

Vizsgaremek Adatbázis Dokumentáció

ElectricGuitarShop



Hallgató:

A hallgató e-mail címe:

Dátum:

Horváth Tamás

thomas.horvath0416@gmail.com

2023. 05. 06.



Tartalomjegyzék

Tar	talomjegyzék	2
	A téma rövid leírása	4
	Adatbázis Diagramm	6
	ElectricGuitarShop – Adatbázis tulajdonságok	7
≣≣	Táblák	8
	dbo.Category	9
	dbo.DatabaseLog	11
	dbo.DictBrand	13
	dbo.DictDeliveryType	15
	dbo.DictPaymentType	16
	dbo.DictTax	18
	dbo.Global	20
	dbo.OrderHeader	22
	dbo.OrderItem	26
	dbo.Partner	29
	dbo.Product	32
	dbo.ProductProperty	35
	dbo.PropertyType	38
	dbo.PropertyValue	40
	dbo.SubCategory	42
	dbo.User	
Ħ	Nézetek	47
	dbo.vCountOfProduct	
	dbo.vPreviousMonthAmount	50
	dbo.vProductInformation	52
	dbo.vQuarterlyStatement	53
	母 dbo.vWhoisAdmin	55
	Tárolt eljárások	56
	dbo.AddOrder	
	dbo.AddPartner	59
	dbo.AddProduct	61
	dbo.MonthlyStatement	63
	dbo.TestDatasImport	65
	dbo.UpdatePartner	67
	dbo.UpdateProduct	69



E_{fx}^{-1}	Table-valued Függvények	71
	dbo.fRepeatCustomer	
101 01 fx	Skalár függvények	73
	dbo.Discount	74
	dbo.EmailAddressCheck	
	dbo.Forint	76
	dbo.ShippingFee	
4	Adatbázis Triggerek	
	* tDLL_LOG	79
101 010	Típus táblák	80
	dbo.ProductIdAndQuantity	81
	dbo.Property	82
1	Felhasználók	83
	⊈ EGS_Admin	84
	₹ EGS_DataAnalyst	85
	1 EGS_EMP	86
	₹ EGS_User	87
Ų.	Adatbázis szerepkörök	88
	DataAnalystRole	89
	db_owner	90
	Emp_Role	91
	Lser_Role	92
<u>.</u>	Alkalmazás szerepkörök	93
	№ WebApp	94
	Mentési stratégia	95



A téma rövid leírása

Az ElectricGuitarShop egy fiktív, online hangszerbolt, amely elektromos gitárok és kiegészítőik forgalmazására specializálódott. Az adatbázis jelenleg csak a teljes rendszer egy részét tartalmazza, néhány funkció, mint például a raktárkezelés és számlázás csak később kerül kiépítésre.

Az adatbázis főbb funkciói a következők:

1. Termékek kezelése:

Az adatbázisban nyilvántartjuk a webáruházban elérhető termékeket, amelyek négy kategóriába és tizennégy alkategóriába sorolhatóak. Az adatok hatékony tárolása és keresése érdekében a termékek különböző tulajdonságai külön táblákban lettek tárolva.

2. Rendelések kezelése:

Az adatbázisban tároljuk a webáruházban leadott rendelések és a hozzájuk tartozó szállítási és fizetési módok adatait. Az adatbázis tartalmazza a megrendelők által megállapított kedvezmények kezelését és a szállítási díjak kalkulációját is. Az adatokra épülő üzleti elemzéshez szükséges lekérdezések tárolt eljárásokon és nézeteken keresztül érhetők el.



3. Partnerek és felhasználók kezelése:

Az adatbázisban rögzítjük a webáruház partnereinek adatait, beleértve az adminisztrátorokat és a vásárlókat. Az aktuális verzióban minden partnernek egy felhasználói fiókja van a user táblában, ahol a felhasználónevek és a titkosított jelszavak is tárolódnak. Az adatbázisban nyilvántartjuk, hogy ki az adminisztrátor és ki a vásárló. A következő fejlesztési szakaszban más típusú partnerek is megjelenhetnek, például beszállítók vagy gyártók, akiknek azonban nem kell hozzáférést kapniuk a webalkalmazáshoz, így nem szükséges számukra az user táblában szerepelni.

4. Biztonság:

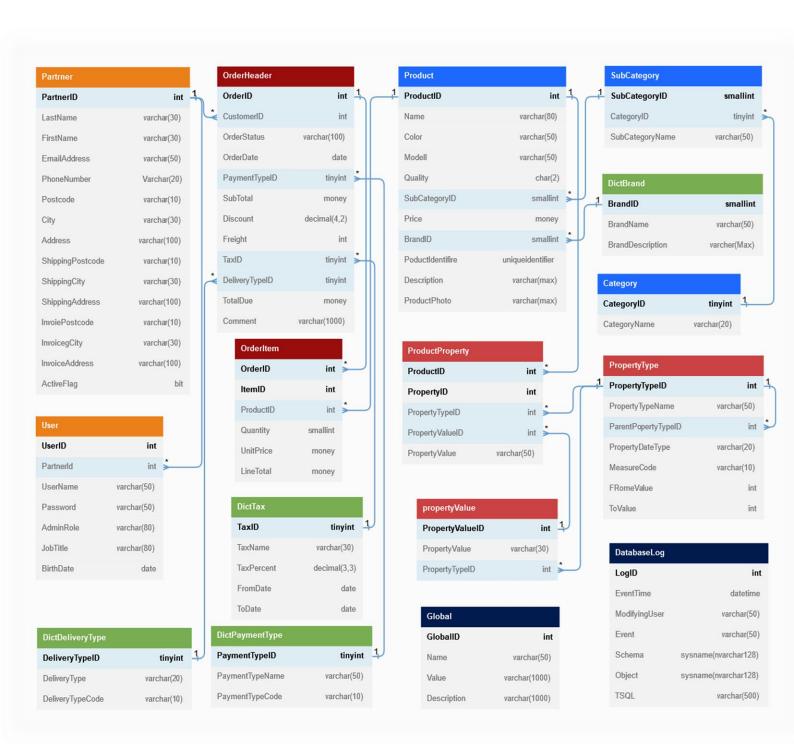
Az adatbázisban biztosítjuk a jogosultságkezelést, hogy a különböző felhasználók csak a számukra szükséges adatokhoz férjenek hozzá. A jogosultságokat külön részre bontva, a webshop alkalmazottainak, a vásárlók számára elérhető felületeknek, az adatelemzőknek és a fejlesztés alatt álló applikációknak külön-külön meghatározzuk.

Az adatbázisban tárolt adatok:

- Termékek és a hozzájuk tartozó tulajdonságok
- A rendelések fej és tétel adatai
- Partnerek és userek adatai
- A forgalmazott márkák neve és leírása
- A szállítás és fizetési módok fajtái
- A különböző adónemek
- Az adatbázisban történő események naplózási adatai
- A változó adatok, valamint elérési utak (Global tábla)



Adatbázis Diagramm





■ ElectricGuitarShop – Adatbázis tulajdonságok

Leírás

Elektromos gitárokat és kiegészítőit áruló webshop adatait tároló adatbázis.

Adatbázis tulajdonságok

Property	Value
Compatibility Level	Max
Creation date	Mar 31 2023
Database size	216.00 MB
Unallocated space	63.05 MB

Adatbázis beállítások

Property	Value
Compatibility Level	160
Database collation	Hungarian_CI_AS
Database status	ONLINE
Recovery model	FULL
Database owner	sa

Files

Name	Туре	File Group	Size	MaxSize	AutoGrowth	File Name
Electric- Guitar- Shop	Data		72,00 MB	unlimited	64,00 MB	C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\DATA\Electric- GuitarShop.mdf
Electric- Guitar- Shop_log	Log		136,00 MB	2048,00 GB	64,00 MB	C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\DATA\Electric- GuitarShop_log.ldf
Large- Text	Data	Large- Text/Img	8,00 MB	unlimited	64,00 MB	C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\DATA\Large- Text.ndf



III Táblák

Objektumok

Name

dbo.Category

A termék kategóriákat tartalmazó tábla.

dbo.DatabaseLog

Az adatbázisban történt DML és DDL utasításokat naplózza.

dbo.DictBrand

A márkákat megjelenítő szótár tábla.

dbo.DictDeliveryType

A kiszállítás fajtáit megjelenítő szótár tábla.

dbo.DictPaymentType

A fizetési módokat megjelenítő szótár tábla.

dbo.DictTax

Az adónemeket tartalmazó szótár tábla.

dbo.Global

A változó adatok tárolására szolgál. Például elérési utak vagy fizetési limitek.

dbo.OrderHeader

A rendelés fej adatait tartalmazó tábla.

dbo.OrderItem

A rendelés tételeit tartalmazó tábla.

dbo.Partner

A partnerek adatait tartalmazó tábla.

dbo.Product

A webshopban forgalmazott termékek adatait tartalmazó tábla.

dbo.ProductProperty

A termékek tulajdonságait tartalmazó tábla.

dbo.PropertyType

A termékek tulajdonságainak a fajtáit tartalmazó tábla.

dbo.PropertyValue

A tulajdonságok értékeit tartalmazó tábla.

dbo.SubCategory

A termékek alkategóráit tartalmazó tábla.

dbo.User

A webshop alkalmazást használó felhasználók adatai. Mivel az adatbázisban csak adminok és customerek szerepelnek ezért a partner tábla minden egyedéhez tartozik egy user. Az adatbázis bővülésével egyéb partner típusok is bekerülhetnek majd, akikhez nem kell felhasználói regisztráció. Például beszállítók.



■ dbo.Category

Leírás

A termék kategóriákat tartalmazó tábla.

Tulajdonságok

Property	Value
Collation	Hungarian_CI_AS
Row Count (~)	4
Created	12:42:34 2023. április 29., szombat
Last Modified	12:45:11 2023. április 29., szombat

Oszlopok

ı	Key	Name	Adattípus	Méret (Bytes)	Nullability
F	°/c	CategoryID A kategória egyedi azonosítója.	tinyint	1	NOT NULL
	ıi.	CategoryName A kategória neve.	varchar(20)	20	NOT NULL

Indexek

Key	Name	Key Columns	Unique
PK <mark>P</mark>	PK_Category_CategoryID Elsődleges kulcs.	CategoryID	True
	UC_Category_CategoryName A névnek egyedinek kell lennie.	CategoryName	True

Туре	Action	Owning Principal
Grant	INSERT	Emp_Role
Grant	SELECT	Emp_Role



```
CREATE TABLE dbo.Category(
CategoryID tinyint NOT NULL,
CategoryName varchar (20) COLLATE Hungarian_CI_AS NOT NULL
) ON PRIMARY

ALTER TABLE dbo.Category ADD CONSTRAINT PK_Category_CategoryID PRIMARY KEY CLUSTERED (CategoryID) ON PRIMARY

ALTER TABLE dbo.Category ADD CONSTRAINT UC_Category_CategoryName UNIQUE NONCLUSTERED (CategoryName) ON PRIMARY

GRANT INSERT ON dbo.Category TO Emp_Role
GRANT SELECT ON dbo.Category TO Emp_Role
```



■ dbo.DatabaseLog

Leírás

Az adatbázisban történt DML és DDL utasításokat naplózza.

Tulajdonságok

Property	Value
Collation	Hungarian_CI_AS
Row Count (~)	499
Created	12:42:04 2023. április 29., szombat
Last Modified	13:11:42 2023. május 6., szombat

Oszlopok

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PKP C	LogID Az esemény azonosítója.	int	4	NOT NULL	1 - 1
	EventTime Az esemény ideje.	datetime	8	NOT NULL	
	ModifyingUser Az eseményt véghezvivő felhasználó.	varchar(50)	50	NOT NULL	
	Event Az esemény elnevezése.	varchar(50)	50	NOT NULL	
	Schema Az esemény melyik sémában történt.	sys.sysname	256	NULL allowed	
	Object Az esemény melyik objektumon történt.	sys.sysname	256	NULL allowed	
	TSQL A DDL események SQL lerása.	varchar(500)	500	NULL allowed	

Indexek

Key	Name	Key Columns	Unique
PK <mark>P</mark> C	PK_DatabaseLog_LogID Elsődleges kulcs.	LogID	True

Triggerek

Name	ANSI Nulls On	Quoted Identifier On	On
tLogSafe Tilltja a táblába való közvetlen írást, módosítást, valamint törlést.	True	True	After Delete Insert Update



```
CREATE TABLE dbo.DatabaseLog(
LogID int NOT NULL IDENTITY(1, 1),
EventTime datetime NOT NULL,
ModifyingUser varchar (50) COLLATE Hungarian_CI_AS NOT NULL,
Event varchar (50) COLLATE Hungarian_CI_AS NOT NULL,
Schema sys.sysname NULL,
Object sys.sysname NULL,
TSQL varchar (500) COLLATE Hungarian_CI_AS NULL
) ON PRIMARY

CREATE TRIGGER dbo.tLogSafe ON dbo.DatabaseLog FOR INSERT, UPDATE, DELETE
AS

IF @@NESTLEVEL = 1 ROLLBACK TRAN

ALTER TABLE dbo.DatabaseLog ADD CONSTRAINT PK_DatabaseLog_LogID PRIMARY KEY CLUSTERED (LogID)
ON PRIMARY
```



■ dbo.DictBrand

Leírás

A márkákat megjelenítő szótár tábla.

