

**Київський національний університет імені
Тараса Шевченка Факультет комп'ютерних наук
та кібернетики**

Звіт

До лабораторної роботи № 1

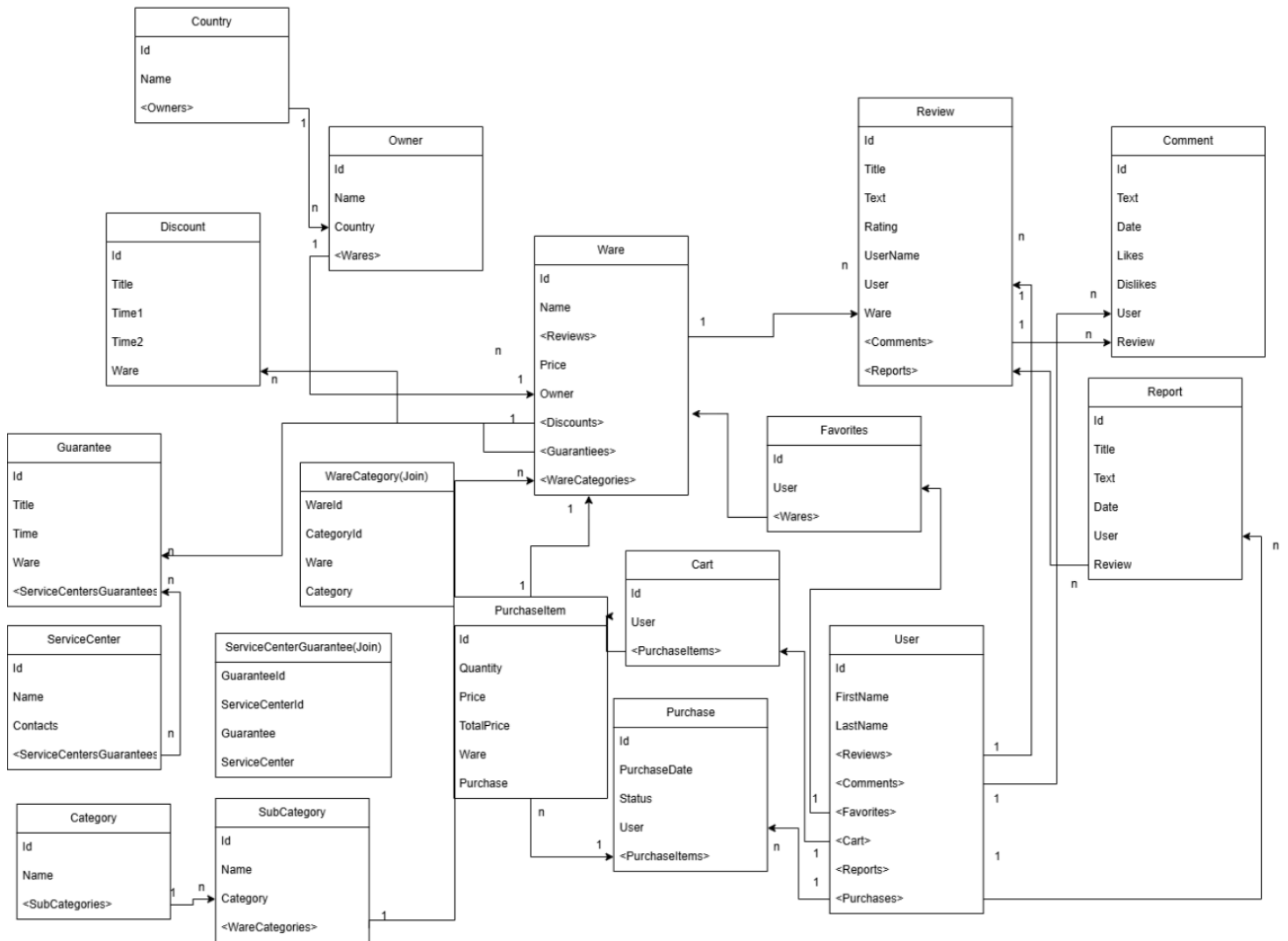
З навчальної дисципліни

«Сучасні технології баз даних»

