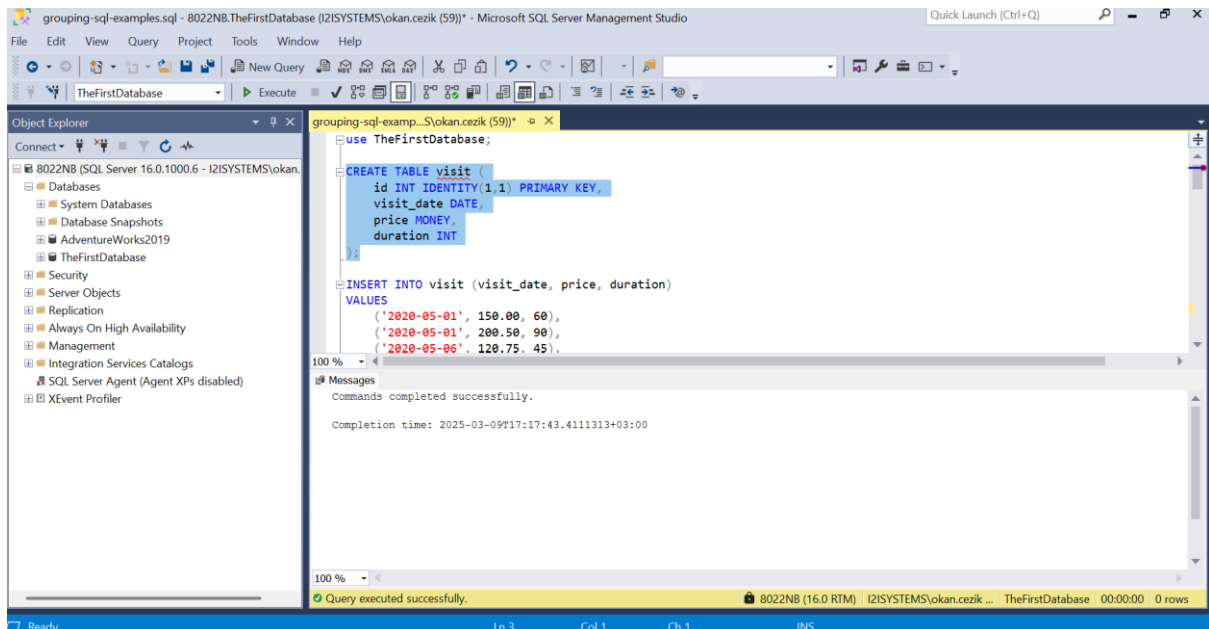


GROUPING EXAMPLES

STEP-1

Creating the visit Table



CREATE TABLE visit: This command creates a new table called visit.

id INT IDENTITY(1,1) PRIMARY KEY:

- id is an integer column that auto-increments with each new record, starting from 1. The IDENTITY(1,1) part specifies that the first value starts at 1 and increments by 1 for each new record.
- The PRIMARY KEY constraint ensures that the id column uniquely identifies each record in the table.

visit_date DATE: A column to store the date of the visit. The data type DATE is used to store dates in the format YYYY-MM-DD.

price MONEY: A column to store the price of the visit, using the MONEY data type to handle currency values.

duration INT: A column to store the duration of the visit in minutes, using the INT (integer) data type.

STEP-2

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'TheFirstDatabase'. The main query window contains the following SQL code:

```
id INT IDENTITY(1,1) PRIMARY KEY,  
visit_date DATE,  
price MONEY,  
duration INT  
);  
  
INSERT INTO visit (visit_date, price, duration)  
VALUES  
('2020-05-01', 150.00, 60),  
('2020-05-01', 200.50, 90),  
('2020-05-06', 120.75, 45),  
('2024-05-06', 180.00, 75);
```

The Messages pane at the bottom shows the execution results:

```
(4 rows affected)  
  
Completion time: 2025-03-09T17:19:02.2987030+03:00
```

The status bar at the bottom indicates 'Query executed successfully.' and '0 rows'.

STEP-3 QUESTIONS

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'TheFirstDatabase'. The main query window contains the following SQL code:

```
INSERT INTO visit (visit_date, price, duration)  
VALUES  
('2020-05-01', 150.00, 60),  
('2020-05-01', 200.50, 90),  
('2020-05-06', 120.75, 45),  
('2024-05-06', 180.00, 75);  
  
-- QUESTION 1) Calculate the number of visitors for each visit date.  
  
SELECT visit_date as "Visit date", COUNT(1) as "Number of visitors"  
FROM visit  
GROUP BY visit_date;
```

The Results pane at the bottom shows the execution results:

Visit date	Number of visitors
2020-05-01	2
2020-05-06	1
2024-05-06	1

The status bar at the bottom indicates 'Query executed successfully.' and '3 rows'.

grouping-sql-examples.sql - 8022NB.TheFirstDatabase (I2ISYSTEMS\okan.cezik (59)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Object Explorer

