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

Topics

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

Objective

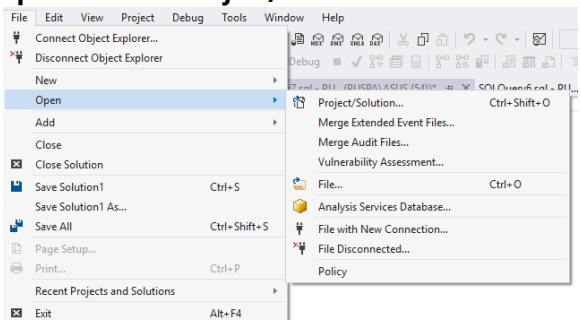
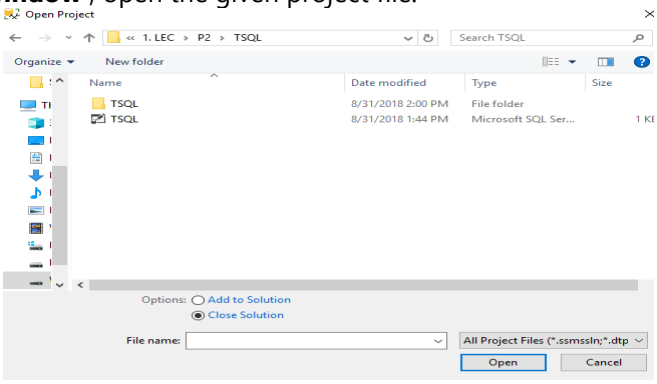
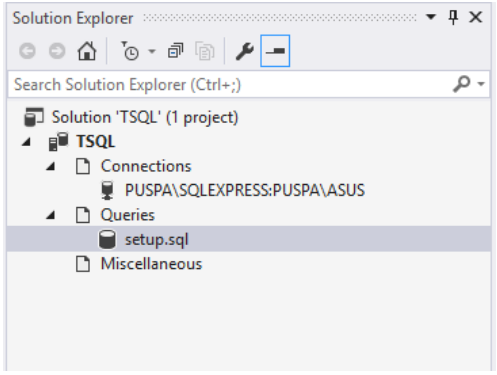
Students are expected to be able to:

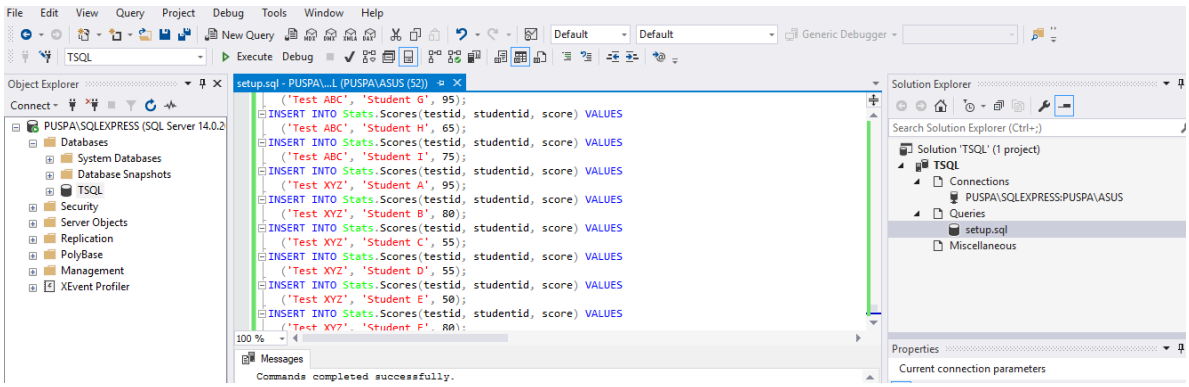
1. Understanding the basic differences between Transact-SQL (T-SQL) and ANSI SQL.
2. Understanding how to create *a database* from an existing SQL file
3. Understand how to execute part or all of a SQL *script* from an existing file.
4. Understanding the concept of using ' *comments* ' in T-SQL.
5. Understand 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.

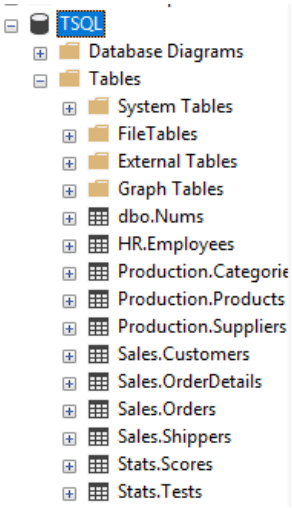
Practical Preparation: Creating a Database from Existing SQL