Tulajdoságok

Property	Value	
Collation	Hungarian_CI_AS	
Row Count (~)	18	
Created	12:42:34 2023. április 29., szombat	
Last Modified	12:45:11 2023. április 29., szombat	

Oszlopok

Key	Name	Data Type	Max Length (Bytes)	Nullability
PKP C	BrandID A brand egyedi azonosítója.	smallint	2	NOT NULL
ı.	BrandName A brand neve.	varchar(50)	50	NOT NULL
	BrandDescription A brand leírása.	varchar(max)	max	NULL allowed

Indexek

Key	Name	Key Columns	Unique
₽ <mark>%</mark> C	PK_DictBrand_BrandID Elsődleges kulcs.	BrandID	True
	UC_DictBrand_BrandName A brand nevének egyedinek kell lennie.	BrandName	True

Туре	Action	Owning Principal
Grant	INSERT	Emp_Role
Grant	SELECT	Emp_Role
Grant	UPDATE	Emp_Role



```
CREATE TABLE dbo.DictBrand(
BrandID smallint NOT NULL,
BrandName varchar (50) COLLATE Hungarian_CI_AS NOT NULL,
BrandDescription varchar (max) COLLATE Hungarian_CI_AS NULL
) ON PRIMARY TEXTIMAGE_ON LargeText/Img

ALTER TABLE dbo.DictBrand ADD CONSTRAINT PK_DictBrand_BrandID PRIMARY KEY CLUSTERED (BrandID)
ON PRIMARY

ALTER TABLE dbo.DictBrand ADD CONSTRAINT UC_DictBrand_BrandName UNIQUE NONCLUSTERED (BrandName) ON PRIMARY

GRANT INSERT ON dbo.DictBrand TO Emp_Role
GRANT SELECT ON dbo.DictBrand TO Emp_Role
GRANT UPDATE ON dbo.DictBrand TO Emp_Role
GRANT UPDATE ON dbo.DictBrand TO Emp_Role
```



dbo.DictDeliveryType

Leírás

A kiszállítás fajtáit megjelenítő szótár tábla.

Tulajdonságok

Property	Value	
Collation	Hungarian_CI_AS	
Row Count (~)	2	
Created	12:42:34 2023. április 29., szombat	
Last Modified	12:45:11 2023. április 29., szombat	

Oszlopok

Key	Name	Data Type	Max Length (Bytes)	Nullability
PK <mark>P</mark> C	DeliveryTypeID Egyedi azonosító.	tinyint	1	NOT NULL
ı.	DeliveryType A kiszállítás fajtája.	varchar(20)	20	NOT NULL
	DeliveryTypeCode A kiszállítás rövid azonosítója.	varchar(10)	10	NULL allowed

Indexek

Key	Name	Key Columns	Unique
P <mark>/</mark> C	PK_DictDeliveryType_DeliveryTypeID Elsődleges kulcs.	DeliveryTypeID	True
	UC_DictDeliveryType_DeliveryType Nem lehet egy szállítót kétszer fölvenni.	DeliveryType	True

```
CREATE TABLE dbo.DictDeliveryType(
DeliveryTypeID tinyint NOT NULL,
DeliveryType varchar (20) COLLATE Hungarian_CI_AS NOT NULL,
DeliveryTypeCode varchar (10) COLLATE Hungarian_CI_AS NULL
) ON PRIMARY

ALTER TABLE dbo.DictDeliveryType ADD CONSTRAINT PK_DictDeliveryType_DeliveryTypeID PRIMARY KEY
CLUSTERED (DeliveryTypeID) ON PRIMARY

ALTER TABLE dbo.DictDeliveryType ADD CONSTRAINT UC_DictDeliveryType_DeliveryType UNIQUE
NONCLUSTERED (DeliveryType) ON PRIMARY
```



■ dbo.DictPaymentType

Leírás

A fizetési módokat megjelenítő szótár tábla.

Tulajdonságok

Property	Value	
Collation	Hungarian_CI_AS	
Row Count (~)	3	
Created	12:42:34 2023. április 29., szombat	
Last Modified	12:45:11 2023. április 29., szombat	

Oszlopok

Key	Name	Data Type	Max Length (Bytes)	Nullability
PKP C	PaymentTypeID Egyedi azonosító.	tinyint	1	NOT NULL
ı.t.	PaymentTypeName A fizetési típus neve.	varchar(50)	50	NOT NULL
	PaymentTypeCode A fizetési típus rövid azonosítója.	varchar(10)	10	NULL allowed

Indexek

Key	Name	Key Columns	Unique
PK <mark>P</mark> C	PK_DictPaymentType_PaymentTypeID Elsődleges kulcs.	PaymentTypeID	True
	UC_DictPaymentType_PaymentTypeName Egy fizetési mód csak egyszer szerepelhet.	PaymentTypeName	True

Туре	Action	Owning Principal
Grant	INSERT	Emp_Role
Grant	SELECT	Emp_Role
Grant	UPDATE	Emp_Role



```
CREATE TABLE dbo.DictPaymentType(
PaymentTypeID tinyint NOT NULL,
PaymentTypeName varchar (50) COLLATE Hungarian_CI_AS NOT NULL,
PaymentTypeCode varchar (10) COLLATE Hungarian_CI_AS NULL
) ON PRIMARY

ALTER TABLE dbo.DictPaymentType ADD CONSTRAINT PK_DictPaymentType_PaymentTypeID PRIMARY KEY
CLUSTERED (PaymentTypeID) ON PRIMARY

ALTER TABLE dbo.DictPaymentType ADD CONSTRAINT UC_DictPaymentType_PaymentTypeName UNIQUE
NONCLUSTERED (PaymentTypeName) ON PRIMARY

GRANT INSERT ON dbo.DictPaymentType TO Emp_Role
GRANT SELECT ON dbo.DictPaymentType TO Emp_Role
GRANT UPDATE ON dbo.DictPaymentType TO Emp_Role
GRANT UPDATE ON dbo.DictPaymentType TO Emp_Role
```



■ dbo.DictTax

Leírás

Az adónemeket tartalmazó szótár tábla.

Tulajdonságok

Property	Value	
Collation	Hungarian_CI_AS	
Row Count (~)	11	
Created	12:42:35 2023. április 29., szombat	
Last Modified	12:45:11 2023. április 29., szombat	

Oszlopok

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK <mark>P</mark> C	TaxID Egyedi azonosító.	tinyint	1	NOT NULL	1 - 1
	TaxName Az adó neve.	varchar(30)	30	NOT NULL	
	TaxPercent Az adó százalékos értéke.	decimal(3,3)	5	NOT NULL	
	FromDate Bevezetés dátuma.	date	3	NULL allowed	
	ToDate Viszzavonás dátuma.	date	3	NULL allowed	

Indexee

Key	Name	Key Columns	Unique
P <mark>/</mark> C	PK_DictTax_TaxID Elsődleges kulcs.	TaxID	True

Megszorítások

Name	Constraint
CK_DictTax_Date Visszavonás dátuma nem lehet a bevezetés előtti dátum.	(ToDate>FromDate)

Туре	Action	Owning Principal
Grant	INSERT	Emp_Role
Grant	SELECT	Emp_Role



```
CREATE TABLE dbo.DictTax(
TaxID tinyint NOT NULL IDENTITY(1, 1),
TaxName varchar (30) COLLATE Hungarian_CI_AS NOT NULL,
TaxPercent decimal (3, 3) NOT NULL,
FromDate date NULL,
ToDate date NULL
) ON PRIMARY

ALTER TABLE dbo.DictTax ADD CONSTRAINT CK_DictTax_Date CHECK ((ToDate>FromDate))

ALTER TABLE dbo.DictTax ADD CONSTRAINT PK_DictTax_TaxID PRIMARY KEY CLUSTERED (TaxID) ON PRIMARY

GRANT INSERT ON dbo.DictTax TO Emp_Role
GRANT SELECT ON dbo.DictTax TO Emp_Role
```



■ dbo.Global

Leírás

A változó adatok tárolására szolgál. Például elérési utak vagy fizetési limitek.

Tulajdonságok

Property	Value	
Collation	Hungarian_CI_AS	
Row Count (~)	10	
Created	12:42:34 2023. április 29., szombat	
Last Modified	12:42:34 2023. április 29., szombat	

Oszlopok

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PKP C	GlobalID Egyedi azonosító.	int	4	NOT NULL	1 - 1
	Name Elnevezés.	varchar(50)	50	NULL allowed	
	Value Érték.	varchar(1000)	1000	NULL allowed	
	Description Rövid leíás, hogy mire szolgál a tárolt elem.	varchar(1000)	1000	NULL allowed	

Indexek

Key	Name	Key Columns	Unique
Pk <mark>/</mark> C	PK_Global_GLobalID Elsődleges kulcs.	GlobalID	True

Туре	Action Owning Principal	
Grant	INSERT	Emp_Role
Grant	SELECT	Emp_Role



```
CREATE TABLE dbo.Global(
GlobalID int NOT NULL IDENTITY(1, 1),
Name varchar (50) COLLATE Hungarian_CI_AS NULL,
Value varchar (1000) COLLATE Hungarian_CI_AS NULL,
Description varchar (1000) COLLATE Hungarian_CI_AS NULL
) ON PRIMARY

ALTER TABLE dbo.Global ADD CONSTRAINT PK_Global_GLobalID PRIMARY KEY CLUSTERED (GlobalID) ON
PRIMARY

GRANT INSERT ON dbo.Global TO Emp_Role
GRANT SELECT ON dbo.Global TO Emp_Role
```



■ dbo.OrderHeader

Leírás

A rendelés fej adatait tartalmazó tábla.

Tulajdonságok

Property	Value
Collation	Hungarian_CI_AS
Row Count (~)	2492
Created	12:42:35 2023. április 29., szombat
Last Modified	12:45:11 2023. április 29., szombat

Oszlopok

Key	Name	Data Type	Computed	Max Length (Bytes)	Nullability	Identity	Default
PK <mark>P</mark> C	OrderID Rendelés azonosító.	int		4	NOT NULL	1 - 1	
FK	Customerid A vásárló azonosító.	int		4	NOT NULL		
	OrderStatus Rendelés állapota. Alapértelmezetten 'Elérhető' a státusz. Az adatbázis funkcióinak bővítésével ez a raktrékészlet állapotától fog függni.	varchar(100)		100	NULL allowed		
	OrderDate Rendelés dátuma.	date		3	NOT NULL		(sysdatetime())
FK	PaymentTypeID Fizetési mód azonosítója.	tinyint		1	NOT NULL		
	SubTotal Részösszeg.	money		8	NULL allowed		
	Discount Kedvezmény. Mértékét a discount függvény kalkulálja	decimal(4,2)		5	NULL allowed		
	Freight A szállítási díj. Az összegét a ShippingFee függvénnyel lehet előállítani.	int		4	NULL allowed		
F/P	TaxID Az adó azonosítója. Alapártelmezetten a 11-es id-hoz tartozó 27% ÁFA kulcs tartzik hozzá. Az adatbázisban feltüntetett árak már tartalmazzák az adót.	tinyint		1	NULL allowed		('11')
FK	DeliveryTypeID	tinyint		1	NOT		



A kiszállítás módját azonosító ld.				NULL	
TotalDue Az előző adotokból számított végösszeg.	money	True	8	NULL allowed	
Comment A megrendeléshez opcionálisan lehet megjegyzést fűzni.	varchar(1000)		1000	NULL allowed	

Számított mezők

Name	Column definition
TotalDue	(CONVERT(money,ceiling((SubTotal-(SubTotal*Discount))+Freight),(0)))

Indexek

Key	Name	Key Columns	Unique
P <mark>%</mark> C	PK_OrderHeader_OrderID Elsődleges kulcs.	OrderID	True

Triggerek

Name	ANSI Nulls On	Quoted Identifier On	On
tOrderHeaderLog A DML utasitásokat figyelő trigger mely az eseményt rögzíti a DatabaseLog táblában.	True	True	After Delete Insert Update

Megszorítások

Name	Checked On Column		Constraint
CK_OrderHeader_Discount A kedvezmény nem lehet negatív érték.	False	Discount	(Discount>=(0))
CK_OrderHeader_Freight A szállítási díjnak 0 és 10000 közé kell esnie.	False	Freight	(Freight>=(0) AND Freight<(10000))
CK_OrderHeader_SubTotal A részösszeg nem lehet negatív érték.	False	SubTotal	(SubTotal>=(0))

ldegen kulcsok

Name	Columns
FK_OrderHeader_DeliveryType_DeliveryTypeID Idegen kulcs a DictDeliveryType tábla felé.	DeliveryTypeID->dbo.DictDeliveryType.DeliveryTypeID
FK_OrderHeader_DictTax_TaxID Idegen kulcs a DictTax tábla felé.	TaxID->dbo.DictTax.TaxID
FK_OrderHeader_Partner_CustomerID Idegen kulcs a partner tábla felé.	Customerid->dbo.Partner.PartnerID
FK_OrderHeader_PaymentType_PaymentTypeID Idegen kulcs a DictPaymentType tábla felé.	PaymentTypeID->dbo.DictPaymentType.PaymentTypeID