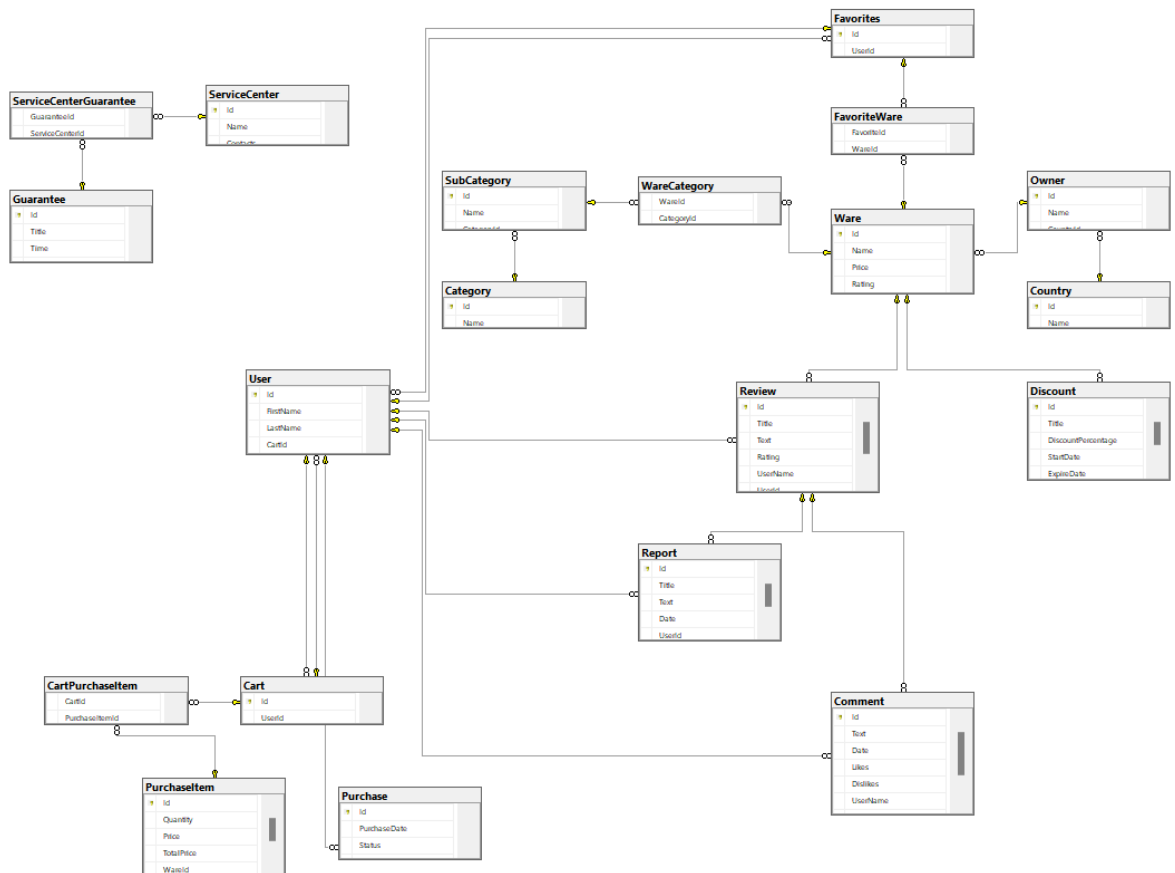
Студента 3 курсу
групи ТК-32
Войцішеного
Боромира

Київ - 2024

Діаграма бази даних, розроблена для розуміння логіки праці інтернет магазину. (*draw.io*)



Модель, реалізована у MsSql:



Ось пояснення до таблиць у базі даних, згідно зі схемою:

1. ServiceCenter

Зберігає інформацію про сервісні центри.

Поля:

- **Id**: Унікальний ідентифікатор сервісного центру.
- **Name**: Назва сервісного центру.
- **City**: Місто, де розташований сервісний центр.

2. Guarantee

Інформація про гарантії для товарів.

Поля:

- **Id**: Унікальний ідентифікатор гарантії.

- **Title**: Назва гарантії.
- **Time**: Тривалість гарантії (в місяцях).

3. **ServiceCenterGuarantee**

Зв'язує гарантії з сервісними центрами.

Поля:

- **GuaranteeId**: Посилання на таблицю **Guarantee**.
- **ServiceCenterId**: Посилання на таблицю **ServiceCenter**.

4. **Category**

Зберігає інформацію про категорії товарів.

Поля:

- **Id**: Унікальний ідентифікатор категорії.
- **Name**: Назва категорії.

5. **SubCategory**

Підкатегорії для товарів.

Поля:

- **Id**: Унікальний ідентифікатор підкатегорії.
- **Name**: Назва підкатегорії.
- **CategoryId**: Посилання на таблицю **Category**.

6. **WareCategory**

Зв'язок між товарами та їх категоріями.

Поля:

- **WareId**: Посилання на таблицю **Ware**.
- **CategoryId**: Посилання на таблицю **Category**.

7. **Ware**

Інформація про товари.

Поля:

- **Id**: Унікальний ідентифікатор товару.
- **Name**: Назва товару.
- **Price**: Ціна товару.

8. **Owner**

Інформація про власників товарів.

Поля:

- **Id**: Унікальний ідентифікатор власника.
- **Name**: Ім'я власника.
- **CountryId**: Посилання на таблицю **Country**.

9. **Country**

Інформація про країни.

Поля:

- **`Id`**: Унікальний ідентифікатор країни.
- **`Name`**: Назва країни.

10. **Discount**

Інформація про знижки.

Поля:

- **`Id`**: Унікальний ідентифікатор знижки.
- **`Title`**: Назва знижки.
- **`DiscountPercentage`**: Відсоток знижки.
- **`StartDate`**: Дата початку дії знижки.
- **`ExpireDate`**: Дата завершення дії знижки.

11. **User**

Інформація про користувачів.

Поля:

- **`Id`**: Унікальний ідентифікатор користувача.
- **`FirstName`**: Ім'я користувача.
- **`LastName`**: Прізвище користувача.
- **`CartId`**: Посилання на таблицю **`Cart`**.

12. **Cart**

Кошик покупок користувача.

Поля:

- **`Id`**: Унікальний ідентифікатор кошика.
- **`UserId`**: Посилання на таблицю **`User`**.

13. **CartPurchaseItem**

Зв'язок між кошиком і товарами, які додаються до нього.

Поля:

- **`CartId`**: Посилання на таблицю **`Cart`**.
- **`PurchaseItemId`**: Посилання на таблицю **`PurchaseItem`**.

14. **PurchaseItem**

Інформація про товари в покупках.

