Course Name

Advanced Database Topics (COMP-8157)

Document Type

Lab 2

Professor

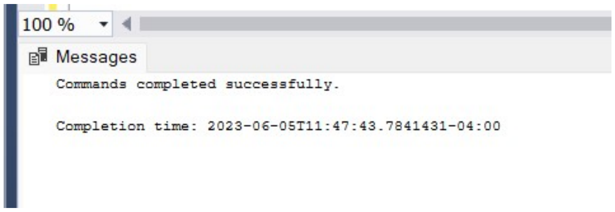
Dr. Shafaq Khan

Student Name	Student ID
Manjinder Singh	110097177

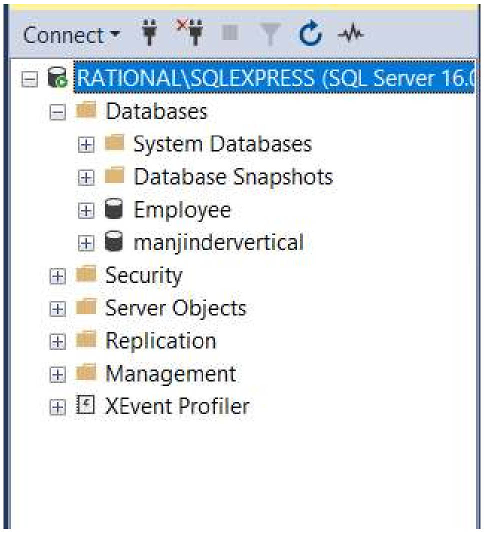
PART-1

1. **Create a database <yourfirstname>vertical.**

```
-- PART 1
-- Answer 1
-- Creating Database with name - manjindervertical.
CREATE DATABASE manjindervertical;

-- Pointing to Database having name - manjindervertical.
       USE manjindervertical;
```



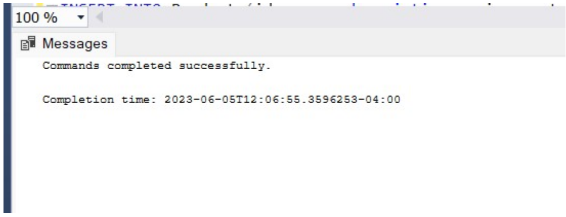
```
Commands completed successfully.

Completion time: 2023-06-05T11:47:43.7841431-04:00
```



2. **Create a table “Product” table with the following columns: id, name, description, price, category, brand, and quantity. (Note: Insert 10 rows of data in this table)**

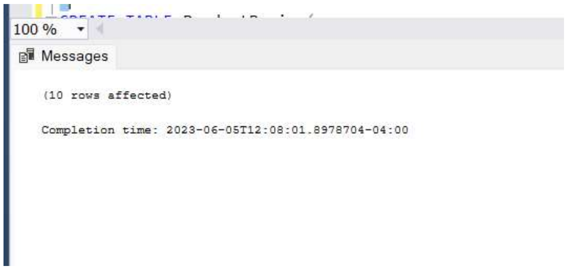
```
-- Answer 2
 -- Creating table - Product with columns id, name, description, price,
category, brand, quantity.
CREATE TABLE Product (
    id INT PRIMARY KEY,
    name VARCHAR(255),
    description VARCHAR(255),
    price DECIMAL(10, 2),
    category VARCHAR(255),
    brand VARCHAR(255),
    quantity INT
);


 -- Inserting 10 records into the table Product.
INSERT INTO Product (id, name, description, price, category, brand, quantity)
VALUES
    (1, 'Stylus Pen', 'Pen used on gadgets', 37.92, 'Pen', 'Camelin', 25),
    (2, 'Comic Book', 'genre related to comic', 49.32, 'Book', 'McGraw Hill',
37),
    (3, 'Ballpoint Pen', 'Blue color ballpoint pen', 14.99, 'Pen', 'National',
98),
    (4, 'Hand Bag', 'used for many purposes', 39.43, 'Bag', 'Louis Vuiton', 19),
    (5, 'Duffle Bag', 'used for gym', 92.11, 'Bag', 'Versace', 82),
    (6, 'Fountain Pen', 'used by experts', 50.83, 'Pen', 'Hero', 78),
    (7, 'Action Book', 'book of type action', 49.95, 'Book', 'Penguin', 105),
    (8, 'Gel Pen', 'makes writing smooth', 11.89, 'Pen', 'Speedball', 48),
    (9, 'Horror Book', 'book of type horror', 89.32, 'Book', 'Pirates', 62),
    (10, 'Tote Bag', 'bag used by students', 36.88, 'Bag', 'Bentley', 48);

-- Executing to show that after inserting 10 rows, it is reflecting in Product
table.
 select * from Product;
```

