JOBSHEET
PRAKTIKUM BASIS DATA LANJUT

Jurusan Teknologi Informasi
POLITEKNIK NEGERI MALANG

WEEK 3

SQL SERVER – DATA TYPE, FUNCTIONS, & TABLE EXPRESSION

Information Technology Department, Malang State Polytechnic

Jobsheet- 3 : Data Types and Functions in Data Types

Advanced Database Course

Supervisor: Advanced Database Teaching Team

SAFRIZAL RAHMAN_19_SIB_2G

Topics

1. Data Type
2. Functions on Data Types

Objective

Students are expected to be able to:

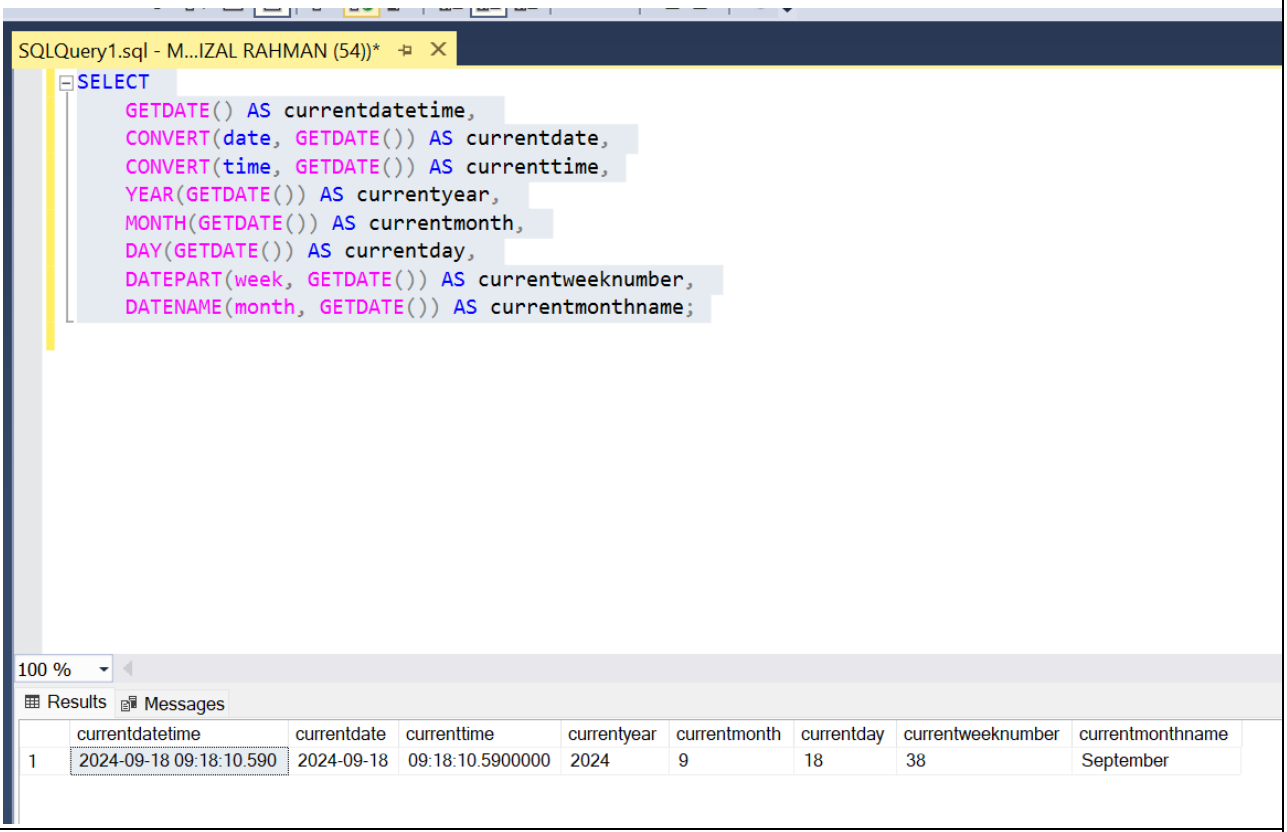
1. Understanding how to perform date & time queries
2. Understanding how to use date & time functions
3. Understanding how to combine character data
4. Understanding how to use character functions

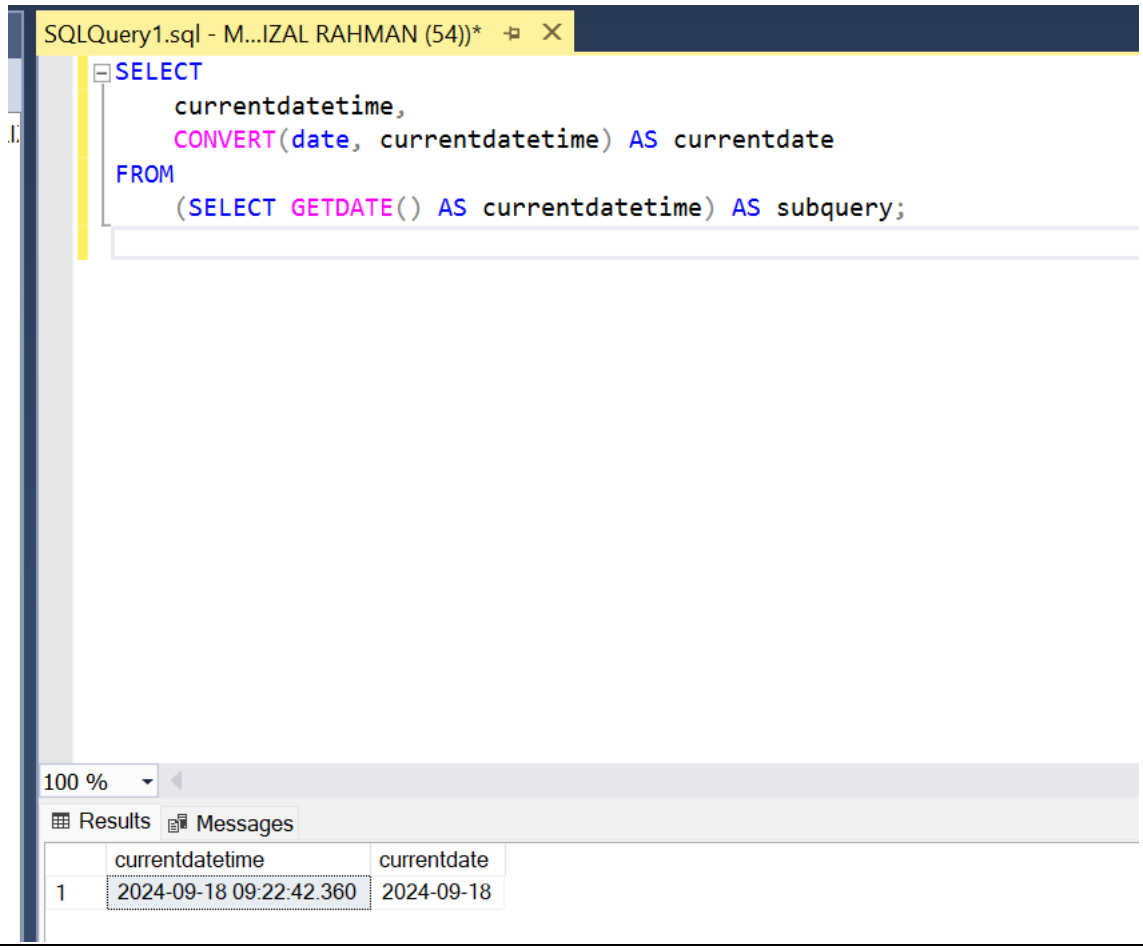
General Instructions

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 2, 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 ○ **Example** : BDL_TI2Z_03_Bang Mudrik.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 – Part 1: Writing a SELECT query to get the current date and time

Step	Information
1	[Question- 1] Write a SELECT query to display the columns containing: a. Current date and time, name the alias *currentdatetime* b. Just the current date, name the alias *currentdate* c. Just the current time (HH:mm:ss), name it alias *current time* d. This year only, name it alias *currentyear* e. Just this month number (number), give it an alias name *currentmonth* f. Only the day number in this month , give it an alias name *currentday* g. Just the number of the nth week *of* the year, give it the alias *currentweeknumber* SELECT GETDATE() AS currentdatetime, CONVERT(date, GETDATE()) AS currentdate, CONVERT(time, GETDATE()) AS currenttime, YEAR(GETDATE()) AS currentyear, MONTH(GETDATE()) AS currentmonth, DAY(GETDATE()) AS currentday, DATEPART(week, GETDATE()) AS currentweeknumber, DATENAME(month, GETDATE()) AS currentmonthname;
	h.Current month name, give alias *currentmonthname* Execute the query, and *screenshot* the results. SQLQuery1.sql - M...IZAL RAHMAN (54))* SELECT GETDATE() AS currentdatetime, CONVERT(date, GETDATE()) AS currentdate, CONVERT(time, GETDATE()) AS currenttime, YEAR(GETDATE()) AS currentyear, MONTH(GETDATE()) AS currentmonth, DAY(GETDATE()) AS currentday, DATEPART(week, GETDATE()) AS currentweeknumber, DATENAME(month, GETDATE()) AS currentmonthname; SELECT GETDATE() AS currentdatetime, CONVERT(date, GETDATE()) AS currentdate, CONVERT(time, GETDATE()) AS currenttime, YEAR(GETDATE()) AS currentyear, MONTH(GETDATE()) AS currentmonth, DAY(GETDATE()) AS currentday, DATEPART(week, GETDATE()) AS currentweeknumber, DATENAME(month, GETDATE()) AS currentmonthname;

3	Compare the results of executing the query in step 2 above with the results in the following image: Results Messages currentdatetime | currentdate | currenttime | currentyear | currentmonth | currentday | currentweeknumber | currentmonthname 1 | 2022-09-11 03:37:18.070 | 2022-09-11 | 03:37:18.0700000 | 2022 | 9 | 11 | 38 | September The values obtained will of course be different because they depend on when the query is executed. 
4	[Question- 2] Can the *currentdatetime alias* be used in [Question-1-b] to replace the *currentdate alias* ? Explain! In SQL, an alias defined in the SELECT clause for a column cannot directly be reused within the same SELECT clause to define or replace another column. However, aliases can be used in subsequent parts of the query, such as in the WHERE, GROUP BY, or HAVING clauses. This means that the alias currentdatetime created for one of the columns in your initial query cannot directly replace currentdate in the same SELECT list.