Step	Information
1	Create a TSQL database `CREATE DATABASE TSQL;`
2	On the **File menu** , click **Open** and click **Project/Solution** . 
3	In the **Open Project window** , open the given project file. 
4	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. 
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
```

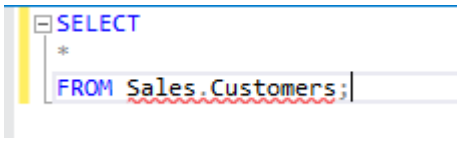
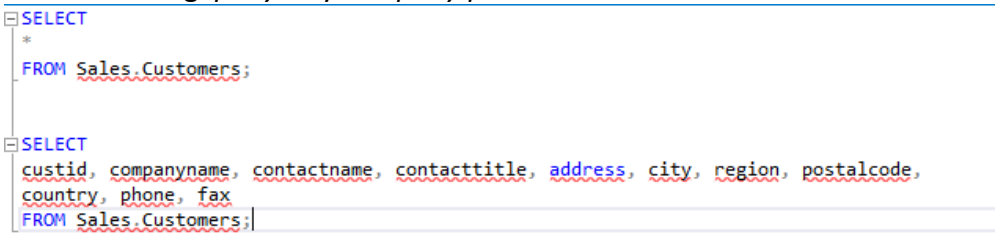
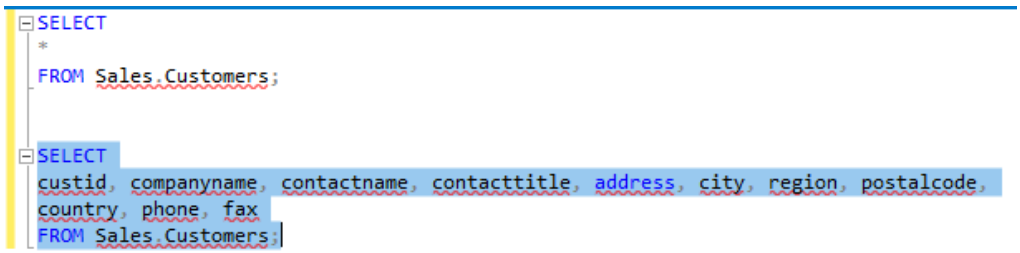
8

The results of the SQL command above are as follows

Results | Messages

	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	Higginbotham, Tom	Order Administrator	Berguvsvägen 5678	Luleå	NULL

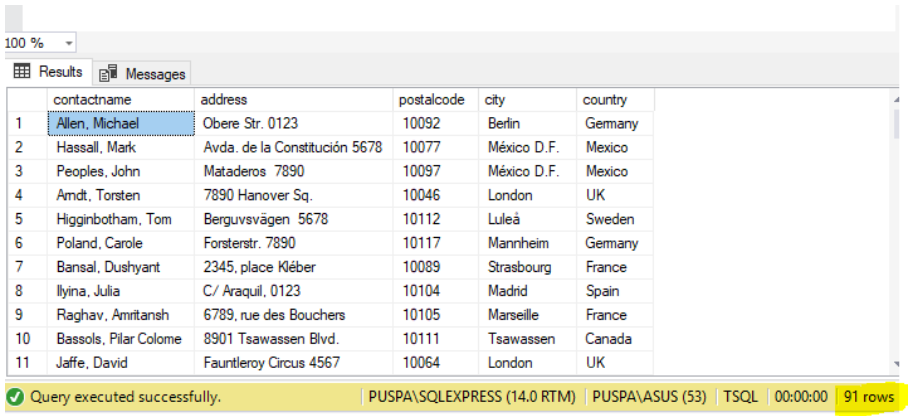
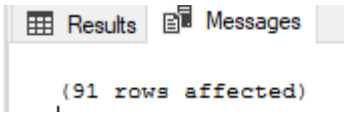
Practical – Part 1: Executing part or all of a SQL *script*

Step	Information
1	Please type the following *query in your query* panel then click *execute* . Note the results displayed.  SELECT * FROM Sales.Customers;
2	Please add the following *query to* your *query panel then click execute* . Note the results displayed  SELECT * FROM Sales.Customers; SELECT custid, companyname, contactname, contacttitle, address, city, region, postalcode, country, phone, fax FROM Sales.Customers;
3	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 * FROM Sales.Customers; SELECT custid, companyname, contactname, contacttitle, address, city, region, postalcode, country, phone, fax FROM Sales.Customers;
4	In *the query* panel please type SELECT * FROM
5	then on the Object Explorer tab – Tables please find the Sales.Customers table. Click the table and drag it to the query pane l . The result is as shown below, after that add a semicolon after the name of the table in question and click execute. SELECT * FROM [Sales].[Customers];

Practical – Part 2: Using the SELECT statement for specific columns

Step	Information

1	In the query panel, please type the script below
	`SELECT`
	`  contactname, address, postalcode, city, country`
	`FROM Sales.Customers;`

2	*Highlights query* above and click *execute*
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  Or you can also go to the messages tab as shown in the image below. 

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

Step	Information

1	In *the query* panel, please type *the script* below
	`SELECT`
	`  country`
	`FROM Sales.Customers;`

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*** **(Question 2)**

4	In *the query* pane, please type *the script* below.
	`SELECT DISTINCT`
	`  country`
	`FROM Sales.Customers;`
	Please click *execute* and observe the results.

5	***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)**

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

Step	Information

1	In *the query* panel, please type *the script* below
	`SELECT`
	`  c.contactname, c.contacttitle`
	`FROM Sales.Customers AS c;`

2	*Highlights query* above and click *execute.* Observe the results

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.
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)**

Practicum – Part 5: Use of CASE

Step	Information

1	In *the query* panel, please type *the script* below
	`SELECT`
	`  p.categoryid, p.productname`
	`FROM Production.Products AS p;`

2	*Highlights query* above and click *execute.* Observe the results

3	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`
	`FROM Production.Products AS p;`

4	*Highlights query* above and click *execute* . Observe the results.
5	***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*** **(Question 5)**
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 iscampaign
FROM Production.Products AS p;
```