Create Table Command Output:



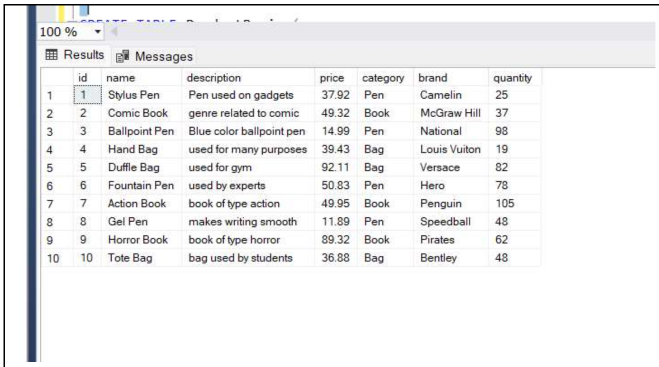
```
Commands completed successfully.

Completion time: 2023-06-05T12:06:55.3596253-04:00
```

Insert Command Output:



```
(10 rows affected)

Completion time: 2023-06-05T12:08:01.8978704-04:00
```

Select Command(just to show table is successfully created and data is inserted)



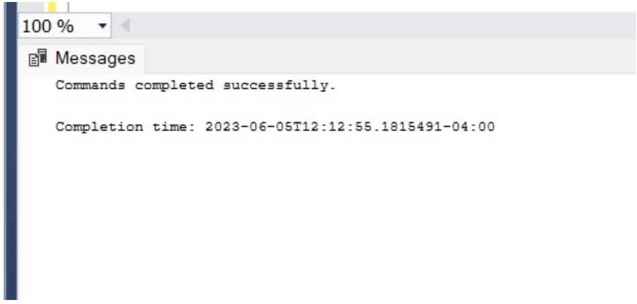
	id	name	description	price	category	brand	quantity
1	1	Stylus Pen	Pen used on gadgets	37.92	Pen	Camelin	25
2	2	Comic Book	genre related to comic	49.32	Book	McGraw Hill	37
3	3	Ballpoint Pen	Blue color ballpoint pen	14.99	Pen	National	98
4	4	Hand Bag	used for many purposes	39.43	Bag	Louis Vuiton	19
5	5	Duffle Bag	used for gym	92.11	Bag	Versace	82
6	6	Fountain Pen	used by experts	50.83	Pen	Hero	78
7	7	Action Book	book of type action	49.95	Book	Penguin	105
8	8	Gel Pen	makes writing smooth	11.89	Pen	Speedball	48
9	9	Horror Book	book of type horror	89.32	Book	Pirates	62
10	10	Tote Bag	bag used by students	36.88	Bag	Bentley	48

3. Apply vertical partitioning by dividing the above table into two partition tables:

i. "ProductBasic" table (Columns: id, name, description and category)

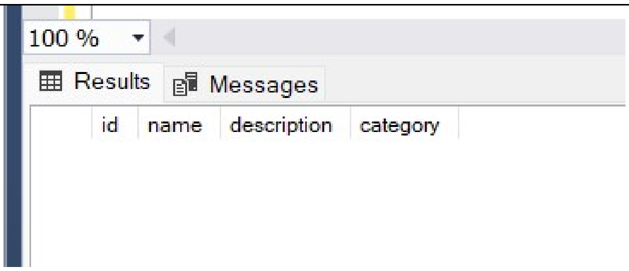
```
-- Answer 3
-- Creating ProductBasic Table for vertical Partitioning
CREATE TABLE ProductBasic (
    id INT PRIMARY KEY,
    name VARCHAR(255),
    description VARCHAR(255),
    category VARCHAR(255)
);

select * from ProductBasic; -- just to show table is successfully created
and data is inserted.

-- Inserting into ProductBasic Table from Product Table
INSERT INTO ProductBasic (id, name, description, category)
SELECT id, name, description, category FROM Product;
```

Create Command Output:



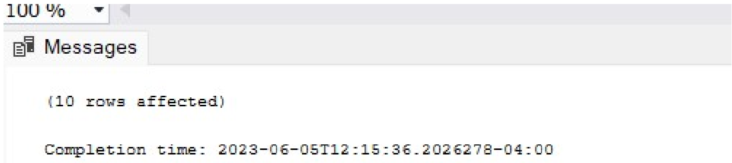
```
Commands completed successfully.

Completion time: 2023-06-05T12:12:55.1815491-04:00
```

Select Command Output(Shows column names only and data hasn't inserted yet.):



id	name	description	category

Insert Command Output:

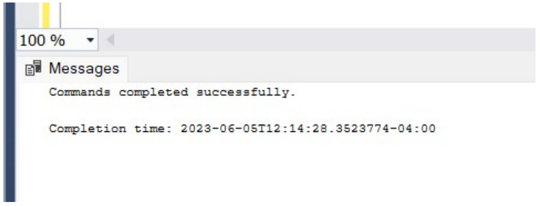


```
(10 rows affected)

Completion time: 2023-06-05T12:15:36.2026278-04:00
```

ii. “ProductDetails” table (Columns: id, price, brand, quantity)

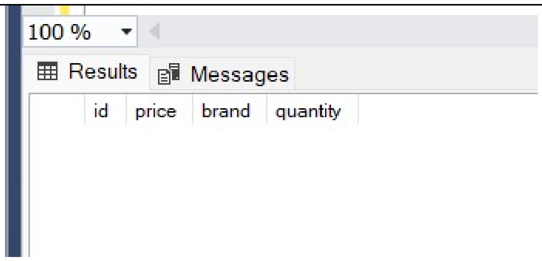
```
-- Creating ProductDetails Table for vertical Partitioning
CREATE TABLE ProductDetails (
    id INT PRIMARY KEY,
    price DECIMAL(10, 2),
    brand VARCHAR(255),
    quantity INT
);

select * from ProductDetails; -- just to show table is successfully created and
data is inserted.

-- Inserting into ProductDetails Table from Product table.
INSERT INTO ProductDetails (id, price, brand, quantity)
SELECT id, price, brand, quantity FROM Product;
```

Create Command Output:



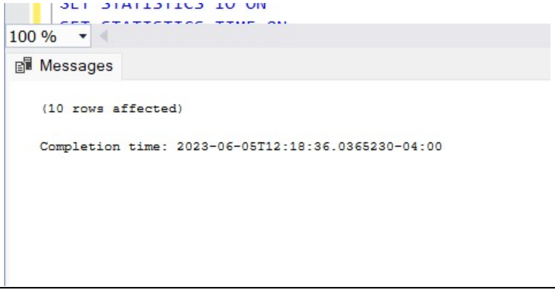
```
Commands completed successfully.

Completion time: 2023-06-05T12:14:28.3523774-04:00
```

Select Command Output(Shows column names only and data hasn’t inserted yet.):



id	price	brand	quantity

Insert Command Output:

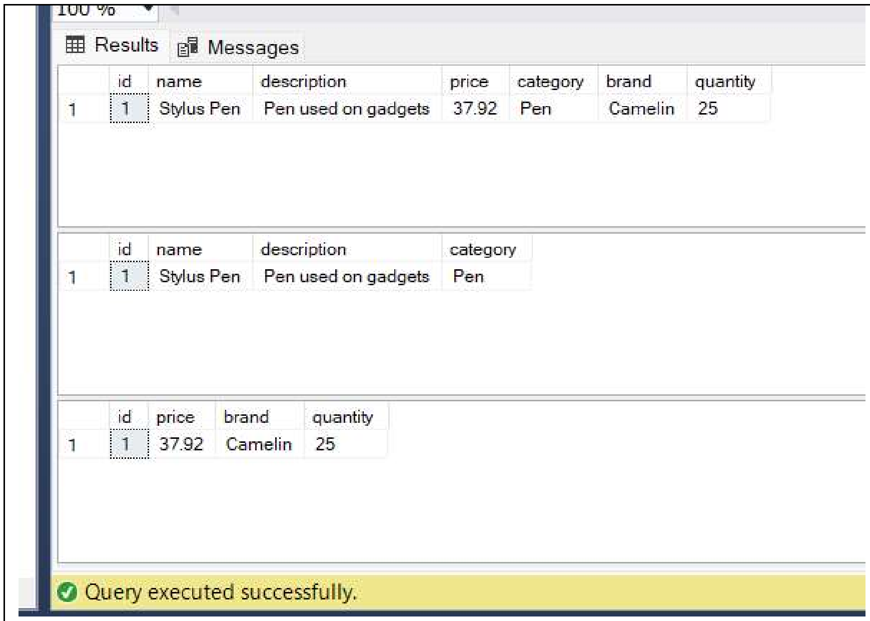


```
(10 rows affected)

Completion time: 2023-06-05T12:18:36.0365230-04:00
```

4. Calculate the query performance of each table by retrieving the same 'id' from three tables.

```
--Answer 4
 -- Computing the query performance of each table by retrieving the same
'id' from three tables.

SET STATISTICS IO ON
SET STATISTICS TIME ON
SELECT *
FROM Product p
WHERE p.id = 1
SET STATISTICS IO OFF
SET STATISTICS TIME OFF;

SET STATISTICS IO ON
SET STATISTICS TIME ON
SELECT *
FROM ProductBasic p
WHERE p.id = 1
SET STATISTICS IO OFF
SET STATISTICS TIME OFF;



SET STATISTICS IO ON
SET STATISTICS TIME ON
SELECT *
FROM ProductDetails p
WHERE p.id = 1
SET STATISTICS IO OFF
SET STATISTICS TIME OFF;
```