Connect - 8022NB (SQL Server 16.0.1000.6 - I2ISYSTEMS\okan.cezik (59))

Databases

- System Databases
- Database Snapshots
- AdventureWorks2019
- TheFirstDatabase

Security

- Server Objects
- Replication
- Always On High Availability
- Management

Integration Services Catalogs

- SQL Server Agent (Agent XPs disabled)
- XEvent Profiler

grouping-sql-examp...S\okan.cezik (59)*

```
-- QUESTION 2) Calculate the average price of visits for each year and month.

SELECT
    YEAR(visit_date) AS year,
    MONTH(visit_date) AS month,
    ROUND(AVG(price), 2) AS avg_price
FROM visit
GROUP BY
    YEAR(visit_date),
    MONTH(visit_date);
```

Results

	year	month	avg_price
1	2020	5	157.08
2	2024	5	180.00

Query executed successfully.

8022NB (16.0 RTM) I2ISYSTEMS\okan.cezik ... TheFirstDatabase 00:00:00 2 rows

Ready Ln 26 Col 1 Ch 1 INS

grouping-sql-examples.sql - 8022NB.TheFirstDatabase (I2ISYSTEMS\okan.cezik (59)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Object Explorer

Connect - 8022NB (SQL Server 16.0.1000.6 - I2ISYSTEMS\okan.cezik (59))

Databases

- System Databases
- Database Snapshots
- AdventureWorks2019
- TheFirstDatabase

Security

- Server Objects
- Replication
- Always On High Availability
- Management

Integration Services Catalogs

- SQL Server Agent (Agent XPs disabled)
- XEvent Profiler

grouping-sql-examp...S\okan.cezik (59)*

```
-- QUESTION 3) Calculate the average duration of visits for each year and month.

SELECT
    YEAR(visit_date) AS year,
    MONTH(visit_date) AS month,
    ROUND(AVG(duration), 2) AS avg_duration
FROM visit
GROUP BY
    YEAR(visit_date),
    MONTH(visit_date)
ORDER BY
    YEAR(visit_date),
    MONTH(visit_date);
```

Results

	year	month	avg_duration
1	2020	5	65
2	2024	5	75

Query executed successfully.

8022NB (16.0 RTM) I2ISYSTEMS\okan.cezik ... TheFirstDatabase 00:00:00 2 rows

Ready Ln 37 Col 1 Ch 1 INS

grouping-sql-examples.sql - 8022NB.TheFirstDatabase (I2ISYSTEMS\okan.cezik (59)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Object Explorer

Connect

8022NB (SQL Server 16.0.1000.6 - I2ISYSTEMS\okan.cezik (59))

Databases

- System Databases
- Database Snapshots
- AdventureWorks2019
- TheFirstDatabase

Security

- Server Objects
- Replication
- Always On High Availability
- Management
- Integration Services Catalogs
- SQL Server Agent (Agent XPs disabled)
- XEvent Profiler

grouping-sql-examp...S\okan.cezik (59) *

```
/* QUESTION 4)
Calculate the average price of visits for each date, but only show dates where more than 3 visits were made.
*/
SELECT
    visit_date,
    ROUND(AVG(price), 2) AS avg_price
FROM visit
GROUP BY visit_date
HAVING COUNT(*) > 3
ORDER BY visit_date;
```

100 %

Results Messages

visit_date	avg_price
------------	-----------

Query executed successfully.

8022NB (16.0 RTM) I2ISYSTEMS\okan.cezik ... TheFirstDatabase 00:00:00 0 rows

Ready Ln 54 Col 1 Ch 1 INS

grouping-sql-examples.sql - 8022NB.TheFirstDatabase (I2ISYSTEMS\okan.cezik (59)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Object Explorer

Connect

8022NB (SQL Server 16.0.1000.6 - I2ISYSTEMS\okan.cezik (59))

Databases

- System Databases
- Database Snapshots
- AdventureWorks2019
- TheFirstDatabase

Security

- Server Objects
- Replication
- Always On High Availability
- Management
- Integration Services Catalogs
- SQL Server Agent (Agent XPs disabled)
- XEvent Profiler

grouping-sql-examp...S\okan.cezik (59) *

```
/* QUESTION 5)
Calculate the average duration of visits where the duration is greater than 5 for each date,
but only show dates where more than 3 visits were made.
*/
SELECT
    visit_date,
    ROUND(AVG(duration), 2) AS avg_duration
FROM visit
WHERE duration > 5
GROUP BY visit_date
HAVING COUNT(*) > 3
ORDER BY visit_date;
```

100 %

Results Messages

visit_date	avg_duration
------------	--------------

Query executed successfully.

8022NB (16.0 RTM) I2ISYSTEMS\okan.cezik ... TheFirstDatabase 00:00:00 0 rows

Ready Ln 67 Col 1 Ch 1 INS