Поля:

- **`Id`**: Унікальний ідентифікатор товару в покупці.
- **`Quantity`**: Кількість товару.
- **`Price`**: Ціна одиниці товару.
- **`TotalPrice`**: Загальна вартість.
- **`WareId`**: Посилання на таблицю **`Ware`**.

15. **Purchase**

Інформація про покупки.

Поля:

- **Id**: Унікальний ідентифікатор покупки.
- **PurchaseDate**: Дата покупки.
- **Status**: Статус покупки.

16. **Favorites**

Зберігає списки улюблених товарів користувача.

Поля:

- **Id**: Унікальний ідентифікатор списку.
- **UserId**: Посилання на таблицю **User**.

17. **FavoriteWare**

Зв'язок між списком улюблених і товарами.

Поля:

- **FavoriteId**: Посилання на таблицю **Favorites**.
- **WareId**: Посилання на таблицю **Ware**.

18. **Review**

Огляди на товари.

Поля:

- **Id**: Унікальний ідентифікатор огляду.
- **Title**: Заголовок огляду.
- **Text**: Текст огляду.
- **Rating**: Оцінка товару.
- **UserName**: Ім'я користувача, який залишив огляд.

19. **Comment**

Коментарі до товарів чи оглядів.

Поля:

- **Id**: Унікальний ідентифікатор коментаря.
- **Text**: Текст коментаря.
- **Date**: Дата створення коментаря.
- **Likes**: Кількість вподобань.
- **Dislikes**: Кількість дизлайків.
- **UserName**: Ім'я користувача, який залишив коментар.

20. **Report**

Звіти про скарги чи проблеми.

Поля:

- **Id**: Унікальний ідентифікатор звіту.
- **Title**: Заголовок звіту.
- **Text**: Опис проблеми.
- **UserId**: Посилання на таблицю **User**.

Процедури, користувацькі функції, тригери

Процедури

- [-] Programmability
 - [-] Stored Procedures
 - + System Stored Procedures
 - + [SQL] dbo.ActivateEntity
 - + [SQL] dbo.AddCartPurchaseItem
 - + [SQL] dbo.AddCategory
 - + [SQL] dbo.AddComment
 - + [SQL] dbo.AddCountry
 - + [SQL] dbo.AddDiscount
 - + [SQL] dbo.AddFavoriteWare
 - + [SQL] dbo.AddGuarantee
 - + [SQL] dbo.AddOwner
 - + [SQL] dbo.AddPurchase
 - + [SQL] dbo.AddReport
 - + [SQL] dbo.AddReview
 - + [SQL] dbo.AddServiceCenter
 - + [SQL] dbo.AddServiceCenterGuaran
 - + [SQL] dbo.AddSubCategory
 - + [SQL] dbo.AddUser
 - + [SQL] dbo.AddWare
 - + [SQL] dbo.AddWareCategory

1) AddCartPurchaseItem

```
USE [InternetShop]
GO
/***** Object:  StoredProcedure [dbo].[AddCartPurchaseItem]
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[AddCartPurchaseItem]
    @CartId INT,
    @PurchaseItemId INT
AS
BEGIN
    INSERT INTO CartPurchaseItem (CartId, PurchaseItemId)
    VALUES (@CartId, @PurchaseItemId);
END;
```

2) AddCategory

```
USE [InternetShop]
GO
/***** Object:  StoredProcedure [dbo].[AddCategory]
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[AddCategory]
    @Id INT,
    @Name NVARCHAR(50)
AS
BEGIN
    INSERT INTO Category (Id, Name)
    VALUES (@Id, @Name);
END;
```

3) AddComment


```

USE [InternetShop]
GO
/***** Object:  StoredProcedure [dbo].[AddComment]    Script Date: 21.11.2024 21
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[AddComment]
    @Id INT,
    @Text NVARCHAR(MAX),
    @Date DATE,
    @Likes INT,
    @Dislikes INT,
    @UserName NVARCHAR(50),
    @UserId INT,
    @ReviewId INT
AS
BEGIN
    INSERT INTO Comment (Id, Text, Date, Likes, Dislikes, UserName, UserId, ReviewId)
    VALUES (@Id, @Text, @Date, @Likes, @Dislikes, @UserName, @UserId, @ReviewId);
END;

```

4) AddCountry

```

USE [InternetShop]
GO
/***** Object:  StoredProcedure [dbo].[AddCountry]
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[AddCountry]
    @Id INT,
    @Name NVARCHAR(50)
AS
BEGIN
    INSERT INTO Country (Id, Name)
    VALUES (@Id, @Name);
END;

```

5) AddDiscount

```

USE [InternetShop]
GO
/***** Object:  StoredProcedure [dbo].[AddDiscount]      Script Date: 21.11.2024 21
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[AddDiscount]
    @Id INT,
    @Title NVARCHAR(50),
    @DiscountPercentage DECIMAL(2, 2),
    @StartDate DATE,
    @ExpireDate DATE,
    @WareId INT
AS
BEGIN
    INSERT INTO Discount (Id, Title, DiscountPercentage, StartDate, ExpireDate, WareId)
    VALUES (@Id, @Title, @DiscountPercentage, @StartDate, @ExpireDate, @WareId);
END;

```

6) AddFavoriteWare

```

USE [InternetShop]
GO
/***** Object:  StoredProcedure [dbo].[AddFavoriteWare]      Script Date: 21.11.2024 21
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[AddFavoriteWare]
    @FavoriteId INT,
    @WareId INT
AS
BEGIN
    INSERT INTO FavoriteWare (FavoriteId, WareId)
    VALUES (@FavoriteId, @WareId);
END;