Output is in order for tables: `Product, ProductBasic, ProductDetails.`



	id	name	description	price	category	brand	quantity
1	1	Stylus Pen	Pen used on gadgets	37.92	Pen	Camelin	25

	id	name	description	category
1	1	Stylus Pen	Pen used on gadgets	Pen

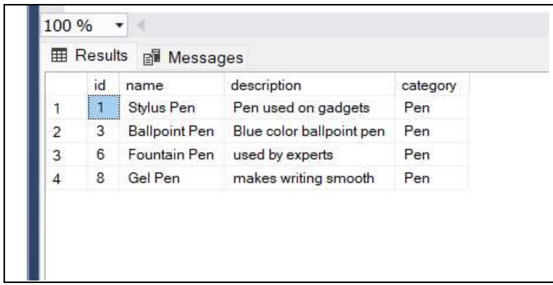
	id	price	brand	quantity
1	1	37.92	Camelin	25

Query executed successfully.

```
-- It takes around less than 1 second for the execution of all three above
queries owing to the reason of less number of records in the table. If in case
more data is present in tables than it will definitely take more time to
retrieve.
-- In addition to this, to compute query performance, you can
-- 1. Ensure Indexing
-- 2. Query Optimization
-- 3. Query Execution Plan
-- 4. Measure Execution Time
-- 5. Analyze Query Performance
-- 6. Index Optimization
-- 7. Query Tuning
-- 8. Monitor and Fine Tune

-- Apart from above, it must be noted that query performance is also dependent
on data volume, hardware resources, query complexity, and indexing. It is
crucial to analyze and optimize query based on requirements.
```

5. Retrieve basic information of all products in a specific category from the “ProductBasic” table.

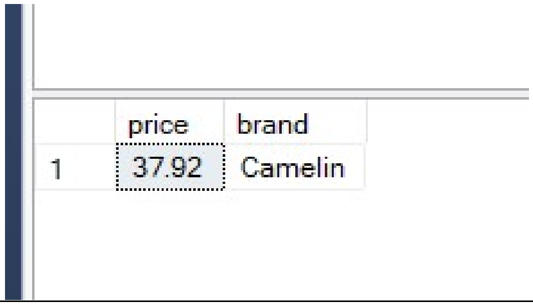
```
        --Answer 5
        -- Retrieving information of all the products listed under specific
        category from ProductBasic Table.
        select * from ProductBasic where category ='Pen';
```



	id	name	description	category
1	1	Stylus Pen	Pen used on gadgets	Pen
2	3	Ballpoint Pen	Blue color ballpoint pen	Pen
3	6	Fountain Pen	used by experts	Pen
4	8	Gel Pen	makes writing smooth	Pen

6. **Retrieve the price and brand of a specific product from the "ProductDetails" table.**

```
--Answer 6
-- Retrieving only the price and brand information of a single product listed
under Product_Details table based on product id(1).
select pd.price, pd.brand
from  ProductDetails pd
where pd.id=1;
```