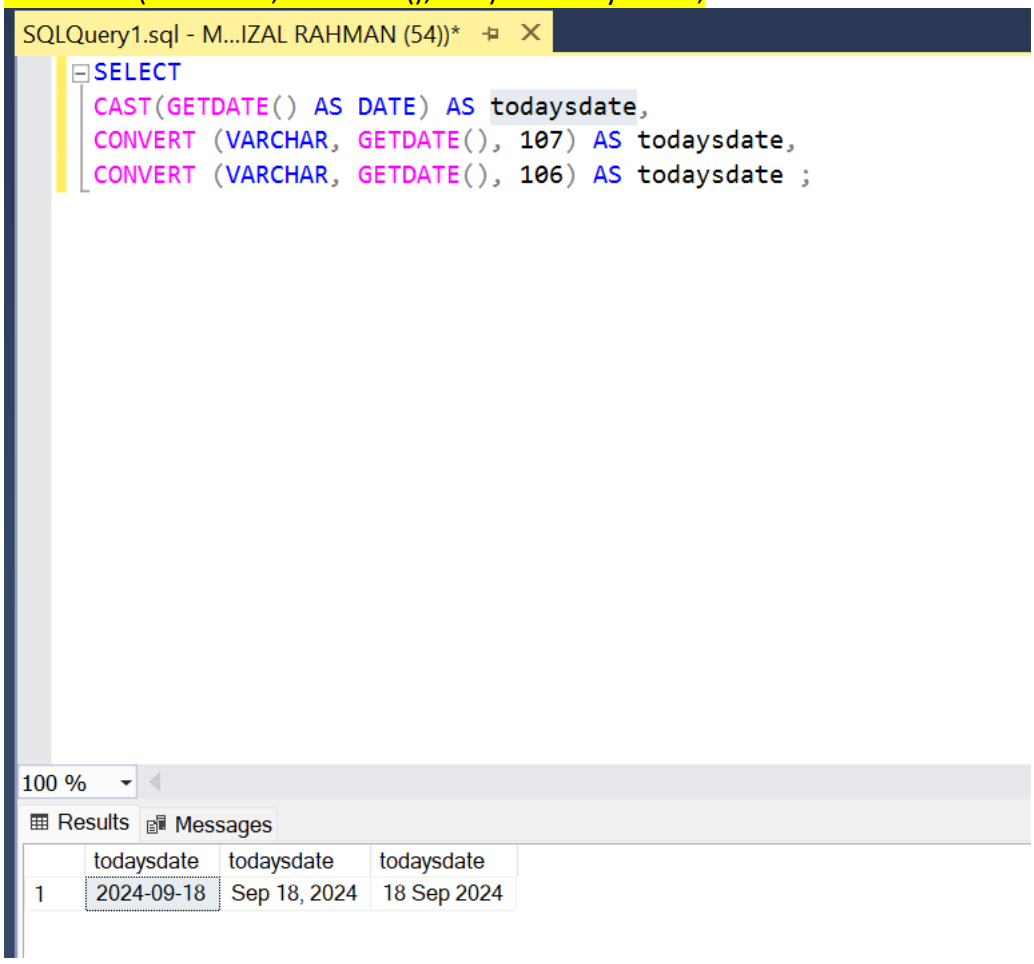


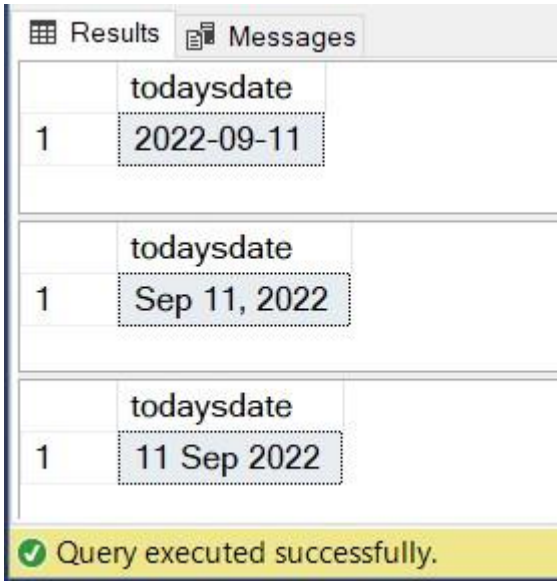
Practical – Part 2: Writing a SELECT query to get *date data type*

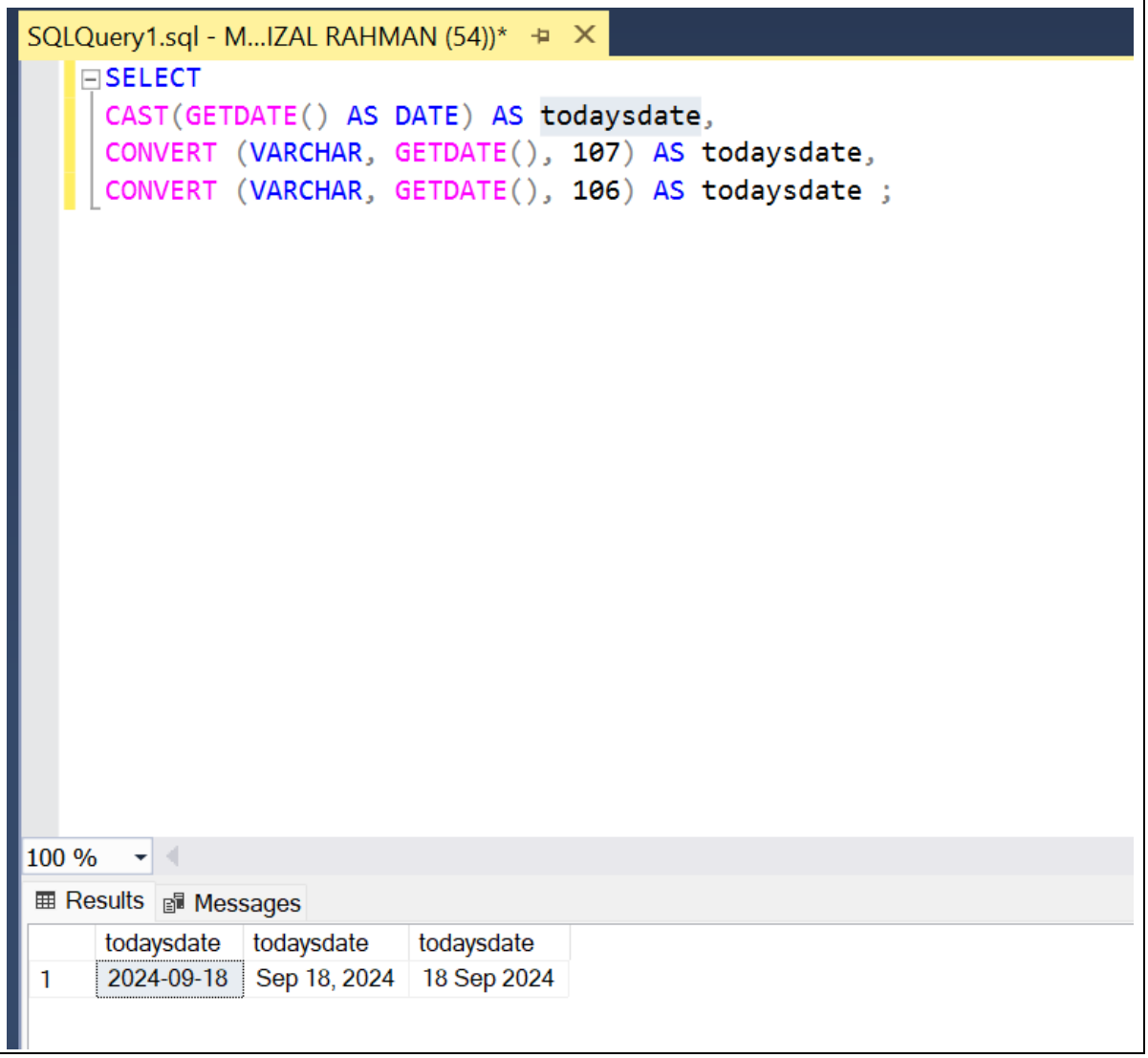
Step	Information

1	[Question- 3] Write a SELECT query using several different T-SQL functions (`CAST` , `CONVERT` , other specific functions, etc.) to display today's date . Name it *todaysdate* as an alias for the column name.

```
SELECT
CAST(GETDATE() AS DATE) AS todaysdate,
CONVERT (VARCHAR, GETDATE(), 107) AS todaysdate,
CONVERT (VARCHAR, GETDATE(), 106) AS todaysdate ;
```



	 Example query results:



Practical – Part 3: Writing SELECT queries that use several *date* and *time functions*

Step	Information
1	[Question- 4] Write a SELECT query that returns several columns containing: a. Date and time 5 months from now. Name the alias *fivemonths* . b. The number of days between the current date and the first column (*fivemonths* in point a above). Name the alias *diffdays* . c. The number of weeks between August 17, 1945 and August 17, 2022. Use the alias *diffweeks* . d. The first day of the month based on the current date and time. Use the alias *firstday* .

```
SELECT
   DATEADD(month, 5, GETDATE()) AS fivemonths,
   DATEDIFF(day, GETDATE(), DATEADD(month, 5, GETDATE())) AS diffdays,
   DATEDIFF(week, '1945-08-17', '2022-08-17') AS diffweeks,
   DATEADD(month, DATEDIFF(month, 0, GETDATE()), 0) AS firstday;
```

2 Execute the query above , and *screenshot* the results. Compare the results obtained with the following results:

Results | Messages

	fivemonths	diffdays	diffweeks	firstday
1	2023-02-11 03:47:14.853	91	4018	2022-09-01 00:00:00.000

