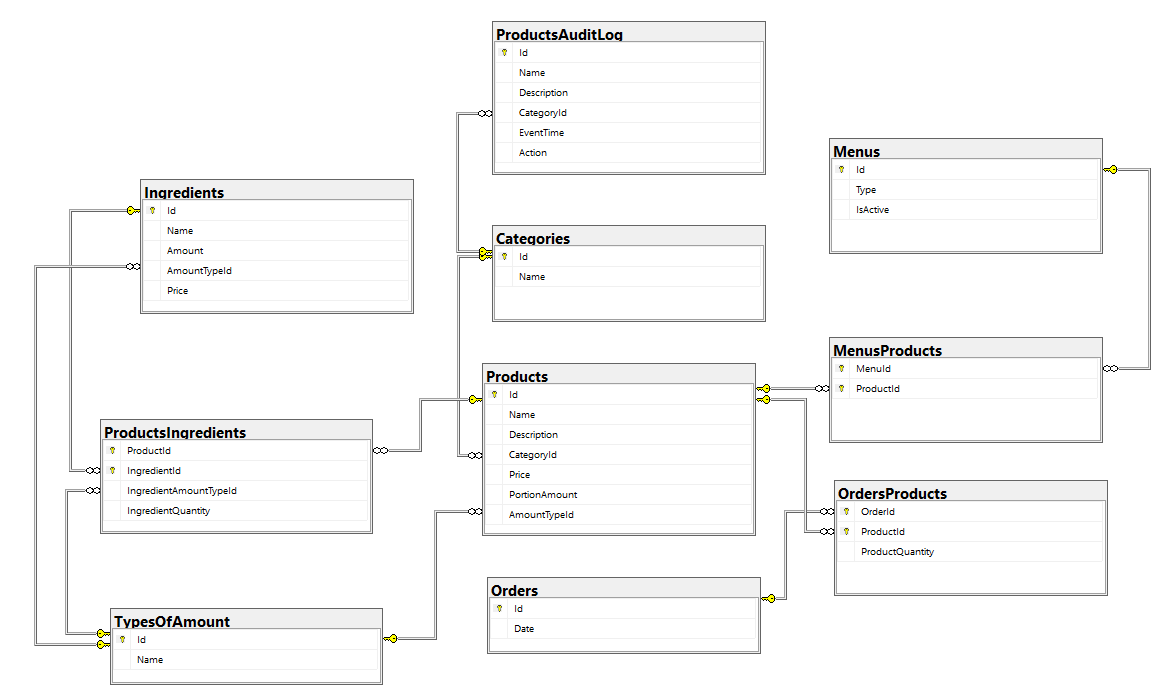
Project Database

**Restaurant**

**Съдържание**

* Таблици
* Ограничения
* Процес на добавяне на съдържание
* Видове заявки
* Тригери
* Индекси
* Изгледи

**Таблиците и техните връзки**

****

Menus – таблица съдържаща различните менюта. Те ще имат различни типове и може да са активни или да не са. Идентификационният номер на менюто се свързва с foreign key-а MenuId от таблица MenusProducts.

MenusProducts – таблица, чрез която е осъществена връзка много към много между менютата и техните елементи (Products).

Products – таблица с елементите (ястия и напитки) от менюто. Всеки продукт си има име, кратко описание, категория, цена, количество на порцията и тип на порцията (gr, ml, pieces).

Categories – таблица в която са изнесени категориите на Products с цел да се спести памет от string-овете, с които се изписват имената на категориите.

Orders – таблица с поръчки и тяхната дата.

OrdersProducts – таблица, чрез която е осъществена връзка много към много между поръчките и елементи от менюто. Тази таблица позволява в една поръчка да има много продукти и един продукт да е в много поръчки, като чрез колона ProductQuantity се разбира колко от дадения продукт в дадената поръчка има.

Ingredients – таблица със съставките, които има в склада (домати, сол и пр.). Всяка съставка има име, налично количество, тип на количеството и цена за килограм.

ProductsIngredients – таблица, чрез която е осъществена връзка много към много между продуктите(елементите на менюто) и техните съставки. Същото така са добавени колони IngredientQuantity и IngredientTypeId, чрез които се разбира от всяка съставка колко се използва за всеки от продуктите, в които участва.