Engedélyek

Туре	Action	Owning Principal
Grant	SELECT	Emp_Role
Grant	UPDATE	Emp_Role

```
CREATE TABLE dbo.OrderHeader(
OrderID int NOT NULL IDENTITY (1, 1),
Customerid int NOT NULL,
OrderStatus varchar (100) COLLATE Hungarian_CI_AS NULL,
OrderDate date NOT NULL CONSTRAINT DF OrderHeader OrderDate DEFAULT (sysdatetime()),
PaymentTypeID tinyint NOT NULL,
SubTotal money NULL,
Discount decimal (4, 2) NULL,
Freight int NULL,
TaxID tinyint NULL CONSTRAINT DF OrderHeader TaxID DEFAULT ('11'),
DeliveryTypeID tinyint NOT NULL,
TotalDue AS (CONVERT(money,ceiling((SubTotal-(SubTotal*Discount))+Freight),(0))),
Comment varchar (1000) COLLATE Hungarian_CI_AS NULL
) ON PRIMARY
CREATE TRIGGER dbo.tOrderHeaderLog
ON dbo.OrderHeader AFTER INSERT , UPDATE, DELETE
BEGIN
    DECLARE @schema sysname = (SELECT OBJECT SCHEMA NAME(object id('orderheader')))
    DECLARE @object sysname = (SELECT OBJECT NAME (parent id) AS TableName FROM sys.triggers
                               WHERE name = 'tOrderHeaderLog');
   DECLARE @operation_type VARCHAR(10);
   IF EXISTS (SELECT * FROM INSERTED)
   BEGIN
        IF EXISTS (SELECT * FROM DELETED)
       BEGIN
           SET @operation_type = 'UPDATE';
       END
        ELSE
        BEGIN
           SET @operation_type = 'INSERT';
    END
    ELSE
    BEGIN
       SET @operation_type = 'DELETE';
    INSERT INTO DatabaseLog (Event, ModifyingUser, EventTime, Schema, Object)
    VALUES (@operation type, USER NAME(), SYSDATETIME(), @schema, @object);
END:
```



```
ALTER TABLE dbo.OrderHeader WITH NOCHECK ADD CONSTRAINT CK OrderHeader Discount CHECK
((Discount>=(0)))
ALTER TABLE dbo.OrderHeader WITH NOCHECK ADD CONSTRAINT CK OrderHeader Freight CHECK
((Freight>=(0) AND Freight<(10000)))
ALTER TABLE dbo.OrderHeader WITH NOCHECK ADD CONSTRAINT CK OrderHeader SubTotal CHECK ((Sub-
Total>=(0)))
ALTER TABLE dbo.OrderHeader ADD CONSTRAINT PK OrderHeader OrderID PRIMARY KEY CLUSTERED
(OrderID) ON PRIMARY
ALTER TABLE dbo.OrderHeader ADD CONSTRAINT FK OrderHeader DeliveryType DeliveryTypeID FOREIGN
KEY (DeliveryTypeID) REFERENCES dbo.DictDeliveryType (DeliveryTypeID)
ALTER TABLE dbo.OrderHeader ADD CONSTRAINT FK_OrderHeader_DictTax_TaxID FOREIGN KEY (TaxID)
REFERENCES dbo.DictTax (TaxID)
GO
ALTER TABLE dbo.OrderHeader ADD CONSTRAINT FK_OrderHeader_Partner_CustomerID FOREIGN KEY
(Customerid) REFERENCES dbo.Partner (PartnerID)
ALTER TABLE dbo.OrderHeader ADD CONSTRAINT FK_OrderHeader_PaymentType_PaymentTypeID FOREIGN
KEY (PaymentTypeID) REFERENCES dbo.DictPaymentType (PaymentTypeID)
GRANT SELECT ON dbo.OrderHeader TO Emp Role
GRANT UPDATE ON dbo.OrderHeader TO Emp Role
```



dbo.OrderItem

Leírás

A rendelés tételeit tartalmazó tábla.

Tulajdonságok

Property	Value
Row Count (~)	7198
Created	12:42:35 2023. április 29., szombat
Last Modified	12:45:11 2023. április 29., szombat

Oszlopok

Key	Name	Data Type	Computed	Max Length (Bytes)	Nullability	Identity
PKPFKP C	OrderID A rendelés azonosító.	int		4	NOT NULL	
P <mark>/</mark> C	ItemID A rendelés tételének azonosítója.	int		4	NOT NULL	1 - 1
.i.F≯	ProductID A termék azonosítója.	int		4	NOT NULL	
⊞	Quantity A rendelt termék darabszáma.	smallint		2	NOT NULL	
	UnitPrice A rendelt termék egységára.	money		8	NOT NULL	
	LineTotal A darabszámból és egységárból kalkulált végösszeg.	money	True	8	NOT NULL	

Számított mezők

Name	Column definition
LineTotal	(isnull(Unitprice*Quantity,(0)))



Indexek

Key	Name	Key Columns	Unique
₽ <mark>%</mark> C	PK_OrderItem_ItemID_OrderID Elsődleges kulcs.(összetett kulcs)	ItemID, OrderID	True
	IX_OrderItem_ProductID Non-clusteres index a ProductId-n.	ProductID	

Triggerek

Name	ANSI Nulls On	Quoted Identifier On	On
tOrderItemLog A DML utasításokat figyelő trigger, ami az eseményeket rögzíti a DatabaseLog táblába.	True	True	After Delete Insert Update

Megszorítások

Name	Checked	On Column	Constraint
CK_OrderItem_Quantity A darabszám nem lehet negatív érték és nem lehet 100 fölötti érték.	False	Quantity	(Quantity>(0) AND Quantity<(100))
CK_OrderItem_UnitPrice Az egységár nem lehet negatív érték.	False	UnitPrice	(UnitPrice>=(0))

ldegen kulcsok

Name	Columns
FK_OrderItem_OrderHeader_OrderID Idegen kulcs az OrderHeader tábla felé.	OrderID->dbo.OrderHeader.OrderID
FK_OrderItem_Product_ProductID Idegen kulcs a Product tábla felé.	ProductID->dbo.Product.ProductID

Туре	Action	Owning Principal
Grant	SELECT	Emp_Role
Grant	UPDATE	Emp_Role



```
CREATE TABLE dbo.OrderItem(
OrderID int NOT NULL,
ItemID int NOT NULL IDENTITY(1, 1),
ProductID int NOT NULL,
Quantity smallint NOT NULL,
UnitPrice money NOT NULL,
LineTotal AS (isnull(Unitprice*Quantity, (0)))
) ON PRIMARY
CREATE TRIGGER dbo.tOrderItemLog
ON dbo.OrderItem AFTER INSERT , UPDATE, DELETE
AS
BEGIN
   DECLARE @schema sysname = (SELECT OBJECT SCHEMA NAME(object id('orderitem')))
   DECLARE @object sysname = (SELECT OBJECT_NAME(parent_id) AS TableName FROM sys.triggers
                              WHERE name = 'tOrderItemLog');
   DECLARE @operation_type VARCHAR(10);
   IF EXISTS (SELECT * FROM INSERTED)
       IF EXISTS (SELECT * FROM DELETED)
           SET @operation type = 'UPDATE';
       END
       ELSE
       BEGIN
         SET @operation type = 'INSERT';
       END
   END
   ELSE
   BEGIN
       SET @operation type = 'DELETE';
    INSERT INTO DatabaseLog (Event, ModifyingUser, EventTime,Schema,Object)
   VALUES (@operation type, USER NAME(), SYSDATETIME(), @schema, @object);
END;
ALTER TABLE dbo.OrderItem WITH NOCHECK ADD CONSTRAINT CK OrderItem Quantity CHECK
((Quantity>(0) AND Quantity<(100)))
ALTER TABLE dbo.OrderItem WITH NOCHECK ADD CONSTRAINT CK OrderItem UnitPrice CHECK ((Unit-
Price>=(0)))
ALTER TABLE dbo.OrderItem ADD CONSTRAINT PK OrderItem ItemID OrderID PRIMARY KEY CLUSTERED
(ItemID, OrderID) ON PRIMARY
CREATE NONCLUSTERED INDEX IX_OrderItem_ProductID ON dbo.OrderItem (ProductID) ON PRIMARY
ALTER TABLE dbo.OrderItem ADD CONSTRAINT FK OrderItem OrderHeader OrderID FOREIGN KEY (Order-
ID) REFERENCES dbo.OrderHeader (OrderID)
ALTER TABLE dbo.OrderItem ADD CONSTRAINT FK OrderItem Product ProductID FOREIGN KEY (Product-
ID) REFERENCES dbo.Product (ProductID)
GRANT SELECT ON dbo.OrderItem TO Emp Role
GRANT UPDATE ON dbo.OrderItem TO Emp Role
```



dbo.Partner

Leírás

A partnerek adatait tartalmazó tábla.

Tulajdonságok

Property	Value
Collation	Hungarian_CI_AS
Row Count (~)	433
Created	12:42:35 2023. április 29., szombat
Last Modified	12:45:11 2023. április 29., szombat

Oszlopok

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
P/ <mark>P</mark> C	PartnerID A partner egyedi azonosítója.	int	4	NOT NULL	1 - 1
	LastName Vezeték név.	varchar(30)	30	NOT NULL	
	FirstName Keresztnév.	varchar(30)	30	NOT NULL	
ф	EmailAddress Email cím.	varchar(50)	50	NULL allowed	
	PhoneNumber Telefonszám.	varchar(20)	20	NULL allowed	
	Postcode Irányító szám.	varchar(10)	10	NULL allowed	
	City Város	varchar(30)	30	NULL allowed	
	Address Cím.	varchar(100)	100	NULL allowed	
	ShippingPostcode Szállítási cím irányító száma, ha van.	varchar(10)	10	NULL allowed	
	ShippingCity Szállítási cím városa, ha van.	varchar(30)	30	NULL allowed	
	ShippingAddress A szállításí cím cím része, ha van.	varchar(100)	100	NULL allowed	
	InvoicePostcode A számlázási cím irányító száma, ha van.	varchar(10)	10	NULL allowed	
	InvoicegCity A számlázási cím városa, ha van.	varchar(30)	30	NULL allowed	
	InvoiceAddress A számlázási cím cím része, ha van.	varchar(100)	100	NULL allowed	
	ActiveFlag Megmutatja, hogy a partner aktív e még a rendszerben.	bit	1	NOT NULL	



Indexek

Key	Name	Key Columns	Unique
P/C	PK_Partner_PartnerID Elsődleges kulcs.	PartnerID	True
	UC_Partner_EmailAddres Egy email címmel csak egy partner regisztrálhat.	EmailAddress	True

Triggerek

Name	ANSI Nulls On	Quoted Identifier On	On
tPartnerLog A DML utasításokat figyelő trigger, ami az események adatait rögzítik a DatabaseLog táblába.	True	True	After Delete Insert Update

Туре	Action	Owning Principal
Grant	INSERT	Emp_Role
Grant	SELECT	Emp_Role
Grant	UPDATE	Emp_Role



```
CREATE TABLE dbo.Partner(
PartnerID int NOT NULL IDENTITY (1, 1),
LastName varchar (30) COLLATE Hungarian_CI_AS NOT NULL,
FirstName varchar (30) COLLATE Hungarian CI AS NOT NULL,
EmailAddress varchar (50) COLLATE Hungarian CI AS NULL,
PhoneNumber varchar (20) COLLATE Hungarian CI AS NULL,
Postcode varchar (10) COLLATE Hungarian CI AS NULL,
City varchar (30) COLLATE Hungarian CI AS NULL,
Address varchar (100) COLLATE Hungarian CI AS NULL,
ShippingPostcode varchar (10) COLLATE Hungarian_CI_AS NULL,
ShippingCity varchar (30) COLLATE Hungarian CI AS NULL,
ShippingAddress varchar (100) COLLATE Hungarian CI AS NULL,
InvoicePostcode varchar (10) COLLATE Hungarian CI AS NULL,
InvoicegCity varchar (30) COLLATE Hungarian CI AS NULL,
InvoiceAddress varchar (100) COLLATE Hungarian CI AS NULL,
ActiveFlag bit NOT NULL
) ON PRIMARY
CREATE TRIGGER dbo.tPartnerLog
ON dbo.Partner AFTER INSERT , UPDATE, DELETE
BEGIN
   DECLARE @schema varchar(20) = (SELECT OBJECT SCHEMA NAME(object id('Partner')))
   DECLARE @object sysname = (SELECT OBJECT NAME (parent id) AS TableName FROM sys.triggers
                               WHERE name = 'tPartnerLog');
   DECLARE @operation_type VARCHAR(10);
   IF EXISTS (SELECT * FROM INSERTED)
   BEGIN
       IF EXISTS (SELECT * FROM DELETED)
           SET @operation type = 'UPDATE';
       END
       ELSE
       BEGIN
          SET @operation type = 'INSERT';
   END
   ELSE
       SET @operation type = 'DELETE';
    INSERT INTO DatabaseLog (Event, ModifyingUser, EventTime, Schema, Object)
   VALUES (@operation type, USER NAME(), SYSDATETIME(), @schema, @object);
END;
ALTER TABLE dbo.Partner ADD CONSTRAINT PK Partner PartnerID PRIMARY KEY CLUSTERED (PartnerID)
ON PRIMARY
ALTER TABLE dbo.Partner ADD CONSTRAINT UC_Partner_EmailAddres UNIQUE NONCLUSTERED (Email-
Address) ON PRIMARY
GRANT INSERT ON dbo.Partner TO Emp Role
GRANT SELECT ON dbo.Partner TO Emp Role
GRANT UPDATE ON dbo.Partner TO Emp Role
```



dbo.Product

Leírás

A webshopban forgalmazott termékek adatait tartalmazó tábla.