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

```
SELECT
    DATEADD(month, 5, GETDATE()) AS fivemonths,
    DATEDIFF(day, GETDATE(), DATEADD(month, 5, GETDATE())) AS diffdays,
    DATEDIFF(week, '1945-08-17', '2022-08-17') AS diffweeks,
    DATEADD(month, DATEDIFF(month, 0, GETDATE()), 0) AS firstday;
```

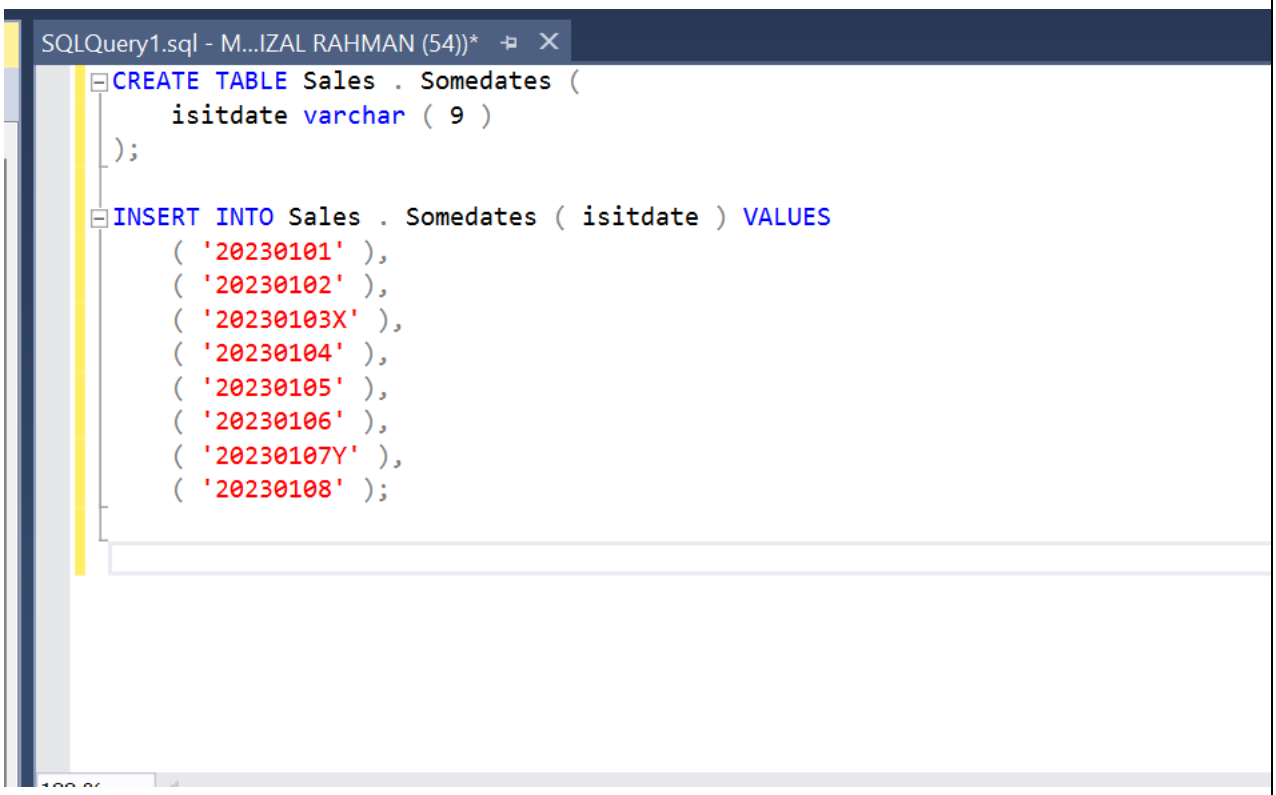
100 %

Results | Messages

	fivemonths	diffdays	diffweeks	firstday
1	2025-02-18 09:32:47.833	153	4018	2024-09-01 00:00:00.000

Lab – Part 4: Observation on Sales.Somedates table

Step	Information

1	Write a T-SQL query to create a table named *Sales.Somedates* with the following contents, then execute it.
	
	`CREATE TABLE Sales . Somedates (`
	`  isitdate varchar ( 9 )`
	`);`
	
	`INSERT INTO Sales . Somedates ( isitdate ) VALUES`
	`  ( '20230101' ),`
	`  ( '20230102' ),`
	`  ( '20230103X' ),`
	`  ( '20230104' ),`
	`  ( '20230105' ),`
	`  ( '20230106' ),`
	`  ( '20230107Y' ),`
	`  ( '20230108' );`
	
	 
	
	`Completion time: 2024-09-18T09:35:03.1694447+07:00`

2

[Question- 5] Write a T-SQL query to get a column named *isitdate* in the *Sales.Somedates table* . Then create a new column named *converteddate of the date* data type based on *the isitdate* column . If the data in the *isitdate column* cannot be converted to the *date data type , return* NULL.

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

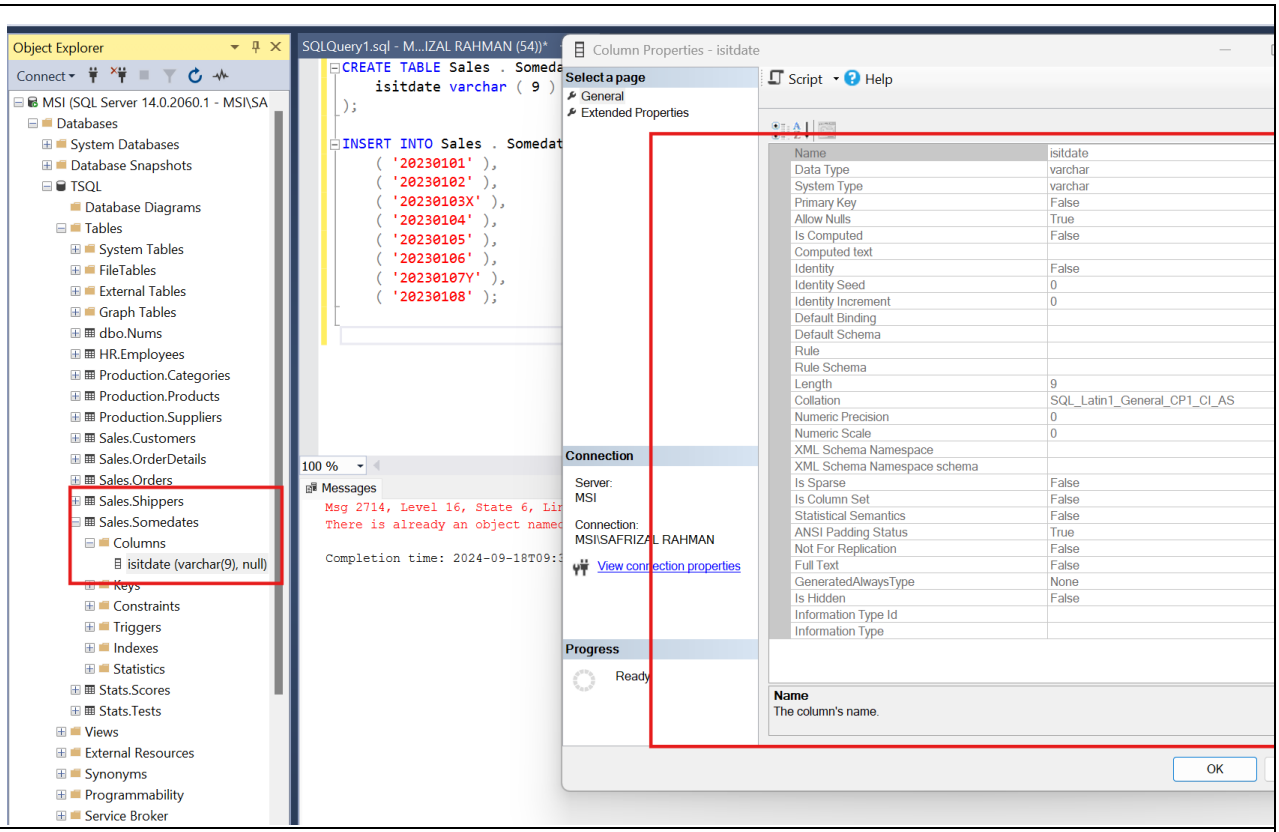
```
SELECT
    isitdate,
    CASE
        WHEN ISDATE(isitdate) = 1 THEN CAST(isitdate AS date)
        ELSE NULL
    END AS converteddate
FROM
    Sales.Somedates;
```