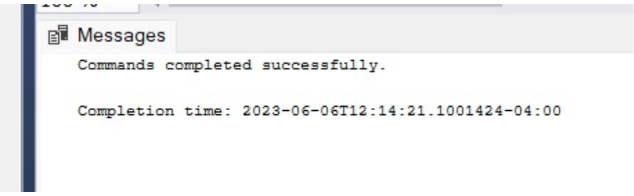


	price	brand
1	37.92	Camelin

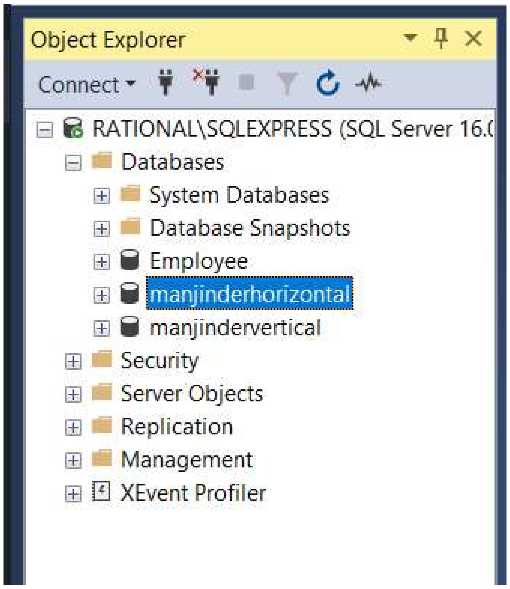
PART-2

1. Create a database horizontal.

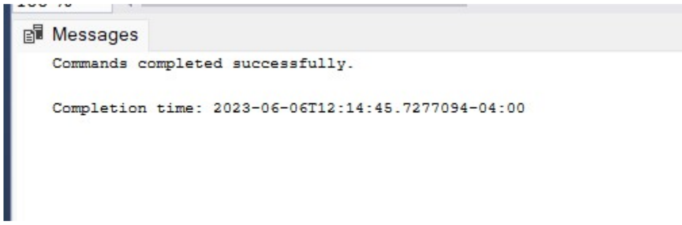
```
-- Answer 1
-- Creating Database with name - manjinderhorizontal
CREATE DATABASE manjinderhorizontal;
```



```
Commands completed successfully.

Completion time: 2023-06-06T12:14:21.1001424-04:00
```



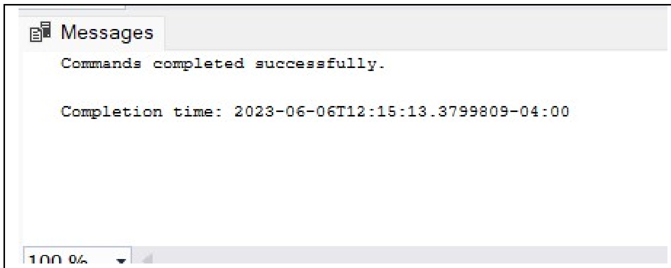
```sql
-- Pointing to Database having name - manjinderhorizontal
USE manjinderhorizontal;
```



```
Commands completed successfully.

Completion time: 2023-06-06T12:14:45.7277094-04:00
```

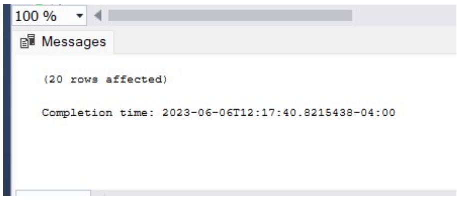
2. Create a table "Birthday" table with the following columns: s.no, name, date, month (01 - 06) and year. (Note: Insert 20 rows of data in this table)

```sql
-- Answer 2
-- Creating Table - Birthday with columns s_no,name,date,month,year
CREATE TABLE Birthday (
  s_no INT,
  name VARCHAR(50),
  date INT,
  month INT,
  year INT
);
```



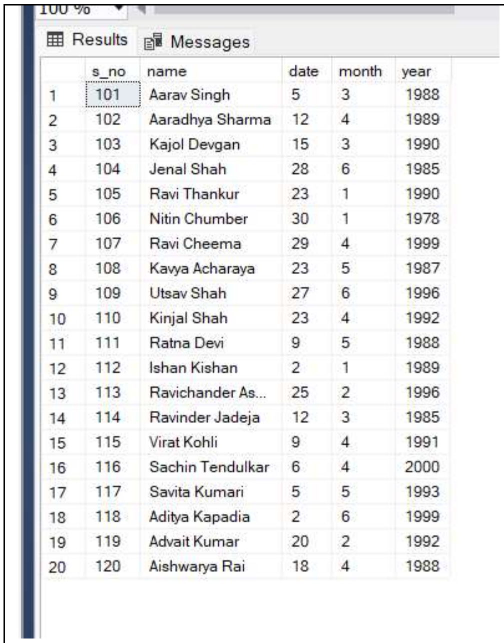
```
Commands completed successfully.

Completion time: 2023-06-06T12:15:13.3799809-04:00
```

```sql
-- Inserting 20 records into the table Birthday
INSERT INTO Birthday (s_no, name, date, month, year)
VALUES
(101, 'Aarav Singh', 5, 3, 1988),
(102, 'Aaradhya Sharma', 12, 4, 1989),
(103, 'Kajol Devgan', 15, 3, 1990),
(104, 'Jenal Shah', 28, 6, 1985),
(105, 'Ravi Thankur', 23, 1, 1990),
(106, 'Nitin Chumber', 30, 1, 1978),
(107, 'Ravi Cheema', 29, 4, 1999),
(108, 'Kavya Acharaya', 23, 5, 1987),
(109, 'Utsav Shah', 27, 6, 1996),
(110, 'Kinjal Shah', 23, 4, 1992),
(111, 'Ratna Devi',9, 5, 1988),
(112, 'Ishan Kishan', 2, 1, 1989),
(113, 'Ravichander Ashwin', 25, 2, 1996),
(114, 'Ravinder Jadeja', 12, 3, 1985),
(115, 'Virat Kohli', 9, 4, 1991),
(116, 'Sachin Tendulkar', 6, 4, 2000),
(117, 'Savita Kumari', 5, 5, 1993),
(118, 'Aditya Kapadia', 2, 6, 1999),
(119, 'Advait Kumar', 20, 2, 1992),
(120, 'Aishwarya Rai', 18, 4, 1988);
```



```
(20 rows affected)

Completion time: 2023-06-06T12:17:40.8215438-04:00
```