Tulajdonságok

Property	Value
Collation	Hungarian_CI_AS
Row Count (~)	368
Created	12:42:34 2023. április 29., szombat
Last Modified	12:45:11 2023. április 29., szombat

Oszlopok

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
PKP C	ProductID A termék egyedi azonosítója.	int	4	NOT NULL	1 - 1	
	Name A termék neve.	varchar(80)	80	NOT NULL		
	Color A termék színe.	varchar(50)	50	NULL allowed		
	Modell A termék modell elnevezése.	varchar(50)	50	NULL allowed		
	Quality A termék minőségi besorolása. P = Premium, S = Standard	char(2)	2	NULL allowed		
F/P	SubCategoryID Az alkategória azonosítója.	smallint	2	NULL allowed		
	Price A termék lista ára.	money	8	NOT NULL		
FK	BrandID A tremék márkáját azonosító id.	smallint	2	NOT NULL		
ı.t.	ProductIdentifier Egyedi azonosító, amit a raktári rendszer applikációi használnak.	uniqueidentifier	16	NOT NULL		(newid())
	Description A termék leírása.	varchar(max)	max	NULL allowed		
	ProductPhoto A termék fotóinak url címe.	varchar(max)	max	NULL allowed		

Indexek

Key	Name	Key Columns	Unique
PKP C	PK_Product_ProductID Elsődleges kulcs.	ProductID	True
	AK_Product_ProductIdentifier Egy non-clusteres index a gyorsabb keresés érdekében.	ProductIdentifier	



Triggerek

Name	ANSI Nulls On	Quoted Identifier On	On
tProductLog A DML utasításokat figyelő trigger, ami az események adatait rögzíti a DatabaseLog táblába.	True	True	After Delete Insert Update

Megszorítások

Name	Checked	On Column	Constraint
CK_Product_Price A listaár nem lehet negatív.	False	Price	(Price>=(0))
CK_Product_Quality A Quality csak P és S vagy null értékeket vehet fel.	False	Quality	(Quality='P' OR Quality='S')

ldegen kulcsok

Name	Columns
FK_Product_DictBrand_BrandID Idegen kulcs a DictBrand tábla felé.	BrandID->dbo.DictBrand.BrandID
FK_Product_SubCategory_SubCategoryID Idegen kulcs a Subcategory tábla felé.	SubCategoryID->dbo.SubCategory.SubCategoryID

Engedélyek

Туре	Action	Owning Principal
Grant	INSERT	Emp_Role
Grant	SELECT	WebApp
Grant	SELECT	User_Role
Grant	SELECT	Emp_Role
Grant	UPDATE	Emp_Role

```
CREATE TABLE dbo.Product(
ProductID int NOT NULL IDENTITY(1, 1),
Name varchar (80) COLLATE Hungarian_CI_AS NOT NULL,
Color varchar (50) COLLATE Hungarian_CI_AS NULL,
Modell varchar (50) COLLATE Hungarian_CI_AS NULL,
Quality char (2) COLLATE Hungarian_CI_AS NULL,
SubCategoryID smallint NULL,
Price money NOT NULL,
BrandID smallint NOT NULL,
ProductIdentifier uniqueidentifier NOT NULL CONSTRAINT DF_Product_ProductIdentifier DEFAULT (newid()),
Description varchar (max) COLLATE Hungarian_CI_AS NULL,
ProductPhoto varchar (max) COLLATE Hungarian_CI_AS NULL)
```



```
CREATE TRIGGER dbo.tProductLog
ON dbo.Product AFTER INSERT , UPDATE, DELETE
AS
BEGIN
   DECLARE @schema varchar(20) = (SELECT OBJECT SCHEMA NAME(object id('Product')))
   DECLARE @object sysname = (SELECT OBJECT NAME (parent id) AS TableName FROM sys.triggers
                              WHERE name = 'tProductLog');
   DECLARE @operation type VARCHAR(10);
   IF EXISTS(SELECT * FROM INSERTED)
       IF EXISTS (SELECT * FROM DELETED)
          SET @operation_type = 'UPDATE';
       END
       ELSE
        SET @operation type = 'INSERT';
       END
   END
   ELSE
   BEGIN
       SET @operation_type = 'DELETE';
   INSERT INTO DatabaseLog (Event, ModifyingUser, EventTime, Schema, Object)
   VALUES (@operation type, USER NAME(), SYSDATETIME(), @schema, @object);
END;
ALTER TABLE dbo.Product WITH NOCHECK ADD CONSTRAINT CK Product Price CHECK ((Price>=(0)))
ALTER TABLE dbo.Product WITH NOCHECK ADD CONSTRAINT CK Product Quality CHECK ((Quality='P' OR
Quality='S'))
ALTER TABLE dbo.Product ADD CONSTRAINT PK Product ProductID PRIMARY KEY CLUSTERED (ProductID)
CREATE NONCLUSTERED INDEX AK Product ProductIdentifier ON dbo.Product (ProductIdentifier) ON
ALTER TABLE dbo.Product ADD CONSTRAINT FK_Product_DictBrand_BrandID FOREIGN KEY (BrandID)
REFERENCES dbo.DictBrand (BrandID)
ALTER TABLE dbo.Product ADD CONSTRAINT FK Product SubCategory SubCategoryID FOREIGN KEY (Sub-
CategoryID) REFERENCES dbo.SubCategory (SubCategoryID)
GRANT INSERT ON dbo.Product TO Emp Role
GRANT SELECT ON dbo.Product TO Emp Role
GRANT UPDATE ON dbo.Product TO Emp Role
GRANT SELECT ON dbo.Product TO User Role
GRANT SELECT ON dbo.Product TO WebApp
```



■ dbo.ProductProperty

Leírás

A termékek tulajdonságait tartalmazó tábla.

Tulajdonságok

Property	Value	
Collation	Hungarian_CI_AS	
Row Count (~)	1079	
Created	12:42:35 2023. április 29., szombat	
Last Modified	12:45:11 2023. április 29., szombat	

Oszlopok

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK P FKP C	ProductID A termék azonosítója.	int	4	NOT NULL	
PK <mark>P</mark> C	PropertyID Tulajdonság azonosítója.	int	4	NOT NULL	1 - 1
FK	PropertyTypeID A tulajdonság fajtájának azonosítója.	int	4	NULL allowed	
FK	PropertyValueID A tulajdonság értékének azonosítója.	int	4	NULL allowed	
	PropertyValue Maga a tulajdonság értéke.	varchar(50)	50	NULL allowed	

Indexek

Key	Name	Key Columns	Unique
PKP C	PK_ProductProperty_ProductID_PropertyID Elsődleges kulcs. (összetett kulcs)	ProductID, PropertyID	True

Triggerek

Name	ANSI Nulls On	Quoted Identifier On	On
tPropertyLog A DML utasításokat figyelő trigger, ami az események adatait rögzíti a DatabaseLog táblába.	True	True	After Delete Insert Update



ldegen kulcsok

Name	Columns
FK_ProductProperty_Product_ProductID Idegen kulcs a Product tábla felé.	ProductID->dbo.Product.ProductID
FK_ProductProperty_PropertyType_PropertyTypeID Idegen kulcs a PropertyType tábla felé.	PropertyTypeID->dbo.PropertyType.PropertyTypeID
FK_ProductProperty_PropertyValue_PropertyValueID Idegen kulcs a PropertyValue tábla felé.	PropertyValueID->dbo.PropertyValue.PropertyValueID

Engedélyek

Туре	Action	Owning Principal
Grant	INSERT	Emp_Role
Grant	SELECT	Emp_Role
Grant	UPDATE	Emp_Role

```
CREATE TABLE dbo.ProductProperty(
ProductID int NOT NULL,
PropertyID int NOT NULL IDENTITY(1, 1),
PropertyTypeID int NULL,
PropertyValueID int NULL,
PropertyValue varchar (50) COLLATE Hungarian_CI_AS NULL
) ON PRIMARY
GO
CREATE TRIGGER dbo.tPropertyLog
ON dbo.ProductProperty AFTER INSERT , UPDATE, DELETE
AS
BEGIN
   DECLARE @schema varchar(20) = (SELECT OBJECT SCHEMA NAME(object id('ProductProperty')))
   DECLARE @object sysname = (SELECT OBJECT_NAME(parent_id) AS TableName FROM sys.triggers
WHERE name = 'tPropertyLog');
   DECLARE @operation_type VARCHAR(10);
   IF EXISTS(SELECT * FROM INSERTED)
   BEGIN
       IF EXISTS (SELECT * FROM DELETED)
       BEGIN
           SET @operation type = 'UPDATE';
       END
       ELSE
        BEGIN
           SET @operation type = 'INSERT';
        END
    END
    ELSE
    BEGIN
```



```
SET @operation_type = 'DELETE';
    END
    INSERT INTO DatabaseLog (Event, ModifyingUser, EventTime, Schema, Object)
    VALUES (@operation_type, USER_NAME(), SYSDATETIME(), @schema, @object);
END;
ALTER TABLE dbo.ProductProperty ADD CONSTRAINT PK ProductProperty ProductID PropertyID PRIMARY
KEY CLUSTERED (ProductID, PropertyID) ON PRIMARY
ALTER TABLE dbo.ProductProperty ADD CONSTRAINT FK_ProductProperty_Product_ProductID FOREIGN
KEY (ProductID) REFERENCES dbo.Product (ProductID)
ALTER TABLE dbo.ProductProperty ADD CONSTRAINT FK ProductProperty PropertyType PropertyTypeID
FOREIGN KEY (PropertyTypeID) REFERENCES dbo.PropertyType (PropertyTypeID)
ALTER TABLE dbo.ProductProperty ADD CONSTRAINT FK_ProductProperty_PropertyValue_PropertyValue
ID FOREIGN KEY (PropertyValueID) REFERENCES dbo.PropertyValue (PropertyValueID)
GRANT INSERT ON dbo.ProductProperty TO Emp_Role
GRANT SELECT ON dbo.ProductProperty TO Emp_Role
GRANT UPDATE ON dbo.ProductProperty TO Emp Role
```



■ dbo.PropertyType

Leírás

A termékek tulajdonságainak a fajtáit tartalmazó tábla.

Tulajdonságok

Property	Value	
Collation	Hungarian_CI_AS	
Row Count (~)	19	
Created	12:42:35 2023. április 29., szombat	
Last Modified	12:45:11 2023. április 29., szombat	

Oszlopok

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK P C	PropertyTypeID A tulajdonág típusát azonosító egyedi érték.	int	4	NOT NULL	1 - 1
	PropertyTypeName A tulajdonság típusának a neve.	varchar(50)	50	NULL allowed	
F	ParentPropertyTypeID A szülő tulajdonság típus azonosítója.	int	4	NULL allowed	
	PropertyDateType A tulajdonság adattípusa. pl.: list,int stb.	varchar(20)	20	NULL allowed	
	MeasureCode A tulajdonság mértékegysége.	varchar(10)	10	NULL allowed	
	FromValue A felvehető érték minimuma.	int	4	NULL allowed	
	ToValue A felvehető érték maximuma.	int	4	NULL allowed	

Indexek

Key	,	Name	Key Columns	Unique
PK		PK_PropertyType_PropertyTypeID Elsődleges kulcs.	PropertyTypeID	True

ldegen kulcsok

Name	Columns
FK_PropertyType_PropertyType_ParentPropertyType Idegen kulcs a tábla PropertyTypeId-ja felé.	ParentPropertyTypeID->dbo.PropertyType.PropertyTypeID



Engedélyek

Туре	Action	Owning Principal
Grant	INSERT	Emp_Role
Grant	SELECT	Emp_Role
Grant	UPDATE	Emp_Role

```
CREATE TABLE dbo.PropertyType(
PropertyTypeID int NOT NULL IDENTITY(1, 1),
PropertyTypeName varchar (50) COLLATE Hungarian_CI_AS NULL,
ParentPropertyTypeID int NULL,
PropertyDateType varchar (20) COLLATE Hungarian_CI_AS NULL,
MeasureCode varchar (10) COLLATE Hungarian_CI_AS NULL,
FromValue int NULL,
ToValue int NULL
) ON PRIMARY
ALTER TABLE dbo.PropertyType ADD CONSTRAINT PK_PropertyType_PropertyTypeID PRIMARY KEY
CLUSTERED (PropertyTypeID) ON PRIMARY
ALTER TABLE dbo.PropertyType ADD CONSTRAINT FK PropertyType PropertyType ParentPropertyType
FOREIGN KEY (ParentPropertyTypeID) REFERENCES dbo.PropertyType (PropertyTypeID)
GRANT INSERT ON dbo.PropertyType TO Emp_Role
GRANT SELECT ON dbo.PropertyType TO Emp_Role
GRANT UPDATE ON dbo.PropertyType TO Emp Role
```



■ dbo.PropertyValue

Leírás

A tulajdonságok értékeit tartalmazó tábla.

Tulajdonságok

Property	Value	
Collation	Hungarian_CI_AS	
Row Count (~)	48	
Created	12:42:35 2023. április 29., szombat	
Last Modified	12:45:11 2023. április 29., szombat	

Oszlopok

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK <mark>P</mark> C	PropertyValueID A tulajdonság egyedi azonosítója.	int	4	NOT NULL	1 - 1
	PropertyValue A tulajdonság értéke.	varchar(30)	30	NULL allowed	
F	PropertyTypeID A tulajdonség a PropertyType tábla melyik id-jéhez kapcsolódik.	int	4	NULL allowed	

Indexek

Key	Name	Key Columns	Unique
PK <mark>P</mark> C	PK_PropertyValue_PropertyValueID Elsődleges kulcs.	PropertyValueID	True

ldegen kulcsok

Name	Columns
FK_PropertyValue_PropertyType_PropertyTypeID Idegen kulcs a PropertyType tábla felé.	PropertyTypeID->dbo.PropertyType.PropertyTypeID

Туре	Action	Owning Principal
Grant	INSERT	Emp_Role
Grant	SELECT	Emp_Role
Grant	UPDATE	Emp_Role



```
CREATE TABLE dbo.PropertyValue(
PropertyValueID int NOT NULL IDENTITY(1, 1),
PropertyValue varchar (30) COLLATE Hungarian_CI_AS NULL,
PropertyTypeID int NULL
) ON PRIMARY

ALTER TABLE dbo.PropertyValue ADD CONSTRAINT PK_PropertyValue_PropertyValueID PRIMARY KEY
CLUSTERED (PropertyValueID) ON PRIMARY
ALTER TABLE dbo.PropertyValue ADD CONSTRAINT FK_PropertyValue_PropertyType_PropertyTypeID
FOREIGN KEY (PropertyTypeID) REFERENCES dbo.PropertyType (PropertyTypeID)

GRANT INSERT ON dbo.PropertyValue TO Emp_Role
GRANT SELECT ON dbo.PropertyValue TO Emp_Role
GRANT UPDATE ON dbo.PropertyValue TO Emp_Role
GRANT UPDATE ON dbo.PropertyValue TO Emp_Role
```



■ dbo.SubCategory

Leírás

A termékek alkategóráit tartalmazó tábla.