100 %

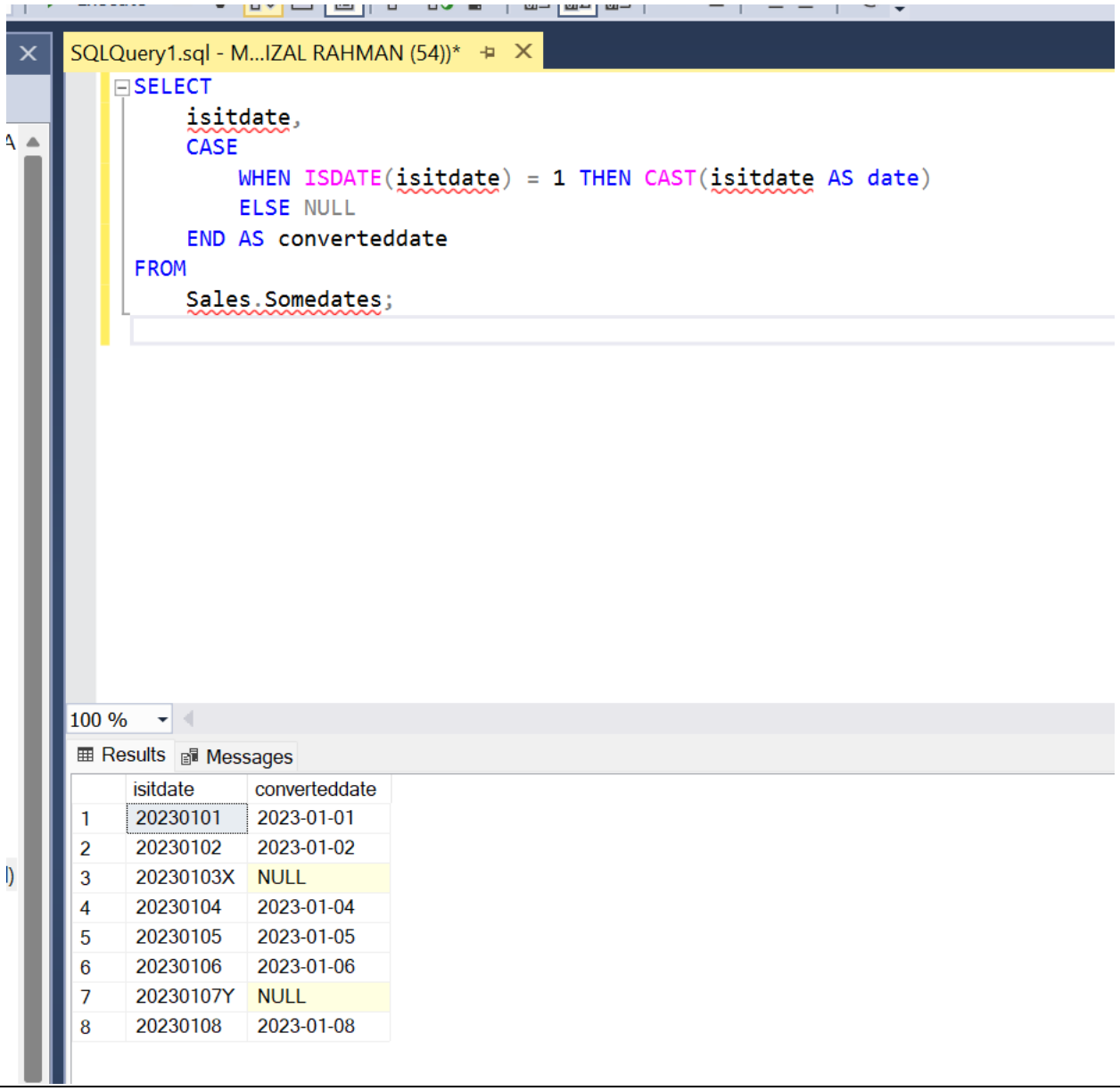
Results | Messages

	isitdate	converteddate
1	20230101	2023-01-01
2	20230102	2023-01-02
3	20230103X	NULL
4	20230104	2023-01-04
5	20230105	2023-01-05
6	20230106	2023-01-06
7	20230107Y	NULL
8	20230108	2023-01-08

```
Completion time: 2024-09-18T09:35:03.1694447+07:00
```

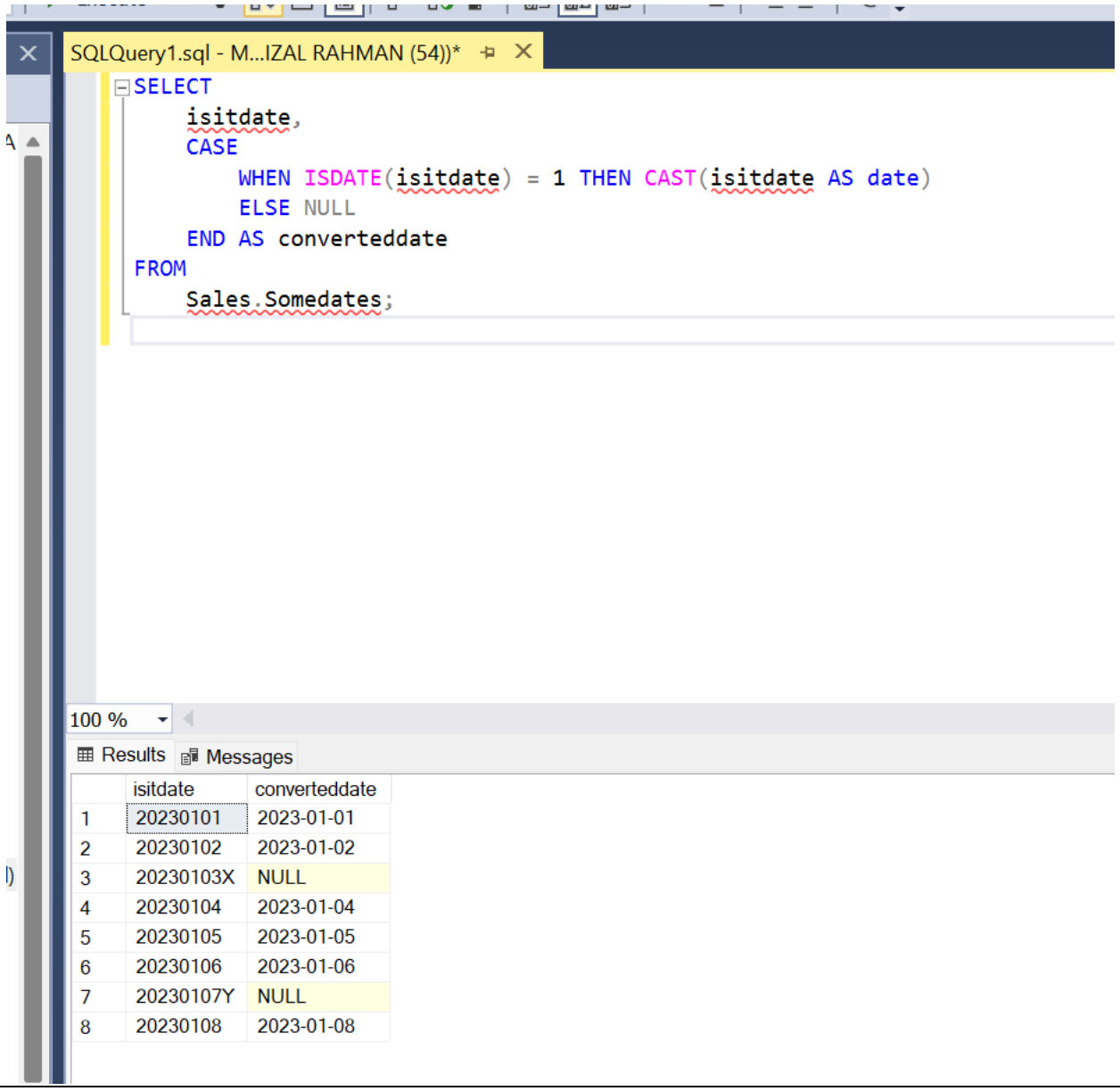


3	Execute step 2 above, and *screenshot* the result.



	isitdate	converteddate
1	20230101	2023-01-01
2	20230102	2023-01-02
3	20230103X	NULL
4	20230104	2023-01-04
5	20230105	2023-01-05
6	20230106	2023-01-06
7	20230107Y	NULL
8	20230108	2023-01-08

4	[Question- 6] What is the difference between `the SYSDATETIME` and `CURRENT_TIMESTAMP functions` ? Show the difference in the results of the two functions. The SYSDATETIME function returns the current date and time with more precision (including fractional seconds), while CURRENT_TIMESTAMP returns the current date and time with less precision. Here is a query to show the difference:

	 isitdate converteddate 1 20230101 2023-01-01 2 20230102 2023-01-02 3 20230103X NULL 4 20230104 2023-01-04 5 20230105 2023-01-05 6 20230106 2023-01-06 7 20230107Y NULL 8 20230108 2023-01-08
5	[Question- 7] What is the general format of `the DATE type` ? The general format of the DATE type in SQL Server is YYYY-MM-DD. This format is used to store date values without time components.
6	**Conclusion** : After testing this section, students will be able to know how to display the date and time with T-SQL.

Practical – Part 5: Writing Queries Using *Date* and *Time Functions*

Step	Information

1	**Scenario** : The Sales Department wants sales reports in different time periods. The Sales staff wants to analyze sales data based on customers, products, and orders made at the end of the month. To be able to create the report, you as the DB Admin must write a SELECT query using various *date* and *time functions* .
2	[Question- 8] Write a SELECT query to get unique data in the *custid, shipname, shipdate columns* in the *Sales.Orders table* . Filter the results to only display orders in March 2008. SELECT DISTINCT custid, shipname, shippeddate FROM Sales.Orders WHERE shippeddate >= '2008-03-01' AND shippeddate < '2008-04-01';

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

```
SELECT DISTINCT
    custid,
    shipname,
    shippeddate
FROM
    Sales.Orders
WHERE
    shippeddate >= '2008-03-01' AND shippeddate < '2008-04-01';
```

100 %

Results | Messages

	custid	shipname	shippeddate
1	1	Destination LOUIE	2008-03-24 00:00:00.000
2	2	Destination RAIGI	2008-03-11 00:00:00.000
3	4	Ship to 4-A	2008-03-09 00:00:00.000
4	4	Ship to 4-B	2008-03-25 00:00:00.000
5	5	Ship to 5-A	2008-03-03 00:00:00.000
6	6	Ship to 6-B	2008-03-20 00:00:00.000
7	9	Ship to 9-B	2008-03-24 00:00:00.000
8	9	Ship to 9-C	2008-03-23 00:00:00.000
9	10	Destination OLSSJ	2008-03-11 00:00:00.000
10	10	Destination OLSSJ	2008-03-17 00:00:00.000
11	10	Destination OLSSJ	2008-03-27 00:00:00.000
12	10	Destination XJIBQ	2008-03-13 00:00:00.000
13	11	Destination NZASL	2008-03-16 00:00:00.000
14	11	Destination NZASL	2008-03-19 00:00:00.000
15	12	Destination QTHBC	2008-03-13 00:00:00.000
16	15	Destination EVHYA	2008-03-30 00:00:00.000
17	20	Destination CUVPF	2008-03-31 00:00:00.000
18	24	Destination NCKKO	2008-03-03 00:00:00.000
19	24	Destination YCMPK	2008-03-20 00:00:00.000
20	25	Destination QOCBL	2008-03-12 00:00:00.000
21	28	Destination CIRQO	2008-03-26 00:00:00.000
22	29	Destination WOF...	2008-03-18 00:00:00.000
23	30	Destination GGQIR	2008-03-19 00:00:00.000
24	30	Destination IIYDD	2008-03-05 00:00:00.000
25	31	Destination GWP...	2008-03-23 00:00:00.000
26	32	Destination AVQUS	2008-03-18 00:00:00.000
27	34	Destination DPCVR	2008-03-04 00:00:00.000
28	34	Destination DPCVR	2008-03-05 00:00:00.000
29	34	Destination JPAIY	2008-03-13 00:00:00.000
30	34	Destination SCQXA	2008-03-02 00:00:00.000

