VISOKA ŠKOLA STRUKOVNIH STUDIJA ZA INFORMACIONE I KOMUNIKACIONE TEHNOLOGIJE

DOKUMENTACIJA

Praktikum iz baze podataka

Mateja Mastelica 73/20

Smer: Internet tehnologije

Modul: Web programiranje

Sadrzaj

Dizajn baze podataka

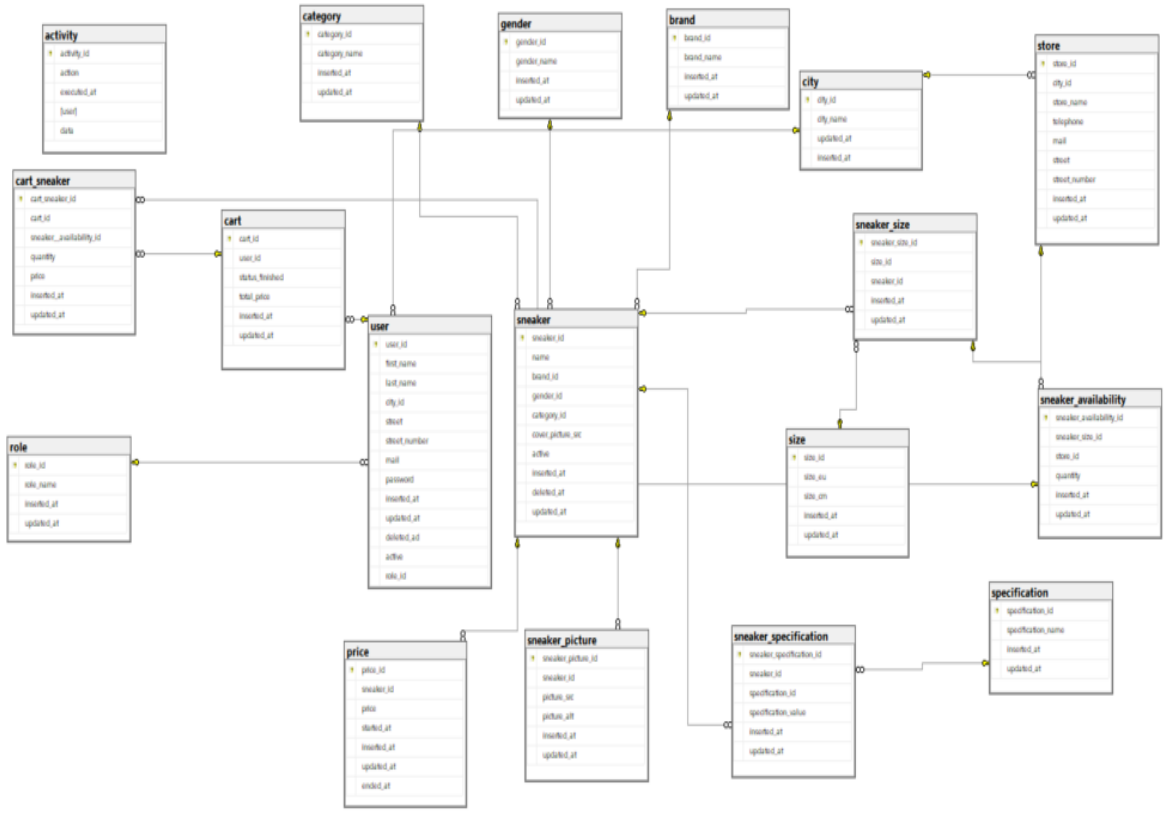
Kratak opis

Baza podataka u ovom projektu predstavlja lanac prodavnica patika u Srbiji,u kojoj kupovinu mogu da vrse iskljucivo registrovani kupci.

Realizovana je kroz 18 tabela, 13 pogleda, 8 uskladištenih procedura, 3 skalarne i 2 tabelarne funckije.

U bazi je kreiran korisnik (programer,Programer123 ali na mom laptopu javlja grešku prilikom logovanja i onda nisam proverio sve) kojem je zabranjen pristup svim tabelama, a pogledima, funckijama i uskladištenim procedurama korisnik može da pristupa samo preko SELECT.

Dijagram


activity
category
gender
brand
store
city
cart_sneaker
cart
sneaker_size
user
sneaker
role
sneaker_availability
size
price
sneaker_picture
sneaker_specification
specification

Tabele

Tabela brand

Opis:

Tabela brand predstavlja brendove(lookup tabela) koji su dostupni u sklopu prodavnica,povezana je sa tabelom sneaker,za nju postoji uskladistena procedura add_new_brand koji pri unosu novog brenda proverava da li takav vec psotoji u bazi

ucestvuje u vecini pogleda koji su vezani za modele patika,ali o tome vise u nastavku

Kolone:

- brand_id int IDENTITY(1,1) PRIMARY KEY
- brand_name NVARCHAR(255) NOT NULL
- inserted_at DATETIME
- updated_at DATETIME

Tabela category

Opis:

Tabela brand predstavlja kateborije(lookup tabela) koji su dostupni u sklopu prodavnica,povezana je sa tabelom sneaker,

ucestvuje u vecini pogleda koji su vezani za modele patika

Kolone:

- category_id int IDENTITY(1,1) PRIMARY KEY
- category_name NVARCHAR(255) NOT NULL
- inserted_at DATETIME
- updated_at DATETIME

Tabela gender

Opis:

Tabela gender predstavlja polove(lookup tabela) koji su dostupni u sklopu prodavnica,povezana je sa tabelom sneaker

ucestvuje u vecini pogleda koji su vezani za modele patika,ali o tome vise u nastavku