Tulajdonságok

Property	Value	
Collation	Hungarian_CI_AS	
Row Count (~)	14	
Created	12:42:34 2023. április 29., szombat	
Last Modified	12:45:11 2023. április 29., szombat	

Oszlopok

Key	Name	Data Type	Max Length (Bytes)	Nullability
PKP C	SubCategoryID Az egyedi azonosító.	smallint	2	NOT NULL
F/P	CategoryID A kategória azonosítója.	tinyint	1	NOT NULL
ı.	SubCategoryName Az alkategória elnevezése.	varchar(50)	50	NOT NULL

Indexek

Key	Name	Key Columns	Unique
PK C	PK_Subcategory_SubcategoryID Elsődleges kulcs.	SubCategoryID	True
	UC_SubCategory_SubCategoryName Az alkategória elnevezése egyedi kell, hogy legyen.	SubCategoryName	True

Idegen kulcsok

Name	Columns
FK_Subcategory_CategoryID Idegen kulcs a Category tábla felé.	CategoryID->dbo.Category.CategoryID

Туре	Action	Owning Principal
Grant	INSERT	Emp_Role
Grant	SELECT	Emp_Role



```
CREATE TABLE dbo.SubCategory(
SubCategoryID smallint NOT NULL,
CategoryID tinyint NOT NULL,
SubCategoryName varchar (50) COLLATE Hungarian_CI_AS NOT NULL
) ON PRIMARY

ALTER TABLE dbo.SubCategory ADD CONSTRAINT PK_Subcategory_SubcategoryID PRIMARY KEY CLUSTERED (SubCategoryID) ON PRIMARY
ALTER TABLE dbo.SubCategory ADD CONSTRAINT UC_SubCategory_SubCategoryName UNIQUE NONCLUSTERED (SubCategoryName) ON PRIMARY
ALTER TABLE dbo.SubCategory ADD CONSTRAINT FK_Subcategory_Category_CategoryID FOREIGN KEY (CategoryID) REFERENCES dbo.Category (CategoryID)

GRANT INSERT ON dbo.SubCategory TO Emp_Role
GRANT SELECT ON dbo.SubCategory TO Emp_Role
```



dbo.User

Leírás

A webshop alkalmazást használó felhasználók adatai. Mivel az adatbázisban csak adminok és customerek szerepelnek ezért a partner tábla minden egyedéhez tartozik egy user. Az adatbázis bővülésével egyéb partner típusok is bekerülhetnek majd, akikhez nem kell felhasználói regisztráció. Például beszállítók.

Tulajdonságok

Property	Value
Collation Hungarian_CI_AS	
Row Count (~)	433
Created	12:42:35 2023. április 29., szombat
Last Modified	12:45:11 2023. április 29., szombat

Oszlopok

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK <mark>P</mark> C	UserID A felhasználó egyedi azonosítója.	int	4	NOT NULL	1000 - 1
FK	PartnerID A hozzá kapcsolódó partner azonostó.	int	4	NOT NULL	
	UserName Felhasználó név	varchar(50)	50	NULL allowed	
	Password Jelszó.	varchar(100)	100	NULL allowed	
	IsAdmin 1 = Administrator, 0 = Customer	bit	1	NOT NULL	
	AdminRole A dolgozók szerepköre.	varchar(80)	80	NULL allowed	
	JobTitle A dolgozók titulusa.	varchar(80)	80	NULL allowed	
	BirthDate A dolgozók születési dátuma.	date	3	NULL allowed	



Indexek

Key	Name	Key Columns	Unique
P1/C	PK_User_UserID Elsődleges kulcs.	UserID	True

Triggerek

Name	ANSI Nulls On	Quoted Identifier On	On
tUserLog A DML utasításokat figyelő trigger, ami az események adatait rögzíti a DatabaseLog táblába.	True	True	After Delete Insert Update

ldegen kulcsok

Name	Columns
FK_User_Partner_PartnerID Idegen kulcs a partner tábla felé.	PartnerID->dbo.Partner.PartnerID

Туре	Action	Owning Principal
Grant	SELECT	WebApp
Grant	SELECT	User_Role
Grant	SELECT	Emp_Role
Grant	UPDATE	Emp_Role



```
CREATE TABLE dbo.User(
UserID int NOT NULL IDENTITY (1000, 1),
PartnerID int NOT NULL,
UserName varchar (50) COLLATE Hungarian_CI_AS NULL,
Password varchar (100) COLLATE Hungarian CI AS MASKED WITH (FUNCTION = 'default()') NULL,
IsAdmin bit NOT NULL,
AdminRole varchar (80) COLLATE Hungarian CI AS NULL,
JobTitle varchar (80) COLLATE Hungarian CI AS NULL,
BirthDate date NULL
) ON PRIMARY
GO
CREATE TRIGGER dbo.tUserLog
ON dbo.User AFTER INSERT , UPDATE, DELETE
BEGIN
   DECLARE @schema varchar(20) = (SELECT OBJECT_SCHEMA_NAME(object_id('User')))
    DECLARE @object sysname = (SELECT OBJECT NAME (parent id) AS TableName FROM sys.triggers
WHERE name = 'tUserLog');
   DECLARE @operation_type VARCHAR(10);
   IF EXISTS(SELECT * FROM INSERTED)
    BEGIN
       IF EXISTS (SELECT * FROM DELETED)
           SET @operation type = 'UPDATE';
       END
        ELSE
        BEGIN
          SET @operation_type = 'INSERT';
        END
   END
   ELSE
       SET @operation_type = 'DELETE';
    INSERT INTO DatabaseLog (Event, ModifyingUser, EventTime,Schema,Object)
   VALUES (@operation type, USER NAME(), SYSDATETIME(), @schema, @object)
END;
ALTER TABLE dbo.User ADD CONSTRAINT PK User UserID PRIMARY KEY CLUSTERED (UserID) ON PRIMARY
ALTER TABLE dbo.User ADD CONSTRAINT FK User Partner PartnerID FOREIGN KEY (PartnerID)
REFERENCES dbo.Partner (PartnerID)
GRANT SELECT ON dbo.User TO Emp Role
GRANT UPDATE ON dbo.User TO Emp Role
GRANT SELECT ON dbo.User TO User Role
GRANT SELECT ON dbo.User TO WebApp
```



Nézetek

Objektumok

Name

dbo.vCountOfProduct

A webshop aktív időszakában(2020-2022) mennyi egy termék összesen eladott darabszáma, az összes bevétel (a kedvezmények levonásra kerültek), a bevételből mennyi az adó összege.

dbo.vPreviousMonthAmount

A view termékenként hasonlítja össze az előző hónap összegzett bevételét a LineTotal alapján és az eltérést %-osan jelenítjük meg.

dbo.vProductInformation

A termék tulajdonságait egy táblázatban bemutató nézet.

dbo.vQuarterlyStatement

A 2020-as,2021-es és 2022-es év forgalmát jeleníti meg termékenként és negyedéves bontásban.

dbo.vWhoisAdmin

Megmutatja az adatbázisban szereplő administrátorok adatait valamit a, hogy melyikük actív és melyikük nincs már a cégnél.



dbo.vCountOfProduct

Leírás

A webshop aktív időszakában (2020-2022) mennyi egy termék összesen eladott darabszáma, az összes bevétel (a kedvezmények levonásra kerültek), a bevételből mennyi az adó összege.

Tulajdonságok

Property	Value	
Collation	Hungarian_CI_AS	
Created 20:26:33 2023. március 30., csütörtök		
Last Modified	9:59:22 2023. május 6., szombat	

Oszlopok

Name	Data Type	Max Length (Bytes)
ProductID A termék azonosítója.	int	4
Name A termék neve.	varchar(80)	80
CategoryName A termékhez tartozó kategória neve.	varchar(20)	20
SubCategoryName A termékhez tartozó alkategória neve.	varchar(50)	50
CountOfQuantity A termék összes rendelt darabszáma.	int	4
TotalUnitPrice Az termék utáni összes bevétel, amiből a rendeléseknél kapott kedvezmények levonásra kerültek.	varchar(50)	50
TotalTax A termék ára tartalmazza a 27% Áfát. Ez a mező megmutatja, a bevételből mennyi az adó összege.	varchar(50)	50

Туре	Action	Owning Principal
Grant	SELECT	DataAnalystRole





dbo.vPreviousMonthAmount

Leírás

A view termékenként hasonlítja össze az előző hónap összegzett bevételét a LineTotal alapján és az eltérést %-osan jelenítjük meg.

Tulajdonságok

Property	Value
Collation	Hungarian_CI_AS
Created	16:52:07 2023. április 12., szerda
Last Modified	9:59:22 2023. május 6., szombat

Oszlopok

Name	Data Type	Max Length (Bytes)
ProductID A termék azonosítója.	int	4
ProductName A termék neve.	varchar(80)	80
Year A vizsgált év.	int	4
Month A vizsgált hónap.	int	4
Actual A tényleges bevétel az adott hónapban.	money	8
Old Az előző havi bevétel.	money	8
Percent Az eltérés százalékos megjelenítése.	money	8

Туре	Action	Owning Principal
Grant	SELECT	DataAnalystRole





dbo.vProductInformation

Leírás

A termék tulajdonságait egy táblázatban bemutató nézet.

Tulajdonságok

Property	Value
Collation	Hungarian_CI_AS
Created	9:59:22 2023. május 6., szombat
Last Modified	9:59:22 2023. május 6., szombat

Oszlopok

Name	Data Type	Max Length (Bytes)
ProductID A termék azonosítója.	int	4
Name A termék neve.	varchar(80)	80
CategoryName A termék melyik kategóriába tertozik.	varchar(20)	20
SubCategoryName A termék melyik alkategóriába tartozik.	varchar(50)	50
Color A termék színe.	varchar(50)	50
Modell A termék modell elnevezése.	varchar(50)	50
ParentPropertyTypeName A tulajdonság fő besorolása.	varchar(50)	50
PropertyTypeName A tulajdonság fajtájánalk neve.	varchar(50)	50
Property A tulajdonság értéke.	varchar(61)	61

```
CREATE VIEW dbo.vProductInformation AS

SELECT P.ProductID, P.Name, C.CategoryName, SC.SubCategoryName, P.Color,

P.Modell, PPT.PropertyTypeName ParentPropertyTypeName, PT.PropertyTypeName,

COALESCE (PP.PropertyValue + COALESCE (' ' + PT.MeasureCode, ''), PV.PropertyValue)

Property

FROM Product P

LEFT JOIN ProductProperty PP ON P.ProductID = PP.ProductID

LEFT JOIN PropertyType PT ON PP.PropertyTypeID = PT.PropertyTypeID

LEFT JOIN PropertyType PPT ON PT.ParentPropertyTypeID = PPT.PropertyTypeID

LEFT JOIN PropertyValue PV ON PP.PropertyValueID = PV.PropertyValueID

LEFT JOIN dbo.SubCategory SC ON P.SubCategoryID = SC.SubCategoryID

LEFT JOIN dbo.Category C ON SC.CategoryID = C.CategoryID
```



dbo.vQuarterlyStatement

Leírás

A 2020-as,2021-es és 2022-es év forgalmát jeleníti meg termékenként és negyedéves bontásban.

Tulajdonságok

Property	Value	
Collation	Hungarian_CI_AS	
Created	16:39:30 2023. április 12., szerda	
Last Modified	9:59:22 2023. május 6., szombat	

Oszlopok

Name	Data Type	Max Length (Bytes)
ProductID A termék azonosítója.	int	4
Name A termék neve.	varchar(80)	80
Y2020Q1 2020 első negyedévi bevétel.	money	8
Y2020Q2 2020 második negyedévi bevétel.	money	8
Y2020Q3 2020 harmadik negyedévi bevétel.	money	8
Y2020Q4 2020 negyedik negyedévi bevétel.	money	8
Y2021Q1 2021 első negyedévi bevétel.	money	8
Y2021Q2 2021 második negyedévi bevétel.	money	8
Y2021Q3 2021 harmadik negyedévi bevétel.	money	8
Y2021Q4 2021 negyedik negyedévi bevétel.	money	8
Y2022Q1 2022 első negyedévi bevétel.	money	8
Y2022Q2 2022 második negyedévi bevétel.	money	8
Y2022Q3 2022 harmadik negyedévi bevétel.	money	8
Y2022Q4 2022 negyedik negyedévi bevétel.	money	8

Туре	Action	Owning Principal
Grant	SELECT	DataAnalystRole



```
{\it CREATE} \qquad {\it VIEW } \ {\it dbo.vQuarterlyStatement } \ {\it AS}
   SELECT x.ProductID, x.Name, x.Y2020Q1, x.Y2020Q2, x.Y2020Q3, x.Y2020Q4,
                                 x.Y2021Q1, x.Y2021Q2, x.Y2021Q3, x.Y2021Q4,
                                 x.Y2022Q1, x.Y2022Q2, x.Y2022Q3, x.Y2022Q4
    FROM (
        SELECT p.ProductID, p.Name, 'Y' + CAST(YEAR(OH.OrderDate) AS char(4)) +
            'Q' + CAST (DATEPART (q,OH.OrderDate) AS char(1)) Period, OI.LineTotal
        FROM OrderHeader OH
        INNER JOIN orderitem OI ON OH.OrderID = OI.OrderID
        INNER JOIN Product P ON OI.ProductID = P.ProductID
        WHERE OH.OrderDate BETWEEN '20200101' AND '20221231') D
   PIVOT
    (SUM(LineTotal)
        FOR Period IN (Y2020Q1, Y2020Q2, Y2020Q3, Y2020Q4, Y2021Q1, Y2021Q2, Y2021Q3,
                       Y2021Q4, Y2022Q1, Y2022Q2, Y2022Q3, Y2022Q4)) x
GO
GRANT SELECT ON dbo.vQuarterlyStatement TO DataAnalystRole
```



dbo.vWhoisAdmin

Leírás

Megmutatja az adatbázisban szereplő adminisztrátorok adatait valamit, hogy melyikük actív és melyikük nincs már a cégnél.