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

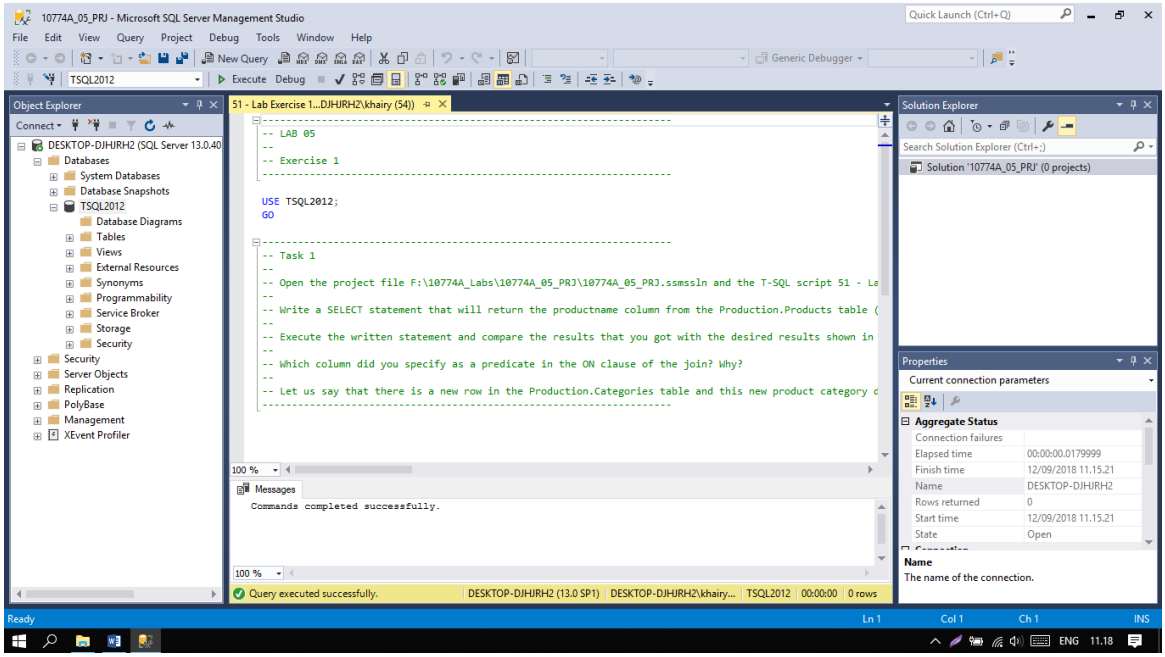
8 Based on question number 6, please display data that is in the 'seafood' category only and use the *ALIAS command* to change the column name as shown in the image below. **Capture your SQL command and how many *rows* are produced (Question 7)**

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

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

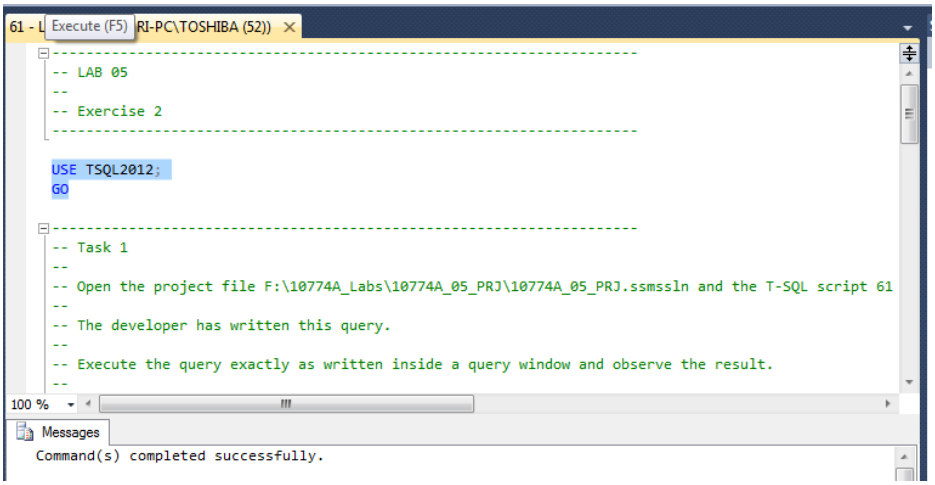
Practical – Part 6 : Creating an Inner Join Query

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** ".
	