TypesOfAmount – подобно на таблица Categories, тази таблица е с цел пестене на памет от string-ове. Използвана е за да се изведе информацията за типа на количеството от Products, ProductsIngredients и Ingredients.

ProductsAuditLog – Таблица, чрез която бихме могли да получим информация за това кога е добавен или премахнат даден продукт. Всеки одит си има уникален идентификационен номер, име на продукт, кратко описание, категория, дата на промяна и каква е извършената промяна, съответно (insert/delete/update)

Всички колони, които са с числови стойности и не бива да са отрицателни, са с поставени съотвени ограничения. Повече за ограничения - малко по-надолу.

**Ограничения**

Ограниченията, които са добавени, са от следния тип:

1. PRIMARY KEY & FOREIGN KEY - могат да се видят на диаграмата
2. CHECK, който не позволява в базата отрицателни записи на следните колони:
   1. PortionAmount, Price в таблицата Products;
   2. ProductQuantity в таблицата OrdersProducts;
   3. Amount, Price в таблицата Ingredients;
   4. IngredientQuantity в таблицата ProductsIngredients;
3. Ограничение на дължината на символен низ:
   1. Description в таблицата Products, 120 символа;
   2. Name в таблицата Products, 50 символа;
   3. Name в таблицата Categories, 20 символа;
   4. Name в таблицата TypesOfAmount, 20 символа;
   5. Name в таблицата Products, 50 символа;
   6. Type в таблицата Menus, 30 символа;
4. NOT NULL ограничения - навсякъде без Id колоните;

**Добавяне на съдържание на таблиците.**

Процесът на добавяне започна първо с добавянето на основните категории продукти в таблицата Categories, като за всяка категория се добавя ключ(id) и име, които са съответно: 1 - Main Dish, 2 - Salad, 3 - Starter, 4 - Pasta and Risotto, 5 -Pizza, 6 - Sushi, 7 - Dessert, 8 - Drink, 9 - Soups. Например :

INSERT INTO Categories

Values(2,'Salad');

В този пример се добави категория с ключ(id) 2 и име ‘Salad’.

След това се добавиха основните типове на количество в таблицата TypesOfAmount, в което може да са съставките, а именно: 1 - grams, 2 - milliliters, 3 - units. Например:

INSERT INTO TypesOfAmount

Values(1,'grams');

В този пример се добави тип с ключ(id) 1, който е ‘grams’.

След това се добавиха основните съставки в таблицата Ingredients, от които могат да се състоят нашите продукти. Като за всяка съставка се добави ключ(id), име, наличност към момента в склада, в какъв тип се измерва количеството и, както и колко е цената и. Трябва да се отбележи, че цената за продуктите, които се измерват съответно в грамове и милилитри е цената за един килограм и един литър съответно. Например:

INSERT INTO Ingredients

Values(1,'tomatoes',5000,1,2);

В този пример ключът(id) е 1, името е “tomatoes”, наличното количество е 5000 грама към момента в склада, измерва се в грамове и цената за един килограм е 2 лева.

След това се добавиха различните продукти в таблицата Products, като при добавянето на всеки продукт се задава ключ(id) на продукта, име на продукта, съдържанието на продукта, категорията на продукта чрез нейният ключ(id), цената на продукта, количеството на продукта и типа на количеството, в което е продукта чрез неговият ключ(id). Например:

INSERT INTO Products

Values (6,'Chicken Soup','chicken, peppers, potatoes, carrots, onion, parsley',9,4.5,350,2);

В този пример ключът(id) е 6, името е 'Chicken Soup', съдържанието е 'chicken, peppers, potatoes, carrots, onion, parsley' ,ключът(id) на категорията е 9 т.е Soup, цената е 4,5 лева, а количеството е 350 милилитра, тъй като ключът(id) на типа на типа на количеството, в което е продукта е 2 т.е “milliliters”.