3	Execute step 2 above, and *screenshot* the result. Compare it with the result in the following image: 

Results | Messages

	custid	shipname	shippeddate
1	1	Destination LOUIE	2008-03-24 00:00:00.000
2	2	Destination RAIGI	2008-03-11 00:00:00.000
3	4	Ship to 4-A	2008-03-09 00:00:00.000
4	4	Ship to 4-B	2008-03-25 00:00:00.000
5	5	Ship to 5-A	2008-03-03 00:00:00.000
6	6	Ship to 6-B	2008-03-20 00:00:00.000
7	9	Ship to 9-B	2008-03-24 00:00:00.000
8	9	Ship to 9-C	2008-03-23 00:00:00.000

DESKTOP-EIPTP8V (15.0 RTM) | DESKTOP-EIPTP8V\milyu ... | TSQL | 00:00:00 | 67 rows

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

```
SELECT DISTINCT
    custid,
    shipname,
    shippeddate
FROM
    Sales.Orders
WHERE
    shippeddate >= '2008-03-01' AND shippeddate < '2008-04-01';
```

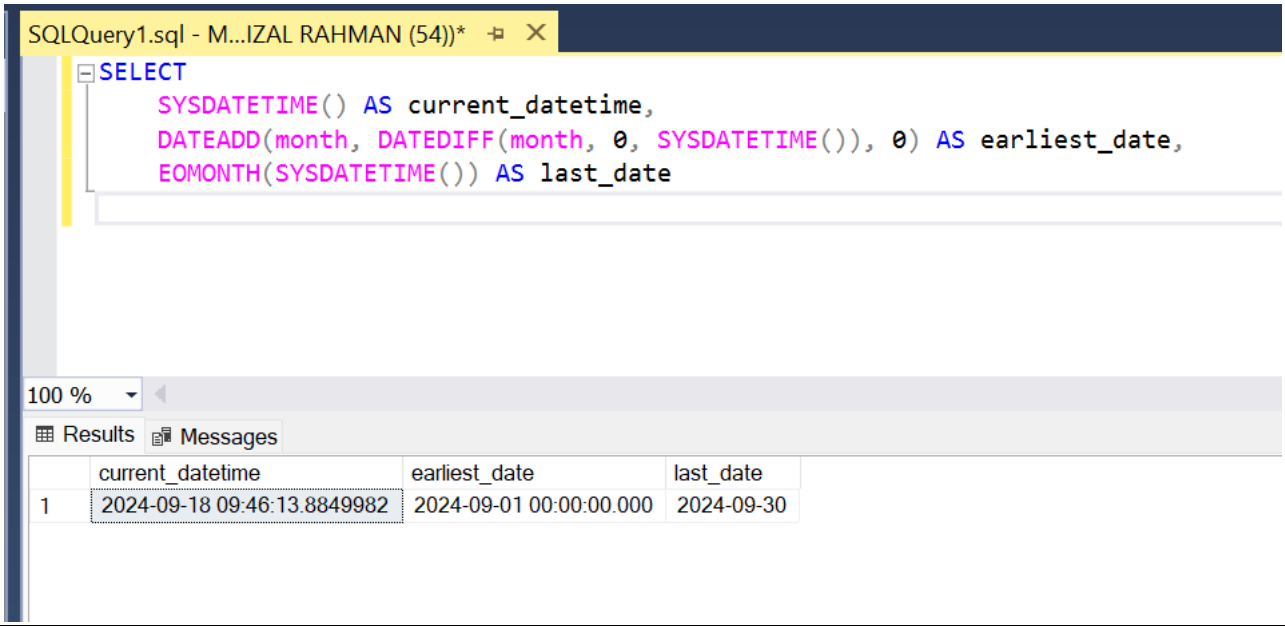
100 %

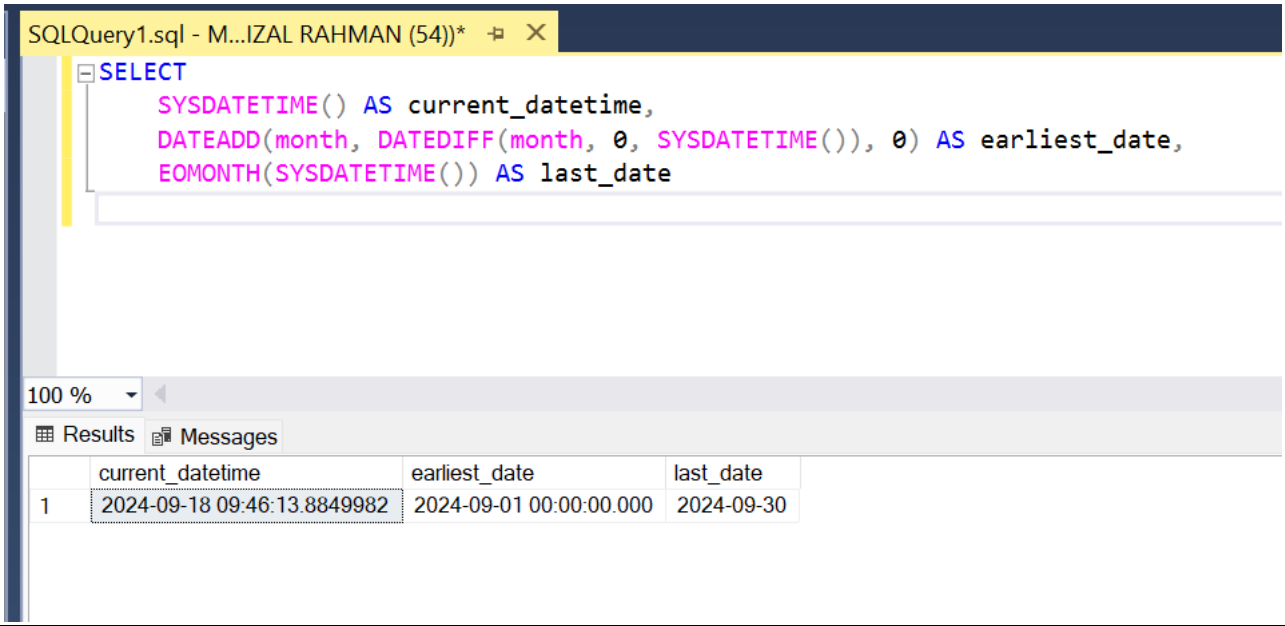
Results | Messages

	custid	shipname	shippeddate
1	1	Destination LOUIE	2008-03-24 00:00:00.000
2	2	Destination RAIGI	2008-03-11 00:00:00.000
3	4	Ship to 4-A	2008-03-09 00:00:00.000
4	4	Ship to 4-B	2008-03-25 00:00:00.000
5	5	Ship to 5-A	2008-03-03 00:00:00.000
6	6	Ship to 6-B	2008-03-20 00:00:00.000
7	9	Ship to 9-B	2008-03-24 00:00:00.000
8	9	Ship to 9-C	2008-03-23 00:00:00.000
9	10	Destination OLSSJ	2008-03-11 00:00:00.000
10	10	Destination OLSSJ	2008-03-17 00:00:00.000
11	10	Destination OLSSJ	2008-03-27 00:00:00.000
12	10	Destination XJIBQ	2008-03-13 00:00:00.000
13	11	Destination NZASL	2008-03-16 00:00:00.000
14	11	Destination NZASL	2008-03-19 00:00:00.000
15	12	Destination QTHBC	2008-03-13 00:00:00.000
16	15	Destination EVHYA	2008-03-30 00:00:00.000
17	20	Destination CUVPF	2008-03-31 00:00:00.000
18	24	Destination NCKKO	2008-03-03 00:00:00.000
19	24	Destination YCMPK	2008-03-20 00:00:00.000
20	25	Destination QOCBL	2008-03-12 00:00:00.000
21	28	Destination CIRQO	2008-03-26 00:00:00.000
22	29	Destination WOF...	2008-03-18 00:00:00.000
23	30	Destination GGQIR	2008-03-19 00:00:00.000
24	30	Destination IIYDD	2008-03-05 00:00:00.000
25	31	Destination GWP...	2008-03-23 00:00:00.000
26	32	Destination AVQUS	2008-03-18 00:00:00.000
27	34	Destination DPCVR	2008-03-04 00:00:00.000
28	34	Destination DPCVR	2008-03-05 00:00:00.000
29	34	Destination JPAIY	2008-03-13 00:00:00.000
30	34	Destination SCQXA	2008-03-02 00:00:00.000

Lab – Part 6 : Writing Queries SELECT to calculate the first and last day in 1 month

Step	Information
1	[Question-9] Write a SELECT query displaying the following 3 columns: a. Date and time when you worked on this jobsheet b. **The earliest** date of the month when you worked on this jobsheet c. **last** date of the month when you worked on this jobsheet
2	[Question-10] Execute step 1 above and screenshot the results. What can you conclude from this experiment?





Practical – Part 7: Writing a SELECT query to generate order data for the last 5 days in 1 month

Step	Information
1	[Question- 11] Write a SELECT query to display the *orderid* , *custid* , *orderdate* , and *shipaddress columns* from the *Sales.Orders table* . Filter the results to only display orders from the last 5 days in a month.

