 Hanover Sq.	London	NULL
5	5	Customer HGVLZ	Higinbotham, Tom	Order Administrator	Berguvsvägen 5678	Luleå	NULL

8

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists the database structure, including databases, tables, and stored procedures. The central pane displays a query results grid and a query editor window. The query editor contains the same T-SQL code as in step 7, which retrieves customer data from the Sales.Customers table. The results grid shows 25 rows of customer information from various countries like Germany, Mexico, France, Spain, and Brazil. The bottom status bar indicates the query was executed successfully and provides runtime details.

custid	companyname	contactname	contacttitle	address	city	region
5	Customer NRZBB	Higinbotham, Tom	Order Administrator	Berguvsvägen 5678	Luleå	NULL
6	Customer MLTDN	Peoples, John	Owner	Mataderos 7890	México D.F.	NULL
7	Customer KBUDE	Amdt, Torsten	Sales Representative	7890 Hanover Sq.	London	NULL
8	Customer HFBZG	Hassall, Mark	Owner	Avda. de la Constitución 5678	México D.F.	NULL
9	Customer HGVLZ	Allen, Michael	Sales Representative	Obere Str. 0123	Berlin	NULL
10	Customer EEEVY	Ilyna, Julia	Owner	2345, place Küber	Strasbourg	NULL
11	Customer VVBUH	Raghav, Amitansh	Owner	6789, rue des Bouchers	Marseille	NULL
12	Customer EEEVY	Bassols, Pilar Colomé	Accounting Manager	8901 Trawassasse Blvd	Tsawassen	BC
13	Customer VVBUH	De Walle, David	Sales Representative	Faubourg Cirrus 4567	London	UK
14	Customer PSNMQ	Reinhold, Werner	Sales Agent	5678, rue du Commerce	Buenos Aires	NULL
15	Customer VVBUH	Ray, Nancy	Sales Agent	3456, rue des Quinze Otages	Nantes	NULL
16	Customer VVBUH	Bento, Almudena	Marketing Manager	Serrras de Granada 7890	México D.F.	Mexico
17	Customer VVBUH	Jeffito, Jacek	Owner	123 Main Street	Bern	NULL
18	Customer VVBUH	Richardson, Shuan	Sales Associate	Avenida Lameiras, 6789	Sao Paulo	SP
19	Customer VVBUH	Bekky, Dana	Sales Representative	Berkley Gardens 0123 Brewery	London	NULL
20	Customer VVBUH	Rizaldy, Afif	Owner	3456, rue des Quinze Otages	Aachen	NULL
21	Customer VVBUH	Kane, John	Sales Manager	3456, rue des Quinze Otages	Nantes	NULL
22	Customer VVBUH	Russo, Giuseppe	Marketing Assistant	5678 King George	Graz	NULL
23	Customer VVBUH	Bueno, Janana Bur.	Accounting Manager	Kirchasse 9012	London	NULL
24	Customer VVBUH	Khanna, Karan	Assistant Sales Ag.	Rua Orléas, 3456	Graz	NULL
25	Customer VVBUH	San Juan, Patricia	Owner	C/ Moratalaz, 5678	Sao Paulo	SP
		Carlson, Jason	Marketing Manager	4567, chaussée de Touraine	Madrid	NULL



Practical –

Part 1: Executing part or all of a SQL script

Step	Information
1	<p>Please type the following <i>query</i> in your <i>query panel</i> then click <i>execute</i> . Note the results displayed.</p> <pre>SELECT * FROM Sales.Customers;</pre>
2	<p>Please add the following <i>query</i> to your <i>query panel</i> then click <i>execute</i> . Note the results</p> <pre>SELECT * FROM Sales.Customers;</pre> <pre>SELECT custid, companyname, contactname, contacttitle, address, city, region, postalcode, country, phone, fax FROM Sales.Customers;</pre>



Practical –

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists the database structure, including the 'Sales' schema which contains the 'Customers' table. The 'Results' tab in the center displays the output of a SELECT query:

```
SELECT * FROM Sales.Customers;
```

CustomerID	CustomerName	ContactName	ContactTitle	Address	City	Region	PostalCode	Country	Phone	Fax
1	Customer NR2BB	Allen, Michael	Sales Representative	Obersee Str. 9223	Berlin	NULL	10082	Germany	(03) 3456789	030-0123456
2	Customer MLLTON	Hassall, Mark	Owner	Avenida de la Constitución 5678	México D.F.	NULL	10077	Mexico	(5) 789-0123	(5) 456-7890
3	Customer KB1DE	Peebles, John	Owner	Matadero 7890	México D.F.	NULL	10097	Mexico	(5) 123-4567	NULL
4	Customer HF1BGZ	Arndt, Torsten	Sales Representative	7890 Hanover Sq.	London	NULL	10046	UK	(171) 456-7890	(171) 456-7891
5	Customer HGWLZ	Hegghotham, Tom	Order Administrator	Berguvsgatan 5678	Luleå	NULL	10112	Sweden	0921-6789 01	0921-23 45 67
6	Customer QJ1WH	Wenche, Yngvild	Sales Representative	Perlungsgatan 12345	Munich	NULL	10117	Germany	(081) 5678 901	(081) 5678 905
7	Customer QX1LA	Bansal, Chudrang	Marketing Manager	2345, place Kébar	Strasbourg	NULL	10080	France	67 89 01 23	67 89 01 24
8	Customer QU1WH	Ilyana, Julia	Owner	C/ Aragón 0123	Madrid	NULL	10104	Spain	(91) 345 67 89	(91) 012 34 96
9	Customer RTXGC	Raghav, Amritansh	Owner	6789, rue des Bouchers	Marseille	NULL	10106	France	23 45 67 89	23 45 67 80
10	Customer EEA1V	Bassett, Pilar Colome	Accounting Manager	8901 Tswewesen Blvd.	Tsawwassen	BC	10111	Canada	(604) 901-2345	(604) 678-9012
11	Customer UBAUJ	Jaffe, David	Sales Representative	Faurency Circus 4567	London	NULL	10064	UK	(171) 789-0123	NULL
12	Customer VMLDG	Reuter, Sven	Sales Agent	8901 Tswewesen Blvd.	Buenos Aires	NULL	10065	Argentina	(10) 123-4567	(10) 123-4567
13	Customer VNLQG	Bento, Ahmadina	Marketing Manager	Sierras de Granada 7890	México D.F.	NULL	10056	Mexico	(5) 456-7890	(5) 123-4567
14	Customer VNMAF	Jelitto, Jacek	Owner	Haupstr. 0123	Bern	NULL	10065	Switzerland	0452-678901	NULL
15	Customer JUWXXX	Richardson, Shawna	Sales Associate	Air dos Luxedas, 6789	Sao Paulo	SP	10087	Brazil	(11) 012-3456	NULL
16	Customer GYBVB	Bekky, Dana	Sales Representative	Bentley Gardens 0123 Brewery	London	NULL	10039	UK	(171) 234-5678	(171) 234-5679
17	Customer JUWXXX	Gomes, Telma	Order Administrator	8901 Tswewesen Blvd.	München	NULL	10072	Germany	02 123 45 67	02 123 45 68
18	Customer BSVAR	Parker, Ted	Owner	3456, rue des Quinze Otages	Nantes	NULL	10041	France	89 01 23 45	89 01 23 46
19	Customer RFNQC	Bosseman, Randall	Sales Agent	5678 King George	London	NULL	10110	UK	(171) 345-4789	(171) 345-4780
20	Customer TH1DP	Kane, John	Sales Manager	Kirchasse 9012	Graz	NULL	10059	Austria	1234-5678	9012-3456
21	Customer KDPX	Russo, Giuseppe	Marketing Assistant	Rua One, 3456	Sao Paulo	SP	10099	Brazil	(11) 456-7890	NULL

MSI (14.0 RTM) | MS(SAFRIZAL RAHMAN (S4)) | TSQL | 00:00:00 | 91 rows



Practical –

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer pane, there are two open queries:

```

File Edit View Query Project Tools Window Help
New Query Execute Options Find Replace Go Back Go Forward Refresh Stop Stop All Stop Task List Stop Task
Object Explorer
MS (SQL Server 14.0.2056.2 - MSA)
Databases
System Databases
Database Snapshots
TSQL
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
o.Nums
HR.Employees
Production.Categories
Columns
Keys
Constraints
Triggers
Indexes
Statistics
Production.Products
Production.Suppliers
Sales.Customers
Sales.OrderDetails
Sales.Orders
Sales.Shippers
Stats.Scores
Stats.Tests
External Resources
Synonyms
Programmability
Service Broker
Storage
Security
Server Objects

```

The first query in the Results pane is:

```

SELECT
    custid, companyname, contactname, contacttitle, address, city, region, postalcode,
    country, phone, fax
FROM Sales.Customers;

```

The second query in the Results pane is:

```

SELECT
    custid, companyname, contactname, contacttitle, address, city, region, postalcode,
    country, phone, fax
FROM Sales.Customers;

```

The results of the second query are displayed in a grid:

	custid	companyname	contactname	contacttitle	address	city	region	postalcode	country	phone	fax
1	1	Customer NR289	Allen, Michael	Sales Representative	Obere Str. 92/123	Berlin	NULL	10082	Germany	(030) 3456789	(030) 6123456
2	2	Customer MLTON	Hassall, Mark	Owner	Avia de la Constitución 5678	México D.F.	NULL	10077	Mexico	(5) 769-0123	(5) 456-7890
3	3	Customer KBJDE	Popek, John	Owner	Matadero 7890	México D.F.	NULL	10097	Mexico	(5) 123-4567	NULL
4	4	Customer HBZGZ	Andrt, Torsten	Sales Representative	7890 Hanover Sq	London	NULL	10040	UK	(171) 456-7891	(171) 456-7891
5	5	Customer HZBTR	Hogemann, Tom	Order Administrator	Bergweg 123	Lübeck	NULL	10012	Sweden	(031) 456-7890	(031) 456-67
6	6	Customer QHJYV	Borsig, Dorothy	Sales Representative	Perthesstr. 7890	Mannheim	NULL	10111	Germany	(061) 456-7890	(061) 456-12345
7	7	Customer QXWLA	Ilyna, Julia	Marketing Manager	Zürichstrasse 2345	Strasbourg	NULL	10088	France	67.88.01.23	67.88.01.24
8	8	Customer QWHWH	Cormier, Corinne	Marketing Manager	C/ Aragó 0123	Madrid	NULL	10104	Spain	(91) 345.67.89	(91) 012.34.56
9	9	Customer RTXGC	Raghav, Amitansh	Owner	6789, rue des Bouchers	Marseille	NULL	10105	France	23.45.67.89	23.45.67.80
10	10	Customer EEAJV	Bassols, Pilar Colom	Accounting Manager	8901 Tsawassen Blvd.	Tsawassen	BC	10111	Canada	(604) 901-2345	(604) 678-9012

Below the results, a message indicates: "Query executed successfully."

displayed

Make a selection on one of the existing *queries* then click execute . Note the results displayed. **What is the difference with the results in the second step above? (Question 1)**

SELECT

custid, companyname, contactname, contacttitle, address, city, region, postalcode, country, phone, fax

FROM Sales.Customers;

3

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer pane, there are two open queries:

```

File Edit View Query Project Tools Window Help
New Query Execute Options Find Replace Go Back Go Forward Refresh Stop Stop All Stop Task List Stop Task
Object Explorer
MS (SQL Server 14.0.2056.2 - MSA)
Databases
System Databases
Database Snapshots
TSQL
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
o.Nums
HR.Employees
Production.Categories
Columns
Keys
Constraints
Triggers
Indexes
Statistics
Production.Products
Production.Suppliers
Sales.Customers
Sales.OrderDetails
Sales.Orders
Sales.Shippers
Stats.Scores
Stats.Tests
External Resources
Synonyms
Programmability
Service Broker
Storage
Security
Server Objects

```

The first query in the Results pane is:

```

SELECT
    custid, companyname, contactname, contacttitle, address, city, region, postalcode,
    country, phone, fax
FROM Sales.Customers;

```

The second query in the Results pane is:

```

SELECT
    custid, companyname, contactname, contacttitle, address, city, region, postalcode,
    country, phone, fax
FROM Sales.Customers;

```



Practical –

SQLQuery2.sql - MSLTSQL (MSI\SAFRIZAL RAHMAN (54)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Quick Launch (Ctrl+Q)

Object Explorer

Connect to... TSQL

SQLQuery2.sql M...IZAL RAHMAN (54) * SQLQuery1.sql M...IZAL RAHMAN (55)*

SELECT
 custid, companyname, contactname, contacttitle, address, city, region, postalcode,
 country, phone, fax
FROM Sales.Customers;

100 %

Results # Messages

custid	companyname	contactname	contacttitle	address	city	region	postalcode	country	phone	fax
1	Customer MILTON	Arlen Michael	Sales Representative	Ober Str. 123	Berlin	Nord	10992	Germany	(0) 30-456-7898	001-0123456
2	Customer BLONDE	Hauska, Mark	Owner	Avenida de la Constitución 5678	México D.F.	Niúll	10077	Mexico	(0) 789-0123	(0) 456-7890
3	Customer KUBLDE	Pepites, John	Owner	Mataderos 7890	México D.F.	Niúll	10097	Mexico	(0) 123-4567	NULL
4	Customer HFBZG	Arndt, Torsten	Sales Representative	7890 Hanover Sq	London	Niúll	10046	UK	(171) 456-7898	(171) 456-7891
5	Customer HOWLZ	Higginbottom, Tom	Order Administrator	Bergweg 289	Luleå	Niúll	10112	Sweden	0921-67 8910	0921-23 45 67
6	Customer CUSTYAR	Cardle, Jennifer	Sales Representative	2345, place Kléber	Mannheim	Niúll	10099	Germany	0611-123456	0611-123456
7	Customer CUSTYCA	Banke, Markward	Marketing Manager	2345, place Kléber	Strasbourg	Niúll	10089	France	070-01 23	67 8910 24
8	Customer CUSTYHW	Birya, Jutta	Owner	C/Andrés 0123	Madrid	Niúll	10104	Spain	(01) 456-7890	(01) 456-7890
9	Customer RTIGC	Raghav, Ananthash	Owner	6789, rue des Touches	Marseille	Niúll	10105	France	23 45 67 89	23 45 67 89
10	Customer EAEVL	Bassols, Pilar Colomé	Accounting Manager	8901 Tsawassen Blvd	Tsawassen	BC	10111	Canada	(604) 901-2345	(604) 678-9012
11	Customer CUSTPBG	Jaffe, David	Sales Representative	Fauntleroy Circus 4567	London	Niúll	10049	UK	(171) 789-0123	NULL
12	Customer VMLQG	Bentz, Almudena	Sales Agent	1234, rue du Commerce	Buenos Aires	Niúll	10057	Argentina	(01) 800-1234	(01) 800-1234
13	Customer WMMAP	Jelatto, Jacek	Marketing Manager	Serrana de Granada 7890	Mendoza D.F.	Niúll	10058	Argentina	(01) 800-1234	(01) 800-1234
14	Customer RFNQC	Kane, John	Owner	Hauptstr. 0123	Bern	Niúll	10065	Switzerland	0424-678901	NULL
15	Customer JWWKK	Richardson, Shawne	Sales Associate	Av dos Usinas, 6789	Sao Paulo	SP	10087	Brazil	(11) 210-3456	NULL
16	Customer GYBPK	Birkby, Dana	Sales Representative	Berkley Gates 0123 Brewery	London	Niúll	10039	UK	(171) 234-5678	(171) 234-5679
17	Customer FEVNN	Jones, Tania	Order Administrator	Watsenweg 4567	Aachen	Niúll	10067	Germany	0241-789012	0241-345678
18	Customer CUSTYCH	Chapman, Lorraine	Owner	1234, rue de la Paix	Paris	Niúll	10068	France	01 23 45 67 89	01 23 45 67 89
19	Customer CUSTYR	Bosman, Randall	Sales Agent	5678, King George	London	Niúll	10110	UK	(171) 345-6789	(171) 345-6780
20	Customer THDHP	Kane, John	Sales Manager	Kirchspie 9012	Graz	Niúll	10059	Austria	1234-5678	9012-3456
21	Customer KIDPX	Russo, Giuseppe	Marketing Assistant	Rua Orós, 3456	Sao Paulo	SP	10096	Brazil	(11) 456-7890	NULL

Query executed successfully.

MSI (14.0 RTM) MSI\SAFRIZAL RAHMAN (54) TSQL 00:00:00 91 rows

The two SQL queries you've provided are similar but differ in the level of detail they return:

1. First Query:

```
sql  
SELECT  
FROM S
```

- This query selects all columns from the Sales.Customers table. The asterisk (*) is a wildcard that tells the database to return every column for each row in the table.

2. Second Query:

```
sql  
SELECT  
    custid,  
    country  
FROM S
```

- This query selects specific columns from the Sales.Customers table. It explicitly lists the columns custid, companyname, contactname, contacttitle, address, city, region.



