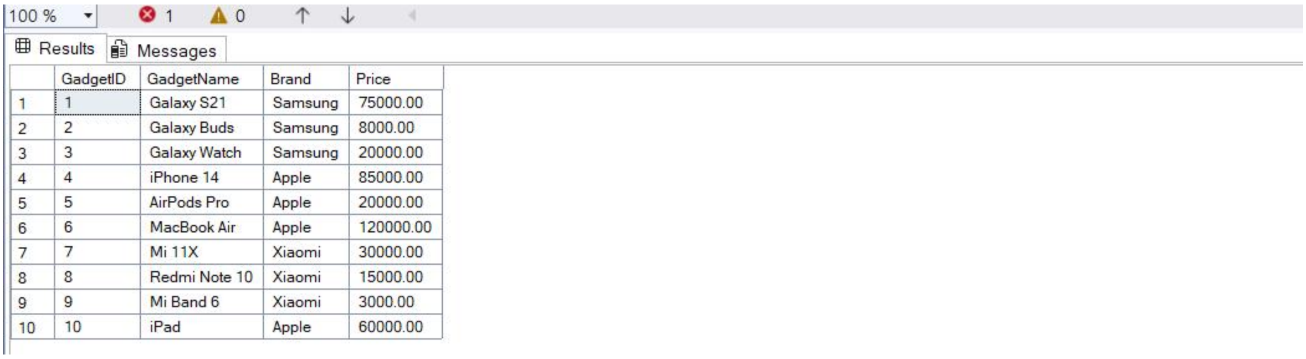
Advanced SQL Exercises for Online Retail Store

Exercise 1: Ranking and Window Functions

Goal: Use ROW_NUMBER(), RANK(), DENSE_RANK(), OVER(), and PARTITION BY.
Scenario:
Find the top 3 most expensive products in each category using different ranking functions.
Steps:
1. Use ROW_NUMBER() to assign a unique rank within each category.
2. Use RANK() and DENSE_RANK() to compare how ties are handled.
3. Use PARTITION BY Category and ORDER BY Price DESC

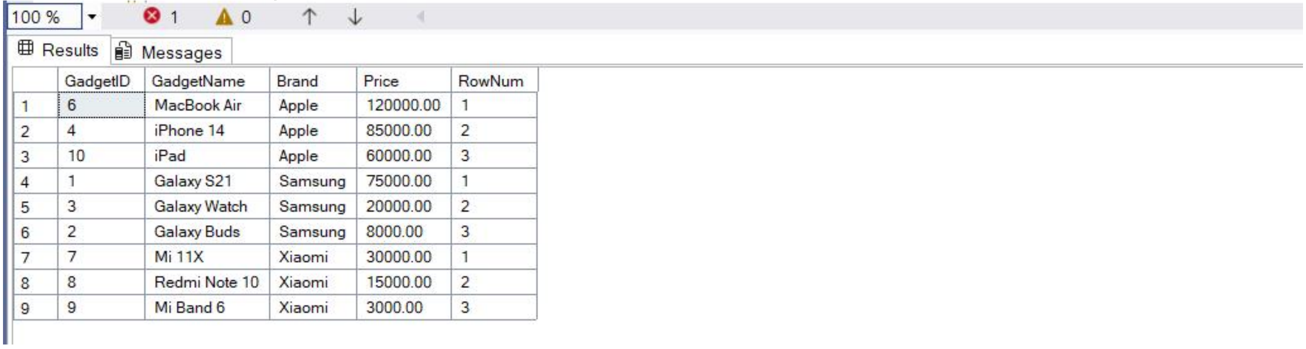
# Creation , innserting and using the database :

```
CREATE DATABASE DOTNETFSE;
USE DOTNETFSE;

CREATE TABLE Gadgets(
    GadgetID INT PRIMARY KEY,
    GadgetName NVARCHAR(100),
    Brand NVARCHAR(50),
    Price DECIMAL(10, 2)
);

INSERT INTO Gadgets (GadgetID, GadgetName, Brand, Price) VALUES
(1, 'Galaxy S21', 'Samsung', 75000),
(2, 'Galaxy Buds', 'Samsung', 8000),
(3, 'Galaxy Watch', 'Samsung', 20000),
(4, 'iPhone 14', 'Apple', 85000),
(5, 'AirPods Pro', 'Apple', 20000),
(6, 'MacBook Air', 'Apple', 120000),
(7, 'Mi 11X', 'Xiaomi', 30000),
(8, 'Redmi Note 10', 'Xiaomi', 15000),
(9, 'Mi Band 6', 'Xiaomi', 3000),
(10, 'iPad', 'Apple', 60000);


SELECT * FROM GADGETS;
```