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

```
SELECT
    orderid,
    custid,
    orderdate,
    shipaddress
FROM
    Sales.Orders
WHERE
    DAY(orderdate) > DAY(EOMONTH(orderdate)) - 5;
```

100 %

Results | Messages

	orderid	custid	orderdate	shipaddress
1	10267	25	2006-07-29 00:00:00.000	Berliner Platz 0123
2	10268	33	2006-07-30 00:00:00.000	5ª Ave. Los Palos Grandes 5678
3	10269	89	2006-07-31 00:00:00.000	8901 - 12th Ave. S.
4	10290	15	2006-08-27 00:00:00.000	Av. dos Lusíadas, 4567
5	10291	61	2006-08-27 00:00:00.000	Rua da Panificadora, 5678
6	10292	81	2006-08-28 00:00:00.000	Av. Inês de Castro, 6789
7	10293	80	2006-08-29 00:00:00.000	Avda. Azteca 4567
8	10294	65	2006-08-30 00:00:00.000	7890 Milton Dr.
9	10315	38	2006-09-26 00:00:00.000	Garden House Crowther Way 9012
10	10316	65	2006-09-27 00:00:00.000	8901 Milton Dr.
11	10317	48	2006-09-30 00:00:00.000	6789 Chiaroscuro Rd.
12	10339	51	2006-10-28 00:00:00.000	8901 rue St. Laurent
13	10340	9	2006-10-29 00:00:00.000	8901, rue des Bouchers
14	10341	73	2006-10-29 00:00:00.000	Vinbæltet 1234
15	10342	25	2006-10-30 00:00:00.000	Berliner Platz 0123
16	10343	44	2006-10-31 00:00:00.000	Magazinweg 4567
17	10363	17	2006-11-26 00:00:00.000	Walserweg 7890
18	10364	19	2006-11-26 00:00:00.000	3456 King George
19	10365	3	2006-11-27 00:00:00.000	Mataderos 3456
20	10366	29	2006-11-28 00:00:00.000	Rambla de Cataluña, 0123
21	10367	83	2006-11-28 00:00:00.000	Smagsloget 1234
22	10368	20	2006-11-29 00:00:00.000	Kirchgasse 9012
23	10396	25	2006-12-27 00:00:00.000	Berliner Platz 0123
24	10397	60	2006-12-27 00:00:00.000	Estrada da saúde n. 2345
25	10398	71	2006-12-30 00:00:00.000	9012 Suffolk Ln.
26	10399	83	2006-12-31 00:00:00.000	Smagsloget 2345

Query executed successfully.

2	Execute step 1 above and screenshot the result. Compare it with the result in the following image:

Results | Messages

	orderid	custid	orderdate	shipaddress
1	10267	25	2006-07-29 00:00:00.000	Berliner Platz 0123
2	10268	33	2006-07-30 00:00:00.000	5ª Ave. Los Palos Grandes 5678
3	10269	89	2006-07-31 00:00:00.000	8901 - 12th Ave. S.
4	10290	15	2006-08-27 00:00:00.000	Av. dos Lusíadas, 4567
5	10291	61	2006-08-27 00:00:00.000	Rua da Panificadora, 5678
6	10292	81	2006-08-28 00:00:00.000	Av. Inês de Castro, 6789
7	10293	80	2006-08-29 00:00:00.000	Avda. Azteca 4567
8	10294	65	2006-08-30 00:00:00.000	7890 Milton Dr.
9	10315	38	2006-09-26 00:00:00.000	Garden House Crowther Way 9012

Query exec... | DESKTOP-EIPTP8V (15.0 RTM) | DESKTOP-EIPTP8V\milyu ... | TSQL | 00:00:00 | 140 rows

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

```
SELECT
    orderid,
    custid,
    orderdate,
    shipaddress
FROM
    Sales.Orders
WHERE
    DAY(orderdate) > DAY(EOMONTH(orderdate)) - 5;
```

100 %

Results | Messages

	orderid	custid	orderdate	shipaddress
1	10267	25	2006-07-29 00:00:00.000	Berliner Platz 0123
2	10268	33	2006-07-30 00:00:00.000	5ª Ave. Los Palos Grandes 5678
3	10269	89	2006-07-31 00:00:00.000	8901 - 12th Ave. S.
4	10290	15	2006-08-27 00:00:00.000	Av. dos Lusíadas, 4567
5	10291	61	2006-08-27 00:00:00.000	Rua da Panificadora, 5678
6	10292	81	2006-08-28 00:00:00.000	Av. Inês de Castro, 6789
7	10293	80	2006-08-29 00:00:00.000	Avda. Azteca 4567
8	10294	65	2006-08-30 00:00:00.000	7890 Milton Dr.
9	10315	38	2006-09-26 00:00:00.000	Garden House Crowther Way 9012
10	10316	65	2006-09-27 00:00:00.000	8901 Milton Dr.
11	10317	48	2006-09-30 00:00:00.000	6789 Chiaroscuro Rd.
12	10339	51	2006-10-28 00:00:00.000	8901 rue St. Laurent
13	10340	9	2006-10-29 00:00:00.000	8901, rue des Bouchers
14	10341	73	2006-10-29 00:00:00.000	Vinbæltet 1234
15	10342	25	2006-10-30 00:00:00.000	Berliner Platz 0123
16	10343	44	2006-10-31 00:00:00.000	Magazinweg 4567
17	10363	17	2006-11-26 00:00:00.000	Walserweg 7890
18	10364	19	2006-11-26 00:00:00.000	3456 King George
19	10365	3	2006-11-27 00:00:00.000	Mataderos 3456
20	10366	29	2006-11-28 00:00:00.000	Rambla de Cataluña, 0123
21	10367	83	2006-11-28 00:00:00.000	Smagsloget 1234
22	10368	20	2006-11-29 00:00:00.000	Kirchgasse 9012
23	10396	25	2006-12-27 00:00:00.000	Berliner Platz 0123
24	10397	60	2006-12-27 00:00:00.000	Estrada da saúde n. 2345
25	10398	71	2006-12-30 00:00:00.000	9012 Suffolk Ln.
26	10399	83	2006-12-31 00:00:00.000	Smagsloget 2345

Query executed successfully.

3	**Conclusion** : After this trial, students will be able to know how to use various date and time functions in T-SQL.

Practical – Part 8 : Writing a SELECT query to combine 2 columns

Step	Information
1	**Scenario** : Marketing staff needs a more concise report when showing it to customers, by combining 2 da
2	[Question-1 2] Write a SELECT query against the *Sales.Customers table* and get *the contactname* and *cit* two columns so that it looks like this:

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

```
SELECT
    contactname + ', ' + city + ' (city:' + city + ')' AS contactdetails
FROM
    Sales.Customers;
```

100 %

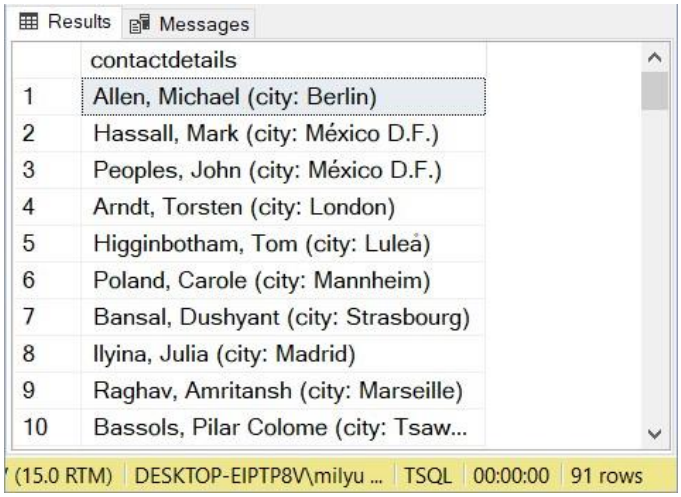
Results | Messages

	contactdetails
1	Allen, Michael, Berlin (city:Berlin)
2	Hassall, Mark, México D.F. (city:México D.F.)
3	Peoples, John, México D.F. (city:México D.F.)
4	Arndt, Torsten, London (city:London)
5	Higginbotham, Tom, Luleå (city:Luleå)
6	Poland, Carole, Mannheim (city:Mannheim)
7	Bansal, Dushyant, Strasbourg (city:Strasbourg)
8	Ilyina, Julia, Madrid (city:Madrid)
9	Raghav, Amritansh, Marseille (city:Marseille)
10	Bassols, Pilar Colome, Tsawassen (city:Tsawassen)
11	Jaffe, David, London (city:London)
12	Ray, Mike, Buenos Aires (city:Buenos Aires)
13	Benito, Almudena, México D.F. (city:México D.F.)
14	Jelitto, Jacek, Bern (city:Bern)
15	Richardson, Shawn, Sao Paulo (city:Sao Paulo)
16	Birkby, Dana, London (city:London)
17	Jones, TiAnna, Aachen (city:Aachen)
18	Rizaldy, Arif, Nantes (city:Nantes)
19	Boseman, Randall, London (city:London)
20	Kane, John, Graz (city:Graz)
21	Russo, Giuseppe, Sao Paulo (city:Sao Paulo)
22	Bueno, Janaina Burdan, Neville, Madrid (city:Madri...
23	Khanna, Karan, Lille (city:Lille)
24	San Juan, Patricia, Bräcke (city:Bräcke)
25	Carlson, Jason, München (city:München)
26	Koch, Paul, Nantes (city:Nantes)
27	Schmöllerl, Martin, Torino (city:Torino)
28	Cavaglieri, Giorgio, Lisboa (city:Lisboa)
29	Kolesnikova, Katerina, Barcelona (city:Barcelona)
30	Shabalin, Rostislav, Sevilla (city:Sevilla)
31	Cheng, Yao-Qiang, Campinas (city:Campinas)
32	Krishnan, Venky, Eugene (city:Eugene)
33	Sigurdarson, Hallur , Caracas (city:Caracas)
34	Cohen, Shy, Rio de Janeiro (city:Rio de Janeiro)

Query executed successfully.

Allen, Michael (city:Berlin,)

3	Execute the query in step 1 and screenshot the result. Compare it with the result shown in the following



Results | Messages

	contactdetails
1	Allen, Michael (city: Berlin)
2	Hassall, Mark (city: México D.F.)
3	Peoples, John (city: México D.F.)
4	Arndt, Torsten (city: London)
5	Higginbotham, Tom (city: Luleå)
6	Poland, Carole (city: Mannheim)
7	Bansal, Dushyant (city: Strasbourg)
8	Ilyina, Julia (city: Madrid)
9	Raghav, Amritansh (city: Marseille)
10	Bassols, Pilar Colome (city: Tsaw...