Practical –

	<p>postalcode, country, phone, and fax, meaning only these columns will be returned in the result set.</p> <p>Difference in Results:</p> <ul style="list-style-type: none">- The first query (SELECT) will return all columns in the Sales.Customers table, regardless of how many columns the table contains.- The second query will return only the specified columns, which might be fewer than the total number of columns in the table. <p>What to Observe in Execution:</p> <p>When you execute these queries:</p> <ul style="list-style-type: none">- For the first query, you'll see a result set that includes every column available in the Sales.Customers table.- For the second query, you'll see a more focused result set, showing only the columns explicitly listed in the query. <p>This difference is important when you want to limit the data returned, especially if you're only interested in certain attributes of the customers and not the entire dataset.</p>
4	In the query panel please type <pre>SELECT * FROM</pre>
5	then on the Object Explorer tab – Tables please find the Sales.Customers table. Click the table and drag it to the query pane I . The result is as shown below, after that add a semicolon after the name of the table in question and click execute. <pre>SELECT * FROM [Sales].[Customers];</pre>



Practical –

The screenshot shows the SQL Server Management Studio interface. In the top pane, there are two tabs: 'SQLQuery2.sql - M...IZAL RAHMAN (54)*' and 'SQLQuery1.sql - M...IZAL RAHMAN (55)*'. The second tab is active and contains the following SQL script:

```
SELECT *
FROM
[Sales].[Customers];
```

In the bottom pane, under the 'Results' tab, the output of the query is displayed in a grid. The columns are: custid, companyname, contactname, contacttitle, address, city, region, postalcode, country, phone, and fax. The data includes 8 rows of customer information from the Sales.Customers table.

custid	companyname	contactname	contacttitle	address	city	region	postalcode	country	phone	fax
1	Customer NRZBB	Allen, Michael	Sales Representative	Obere Str. 0123	Berlin	NULL	10092	Germany	030-3456789	030-0123456
2	Customer MLTON	Hassall, Mark	Owner	Avenida de la Constitución 5678	México D.F.	NULL	10077	Mexico	(5) 789-0123	(5) 456-7890
3	Customer KBUDE	Peoples, John	Owner	Mataderos 7890	México D.F.	NULL	10097	Mexico	(5) 123-4567	NULL
4	Customer HFBZG	Arndt, Torsten	Sales Representative	7890 Hanover Sq.	London	NULL	10046	UK	(171) 456-7890	(171) 456-7891
5	Customer HGVLZ	Higinbotham, Tom	Order Administrator	Berguvsvägen 5678	Luleå	NULL	10112	Sweden	0921-67 89 01	0921-23 45 67
6	Customer XHXJV	Poland, Carole	Sales Representative	Forsterstr. 7890	Mannheim	NULL	10117	Germany	0621-67890	0621-12345
7	Customer QXVLA	Bansal, Dushyant	Marketing Manager	2345, place Kléber	Strasbourg	NULL	10089	France	67.89.01.23	67.89.01.24
8	Customer QUHWH	Ivina, Julia	Owner	C/ Araujo, 0123	Madrid	NULL	10104	Spain	(91) 345 67 89	(91) 012 34 56

Part 2: Using the SELECT statement for specific columns

Step	Information
1	In the query panel, please type the script below <pre>SELECT contactname, address, postalcode, city, country FROM Sales.Customers;</pre>
2	<i>Highlights query above and click execute</i>

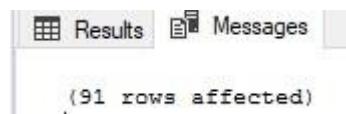
**Practical –****3**

Please observe the results. How many *rows* are produced? To find out, you can do it on the results tab as shown in the image below

	contactname	address	postalcode	city	country
1	Allen, Michael	Obere Str. 0123	10092	Berlin	Germany
2	Hassall, Mark	Avda. de la Constitución 5678	10077	México D.F.	Mexico
3	Peoples, John	Mataderos 7890	10097	México D.F.	Mexico
4	Andt, Torsten	7890 Hanover Sq.	10046	London	UK
5	Higinbotham, Tom	Berguvsvägen 5678	10112	Luleå	Sweden
6	Poland, Carole	Forsterstr. 7890	10117	Mannheim	Germany
7	Bansal, Dushyant	2345, place Kléber	10089	Strasbourg	France
8	Ilyina, Julia	C/ Araquil, 0123	10104	Madrid	Spain
9	Raghav, Amritansh	6789, rue des Bouchers	10105	Marseille	France
10	Bassols, Pilar Colome	8901 Tsawassen Blvd.	10111	Tsawassen	Canada
11	Jaffe, David	Fauntleroy Circus 4567	10064	London	UK

Query executed successfully. | PUSPA\SQLEXPRESS (14.0 RTM) | PUSPA\ASUS (53) | TSQL | 00:00:00 | 91 rows

Or you can also go to the messages tab as shown in the image below.

**Part 3: Using the SELECT statement to display data uniquely / DISTINCT**

Step	Information
1	In the query panel, please type the script below <pre>SELECT country FROM Sales.Customers;</pre>
2	Highlights query above and click execute
3	Please observe the results. Is there any duplicate data? If YES, why? Capture the results of executing the SQL script above



Practical –

SQLQuery2.sql - M...IZAL RAHMAN (54)*		X	SQLQuery1.sql - M...IZAL RAHMAN (55)*
100 %	SELECT	country	
Results	Messages		
	country		
45	USA		
46	Venezuela		
47	Venezuela		
48	USA		
49	Italy		
50	Belgium		
51	Canada		
52	Germany		
53	UK		
54	Argentina		
55	USA		
56	Germany		
57	France		
58	Mexico		
59	Austria		
60	Portugal		
61	Brazil		
62	Brazil		
63	Germany		
64	Argentina		
65	USA		
66	Italy		
67	Brazil		
68	Switzerland		
69	Spain		
70	Norway		
71	USA		
72	UK		
73	Denmark		
74	France		
75	USA		
76	Belgium		
77	USA		
78	USA		
79	Germany		
80	Mexico		
81	Brazil		
82	USA		

Duplicate

The query you executed:

```
sql  
SELECT country  
FROM Sales.Customers;
```

returns a list of all the countries associated with the customers in the Sales.Customers table. The result you've provided shows the country for each customer record in the table.

What the Query Does:



Practical –

	<p>- SELECT country: This part specifies that you want to retrieve the country column.</p> <p>- FROM Sales.Customers;: This indicates the table from which you want to retrieve the data.</p> <p>Understanding the Output:</p> <p>- Each row in the result corresponds to a country value from a customer record.</p> <p>- The same country might appear multiple times because multiple customers can be from the same country.</p> <p>Additional Considerations:</p> <p>If you want to see a list of unique countries (i.e., each country appearing only once), you can modify the query using DISTINCT:</p> <pre>sql SELECT DISTINCT country FROM Sales.Customers;</pre> <p>This will return each country only once, regardless of how many customers are from that country.</p> <p>If you want to count how many customers are from each country, you could use:</p> <pre>sql SELECT country, COUNT() as customer_count FROM Sales.Customers GROUP BY country;</pre> <p>This query will return each country along with the number of customers from that country.</p>
4	<p>In <i>the query pane</i>, please type <i>the script</i> below.</p> <pre>SELECT DISTINCT country FROM Sales.Customers;</pre> <p>Please click <i>execute</i> and observe the results.</p>



Practical –

Is there any duplicate data? Explain the difference in results in step 4 and step 3!? What are the benefits of the DISTINCT command? Capture the results of executing the SQL script above (Question 3)

If we use (DISTINCT) returns the unique values of a specified column in the query results.

Each row in the results corresponds to a country value from the customer data.

The same country may appear multiple times because some customers may be from the same country.

Therefore it can be concluded that distinct reduces duplication

5

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays the database structure of 'MSI SQL Server 14.0.2056.2 - MSI\SA'. In the center, the 'SQLQuery2.sql' window contains the following SQL code:

```
SELECT DISTINCT country
FROM Sales.Customers;
```

The results pane below shows the output of the query, listing 21 unique countries:

country
Argentina
Austria
Belgium
Brazil
Canada
Denmark
Finland
France
Germany
Ireland
Italy
Mexico
Norway
Poland
Portugal
Spain
Sweden
Switzerland
UK
USA
Venezuela

At the bottom of the results pane, a green status bar indicates: 'Query executed successfully.'



Practical –

--	--

Part 4: Using *ALIAS* for table names and column names

Step	Information																																								
1	<p>In the query panel, please type the script below</p> <pre>SELECT c.contactname, c.contacttitle FROM Sales.Customers AS c;</pre>																																								
2	<p>Highlights query above and click execute. Observe the results</p> <p>The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a tree view of databases, tables, and other objects. The main window contains a query editor with the following T-SQL code:</p> <pre>SELECT c.contactname, c.contacttitle FROM Sales.Customers AS c;</pre> <p>Below the query editor is a results grid titled "Results". It displays two columns: "contactname" and "contacttitle". The data consists of 10 rows, each representing a customer from the Sales.Customers table. The results are as follows:</p> <table border="1"><thead><tr><th>contactname</th><th>contacttitle</th></tr></thead><tbody><tr><td>Allen, Michael</td><td>Sales Representative</td></tr><tr><td>Hassall, Mark</td><td>Owner</td></tr><tr><td>Peoples, John</td><td>Owner</td></tr><tr><td>Andt, Torsten</td><td>Sales Representative</td></tr><tr><td>Higinbotham, Tom</td><td>Order Administrator</td></tr><tr><td>Poland, Carole</td><td>Sales Representative</td></tr><tr><td>Bansal, Dushyant</td><td>Marketing Manager</td></tr><tr><td>Ilyna, Julia</td><td>Owner</td></tr><tr><td>Raghav, Amitansh</td><td>Owner</td></tr><tr><td>Bassols, Pilar Coloma</td><td>Accounting Manager</td></tr><tr><td>Jaffe, David</td><td>Sales Representative</td></tr><tr><td>Ray, Mike</td><td>Sales Agent</td></tr><tr><td>Benito, Almudena</td><td>Marketing Manager</td></tr><tr><td>Jelitto, Jacek</td><td>Owner</td></tr><tr><td>Richardson, Shawn</td><td>Sales Associate</td></tr><tr><td>Birkby, Dana</td><td>Sales Representative</td></tr><tr><td>Jones, Tianna</td><td>Order Administrator</td></tr><tr><td>Rizaldy, Arif</td><td>Owner</td></tr><tr><td>Annaman, Ranjith</td><td>Sales Agent</td></tr></tbody></table> <p>A yellow bar at the bottom of the results grid says "Query executed successfully."</p>	contactname	contacttitle	Allen, Michael	Sales Representative	Hassall, Mark	Owner	Peoples, John	Owner	Andt, Torsten	Sales Representative	Higinbotham, Tom	Order Administrator	Poland, Carole	Sales Representative	Bansal, Dushyant	Marketing Manager	Ilyna, Julia	Owner	Raghav, Amitansh	Owner	Bassols, Pilar Coloma	Accounting Manager	Jaffe, David	Sales Representative	Ray, Mike	Sales Agent	Benito, Almudena	Marketing Manager	Jelitto, Jacek	Owner	Richardson, Shawn	Sales Associate	Birkby, Dana	Sales Representative	Jones, Tianna	Order Administrator	Rizaldy, Arif	Owner	Annaman, Ranjith	Sales Agent
contactname	contacttitle																																								
Allen, Michael	Sales Representative																																								
Hassall, Mark	Owner																																								
Peoples, John	Owner																																								
Andt, Torsten	Sales Representative																																								
Higinbotham, Tom	Order Administrator																																								
Poland, Carole	Sales Representative																																								
Bansal, Dushyant	Marketing Manager																																								
Ilyna, Julia	Owner																																								
Raghav, Amitansh	Owner																																								
Bassols, Pilar Coloma	Accounting Manager																																								
Jaffe, David	Sales Representative																																								
Ray, Mike	Sales Agent																																								
Benito, Almudena	Marketing Manager																																								
Jelitto, Jacek	Owner																																								
Richardson, Shawn	Sales Associate																																								
Birkby, Dana	Sales Representative																																								
Jones, Tianna	Order Administrator																																								
Rizaldy, Arif	Owner																																								
Annaman, Ranjith	Sales Agent																																								



Practical –

3

In the query panel, please type the script below.

```
SELECT  
    c.contactname AS Name, c.contacttitle AS Title, c.companyname AS [Company Name]  
FROM Sales.Customers AS c;
```

4

Highlights query above and click execute . Observe the results.

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays the database structure, including the Sales database and its tables like Customers, Employees, and Products. The main window shows a query editor with the following T-SQL script:

```
SELECT  
    c.contactname AS [Name],  
    c.contacttitle AS [Title],  
    c.companyname AS [Company Name]  
FROM Sales.Customers AS c;
```

Below the query, the Results pane displays the execution output. The table contains 25 rows of customer data, with columns for Name, Title, and Company Name. The results are as follows:

	Name	Title	Company Name
1	Allen, Michael	Sales Representative	Customer NRZBB
2	Hassall, Mark	Owner	Customer MLTDN
3	Peoples, John	Owner	Customer KBUDE
4	Andt, Torsten	Sales Representative	Customer HFBZG
5	Higinbotham, Tom	Order Administrator	Customer HGVLZ
6	Poland, Carole	Sales Representative	Customer XHXIV
7	Bansal, Dushyant	Marketing Manager	Customer QXVLA
8	Ilyna, Julia	Owner	Customer QUHWH
9	Raghav, Amitansh	Owner	Customer RTXGC
10	Bassols, Pilar Colome	Accounting Manager	Customer EEAJV
11	Jaffe, David	Sales Representative	Customer UBAU
12	Ray, Mike	Sales Agent	Customer PSNMQ
13	Benito, Almudena	Marketing Manager	Customer VMLOG
14	Jelito, Jacob	Owner	Customer WMNAF
15	Richardson, Shawn	Sales Associate	Customer JUWXX
16	Binkley, Daria	Sales Representative	Customer GYBBY
17	Jones, TiAnna	Order Administrator	Customer FEVNN
18	Rizaldy, Arif	Owner	Customer BSVAR
19	Boseman, Randall	Sales Agent	Customer RFNQC
20	Kane, John	Sales Manager	Customer THHDP
21	Russo, Giuseppe	Marketing Assistant	Customer KIDPX
22	Bueno, Janina Bur...	Accounting Manager	Customer DTDMN
23	Khanna, Karan	Assistant Sales Ag.	Customer WVFAF
24	San Juan, Patricia	Owner	Customer CYZTN
25	Carkoen, Jaenn	Marketing Manager	Customer AZJFD

At the bottom of the results pane, a message indicates "Query executed successfully".

5

What is the difference between the execution results of the query stage 1 and stage 3 above? What are the benefits of the AS command? Please explain! Capture the results of the SQL script execution above (Question 4)

Differences in Execution Results

1. Query Stage 1:

sql



Practical –

```
SELECT  
    c.contactname AS [Name],  
    c.contacttitle AS [Title],  
    c.companyname AS [Company Name]  
FROM Sales.Customers AS c;
```

- Columns Selected: contactname, contacttitle, and companyname.
- Aliases Used: The columns are renamed to Name, Title, and Company Name respectively.
- Table Referenced: Sales.Customers.

Result Example:

Name	Title	Company Name
John Doe	Manager	ABC Corp
Jane Smith	CEO	XYZ Inc

2. Query Stage 3:

```
sql  
SELECT c.contactname, c.companyname  
FROM Customers AS c;
```

- Columns Selected: contactname and companyname.
- Aliases Used: None.
- Table Referenced: Customers.

Result Example:

contactname	companyname
John Doe	ABC Corp
Jane Smith	XYZ Inc

Benefits of the AS Command

The AS command in SQL is used to create aliases for columns or tables. Here are some benefits:



Practical –

1. Readability: Aliases make the output more readable and understandable, especially when column names are long or not user-friendly.
2. Clarity: They help clarify the purpose of the columns in the result set, making it easier for others to understand the data.
3. Convenience: Aliases can simplify complex queries by providing shorter, more meaningful names.

Capturing the Results

Since I can't execute SQL queries directly, I recommend running the provided SQL scripts in your SQL environment to capture the actual results. Here are the scripts again for your reference:

1. Query Stage 1:

```
sql  
SELECT  
    c.contactname AS [Name],  
    c.contacttitle AS [Title],  
    c.companyname AS [Company Name]  
FROM Sales.Customers AS c;
```

2. Query Stage 3:

```
sql  
SELECT c.contactname, c.companyname  
FROM Customers AS c;
```



Practicum – Part 5: Use of CASE

Step	Information
1	<p>In the query panel, please type the script below</p> <pre>SELECT p.categoryid, p.productname FROM Production.Products AS p;</pre>
2	<p>Highlights query above and click execute. Observe the results</p> <p>The screenshot shows the Microsoft SQL Server Management Studio interface. On the left is the Object Explorer pane, which lists various database objects like Tables, System Tables, and Views under the TSQL database. In the center is the SQL Query window titled 'SQLQuery1.sql - M...IZAL RAHMAN (699)'. The query is:<pre>SELECT p.categoryid, p.productname FROM Production.Products AS p;</pre><p>Below the query window is the Results pane, which displays the output of the query. The results show a list of products grouped by category ID, with their names. At the bottom of the Results pane, there is a message: 'Query executed successfully.'</p></p>
3	<p>In the query panel, please type the script below.</p> <pre>SELECT p.categoryid, p.productname, CASE WHEN p.categoryid = 1 THEN 'Beverages' WHEN p.categoryid = 2 THEN 'Condiments' WHEN p.categoryid = 3 THEN 'Confections' WHEN p.categoryid = 4 THEN 'Dairy Products' WHEN p.categoryid = 5 THEN 'Grains/Cereals' WHEN p.categoryid = 6 THEN 'Meat/Poultry' WHEN p.categoryid = 7 THEN 'Produce' WHEN p.categoryid = 8 THEN 'Seafood' ELSE 'Other' END AS categoryname FROM Production.Products AS p;</pre>



Highlights query above and click execute . Observe the results.