Tulajdonságok

Property	Value
Collation	Hungarian_CI_AS
Created	13:50:19 2023. április 2., vasárnap
Last Modified	9:59:22 2023. május 6., szombat

Oszlopok

Name	Data Type	Max Length (Bytes)
UserID A felhasználó egyedi azonosítója.	int	4
AdminName Az adminisztrátor teljes neve.	varchar(61)	61
EmailAddress Az adminisztrátor email címe.	varchar(50)	50
PhoneNumber Az adminisztrátor telefonszáma.	varchar(20)	20
BirthDate Az adminisztrátor születési dátuma.	date	3
JobTitle Az adminisztrátor titulusa.	varchar(80)	80
Status Az adminisztrátor státusza. (Aktív, Inaktív)	varchar(7)	7

```
CREATE VIEW dbo.vWhoisAdmin AS

SELECT u.UserID ,CONCAT(p.LastName, + ' ' + p.FirstName) AdminName,

p.EmailAddress,p.PhoneNumber,u.BirthDate,u.JobTitle,

CASE p.ActiveFlag WHEN '1' THEN 'Aktív'

ELSE 'Inaktív'

END AS "Status"

FROM dbo.User u

INNER JOIN dbo.Partner p ON u.PartnerID = p.PartnerID

WHERE u.IsAdmin = 1
```



🖺 Tárolt eljárások

Objektumok

Name

dbo.AddOrder

Új rendelést fölvevő tárolt eljárás.

A tárolt eljárás megvizsgálja, hogy van e létező felhasználó a webshoppban. A rendelés regisztrációhoz kötött. Nemlétező felhasználó esetén 1-es hibakód.

Megvizsgálja, hogy létezik-e a megadott azonosítójú termék és a darabszám ki van e töltve és legalább egy terméket rendeltek e. Hibás adatok esetén a hibakód: 2.

Egyéb rendszerhiba esetén (nem létező paymenttypeid vagy deliverytypeid) esetén a hibakód:3.

dbo.AddPartner

Új partner és user felvétele.

Az adatbázis jelen szakaszában minden partner vagy vásárló vagy adminisztrátor így minden partnernek kell egy felhasználó fiók is a rendszerben.

A tárolt eljárás vizsgálja, hogy a megadott emailcímhez tartozik e már felhasználó. Valamint, hogy az email cím helyesen van megadva (EmailAddressCheck függvény). Egy email címhez csak egy felhasználó tartozhat. Hibakód: 1.

Egyéb rendszerhiba vagy megszorítás ütközés esetén a hibakód: 2.

Sikeres tranzakció esetén a visszatérési érték: 0.

dbo.AddProduct

Új termék fölvételét végző tárolt eljárás.

Vizsgálja, hogy létezik-e ilyen nevű termék az adatbázisban és a név mező ki van e töltve(kötelező). Ha igen akor hibakód= 1.

A tesztadatoknál vannak azonos nevű termékek, de új terméket nem lehet már azonos néven fölvenni, ezért itt ellenőrzöm és nem constraint-el)

Egyéb rendszerhiba vagy megszorítás ütközés esetén hibakód= 2.

Sikeres tranzakció esetén visszatérési érték = 0.

dbo.MonthlyStatement

A havi bevétel az egyes termékekre nézve egy paraméterként megadott évben.

A forint megielenetíse a megrendelő kifejezett kérése volt! Készült hozzá egy függyény arra az esetre ha még szükség lenne rá.

(A kivitelezés így nem javasolt mert az érték így string-ként kerül elmentésre így nem lehet vele numerikus műveleteket végezni.)

dbo.TestDatasImport

Az adatbázis teszteléséhez szükséges tesztadatok feltöltését végző tárolt eljárás ami a globál táblában elmentett elérési utakból tölti fel az ott található csv fájlokat a megfelelő táblákba.

dbo.UpdatePartner

A partner és user tábla adatait módosító tárolt eljárás.

A tárolt eljárás megvizsgálja, hogy a létezik-e a partner a rendszerben és az email cím helyesen van e megadva. Ha nem akkor hibakód: 1.

Egyéb rendszerhiba esetén (létező emailcím) a hibakód: 2.

Ha nincs új adat megadva akkor alapértelmezetten a régi adat marad a táblában.

dbo.UpdateProduct

A termék árát vagy leírását módosító tárolt eljárás. A termék egyéb paramétereinek a változtatása esetén új tételként kell fölvenni a terméket.

Vizsgálja, hogy a termék létezik-e az adatbázisban. Ha nem akkor hibakód: 1.

Ha a módosított ár ütközik a megszorításokkal akkor hibakód:2.

Sikeres tranzakció esetén a visszatérési érték: 0.

H nincs megadva új érték akkor alapértelmezetten a régi marad a táblában.



dbo.AddOrder

Leírás

Új rendelést fölvevő tárolt eljárás.

A tárolt eljárás megvizsgálja, hogy van e létező felhasználó a webshoppban. A rendelés regisztrációhoz kötött. Nemlétező felhasználó esetén 1-es hibakód.

Megvizsgálja, hogy létezik-e a megadott azonosítójú termék és a darabszám ki van e töltve és legalább egy terméket rendeltek e. Hibás adatok esetén a hibakód: 2.

Egyéb rendszerhiba esetén (nem létező paymenttypeid vagy deliverytypeid) esetén a hibakód:3.

Paraméterek

Name	Data Type	Max Length (Bytes)	Direction
@Customerld A vevő azonosítója.	int	4	
@PaymentTypeId A fizetési mód azonosítója.	int	4	
@DeliveryTypeld A kiszállítási mód azonosítoja.	int	4	
@ProductldsAndQuantities A temék azonosítót és megrendelt darabszámotmegadó változó.	ProductIdAnd- Quantity	max	
@Comment A rendeléshez opcionálisan megadható megjegyzés.	varchar(1000)	1000	
@Orderstatus A rendelés státuszát visszadó érték. Alapértelmezetten "Elérhető".	varchar(100)	100	
@result Visszatérési érték a sikeres tranzakciohoz vagy esetleges hibák értelmezéséhez.	int	4	Out

Engedélyek

Туре	Action	Owning Principal
Grant	EXECUTE	WebApp
Grant	EXECUTE	User_Role

```
CREATE PROCEDURE dbo.AddOrder

@CustomerId int,
@PaymentTypeId int,
@DeliveryTypeId int,
@ProductIdsAndQuantities as dbo.ProductIdAndQuantity readonly,
@Comment varchar(1000) = Null ,
@Orderstatus varchar(100) = 'Elérhető',
@result INT OUTPUT
AS
```



```
BEGIN
   SET NOCOUNT ON;
   BEGIN TRY
       BEGIN TRANSACTION
       IF NOT EXISTS (SELECT * FROM dbo.Partner WHERE PartnerID = @CustomerId)
        BEGIN
           SET @result = 1;
        ELSE IF EXISTS (
           SELECT 1
           FROM @ProductIdsAndQuantities p
           LEFT JOIN Product pr ON p.ProductId = pr.ProductID
           WHERE p.Quantity < 1 OR pr.ProductID IS NULL
        BEGIN
           SET @result = 2;
        END
        ELSE
        BEGIN
           INSERT dbo.OrderHeader (Customerid, PaymentTypeID, DeliveryTypeID, Comment, Order
                                    Status)
           VALUES (@CustomerId, @PaymentTypeId, @DeliveryTypeId,@Comment,@Orderstatus)
            DECLARE @orderId int = scope identity()
            INSERT dbo.OrderItem (OrderID, ProductID, Quantity, UnitPrice)
            SELECT @orderId, p.ProductID, p.Quantity, pr.Price
            FROM @ProductIdsAndQuantities p
            INNER JOIN Product pr ON pr.ProductID = p.ProductId
            UPDATE dbo.OrderHeader
            SET SubTotal = (SELECT SUM(linetotal) LineTotal FROM OrderItem
                           WHERE OrderID = @orderId GROUP BY orderId)
            WHERE OrderID = @orderId
            UPDATE dbo.OrderHeader
            SET Freight = dbo.ShippingFee(SubTotal), Discount = dbo.Discount(SubTotal)
            WHERE OrderID = @orderId
           SET @result = 0;
        END
        IF XACT STATE() > 0 COMMIT TRAN
   END TRY
    BEGIN CATCH
       IF XACT STATE() = -1
           ROLLBACK TRAN;
        ELSE IF XACT STATE() = 1
           COMMIT TRAN;
        SET @result = 3;
    END CATCH
END
GRANT EXECUTE ON dbo.AddOrder TO User_Role
GRANT EXECUTE ON dbo.AddOrder TO WebApp
```



dbo.AddPartner

Leírás

Új partner és user felvétele.

Az adatbázis jelen szakaszában minden partner vagy vásárló vagy adminisztrátor így minden partnernek kell egy felhasználó fiók is a rendszerben.

A tárolt eljárás vizsgálja, hogy a megadott emailcímhez tartozik e már felhasználó. Valamint, hogy az email cím helyesen van megadva(EmailAddressCheck függvény). Egy email címhez csak egy felhasználó tartozhat. Hibakód: 1.

Egyéb rendszerhiba vagy megszorítás ütközés esetén a hibakód: 2.

Sikeres tranzakció esetén a visszatérési érték: 0.

Paraméterk

Name	Data Type	Max Length (Bytes)	Direction
@LastName Vezetéknév.	varchar(30)	30	
@FirstName Kereszt név.	varchar(30)	30	
@EmailAddress Email cím.	varchar(50)	50	
@PhoneNumber Telefonszám.	varchar(20)	20	
@PostCode Irányító szám.	varchar(10)	10	
@City Város.	varchar(30)	30	
@Address Cím.	varchar(100)	100	
@ActiveFlag Aktív, inaktív státusz megadása.	bit	1	
@UserName Felhasználó név.	varchar(50)	50	
@Password Jelszó.	varchar(100)	100	
@IsAdmin Adminisztror = 1, vevő=0.	bit	1	
@result Viszatérési érték.	int	4	Out

Туре	Action	Owning Principal
Grant	EXECUTE	WebApp
Grant	EXECUTE	User_Role
Grant	EXECUTE	Emp_Role



```
CREATE PROCEDURE dbo.AddPartner
   @LastName varchar(30),
   @FirstName varchar(30),
   @EmailAddress varchar(50),
   @PhoneNumber Varchar(20) ,
   @PostCode varchar(10),
   @City Varchar(30),
   @Address Varchar(100),
   @ActiveFlag bit = 1, -- 1 active , 0 inactive
   @UserName varchar(50),
   @Password varchar(100),
   @IsAdmin bit = 0,--1 admin, 0 user
   @result INT OUTPUT
AS
BEGIN
   SET NOCOUNT ON
   BEGIN TRY
       BEGIN TRAN;
           IF NOT EXISTS (SELECT * FROM dbo.Partner WHERE EmailAddress = @EmailAddress OR
                          dbo.EmailAddressCheck(@EmailAddress) = 1)
            BEGIN
            INSERT dbo.Partner (LastName, FirstName, EmailAddress, Phone
                               Number, Postcode, City, Address, ActiveFlag)
            VALUES (@LastName, @FirstName, @EmailAddress, @PhoneNumber,
                   @PostCode,@City,@Address,@ActiveFlag)
            DECLARE @PartnerId int = scope identity()
            INSERT dbo.User (PartnerId, UserName, Password, IsAdmin)
            VALUES (@PartnerId, @UserName, @Password, @IsAdmin)
            SET @result = 0
            END
            ELSE
               BEGIN
                SET @result = 1
               END
        IF XACT STATE() = 1 COMMIT TRAN
   END TRY
   BEGIN CATCH
       IF XACT STATE() = -1
           ROLLBACK TRAN;
       ELSE IF XACT STATE() = 1
           COMMIT TRAN;
       SET @result = 2;
   END CATCH
END
GRANT EXECUTE ON dbo.AddPartner TO Emp_Role
GRANT EXECUTE ON dbo.AddPartner TO User Role
GRANT EXECUTE ON dbo.AddPartner TO WebApp
```



dbo.AddProduct

Leírás

Új termék fölvételét végző tárolt eljárás.

Vizsgálja, hogy létezik-e ilyen nevű termék az adatbázisban és a név mező ki van e töltve(kötelező). Ha igen akor hibakód= 1.

A tesztadatoknál vannak azonos nevű termékek, de új terméket nem lehet már azonos néven fölvenni, ezért itt ellenőrzöm és nem constraint-el)

Egyéb rendszerhiba vagy megszorítás ütközés esetén hibakód= 2.

Sikeres tranzakció esetén visszatérési érték = 0.