(15.0 RTM) | DESKTOP-EIPTP8V\milyu ... | TSQL | 00:00:00 | 91 rows

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

```
SELECT
    contactname + ', ' + city + ' (city:' + city + ')' AS contactdetails
FROM
    Sales.Customers;
```

100 %

Results | Messages

	contactdetails
1	Allen, Michael, Berlin (city:Berlin)
2	Hassall, Mark, México D.F. (city:México D.F.)
3	Peoples, John, México D.F. (city:México D.F.)
4	Arndt, Torsten, London (city:London)
5	Higginbotham, Tom, Luleå (city:Luleå)
6	Poland, Carole, Mannheim (city:Mannheim)
7	Bansal, Dushyant, Strasbourg (city:Strasbourg)
8	Ilyina, Julia, Madrid (city:Madrid)
9	Raghav, Amritansh, Marseille (city:Marseille)
10	Bassols, Pilar Colome, Tsawassen (city:Tsawassen)
11	Jaffe, David, London (city:London)
12	Ray, Mike, Buenos Aires (city:Buenos Aires)
13	Benito, Almudena, México D.F. (city:México D.F.)
14	Jelitto, Jacek, Bern (city:Bern)
15	Richardson, Shawn, Sao Paulo (city:Sao Paulo)
16	Birkby, Dana, London (city:London)
17	Jones, TiAnna, Aachen (city:Aachen)
18	Rizaldy, Arif, Nantes (city:Nantes)
19	Boseman, Randall, London (city:London)
20	Kane, John, Graz (city:Graz)
21	Russo, Giuseppe, Sao Paulo (city:Sao Paulo)
22	Bueno, Janaina Burdan, Neville, Madrid (city:Madri...
23	Khanna, Karan, Lille (city:Lille)
24	San Juan, Patricia, Bräcke (city:Bräcke)
25	Carlson, Jason, München (city:München)
26	Koch, Paul, Nantes (city:Nantes)
27	Schmöllerl, Martin, Torino (city:Torino)
28	Cavaglieri, Giorgio, Lisboa (city:Lisboa)
29	Kolesnikova, Katerina, Barcelona (city:Barcelona)
30	Shabalin, Rostislav, Sevilla (city:Sevilla)
31	Cheng, Yao-Qiang, Campinas (city:Campinas)
32	Krishnan, Venky, Eugene (city:Eugene)
33	Sigurdarson, Hallur , Caracas (city:Caracas)
34	Cohen, Shy, Rio de Janeiro (city:Rio de Janeiro)

Query executed successfully.

Practical – Part 9 : Writing a SELECT query to display all customers based on the first character in the contact name.

Step	Information
1	[Question- 13] Write a SELECT query to display *the contactname* and *contacttitle columns* from the *Sales.Customers table* . Filter to display only contact names whose first character is 'A' through 'G'.

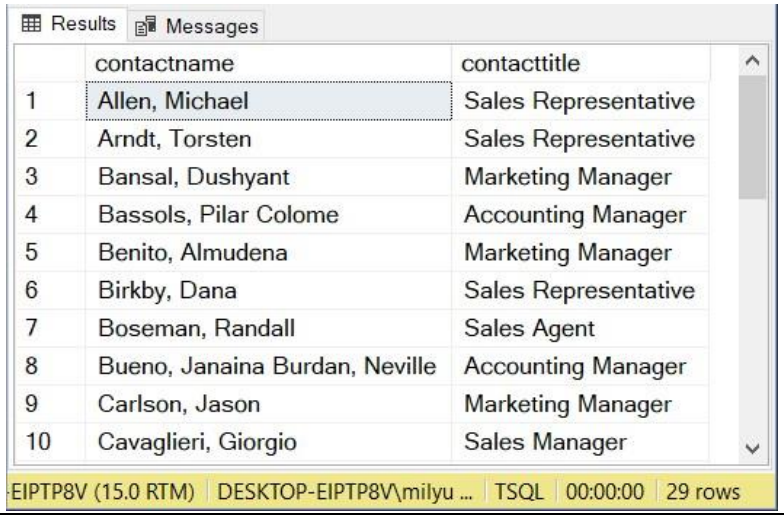
SQLQuery1.sql - M...IZAL RAHMAN (54))*

```
SELECT
    contactname,
    contacttitle
FROM
    Sales.Customers
WHERE
    LEFT(contactname, 1) BETWEEN 'A' AND 'G';
```

100 %

Results | Messages

	contactname	contacttitle
1	Allen, Michael	Sales Representative
2	Arndt, Torsten	Sales Representative
3	Bansal, Dushyant	Marketing Manager
4	Bassols, Pilar Colome	Accounting Manager
5	Benito, Almudena	Marketing Manager
6	Birkby, Dana	Sales Representative
7	Boseman, Randall	Sales Agent
8	Bueno, Janaina Burdan, Neville	Accounting Manager
9	Carlson, Jason	Marketing Manager
10	Cavaglieri, Giorgio	Sales Manager
11	Cheng, Yao-Qiang	Sales Associate
12	Cohen, Shy	Accounting Manager
13	Cr?ciun, Ovidiu V.	Sales Associate
14	De Oliveira, Jose	Sales Representative
15	Deshpande, Anu	Marketing Manager
16	Dressler, Marlies	Accounting Manager
17	Duerr, Bernard	Marketing Manager
18	Dupont-Roc, Patrice	Marketing Assistant
19	Egelund-Muller, Anja	Sales Representative
20	Fakhouri, Fadi	Sales Representative
21	Florczyk, Krzysztof	Accounting Manager
22	Gaffney, Lawrie	Sales Representative
23	Garden, Euan	Assistant Sales Agent
24	Ginters, Kaspars	Owner
25	Gonzalez, Nuria	Owner
26	Gulbis, Katrin	Accounting Manager
27	Geschwandtner, Jens	Owner
28	Fonteneau, Karl	Sales Manager

2	Execute the query in step 1 above and screenshot the result. Compare it with the result shown in the foll image:
	

Results | Messages

	contactname	contacttitle
1	Allen, Michael	Sales Representative
2	Arndt, Torsten	Sales Representative
3	Bansal, Dushyant	Marketing Manager
4	Bassols, Pilar Colome	Accounting Manager
5	Benito, Almudena	Marketing Manager
6	Birkby, Dana	Sales Representative
7	Boseman, Randall	Sales Agent
8	Bueno, Janaina Burdan, Neville	Accounting Manager
9	Carlson, Jason	Marketing Manager
10	Cavaglieri, Giorgio	Sales Manager

EIPTP8V (15.0 RTM) | DESKTOP-EIPTP8V\milyu ... | TSQL | 00:00:00 | 29 rows

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

```
SELECT
    contactname,
    contacttitle
FROM
    Sales.Customers
WHERE
    LEFT(contactname, 1) BETWEEN 'A' AND 'G';
```

100 %

Results | Messages

	contactname	contacttitle
1	Allen, Michael	Sales Representative
2	Arndt, Torsten	Sales Representative
3	Bansal, Dushyant	Marketing Manager
4	Bassols, Pilar Colome	Accounting Manager
5	Benito, Almudena	Marketing Manager
6	Birkby, Dana	Sales Representative
7	Boseman, Randall	Sales Agent
8	Bueno, Janaina Burdan, Neville	Accounting Manager
9	Carlson, Jason	Marketing Manager
10	Cavaglieri, Giorgio	Sales Manager
11	Cheng, Yao-Qiang	Sales Associate
12	Cohen, Shy	Accounting Manager
13	Cr?ciun, Ovidiu V.	Sales Associate
14	De Oliveira, Jose	Sales Representative
15	Deshpande, Anu	Marketing Manager
16	Dressler, Marlies	Accounting Manager
17	Duerr, Bernard	Marketing Manager
18	Dupont-Roc, Patrice	Marketing Assistant
19	Egelund-Muller, Anja	Sales Representative
20	Fakhouri, Fadi	Sales Representative
21	Florczyk, Krzysztof	Accounting Manager
22	Gaffney, Lawrie	Sales Representative
23	Garden, Euan	Assistant Sales Agent
24	Ginters, Kaspars	Owner
25	Gonzalez, Nuria	Owner
26	Gulbis, Katrin	Accounting Manager
27	Geschwandtner, Jens	Owner
28	Fonteneau, Karl	Sales Manager

3	**Conclusion** : After this trial, students should be able to understand and know how to combine character

Practical – Part 10 : Writing a SELECT query using the SUBSTRING function

Step	Information
1	[Question- 14] Write a SELECT query to display the *contactname column* from the *Sales.Customers table* and *replace* all commas with empty strings. Then, based on this column, add a column named *lastname* containing all the characters before the comma using the SUBSTRING function .