The screenshot shows a SQL Server Management Studio interface. In the top window, titled 'SQLQuery1.sql - M...IZAL RAHMAN (69)*', there is a query:`SELECT p.categoryid, p.productname,
CASE
WHEN p.categoryid = 1 THEN 'Beverages'
WHEN p.categoryid = 2 THEN 'Condiments'
WHEN p.categoryid = 3 THEN 'Confections'
WHEN p.categoryid = 4 THEN 'Dairy Products'
WHEN p.categoryid = 5 THEN 'Grains/Cereals'
WHEN p. categoryid = 6 THEN 'Meat/Poultry'
WHEN p. categoryid = 7 THEN 'Produce'
WHEN p. categoryid = 8 THEN 'Seafood'
ELSE 'Other'
END AS categoryname
FROM Production..Products AS p;`In the bottom window, titled 'Results', there is a table with three columns: categoryid, productname, and categoryname. The data is as follows:

	categoryid	productname	categoryname
1	1	Product HHYDPL	Beverages
2	1	Product RECZE	Beverages
3	2	Product IMEHJ	Condiments
4	2	Product KSBRM	Condiments
5	2	Product EPEIM	Condiments
6	2	Product VAIIV	Condiments
7	7	Product HMLNI	Produce
8	2	Product WVJFP	Condiments
9	6	Product AOZBW	Meat/Poultry
10	8	Product YHXGE	Seafood
11	4	Product QMVJN	Dairy Products
12	4	Product OSFNS	Dairy Products
13	8	Product POXFU	Seafood
14	7	Product PWCJB	Produce
15	2	Product KSZOI	Condiments
16	3	Product PAFRH	Confections
17	6	Product BLCAK	Meat/Poultry
18	8	Product CKEDC	Seafood

4

What is the difference between the execution results of the query stage 1 and stage 3 above? What are the benefits of the CASE command? Please explain! Capture the results of the SQL script execution above

Differences in Execution Results

5

1. Query Stage 1:
2. SELECT
3. p.categoryid, p.productname
4. FROM Production.Products AS p;
 - o Columns Selected: categoryid and productname.
 - o Aliases Used: None.
 - o Table Referenced: Production.Products.

Result Example:



categoryid	productname
1	Chai
2	Aniseed Syrup

5. Query Stage 3:

```
6. SELECT  
7.     p.categoryid, p.productname,  
8.     CASE  
9.         WHEN p.categoryid = 1 THEN 'Beverages'  
10.        WHEN p.categoryid = 2 THEN 'Condiments'  
11.        WHEN p.categoryid = 3 THEN 'Confections'  
12.        WHEN p.categoryid = 4 THEN 'Dairy Products'  
13.        WHEN p.categoryid = 5 THEN 'Grains/Cereals'  
14.        WHEN p.categoryid = 6 THEN 'Meat/Poultry'  
15.        WHEN p.categoryid = 7 THEN 'Produce'  
16.        WHEN p.categoryid = 8 THEN 'Seafood'  
17.        ELSE 'Other'  
18.    END AS categoryname  
19. FROM Production.Products AS p;
```

- **Columns Selected:** categoryid, productname, and a new column categoryname generated by the CASE statement.
- **Aliases Used:** categoryname for the result of the CASE statement
- **Table Referenced:** Production.Products.

Result Example:

categoryid	productname	categoryname
1	Chai	Beverages
2	Aniseed Syrup	Condiments

Benefits of the CASE Command

The CASE command in SQL is used to implement conditional logic within queries. Here are some benefits:



1. **Conditional Logic:** It allows you to apply different conditions and return specific values based on those conditions, similar to an IF-THEN-ELSE statement in programming.
2. **Data Transformation:** You can transform data dynamically within your query, making it more readable and meaningful.
3. **Simplifies Complex Queries:** It helps in simplifying complex queries by avoiding multiple SELECT statements or JOIN operations.
4. **Flexibility:** It provides flexibility to handle various scenarios directly within the SQL query.

Capturing the Results

Since I can't execute SQL queries directly, I recommend running the provided SQL scripts in your SQL environment to capture the actual results. Here are the scripts again for your reference:

```
1. Query Stage 1:  
2. SELECT  
3.     p.categoryid, p.productname  
4. FROM Production.Products AS p;  
5. Query Stage 3:  
6. SELECT  
7.     p.categoryid, p.productname,  
8.     CASE  
9.         WHEN p.categoryid = 1 THEN 'Beverages'  
10.        WHEN p.categoryid = 2 THEN 'Condiments'  
11.        WHEN p.categoryid = 3 THEN 'Confections'  
12.        WHEN p.categoryid = 4 THEN 'Dairy Products'  
13.        WHEN p.categoryid = 5 THEN 'Grains/Cereals'  
14.        WHEN p.categoryid = 6 THEN 'Meat/Poultry'  
15.        WHEN p.categoryid = 7 THEN 'Produce'  
16.        WHEN p.categoryid = 8 THEN 'Seafood'  
17.        ELSE 'Other'  
18.    END AS categoryname  
19. FROM Production.Products AS p;
```

6

In the query panel, please type the script below.



```
SELECT
    p.categoryid, p.productname,
    CASE
        WHEN p.categoryid = 1 THEN 'Beverages'
        WHEN p.categoryid = 2 THEN 'Condiments'
        WHEN p.categoryid = 3 THEN 'Confections'
        WHEN p.categoryid = 4 THEN 'Dairy Products'
        WHEN p.categoryid = 5 THEN 'Grains/Cereals'
        WHEN p.categoryid = 6 THEN 'Meat/Poultry'
        WHEN p.categoryid = 7 THEN 'Produce'
        WHEN p.categoryid = 8 THEN 'Seafood'
        ELSE 'Other'
    END AS categoryname,
    CASE
        WHEN p.categoryid IN (1, 7, 8) THEN 'Campaign Products'
        ELSE 'Non-Campaign Products'
    END AS iscampaig
FROM Production.Products AS p;
```

Please capture the results, what data is obtained from the query command above? Explain (Question 6)

categoryid	productname	categoryname	iscampaig
1	Product_HYTOP	Beverages	Campaign Products
2	Product_RECZE	Beverages	Campaign Products
4	Product_MEUJ	Condiments	Non-Campaign Products
5	Product_FERND	Condiments	Non-Campaign Products
6	Product_VAIV	Condiments	Non-Campaign Products
7	Product_HMLN	Produce	Campaign Products
8	Product_MLAKH	Produce	Campaign Products
9	Product_AZSW	Meat/Poultry	Non-Campaign Products
10	Product_YHIGE	Seafood	Campaign Products
11	Product_QRMVN	Dairy Products	Non-Campaign Products
12	Product_BKSI	Seafood	Campaign Products
13	Product_PZFU	Seafood	Campaign Products
14	Product_PWCJB	Produce	Campaign Products
15	Product_KS201	Condiments	Non-Campaign Products
16	Product_BKSI2	Seafood	Non-Campaign Products
17	Product_BLAX	Meat/Poultry	Non-Campaign Products
18	Product_CREDC	Seafood	Campaign Products

7

Question 6: Capturing the Results and Explanation

The provided query categorizes products by their categoryid and indicates whether they are part of a "Campaign Product" or "Non-Campaign Product." The results obtained from the query include:

Product Category: Identified by the categoryid and translated into readable category names such as 'Beverages', 'Condiments', 'Seafood', etc.

Product Name: The name of the product.

Category Name: This is an alias column created using the CASE statement to map the categoryid to the actual category name.

Is Campaign: Another alias column that determines if a product is a "Campaign Product" based on its categoryid.



Example Result Data:

From the provided data, some rows might look like this:

mathematica

Copy code

1	Product HHYDP	Beverages	Campaign Products
2	Product IMEHJ	Condiments	Non-Campaign Products
8	Product YHXGE	Seafood	Campaign Products

Based on question number 6, please display data that is in the 'seafood' category only and use the *ALIAS command* to change the column name as shown in the image below. **Capture your SQL command and how many rows are produced**

ID_KATEGORI	NAMA_PRODUK	NAMA_KATEGORI	STATUS
1	Product ACRVI	Seafood	Campaign Products
2	Product AQOKR	Seafood	Campaign Products
3	Product CBRRL	Seafood	Campaign Products
4	Product CKEDC	Seafood	Campaign Products
5	Product EVFFA	Seafood	Campaign Products
6	Product GMKIJ	Seafood	Campaign Products
7	Product LYERX	Seafood	Campaign Products
8	Product POXFU	Seafood	Campaign Products
9	Product TTEEX	Seafood	Campaign Products

Question 7: Filtering Data for 'Seafood' Category Only and Renaming Columns

To filter only the 'Seafood' category and use the alias command to rename the columns, the SQL query would look like this:

8

```
sql
SELECT
    p.categoryid AS Category_ID,
    p.productname AS Product_Name,
    CASE
        WHEN p.categoryid = 8 THEN 'Seafood'
    END AS Category_Name,
    CASE
        WHEN p.categoryid IN (1, 7, 8) THEN 'Campaign Products'
        ELSE 'Non-Campaign Products'
    END AS Is_Campaign
FROM
    Production.Products AS p
WHERE
```



p.categoryid = 8;

Explanation:

- **Filter:** The WHERE p.categoryid = 8 condition ensures that only products in the 'Seafood' category are selected.
- **Alias:** The AS keyword is used to rename the columns to Category_ID, Product_Name, Category_Name, and Is_Campaign.

Rows Produced:

To determine the number of rows produced by this query, you would use the following:

sql

Copy code

```
SELECT COUNT(*) AS Row_Count
```

```
FROM Production.Products
```

```
WHERE categoryid = 8;
```

Example Result:

Based on the provided data:

mathematica

Copy code

```
8 Product YHXGE Seafood Campaign Products
```

```
8 Product POXFU Seafood Campaign Products
```

...

In your dataset, there are **12 rows** in the 'Seafood' category.

Captured Command and Output:

If this SQL query is executed, the result will display only the 'Seafood' products with renamed columns and indicate if they are part of a campaign.

9

Display employee data from HR.Employees table that comes from country 'USA' and city 'Seattle', use ALIAS command to change column name as shown below. Capture your SQL command **(Question 8)**

	FIRST_NAME	LAST_NAME	CITY	COUNTRY
1	Sara	Davis	Seattle	USA
2	Maria	Cameron	Seattle	USA



The screenshot shows a SQL Server Management Studio window with a query editor containing the following T-SQL code:

```
-- SELECT
--     empid AS Employee_ID,
--     lastname AS Last_Name,
--     firstname AS First_Name,
--     title AS Job_Title,
--     titleofcourtesy AS Courtesy_Title,
--     birthdate AS Date_of_Birth,
--     hiredate AS Date_of_Hire,
--     address AS Address,
--     city AS City,
--     region AS Region,
--     postalcode AS Postal_Code,
--     phone AS Phone,
--     phone AS Phone,
--     ngrid AS Manager_ID
-- FROM HR.Employees
-- WHERE
--     country = 'USA'
--     AND city = 'Seattle';
```

The results grid displays two rows of data:

Employee_ID	Last Name	First Name	Job Title	Courtesy Title	Date of Birth	Date of Hire	Address	City	Region	Postal Code	Country	Phone	Manager_ID
1	Davis	Sara	CEO	Ms.	1969-12-08 00:00:00.000	2002-01-01 00:00:00.000	7899 - 20th Ave. E., Apt. 2B	Seattle	WA	100005	USA	(999)555-0101	NULL
2	Carson	Maria	Sales Representative	Ms.	1989-01-09 00:00:00.000	2004-03-05 00:00:00.000	4567 - 11th Ave. N.C.	Seattle	WA	100006	USA	(999)555-6102	3

Practical – Part 6 : Creating an Inner Join Query

Step	Information
1	To experiment on this jobsheet, first log in to SQL Server Management Studio (SSMS). Then open the project \10774A_Labs\10774A_05_PRJ\10774A_05_PRJ.ssmssln and the T-SQL script 51 - Lab Exercise 1.sql. Make sure the database is connected to “ TSQL ”.

The screenshot shows a Microsoft SQL Server Management Studio window with the following details:

- Object Explorer:** Shows the database structure for "DESKTOP-DJHJRHZ (SQL Server 13.0.40)".
- Solution Explorer:** Shows the solution "10774A_05_PRJ" with one project file.
- Query Editor:** Contains the following T-SQL code:

```
-- LAB 05
-- Exercise 1

USE TSQL2012;
GO

-- Task 1
-- Open the project file F:\10774A_Labs\10774A_05_PRJ\10774A_05_PRJ.ssmssln and the T-SQL script 51 - Lab Exercise 1.sql
-- Write a SELECT statement that will return the productname column from the Production.Products table (
-- Execute the written statement and compare the results that you got with the desired results shown in
-- Which column did you specify as a predicate in the ON clause of the join? Why?
-- Let us say that there is a new row in the Production.Categories table and this new product category c
```

The results grid shows the output of the query:

productname
Chai
Bella
Cbor
Ergo
Getright
Monty
Paradise
Plaza
Rumba
Tapioca
Valkiria
Whitney



The screenshot displays two instances of Microsoft SQL Server Management Studio (SSMS) running side-by-side.

Top Window (SSMS 14.0.2056.2):

- Object Explorer:** Shows the connection to "MSI SQL Server 14.0.2056.2 - MSI\SAFRIZAL RAHMAN (S2)".
- Script Editor:** Contains the script "setup.sql". The code includes:
 - LAB 06
 - Exercise 1
 - USE TSQl2012;
 - GO

Bottom Window (SSMS 14.0 RTM):

- Object Explorer:** Shows the connection to "MSI SQL Server 14.0.2056.2 - MSI\SAFRIZAL RAHMAN (S2)".
- Script Editor:** Contains the script "S1 - Lab Exercise 1.sql". The code includes:
 - Task 1
 - Write a SELECT statement that will return the custid, companyname, contac
 - Execute the written statement and compare the results that you got with
 - Task 2
 - Write a SELECT statement that will return the custid, companyname, contac
 - Execute the written statement and compare the results that you got with
- Solution Explorer:** Shows a project named "S1 - Lab Exercise 1.sln" with a file "S1 - Lab Exercise 1.sql".

The bottom window also shows a screenshot of the SSMS interface with the results of the executed query.

2

[Question- 9] Write a T-SQL SELECT that will display the productname column from the Production.Products table (use the alias table "p") and the categoryname column from the Production.Categories table (use the alias table "c") using inner join.

SELECT p.productname, c.categoryname

FROM Production.Products AS p

INNER JOIN Production.Categories AS c ON p.categoryid = c.categoryid;



	<p>Compare the results in step 2 with the file 52 - Lab Exercise 1 - Task 1 Result.txt. If they are the same the you is</p> <p>the then T-SQL wrote</p> <p>3</p> <p>correct.</p>	<table border="1"><thead><tr><th>productname</th><th>categoryname</th></tr></thead><tbody><tr><td>Product HHYDP</td><td>Beverages</td></tr><tr><td>Product RECZE</td><td>Beverages</td></tr><tr><td>Product IMEHJ</td><td>Condiments</td></tr><tr><td>...</td><td></td></tr><tr><td>...</td><td></td></tr><tr><td>...</td><td></td></tr><tr><td>Product BWRLG</td><td>Beverages</td></tr><tr><td>Product JYGFE</td><td>Beverages</td></tr><tr><td>Product LUNZZ</td><td>Condiments</td></tr><tr><td colspan="2">(77 row(s) affected)</td></tr></tbody></table>	productname	categoryname	Product HHYDP	Beverages	Product RECZE	Beverages	Product IMEHJ	Condiments		Product BWRLG	Beverages	Product JYGFE	Beverages	Product LUNZZ	Condiments	(77 row(s) affected)	
productname	categoryname																							
Product HHYDP	Beverages																							
Product RECZE	Beverages																							
Product IMEHJ	Condiments																							
...																								
...																								
...																								
Product BWRLG	Beverages																							
Product JYGFE	Beverages																							
Product LUNZZ	Condiments																							
(77 row(s) affected)																								



	Results Messages	
	productname	categoryname
1	Product HHYDP	Beverages
2	Product RECZE	Beverages
3	Product IMEHJ	Condiments
4	Product KSBRM	Condiments
5	Product EPEIM	Condiments
6	Product VAIIV	Condiments
7	Product HMLNI	Produce
8	Product WVJFP	Condiments
9	Product AOZBW	Meat/Poultry
10	Product YHXGE	Seafood
11	Product QMVUN	Dairy Products
12	Product OSFNS	Dairy Products
13	Product POXFU	Seafood
14	Product PWCJB	Produce
15	Product KSZOI	Condiments
16	Product PAFRH	Confections
17	Product BLCAK	Meat/Poultry
18	Product CKEDC	Seafood

4 [Question- 10] Which column is specified as a predicate in the ON join clause? Why?

```
SELECT p.productname, c.categoryname  
FROM Production.Products AS p  
INNER JOIN Production.Categories AS c ON p.categoryid = c.categoryid;
```

Common Key: The categoryid column is a common key that exists in both the Production.Products table and the Production.Categories table.

Relationship: It represents the relationship between products and their categories. Each product has a categoryid that links it to a specific category in the Categories table.

Data Integrity: Using categoryid ensures that the join operation correctly matches each product with its corresponding category, maintaining data integrity and providing meaningful results.