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.
3	Compare the results in step 2 with the file 52 - Lab Exercise 1 - Task 1 Result.txt. If they are the same then the T-SQL you wrote is correct.
4	**[Question- 10]** Which column is specified as a predicate in the ON join clause? Why?
5	**Conclusion :** After carrying out this part of the practicum, students know and understand how to perform an INNER JOIN on two tables.

52 - Lab Exercise 1 - Task 1 Result.txt | 51 - Lab Exercise 1...RI-PC\TOSHIBA (52))

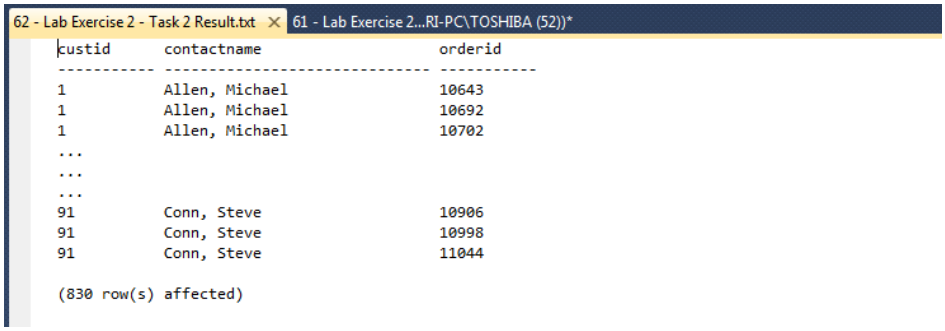
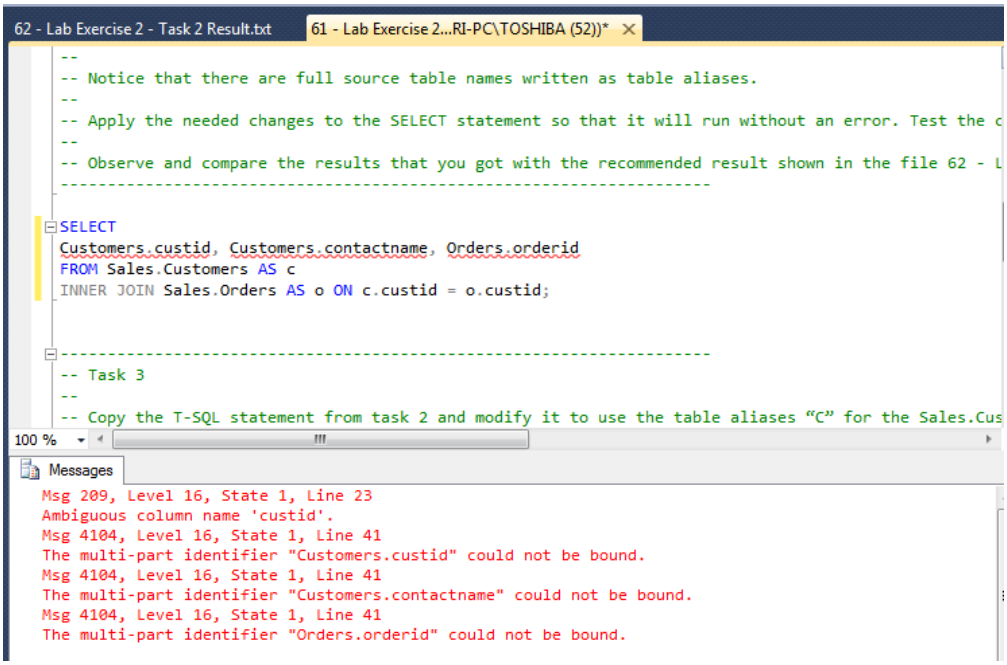
```
productname                              categoryname
---------------------------------------- ---------------
Product HHYDP                            Beverages
Product RECZE                            Beverages
Product IMEHJ                            Condiments
...
...
...
Product BWRLG                            Beverages
Product JYGFE                            Beverages
Product LUNZZ                            Condiments

(77 row(s) affected)
```