```

7) AddGuarantee

```

USE [InternetShop]
GO
/***** Object:  StoredProcedure [dbo].[AddGuarantee]      s
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[AddGuarantee]
    @Id INT,
    @Title NVARCHAR(100),
    @Time INT,
    @WareId INT
AS
BEGIN
    INSERT INTO Guarantee (Id, Title, Time, WareId)
    VALUES (@Id, @Title, @Time, @WareId);
END;

```

8) AddOwner

```

USE [InternetShop]
GO
/***** Object:  StoredProcedure [dbo].[AddOwner]
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[AddOwner]
    @Id INT,
    @Name NVARCHAR(50),
    @CountryId INT
AS
BEGIN
    INSERT INTO Owner (Id, Name, CountryId)
    VALUES (@Id, @Name, @CountryId);
END;

```

9) AddPurchase

```

USE [InternetShop]
GO
/***** Object:  StoredProcedure [dbo].[AddPurchase]    Script Da
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[AddPurchase]
    @Id INT,
    @PurchaseDate DATE,
    @Status INT,
    @UserId INT
AS
BEGIN
    INSERT INTO Purchase (Id, PurchaseDate, Status, UserId)
    VALUES (@Id, @PurchaseDate, @Status, @UserId);
END;

```

10) AddReport

```

USE [InternetShop]
GO
/***** Object:  StoredProcedure [dbo].[AddReport]    Script Date: 21.
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[AddReport]
    @Id INT,
    @Title NVARCHAR(50),
    @Text NVARCHAR(MAX),
    @Date DATETIME,
    @UserId INT,
    @ReviewId INT
AS
BEGIN
    INSERT INTO Report (Id, Title, Text, Date, UserId, ReviewId)
    VALUES (@Id, @Title, @Text, @Date, @UserId, @ReviewId);
END;

```

11) AddReview

```

USE [InternetShop]
GO
/***** Object:  StoredProcedure [dbo].[AddReview]      Script Date: 21.11.202
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[AddReview]
    @Id INT,
    @Title NVARCHAR(50),
    @Text NVARCHAR(MAX),
    @Rating INT,
    @UserName NVARCHAR(50),
    @UserId INT,
    @WareId INT
AS
BEGIN
    INSERT INTO Review (Id, Title, Text, Rating, UserName, UserId, WareId)
    VALUES (@Id, @Title, @Text, @Rating, @UserName, @UserId, @WareId);
END;

```

12) AddServiceCenter

```

SQLQuery54.sql - P...AMING\bons (105)  SQLQuery53.sql - P...AMING\bons (10
USE [InternetShop]
GO
/***** Object:  StoredProcedure [dbo].[AddServiceCenter]
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[AddServiceCenter]
    @Id INT,
    @Name NVARCHAR(50),
    @Contacts NVARCHAR(MAX)
AS
BEGIN
    INSERT INTO ServiceCenter (Id, Name, Contacts)
    VALUES (@Id, @Name, @Contacts);
END;

```

13) AddServiceCenterGuarantee

```

USE [InternetShop]
GO
/***** Object:  StoredProcedure [dbo].[AddServiceCenterGuarantee] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[AddServiceCenterGuarantee]
    @GuaranteeId INT,
    @ServiceCenterId INT
AS
BEGIN
    INSERT INTO ServiceCenterGuarantee (GuaranteeId, ServiceCenterId)
    VALUES (@GuaranteeId, @ServiceCenterId);
END;

```

14) AddSubCategory

```

USE [InternetShop]
GO
/***** Object:  StoredProcedure [dbo].[AddSubCategory] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[AddSubCategory]
    @Id INT,
    @Name NVARCHAR(50),
    @CategoryId INT
AS
BEGIN
    INSERT INTO SubCategory (Id, Name, CategoryId)
    VALUES (@Id, @Name, @CategoryId);
END;

```

15) AddUser

```

USE [InternetShop]
GO
/***** Object:  StoredProcedure [dbo].[AddUser]    Script Date: 21.11.2017 14:56:12 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[AddUser]
    @Id INT,
    @FirstName NVARCHAR(50),
    @LastName NVARCHAR(50),
    @CartId INT,
    @FavoriteId INT
AS
BEGIN
    INSERT INTO [User] (Id, FirstName, LastName, CartId, FavoriteId)
    VALUES (@Id, @FirstName, @LastName, @CartId, @FavoriteId);
END;

```

16) AddWare

```

USE [InternetShop]
GO
/***** Object:  StoredProcedure [dbo].[AddWare]    Script Date: 21.11.2017 14:56:12 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[AddWare]
    @Id INT,
    @Name VARCHAR(50),
    @Price DECIMAL(18, 2),
    @Rating INT,
    @OwnerId INT
AS
BEGIN
    INSERT INTO Ware(Id, Name, Price, Rating, OwnerId)
    VALUES(@Id, @Name, @Price, @Rating, @OwnerId);
END;

```

17) AddWareCategory

```
USE [InternetShop]
GO
/***** Object:  StoredProcedure [dbo].[AddWareCategory]
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[AddWareCategory]
    @WareId INT,
    @CategoryId INT
AS
BEGIN
    INSERT INTO WareCategory (WareId, CategoryId)
    VALUES (@WareId, @CategoryId);
END;
```

Давайте подивимось на приклад виконання однієї з процедур:

PavillionGaming\S...etShop - dbo.Owner					
	Id	Name	CountryId	IsDeleted	LastModify...
▶	1	TechStore	1	False	2024-11-21 ...
	2	GadgetPro	2	False	2024-11-21 ...
	3	HomeAppli...	3	False	2024-11-21 ...
*	NULL	NULL	NULL	NULL	NULL

Рис.1 – до виконання процедури

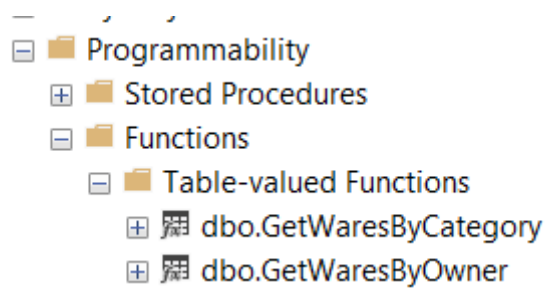
Виконаємо:

```
EXEC AddOwner
    @Id = 4,
    @Name = 'IVANKO',
    @CountryId = 3;
```

PavillionGaming\S...etShop - dbo.Owner					
SQLQuery64.sql - P...GAMING\boris					
	Id	Name	CountryId	IsDeleted	LastModify...
▶	1	TechStore	1	False	2024-11-21 ...
	2	GadgetPro	2	False	2024-11-21 ...
	3	HomeAppli...	3	False	2024-11-21 ...
	4	IVANKO	3	False	2024-11-21 ...
*	NULL	NULL	NULL	NULL	NULL

Рис.2 – після виконання процедури

Користувацькі функції



1) GetWaresByCategory

```

USE [InternetShop]
GO
/***** Object: UserDefinedFunction [dbo].[GetWaresByCategory]
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER FUNCTION [dbo].[GetWaresByCategory] (@CategoryId INT)
RETURNS TABLE
AS
RETURN
(
    SELECT
        Ware.Id AS WareId,
        Ware.Name AS WareName,
        Ware.Price AS WarePrice
    FROM
        Ware
    INNER JOIN WareCategory
        ON Ware.Id = WareCategory.WareId
    WHERE
        WareCategory.CategoryId = @CategoryId
);

```

2) GetWaresByOwner

```

USE [InternetShop]
GO
/***** Object:  UserDefinedFunction [dbo].[GetWaresByOwner]
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER FUNCTION [dbo].[GetWaresByOwner](@OwnerId INT)
RETURNS TABLE
AS
RETURN
(
    SELECT
        Ware.Id AS WareId,
        Ware.Name AS WareName,
        Ware.Price AS WarePrice
    FROM
        Ware
    WHERE
        Ware.OwnerId = @OwnerId
);

```

Давайте виконаємо одну функцію для прикладу.

SQLQuery72.sql - P...GAMING\boris (55))* ✕

```

-- Передбачається, що Owner з Id = 1 існує
SELECT * FROM dbo.GetWaresByOwner(1);