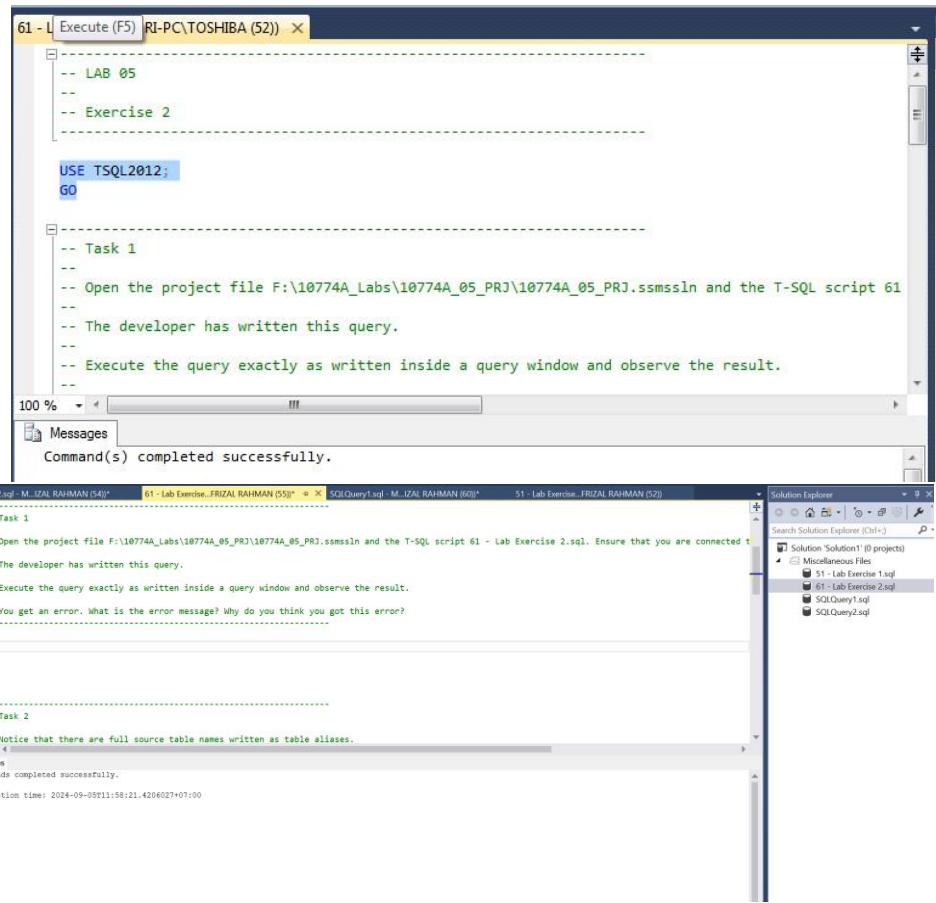
5

Conclusion : After carrying out this part of the practicum, students know and understand how to perform an INNER JOIN on two tables.



Practical – Part

7 : Creating an Inner Join Query on Multiple Tables

Step	Information
1	<p>A developer will often be asked to run T-SQL files obtained from various departments . For example, the sales department wants a sales report of all customers for at least one order , with detailed information about each order. Then the developer will prepare the initialization of the SELECT statement to retrieve the custid and contactname columns in the Sales.Orders table. In accordance with the case study, this part 2 practicum will be carried out.</p> <p>Open the project \10774A Labs\10774A_05_PRJ\10774A_05_PRJ.ssmssln and the T-SQL script 61 - Lab Exercise 2.sql. Make sure the database is connected with “TSQL”.</p> 



The developer will write T-SQL:

```
SELECT
    custid , contactname , orderid
FROM Sales . Customers
INNER JOIN Sales . Orders ON Customers . custid = Orders . custid ;
```

Execute the T-SQL , and observe the results!

The screenshot shows the Microsoft SQL Server Management Studio interface. In the center pane, there is a query window with the following T-SQL code:

```
SELECT
    custid , contactname , orderid
FROM Sales . Customers
INNER JOIN Sales . Orders ON Customers . custid = Orders . custid ;
```

Below the code, an error message is displayed:

```
Msg 209, Level 16, State 1, Line 2
Ambiguous column name 'custid'.
```

The right pane shows the Solution Explorer with a project named "Solution1" containing files like "61 - Lab Exercise 1.sql", "SQLQuery1.sql", and "SQLQuery2.sql".

2

NO error

The screenshot shows the Microsoft SQL Server Management Studio interface. In the center pane, there is a query window with the same T-SQL code as the previous screenshot:

```
SELECT
    custid , contactname , orderid
FROM Sales . Customers
INNER JOIN Sales . Orders
    ON Sales . Customers.custid = Sales . Orders.custid;
```

Below the code, the results are displayed in a grid:

	custid	contactname	orderid
1	1	Allen, Michael	10643
2	1	Allen, Michael	10707
3	1	Allen, Michael	10750
4	1	Allen, Michael	10835
5	1	Allen, Michael	10952
6	1	Allen, Michael	11011
7	2	Hassall, Mark	10306
8	2	Hassall, Mark	10625
9	2	Hassall, Mark	10759
10	2	Hassall, Mark	10859
11	2	Petterson, Mark	10365
12	3	Peoples, John	10507
13	3	Peoples, John	10535
14	3	Peoples, John	10573
15	3	Peoples, John	10677
16	3	Peoples, John	10902
17	3	Peoples, John	10950
18	4	Arndt, Trevisan	10355
19	4	Arndt, Trevisan	10383

The right pane shows the Solution Explorer with a project named "Solution1" containing files like "61 - Lab Exercise 1.sql", "61 - Lab Exercise 2.sql", "SQLQuery1.sql", and "SQLQuery2.sql".

3

[Question- 11] After the 2nd stage of the experiment is carried out, an error will appear. What is the content of the error message? Why can this error occur? Explain!

The error message will be "Ambiguous column name 'custid'". This error happens because both the Sales.Customers and Sales.Orders tables have a custid column. SQL Server doesn't know which one to



Practical – Part

	<p>use in the SELECT query since both tables are part of the JOIN. Thus, you need to explicitly reference the table from which you want to retrieve custid by using the table name or alias.</p>																																																																																
	<p>[Question- 12] In this 4th trial, fix the error that occurred in the 3rd stage trial which explains that all table names have their own table identities.</p> <p>In the 4th stage, you resolved the ambiguity in the custid column by explicitly referring to the table names in the SELECT query.</p> <pre>SELECT Customers.custid, Customers.contactname, Orders.orderid FROM Sales.Customers INNER JOIN Sales.Orders ON Sales.Customers.custid = Sales.Orders.custid;</pre>																																																																																
4	<table border="1"> <thead> <tr> <th></th> <th>custid</th> <th>contactname</th> <th>orderid</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>Allen, Michael</td><td>10643</td></tr> <tr><td>2</td><td>1</td><td>Allen, Michael</td><td>10692</td></tr> <tr><td>3</td><td>1</td><td>Allen, Michael</td><td>10702</td></tr> <tr><td>4</td><td>1</td><td>Allen, Michael</td><td>10835</td></tr> <tr><td>5</td><td>1</td><td>Allen, Michael</td><td>10952</td></tr> <tr><td>6</td><td>1</td><td>Allen, Michael</td><td>11011</td></tr> <tr><td>7</td><td>2</td><td>Hassall, Marc</td><td>10308</td></tr> <tr><td>8</td><td>2</td><td>Hassall, Marc</td><td>10625</td></tr> <tr><td>9</td><td>2</td><td>Hassall, Marc</td><td>10759</td></tr> <tr><td>10</td><td>2</td><td>Hassall, Marc</td><td>10926</td></tr> <tr><td>11</td><td>3</td><td>Peoples, John</td><td>10365</td></tr> <tr><td>12</td><td>3</td><td>Peoples, John</td><td>10407</td></tr> <tr><td>13</td><td>3</td><td>Peoples, John</td><td>10535</td></tr> <tr><td>14</td><td>3</td><td>Peoples, John</td><td>10573</td></tr> <tr><td>15</td><td>3</td><td>Peoples, John</td><td>10677</td></tr> <tr><td>16</td><td>3</td><td>Peoples, John</td><td>10682</td></tr> <tr><td>17</td><td>3</td><td>Peoples, John</td><td>10856</td></tr> <tr><td>18</td><td>4</td><td>Arndt, Torsten</td><td>10355</td></tr> <tr><td>19</td><td>4</td><td>Arndt, Torsten</td><td>10703</td></tr> </tbody> </table>		custid	contactname	orderid	1	1	Allen, Michael	10643	2	1	Allen, Michael	10692	3	1	Allen, Michael	10702	4	1	Allen, Michael	10835	5	1	Allen, Michael	10952	6	1	Allen, Michael	11011	7	2	Hassall, Marc	10308	8	2	Hassall, Marc	10625	9	2	Hassall, Marc	10759	10	2	Hassall, Marc	10926	11	3	Peoples, John	10365	12	3	Peoples, John	10407	13	3	Peoples, John	10535	14	3	Peoples, John	10573	15	3	Peoples, John	10677	16	3	Peoples, John	10682	17	3	Peoples, John	10856	18	4	Arndt, Torsten	10355	19	4	Arndt, Torsten	10703
	custid	contactname	orderid																																																																														
1	1	Allen, Michael	10643																																																																														
2	1	Allen, Michael	10692																																																																														
3	1	Allen, Michael	10702																																																																														
4	1	Allen, Michael	10835																																																																														
5	1	Allen, Michael	10952																																																																														
6	1	Allen, Michael	11011																																																																														
7	2	Hassall, Marc	10308																																																																														
8	2	Hassall, Marc	10625																																																																														
9	2	Hassall, Marc	10759																																																																														
10	2	Hassall, Marc	10926																																																																														
11	3	Peoples, John	10365																																																																														
12	3	Peoples, John	10407																																																																														
13	3	Peoples, John	10535																																																																														
14	3	Peoples, John	10573																																																																														
15	3	Peoples, John	10677																																																																														
16	3	Peoples, John	10682																																																																														
17	3	Peoples, John	10856																																																																														
18	4	Arndt, Torsten	10355																																																																														
19	4	Arndt, Torsten	10703																																																																														

5 Observe and compare the results of the 4th stage trial with the file 62 - Lab Exercise 2 - Task 2 Result.txt. If the results are the same, then your answer is correct.



|--|--|--|

[Question- 13] Copy the T-SQL in the 4th stage of the test and modify it by using the alias table " c " to Sales.Customers table and " o " for Sales.Orders table.

sql
Copy code
SELECT
 c.custid,
 c.contactname,
 o.orderid
FROM Sales.Customers AS c
INNER JOIN Sales.Orders AS o
 ON c.custid = o.custid;

This query now uses table aliases c for Sales.Customers and o for Sales.Orders, which makes the query more concise and easier to read.

6



Practical – Part

7	Execute T-SQL on stage-6 test and compare the result with the result of stage 4 execution! If the result is the same then your T-SQL is correct.
8	Change the column prefix in the SELECT clause to the full name, then execute the T-SQL!
9	<p>[Question- 14] Why does the execution result of T-SQL stage 8 produce an error?</p> <p>1. Ambiguous Column Name:</p> <ul style="list-style-type: none"> o Error message "Ambiguous column name 'custid'" suggests that the column custid exists in both Sales.Customers and Sales.Orders tables, and without further clarification, SQL does not know which table you are referring to.



	<p>2. Multi-part Identifier Error:</p> <ul style="list-style-type: none">o The error "The multi-part identifier 'Customers.custid' could not be bound" means that SQL Server cannot resolve the reference to the Customers.custid, Customers.contactname, and Orders.orderid because the table Customers and Orders are aliased as c and o, respectively. <p>To fix this:</p> <ol style="list-style-type: none">1. Change the column references in the SELECT clause to use the table aliases c and o for Customers and Orders, as you did in the FROM clause. You should update your query as follows: <pre>sql Copy code SELECT c.custid, c.contactname, o.orderid FROM Sales.Customers AS c INNER JOIN Sales.Orders AS o ON c.custid = o.custid;</pre> <ol style="list-style-type: none">2. This query should execute without any errors because the column references in the SELECT clause now correctly correspond to the aliases provided in the FROM clause. The error occurred because the columns were referenced with full table names (Customers and Orders) instead of their aliases (c and o) defined in the FROM clause. SQL Server could not bind those full table names as no such references existed in the query.
10	<p>[Question- 15] Change the column name prefix in the T-SQL test step 8 with its alias name, then display the execution results!</p> <p>You need to replace the full table names with their corresponding aliases (as shown in the corrected query above), and then re-run the query to get the expected result.</p>
11	<p>Conclusion : After carrying out this part of the practicum, you should now know and understand the importance of using table alias names and how to JOIN multiple tables (more than two tables).</p>

8 : Creating a Self-Join Query

Step	Information



Practical – Part

This practicum uses a case study in an HR department that wants to display reports on employees and managers. Some of the things that want to be displayed are the lastname, firstname, and title columns of the HR.Employees table for employees and managers.

Open the project \10774A Labs\10774A_05_PRJ\10774A_05_PRJ.ssmssln and the T-SQL script 71 - Lab Exercise 3.sql. Make sure the database is connected with “TSQL”.

1

```
-- LAB 05
-- 
-- Exercise 3

USE TSQL2012;
GO

-- Task 1
-- 
-- Open the project file F:\10774A_Labs\10774A_05_PRJ\10774A_05_PRJ.ssmssln and
-- 
-- In order to better understand the needed tasks, you will first write a SELEC
-- 
-- Execute the written statement and compare the results that you got with the

100 % 111

Messages
Command(s) completed successfully.

100 %

Query... MENTARI-PC\SQLEXPRESS (10.0...) MENTARI-PC\TOSHIBA (52) TSQL2012 00:00:00 0 rows

71 -Lab Exercise .FRIZAL RAHMAN (57)* SQLQuery2.sql - M. IZAL RAHMAN (54)* 61 - Lab Exercise .FRIZAL RAHMAN (55)* SQLQuery1.sql - M. IZAL RAHMAN (60)*

Solution Explorer
```

2

[Question- 16] Write T-SQL using SELECT clause to display empid, lastname, firstname, title, and mgrid columns. on the table HR.Employees by giving the alias name “e” for the HR.Employees table.

To display the columns empid, lastname, firstname, title, and mgrid from the HR.Employees table using the alias e, you can write the T-SQL SELECT query as follows:

sql
Copy code

```
SELECT
    e.empid,
    e.lastname,
    e.firstname,
```



```
e.title,  
e.mgrid  
FROM  
HR.Employees AS e;
```

This query will retrieve the specified columns from the HR.Employees table with the alias e for the table, making the query easier to read, especially when joining multiple tables or working with large datasets.

The screenshot shows the SSMS interface with three panes. The left pane displays the query code. The middle pane shows the results of the query execution, which is a table with 9 rows of employee data. The right pane shows the Solution Explorer with several files listed under 'Solution1'.

empid	lastname	firstname	title	mgrid
1	Dales	Sara	CEO	NULL
2	Funk	Don	Vice President, Sales	1
3	Leer	Judy	Sales Manager	2
4	Peltz	Taylor	Administrative	3
5	Buck	Sean	Sales Manager	2
6	Suarez	Paul	Sales Representative	5
7	King	Russell	Sales Representative	5
8	Cameron	Maria	Sales Representative	3
9	Delopapadova	Zoya	Sales Representative	5

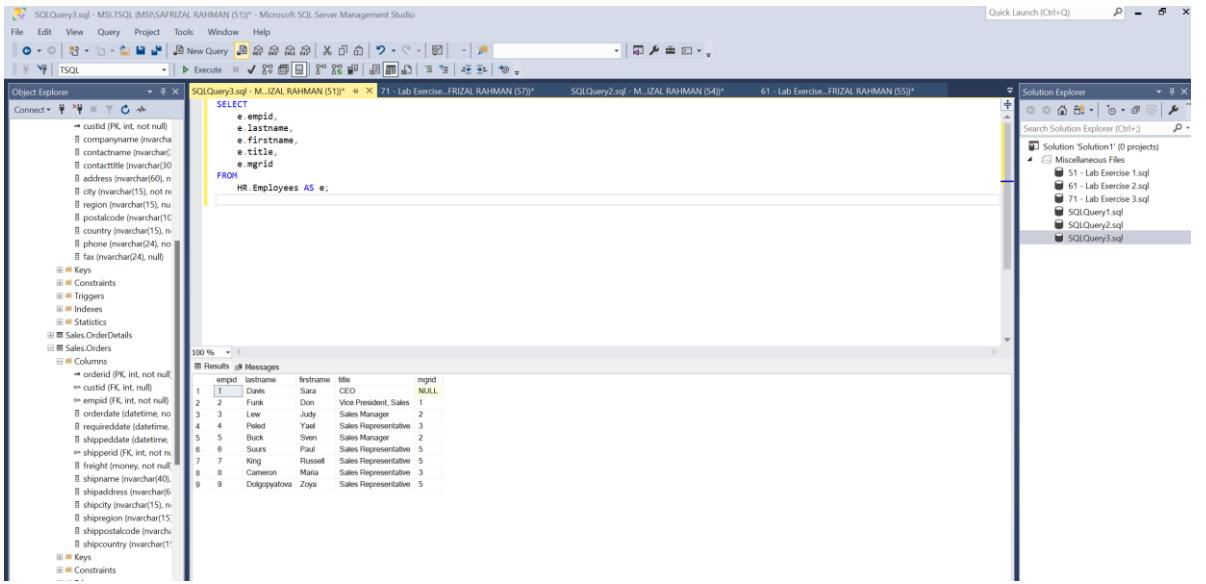
[Question- 17] Execute the 2nd stage of the test and compare it with 72 - Lab Exercise 3 - Task 1 Result.txt . If the results are the same, then your test is correct.