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

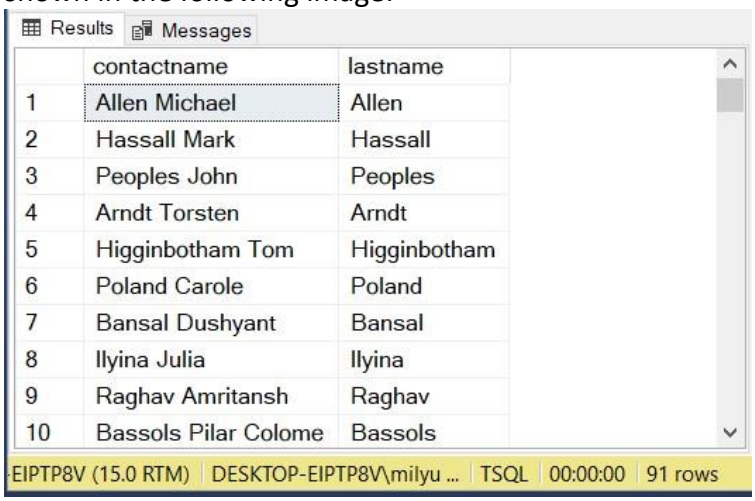
```
SELECT
    REPLACE(contactname, ',', '') AS contactname,
    SUBSTRING(contactname, 1, CHARINDEX(',', contactname) - 1) AS lastname
FROM
    Sales.Customers;
```

100 %

Results | Messages

	contactname	lastname
1	Allen Michael	Allen
2	Hassall Mark	Hassall
3	Peoples John	Peoples
4	Arndt Torsten	Arndt
5	Higginbotham Tom	Higginbotham
6	Poland Carole	Poland
7	Bansal Dushyant	Bansal
8	Ilyina Julia	Ilyina
9	Raghav Amritansh	Raghav
10	Bassols Pilar Colome	Bassols
11	Jaffe David	Jaffe
12	Ray Mike	Ray
13	Benito Almudena	Benito
14	Jelitto Jacek	Jelitto
15	Richardson Shawn	Richardson
16	Birkby Dana	Birkby
17	Jones TiAnna	Jones
18	Rizaldy Arif	Rizaldy
19	Boseman Randall	Boseman
20	Kane John	Kane
21	Russo Giuseppe	Russo
22	Bueno Janaina Burdan Neville	Bueno
23	Khanna Karan	Khanna
24	San Juan Patricia	San Juan
25	Carlson Jason	Carlson
26	Koch Paul	Koch
27	Schmöllerl Martin	Schmöllerl
28	Cavaglieri Giorgio	Cavaglieri

Query executed successfully.

2	Execute the query in step 1 above and screenshot the result. Compare it with the result shown in the following image: 

Results | Messages

	contactname	lastname
1	Allen Michael	Allen
2	Hassall Mark	Hassall
3	Peoples John	Peoples
4	Arndt Torsten	Arndt
5	Higginbotham Tom	Higginbotham
6	Poland Carole	Poland
7	Bansal Dushyant	Bansal
8	Ilyina Julia	Ilyina
9	Raghav Amritansh	Raghav
10	Bassols Pilar Colome	Bassols

-EIPTP8V (15.0 RTM) | DESKTOP-EIPTP8V\milyu ... | TSQL | 00:00:00 | 91 rows

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

```
SELECT
    REPLACE(contactname, ',', '') AS contactname,
    SUBSTRING(contactname, 1, CHARINDEX(',', contactname) - 1) AS lastname
FROM
    Sales.Customers;
```

100 %

Results | Messages

	contactname	lastname
1	Allen Michael	Allen
2	Hassall Mark	Hassall
3	Peoples John	Peoples
4	Arndt Torsten	Arndt
5	Higginbotham Tom	Higginbotham
6	Poland Carole	Poland
7	Bansal Dushyant	Bansal
8	Ilyina Julia	Ilyina
9	Raghav Amritansh	Raghav
10	Bassols Pilar Colome	Bassols
11	Jaffe David	Jaffe
12	Ray Mike	Ray
13	Benito Almudena	Benito
14	Jelitto Jacek	Jelitto
15	Richardson Shawn	Richardson
16	Birkby Dana	Birkby
17	Jones TiAnna	Jones
18	Rizaldy Arif	Rizaldy
19	Boseman Randall	Boseman
20	Kane John	Kane
21	Russo Giuseppe	Russo
22	Bueno Janaina Burdan Neville	Bueno
23	Khanna Karan	Khanna
24	San Juan Patricia	San Juan
25	Carlson Jason	Carlson
26	Koch Paul	Koch
27	Schmöllerl Martin	Schmöllerl
28	Cavaglieri Giorgio	Cavaglieri

Query executed successfully.

Practical – Part 11 : Writing a SELECT query to change the customer code

Step	Information
1	[Question-1 5] Write a SELECT query to display the *custid column* from the *Sales.Customers table* . Based on this column, add a column containing the 6-digit customer code, formatted with the letter C and a leading 0. For example, *a custid* with code 1 is displayed as `C00001` , etc.

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

```
SELECT
    custid,
    'C' + RIGHT('00000' + CAST(custid AS VARCHAR), 5) AS newcustid
FROM
    Sales.Customers;
```

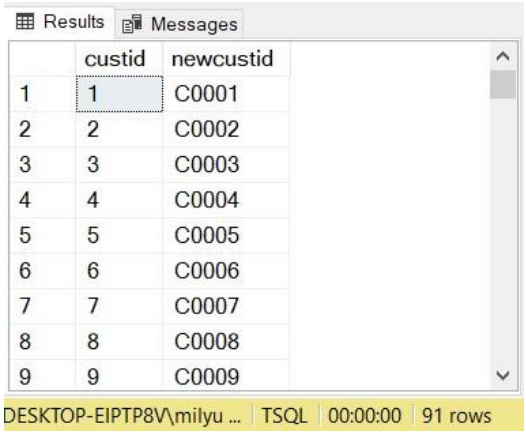
100 %

Results | Messages

	custid	newcustid
1	1	C00001
2	2	C00002
3	3	C00003
4	4	C00004
5	5	C00005
6	6	C00006
7	7	C00007
8	8	C00008
9	9	C00009
10	11	C00011
11	12	C00012
12	13	C00013
13	14	C00014
14	16	C00016
15	17	C00017
16	18	C00018
17	19	C00019
18	20	C00020
19	22	C00022
20	23	C00023
21	24	C00024
22	25	C00025
23	26	C00026
24	27	C00027
25	28	C00028
26	29	C00029
27	30	C00030
28	39	C00039

Query executed successfully.

2	Execute the query in step 1 above and *screenshot* the result. Compare it with the result shown in the following image:



	custid	newcustid
1	1	C0001
2	2	C0002
3	3	C0003
4	4	C0004
5	5	C0005
6	6	C0006
7	7	C0007
8	8	C0008
9	9	C0009

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

```
SELECT
    custid,
    'C' + RIGHT('00000' + CAST(custid AS VARCHAR), 5) AS newcustid
FROM
    Sales.Customers;
```

100 %

Results | Messages

	custid	newcustid
1	1	C00001
2	2	C00002
3	3	C00003
4	4	C00004
5	5	C00005
6	6	C00006
7	7	C00007
8	8	C00008
9	9	C00009
10	11	C00011
11	12	C00012
12	13	C00013
13	14	C00014
14	16	C00016
15	17	C00017
16	18	C00018
17	19	C00019
18	20	C00020
19	22	C00022
20	23	C00023
21	24	C00024
22	25	C00025
23	26	C00026
24	27	C00027
25	28	C00028
26	29	C00029
27	30	C00030
28	39	C00039

Query executed successfully.

q

Practical – Part 14 : Writing a SELECT query to display the number of occurrences of a character

Step	Information
1	[Question-1 6] Write a SELECT query to display *the contactname column* from the *Sales.Customers table* . Based on this column, add a column that displays the number of 'a' characters in the contact name. (Hint: Use the `REPLACE` and `LEN string functions`).

Sort the results by largest.

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

```
SELECT
    contactname,
    LEN(contactname) - LEN(REPLACE(contactname, 'a', '')) AS numberofa
FROM
    Sales.Customers
ORDER BY
    numberofa DESC;
```

100 %

Results | Messages

	contactname	numberofa
1	Raghav, Amritansh	4
2	Bueno, Janaina Burdan, Neville	4
3	Khanna, Karan	4
4	San Juan, Patricia	4
5	Marinova, Nadejda	4
6	Syamala, Manoj	4
7	Larsson, Katarina	4
8	Tuntisangaroon, Sittichai	3
9	Wojciechowska, Agnieszka	3
10	Sunkammurali, Krishna	3
11	Szymczak, Rados?aw	3
12	Boseman, Randall	3
13	Kolesnikova, Katerina	3
14	Shabalin, Rostislav	3
15	Hassall, Mark	3
16	Bansal, Dushyant	3
17	Ilyina, Julia	2
18	Allen, Michael	2
19	Poland, Carole	2
20	Bassols, Pilar Colome	2
21	Jaffe, David	2
22	Richardson, Shawn	2
23	Birkby, Dana	2
24	Jones, TiAnna	2
25	Rizaldy, Arif	2
26	Cheng, Yao-Qiang	2
27	Sigurdarson, Hallur	2
28	Benito, Almudena	2

Query executed successfully.

2	Execute the query in step 1 above and *screenshot* the result. Compare it with the result shown in the following image:

Results | Messages

	contactname	numberofa
1	Allen, Michael	2
2	Hassall, Mark	3
3	Peoples, John	0
4	Arndt, Torsten	1
5	Higginbotham, Tom	1
6	Poland, Carole	2
7	Bansal, Dushyant	3
8	Ilyina, Julia	2
9	Raghav, Amritansh	4

DESKTOP-EIPTP8V\milyu ... | TSQL | 00:00:00 | 91 rows

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

```
SELECT
    contactname,
    LEN(contactname) - LEN(REPLACE(contactname, 'a', '')) AS numberofa
FROM
    Sales.Customers
ORDER BY
    numberofa DESC;
```

100 %

Results | Messages

	contactname	numberofa
1	Raghav, Amritansh	4
2	Bueno, Janaina Burdan, Neville	4
3	Khanna, Karan	4
4	San Juan, Patricia	4
5	Marinova, Nadejda	4
6	Syamala, Manoj	4
7	Larsson, Katarina	4
8	Tuntisangaroon, Sittichai	3
9	Wojciechowska, Agnieszka	3
10	Sunkammurali, Krishna	3
11	Szymczak, Rados?aw	3
12	Boseman, Randall	3
13	Kolesnikova, Katerina	3
14	Shabalin, Rostislav	3
15	Hassall, Mark	3
16	Bansal, Dushyant	3
17	Ilyina, Julia	2
18	Allen, Michael	2
19	Poland, Carole	2
20	Bassols, Pilar Colome	2
21	Jaffe, David	2
22	Richardson, Shawn	2
23	Birkby, Dana	2
24	Jones, TiAnna	2
25	Rizaldy, Arif	2
26	Cheng, Yao-Qiang	2
27	Sigurdarson, Hallur	2
28	Benito, Almudena	2

Query executed successfully.

3	Conclusion: After the trial is conducted, students can find out how to use various character functions.

-- Have a great time doing it -