Kolone:

- gender_id int IDENTITY(1,1) PRIMARY KEY
- gender_name NVARCHAR(255) NOT NULL
- inserted_at DATETIME
- updated_at DATETIME

Tabela city

Opis:

Tabela city predstavlja gradove(lookup tabela) koji su dostupni u sklopu drzave(nisu upisani svi,nego samo 10ak,u kojima BUZZ ima svoje prodavnice)

povezana je sa tabelama store(prodavnice u kojim gradovima se nalaze) i user(registrovani kupci iz kog su grada,zbog kasnijeg dostavljanja patika)

Kolone:

- city_id int IDENTITY(1,1) PRIMARY KEY
- city_name NVARCHAR(255) NOT NULL
- inserted_at DATETIME
- updated_at DATETIME

Tabela role

Opis:

Tabela role predstavlja uloge(lookup tabela) koje su dostupne u okviru aplikacije,ucestvuje u pogledu koji prikazuje sve korisnike aplikacije/sajta/shopa

Kolone:

- role_id int IDENTITY(1,1) PRIMARY KEY
- role_name NVARCHAR(255) NOT NULL
- inserted_at DATETIME
- updated_at DATETIME

Tabela size

Opis:

Tabela size predstavlja velicinu patike od broja 36-47,sa EU,US,UK standardom i duzinom gazista u cm,spaja se sa tabelom sneaker,tacnije sneaker_size,u relaciji VISE:VISE

Kolone:

- size_id int IDENTITY(1,1) PRIMARY KEY
- size_eu int NOT NULL
- size_us int NOT NULL
- size_uk int NOT NULL
- size_cm NVARCAHR(20) NOT NULL
- inserted_at DATETIME
- updated_at DATETIME

Tabela sneaker

Opis:

Glavna tabela u bazi,u kojoj postoji ime modela,i cover slika,spaja se sa tabelama brand,gender,category,price,sneaker_picture,sneaker_size,sneaker_specification

Kolone:

- sneaker_id int IDENTITY(1,1) PRIMARY KEY
- sneaker_name NVARCHAR(255) NOT NULL
- brand_id INT NOT NULL
- category_id INT NOT NULL
- gender_id INT NOT NULL
- cover_picture_src NVARCHAR(255) NOT NULL
- active bit
- inserted_at DATETIME
- updated_at DATETIME

Tabela store

Opis:

Takodje jako bitna tabela u bazi,koja predtavlja sve prodavnice u Srbiji,vezana je za city,sneaker_availability

Kolone:

- store_id int IDENTITY(1,1) PRIMARY KEY
- store_name NVARCHAR(255) NOT NULL
- city_id INT NOT NULL
- telephone NVARCHAR(20) NOT NULL
- mail NVARCHAR(20) NOT NULL
- street NVARCHAR(255) NOT NULL
- street_number NVARCHAR(255) NOT NULL
- inserted_at DATETIME
- updated_at DATETIME

Tabela sneaker_size

Opis:

Vezivna tabela u kojoj se nalaze modeli i njihovi brojevi/velicine,spojena sa sneaker i size,ucestvuje u pogledima koji su vezani za modele dostupne u radnjama

Kolone:

- sneaker_size_id int IDENTITY(1,1) PRIMARY KEY
- sneaker_id INT NOT NULL
- size_id INT NOT NULL
- inserted_at DATETIME
- updated_at DATETIME

Tabela sneaker_picture

Opis:

Tabela koja cuva slike(sporedne ili kako vec da ih nazovem,one koje slike koje nisu naslovna)vezane za patike,spojena sa sneaker

Kolone:

- sneaker_picture_id int IDENTITY(1,1) PRIMARY KEY
- sneaker_id INT NOT NULL
- picture_Src NVARCHAR(255) NOT NULL
- picture_alt NVARCHAR(255) NOT NULL
- inserted_at DATETIME
- updated_at DATETIME

Tabela specification

Opis:

Tabela specifiakcije u kojoj se nalaze nazivi specifikacija modela spaja se sa sneaker_specification

Kolone:

- specification_id int IDENTITY(1,1) PRIMARY KEY
- specification_name NVARHCAR(255) NOT NULL
- inserted_at DATETIME
- updated_at DATETIME

Tabela sneaker_specification

Opis:

Vezivna tabela u kojoj su vrednosti specifikacija modela patika spaja se sa sneaker,specifaication

Kolone:

- sneaker_specification_id int IDENTITY(1,1) PRIMARY KEY
- specification_id INT NOT NULL
- sneaker_id INT NOT NULL
- inserted_at DATETIME
- updated_at DATETIME

Tabela user

Opis:

Tabela sa svim korisnicima sajta,spaja se sa role,cart

Kolone:

- user_id int IDENTITY(1,1) PRIMARY KEY
- first_name NVARHCAR(255) NOT NULL
- last_name NVARHCAR(255) NOT NULL
- city_id INT NOT NULL
- password NVARCHAR(20) NOT NULL
- mail NVARCHAR(20) NOT NULL
- street NVARCHAR(255) NOT NULL
- street_number NVARCHAR(255) NOT NULL
- role_id INT NOT NULL
- active bit
- inserted_at DATETIME
- updated_at DATETIME

Tabela cart

Opis:

Tabela cart sa zavrsenim kupovinama korisnika,spaja se sa user,cart_sneaker,polje total_price je trebalo da bude okidac,koji racuna ukupnu cenu cele kupovine,ali je u ovom slucaju

hard kodovano

Kolone:

- cart_id int IDENTITY(1,1) PRIMARY KEY
- user_id int NOT NULL
- status_finished bit
- total_price decimal(16,2)
- inserted_at DATETIME
- updated_at DATETIME

Tabela cart_sneaker

Opis:

Tabela cart_sneaker je tabela sa pojedinacnim artiklima sa jedne kupovine,kolicina i model koji je kupljen,kao i njena cena ova tabela je isto trebala da ima okidac koji na osnovu

kolicine oduzima kolicinu iz radnje iz koje je poruceno i okidac koji na osnovu kolicine i proizvoda racuna cenu za taj proizvod,ali je i ovde hardkokovani,spaja se sa cart,sneaker_availability

Kolone:

- cart_sneaker_id int IDENTITY(1,1) PRIMARY KEY
- cart_id int NOT NULL
- sneaker_availability_id INT NOT NULL
- price decimal(16,2)
- quanitity int
- inserted_at DATETIME
- updated_at DATETIME

Tabela price

Opis:

Tabela price je tabele sa svim cenama koje su proizvodi imali sa evidencijom od kad je aktuelna cena,pa do kad se zavrsava,spaja se sa sneaker i ucestvuje u nekim pogledima

Kolone:

- price_id int IDENTITY(1,1) PRIMARY KEY
- sneaker_id int NOT NULL
- started_at DATETIME NOT NULL
- ended_at DATETIME NOT NULL
- price decimal(16,2)
- quanitity int
- inserted_at DATETIME
- updated_at DATETIME

Tabela sneaker_availability

Opis:

Tabela sneaker_availability je vezivna tabela koja sadrzi evidenciju,modela sa dostupnim brojevima i kolicinom,po svim psotojecim prodavnicama,spaja se sa store,sneaker_size,cart_sneaker

Kolone:

- sneaker_availability_id int IDENTITY(1,1) PRIMARY KEY
- sneaker_size_id int NOT NULL
- store_id int NOT NULL
- quanitity int
- inserted_at DATETIME
- updated_at DATETIME

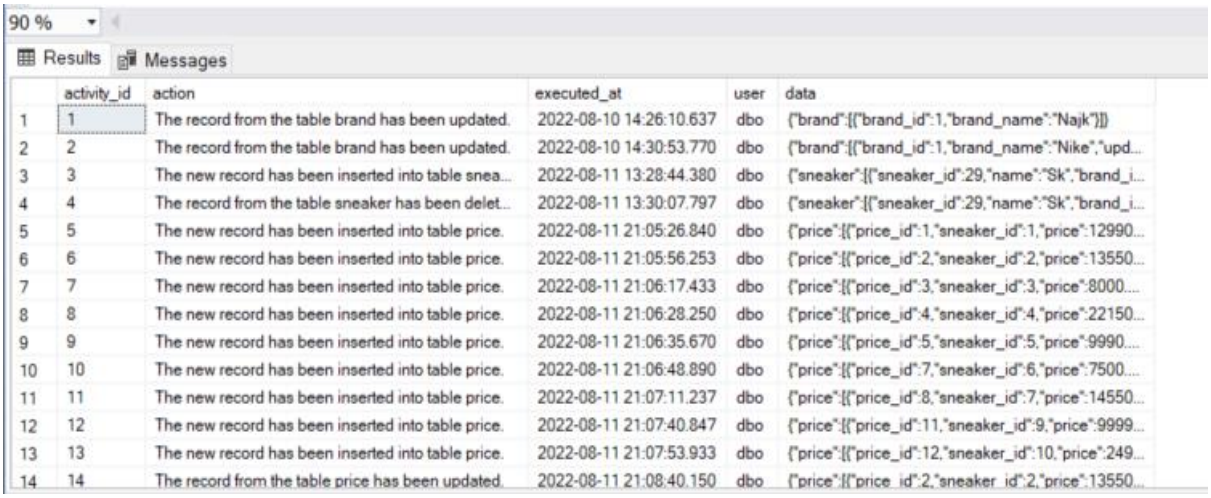
Tabela activity

Opis:

Ova tabela nije povezana sa drugim tabelama I u njoj se vrši upis aktivnost korisnika, sa komentarom I datumom izvršavanja.

Kolone:

- activitiy_id bigint IDENTITY(1,1) PRIMARY KEY
- [user] NVARCHAR(255) NOT NULL
- action nvarchar(255)
- data NVARCHAR(MAX)
- excuted_at datetime



90 %

Results | Messages

	activity_id	action	executed_at	user	data
1	1	The record from the table brand has been updated.	2022-08-10 14:26:10.637	dbo	{"brand":[{"brand_id":1,"brand_name":"Najk"}]}
2	2	The record from the table brand has been updated.	2022-08-10 14:30:53.770	dbo	{"brand":[{"brand_id":1,"brand_name":"Nike","upd...
3	3	The new record has been inserted into table snea...	2022-08-11 13:28:44.380	dbo	{"sneaker":[{"sneaker_id":29,"name":"Sk","brand_i...
4	4	The record from the table sneaker has been delet...	2022-08-11 13:30:07.797	dbo	{"sneaker":[{"sneaker_id":29,"name":"Sk","brand_i...
5	5	The new record has been inserted into table price.	2022-08-11 21:05:26.840	dbo	{"price":[{"price_id":1,"sneaker_id":1,"price":12990...
6	6	The new record has been inserted into table price.	2022-08-11 21:05:56.253	dbo	{"price":[{"price_id":2,"sneaker_id":2,"price":13550...
7	7	The new record has been inserted into table price.	2022-08-11 21:06:17.433	dbo	{"price":[{"price_id":3,"sneaker_id":3,"price":8000....
8	8	The new record has been inserted into table price.	2022-08-11 21:06:28.250	dbo	{"price":[{"price_id":4,"sneaker_id":4,"price":22150...
9	9	The new record has been inserted into table price.	2022-08-11 21:06:35.670	dbo	{"price":[{"price_id":5,"sneaker_id":5,"price":9990....
10	10	The new record has been inserted into table price.	2022-08-11 21:06:48.890	dbo	{"price":[{"price_id":7,"sneaker_id":6,"price":7500....
11	11	The new record has been inserted into table price.	2022-08-11 21:07:11.237	dbo	{"price":[{"price_id":8,"sneaker_id":7,"price":14550...
12	12	The new record has been inserted into table price.	2022-08-11 21:07:40.847	dbo	{"price":[{"price_id":11,"sneaker_id":9,"price":9999...
13	13	The new record has been inserted into table price.	2022-08-11 21:07:53.933	dbo	{"price":[{"price_id":12,"sneaker_id":10,"price":249...
14	14	The record from the table price has been updated.	2022-08-11 21:08:40.150	dbo	{"price":[{"price_id":2,"sneaker_id":2,"price":13550...