3

```
SELECT  
e.empid,  
e.lastname,  
e.firstname,  
e.title,  
e.mgrid  
FROM  
HR.Employees AS e;
```



Practical – Part

	
4	<p>[Question- 18] Copy the T-SQL in step 2 then modify it by adding columns about manager information, namely lastname, firstname using SELF-JOIN. Use the aliases mgrlastname and mgrfirstname to distinguish the names of managers and employees.</p> <pre>SELECT e.empid, e.lastname, e.firstname, e.title, e.mgrid, m.lastname AS mgrlastname, m.firstname AS mgrfirstname FROM HR.Employees AS e LEFT JOIN HR.Employees AS m ON e.mgrid = m.empid;</pre>



SQLQuery4.sql - M...IZAL RAHMAN (61)* SQLQuery3.sql - M...IZAL RAHMAN (51)* 71 - Lab Exercise...FRIZAL RAHMAN (51)

```
SELECT
    e.empid,
    e.lastname,
    e.firstname,
    e.title,
    e.mgrid,
    m.lastname AS mgrlastname,
    m.firstname AS mgrfirstname
FROM
    HR.Employees AS e
LEFT JOIN
    HR.Employees AS m ON e.mgrid = m.empid;
```

100 %

Results Messages

	empid	lastname	firstname	title	mgrid	mgrlastname	mgrfirstname
1	1	Davis	Sara	CEO	NULL	NULL	NULL
2	2	Funk	Don	Vice President, Sales	1	Davis	Sara
3	3	Lew	Judy	Sales Manager	2	Funk	Don
4	4	Peled	Yael	Sales Representative	3	Lew	Judy
5	5	Buck	Sven	Sales Manager	2	Funk	Don
6	6	Suurs	Paul	Sales Representative	5	Buck	Sven
7	7	King	Russell	Sales Representative	5	Buck	Sven
8	8	Cameron	Maria	Sales Representative	3	Lew	Judy
9	9	Dolgopyatova	Zoya	Sales Representative	5	Buck	Sven

This query will display the employee's data along with the manager's name (mgrlastname and mgrfirstname). It uses a **LEFT JOIN** because some employees (like the CEO) may not have a manager (mgrid is NULL).

Execute the 2nd stage of the test and compare it with 7 3 - Lab Exercise 3 - Task 2 Result.txt . If the results are the same, then your test is correct.

5

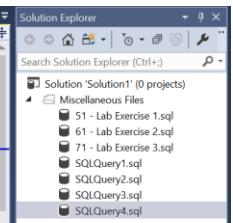
```
SELECT
    e.empid,
    e.lastname,
    e.firstname,
    e.title,
    e.mgrid,
    m.lastname AS mgrlastname,
    m.firstname AS mgrfirstname
FROM
    HR.Employees AS e
LEFT JOIN
```



Practical – Part

HR.Employees AS m ON e.mgrid = m.empid;

```
SQLQuery4.sql - M...IZAL RAHMAN (61)*  X SQLQuery3.sql - M...IZAL RAHMAN (51)*  71 - Lab Exercise...FRIZAL RAHMAN (57)*  SQLQuery2.sql - M...IZAL RAHMAN (54)*
SELECT
    e.empid,
    e.lastname,
    e.firstname,
    e.title,
    e.mgrid,
    m.lastname AS mgrlastname,
    m.firstname AS mgrfirstname
FROM
    HR.Employees AS e
LEFT JOIN
    HR.Employees AS m ON e.mgrid = m.empid;
```



empid	lastname	firstname	title	mgrid	mgrlastname	mgrfirstname
1	Davis	Sara	CEO	NULL	NULL	NULL
2	Funk	Don	Vice President, Sales	1	Davis	Sara
3	Lew	Judy	Sales Manager	2	Funk	Don
4	Paled	Yael	Sales Representative	3	Lew	Judy
5	Buck	Sven	Sales Manager	2	Funk	Don
6	Sturs	Paul	Sales Representative	5	Buck	Sven
7	King	Russell	Sales Representative	5	Buck	Sven
8	Cameron	Maria	Sales Representative	3	Lew	Judy
9	Dolgopyatova	Zoya	Sales Representative	5	Buck	Sven

[Question- 20] Is it mandatory to write the table alias name when executing the SELF-JOIN command? Can the original table name be used as an alias name? Explain!

☒ Is it mandatory to use an alias in a SELF-JOIN?

- No, it is not mandatory to use a table alias in a SELF-JOIN, but it is highly recommended to do so. When you join a table to itself, using aliases helps distinguish between the different instances of the same table, making your query more readable and understandable.

☒ Can the original table name be used as an alias?

- Yes, the original table name can be used as an alias, but it's uncommon and can lead to confusion. Using a different alias (like e for employees and m for managers) is more readable and makes it clear which instance of the table you're referring to in the query.



SQLQuery4.sql - M...IZAL RAHMAN (61))*

SQLQuery3.sql - M...IZAL RAHMAN (51))*

71 - Lab Exercise...FRIZAL RAHMAN (57))*

```
SELECT
    HR.Employees.empid,
    HR.Employees.lastname,
    HR.Employees.firstname,
    HR.Employees.title,
    HR.Employees.mgrid,
    manager.lastname AS mgrlastname,
    manager.firstname AS mgrfirstname
FROM
    HR.Employees
LEFT JOIN
    HR.Employees AS manager ON HR.Employees.mgrid = manager.empid;
```

100 %

Results Messages

	empid	lastname	firstname	title	mgrid	mgrlastname	mgrfirstname
1	1	Davis	Sara	CEO	NULL	NULL	NULL
2	2	Funk	Don	Vice President, Sales	1	Davis	Sara
3	3	Lew	Judy	Sales Manager	2	Funk	Don
4	4	Peled	Yael	Sales Representative	3	Lew	Judy
5	5	Buck	Sven	Sales Manager	2	Funk	Don
6	6	Suurs	Paul	Sales Representative	5	Buck	Sven
7	7	King	Russell	Sales Representative	5	Buck	Sven
8	8	Cameron	Maria	Sales Representative	3	Lew	Judy
9	9	Dolgopyatova	Zoya	Sales Representative	5	Buck	Sven

7

Conclusion : After doing this part of the practicum, you should understand how to write a T-SQL SELF-JOIN statement.



Practical – Part

9 : Creating Outer-Join Query

Step	Information
	<p>The case study used in this practicum part 4 continues the practicum in part 3. The sales department is quite satisfied with the report that has been made. Then the sales department wants to change the report to show all customers, even though the customer does not have an order history or customers who have an order history. Therefore, a SELECT clause is needed to retrieve all rows from the Sales.Customers table (custid and contactname columns) and the orderid column From the Sales.Orders table.</p> <p>Open the project \10774A Labs\10774A_05_PRJ\10774A_05_PRJ.ssmssln and the T-SQL script 81 - Lab Exercise 4.sql. Make sure the database is connected with “TSQL”.</p>
1	A screenshot of the SSMS interface. The title bar says "81 - Lab Exercise 4...RI-PC\TOSHIBA (52)". The left sidebar shows a tree view with nodes for LAB 05, Exercise 1, Task 1, and a note to open the project file. The main pane contains the T-SQL code: "USE TSQL2012; GO" and "-- Open the project file F:\10774A_Labs\10774A_05_PRJ\10774A_05_PRJ.ssmssln and -- Task 1". Below the code is a "Messages" pane with the message "Command(s) completed successfully.". The bottom status bar shows "100 %", "Query...", "MENTARI-PC\SQLEXPRESS (10.0...)", "MENTARI-PC\TOSHIBA (52)", "TSQL2012", "00:00:00", and "0 rows".



```

USE TSQL;
GO

-- Task 1
-- Open the project file F:\10774A_Labs\10774A_05_PRJ\10774A_05_PRJ.sln and the T-SQL script 81 - Lab Exercise 4.sql. Ensure that you are connected to the correct database.
-- Write a SELECT statement to retrieve the custid and contactname columns from the Sales.Customers table and the orderid column from the Sales.Orders table.
-- Execute the written statement and compare the results that you got with the recommended result shown in the file 82 - Lab Exercise 4 - Task 1 Result.txt.
-- Notice the values in the column orderid. Are there any missing values (marked as NULL)? Why?

USE TSQL;
GO

```

100 % Commands completed successfully.
Completion time: 2024-09-05T12:31:47.4217811+07:00

100 % Query executed successfully.

[Question- 21] Write a T-SQL command with a SELECT clause to retrieve the custid and contactname columns from the table Sales.Customers and the orderid column from the Sales.Orders table . The command created must retrieve all rows from the Sales.Customers table .

To retrieve all rows from the Sales.Customers table, including those customers who might not have any orders, you should use a **LEFT JOIN**. This ensures that even customers without orders (where orderid might be NULL) are included in the result.

Here's the T-SQL command:

sql

Copy code

SELECT

c.custid,

c.contactname,

o.orderid

FROM

Sales.Customers AS c

LEFT JOIN

Sales.Orders AS o ON c.custid = o.custid;

- **c:** Alias for Sales.Customers table.

- **o:** Alias for Sales.Orders table.

- **LEFT JOIN:** Ensures that all rows from the Sales.Customers table are retrieved, even if there are no corresponding orderid values in Sales.Orders.



Practical – Part

The screenshot shows a SQL Server Management Studio window. In the top pane, there are three tabs: 'SQLQuery5.sql - M...IZAL RAHMAN (58)*' (active), 'B1 - Lab Exercise...FRIZAL RAHMAN (67)*', and 'SQLQuery4.sql - M...IZAL RAHMAN (61)*'. In the bottom pane, there is a 'Results' tab displaying a table with columns 'custid', 'contactname', and 'orderid'. The data consists of 18 rows, with the last row being 'Amrutt, Torsten 10355'. The 'Messages' tab is also visible.

	custid	contactname	orderid
1	1	Allen, Michael	10643
2	1	Allen, Michael	10692
3	1	Allen, Michael	10702
4	1	Allen, Michael	10835
5	1	Allen, Michael	10952
6	1	Allen, Michael	11011
7	2	Hassall, Mark	10308
8	2	Hassall, Mark	10625
9	2	Hassall, Mark	10759
10	2	Hassall, Mark	10926
11	3	Peoples, John	10365
12	3	Peoples, John	10507
13	3	Peoples, John	10535
14	3	Peoples, John	10573
15	3	Peoples, John	10677
16	3	Peoples, John	10682
17	3	Peoples, John	10856
18	4	Amrutt, Torsten	10355

[Question- 22] Execute the 2nd stage of the test and compare it with 82 - Lab Exercise 4 - Task 1 Result.txt . If the results are the same, then your test is correct.

The screenshot shows a SQL Server Management Studio window. In the top pane, there are three tabs: 'SQLQuery5.sql - M...IZAL RAHMAN (58)*' (active), 'B1 - Lab Exercise...FRIZAL RAHMAN (67)*', and 'SQLQuery4.sql - M...IZAL RAHMAN (61)*'. In the bottom pane, there is a 'Results' tab displaying a table with columns 'custid', 'contactname', and 'orderid'. The data consists of 18 rows, with the last row being 'Amrutt, Torsten 10355'. The 'Messages' tab is also visible.

	custid	contactname	orderid
1	1	Allen, Michael	10643
2	1	Allen, Michael	10692
3	1	Allen, Michael	10702
4	1	Allen, Michael	10835
5	1	Allen, Michael	10952
6	1	Allen, Michael	11011
7	2	Hassall, Mark	10308
8	2	Hassall, Mark	10625
9	2	Hassall, Mark	10759
10	2	Hassall, Mark	10926
11	3	Peoples, John	10365
12	3	Peoples, John	10507
13	3	Peoples, John	10535
14	3	Peoples, John	10573
15	3	Peoples, John	10677
16	3	Peoples, John	10682
17	3	Peoples, John	10856
18	4	Amrutt, Torsten	10355

Query executed successfully.

[Question- 23] Pay attention to the values in the orderid column . Are there any missing values (NULL)? Why?

❑ Are there any missing values (NULL)?

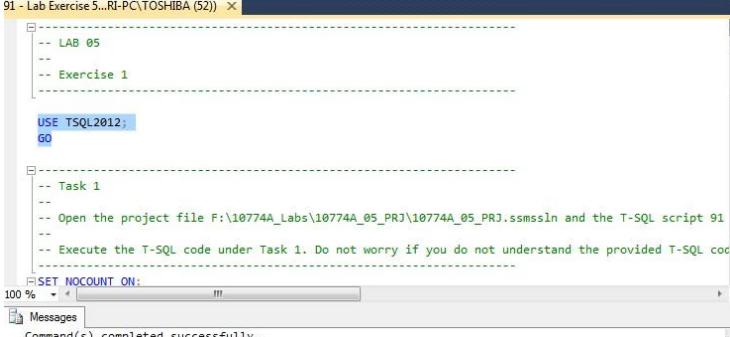
- Yes, there might be NULL values in the orderid column. These would occur for customers who have not placed any orders.

❑ Why are there NULL values?



	<ul style="list-style-type: none">The query uses a LEFT JOIN between Sales.Customers and Sales.Orders. A LEFT JOIN retrieves all rows from the left table (Sales.Customers) and matches them with the right table (Sales.Orders). If a customer does not have any corresponding orders, the orderid column will be NULL for that customer.
5	Conclusion : After doing this part of the practicum, you should understand how to write the TSQL OUTER-JOIN statement .

10 : Creating a Cross-Join Query

Step	Information
	This case study begins with the HR department wanting to set up a personal calendar for each employee. The IT department will provide a T-SQL code that generates all days in the past year. Therefore, <i>the developer</i> will use the SELECT clause to return all rows from the calendar table for each row in the HR.Employees table. Open the project \10774A Labs\10774A_05_PRJ\10774A_05_PRJ.ssmssln and the T-SQL script 91 - Lab Exercise 5.sql. Make sure the database is connected with “TSQL”.
1	 <pre>91 - Lab Exercise 5...RI-PC\TOSHIBA (52) X -- LAB 05 -- Exercise 1 USE TSQL2012; GO -- Task 1 -- Open the project file F:\10774A_Labs\10774A_05_PRJ\10774A_05_PRJ.ssmssln and the T-SQL script 91 -- Execute the T-SQL code under Task 1. Do not worry if you do not understand the provided T-SQL code. SET NOCOUNT ON;</pre>



Practical – Part

```
-- Exercise 1
-- Task 1
-- Open the project file F:\10774A_Labs\10774A_05_PRJ\10774A_05_PRJ.sln and the T-SQL script 91 - Lab Exercise 5.sql. Ensure that you are connected to the correct database.
-- Execute the T-SQL code under Task 1. Do not worry if you do not understand the provided T-SQL code, as it is used here to provide a more realistic example.
SET NOCOUNT ON;

IF OBJECT_ID('HR.Calendar') IS NOT NULL
    DROP TABLE HR.Calendar;

CREATE TABLE HR.Calendar (
    calendardate DATE CONSTRAINT PK_Calendar PRIMARY KEY
);
```

Calendardate
2024-01-01
2024-01-02
2024-01-03
2024-01-04
2024-01-05
2024-01-06
2024-01-07
2024-01-08
2024-01-09
2024-01-10
2024-01-11
2024-01-12
2024-01-13
2024-01-14
2024-01-15
2024-01-16
2024-01-17
2024-01-18
2024-01-19
2024-01-20
2024-01-21
2024-01-22
2024-01-23
2024-01-24
2024-01-25
2024-01-26
2024-01-27
2024-01-28
2024-01-29
2024-01-30
2024-01-31

Query executed successfully.

[Question- 24] Run the T-SQL code under task 1. Display the output! (Don't worry if you don't understand the T-SQL code. The next step will provide a more concrete example of how CROSSJOIN is implemented.)