```

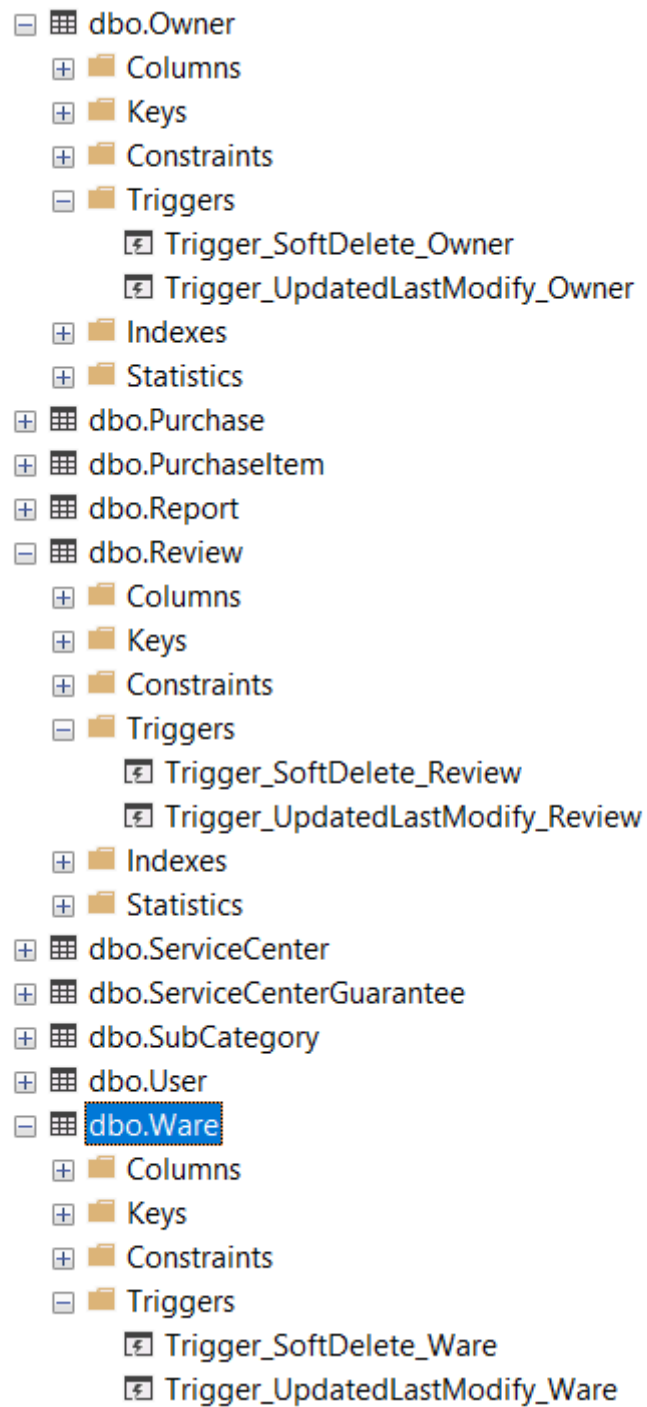
100 %

Results Messages

	WareId	WareName	WarePrice
1	1	Updated Ware	999.99
2	2	MacBook Pro	1999.99

PavillionGaming\S...netShop - dbo.Ware SQLQuery72.sql - P...GAMING\boris (55))*							
	Id	Name	Price	Rating	OwnerId	IsDeleted	LastMod
▶	1	Updated W...	999,99	4	1	False	2024-11-
	2	MacBook Pro	1999,99	4	1	False	2024-11-
	3	Samsung G...	899,99	4	2	False	2024-11-
	4	LG Washing ...	599,99	4	3	False	2024-11-
	5	Modern Sofa	499,99	4	3	False	2024-11-
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Тригери



1) SoftDelete (на прикладі Ware)

```

USE [InternetShop]
GO
/***** Object: Trigger [dbo].[Trigger_SoftDelete_Ware]
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER TRIGGER [dbo].[Trigger_SoftDelete_Ware]
ON [dbo].[Ware]
INSTEAD OF DELETE
AS
BEGIN
    -- Оновлення IsDeleted замість видалення запису
    UPDATE Ware
    SET IsDeleted = 1
    FROM Ware
    INNER JOIN deleted ON Ware.Id = deleted.Id;
END;

```

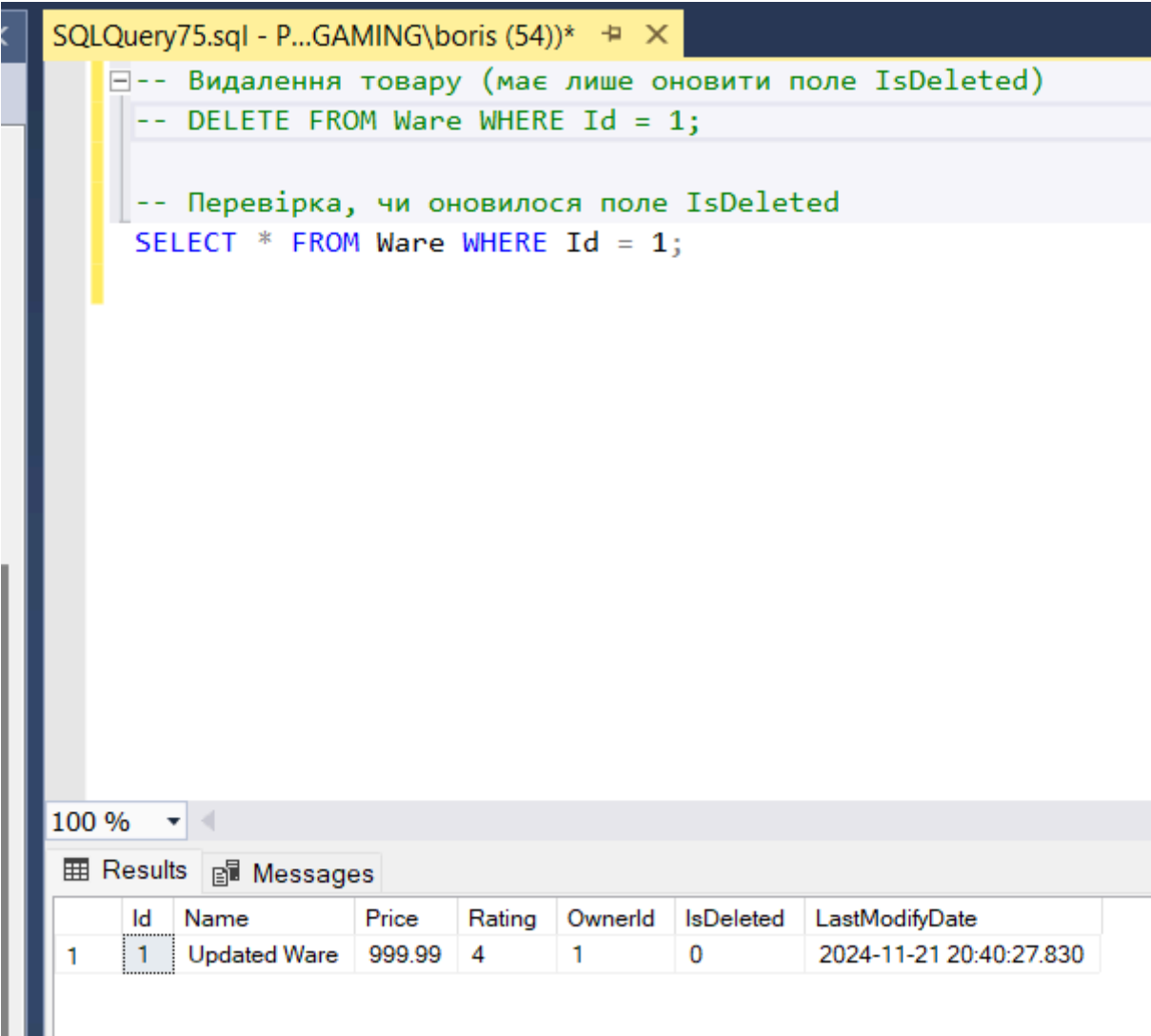
2) UpdatedLastModify (на прикладі Ware)

```

USE [InternetShop]
GO
/***** Object: Trigger [dbo].[Trigger_UpdatedLastModify_Ware]
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER TRIGGER [dbo].[Trigger_UpdatedLastModify_Ware]
ON [dbo].[Ware]
AFTER UPDATE
AS
BEGIN
    -- Оновлення поля LastModifyDate при зміні запису
    UPDATE Ware
    SET LastModifyDate = GETDATE()
    FROM Ware
    INNER JOIN inserted ON Ware.Id = inserted.Id;
END;