Pogledi

view_all_avalibility_models_in_nis

Opis:

Pogled koji predstavlja sve aktivne modele koji se nalaze u prodavnici u Nisu, ucestvuju tabele: brand, gender, sneaker_size, store, city, size, sneaker_availability, category.

SELECT dbo.sneaker.name, dbo.gender.gender_name, dbo.brand.brand_name, dbo.city.city_name, dbo.store.store_name, dbo.sneaker_availability.quantity, dbo.size.size_eu, dbo.size.size_cm

FROM dbo.city INNER JOIN

dbo.store ON dbo.city.city_id = dbo.store.city_id INNER JOIN

dbo.sneaker_availability ON dbo.store.store_id = dbo.sneaker_availability.store_id INNER JOIN

dbo.brand INNER JOIN

dbo.sneaker ON dbo.brand.brand_id = dbo.sneaker.brand_id INNER JOIN

dbo.gender ON dbo.sneaker.gender_id = dbo.gender.gender_id INNER JOIN

dbo.sneaker_size ON dbo.sneaker.sneaker_id = dbo.sneaker_size.sneaker_id ON dbo.sneaker_availability.sneaker_size_id = dbo.sneaker_size.sneaker_size_id INNER JOIN

dbo.size ON dbo.sneaker_size.size_id = dbo.size.size_id

WHERE (dbo.city.city_name = N'Niš')

	name	gender_name	brand_name	city_name	store_name	quantity	size_eu	size_cm
1	Jordan 6 Rings	Muški	Nike	Niš	PJ Buzz Niš	34	44	27,1cm
2	Air More Uptempo	Muški	Nike	Niš	PJ Delta Niš	136	44	27,1cm
3	Air More Uptempo	Muški	Nike	Niš	PJ Buzz Niš	28	44	27,1cm
4	Air Max 270	Muški	Nike	Niš	PJ Buzz Niš	144	43	26,7cm
5	AIR JORDAN 1 MID	Ženski	Nike	Niš	PJ Delta Niš	122	38	23,3cm
6	AIR JORDAN 1 MID	Ženski	Nike	Niš	PJ Buzz Niš	90	39	24,2cm
7	Supersat BOLD	Ženski	Adidas	Niš	PJ Delta Niš	46	39	24,2cm
8	Adidas Astir	Ženski	Adidas	Niš	PJ Buzz Niš	145	37	22,9cm
9	Continental 80	Muški	Adidas	Niš	PJ Buzz Niš	77	46	28,4cm
10	RS-Z Reivent	Ženski	Puma	Niš	PJ Buzz Niš	22	37	22,9cm
11	Chuck 70 Explore Waterproof	Muški	All Star Convers	Niš	PJ Buzz Niš	33	43	26,7cm

view_all_models_higher_than_40_size

Opis:

Pogled koji predstavlja sve aktivne modele koji su dostupni u velicini vecoj od broja 40, ucestvuju tabele: brand, gender, sneaker, sneaker_size, store, city, size, sneaker_availability, category.

SELECT dbo.sneaker.cover_picture_src, dbo.sneaker.name, dbo.sneaker_availability.quantity, dbo.size.size_eu, dbo.store.store_name, dbo.city.city_name, dbo.brand.brand_name

FROM dbo.brand INNER JOIN

dbo.sneaker ON dbo.brand.brand_id = dbo.sneaker.brand_id INNER JOIN

dbo.sneaker_size INNER JOIN

dbo.size ON dbo.sneaker_size.size_id = dbo.size.size_id INNER JOIN

dbo.sneaker_availability ON dbo.sneaker_size.sneaker_size_id = dbo.sneaker_availability.sneaker_size_id INNER JOIN

dbo.store ON dbo.sneaker_availability.store_id = dbo.store.store_id INNER JOIN

dbo.city ON dbo.store.city_id = dbo.city.city_id ON dbo.sneaker.sneaker_id = dbo.sneaker_size.sneaker_id