The T-SQL code under Task 1 creates and populates the HR.Calendar table with dates for the current year. Here is the code:

SQL

```
SET NOCOUNT ON;
```

```
IF OBJECT_ID('HR.Calendar') IS NOT NULL
    DROP TABLE HR.Calendar;
```

```
CREATE TABLE HR.Calendar (
    calendardate DATE CONSTRAINT PK_Calendar PRIMARY KEY
);
```

```
DECLARE
    @startdate DATE = DATEFROMPARTS(YEAR(SYSDATETIME()), 1, 1),
    @enddate DATE = DATEFROMPARTS(YEAR(SYSDATETIME()), 12, 31);
```

```
WHILE @startdate <= @enddate
BEGIN
    INSERT INTO HR.Calendar (calendardate)
    VALUES (@startdate);
```

```
    SET @startdate = DATEADD(DAY, 1, @startdate);
```



```
END;

SET NOCOUNT OFF;

GO

-- Observe the HR.Calendar table
SELECT
    calendardate
FROM HR.Calendar;
```

The screenshot shows the execution of a SQL script in SQLQuery1.sql. The script creates the HR.Calendar table if it does not exist, dropping it first. It then creates a primary key constraint on the calendardate column and inserts all dates from January 1, 2024, to January 31, 2024, into the table. The results pane displays the generated calendar dates.

calendardate
2024-01-01
2024-01-02
2024-01-03
2024-01-04
2024-01-05
2024-01-06
2024-01-07
2024-01-08
2024-01-09
2024-01-10
2024-01-11
2024-01-12
2024-01-13
2024-01-14
2024-01-15
2024-01-16
2024-01-17
2024-01-18
2024-01-19

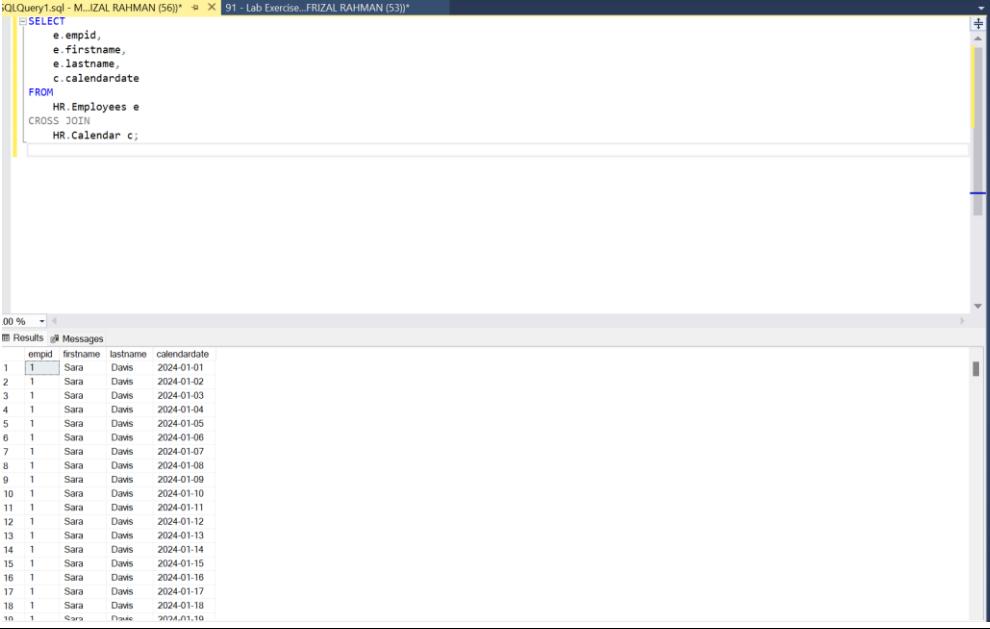
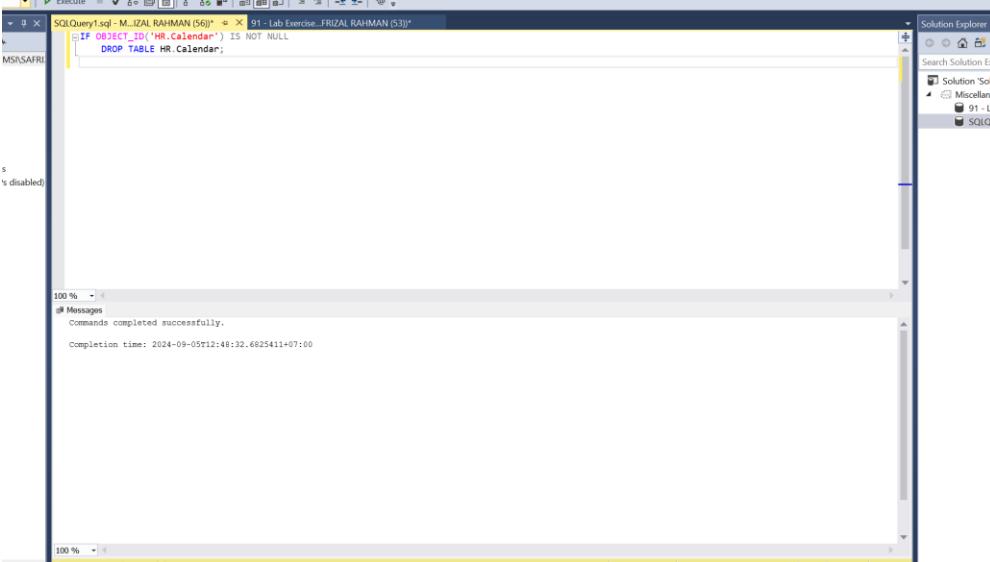
Query executed successfully.



Practical – Part

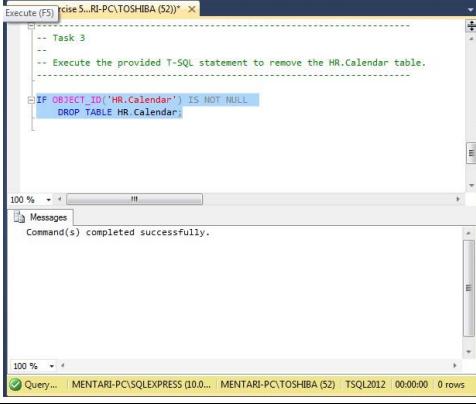
3	<p>[Question- 25] Write a SELECT command to retrieve values from the empid, firstname, and lastname columns from the HR.Employees table and the calendardate column from the HR.Calendar table</p> <p>.</p> <pre>SELECT e.empid, e.firstname, e.lastname, c.calendardate FROM HR.Employees e CROSS JOIN HR.Calendar c;</pre>



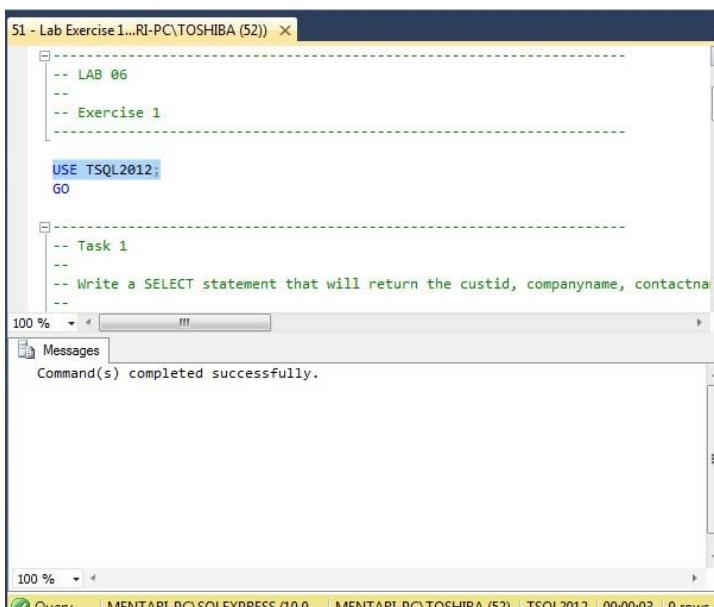
	 <p>The screenshot shows a SQL Server Management Studio window. In the center is a results grid titled 'Results' with columns: empid, firstname, lastname, and calendardate. The data consists of 18 rows, each with empid 1, firstname Sara, lastname Davis, and a date from 2024-01-01 to 2024-01-18. The Solution Explorer on the right shows a project named '91 - Lab Exercise 5' containing a file 'SQLQuery1.sql'.</p>
	<p>[Question-2 6] Execute the 3rd stage test and compare it with the file 92 - Lab Exercise 5 - Task 2 Result.txt . If the results are the same, then your test is correct.</p> <p>By completing these tasks, you've demonstrated your understanding of how to write and execute T-SQL CROSS JOIN queries. If you have any further questions or need additional assistance, feel free to ask!</p>
4	 <p>The screenshot shows a SQL Server Management Studio window with a single command in the query editor: 'IF OBJECT_ID('HR.Calendar') IS NOT NULL DROP TABLE HR.Calendar;'. The results pane shows a message: 'Commands completed successfully.' and a completion time of '2024-09-05T12:48:32.682Z411+07:00'. The Solution Explorer on the right shows a project named 'Solution1' containing files '91 - Lab Exercise 5.sql' and 'SQLQuery1.sql'.</p>
5	Drop the HR.Calendar table by executing the T-SQL code below task 3.



Practical – Part

	
6	Conclusion : After completing this practical section, you will understand how to write T-SQL CROSS-JOIN code .

Practical – Part 11 : Writing Queries Who Will Filter Data with WHERE clause

Step	Information
1	<p>The scenario in this practicum uses the problems in the marketing department. The marketing department is working on several campaigns for old customers. The marketing staff needs a different customer list according to several business rules. Therefore, <i>the developer</i> will write a SELECT command to retrieve the desired rows from the Sales.Customers table.</p> <p>Open the project \10774A Labs\10774A_06_PRJ\10774A_06_PRJ.ssmssln and the T-SQL script 51 - Lab Exercise 1.sql. Make sure the database is connected with “TSQL”.</p> 



```

-- Compare the results that you got with the desired results shown in the file 55 - Lab Exercise 1 - Task 4 Result.txt.
-- Is the result the same as in the first T-SQL statement? Why? What is the difference between specifying the predicate in the ON clause and in the WHERE clause?
-----  

SELECT
    c.custid, c.companyname, o.orderid
FROM Sales.Customers AS c
LEFT OUTER JOIN Sales.Orders AS o ON c.custid = o.custid AND c.city = 'Paris';
-----  

-- Task 5
-- Write a T-SQL statement to retrieve customers from the Sales.Customers table that do not have matching orders in the Sales.Orders table. Matching customers are those who have at least one order.
-- Execute the written statement and compare the results that you got with the desired results shown in the file 56 - Lab Exercise 1 - Task 5 Result.txt.
-----  


```

Results

custid	companyname	orderid
1	Customer NRZBB	NULL
2	Customer MLTDN	NULL
3	Customer KBUDE	NULL
4	Customer HFBZG	NULL
5	Customer HGVLZ	NULL
6	Customer XHXIV	NULL
7	Customer QXVLA	NULL
8	Customer QURWH	NULL
9	Customer RTXGC	NULL
10	Customer EELAV	NULL
11	Customer UBAHU	NULL
12	Customer PSNMQ	NULL
13	Customer VMLQG	NULL
14	Customer WNMAF	NULL
15	Customer JUWKK	NULL
16	Customer GYBBY	NULL
17	Customer FEVNN	NULL
18	Customer BSVAR	NULL
19	Customer REMOC	NULL

Query executed successfully.

Write a SELECT statement that will return the column values from a table, Then filter the results to only customers who are from “Brazil”!

```

SELECT
    custid , companyname , contactname , address , city , country ,
    telephone FROM Sales . Customers WHERE
    country = Brazil' ;

```

Use of the N prefix for literal characters (`N'Brazil'`). This prefix is used because the country column is a Unicode data type. When expressing Unicode characters literally, the N character (for National) is specified as the prefix.

2

```

SELECT
    custid,
    companyname,
    contactname,
    address,
    city,
    country,
    telephone
FROM Sales.Customers
WHERE country = N'Brazil';

```

```

SELECT
    custid,
    companyname,
    contactname,
    address,

```



Practical – Part

```

city,
country,
phone -- Replace 'phone' with the correct column name if 'telephone' is incorrect
FROM Sales.Customers
WHERE country = N'Brazil';

```

The screenshot shows the SQL Server Management Studio interface. In the center pane, there is a query window with the following T-SQL code:

```

SELECT
    custid,
    companyname,
    contactname,
    address,
    city,
    country,
    phone -- Replace 'phone' with the correct column name if 'telephone' is incorrect
FROM Sales.Customers
WHERE country = N'Brazil';

```

Below the query window is the 'Results' tab, which displays the following table of customer data:

	custid	companyname	contactname	address	city	country	phone
1	15	Customer JUWXX	Richardson, Shawn	Av. dos Lusadas, 6789	Sao Paulo	Brazil	(11) 234-5678
2	21	Customer KIDPX	Russo, Giuseppe	Rua Orós, 3456	Sao Paulo	Brazil	(11) 456-7890
3	31	Customer YJCBX	Cheng, Yao-Qiang	Av. Brasil, 5678	Campinas	Brazil	(11) 567-8901
4	34	Customer IBVRG	Cohen, Shy	Rua do Pão, 7890	Rio de Janeiro	Brazil	(21) 789-0123
5	61	Customer WULWD	Florczyk, Krzysztof	Rua da Parificadora, 1234	Rio de Janeiro	Brazil	(21) 678-9012
6	62	Customer WFZJL	Misee, Anna	Alameda dos Canários, 1234	Sao Paulo	Brazil	(11) 901-2345
7	67	Customer QVEPD	Garden, Euan	Av. Copacabana, 6789	Rio de Janeiro	Brazil	(21) 345-789
8	81	Customer YQVWW	Nagel, Jean-Philippe	Av. Irís de Castro, 1234	Sao Paulo	Brazil	(11) 123-567
9	88	Customer SRQVM	Li, Yan	Rua do Mercado, 4567	Resende	Brazil	(14) 234-5678

[Question- 27] Execute the 2nd stage of the test and compare it with the file 52 - Lab Exercise 1 - Task 1 Result.txt . If the results are the same, then your test is correct.

The screenshot shows the SQL Server Management Studio interface. In the center pane, there is a query window with the following T-SQL code:

```

SELECT
    custid,
    companyname,
    contactname,
    address,
    city,
    country,
    phone
FROM Sales.Customers
WHERE country IN (N'Brazil', N'UK', N'USA');

```

Below the query window is the 'Results' tab, which displays the following table of customer data:

	custid	companyname	contactname	address	city	country	phone
1	4	Customer HFBZG	Arndt, Torsten	7890 Hanover Sq	London	UK	(171) 456-7890
2	11	Customer UBBAU	Jaffe, David	Fauntleroy Circus 4567	London	UK	(171) 789-0123
3	15	Customer JUWXX	Richardson, Shawn	Av. dos Lusadas, 6789	Sao Paulo	Brazil	(11) 012-3456
4	16	Customer GYBYB	Birkby, Dana	Berkeley Gardens 0123 Brewery	London	UK	(171) 234-5678
5	19	Customer RFNQC	Boseman, Randall	5678 King George	London	UK	(171) 456-7899
6	21	Customer KIDPX	Russo, Giuseppe	Rua Orós, 3456	Sao Paulo	Brazil	(11) 456-7890
7	31	Customer YJC BX	Cheng, Yao-Qiang	Av. Brasil, 5678	Campinas	Brazil	(11) 567-8901
8	32	Customer YSIQX	Krishnan, Venky	6789 Baker Blvd	Eugene	USA	(503) 555-0122
9	34	Customer IBVRG	Cohen, Shy	Rua do Pão, 7890	Rio de Janeiro	Brazil	(21) 789-0123
10	36	Customer LVJSO	Smith, Denise	City Center Plaza 2345 Main St.	Eggn	USA	(503) 555-0126
11	38	Customer LIUCA	Lee, Frank	Garden House Crowther Way 3456	Cowes	UK	(196) 567-8901
12	43	Customer USIJQ	Deshpande, Anu	8901 Orchestra Terrace	Walla Walla	USA	(509) 555-0119
13	45	Customer QXPPT	Sunkamurrali, Krishna	1234 Polk St. Suite 5	San Francisco	USA	(415) 555-0118
14	48	Customer DMB	Szymczak, Radosław	9012 Chiaroscuro Rd	Portland	USA	(503) 555-0117
15	53	Customer CGSG	Mallik, Ken	South House, 1234 Queensbridge	London	UK	(171) 555-1234
16	55	Customer KZQZD	Egelund-Müller, Anja	7890 Baker St.	Avebury	USA	(607) 555-0115
17	61	Customer WULWD	Florczyk, Krzysztof	Rua da Parificadora, 1234	Rio de Janeiro	Brazil	(21) 678-9012
18	62	Customer WFZJL	Misee, Anna	Alameda dos Canários, 1234	Sao Paulo	Brazil	(11) 901-2345
19	65	Customer MYIUC	Morita, Mirella	4789 Miller Dr.	Athens	USA	(404) 434-0123

At the bottom of the results window, it says "Query executed successfully."



The screenshot shows a SQL Server Management Studio window with two panes. The left pane contains a query editor with the following T-SQL code:

```
SELECT
    custid,
    companyname,
    contactname,
    address,
    city,
    country,
    phone -- Correct column name based on your output
FROM Sales.Customers
WHERE country = N'Brazil';
```

The right pane shows the results of the query, which is a table with the following data:

	custid	companyname	contactname	address	city	country	phone
1	15	Customer JUWXK	Richardson, Shawn	Av. dos Lusíadas, 6789	Sao Paulo	Brazil	(11) 012-3456
2	21	Customer KIDPX	Russo, Giuseppe	Rua Orós, 3456	Sao Paulo	Brazil	(11) 456-7890
3	31	Customer YJCBX	Cheng, Yao-Qiang	Av. Brasil, 5678	Campinas	Brazil	(11) 567-8901
4	34	Customer IBVRG	Cohen, Shy	Rua do Pão, 7890	Rio de Janeiro	Brazil	(21) 789-0123
5	61	Customer WULWD	Florczyk, Krzysztof	Rua da Panificadora, 1234	Rio de Janeiro	Brazil	(21) 678-9012
6	62	Customer WFIZJ	Misic, Anna	Alameda dos Canários, 1234	Sao Paulo	Brazil	(11) 901-2345
7	67	Customer QVEPD	Garden, Euan	Av. Copacabana, 6789	Rio de Janeiro	Brazil	(21) 345-6789
8	81	Customer YQQWW	Nagel, Jean-Philippe	Av. Inês de Castro, 1234	Sao Paulo	Brazil	(11) 123-4567
9	88	Customer SRQVM	Li, Yan	Rua do Mercado, 4567	Resende	Brazil	(14) 234-5678



[Question- 28] Write a SELECT command that will return values in the custid, companyname, contactname, address, city, columns. country, and phone in the Sales.Customers table , then filter the results only for “Brazil, UK and USA” (Use the IN predicate in the WHERE clause).