След като се добави всеки продукт се добавя и какви съставки съдържа той, като това се направи в таблицата ProductsIngredients и на първо място се добавя ключът(id) на продукта, на второ място ключът(id) на съставката, която съдържа, на трето ключът(id) на типът на количеството, в което се съдържа и на четвърто колко от това количество се съдържа. Например:

INSERT INTO ProductsIngredients

Values(6,12,1,50);

Като в този пример се казва, че в продукта с ключ(id) 6, а именно 'Chicken Soup', се съдържа съставката с ключ(id) 12, а именно “chicken”, ключът(id) на типа на количеството, в което се измерва тази съставка е 1 т.е ‘grams’ и продукта съдържа 50 грама от съставката.

След това се добавиха различните менюта в таблицата Menus, като на всяко меню се добави ключ(id), име и дали е активно, ако е активно третата стойност е 1, ако не е активно третата стойност е 0. Например:

INSERT INTO Menus

Values(1,'Drinks',1);

В този пример се добави меню с ключ(id) 1, чието име е ‘Drinks’ и е активно към момента.

След като се добави меню се добавят и продукти, които са в това меню, като това се случва в таблицата MenusProducts, където първата стойност е ключа(id) на менюто, а втората ключа(id) на продукта. Например:

INSERT INTO MenusProducts

Values(1,14);

В този пример в менюто с ключ(id) 1, а именно Drinks се добавя продукта с ключ(id) 14, а именно Coca-Cola. Тук е мястото да отбележим, че за всеки продукт, който се предлага в бутилки, в таблицата Ingredients той е добавен като продукт и типът му na количеството е units, тъй като Ingredients един вид е таблица за склада на ресторанта.

След това се добавиха поръчките в таблицата Orders, като в тази таблица се добавя само ключът(id) на поръчката и дата, на която е взета. Например:

INSERT INTO Orders

Values(12,'2020-4-28');

В този пример поръчката е с ключ(id) 12, а датата, на която е взета е 28 април 2020 година.

След добавянето на една поръчка в таблицата OrdersProducts се добавиха продукти в съответната поръчка, като всеки кортеж съдържа ключ(id) на поръчката, ключ(id) на продукта и колко бройки от този продукт е поръчано. Например:

INSERT INTO OrdersProducts

Values(15,60,2);

В този пример се добави в поръчката с ключ 15, продукта с ключ 60, а именно ‘water’ и колко бройки от този продукт е поръчано, а именно - 2.

Така процесът по добавяне на данни към таблиците приключи.

**Group by заявки**

*/\* 1. Заявка, която извежда името на продукт и броят поръчвания в поръчки. Принтирайте ги от най-срещаните продукти към най-малко срещаните.\*/*

SELECT pr.Name AS Product, COUNT(op.orderId) AS AmountOfOrders

FROM OrdersProducts op

JOIN Products pr ON op.ProductId = pr.Id

GROUP BY pr.Name

ORDER BY AmountOfOrders DESC;

*/\* 2. Заявка, която извежда номер на поръчка, датата,*

*както и цялата сума, която трябва да се заплати за поръчки, които са поръчали поне 7 продукта. Нека са подредени*

*датата на издаване. \*/*

SELECT o.Id as OrderId,o.Date,

SUM(op.ProductQuantity\*pr.Price) AS TotalSum

FROM OrdersProducts op

JOIN Orders o ON op.OrderId = o.Id

JOIN Products pr ON pr.Id = op.ProductId

GROUP BY o.Id, o.Date

HAVING COUNT(op.ProductId) > 7

ORDER BY o.Date ASC;

*/\* 3. Заявка, която извежда храни, тяхната производствена*

*цена (в зависимост от продуктите, които съдържа) и цената, на която се продава в ресторанта. Подредени от най-висока към най-ниска цена на производство.\*/*

SELECT pr.Name AS Product, SUM((i.Price/1000)\*pin.IngredientQuantity) AS PriceProduction, pr.Price

FROM Ingredients i

JOIN ProductsIngredients pin ON i.Id = pin.IngredientId

JOIN Products pr ON pin.ProductId = pr.Id

JOIN Categories c ON c.id = pr.CategoryId

WHERE c.Name NOT LIKE 'Drink'

GROUP BY pr.Name, pr.Price

ORDER BY PriceProduction DESC;