Practical – Part 7 : Creating an Inner Join Query on Multiple Tables

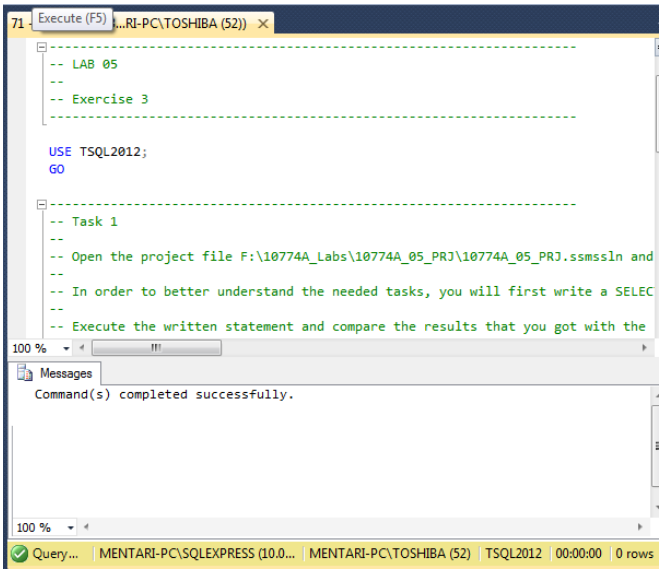
Step	Information
1	A *developer* will often be asked to run T-SQL files obtained from various departments . For example, the sales department wants a sales report of all customers for at least one order , with detailed information about each order. Then *the developer* will prepare the initialization 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. 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".
	

2	*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!

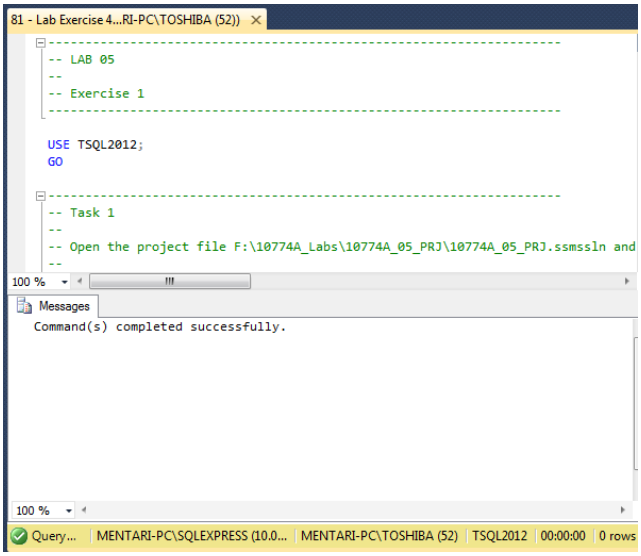
3	**[Question- 11]** After the 2nd stage of the experiment is carried out, an error will appear. What is the content of the error message? Why can this error occur? Explain!
4	**[Question- 12]** In this 4th trial, fix the error that occurred in the 3rd stage trial which explains that all table names have their own table identities.
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.

	
6	**[Question- 13]** Copy the T-SQL in the 4th stage of the test and modify it by using the alias table '' c '' to Sales.Custumers table and '' o '' for Sales.Orders table.
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	**[Question- 14]** Why does the execution result of T-SQL stage 8 produce an error?
10	**[Question- 15]** Change the column name prefix in the T-SQL test step 8 with its alias name, then display the execution results!
11	**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).

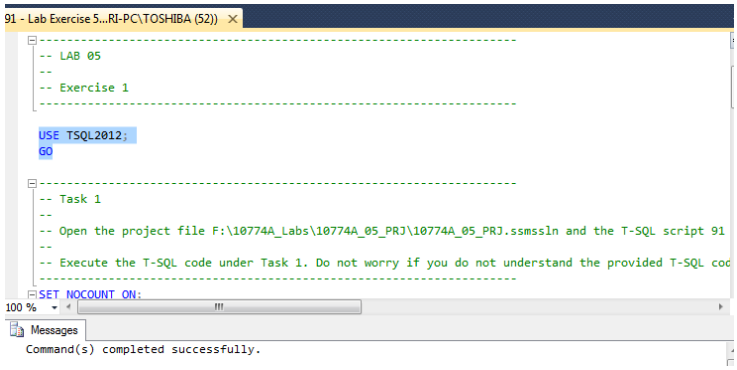
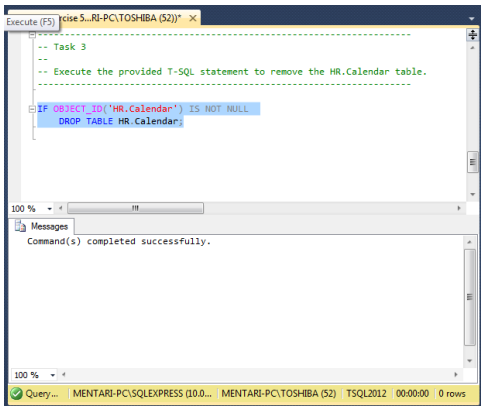
Practical – Part 8 : Creating a Self-Join Query

Step	Information
1	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".
	