4

The screenshot shows the SQL Server Management Studio interface with three tabs: SQLQuery3.sql, SQLQuery2.sql, and 51 - Lab Exercise...FRIZAL RAHMAN. The SQLQuery3.sql tab contains the following query:

```
SELECT
    custid,
    companyname,
    contactname,
    address,
    city,
    country,
    phone -- Correct column name based on your output
FROM Sales.Customers
WHERE country = 'N'Brazil';
```

The results pane displays a table with 9 rows of customer data from Brazil:

	custid	companyname	contactname	address	city	country	phone
1	15	Customer JWYXK	Richardson, Shaw	Av. dos Lusiadas, 6789	Sao Paulo	Brazil	(11) 012-3456
2	21	Customer KIDPX	Russo, Giuseppe	Rua Orós, 3456	Sao Paulo	Brazil	(11) 456-7890
3	31	Customer YJC BX	Cheng, Yao-Qiang	Av. Brasil, 5678	Campinas	Brazil	(11) 567-8901
4	34	Customer IBVRG	Cohen, Shy	Rua do Paço, 7890	Rio de Janeiro	Brazil	(21) 789-0123
5	61	Customer WULWD	Florczyk, Krzysztof	Rua da Panificadora, 1234	Rio de Janeiro	Brazil	(21) 678-9012
6	62	Customer WFIZI	Misic, Anna	Alameda dos Canários, 1234	Sao Paulo	Brazil	(11) 901-2345
7	67	Customer QVEPD	Garden, Euan	Av. Copacabana, 6789	Rio de Janeiro	Brazil	(21) 345-6789
8	81	Customer YQQWW	Nagel, Jean-Philippe	Av. Inês de Castro, 1234	Sao Paulo	Brazil	(11) 123-4567
9	88	Customer SRQVM	Li, Yan	Rua do Mercado, 4567	Resende	Brazil	(14) 234-5678

[Question-2 9] Execute the 3rd stage test and compare it with file 53 - Lab Exercise 1 - Task 2 Result.txt . If the results are the same, then your test is correct.

5

The screenshot shows the SQL Server Management Studio interface with three tabs: SQLQuery3.sql, SQLQuery2.sql, and 51 - Lab Exercise...FRIZAL RAHMAN. The SQLQuery3.sql tab contains the following query:

```
SELECT
    custid,
    companyname,
    contactname,
    address,
    city,
    country,
    phone
FROM Sales.Customers
WHERE contactname LIKE 'A%';
```

The results pane displays a table with 2 rows of customer data where contactname starts with 'A':

	custid	companyname	contactname	address	city	country	phone
1	1	Customer NRZBB	Allen, Michael	Obere Str. 0123	Berlin	Germany	030-3456789
2	4	Customer HFBZG	Arndt, Torsten	7890 Hanover Sq.	London	UK	(171) 456-7890



SQLQuery3.sql - M...IZAL RAHMAN (57)* 51 - Lab Exercise...FRIZAL RAHMAN (60)* SQLQuery2.sql - M...IZAL RAHMAN (59)* 51 - Lab Exercise...FRIZAL RAHMAN (54)

```

SELECT
    custid,
    companyname,
    contactname,
    address,
    city,
    country,
    phone
FROM Sales.Customers
WHERE country IN (N'Brazil', N'UK', N'USA');

```

100 %

Results Messages

	custid	companyname	contactname	address	city	country	phone
1	4	Customer HFBZG	Amdt, Torsten	7890 Hanover Sq.	London	UK	(171) 456-7890
2	11	Customer UBHAU	Jaffe, David	Fauntleroy Circus 4567	London	UK	(171) 789-0123
3	15	Customer JUWXK	Richardson, Shawn	Av. dos Lusíadas, 6789	Sao Paulo	Brazil	(11) 012-3456
4	16	Customer GYBBY	Birkby, Dana	Berkeley Gardens 0123 Brewery	London	UK	(171) 234-5678
5	19	Customer RFNQC	Boseman, Randall	5678 King George	London	UK	(171) 345-6789
6	21	Customer KIDPX	Russo, Giuseppe	Rua Orós, 3456	Sao Paulo	Brazil	(11) 456-7890
7	31	Customer YJC BX	Cheng, Yao-Qiang	Av. Brasil, 5678	Campinas	Brazil	(11) 567-8901
8	32	Customer YSIQX	Krishnan, Venky	6789 Baker Blvd	Eugene	USA	(503) 555-0122
9	34	Customer IBVRG	Cohen, Shy	Rua do Pão, 7890	Rio de Janeiro	Brazil	(21) 789-0123
10	36	Customer LVJSO	Smith, Denise	City Center Plaza 2345 Main St.	Elgin	USA	(503) 555-0126
11	38	Customer LJUCA	Lee, Frank	Garden House Crowther Way 3456	Cowes	UK	(198) 567-8901
12	43	Customer UISOJ	Deshpande, Anu	8801 Orchestra Terrace	Walla Walla	USA	(509) 555-0119
13	45	Customer QXPPT	Sunkammuri, Krishna	1234 Polk St. Suite 5	San Francisco	USA	(415) 555-0118
14	48	Customer DVFM B	Szymczak, Radosław	9012 Chiaroscuro Rd.	Portland	USA	(503) 555-0117
15	53	Customer GCJSG	Mallit, Ken	South House 1234 Queensbridge	London	UK	(171) 890-1234
16	55	Customer KZQZT	Egelund-Müller, Anja	7890 Bering St.	Anchorage	USA	(907) 555-0115
17	61	Customer WULWD	Florczyk, Krzysztof	Rua da Panificadora, 1234	Rio de Janeiro	Brazil	(21) 678-9012
18	62	Customer WFIZJ	Misic, Anna	Alameda dos Canários, 1234	Sao Paulo	Brazil	(11) 901-2345

The IT department has written T-SQL code to return values in the custid, companyname columns in the Sales.Customers table and the orderid column in the Sales.Orders table as below:

```

SELECT c . custid , c . companyname , o . orderid
FROM Sales . Customers AS c
LEFT OUTER JOIN Sales . Orders AS o ON c . custid = o . custid AND c . city = 'Paris' ;

```

SQLQuery3.sql - M...IZAL RAHMAN (57)* 51 - Lab Exercise...FRIZAL RAHMAN (60)* SQLQuery2.sql - M...IZAL RAHMAN (59)* 51 - Lab Exercise...FRIZAL RAHMAN (54)

```

SELECT c . custid , c . companyname , o . orderid
FROM Sales . Customers AS c
LEFT OUTER JOIN Sales . Orders AS o ON c . custid = o . custid AND c . city = 'Paris' ;

```

6

100 %

Results Messages

	custid	companyname	orderid
1	1	Customer NRZBB	NULL
2	2	Customer MLTDN	NULL
3	3	Customer KBUDE	NULL
4	4	Customer HFBZG	NULL
5	5	Customer HGVLZ	NULL
6	6	Customer XHXJV	NULL
7	7	Customer QXVLA	NULL
8	8	Customer QUHWH	NULL
9	9	Customer RTXGC	NULL
10	10	Customer EALV	NULL
11	11	Customer UBHAU	NULL
12	12	Customer PSNMQ	NULL
13	13	Customer VMLOG	NULL
14	14	Customer WNMAF	NULL
15	15	Customer JUWXK	NULL
16	16	Customer GYBBY	NULL
17	17	Customer FEVNN	NULL
18	18	Customer RSVAR	NULL



Query execution in the 7th stage of the trial. Note two things, first the query will retrieve all rows in the Sales.Customers table . Second, the use of the comparison operator with the ON clause makes the city column more specific, namely the same as the value "Paris".

7

```
-- Is the result the same as in the first T-SQL statement? Why? What is the difference?  
--  
SELECT  
    c.custid, c.companyname, o.orderid  
FROM Sales.Customers AS c  
LEFT OUTER JOIN Sales.Orders AS o ON c.custid = o.custid AND c.city = 'Paris';  
  
-- Task 5  
--  
-- Write a T-SQL statement to retrieve customers from the Sales.Customers table  
--  
100 % 4 !!!  
Results Messages  
custid companyname orderid  
1 Customer NRZBB NULL  
2 Customer MLTDN NULL  
3 Customer KBUDE NULL  
4 Customer HFBZG NULL  
5 Customer HGVLZ NULL  
6 Customer XHXJV NULL  
7 Customer QXVLA NULL  
8 Customer QUHWH NULL  
9 Customer RTXGC NULL  
Query... | MENTARI-PC\SQLEXPRESS (10.0... | MENTARI-PC\TOSHIBA (52) | TSQl2012 | 00:00:00 | 94 rows
```

[Question-30] Copy the T-SQL Code in step 7 then modify it with the comparison operator for the city column in the WHERE clause. After that execute the code, show the result!

8

```
SQLQuery4.sql - M...IZAL RAHMAN (55)*  SQLQuery3.sql - M...IZAL RAHMAN (57)*  51 - Lab Exercise...FRIZAL RAHMAN (60)*  SQLQuery2.sql - M...IZAL RAHMAN (59)*  
SELECT  
    c.custid, c.companyname, o.orderid  
FROM Sales.Customers AS c  
LEFT OUTER JOIN Sales.Orders AS o ON c.custid = o.custid  
WHERE c.city = 'Paris';  
  
100 % 4 !!!  
Results Messages  
custid companyname orderid  
1 57 Customer WVAXS NULL  
2 74 Customer YSHXL 10738  
3 74 Customer YSHXL 10907  
4 74 Customer YSHXL 10964  
5 74 Customer YSHXL 11043  
Query... | MENTARI-PC\SQLEXPRESS (10.0... | MENTARI-PC\TOSHIBA (52) | TSQl2012 | 00:00:00 | 94 rows
```



The screenshot shows a SQL Server Management Studio window with three tabs at the top: 'SQLQuery3.sql - M...IZAL RAHMAN (57)*' (active), '51 - Lab Exercise...FRIZAL RAHMAN (60)*', and 'SQLQuery2.sql - M...IZAL RAHMAN (59)*'. The active query window contains the following T-SQL code:

```

SELECT
    c.custid,
    c.companyname
FROM Sales.Customers AS c
LEFT OUTER JOIN Sales.Orders AS o
ON c.custid = o.custid
WHERE o.custid IS NULL;

```

The 'Results' pane below displays the output of the query:

	custid	companyname
1	22	Customer DTDMN
2	57	Customer WVAXS

Compare the results of step 9 with file 55 - Lab Exercise 1 - Task 4 Result.txt . If the results are the same, then your test is correct.

9

custid	companyname	orderid
57	Customer WVAXS	NULL
74	Customer YSHXL	10738
74	Customer YSHXL	10907
74	Customer YSHXL	10964
74	Customer YSHXL	11043

(5 row(s) affected)

10

Conclusion : After completing the practicum and answering the questions in this section, you should understand how to filter data rows from one or more tables using the WHERE clause with logical operator predicates.

Practical – Part 11 : Writing Queries Which Will Sort Data with clause ORDER BY

Step	Information
1	<p>The case study in this lab is based on a problem in the sales department. The sales department wants to create a report that shows all orders with some customer information. In addition, there is an additional request to sort the data based on order dates and the customer IDs. The order rows in the previous lab were displayed without using the ORDER BY clause, therefore specifically for this lab section the WHERE command will be followed by the ORDER BY clause.</p> <p>Open the project \10774A Labs\10774A_06_PRJ\10774A_06_PRJ.ssmssln and the T-SQL script 61 - Lab Exercise 2.sql . Make sure the database is connected with “TSQL”.</p>



The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. At the top, there are two tabs: "61 - Lab Exercise 2...RI-PC\TOSHIBA (54)" and "51 - Lab Exercise 1...RI-PC\TOSHIBA (52)". The left pane displays a tree view of database objects under "LAB 06" and "Exercise 2". The right pane contains a query editor window with the following T-SQL script:

```
USE TSQl2012;
GO

-- Task 1
-- Open the project file F:\10774A_Labs\10774A_06_PRJ\10774A_06_PRJ.ssmssln and
--
```

The "Messages" tab at the bottom of the query editor shows the message: "Command(s) completed successfully."

Below the main window, there are two other tabs: "61 - Lab Exercise_FRIZAL RAHMAN (61)*" and "SQLQuery4.sql - M...IZAL RAHMAN (55)*". The "61 - Lab Exercise_FRIZAL RAHMAN (61)*" tab contains a large block of T-SQL comments describing tasks and results. The "Solution Explorer" window on the right lists files for "Solution1": "51 - Lab Exercise 1.sql", "51 - Lab Exercise 2.sql", "61 - Lab Exercise 2.sql", "91 - Lab Exercise 5.sql", "SQLQuery1.sql", "SQLQuery2.sql", "SQLQuery3.sql", and "SQLQuery4.sql".

2

[Question- 31] Write a SELECT command to retrieve the custid, custname columns from the Sales.Customers table and the orderid, orderdate columns from the Sales.Orders table ! Filter the results only for orders on or after April 1, 2008. Then sort the results based on orderdate in descending order and custid in ascending order!



The screenshot shows the SSMS interface with multiple tabs open. The main tab displays a query result grid with 88 rows. The columns are labeled: custid, contactname, orderid, and orderdate. The results show various customer records from the Sales.Customers and Sales.Orders tables, ordered by orderdate in descending order and custid in ascending order. The grid includes column headers and a total row count of 88.

	custid	contactname	orderid	orderdate
1	9	Raghav, Amitansh	11076	2008-05-06 00:00:00.000
2	65	Moore, Michael	11077	2008-05-06 00:00:00.000
3	68	Myrcha, Jacek	11075	2008-05-06 00:00:00.000
4	73	Gonzalez, Nuria	11074	2008-05-06 00:00:00.000
5	20	Kane, John	11072	2008-05-05 00:00:00.000
6	44	Louverdis, George	11070	2008-05-05 00:00:00.000
7	46	Dressler, Marlies	11071	2008-05-05 00:00:00.000
8	58	Fakhouri, Fadi	11073	2008-05-05 00:00:00.000
9	17	Jones, TiAnna	11067	2008-05-04 00:00:00.000
10	62	Misic, Anna	11068	2008-05-04 00:00:00.000
11	80	Geschwandtner, Jens	11069	2008-05-04 00:00:00.000
12	46	Dressler, Marlies	11065	2008-05-01 00:00:00.000
13	71	Navarro, Tomás	11064	2008-05-01 00:00:00.000
14	89	Smith Jr., Ronaldo	11066	2008-05-01 00:00:00.000
15	27	Schmollet, Martin	11060	2008-04-30 00:00:00.000
16	32	Krishnan, Venky	11061	2008-04-30 00:00:00.000
17	37	Crăciun, Ovidiu V.	11063	2008-04-30 00:00:00.000
18	66	Voss, Florian	11062	2008-04-30 00:00:00.000
19	6	Poland, Carola	11058	2008-04-29 00:00:00.000

MSI (14.0 RTM) | MSI\SAFRIZAL RAHMAN (52) | TSQL | 00:00:00 | 88 rows
Query executed successfully.

[Question- 32] Execute the 2nd stage of the test and compare it with the file 62 - Lab Exercise 2 - Task 1 Result.txt . If the results are the same, then your test is correct.

The screenshot shows the SSMS interface with multiple tabs open. The main tab displays a query result grid with 88 rows. The columns are labeled: custid, contactname, orderid, and orderdate. The results show various customer records from the Sales.Customers and Sales.Orders tables, ordered by orderdate in descending order and custid in ascending order. The grid includes column headers and a total row count of 88.

	custid	contactname	orderid	orderdate
1	9	Raghav, Amitansh	11076	2008-05-06 00:00:00.000
2	65	Moore, Michael	11077	2008-05-06 00:00:00.000
3	68	Myrcha, Jacek	11075	2008-05-06 00:00:00.000
4	73	Gonzalez, Nuria	11074	2008-05-06 00:00:00.000
5	20	Kane, John	11072	2008-05-05 00:00:00.000
6	44	Louverdis, George	11070	2008-05-05 00:00:00.000
7	46	Dressler, Marlies	11071	2008-05-05 00:00:00.000
8	58	Fakhouri, Fadi	11073	2008-05-05 00:00:00.000
9	17	Jones, TiAnna	11067	2008-05-04 00:00:00.000
10	62	Misic, Anna	11068	2008-05-04 00:00:00.000
11	80	Geschwandtner, Jens	11069	2008-05-04 00:00:00.000
12	46	Dressler, Marlies	11065	2008-05-01 00:00:00.000
13	71	Navarro, Tomás	11064	2008-05-01 00:00:00.000
14	89	Smith Jr., Ronaldo	11066	2008-05-01 00:00:00.000
15	27	Schmollet, Martin	11060	2008-04-30 00:00:00.000
16	32	Krishnan, Venky	11061	2008-04-30 00:00:00.000
17	37	Crăciun, Ovidiu V.	11063	2008-04-30 00:00:00.000
18	66	Voss, Florian	11062	2008-04-30 00:00:00.000
19	6	Poland, Carola	11058	2008-04-29 00:00:00.000

3



The T-SQL command from the previous practicum followed by the WHERE command is as follows:

```
SELECT
e . empid , e . lastname , e . firstname , e . title , e . mgrid , m
. lastname AS mgrlastname , m . firstname AS mgrfirstname FROM HR .
Employees AS e
INNER JOIN HR . Employees AS m ON e . mgrid = m . empid WHERE
mgrlastname = N'Buck' ;
```

[Question- 33] Execute the T-SQL command at stage 3. Did an error occur? What is the error message? What do you think is the cause?

Msg 207, Level 16, State 1, Line 4

Invalid column name 'mgrlastname'.

[Question-3 4] Make changes to the T-SQL command to fix the error in the 3rd trial, then execute it! Compare the execution results with the file 63 - Lab Exercise 2 - Task 2 Result.txt. If the same, then the test result is correct.