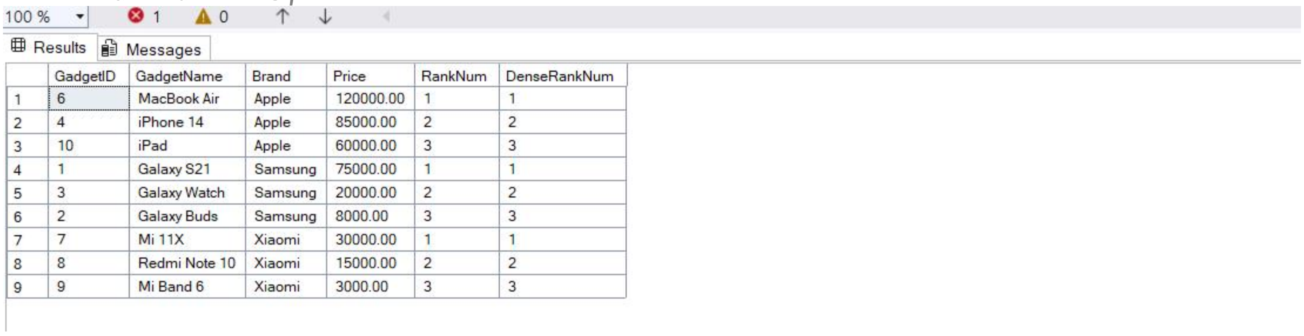
	GadgetID	GadgetName	Brand	Price
1	1	Galaxy S21	Samsung	75000.00
2	2	Galaxy Buds	Samsung	8000.00
3	3	Galaxy Watch	Samsung	20000.00
4	4	iPhone 14	Apple	85000.00
5	5	AirPods Pro	Apple	20000.00
6	6	MacBook Air	Apple	120000.00
7	7	Mi 11X	Xiaomi	30000.00
8	8	Redmi Note 10	Xiaomi	15000.00
9	9	Mi Band 6	Xiaomi	3000.00
10	10	iPad	Apple	60000.00

```
-- . Use ROW_NUMBER() to assign a unique rank within each category --
SELECT *
FROM (
    SELECT
        GadgetID, GadgetName, Brand, Price,
        ROW_NUMBER() OVER (PARTITION BY Brand ORDER BY Price DESC) AS RowNum
    FROM Gadgets
) AS Ranked
WHERE RowNum <= 3;
```



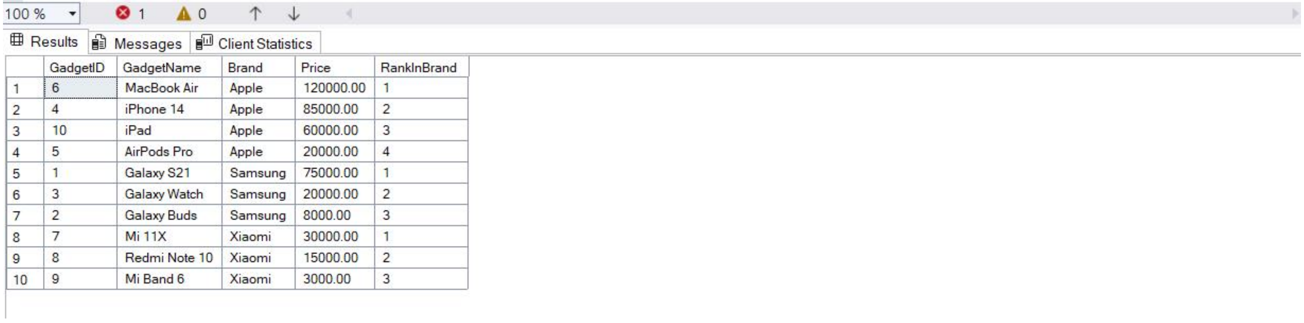
	GadgetID	GadgetName	Brand	Price	RowNum
1	6	MacBook Air	Apple	120000.00	1
2	4	iPhone 14	Apple	85000.00	2
3	10	iPad	Apple	60000.00	3
4	1	Galaxy S21	Samsung	75000.00	1
5	3	Galaxy Watch	Samsung	20000.00	2
6	2	Galaxy Buds	Samsung	8000.00	3
7	7	Mi 11X	Xiaomi	30000.00	1
8	8	Redmi Note 10	Xiaomi	15000.00	2
9	9	Mi Band 6	Xiaomi	3000.00	3

```
-- Use RANK() and DENSE_RANK() to compare how ties are handled
SELECT *
FROM (
    SELECT
        GadgetID, GadgetName, Brand, Price,
        RANK() OVER (PARTITION BY Brand ORDER BY Price DESC) AS RankNum,
        DENSE_RANK() OVER (PARTITION BY Brand ORDER BY Price DESC) AS DenseRankNum
    FROM Gadgets
) AS Ranked
WHERE RankNum <= 3;
```



	GadgetID	GadgetName	Brand	Price	RankNum	DenseRankNum
1	6	MacBook Air	Apple	120000.00	1	1
2	4	iPhone 14	Apple	85000.00	2	2
3	10	iPad	Apple	60000.00	3	3
4	1	Galaxy S21	Samsung	75000.00	1	1
5	3	Galaxy Watch	Samsung	20000.00	2	2
6	2	Galaxy Buds	Samsung	8000.00	3	3
7	7	Mi 11X	Xiaomi	30000.00	1	1
8	8	Redmi Note 10	Xiaomi	15000.00	2	2
9	9	Mi Band 6	Xiaomi	3000.00	3	3

```
--Use PARTITION BY Category and ORDER BY Price DESC
SELECT
    GadgetID,GadgetName,Brand,Price,
    RANK() OVER (PARTITION BY Brand ORDER BY Price DESC) AS RankInBrand
FROM
Gadgets;
```



	GadgetID	GadgetName	Brand	Price	RankInBrand
1	6	MacBook Air	Apple	120000.00	1
2	4	iPhone 14	Apple	85000.00	2
3	10	iPad	Apple	60000.00	3
4	5	AirPods Pro	Apple	20000.00	4
5	1	Galaxy S21	Samsung	75000.00	1
6	3	Galaxy Watch	Samsung	20000.00	2
7	2	Galaxy Buds	Samsung	8000.00	3
8	7	Mi 11X	Xiaomi	30000.00	1
9	8	Redmi Note 10	Xiaomi	15000.00	2
10	9	Mi Band 6	Xiaomi	3000.00	3