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.
3	**[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.
4	**[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.
5	**[Question- 19]** 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.
6	**[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!
7	**Conclusion** : After doing this part of the practicum, you should understand how to write a T-SQL SELF-JOIN statement.

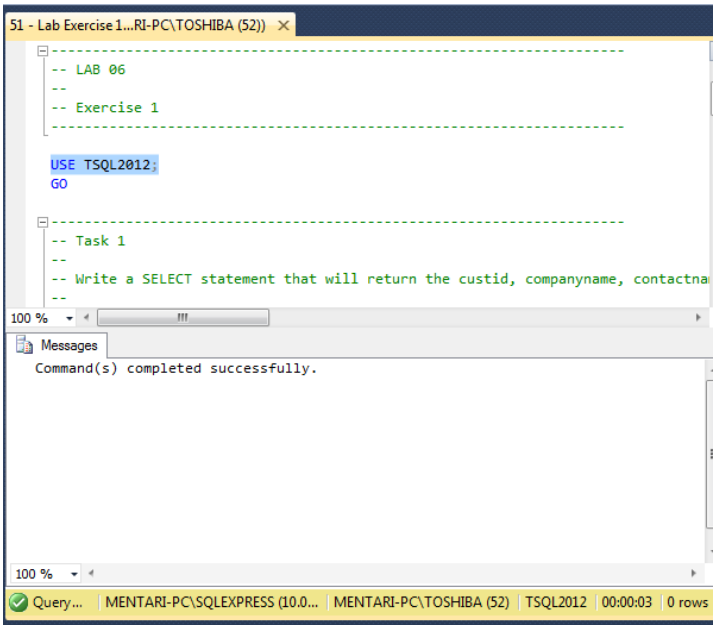
Practical – Part 9 : Creating Outer-Join Query

Step	Information
1	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. 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".
	
2	**[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 .
3	**[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.
4	**[Question- 23]** Pay attention to the values in the orderid column . Are there any missing values (NULL)? Why?
5	**Conclusion** : After doing this part of the practicum, you should understand how to write the T-SQL OUTER-JOIN statement .

Practical – Part 10 : Creating a Cross-Join Query

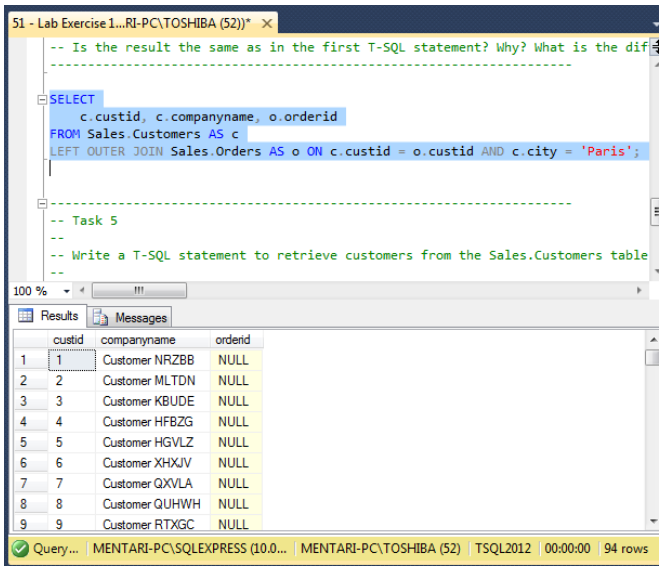
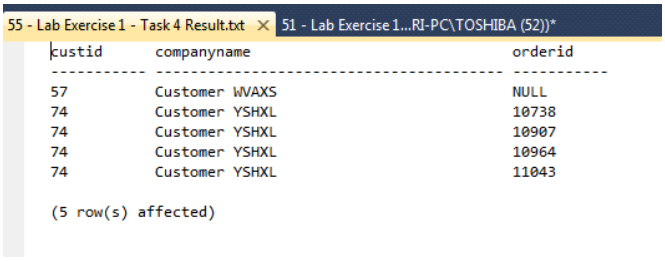
Step	Information
1	This case study begins with the HR department wanting to set up a personal calendar for each employee. The IT department will provide a T-SQL code that generates all days in the past year. Therefore, *the developer* will use the SELECT clause to return all rows from the calendar table 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".
	
2	**[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 CROSS-JOIN is implemented.)
3	**[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 .
4	**[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.
5	Drop the HR.Calendar table by executing the T-SQL code below task 3.
	