The screenshot shows the SQL Server Management Studio interface. At the top, there are three tabs: 'SQLQuery5.sql - M...IZAL RAHMAN (52)*', '61 - Lab Exercise...FRIZAL RAHMAN (61)*', and 'SQLQuery4.sql - M...IZAL RAHMAN (55)*'. Below them is a 'Results' tab with a 'Messages' section containing the error message: 'Msg 207, Level 16, State 1, Line 4 Invalid column name 'mgrlastname''. The main query pane contains the following T-SQL code:

```
SELECT
e.empid,
e.lastname,
e.firstname,
e.title,
e.mgrid,
m.lastname AS mgrlastname,
m.firstname AS mgrfirstname
FROM HR.Employees AS e
INNER JOIN HR.Employees AS m ON e.mgrid = m.empid
WHERE m.lastname = N'Buck';
```

The 'Results' tab also displays the output of the query, which is identical to the content of the '63 - Lab Exercise 2 - Task 2 Result.txt' file shown below it. Both the 'Results' pane and the 'File' pane show the following data:

	empid	lastname	firstname	title	mgrid	mgrlastname	mgrfirstname
1	6	Suurs	Paul	Sales Representative	5	Buck	Sven
2	7	King	Russell	Sales Representative	5	Buck	Sven
3	9	Dolgopyatova	Zoya	Sales Representative	5	Buck	Sven

(3 row(s) affected)



[Question- 35] Copy the T-SQL command in experiment 4, and modify it to produce all employees ORDER BY manager's first name. Initially test using the table's original name, then test using the table's alias name! Execute the T-SQL and compare the results to the 64 - Lab Exercise 2 - Task 3 Result.txt file . If the results are the same, then the experiment was correct.

empid	lastname	firstname	title	mgrid	mgrlastname
3	Lew	Judy	Sales Manager	2	Funk
5	Buck	Sven	Sales Manager	2	Funk
4	Peled	Yael	Sales Representative	3	Lew
8	Cameron	Maria	Sales Representative	3	Lew
2	Funk	Don	Vice President, Sales	1	Davis
6	Suurs	Paul	Sales Representative	5	Buck
7	King	Russell	Sales Representative	5	Buck
9	Dolgopyatova	Zoya	Sales Representative	5	Buck

(8 row(s) affected)

5

```

SELECT
    e.empid,
    e.lastname,
    e.firstname,
    e.title,
    e.mgrid,
    m.lastname AS mgrlastname,
    m.firstname AS mgrfirstname
FROM HR.Employees AS e
INNER JOIN HR.Employees AS m ON e.mgrid = m.empid
ORDER BY m.firstname;

```

The screenshot shows the SQL Server Management Studio interface. It has three tabs open: SQLQuery5.sql, SQLQuery4.sql, and SQLQuery3.sql. The SQLQuery5.sql tab contains the provided T-SQL code. The Results tab shows the execution of the query, displaying the same data as the text above. The Solution Explorer on the right shows a solution named 'Solution1' with various files listed.

empid	lastname	firstname	title	mgrid	mgrlastname	mgrfirstname
1 3	Lew	Judy	Sales Manager	2	Funk	Don
2 5	Buck	Sven	Sales Manager	2	Funk	Don
3 4	Peled	Yael	Sales Representative	3	Lew	Judy
4 8	Cameron	Maria	Sales Representative	3	Lew	Judy
5 2	Funk	Don	Vice President, Sales	1	Davis	Sara
6 6	Suurs	Paul	Sales Representative	5	Buck	Sven
7 7	King	Russell	Sales Representative	5	Buck	Sven
8 9	Dolgopyatova	Zoya	Sales Representative	5	Buck	Sven

[Question-3 6] Why can we use column names according to the original table name or use table alias names?

6

```

SELECT
    e.empid,
    e.lastname,

```



	<pre> e.firstname, e.title, e.mgrid, m.lastname AS mgrlastname, m.firstname AS mgrfirstname FROM HR.Employees AS e INNER JOIN HR.Employees AS m ON e.mgrid = m.empid ORDER BY mgrfirstname; </pre> <p>Clarity and Precision: Using table aliases helps make your queries clearer, especially when joining multiple tables. It ensures that the columns are correctly referenced and avoids ambiguity.</p> <p>Flexibility: SQL allows both approaches for flexibility. If table aliases are used, they provide a more concise way to write queries, and they can be used interchangeably with original table names if needed.</p>
7	<p>Conclusion : After working on the practical work and questions in this section, you should now understand how to use the ORDER BY clause .</p>

Practical – Part 12 : Writing Queries Who Will Do Data Filtering with clauses TOP

Step	Information
1	<p>Part 8 of the lab uses a case study on the sales department. The sales department wants to create an additional report that shows the order invoices and the 10 percent of the most expensive products that have been sold.</p> <p>Open the project \10774A_Labs\10774A_06_PRJ\10774A_06_PRJ.ssmssln and the T-SQL script 71 - Lab Exercise 3.sql . Make sure the database is connected with “TSQL”.</p>



The screenshot shows a SQL Server Management Studio (SSMS) interface. On the left, a script window displays a series of comments and a T-SQL command:

```
-- LAB 06
-- 
-- Exercise 3
-- 
USE TSQL;
GO

-- Task 1
-- 
-- Open the project file F:\10774A_Labs\10774A_06_PRJ\10774A_06_PRJ.ssmssln and the SQL script 71 - Lab Exercise 3.sql. Ensure that you are connected to ...
-- 
-- Write a SELECT statement against the Sales.Orders table and retrieve the orderid and orderdate columns. Retrieve the last 20 orders, based on orderdat ...
-- 
-- Execute the written statement and compare the results that you got with the recommended result shown in the file 72 - Lab Exercise 3 - Task 1 Result.t ...
-- 
```

Below the script window, the status bar shows "Completion time: 2024-09-05T13:25:23.9829881+07:00". On the right, the Solution Explorer window lists several files under "Solution 'Solution1' (0 projects)" and "Miscellaneous Files".

[Question- 37] Write a SELECT command to display the productname and unitprice columns in the Production.Products table sorted descending by unitprice! Show the execution results!

2

The screenshot shows the SSMS interface with a query window containing the following T-SQL code:

```
SELECT
    productname,
    unitprice
FROM Production.Products
ORDER BY unitprice DESC;
```

Below the query window, the Results pane displays the output:

	productname	unitprice
1	Product QDOMO	263.50
2	Product VJXYN	123.79
3	Product AOZBW	97.00
4	Product QHFFP	81.00
5	Product CKEDC	62.50
6	Product UKXRI	55.00
7	Product APITJ	53.00
8	Product WUXYK	49.30
9	Product ZZHR	46.00
10	Product OFBNT	45.60
11	Product SMOIH	43.90
12	Product ICKNK	43.90
13	Product WVJFP	40.00
14	Product BLCA	39.00
15	Product OSFNS	38.00
16	Product VKCMF	38.00
17	Product COAXA	36.00
18	Product GEEOO	34.80
19	Product WHRYK	34.00

At the bottom of the Results pane, a message indicates "Query executed successfully." and shows the session details: MSI (14.0 RTM) | MSI\SAFRIZAL RAHMAN (65) | TSQl | 00:00:00 | 77 rows.



[Question- 38] Copy and modify the T-SQL command in trial 2 with the limitation that only 10 percent of the child products are displayed based on unitprice ordering! Execute the command, and compare whether it is in accordance with the file 73 - Lab Exercise 3 - Task 2 Result.txt.

3

```
73 - Lab Exercise 3 - Task 2 Result.txt 71 - Lab Exercise 3...RI-PC\TOSHIBA (52)*  
productname          unitprice  
-----  
Product QDOMO        263.50  
Product VJXYN        123.79  
Product AOZBW        97.00  
Product QHFFP        81.00  
Product CKEDC        62.50  
Product UKXRI        55.00  
Product APITJ        53.00  
Product WUXYK        49.30  
  
(8 row(s) affected)  
  
;WITH SortedProducts AS (  
    SELECT  
        productname,  
        unitprice,  
        ROW_NUMBER() OVER (ORDER BY unitprice DESC) AS RowNum,  
        COUNT(*) OVER () AS TotalRows  
    FROM Production.Products  
)  
SELECT  
    productname,  
    unitprice  
FROM SortedProducts  
WHERE RowNum <= (TotalRows * 0.10);
```

	productname	unitprice
1	Product QDOMO	263.50
2	Product VJXYN	123.79
3	Product AOZBW	97.00
4	Product QHFFP	81.00
5	Product CKEDC	62.50
6	Product UKXRI	55.00
7	Product APITJ	53.00

4

[Question- 39] Is it possible to implement the 5 trial T-SQL command using the OFFSET-FETCH clause?



Yes, you can use the OFFSET-FETCH clause to achieve similar results, but it works with a fixed number of rows or pages, not percentages directly. However, you can combine it with a row count calculation. For instance, if you know the total number of rows, you can calculate the number of rows to fetch for 10% and use OFFSET-FETCH to retrieve that subset:

sql

Copy code

```
;WITH SortedProducts AS (
    SELECT
        productname,
        unitprice,
        ROW_NUMBER() OVER (ORDER BY unitprice DESC) AS RowNum,
        COUNT(*) OVER () AS TotalRows
    FROM Production.Products
)
SELECT
    productname,
    unitprice
FROM SortedProducts
ORDER BY unitprice DESC
OFFSET 0 ROWS
FETCH NEXT CAST((TotalRows * 0.10) AS INT) ROWS ONLY;
```

In this example, OFFSET 0 ROWS starts at the beginning, and FETCH NEXT CAST((TotalRows * 0.10) AS INT) ROWS ONLY limits the number of rows fetched to approximately 10% of the total rows. Note that FETCH works with exact row numbers, so you need to compute the number of rows based on the total available.

-- First, get the total number of rows

```
DECLARE @TotalRows INT;
```

```
SELECT @TotalRows = COUNT(*)
FROM Production.Products;
```

-- Now, use OFFSET-FETCH to get the top 10% rows

```
WITH SortedProducts AS (
    SELECT
        productname,
        unitprice,
        ROW_NUMBER() OVER (ORDER BY unitprice DESC) AS RowNum
    FROM Production.Products
)
SELECT
    productname,
```



```

unitprice
FROM SortedProducts
ORDER BY unitprice DESC
OFFSET 0 ROWS
FETCH NEXT CAST(@TotalRows * 0.10 AS INT) ROWS ONLY;

```

```

SQLQuery6.sql - M...IZAL RAHMAN (65)* 71 - Lab Exercise...FRIZAL RAHMAN (63)* SQLQuery5.sql - M...IZAL RAHMAN (52)* 61 - Lab Exercise...FRIZAL RAHMAN (61)*
DECLARE @TotalRows INT;

SELECT @TotalRows = COUNT(*)
FROM Production.Products;

-- Now, use OFFSET-FETCH to get the top 10% rows
WITH SortedProducts AS (
    SELECT
        productname,
        unitprice,
        ROW_NUMBER() OVER (ORDER BY unitprice DESC) AS RowNum
    FROM Production.Products
)
SELECT
    productname,
    unitprice
FROM SortedProducts
ORDER BY unitprice DESC
OFFSET 0 ROWS
FETCH NEXT CAST(@TotalRows * 0.10 AS INT) ROWS ONLY;

```

Results

	productname	unitprice
1	Product QDOMO	263.50
2	Product VJXYN	123.79
3	Product AOZBW	97.00
4	Product QHFFP	81.00
5	Product CKEDC	62.50
6	Product UKXRI	55.00
7	Product APITJ	53.00

5

Conclusion : After completing the practical work and questions in this section, you should now understand how to apply the TOP option to the SELECT clause of the T-SQL command.

Practical – Part 13 : Writing Queries Who Will Filter Data with OFFSET-FETCH clause

Step	Information
1	<p>Practical part 9 will implement paging solution to display rows from Sales.Orders table , because the number of rows is too many. On each report page, user can only see 20 rows.</p> <p>Open the project \10774A Labs\10774A_06_PRJ\10774A_06_PRJ.ssmssln and the T-SQL script 81 - Lab Exercise 4.sql . Make sure the database is connected with “TSQL”.</p>



The screenshot shows the SSMS interface with two query windows open:

- Query 1 (Top Window):** Contains T-SQL code to use the TSQLEXPRESS database and a comment about opening the project file. The message pane shows "Command(s) completed successfully."
- Query 2 (Bottom Window):** Contains T-SQL code to use the TSQLEXPRESS database and a comment about writing a SELECT statement to retrieve custid, orderid, and orderdate from the Sales.Orders table, ordered by orderdate and orderid. It also mentions comparing results with a file named "82 - Lab Exercise 4 - Task 1 Result.txt". The message pane shows "Commands completed successfully."

The Solution Explorer on the right lists various files related to the exercise, including "81 - Lab Exercise 4.sql" which is currently selected.

2

[Question- 40] Write a SELECT command to display the custid, orderid, and orderdate columns in the Sales.Orders table . Sort the rows by orderdate and orderid. Take the first 20 rows. Execute the command and compare the results with the file 82 - Lab Exercise 4 - Task 1 Result.txt. If the results are the same, then your test is correct.



The screenshot shows the SQL Server Management Studio interface. In the top tab bar, there are four tabs: 'SQLQuery7.sql - M...IZAL RAHMAN (72)*' (selected), '81 - Lab Exercise...FRIZAL RAHMAN (66)*', 'SQLQuery6.sql - M...IZAL RAHMAN (65)*', and '71 - Lab Exercise...FRIZAL RAHMAN (63)*'. The main area displays a T-SQL query:

```
SELECT
    custid,
    orderid,
    orderdate
FROM Sales.Orders
ORDER BY orderdate, orderid
OFFSET 0 ROWS
FETCH NEXT 20 ROWS ONLY;
```

Below the query, the results pane shows a table with columns 'custid', 'orderid', and 'orderdate'. The data consists of 20 rows, starting from the 21st row of the result set:

	custid	orderid	orderdate
1	85	10248	2006-07-04 00:00:00.000
2	79	10249	2006-07-05 00:00:00.000
3	34	10250	2006-07-08 00:00:00.000
4	84	10251	2006-07-08 00:00:00.000
5	76	10252	2006-07-09 00:00:00.000
6	34	10253	2006-07-10 00:00:00.000
7	14	10254	2006-07-11 00:00:00.000
8	68	10255	2006-07-12 00:00:00.000
9	88	10256	2006-07-15 00:00:00.000
10	35	10257	2006-07-16 00:00:00.000
11	20	10258	2006-07-17 00:00:00.000
12	13	10259	2006-07-18 00:00:00.000
13	56	10260	2006-07-19 00:00:00.000
14	61	10261	2006-07-19 00:00:00.000
15	65	10262	2006-07-22 00:00:00.000
16	20	10263	2006-07-23 00:00:00.000
17	24	10264	2006-07-24 00:00:00.000
18	7	10265	2006-07-25 00:00:00.000
19	87	10266	2006-07-26 00:00:00.000
20		10267	2006-07-29 00:00:00.000

At the bottom of the results pane, it says 'Query executed successfully.'

The screenshot shows a command-line window titled '82 - Lab Exercise 4 - Task 1 Result.txt'. The window displays the same T-SQL query and its results as shown in the previous screenshot. The results are identical, showing 20 rows of data from the 'Sales.Orders' table.

```
custid      orderid      orderdate
-----      -----      -----
85          10248        2006-07-04 00:00:00.000
79          10249        2006-07-05 00:00:00.000
34          10250        2006-07-08 00:00:00.000
...
7           10265        2006-07-25 00:00:00.000
87          10266        2006-07-26 00:00:00.000
25          10267        2006-07-29 00:00:00.000
```

(20 row(s) affected)

❑ ORDER BY orderdate, orderid sorts the result by orderdate first and then by orderid.

❑ OFFSET 0 ROWS skips no rows.

❑ FETCH NEXT 20 ROWS ONLY retrieves the next 20 rows after the offset.

[Question- 41] Write a SELECT statement to display the same results as question no. 43, skip the first 20 rows, and continue with the next 20 rows using the OFFSET-FETCH clause! Execute the statement and compare 83 - Lab Exercise 4 - Task 2 Result.txt. If the results are the same, then your test is correct.

❑ ORDER BY orderdate, orderid sorts the result by orderdate and orderid.

❑ OFFSET 20 ROWS skips the first 20 rows.



? FETCH NEXT 20 ROWS ONLY retrieves the next 20 rows after the offset.

The screenshot shows a SQL Server Management Studio window. In the center is a results grid titled 'Results' with columns 'custid', 'orderid', and 'orderdate'. The data consists of 20 rows of customer ID, order ID, and order date. The first few rows are: 1 33 2006-07-30 00:00:00.000, 2 89 2006-07-31 00:00:00.000, 3 87 2006-08-01 00:00:00.000, 4 75 2006-08-01 00:00:00.000, 5 65 2006-08-02 00:00:00.000, 6 63 2006-08-05 00:00:00.000, 7 85 2006-08-06 00:00:00.000, 8 49 2006-08-07 00:00:00.000, 9 80 2006-08-08 00:00:00.000, 10 52 2006-08-09 00:00:00.000, 11 5 2006-08-12 00:00:00.000, 12 44 2006-08-13 00:00:00.000, 13 5 2006-08-14 00:00:00.000, 14 69 2006-08-14 00:00:00.000, 15 69 2006-08-15 00:00:00.000, 16 46 2006-08-16 00:00:00.000, 17 44 2006-08-19 00:00:00.000, 18 63 2006-08-20 00:00:00.000, 19 63 2006-08-21 00:00:00.000. At the bottom of the results grid, it says '(20 row(s) affected)'. Below the results grid, a status bar shows 'MSI (14.0 RTM) MSI\SAFRIZAL RAHMAN (72) TSQL 00:00:00 20 rows'.

The screenshot shows a text file titled '83 - Lab Exercise 4 - Task 2 Result.txt'. The file contains the same 20 rows of data as the previous screenshot, with columns 'custid', 'orderid', and 'orderdate'. The data is identical to the results grid shown above.

4

Conclusion : After working on the practical work and questions in this section, you should now understand how to use the OFFSET-FETCH clause in T-SQL commands.

-- *Have a great time doing it -*