Paraméterek

Name	Data Type	Max Length (Bytes)	Direction
@ProductName A termék neve. Kötelező megadni.	varchar(80)	80	
@Color Szín.	varchar(80)	80	
@Modell Modell.	varchar(50)	50	
@Quality Minőség. P = prémium, S = standard.	char(2)	2	
@SubCategoryId Alkategória azonosítója.	int	4	
@Price Egységár.	money	8	
@Brandld A márka azonosítója.	smallint	2	
@property Egyéb tulajdonságok értékei.	Property	max	
@result Visszatérési érték.	int	4	Out

Туре	Action	Owning Principal
Grant	EXECUTE	Emp_Role



```
CREATE PROCEDURE dbo.AddProduct
   @ProductName varchar(80),
   @Color varchar(80),
   @Modell Varchar(50) ,
   @Quality Char(2),
   @SubCategoryId int,
   @Price money,
   @BrandId smallint,
   @property as dbo.Property READONLY,
    @result INT OUTPUT
AS
BEGIN
   SET NOCOUNT ON
   BEGIN TRY
     BEGIN TRAN;
        IF @ProductName IS NOT NULL And @ProductName NOT IN (SELECT name FROM dbo.Product)
            INSERT dbo.Product (Name, Color, Modell, Quality, SubCategoryID, Price, BrandID)
            VALUES (@productName, @Color, @Modell, @Quality, @SubCategoryID, @Price, @BrandID)
            DECLARE @ProductId int = scope identity() --Az utolsó Id-t adja vissza
           INSERT dbo.ProductProperty(ProductID, PropertyTypeID, PropertyValueID, PropertyValue)
            SELECT @ProductId,p.PropertyTypeId,p.PropertyValueid,p.PropertyValue
            FROM @property p
            SET @result = 0
        END
        ELSE
           BEGIN
              SET @result = 1
           END
   IF XACT STATE() = 1 COMMIT TRAN
   END TRY
    BEGIN CATCH
       IF XACT_STATE() = -1
           ROLLBACK TRAN;
        ELSE IF XACT STATE() = 1
           COMMIT TRAN;
        SET @result = 2;
    END CATCH
END
GRANT EXECUTE ON dbo.AddProduct TO Emp_Role
```



dbo.MonthlyStatement

Leírás

A havi bevétel az egyes termékekre nézve egy paraméterként megadott évben.

A forint megjelenetíse a megrendelő kifejezett kérése volt! Készült hozzá egy függvény arra az esetre, ha még szükség lenne rá.

(A kivitelezés így nem javasolt mert az érték így string-ként kerül elmentésre így nem lehet vele numerikus műveleteket végezni.)

Paraméterek

Name	Data Type	Max Length (Bytes)
@year A megtekinteni kívánt év.	int	4

Туре	Action	Owning Principal
Grant	EXECUTE	DataAnalystRole



```
CREATE PROCEDURE dbo.MonthlyStatement
     @year int
AS
SET NOCOUNT ON
SELECT ProductID, ProductName,
       ISNULL(dbo.Forint(1), dbo.Forint(0)) January,
       ISNULL(dbo.Forint(2), dbo.forint(0)) February,
       ISNULL(dbo.Forint(3), dbo.forint(0)) March,
       ISNULL(dbo.Forint(4), dbo.forint(0)) April,
       ISNULL(dbo.Forint(5), dbo.forint(0)) May,
      ISNULL(dbo.Forint(6), dbo.forint(0)) June,
      ISNULL(dbo.Forint(7), dbo.forint(0)) July,
      ISNULL(dbo.Forint(8), dbo.forint(0)) August,
      ISNULL(dbo.Forint(9), dbo.forint(0)) September,
       ISNULL(dbo.Forint(10),dbo.forint(0)) October,
       ISNULL(dbo.Forint(11),dbo.forint(0)) November,
       ISNULL(dbo.Forint(12),dbo.forint(0)) December
FROM (
 SELECT ot.ProductID ProductID, p.Name ProductName, MONTH(oh.OrderDate) OrderMonth,
  sum((ot.UnitPrice * ot.Quantity)-((ot.UnitPrice * ot.Quantity)*oh.Discount)) Unitprice
 FROM OrderItem ot
  INNER JOIN OrderHeader oh ON ot.OrderID = oh.OrderID
  INNER JOIN Product p on p.ProductID = ot.ProductID
 WHERE YEAR(oh.OrderDate) = @year
 GROUP BY ot.ProductID, p.Name , MONTH(oh.OrderDate)
) x
PIVOT (
 SUM(unitprice)
  FOR OrderMonth IN (1,2,3,4,5,6,7,8,9,10,11,12)
ORDER BY ProductID
GRANT EXECUTE ON dbo.MonthlyStatement TO DataAnalystRole
```



dbo.TestDatasImport

Leírás

Az adatbázis teszteléséhez szükséges tesztadatok feltöltését végző tárolt eljárás, ami a globál táblában elmentett elérési utakból tölti fel az ott található csv fájlokat a megfelelő táblákba.

```
CREATE
         PROCEDURE dbo.TestDatasImport AS
BEGIN
   BEGIN TRY
       BEGIN TRAN
       DECLARE @SQL NVARCHAR (MAX)
        DECLARE @PartnerDataPath NVARCHAR(500)
        DECLARE @UserDataPath NVARCHAR(500)
       DECLARE @OrderHeaderDataPath NVARCHAR (500)
       DECLARE @OrderItemDataPath NVARCHAR(500)
       DECLARE @PropertyValueDataPath NVARCHAR(500)
       DECLARE @PropertyTypeDataPath NVARCHAR(500)
       DECLARE @ProductPropertyDataPath NVARCHAR(500)
       DECLARE @ProductDataPath NVARCHAR(500)
        SELECT @PartnerDataPath = Value FROM dbo.global WHERE GlobalID = 8
       SELECT @UserDataPath = Value FROM dbo.global WHERE GlobalID = 2
       SELECT @OrderHeaderDataPath = Value FROM dbo.global WHERE GlobalID = 4
       SELECT @OrderItemDataPath = Value FROM dbo.qlobal WHERE GlobalID = 5
       SELECT @ProductDataPath = Value FROM dbo.global WHERE GlobalID = 1
        SELECT @ProductPropertyDataPath = Value FROM dbo.global WHERE GlobalID = 3
        SELECT @PropertyTypeDataPath = Value FROM dbo.global WHERE GlobalID = 6
        SELECT @PropertyValueDataPath = Value FROM dbo.global WHERE GlobalID = 7
        SET @SQL =
            'bulk insert dbo.partner
            from ''' + @PartnerDataPath + '''
            WITH ( FIELDTERMINATOR = '';'', ROWTERMINATOR = ''\n'', CODEPAGE = ''65001'',
                  firstrow = 2,fire_triggers)
            bulk insert dbo.user
            from ''' + @UserDataPath + '''
            WITH ( FIELDTERMINATOR = '';'', ROWTERMINATOR = ''\n'', CODEPAGE = ''65001'',
                 firstrow = 2, fire triggers)
            bulk insert dbo.product
            from ''' + @ProductDataPath + '''
            WITH ( FIELDTERMINATOR = '';'', ROWTERMINATOR =''\n'', CODEPAGE = ''65001'',
                  firstrow = 2, fire_triggers)
```



```
from ''' + @OrderHeaderDataPath + '''
            WITH ( FIELDTERMINATOR = '';'', ROWTERMINATOR = ''\n'', CODEPAGE = ''65001'',
                   firstrow = 2,fire triggers)
            bulk insert dbo.OrderItem
            from ''' + @OrderItemDataPath + '''
            WITH ( FIELDTERMINATOR = '';'', ROWTERMINATOR = ''\n'', CODEPAGE = ''65001'',
                   firstrow = 2,fire triggers)
            bulk insert dbo.ProductProperty
            from ''' + @ProductPropertyDataPath + '''
            WITH (FIELDTERMINATOR = '';'', ROWTERMINATOR = ''\n'', CODEPAGE = ''65001'',
                   firstrow = 2,fire_triggers)
            bulk insert dbo.PropertyValue
            from ''' + @PropertyValueDataPath + '''
            WITH (FIELDTERMINATOR = '';'', ROWTERMINATOR =''\n'', CODEPAGE = ''65001'',
                  firstrow = 2,fire triggers)
           bulk insert dbo.PropertyType
            from ''' + @PropertyTypeDataPath + '''
            WITH (FIELDTERMINATOR = '';'', ROWTERMINATOR = ''\n'', CODEPAGE = ''65001'',
                   firstrow = 2,fire_triggers)'
       EXEC sp executesql @SQL
   COMMIT TRAN;
       RETURN 0;
    END TRY
    BEGIN CATCH
       IF @@TRANCOUNT > 0 ROLLBACK TRAN
       RETURN 1;
   END CATCH
END
```



dbo.UpdatePartner

Leírás

A partner és user tábla adatait módosító tárolt eljárás.

A tárolt eljárás megvizsgálja, hogy a létezik-e a partner a rendszerben és az email cím helyesen van e megadva. Ha nem akkor hibakód: 1.

Egyéb rendszerhiba esetén (létező emailcím) a hibakód: 2.

Ha nincs új adat megadva akkor alapértelmezetten a régi adat marad a táblában.

Paraméterek

Name	Data Type	Max Length (Bytes)	Direction
@PartnerID A partner azonosítója.	int	4	
@LastName Vezetéknév.	varchar(30)	30	
@FirstName Kereszt név.	varchar(30)	30	
@EmailAddress Email cím.	varchar(50)	50	
@PhoneNumber Telefonszám.	varchar(20)	20	
@PostCode Irányító szám.	varchar(10)	10	
@City Város.	varchar(30)	30	
@Address Cím.	varchar(100)	100	
@ActiveFlag Aktív inaktív státusz.	bit	1	
@Password Jelszó.	varchar(100)	100	
@IsAdmin Adminisztrátor = 1, Customer = 0.	bit	1	
@result Viszatérési érték.	int	4	Out

Туре	Action	Owning Principal
Grant	EXECUTE	WebApp
Grant	EXECUTE	User_Role
Grant	EXECUTE	Emp_Role



```
CREATE PROCEDURE dbo.UpdatePartner
   @PartnerID int,
   @LastName\ VARCHAR(30) = NULL,
   @FirstName VARCHAR(30) = NULL,
   @EmailAddress varchar(50) = null,
   @PhoneNumber Varchar(20) = null,
   @PostCode varchar(10) = null,
   @City Varchar(30) = null,
   @Address Varchar(100) = null,
   @ActiveFlag bit , -- 1 active , 0 inactive
   @Password varchar(100) = null,
   @IsAdmin bit = 0 , --1 admin, 0 user
   @result INT OUTPUT
AS
BEGIN
  SET NOCOUNT ON
       BEGIN TRY
            BEGIN TRAN;
                 IF EXISTS (SELECT * FROM partner WHERE partnerId = @partnerId and
                           dbo.EmailAddressCheck(@EmailAddress) = 0 )
                        UPDATE dbo.Partner
                        SET LastName = COALESCE(@LastName, LastName),
                            FirstName = COALESCE(@FirstName, FirstName),
                            EmailAddress = COALESCE(@EmailAddress,EmailAddress),
                            PhoneNumber = COALESCE(@PhoneNumber, PhoneNumber),
                            PostCode = COALESCE (@PostCode, Postcode),
                            City = COALESCE (@City,City),
                            Address = COALESCE (@Address, Address),
                            ActiveFlag = COALESCE(@ActiveFlag, ActiveFlag)
                        WHERE PartnerID = @PartnerID
                        UPDATE dbo.User
                        SET Password = COALESCE (@Password, Password),
                             IsAdmin = COALESCE(@IsAdmin, Isadmin)
                        WHERE PartnerID = @PartnerID
                        SET @result = 0
                   END
                ELSE
                BEGIN
                 SET @result = 1
                END
         IF XACT STATE() = 1 COMMIT TRAN
         END TRY
         BEGIN CATCH
           IF XACT_STATE() = -1
                 ROLLBACK TRAN;
            ELSE IF XACT STATE() = 1
               COMMIT TRAN;
            SET @result = 2
       END CATCH
END
GRANT EXECUTE ON dbo.UpdatePartner TO Emp Role
GRANT EXECUTE ON dbo.UpdatePartner TO User Role, GRANT EXECUTE ON dbo.UpdatePartner TO WebApp
```



dbo.UpdateProduct

Lerás

A termék árát vagy leírását módosító tárolt eljárás. A termék egyéb paramétereinek a változtatása esetén új tételként kell fölvenni a terméket.

Vizsgálja, hogy a termék létezik-e az adatbázisban. Ha nem akkor hibakód: 1.

Ha a módosított ár ütközik a megszorításokkal akkor hibakód:2.

Sikeres tranzakció esetén a visszatérési érték: 0.

H nincs megadva új érték akkor alapértelmezetten a régi marad a táblában.

Paraméterek

Name	Data Type	Max Length (Bytes)	Direction
@productId A termák azonosítója.	int	4	
@newPrice Új egységár.	decimal(10,2)	9	
@newDescription Új leírás a termékhez.	varchar(max)	max	
@result Visszítérési érték.	int	4	Out

Туре	Action	Owning Principal
Grant	EXECUTE	Emp_Role



```
CREATE PROCEDURE dbo.UpdateProduct
   @productId INT,
   @newPrice DECIMAL(10,2) = NULL,
   @newDescription VARCHAR(MAX) = NULL,
   @result INT OUTPUT
AS
BEGIN
   SET NOCOUNT ON
    BEGIN TRY
     BEGIN TRAN;
   IF EXISTS (SELECT * FROM dbo.Product WHERE ProductId = @productId)
       BEGIN
           UPDATE dbo.Product
           SET price = COALESCE(@newPrice, price),
               Description = COALESCE(@newDescription, description)
           WHERE productId = @productId;
    SET @result = 0
       END
       ELSE
           BEGIN
             SET @result = 1
           END
   IF XACT STATE() = 1 COMMIT TRAN
   END TRY
   BEGIN CATCH
      IF XACT STATE() = -1
          ROLLBACK TRAN;
       ELSE IF XACT_STATE() = 1
           COMMIT TRAN;
       SET @result = 2;
   END CATCH
END
GRANT EXECUTE ON dbo.UpdateProduct TO Emp_Role
```



Objektumok

Name

dbo.fRepeatCustomer

A paraméterként megadott évben a megadott számnál többször vásárló vevőket adja vissza függvény.



dbo.fRepeatCustomer

Leírás

A paraméterként megadott évben a megadott számnál többször vásárló vevőket adja vissza függvény.