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

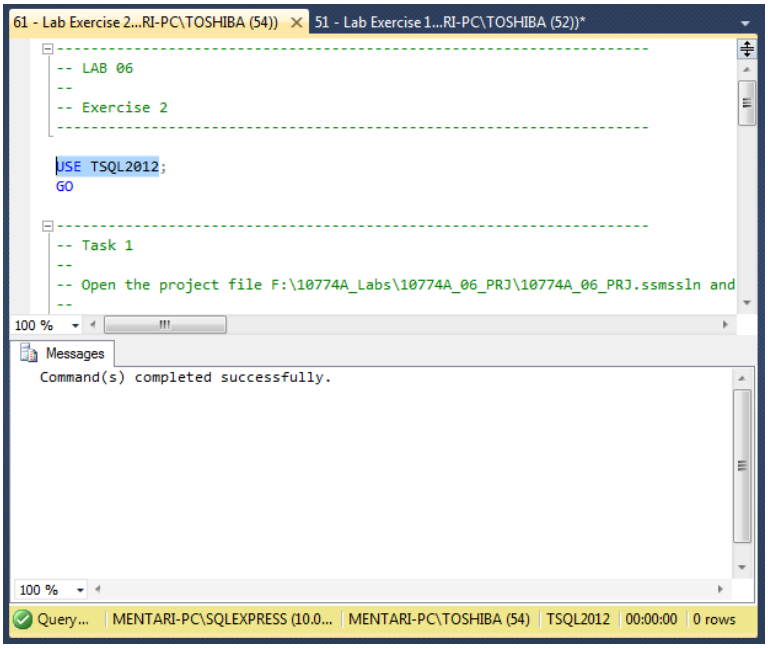
Step	Information
1	The scenario in this practicum uses the problems in the marketing department. The marketing department is working on several campaigns for old customers. The marketing staff needs a different customer list according to several business rules. Therefore, *the developer* will write a SELECT command to retrieve the desired rows from the Sales.Customers table. 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".
	
2	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.
3	**[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.

4	**[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).
5	**[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.

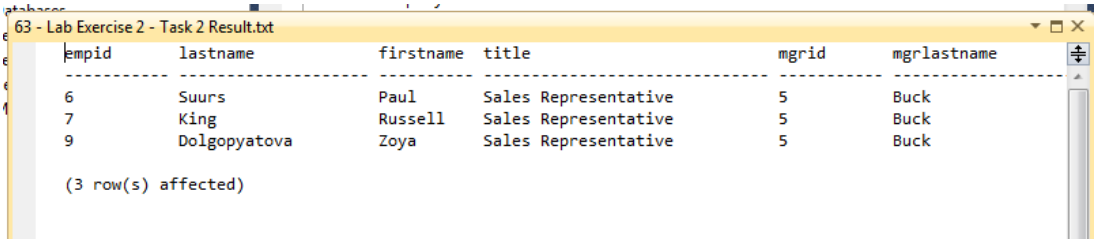
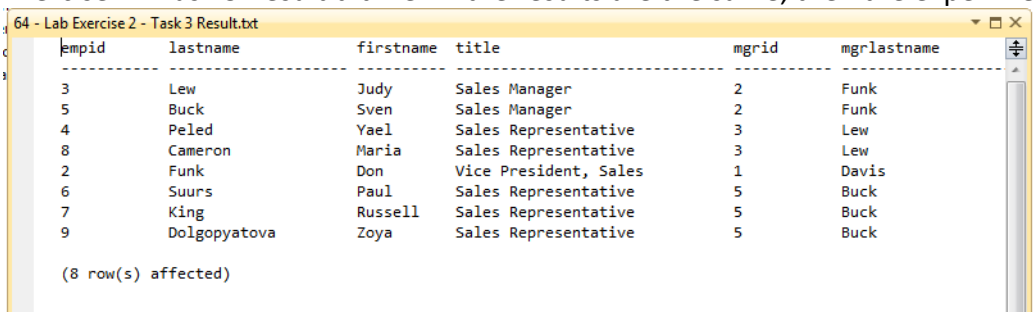
6	The IT department has written T-SQL code to return values in the custid, companyname columns in the Sales.Customers table and the orderid column. in the Sales.Orders table as below:
	`SELECT`
	`c . custid , c . companyname , o . orderid`
	`FROM Sales . Customers AS c`
	`LEFT OUTER JOIN Sales . Orders AS o ON c . custid = o . custid AND c . city = 'Paris' ;`

7	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". 
8	**[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!
9	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. 
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	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. 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".
	
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!
3	**[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 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?
4	**[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. 
5	**[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. 
6	**[Question-3 6]** Why can we use column names according to the original table name or use table alias names?
7	**Conclusion** : After working on the practical work and questions in this section, you should now understand how to use the ORDER BY clause .