WHERE (dbo.size.size_eu > 40) AND (dbo.sneaker_availability.quantity > 0)

	cover_picture_src	name	quantity	size_eu	store_name	city_name	brand_name
1	assets/cover1.jpg	Jordan 6 Rings	245	41	TC BIG Shopping Centar	Beograd	Nike
2	assets/cover1.jpg	Jordan 6 Rings	123	41	PJ Buzz Ada Mall	Beograd	Nike
3	assets/cover1.jpg	Jordan 6 Rings	78	41	PJ Buzz BW	Beograd	Nike
4	assets/cover1.jpg	Jordan 6 Rings	99	41	PJ Buzz Delta City	Beograd	Nike
5	assets/cover1.jpg	Jordan 6 Rings	124	42	PJ Buzz Ušće	Beograd	Nike
6	assets/cover1.jpg	Jordan 6 Rings	14	42	PJ Buzz Kragujevac	Kragujevac	Nike
7	assets/cover1.jpg	Jordan 6 Rings	53	42	PJ Outlet Kraljevo	Kraljevo	Nike
8	assets/cover1.jpg	Jordan 6 Rings	96	42	PJ Outlet Zrenjanin	Zrenjanin	Nike
9	assets/cover1.jpg	Jordan 6 Rings	111	43	PJ Buzz Sombor	Sombor	Nike
10	assets/cover1.jpg	Jordan 6 Rings	34	44	PJ Buzz Niš	Niš	Nike
11	assets/cover1.jpg	Jordan 6 Rings	25	44	PJ Buzz Novi Sad	Novi Sad	Nike
12	assets/cover1.jpg	Jordan 6 Rings	4	45	PJ Buzz Promenada	Novi Sad	Nike
13	assets/cover1.jpg	Jordan 6 Rings	46	45	PJ Buzz Sombor	Sombor	Nike
14	assets/cover1.jpg	Air More Uptempo	57	43	TC BIG Shopping Centar	Beograd	Nike

view_all_models_with_price

Opis:

Pogled koji predstavlja sve aktivne modele sa cenom, ucestvuju tabele: brand, gender, sneaker, category, price.

SELECT dbo.sneaker.name, dbo.brand.brand_name, dbo.category.category_name, dbo.gender.gender_name, dbo.price.price

FROM dbo.sneaker INNER JOIN

dbo.brand ON dbo.sneaker.brand_id = dbo.brand.brand_id INNER JOIN

dbo.category ON dbo.sneaker.category_id = dbo.category.category_id INNER JOIN

dbo.gender ON dbo.sneaker.gender_id = dbo.gender.gender_id INNER JOIN

dbo.price ON dbo.sneaker.sneaker_id = dbo.price.sneaker_id

WHERE (dbo.price.ended_at IS NULL) OR

(dbo.price.ended_at > GETDATE())

	name	brand_name	category_name	gender_name	price
1	Jordan 6 Rings	Nike	Duboke	Muški	12990.00
2	Air More Uptempo	Nike	Duboke	Muški	13550.00
3	Air Force 1	Nike	Plitke	Muški	8000.00
4	Air Max 90	Nike	Plitke	Muški	22150.00
5	Air Max 270	Nike	Plitke	Muški	9990.00
6	AIR JORDAN 1 MID	Nike	Duboke	Ženski	7500.00
7	HUARACHE SB	Nike	Sport	Ženski	14550.99
8	Air Max Structure	Nike	Plitke	Ženski	9999.99
9	AF1 PIXEL	Nike	Plitke	Ženski	24999.50
10	Supersat BOLD	Adidas	Plitke	Ženski	9800.00
11	Adidas Astir	Adidas	Plitke	Ženski	6790.50
12	Superstar	Adidas	Plitke	Muški	17990.00
13	Continental 80	Adidas	Plitke	Muški	19999.99
14	Gazelle	Adidas	Plitke	Muški	23450.00

view_nike_all_male_models

Opis:

Pogled koji predstavlja sve aktivne modele muske brenda najk, ucestvuju tabele: brand, gender, sneaker, category.

SELECT dbo.sneaker.name, dbo.brand.brand_name, dbo.category.category_name, dbo.gender.gender_name

FROM dbo.brand INNER JOIN

dbo.sneaker ON dbo.brand.brand_id = dbo.sneaker.brand_id INNER JOIN

dbo.category ON dbo.sneaker.category_id = dbo.category.category_id INNER JOIN

dbo.gender ON dbo.sneaker.gender_id = dbo.gender.gender_id

WHERE (dbo.brand.brand_name = N'Nike') AND (dbo.gender.gender_name = N'Muški') AND (dbo.sneaker.active = 1)

	name	brand_name	category_name	gender_name
1	Jordan 6 Rings	Nike	Duboke	Muški
2	Air More Uptempo	Nike	Duboke	Muški
3	Air Force 1	Nike	Plitke	Muški
4	Air Max 90	Nike	Plitke	Muški
5	Air Max 270	Nike	Plitke	Muški
6	Air Ljakse	Nike	Plitke	Muški

view_picture_of_puma_rsx

Opis:

Pogled koji predstavlja sve slike puminog modela rsx, ucestvuju tabele: brand, sneaker_picture, sneaker, category.

SELECT dbo.sneaker.name, dbo.sneaker_picture.picture_src, dbo.category.category_name, dbo.brand.brand_name

FROM dbo.sneaker INNER JOIN

dbo.sneaker_picture ON dbo.sneaker.sneaker_id = dbo.sneaker_picture.sneaker_id INNER JOIN

dbo.brand ON dbo.sneaker.brand_id = dbo.brand.brand_id INNER JOIN

dbo.category ON dbo.sneaker.category_id = dbo.category.category_id

WHERE (dbo.sneaker.active = 1) AND (dbo.sneaker.name = N'RS-Z Reivent')