```

Виконаємо один з тригерів, наприклад, «м'яке видалення» для терміналу.



The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows a script named 'SQLQuery75.sql' with the following T-SQL code:

```
-- Видалення товару (має лише оновити поле IsDeleted)
-- DELETE FROM Ware WHERE Id = 1;

-- Перевірка, чи оновилося поле IsDeleted
SELECT * FROM Ware WHERE Id = 1;
```

The bottom pane shows the 'Results' tab with a table containing one row of data:

	Id	Name	Price	Rating	OwnerId	IsDeleted	LastModifyDate
1	1	Updated Ware	999.99	4	1	0	2024-11-21 20:40:27.830

Рис.6 – таблиця до виконання

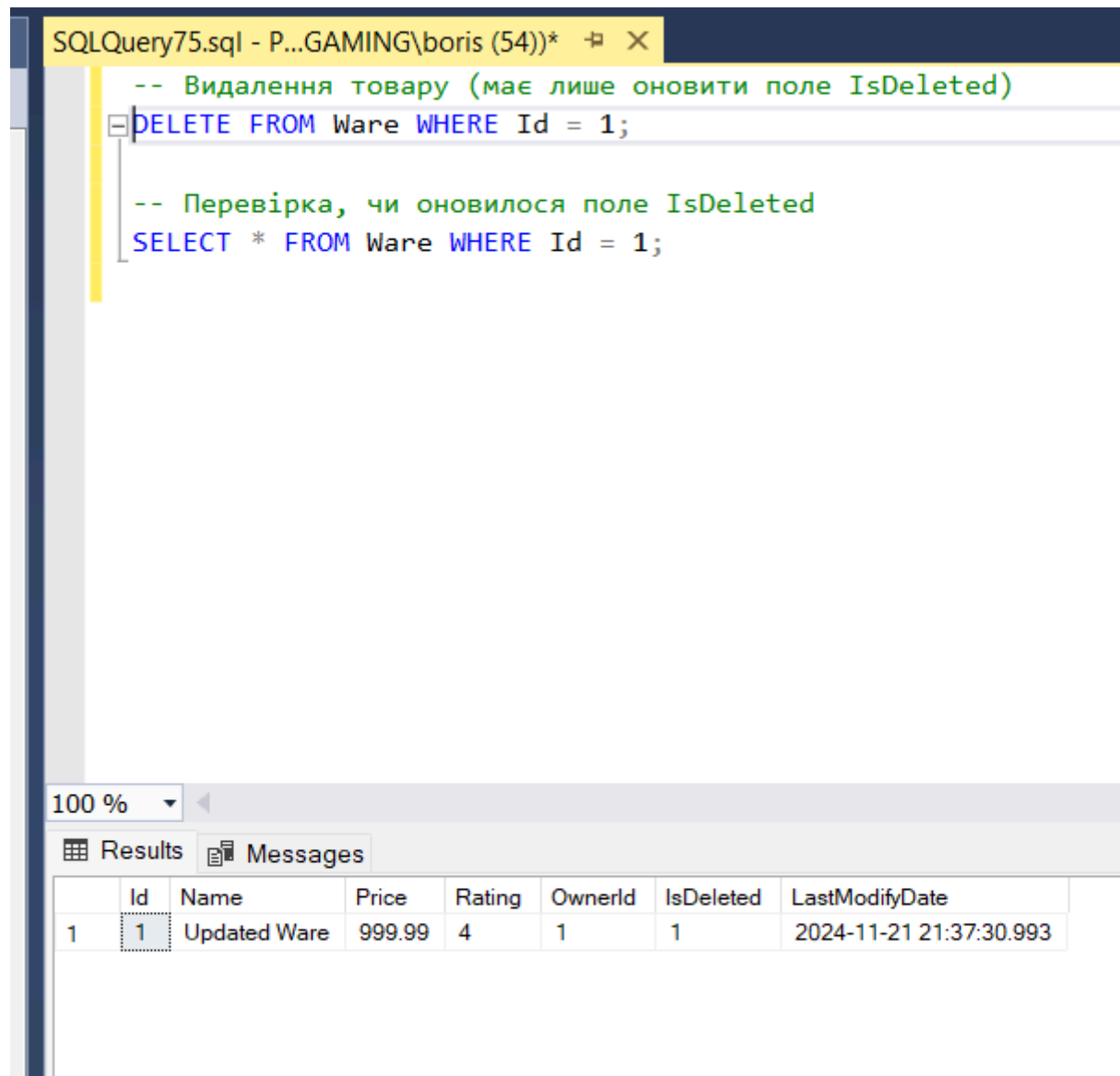


Рис.7 – таблиця після виконання

Як бачимо наш запис не видалився, а просто змінилось значення.