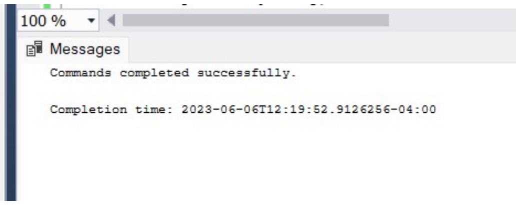
```sql
-- Executing to show that after inserting 20 rows, it is reflecting in Birthday table.
select * from Birthday;
```



	s_no	name	date	month	year
1	101	Aarav Singh	5	3	1988
2	102	Aaradhya Sharma	12	4	1989
3	103	Kajol Devgan	15	3	1990
4	104	Jenal Shah	28	6	1985
5	105	Ravi Thankur	23	1	1990
6	106	Nitin Chumber	30	1	1978
7	107	Ravi Cheema	29	4	1999
8	108	Kavya Acharaya	23	5	1987
9	109	Utsav Shah	27	6	1996
10	110	Kinjal Shah	23	4	1992
11	111	Ratna Devi	9	5	1988
12	112	Ishan Kishan	2	1	1989
13	113	Ravichander As...	25	2	1996
14	114	Ravinder Jadeja	12	3	1985
15	115	Virat Kohli	9	4	1991
16	116	Sachin Tendulkar	6	4	2000
17	117	Savita Kumari	5	5	1993
18	118	Aditya Kapadia	2	6	1999
19	119	Advait Kumar	20	2	1992
20	120	Aishwarya Rai	18	4	1988

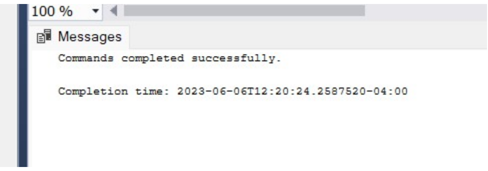
3. Create filegroups within the database to divide them by month.

```
-- Answer 3
-- Creating filegroups within manjinderhorizontal database to divide them based on
month
ALTER DATABASE manjinderhorizontal ADD FILEGROUP January;
ALTER DATABASE manjinderhorizontal ADD FILEGROUP February;
ALTER DATABASE manjinderhorizontal ADD FILEGROUP March;
ALTER DATABASE manjinderhorizontal ADD FILEGROUP April;
ALTER DATABASE manjinderhorizontal ADD FILEGROUP May;
ALTER DATABASE manjinderhorizontal ADD FILEGROUP June;
```