Results Messages

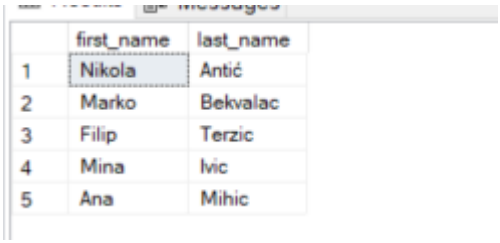
name	picture_src	category_name	brand_name
RS-Z Reivent	assets/puma1.jpg	Sport	Puma
RS-Z Reivent	assets/puma2.jpg	Sport	Puma
RS-Z Reivent	assets/puma3.jpg	Sport	Puma
RS-Z Reivent	assets/puma5.jpg	Sport	Puma
RS-Z Reivent	assets/puma4.jpg	Sport	Puma

view_all_registration_user

Opis:

Pogled koji predstavlja sve registrovane korisnike, ucestvuju tabele: role, user.

```
SELECT dbo.[user].first_name, dbo.[user].last_name
FROM     dbo.[user] INNER JOIN
                  dbo.role ON dbo.[user].role_id = dbo.role.role_id
WHERE  (dbo.role.role_name = N'Korisnik') AND (dbo.[user].active = 1)
```



	first_name	last_name
1	Nikola	Antić
2	Marko	Bekvalac
3	Filip	Terzic
4	Mina	Ivic
5	Ana	Mihic

view_all_sneakers_model

Opis:

Pogled koji predstavlja sve dostupne modele koji postoje, samo njihov brend, kategoriju I za koji su pol, ucestvuju tabele: brand, sneaker, category, gender,

```
SELECT dbo.sneaker.name, dbo.gender.gender_name, dbo.category.category_name, dbo.brand.brand_name
FROM dbo.brand INNER JOIN
    dbo.sneaker ON dbo.brand.brand_id = dbo.sneaker.brand_id INNER JOIN
    dbo.category ON dbo.sneaker.category_id = dbo.category.category_id INNER JOIN
    dbo.gender ON dbo.sneaker.gender_id = dbo.gender.gender_id
WHERE (dbo.sneaker.active = 1)
```

Results Messages

	name	gender_name	category_name	brand_name
1	Jordan 6 Rings	Muški	Duboke	Nike
2	Air More Uptempo	Muški	Duboke	Nike
3	Air Force 1	Muški	Plitke	Nike
4	Air Max 90	Muški	Plitke	Nike
5	Air Max 270	Muški	Plitke	Nike
6	AIR JORDAN 1 MID	Ženski	Duboke	Nike
7	HUARACHE SB	Ženski	Sport	Nike
8	Air Max Structure	Ženski	Plitke	Nike
9	AF1 PIXEL	Ženski	Plitke	Nike
10	Supersat BOLD	Ženski	Plitke	Adidas
11	Adidas Astir	Ženski	Plitke	Adidas
12	Superstar	Muški	Plitke	Adidas
13	Continental 80	Muški	Plitke	Adidas
14	Gazelle	Muški	Plitke	Adidas

view_all_store_in_belgrade

Opis:

Pogled koji predstavlja sve prodavnice u Beogradu, ucestvuju tabele: store, city

```
SELECT dbo.store.store_name, dbo.store.street, dbo.store.street_number, dbo.city.city_name
FROM   dbo.store INNER JOIN
                  dbo.city ON dbo.store.city_id = dbo.city.city_id
WHERE  (dbo.city.city_name = N'Beograd')
```

Results Messages

	store_name	street	street_number	city_name
1	TC BIG Shopping Centar	Višnjička	84	Beograd
2	PJ Outlet Banjica	Crnotravska	4	Beograd
3	PJ Buzz Ada Mall	Vojislava Ilića	141	Beograd
4	PJ Buzz BW	Bulevar Vudroa Vilsona	12	Beograd
5	PJ Buzz Delta City	Jurija Gagarina	133	Beograd
6	PJ Buzz Ušće	Bulevar Mihajla Pupina	4	Beograd

view_gazzele_model_with_all_specifications

Opis:

Pogled koji predstavlja adidasov model Gazelle sa svim njenim specifikacijama, ucestvuju tabele: sneaker, specification, sneaker_specification, brand.

pogled koji predstavlja pumin model Gazelle sa svim njenim specifikacijama

ucestvuju tabele sneaker,specification,sneaker_specification,brand

```
SELECT dbo.sneaker.name, dbo.brand.brand_name, dbo.specification.specification_name, dbo.sneaker_specification.specification_value
FROM   dbo.sneaker INNER JOIN
           dbo.sneaker_specification ON dbo.sneaker.sneaker_id = dbo.sneaker_specification.sneaker_id INNER JOIN
           dbo.specification ON dbo.sneaker_specification.specification_id = dbo.specification.specification_id INNER JOIN
           dbo.brand ON dbo.sneaker.brand_id = dbo.brand.brand_id
WHERE  (dbo.sneaker.active = 1) AND (dbo.sneaker.name = N'Gazelle')
```

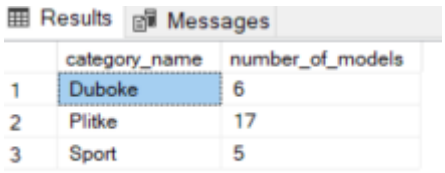
	name	brand_name	specification_name	specification_value
1	Gazelle	Adidas	Boja	Crvena
2	Gazelle	Adidas	Materijal	Prevrnuta koža
3	Gazelle	Adidas	Uložak	Anatomski
4	Gazelle	Adidas	Djon	Guma

view_number_of_product_per_category

Opis:

Pogled koji predstavlja broj modela u radnjama po kategoriji, ucestvuju tabele: sneaker, category.

```
SELECT TOP (100) PERCENT dbo.category.category_name, COUNT(dbo.sneaker.category_id) AS number_of_models
FROM dbo.category INNER JOIN
    dbo.sneaker ON dbo.category.category_id = dbo.sneaker.category_id
GROUP BY dbo.category.category_name
```