Тепер виконаємо тригер на LastMofidy

SQLQuery75.sql - P...GAMING\boris (54))*

```
-- Оновлення товару, щоб перевірити зміну поля LastModified  
-- UPDATE Ware SET Name = 'Updated Ware' WHERE Id = 1;  
  
-- Перевірка поля LastModified  
SELECT * FROM Ware WHERE Id = 1;
```

100 %

Results Messages

	Id	Name	Price	Rating	OwnerId	IsDeleted	LastModifyDate
1	1	Updated Ware	999.99	4	1	1	2024-11-21 21:37:30.993

Рис.8 – таблиця

Дивимось результат в таблиці

The screenshot shows a SQL Server Enterprise Manager interface. At the top, a query window titled 'SQLQuery75.sql - P...GAMING\boris (54))*' contains the following SQL code:

```
-- Оновлення товару, щоб перевірити зміну поля LastModified
UPDATE Ware SET Name = 'Updated Ware' WHERE Id = 1;

-- Перевірка поля LastModified
SELECT * FROM Ware WHERE Id = 1;
```

Below the query window, the 'Results' tab is active, displaying a table with the following data:

	Id	Name	Price	Rating	OwnerId	IsDeleted	LastModifyDate
1	1	Updated Ware	999.99	4	1	1	2024-11-21 21:39:28.820

Рис.9 – результат виконання

От ми видалили і замінили потрібні об'єкти, а якщо нам треба повернути їх статус в активний, але ми не хочемо лізти в таблиці, це буде незручно якщо в нас велика база даних.

Для цього в нас є ще одна *збережена* процедура.

1) ActivateEntity

```
USE [InternetShop]
GO
/***** Object:  StoredProcedure [dbo].[ActivateEntity]
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[ActivateEntity]
    @TableName NVARCHAR(50),
    @Id INT
AS
BEGIN
    DECLARE @SQL NVARCHAR(MAX);

    SET @SQL = '
        UPDATE ' + @TableName + '
        SET IsDeleted = 0
        WHERE Id = @Id AND IsDeleted = 1';

    EXEC sp_executesql @SQL, N'@Id INT', @Id;
END;
```

Давайте спробуємо виконати.

SQLQuery76.sql - P...GAMING\boris (78))

SQLQuery75.sql - P...GAMING\boris (54))*

SELECT * FROM Ware;

100 %

ResultsMessages

	Id	Name	Price	Rating	OwnerId	IsDeleted	LastModifyDate
1	1	iPhone 13	999.99	4	1	1	2024-11-21 21:43:37.660
2	2	MacBook Pro	1999.99	4	1	0	2024-11-21 20:02:08.900
3	3	Samsung Galaxy	899.99	4	2	0	2024-11-21 20:02:08.900
4	4	LG Washing Machine	599.99	4	3	0	2024-11-21 20:02:08.900
5	5	Modern Sofa	499.99	4	3	0	2024-11-21 20:02:08.900

Рис.10 – таблиця до процедури

Змінимо статус товару з Id = 1, виконавши команду

SQLQuery76.sql - P...GAMING\boris (78)SQLQuery75.sql - P...GAMING\boris (54))*

EXEC ActivateEntity 'Ware', 1;

SELECT * FROM Ware;

100 %

ResultsMessages

	Id	Name	Price	Rating	OwnerId	IsDeleted	LastModifyDate
1	1	iPhone 13	999.99	4	1	0	2024-11-21 21:47:05.143
2	2	MacBook Pro	1999.99	4	1	0	2024-11-21 20:02:08.900
3	3	Samsung Galaxy	899.99	4	2	0	2024-11-21 20:02:08.900
4	4	LG Washing Machine	599.99	4	3	0	2024-11-21 20:02:08.900
5	5	Modern Sofa	499.99	4	3	0	2024-11-21 20:02:08.900

Рис.11 – результат виконання процедури

Розрізи даних(Views)

- Views
- System Views

dbo.UserPurchases

dbo.WaresWithCategories

SQLQuery76.sql - P...GAMING\boris (78))

SQLQuery75.sql - P...GAMING\boris (54))*

SELECT * FROM UserPurchases;

100 %

Results

Messages

	UserId	FirstName	LastName	PurchaseDate	PurchaseStatus	WareName	WarePrice
1	1	John	Doe	2024-11-20 00:00:00.000	2	iPhone 13	999.99
2	1	John	Doe	2024-11-21 00:00:00.000	1	MacBook Pro	1999.99
3	2	Jane	Smith	2024-11-22 00:00:00.000	0	Samsung Galaxy	899.99

Рис.12 – результат виконання UserPurchases

SQLQuery76.sql - P...GAMING\boris (78))

SQLQuery75.sql - P...GAMING\boris (54))*

SELECT * FROM WaresWithCategories;

100 %

Results

Messages

	WareId	WareName	WarePrice	CategoryName	SubCategoryName
1	1	iPhone 13	999.99	Electronics	Smartphones
2	1	iPhone 13	999.99	Electronics	Laptops
3	2	MacBook Pro	1999.99	Home Appliances	Washing Machines
4	3	Samsung Galaxy	899.99	Electronics	Smartphones
5	3	Samsung Galaxy	899.99	Electronics	Laptops
6	4	LG Washing Machine	599.99	Furniture	Sofas

Рис.13 – результат виконання WaresWithCategories