*/\* 4. Заявка, която извежда име на меню, продуктът от менюто с най-ниска цена и продуктът с най-висока. \*/*

SELECT m.Type AS Menu, MIN(pr.Price) AS MinPriceProduct, MAX(pr.Price) AS MaxPriceProduct

FROM Products pr

JOIN MenusProducts mp ON pr.Id = mp.ProductId

JOIN Menus m ON mp.MenuId = m.Id

GROUP BY m.Type

*/\* 5. Заявка, която извежда името на съставката и броят ястия, в които се съдържа. Изведете ги откъм най-използваните ястия.\*/*

SELECT i.Name AS Ingredient, COUNT(pr.Name) AS CountProducts

FROM Products pr

JOIN ProductsIngredients pin ON pin.ProductId = pr.Id

JOIN Ingredients i ON i.Id = pin.IngredientId

GROUP BY i.Name

ORDER BY countProducts DESC;

*/\* 6. Заявка, която извежда име на съставка, количество и тип*

*количество, в каквато се използва във всички продукти. Изведете ги от към най-използваните съставки.\*/*

SELECT i.Name AS Ingredient, SUM (pin.IngredientQuantity) AS Quantity, ta.Name AS QuantityUnit

FROM Products pr

JOIN ProductsIngredients pin ON pin.ProductId = pr.Id

JOIN Ingredients i ON i.Id = pin.IngredientId

JOIN TypesOfAmount ta ON ta.Id = i.AmountTypeId

GROUP BY i.Name, ta.Name

ORDER BY ta.Name, Quantity DESC;

*/\* 7. Заявка, която извежда името на продуктите, които са съставени от поне 6 съставки. Извеждаме също кои са съставките, както и категория на продукта. Сортирани са според категорията на продукта.\*/*

SELECT pr.Name AS Product, pr.Description, c.Name AS Category

FROM Products pr

JOIN Categories c ON c.Id = pr.CategoryId

JOIN ProductsIngredients pin ON pin.ProductId = pr.Id

GROUP BY pr.Name, c.Name, pr.Description

HAVING COUNT(pin.IngredientId) > 5

ORDER BY c.Name ASC;

*/\* 8. Заявката извежда име на съставка, количество и количествена единица за всички продукти, които се намират в неактивното меню ( с цел подготвка за пускане). \*/*

SELECT i.Name AS Ingredient, SUM (pin.IngredientQuantity) AS Quantity, ta.Name AS QuantityUnit

FROM Products pr

JOIN ProductsIngredients pin ON pin.ProductId = pr.Id

JOIN Ingredients i ON i.Id = pin.IngredientId

JOIN TypesOfAmount ta ON ta.Id = i.AmountTypeId

JOIN MenusProducts mp ON pr.Id = mp.ProductId

JOIN Menus m ON mp.MenuId = m.Id

WHERE m.IsActive = 0

GROUP BY i.Name, ta.Name

ORDER BY ta.Name, Quantity DESC;

*/\* 9. Заявка, коятo извежда номер на поръчката, дата и обща сума, за тези поръчки, които са поръчвали поне 10 различни продукта \*/*

SELECT op.OrderId, o.Date, SUM(pr.Price) AS TotalSum

FROM Products pr

JOIN OrdersProducts op ON op.ProductId = pr.Id

JOIN Orders o ON o.Id = op.OrderId

GROUP BY op.OrderId, o.Date

HAVING COUNT(DISTINCT op.ProductId) > 9

*/\* 10. Заявка, която извежда активните менюта и колко*

*на брой продукта от него са били поръчвани. \*/*

SELECT m.Type AS Menu, SUM(op.ProductId\*op.ProductQuantity) AS QuantityProducts

FROM Products pr

JOIN MenusProducts mp ON pr.Id = mp.ProductId

JOIN Menus m ON mp.MenuId = m.Id

JOIN OrdersProducts op ON op.ProductId = pr.Id

JOIN Orders o ON o.Id = op.OrderId

WHERE m.IsActive = 1

GROUP BY m.Type;