Results | Messages

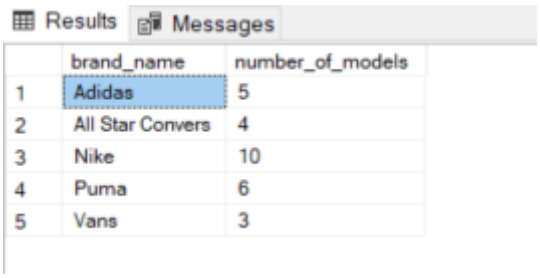
	category_name	number_of_models
1	Duboke	6
2	Plitke	17
3	Sport	5

view_number_of_product_per_brand

Opis:

Pogled koji predstavlja broj modela u radnjama po brendu, ucestvuju tabele: sneaker, brand.

```
SELECT dbo.brand.brand_name, COUNT(dbo.sneaker.brand_id) AS number_of_models
FROM dbo.brand INNER JOIN
        dbo.sneaker ON dbo.brand.brand_id = dbo.sneaker.brand_id
GROUP BY dbo.brand.brand_name
```



Results | Messages

	brand_name	number_of_models
1	Adidas	5
2	All Star Convers	4
3	Nike	10
4	Puma	6
5	Vans	3

view_store_and_models_with_more_than_100_pair

Opis:

Pogled koji prikazuje sve modele koji u radnjama ima na stanju preko 100, ucestvuju tabele: sneaker, category, brand, size, sneaker_size, store, city, sneaker_availability.

```
SELECT dbo.sneaker.name, dbo.brand.brand_name, dbo.size.size_eu, dbo.city.city_name, dbo.store.store_name, dbo.sneaker_availability.quantity
FROM dbo.sneaker_size INNER JOIN
    dbo.size ON dbo.sneaker_size.size_id = dbo.size.size_id INNER JOIN
    dbo.sneaker_availability ON dbo.sneaker_size.sneaker_size_id = dbo.sneaker_availability.sneaker_size_id INNER JOIN
    dbo.store ON dbo.sneaker_availability.store_id = dbo.store.store_id INNER JOIN
    dbo.sneaker ON dbo.sneaker_size.sneaker_id = dbo.sneaker.sneaker_id INNER JOIN
    dbo.brand ON dbo.sneaker.brand_id = dbo.brand.brand_id INNER JOIN
    dbo.city ON dbo.store.city_id = dbo.city.city_id
WHERE (dbo.sneaker_availability.quantity > 100) AND (dbo.sneaker.active = 1)
```

Results | Messages

	name	brand_name	size_eu	city_name	store_name	quantity
1	Jordan 6 Rings	Nike	41	Beograd	TC BIG Shopping Centar	245
2	Jordan 6 Rings	Nike	41	Beograd	PJ Buzz Ada Mall	123
3	Jordan 6 Rings	Nike	42	Beograd	PJ Buzz Ušće	124
4	Jordan 6 Rings	Nike	43	Sombor	PJ Buzz Sombor	111
5	Air More Uptempo	Nike	44	Niš	PJ Delta Niš	136
6	Air More Uptempo	Nike	46	Beograd	PJ Buzz BW	125
7	Air Force 1	Nike	41	Kraljevo	PJ Outlet Kraljevo	137
8	Air Force 1	Nike	44	Beograd	PJ Buzz BW	125
9	Air Max 90	Nike	46	Kraljevo	PJ Outlet Kraljevo	167
10	Air Max 270	Nike	43	Niš	PJ Buzz Niš	144
11	Air Max 270	Nike	43	Novi Sad	PJ Buzz Novi Sad	113
12	AIR JORDAN 1 MID	Nike	36	Kraljevo	PJ Outlet Kraljevo	190
13	AIR JORDAN 1 MID	Nike	38	Niš	PJ Delta Niš	122
14	Air Max Structure	Nike	37	Kraljevo	PJ Outlet Kraljevo	122

view_all_stores

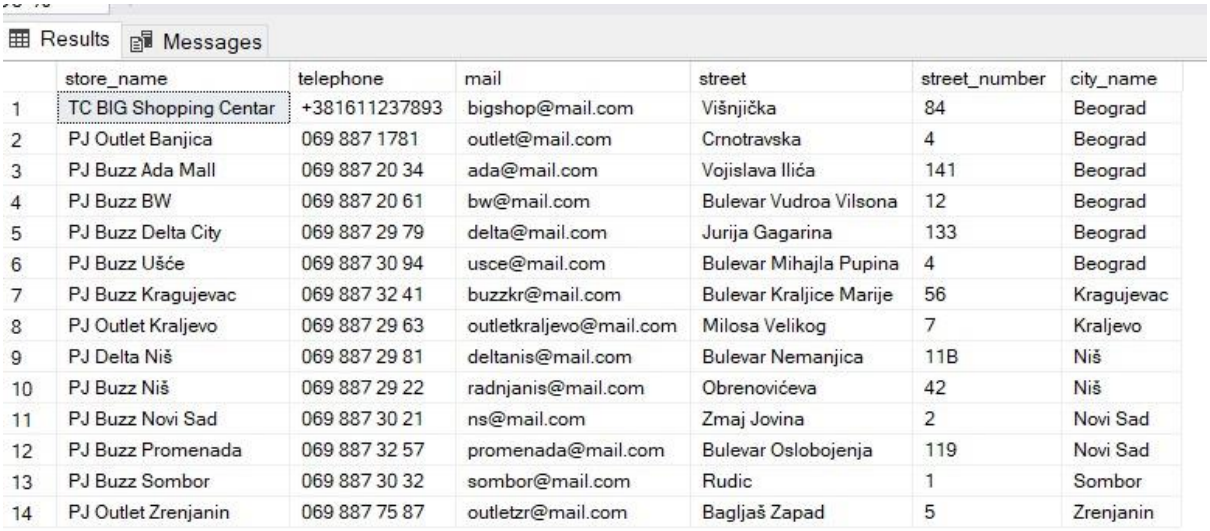
Opis:

Pogled koji prikazuje sve prodavnice, ucestvuju tabele: city, store.