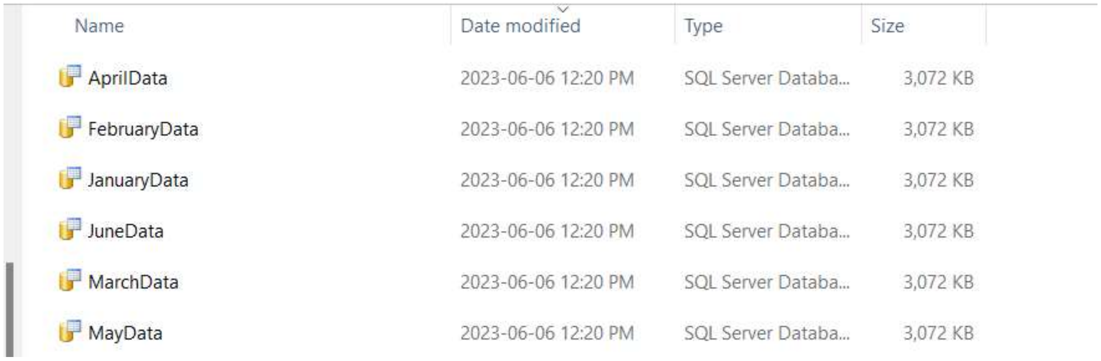
```
Commands completed successfully.

Completion time: 2023-06-06T12:19:52.9126256-04:00
```

```
-- Assigning location on system to every filegroup(segregated by month)
ALTER DATABASE manjinderhorizontal ADD FILE (
    NAME = [JanuaryData],
    FILENAME = 'F:\Sql Server Express Edition
2022\MSSQL16.SQLEXPRESS\MSSQL\DATA\JanuaryData.ndf',
    SIZE = 3072 KB,
    MAXSIZE = UNLIMITED,
    FILEGROWTH = 1024 KB
) TO FILEGROUP [January];

ALTER DATABASE manjinderhorizontal ADD FILE (
    NAME = [FebruaryData],
    FILENAME = 'F:\Sql Server Express Edition 
2022\MSSQL16.SQLEXPRESS\MSSQL\DATA\FebruaryData.ndf',
    SIZE = 3072 KB,
    MAXSIZE = UNLIMITED,
    FILEGROWTH = 1024 KB
) TO FILEGROUP [February];

ALTER DATABASE manjinderhorizontal ADD FILE (
    NAME = [MarchData],
    FILENAME = 'F:\Sql Server Express Edition 
2022\MSSQL16.SQLEXPRESS\MSSQL\DATA\MarchData.ndf',
    SIZE = 3072 KB,
    MAXSIZE = UNLIMITED,
    FILEGROWTH = 1024 KB
) TO FILEGROUP [March];

ALTER DATABASE manjinderhorizontal ADD FILE (
    NAME = [AprilData],
    FILENAME = 'F:\Sql Server Express Edition 
2022\MSSQL16.SQLEXPRESS\MSSQL\DATA\AprilData.ndf',
    SIZE = 3072 KB,
    MAXSIZE = UNLIMITED,
    FILEGROWTH = 1024 KB
) TO FILEGROUP [April];

ALTER DATABASE manjinderhorizontal ADD FILE (
    NAME = [MayData],
    FILENAME = 'F:\Sql Server Express Edition 
2022\MSSQL16.SQLEXPRESS\MSSQL\DATA\MayData.ndf',
    SIZE = 3072 KB,
    MAXSIZE = UNLIMITED,
    FILEGROWTH = 1024 KB
) TO FILEGROUP [May];

ALTER DATABASE manjinderhorizontal ADD FILE (
    NAME = [JuneData],
    FILENAME = 'F:\Sql Server Express Edition 
2022\MSSQL16.SQLEXPRESS\MSSQL\DATA\JuneData.ndf',
    SIZE = 3072 KB,
    MAXSIZE = UNLIMITED,
    FILEGROWTH = 1024 KB
) TO FILEGROUP [June];
```



```
Commands completed successfully.

Completion time: 2023-06-06T12:20:24.2587520-04:00
```

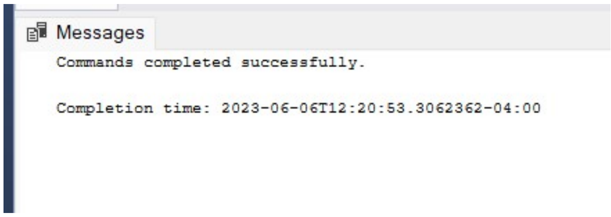
```
-- Screenshot of files from local system to show that files are created successfully.
```



Name	Date modified	Type	Size
AprilData	2023-06-06 12:20 PM	SQL Server Databa...	3,072 KB
FebruaryData	2023-06-06 12:20 PM	SQL Server Databa...	3,072 KB
JanuaryData	2023-06-06 12:20 PM	SQL Server Databa...	3,072 KB
JuneData	2023-06-06 12:20 PM	SQL Server Databa...	3,072 KB
MarchData	2023-06-06 12:20 PM	SQL Server Databa...	3,072 KB
MayData	2023-06-06 12:20 PM	SQL Server Databa...	3,072 KB

4. Create a partition function <yourfirstname>ByMonth (Note: The datatype of the month to be integer)

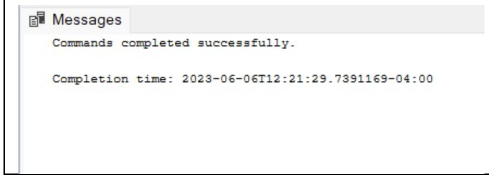
```
-- Answer 4
-- Creating a partition function by Month of type Integer.
CREATE PARTITION FUNCTION manjinderByMonth(INT)
AS RANGE LEFT FOR VALUES (1, 2, 3, 4, 5);
```



```
Commands completed successfully.

Completion time: 2023-06-06T12:20:53.3062362-04:00
```