**Подзаявки**

/\* 1. *Заявка, която извежда имената на артикулите, чиято цена е по-висока от най-високата цена на продукт, поръчан в поръчка преди 2020-04-18. \*/*

SELECT Name

FROM Products

WHERE Price > (

SELECT MAX(Price)

FROM Orders o JOIN OrdersProducts op ON o.Id=op.OrderId JOIN Products p ON p.Id = op.ProductId

WHERE Date < '2020-04-18'

);

*/\* 2. Заявка, която извежда броя на поръчките, в които присъства "Homemade" артикул (напр. Hommemade Chips) \*/*

SELECT COUNT(DISTINCT o.Id)

FROM Orders o

JOIN OrdersProducts op ON o.Id=op.OrderId

WHERE op.ProductId IN (

SELECT Id FROM Products WHERE Name LIKE 'Homemade%'

)

*/\* 3. Заявка, която извежда датата на най-скорошната поръчка, в която присъства салата от гръцкото меню. \*/*

SELECT MAX(o.Date)

FROM Orders o JOIN OrdersProducts op ON op.OrderId = o.Id

WHERE op.ProductId IN (

SELECT mp.ProductId

FROM Menus m JOIN MenusProducts mp ON mp.MenuId=m.Id JOIN Categories c ON c.Id = mp.ProductId

WHERE c.Name = 'Salad' AND m.Type = 'Greek'

)

*/\* 4. Името на продукта, съдържащ най-скъпата съставка. \*/*

SELECT p.Name

FROM Products p Join ProductsIngredients pi ON p.id=pi.ProductId

WHERE pi.IngredientId = (

SELECT Id FROM Ingredients

WHERE Price = (

SELECT MAX(Price) FROM Ingredients

)

)

*/\* 5. Заявка, която извежда съставките на продуктите, които са "Fried" от гръцкото меню.\*/*

SELECT i.Name

FROM ProductsIngredients pi JOIN Ingredients i ON pi.IngredientId=i.Id

WHERE pi.ProductId IN (

SELECT mp.ProductId FROM Menus m JOIN MenusProducts mp ON mp.MenuId=m.Id

WHERE mp.ProductId IN (

SELECT Id FROM Products WHERE Name LIKE 'Fried%'

) AND m.Type = 'Greek'

)

**Прости заявки и заявки върху две и повече релации**

*/\* 1. Заявка, която връща имената на продуктите ,които не съдържат дадена съставка (домат). \*/*

SELECT NAME

FROM Products

WHERE Description

NOT LIKE '%tomatoes%' *,където tomatoes е името на съставката ,която проверяване дали не се съдържа.*

*/\*2. Зявка, която връща Id и име на съставките ,на които количеството им е над 2000 грама ,подредени по азбучен ред и по грамаж в нарастващ ред.\*/*

SELECT Id,Name

FROM Ingredients

WHERE Amount >2000 AND AmountTypeId = 1

ORDER BY Name,Amount ASC

*/\*3.Заявка ,която връща имената и грамажа на порцията на всички десерти с цена до 5 лева ,подредени по цена във възходящ ред.\*/*

SELECT NAME,PortionAmount

FROM Products

WHERE CategoryId = 7 AND Price < 5

ORDER BY Price ASC

*/\*4.Заявка ,която връща имената на всички активни менюта\*/*

SELECT Type

FROM Menus

WHERE IsActive = 1

*/\*5.Заявка ,която връща ID и дата на поръчките ,направени през месец Април 2020г.\*/*

SELECT \*

FROM Orders

WHERE Date

LIKE '%2020-04%'

*/\*1.Заявка, която връща име, описание, цена и размер на порцията на ястията ,които са в категория Пица ,подредени по азбучен ред.\*/*

SELECT Products.Name,[Description],Price,PortionAmount

FROM Products,Categories

WHERE Categories.Name='Pizza' and CategoryId=5

ORDER By Products.Name

*/\*2.Заявка ,която връща общата сума на цената на продуктите ,които са под ID 14 (Coca Cola)*

*в поръчка с ID номер 1.\*/*

SELECT SUM(Price)

FROM OrdersProducts,Orders,Products

WHERE [Date]='2020-04-10'