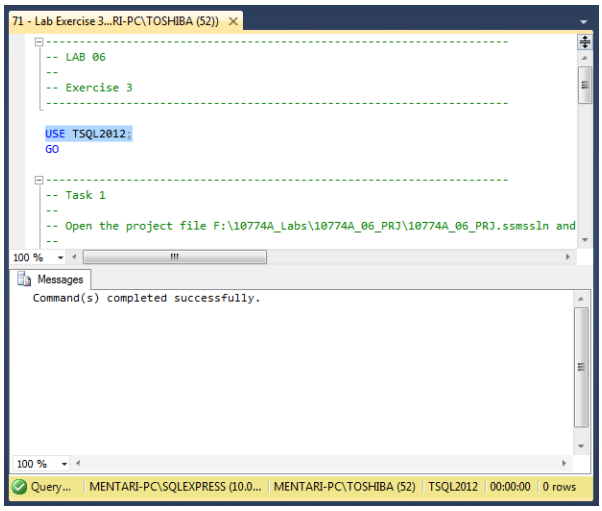
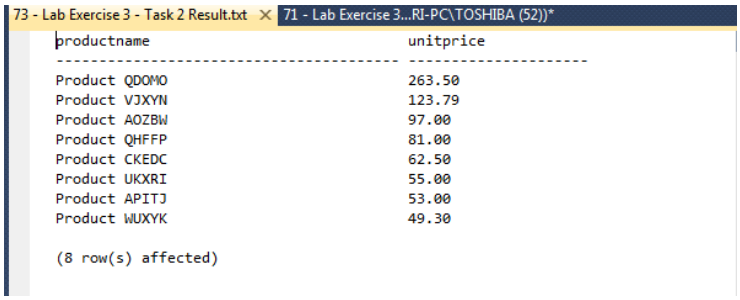
63 - Lab Exercise 2 - Task 2 Result.txt

```
empid       lastname             firstname  title                         mgrid       mgrlastname
----------- -------------------- ---------- ----------------------------- ----------- --------------------
6           Suurs                Paul       Sales Representative          5           Buck
7           King                 Russell    Sales Representative          5           Buck
9           Dolgopyatova         Zoya       Sales Representative          5           Buck

(3 row(s) affected)
```

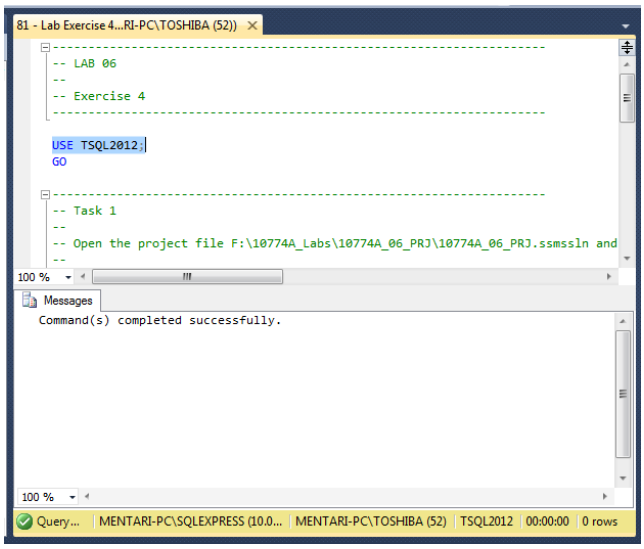
64 - Lab Exercise 2 - Task 3 Result.txt

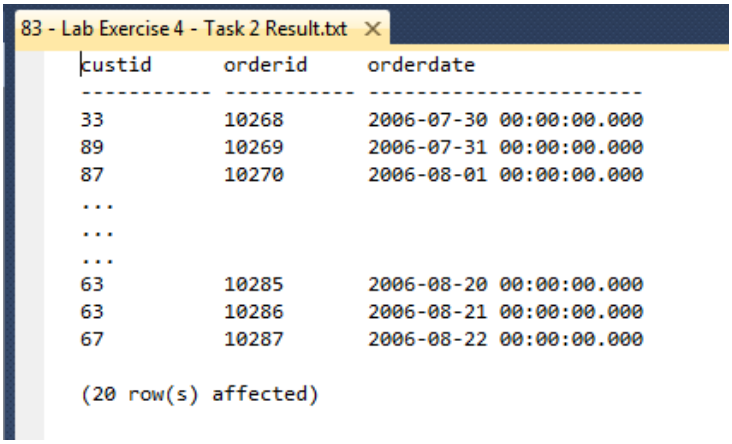
```
empid       lastname             firstname  title                         mgrid       mgrlastname
----------- -------------------- ---------- ----------------------------- ----------- --------------------
3           Lew                  Judy       Sales Manager                 2           Funk
5           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)
```

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

Step	Information
1	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. 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".
	
2	**[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!
3	**[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. 
4	**[Question- 39]** Is it possible to implement the 5 trial T-SQL command using the OFFSET-FETCH clause?
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	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. 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".
	
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. 82 - Lab Exercise 4 - Task 1 Result.txt | 81 - Lab Exercise 4...RI-PC\TOSHIBA (52)) 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)
3	**[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.

	
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 -