5. Create a partition scheme <yourfirstname>ByMonthADT

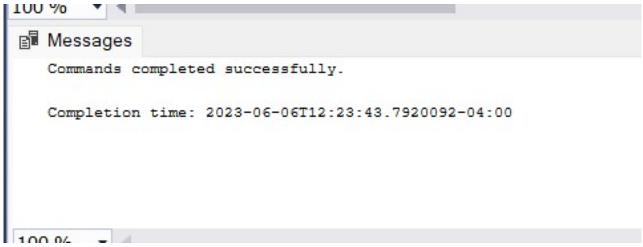
```
-- Answer 5
-- Creating a partition scheme by month Screenshot of files to show that files are
created successfully
CREATE PARTITION SCHEME manjinderByMonthADT
AS PARTITION manjinderByMonth
TO (January, February, March, April, May, June);
```



```
Commands completed successfully.

Completion time: 2023-06-06T12:21:29.7391169-04:00
```

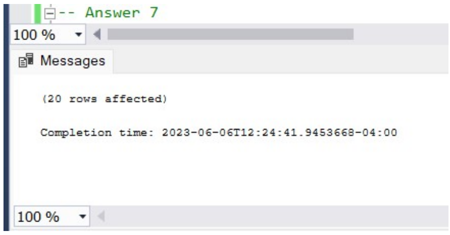
6. Create or modify the table and Specify the partition scheme as the storage location to segment the data out and store it within the appropriate file group.

```
-- Answer 6
-- In this we are creating and modifying the table. Then, specifying the partition
scheme as the storage location to segment the data out and storing it in appropriate
file group.
CREATE TABLE BirthdayMonth(
s_no int IDENTITY(1,1) NOT NULL,
    name VARCHAR(100),
    month int,
PRIMARY KEY (s_no, month))
ON manjinderByMonthADT(month);
```



```
Commands completed successfully.

Completion time: 2023-06-06T12:23:43.7920092-04:00
```

```
INSERT INTO BirthdayMonth (name, month)
VALUES
('Aarav Singh', 3),
('Aaradhya Sharma', 4),
('Kajol Devgan', 3),
('Jenal Shah', 6),
('Ravi Thankur', 1),
('Nitin Chumber', 1),
('Ravi Cheema', 4),
('Kavya Acharaya', 5),
('Utsav Shah', 6),
('Kinjal Shah', 4),
('Ratna Devi',5),
('Ishan Kishan', 1),
('Ravichander Ashwin', 2),
('Ravinder Jadeja', 3),
('Virat Kohli', 4),
('Sachin Tendulkar', 4),
('Savita Kumari', 5),
('Aditya Kapadia', 6),
('Advait Kumar',2),
('Aishwarya Rai',4);
```

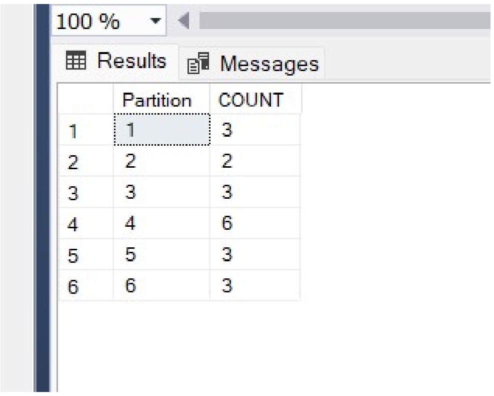


```
-- Answer 7
```

```
(20 rows affected)

Completion time: 2023-06-06T12:24:41.9453668-04:00
```

7. Write a query to check the number of records in each partition.

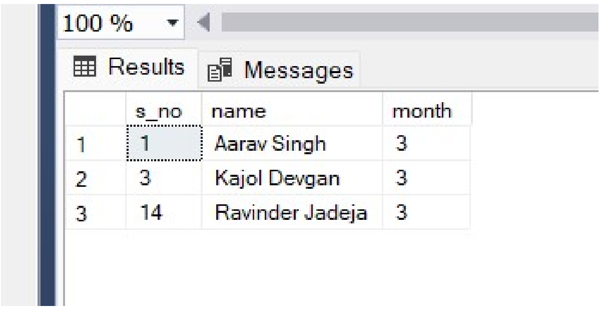
```
-- Answer 7
-- Checking number of records in each paritition i.e. in each month.
SELECT $PARTITION.manjinderByMonth(month) AS Partition, COUNT(*)
AS [COUNT] FROM BirthdayMonth
GROUP BY $PARTITION.manjinderByMonth(month) ORDER
BY Partition;
```



	Partition	COUNT
1	1	3
2	2	2
3	3	3
4	4	6
5	5	3
6	6	3

8. Execute the records in partition number 3

```
-- Answer 8
-- Fetching the records in Partition Number - 3 i.e. March Month
SELECT *
FROM BirthdayMonth
WHERE $PARTITION.manjinderByMonth(month) = 3;
```



	s_no	name	month
1	1	Aarav Singh	3
2	3	Kajol Devgan	3
3	14	Ravinder Jadeja	3