AND OrderId = 1 AND ProductId = 14 AND Products.Id = 14

*/\*3.Заявка ,която връща датата ,ID и количество на продуктите ,на поръчка с ID = 1\*/*

SELECT Date,ProductId,ProductQuantity

FROM Orders,OrdersProducts

WHERE Id = 1 AND OrderId = 1

*/\*4.Заявка ,която връща име ,описание ,цена ,количество и мерната единица за количество на всички продукти ,които се измерват в грамове.\*/*

SELECT Products.Name,[Description],Price,PortionAmount,TypesOfAmount.Name

FROM Products,TypesOfAmount

WHERE AmountTypeId = 1 AND TypesOfAmount.Id= 1

*/\*5.Заявка, която извежда име ,описание ,цена ,количество и мерната единица за количество на всички продукти ,които се измерват в милилитри и са ястия.\*/*

SELECT Products.Name,[Description],Price,PortionAmount,TypesOfAmount.Name

FROM Products,TypesOfAmount

WHERE AmountTypeId = 2 AND TypesOfAmount.Id= 2

AND [Description] LIKE '% %'

**Заявки със съединения**

*/\* 1. Заявка, която да извежда кои продукти се съдържат в повече от едно меню и борят на менютата, в които се съдържа. \*/*

SELECT pr.Name, COUNT(pr.Name) as NumberOfMenus

FROM Products pr

JOIN MenusProducts mp ON pr.id=mp.ProductId

JOIN Menus m ON mp.MenuId=m.Id

GROUP BY pr.Name

HAVING COUNT(pr.Name)>1;

*/\* 2. Седмица на Италия! Заявка, извеждаща всички продукти, техния тип, описание на съставките и цената в Италианското меню. \*/*

SELECT pr.Name, c.Name as 'Type of product', pr.Description, pr.Price, m.Type as 'Type of menu'

FROM Products pr

JOIN MenusProducts mp ON pr.id=mp.ProductId

JOIN Menus m ON mp.MenuId=m.Id

JOIN Categories c ON pr.CategoryId=c.Id

WHERE m.Type='Italian';

*/\* 3. Заявка, която извежда датите, на които са поръчани продукти от тип Салата с цени по-ниски от 10 лева. \*/*

SELECT o.Date, pr.Name, c.Name as 'Type of product' , pr.Price, m.Type as 'Type of menu'

FROM Orders o

JOIN OrdersProducts op ON o.id=op.OrderId

JOIN Products pr ON op.ProductId=pr.Id

JOIN MenusProducts mp ON pr.id=mp.ProductId

JOIN Menus m ON mp.MenuId=m.Id

JOIN Categories c ON pr.CategoryId=c.Id

WHERE c.Name='Salad' AND pr.Price<10;

*/\* 4. Заявка, която извежда продуктите, за направата на които са използвани повече от 100 грама захар \*/*

SELECT pr.Name, c.Name as 'Type of product', i.Name as 'Ingredient', ta.Name as 'Type of ingredient', pi.IngredientQuantity

FROM Products pr

JOIN ProductsIngredients pi ON pr.Id=pi.ProductId

JOIN TypesOfAmount ta ON pi.IngredientAmountTypeId=ta.Id

JOIN Ingredients i ON ta.Id=i.AmountTypeId

JOIN Categories c ON pr.CategoryId=c.Id

WHERE pi.IngredientQuantity<50 AND ta.Name='grams' AND i.Name='sugar';

*/\* 5. Заявка, която извежда всички наливни напитки. \*/*

SELECT pr.Name, c.Name as 'Category', ta.Name as 'Type of Amount', pr.PortionAmount

FROM Products pr

JOIN Categories c ON pr.CategoryId=c.Id

JOIN TypesOfAmount ta ON pr.AmountTypeId=ta.Id

WHERE c.Name='Drink' AND ta.Name='milliliters';

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Тригери:**

*/\**

*Trigger for soft delete Menus*

*Тригер ,който прави soft delete ,тоест когато изтриваме дадено меню ,то просто става неактивно ,не се изтрива ,а стойността на isActive става 0 ,тъй като изтриването на данни от базата*