Paraméterek

Name	Data Type	Max Length (Bytes)
@year	int	4
@CountOfPurchases	int	4

Engedélyek

Туре	Action	Owning Principal
Grant	SELECT	DataAnalystRole

```
CREATE FUNCTION dbo.fRepeatCustomer(@year int,@CountOfPurchases int )

RETURNS TABLE AS RETURN

SELECT oh.CustomerID,concat(p.LastName, ' ' + p.FirstName, ' ' +'(' +u.UserName +')')

CustomerNameAndUserName

FROM OrderHeader oh

INNER JOIN Partner p ON p.PartnerID = oh.Customerid

INNER JOIN user u ON p.PartnerID = u.PartnerID

WHERE Year(oh.OrderDate) = @year

GROUP BY oh.CustomerID,concat(p.LastName, ' ' + p.FirstName, ' ' +'(' +u.UserName +')')

HAVING Count(1) >= @CountOfPurchases

GRANT SELECT ON dbo.fRepeatCustomer TO DataAnalystRole
```





Skalár függvények

Objektumok

Name

dbo.Discount

A kedvezmény mértékét adja vissza a függvény. 700 ezer fölött 20%-os kedvezmény, 500 ezer fölötti rendelés esetén 15% -os kedvezmény és 400 ezer fölötti rendelés esetén 10%-os kedvezmény.

dbo.EmailAddressCheck

Ellenőrzi a beírt email cím helyességét. (tartalmaz e @ jelet és pontot)

A megadott összeg mögé írj a magyar valuta jelét(Ft). Kifejezetten a megrendelő kérése volt ennek a függvénynek a használata.

dbo.ShippingFee

Kiszámítja a szállítási díjjat. A limit és a határérték a global táblában tárolt adat. Jelen állás szerint 25000 forimt összegű vásárlás felett díjtalan a kiszállítás, de az érték alatt 2500ft szállítási díj.





A kedvezmény mértékét adja vissza a függvény. 700 ezer fölött 20%-os kedvezmény, 500 ezer fölötti rendelés esetén 15% -os kedvezmény és 400 ezer fölötti rendelés esetén 10%-os kedvezmény.

Paraméterek

Name	Data Type	Max Length (Bytes)
@amount	money	8
Vásárlás összege, amiből kalkulálható a kedvezmény.		

```
CREATE FUNCTION dbo.Discount (@amount money) RETURNS decimal(4,2) AS

BEGIN

DECLARE @discount decimal(4,2)

SELECT @discount = CASE

WHEN @amount > 700000 THEN 0.2

WHEN @amount > 500000 THEN 0.15

WHEN @amount > 400000 THEN 0.1

ELSE 0

END

RETURN @discount

END
```



dbo.EmailAddressCheck

Leírás

Ellenőrzi a beírt email cím helyességét. (tartalmaz e @ jelet és pontot)

Paraméterek

Name	Data Type	Max Length (Bytes)
@email Az email cím.	varchar(50)	50

```
CREATE FUNCTION dbo.EmailAddressCheck (@email varchar(50)) Returns bit As
BEGIN
DECLARE @result BIT

IF @email like '%@%.%'
    SET @result = 0 -- Helyes email cim!
ELSE
    SET @result = 1 -- Hibás email cim!
    RETURN @result
END
```





A megadott összeg mögé írj a magyar valuta jelét (Ft). Kifejezetten a megrendelő kérése volt ennek a függvénynek a használata.

Paraméterek

Name	Data Type	Max Length (Bytes)
@amount A megadott összeg, amihez hozzáfűzi a Ft végződést.	money	8

```
CREATE FUNCTION dbo.Forint (@amount money) RETURNS varchar(50) AS
BEGIN

DECLARE @result varchar(50)

SET @result = CAST(@amount AS varchar(50)) + ' Ft'

RETURN @result

END
```



dbo.ShippingFee

Leírás

Kiszámítja a szállítási díjjat. A limit és a határérték a global táblában tárolt adat. Jelen állás szerint 25000 forint összegű vásárlás felett díjtalan a kiszállítás, de az érték alatt 2500ft szállítási díj.

Paraméterek

Name	Data Type	Max Length (Bytes)
@TotalDue A rendelés összege, amiből számítható a szállítási díj.	money	8

```
CREATE FUNCTION dbo.ShippingFee (@TotalDue money) RETURNS money AS

BEGIN

DECLARE @limit int

DECLARE @BelowLimitPrice money

DECLARE @AboveLimitPrice money

SET @limit = (Select try_cast(Value as money) from dbo.Global Where Name = 'Limit')

SET @BelowLimitPrice = (Select try_cast(Value as money) from dbo.Global Where Name = 'Below-LimitPrice') -- szállítási díj értéke a global táblából

SET @AboveLimitPrice = 0

RETURN

(SELECT CASE WHEN @TotalDue < @limit THEN @BelowLimitPrice ELSE @AboveLimitPrice END )

END
```



Adatbázis Triggerek

Objektumok

Name

tDLL_LOG

A DDL utasátások adatait naplózza a DatabaseLog táblába. A TSQL mezőbe beírja magát az utasítást is.



∮ tDLL LOG

Leírás

A DDL utasátások adatait naplózza a DatabaseLog táblába. A TSQL mezőbe beírja magát az utasítást is.

```
CREATE TRIGGER tDLL_LOG ON DATABASE

FOR DDL_DATABASE_LEVEL_EVENTS

AS

BEGIN

DECLARE @schema sysname;

DECLARE @object sysname;

DECLARE @data XML;

DECLARE @eventType sysname;

SET @data = EVENTDATA();

SET @eventType = @data.value('(/EVENT_INSTANCE/EventType)1', 'sysname')

SET @schema = @data.value('(/EVENT_INSTANCE/SchemaName)1', 'sysname')

SET @object = @data.value('(/EVENT_INSTANCE/ObjectName)1', 'sysname')

INSERT INTO DatabaseLog (Event, ModifyingUser, EventTime, Schema, Object, TSQL)

VALUES (@eventType, USER_NAME(), GETDATE(), @schema, @object, @data.value('(/EVENT_INSTANCE/TSQLCommand)1', 'varchar(500)'))

END
```





Típus táblák

Objektumok

Name

dbo.ProductIdAndQuantity

Az új rendelés fölvételét kivitelező tárolt eljárás használja ezt a típus táblát ami a productId és az ahoz tartozó darabszám értékpárokat tárolja.

dbo.Property

Új termék fölvételét kivitelező tárolt eljárás használja ezt a típus táblát ami a termékhez tartozó tuljadonságok halmazát tartalmazza.



dbo.ProductIdAndQuantity

Leírás

Az új rendelés fölvételét kivitelező tárolt eljárás használja ezt a típus táblát, ami a productld és az ahoz tartozó darabszám értékpárokat tárolja.

Oszlopok

Name	Data Type	Max Length (Bytes)	Nullability
ProductId A termék azonosítója.	int	4	NULL allowed
Quantity A rendelt termék darabszáma.	int	4	NULL allowed

```
CREATE TYPE dbo.ProductIdAndQuantity AS TABLE

(
ProductId int NULL,
Quantity int NULL
)
```



dbo.Property

Leírás

Új termék fölvételét kivitelező tárolt eljárás használja ezt a típus táblát, ami a termékhez tartozó tuljadonságok halmazát tartalmazza.

Oszlopok

Name	Data Type	Max Length (Bytes)	Nullability
PropertyTypeld A tulajdonság azonosítója.	int	4	NULL allowed
PropertyValueid A fix tulajdonság érték azonosítója.	int	4	NULL allowed
PropertyValue Az egyeni tulajdonság érték.	varchar(50)	50	NULL allowed

```
CREATE TYPE dbo.Property AS TABLE

(
PropertyTypeId int NULL,
PropertyValueid int NULL,
PropertyValue varchar (50) COLLATE Hungarian_CI_AS NULL
)
```



♣ Felhasználók

Objektumok

Name

EGS_Admin

Az adatbázis adminisztrátorai. Teljeskörű hozzáfáráse van az adatbázishoz. A db_owner adatbázis szerepkör tagja.

EGS_DataAnalyst

Adatbázis elemező. Tagja a DataAnalystRole adatbázis szerepkörnek.

EGS EMP

Az webshop alkalmazottaja. Az Emp_role adatbázis szerepkör tagja.

EGS User

A felhasználók álltal használt kliens program. Jogosultság kezelése a User_role adatbázis szerepkörben, aminek a tagja.



▲ EGS_Admin

Leírás

Az adatbázis adminisztrátorai. Teljeskörű hozzáfáráse van az adatbázishoz. A db_owner adatbázis szerepkör tagja.

Tulajdonságok

Property	Value
Туре	SqlUser
Login Name	EGS_Admin
Default Schema	dbo

Adatbázis szintű jogosultságok

Туре	Action
CONNECT	Grant

SQL Script

CREATE USER EGS_Admin FOR LOGIN EGS_Admin



♣ EGS_DataAnalyst

Leírás

Adat elemező felhasználók. Tagja a DataAnalystRole adatbázis szerepkörnek.

Tulajdonságok

Property	Value
Туре	SqlUser
Login Name	EGS_DataAnalyst
Default Schema	dbo

Adatbázis szintű jogosultságok

Туре	Action
CONNECT	Grant

SQL Script

CREATE USER EGS_DataAnalyst FOR LOGIN EGS_DataAnalyst





Az webshop alkalmazottaja. Az Emp_role adatbázis szerepkör tagja.

Tulajdonságok

Property	Value
Туре	SqlUser
Login Name	EGS_EMP
Default Schema	dbo

Adatbázis szintű jogosultságok

Туре	Action
CONNECT	Grant

SQL Script

CREATE USER EGS_EMP FOR LOGIN EGS_EMP



♣ EGS_User

Leírás

A felhasználók álltal használt kliens program. Jogosultság kezelése a User_role adatbázis szerepkörben, aminek a tagja.

Tulajdonságok

Property	Value	
Туре	SqlUser	
Login Name	EGS_User	
Default Schema	dbo	

Adatbázis szintű jogosultságok

Туре	Action
CONNECT	Grant

SQL Script

CREATE USER EGS_User FOR LOGIN EGS_User



Adatbázis szerepkörök

Objektumok

Name

DataAnalystRole

Select joga van a vQuarterlyStatement, vPreviousMonthAmount, vCountOfProduct view-khoz és a fRepeat-Customer függvényhez. Végrehajtási joga van MonthlyStatement tárolt eljáráshoz.

db_owner

Fix adatbázis szerepkör.

Emp_Role

A webshop alkalmazotaihoz tartozó szerepkör.

User_Role

A felhasználók álltal használt kliens program szerepköre.



♣ DataAnalystRole

Leírás

Select joga van a vQuarterlyStatement, vPreviousMonthAmount, vCountOfProduct view-khoz és a f-RepeatCustomer függvényhez. Végrehajtási joga van MonthlyStatement tárolt eljáráshoz.

Tulajdonságok

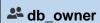
Property	Value
Owner	dbo

Tagok

EGS_DataAnalyst

```
CREATE ROLE DataAnalystRole
AUTHORIZATION dbo
GO
ALTER ROLE DataAnalystRole ADD MEMBER EGS_DataAnalyst
GO
```





Fix adatbázis szerepkör.

Tulajdonságok

Property	Value
Owner	dbo

Tagok

• EGS_Admin

SQL Script

ALTER ROLE db_owner ADD MEMBER EGS_Admin GO



♣ Emp_Role

Leírás

A webshop alkalmazotaihoz tartozó szerepkör.

Tulajdonságok

Property	Value
Owner	dbo

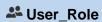
Tagok

• EGS_EMP

SQL Script

CREATE ROLE Emp_Role
AUTHORIZATION dbo
GO
ALTER ROLE Emp_Role ADD MEMBER EGS_EMP





A felhasználók álltal használt kliens program szerepköre.

Tulajdonságok

Property	Value
Owner	dbo

Tagok

• EGS_User

```
CREATE ROLE User_Role
AUTHORIZATION dbo
GO
ALTER ROLE User_Role ADD MEMBER EGS_User
```



Alkalmazás szerepkörök

Name

WebApp

A webshophoz tartozó webalkalmazások szerepköre.





A webshophoz tartozó webalkalmazások szerepköre.

Tulajdonságok

Property	Value
Default Schema	dbo

SQL Script

CREATE APPLICATION ROLE WebApp WITH PASSWORD = 'p@ssw0rd', DEFAULT_SCHEMA = dbo



Mentési stratégia

Adatbázis információk

- Az ElectricGuitarShop adatbázis egy webshop üzlet adatait tartalmazza, beleértve az ügyfelek, termékek, rendelések adatait.
- Az adatbázis mérete jelenleg 216.00 MB.
- Az adatbázis a Microsoft SQL Server Standard verzióján fut és full recovery modellt használ, amely lehetővé teszi a tranzakció-log fájlok mentését és a visszaállítás pontossab elvégzését.

Mentési stratégia

- Az ElectricGuitarShop adatbázis heti egyszer, szombat éjjel 2-kor teljes mentést, valamint naponta éjjel fél 2-kor differenciális mentést végez.
- Az óránkénti tranzakció-log mentést használjuk, hogy biztosítsuk a redundanciát és az adatvesztés elkerülését.
- A mentési folyamatokat az SQL Agent segítségével automatizáljuk, amely lehetővé teszi a jobok ütemezését és konfigurálását a SQL Server Management Studio (SSMS) segítségével.
- A mentési adatokat az adatbázis szerveren tároljuk, azonban a mentési adatokat másik helyre is át lehet menteni.

A mentési folyamat konfigurációja

- A mentési típusok a következők: teljes mentés, differenciális mentés és tranzakció-log mentés
- A mentési feladatokat jobok segítségével végezzük.
- Az ElectricGuitarShop adatbázis teljes mentése után az előző teljes mentés óta változott adatokat tartalmazó differenciális mentéseket készítünk.
- Az óránkénti tranzakció-log mentések az előző teljes vagy differenciális mentés óta végrehajtott tranzakciók mentésére szolgálnak.
- Az ElectricGuitarShop adatbázis mentési fájljait a' C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\' mappában tároljuk.
- Az adatbázis mentéseit a jobok segítségével automatizáljuk.