*\*/*

GO

CREATE OR ALTER TRIGGER tr\_soft\_delete

ON dbo.Menus

INSTEAD OF DELETE

AS

UPDATE dbo.Menus

SET IsActive = 0

WHERE Id IN (SELECT Id FROM deleted)

PRINT 'SUCCESSFULLY DISABLED';

*/\**

*Trigger tracking events around Products*

*Тригер ,показващ дейностите ,които се извършват по таблицата Products ,като insert,delete,update ,записвайки ги в ProductsAuditLog \*/*

GO

CREATE OR ALTER TRIGGER tr\_productEvent

ON dbo.Products

FOR INSERT,UPDATE,DELETE

AS

declare @pId int;

declare @pName varchar(50);

declare @pDescription varchar(150);

declare @pCategoryId int;

declare @pEventTime DATE;

declare @pAction varchar(50);

select @pId = i.Id from inserted

select @pName = i.Name from inserted i;

select @pDescription = i.DESCRIPTION from inserted i;

select @pCategoryId = i.CategoryId from inserted i;

if exists(SELECT \* FROM inserted) and exists (select \* from deleted)

begin

SET @pAction = 'Updated'

insert into ProductsAuditLog

(Id,Name,[Description],CategoryId,EventTime,[Action])

values(@pId,@pName,@pDescription,@pCategoryId,GETDATE(),@pAction);

end

if exists(SELECT \* FROM deleted) and not exists (select \* from inserted)

begin

SET @pAction = 'Deleted'

insert into ProductsAuditLog (Id,Name,[Description],CategoryId,EventTime,[Action])

values(@pId,@pName,@pDescription,@pCategoryId,GETDATE(),@pAction);

end

if exists(SELECT \* FROM inserted) and not exists (select \* from deleted)

begin

SET @pAction = 'Inserted'

insert into ProductsAuditLog (Id,Name,[Description],CategoryId,EventTime,[Action])

values(@pId,@pName,@pDescription,@pCategoryId,GETDATE(),@pAction);

end

PRINT 'TRIGGER fired';

**Индекси:**

**/\*** Колони от таблици, които често се използват в заявки \*/

CREATE iNDEX idx\_Products

ON Products(Name);

CREATE INDEX idx\_Ingredients

ON Ingredients(Name);

CREATE INDEX idx\_Orders

ON Orders(Date);

**Изгледи:**

1)

/\* View за активните Менюта. \*/

CREATE VIEW v\_ActiveMenus

AS

SELECT \*

FROM Menus

WHERE IsActive=1;

Select Type

from v\_ActiveMenus;

2)

/\* View за всички продукти, които могат да бъдат поръчани в момента. \*/

CREATE VIEW v\_ActiveProducts

AS

SELECT pr.Name, c.Name as TypeOfProduct, pr.Description, pr.Price, m.Type as TypeOfMenu

FROM Products pr

JOIN MenusProducts mp ON pr.id=mp.ProductId

JOIN Menus m ON mp.MenuId=m.Id

JOIN Categories c ON pr.CategoryId=c.Id

WHERE m.IsActive=1;

SELECT \*

FROM v\_ActiveProducts;

3)

/\* View за наличните съставки в склада и тяхното количество. \*/

CREATE VIEW v\_Ingredients

AS

SELECT i.Name, i.Amount, ta.Name as TypeOfAmount

FROM Ingredients i JOIN TypesOfAmount ta ON i.AmountTypeId=ta.Id;

SELECT \*

FROM v\_Ingredients

ORDER BY Name;

4)

/\* View за активността на поръчките и натрупаните сметки по дати.\*/

CREATE VIEW v\_Number\_Of\_OrderProducts\_per\_Day

AS

SELECT o.Date, COUNT(Distinct o.Id) AS Orders,

SUM(op.ProductQuantity) AS NumberOrderProducts,

SUM(op.ProductQuantity\*pr.Price) AS Bill

FROM Orders o

JOIN OrdersProducts op ON o.Id=op.OrderId

JOIN Products pr ON op.ProductId=pr.Id

GROUP BY o.Date;

SELECT \*

FROM v\_Number\_Of\_OrderProducts\_per\_Day;