SELECT dbo.store.store_name, dbo.store.telephone, dbo.store.mail, dbo.store.street, dbo.store.street_number, dbo.city.city_name

FROM dbo.city INNER JOIN

dbo.store ON dbo.city.city_id = dbo.store.city_id



Results Messages

	store_name	telephone	mail	street	street_number	city_name
1	TC BIG Shopping Centar	+381611237893	bigshop@mail.com	Višnjička	84	Beograd
2	PJ Outlet Banjica	069 887 1781	outlet@mail.com	Crnotravska	4	Beograd
3	PJ Buzz Ada Mall	069 887 20 34	ada@mail.com	Vojislava Ilića	141	Beograd
4	PJ Buzz BW	069 887 20 61	bw@mail.com	Bulevar Vudroa Vilsona	12	Beograd
5	PJ Buzz Delta City	069 887 29 79	delta@mail.com	Jurija Gagarina	133	Beograd
6	PJ Buzz Ušće	069 887 30 94	usce@mail.com	Bulevar Mihajla Pupina	4	Beograd
7	PJ Buzz Kragujevac	069 887 32 41	buzzkr@mail.com	Bulevar Kraljice Marije	56	Kragujevac
8	PJ Outlet Kraljevo	069 887 29 63	outletkraljevo@mail.com	Milosa Velikog	7	Kraljevo
9	PJ Delta Niš	069 887 29 81	deltanis@mail.com	Bulevar Nemanjica	11B	Niš
10	PJ Buzz Niš	069 887 29 22	radnjanis@mail.com	Obrenovićeva	42	Niš
11	PJ Buzz Novi Sad	069 887 30 21	ns@mail.com	Zmaj Jovina	2	Novi Sad
12	PJ Buzz Promenada	069 887 32 57	promenada@mail.com	Bulevar Oslobojenja	119	Novi Sad
13	PJ Buzz Sombor	069 887 30 32	sombor@mail.com	Rudic	1	Sombor
14	PJ Outlet Zrenjanin	069 887 75 87	outletzr@mail.com	Bagljaš Zapad	5	Zrenjanin

Uskladistene procedure

Nisu radjenje za sve tabele i ako bi trebalo da postoje,ali nije bilo dovoljno vremena,jer je isti nacin rada,za tabele kao sto su brand,category,specification pa je zbog togra uradjeno samo za jednu od tih.

add_new_brand

Koristi se za upis novog brenda u bazu i proveru da li taj brend vec postoji

```
ALTER PROCEDURE [dbo].[add_new_brand]
    -- Add the parameters for the stored procedure here
    @brand_name NVARCHAR(255)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    IF EXISTS (SELECT * FROM brand WHERE brand_name=@brand_name)
        PRINT 'This brand name already has been in database'
    ELSE
        BEGIN
            INSERT INTO brand
                  (brand_name)
            VALUES (@brand_name)
        END
END
```

add_new_price

Koristi se za unos nove cene, parametric su model patika, ceni I datum pocetka.

```
ALTER PROCEDURE [dbo].[add_new_price]
    -- Add the parameters for the stored procedure here
    @sneaker_id int,
    @price decimal(16,2),
    @started_at DATETIME
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    IF @sneaker_id NOT IN (SELECT sneaker_id FROM sneaker)
        PRINT 'This model does not exist in database'
    ELSE IF @started_at < GETDATE()
        PRINT 'You have to choose date in nearly future not in the past'
    ELSE IF @price < 0
        PRINT 'You have to choose price higher than 0'
    ELSE
        BEGIN
            INSERT INTO price
                  (price, started_at, sneaker_id)
            VALUES (@price,@started_at,@sneaker_id)
        END
END
```

add_new_product

Koristi se za upis novog proizvoda.

```
-- Description: <Description,,>
-- =============================================
ALTER PROCEDURE [dbo].[add_new_product]
    -- Add the parameters for the stored procedure here
    @name nvarchar(255),
    @brand_id int,
    @gender_id int,
    @category_id int,
    @cover_picture_src nvarchar(255),
    @active bit

AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    IF (@brand_id NOT IN (SELECT brand_id from brand) OR @gender_id NOT IN (SELECT gender_id FROM gender)
        OR @category_id NOT IN(SELECT category_id FROM category))
        PRINT 'The data were not entered in the correct format'
    ELSE
        BEGIN
            INSERT INTO sneaker (name, brand_id, gender_id, category_id, cover_picture_src, active)
            VALUES (@name,@brand_id,@gender_id,@category_id,@cover_picture_src,@active)
        END
END
```

delete_account

Koristi se za brisanje naloga.

```
-- Description: <Description,,>
-- =============================================
ALTER PROCEDURE [dbo].[delete_account]
    -- Add the parameters for the stored procedure here
    @user_id INT
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    IF (@user_id NOT IN (SELECT user_id from "user"))
        PRINT 'This account was not exist in database'
    ELSE
        BEGIN
            DELETE FROM "user"
            WHERE user_Id=@user_id
        END
END
```

delete_product

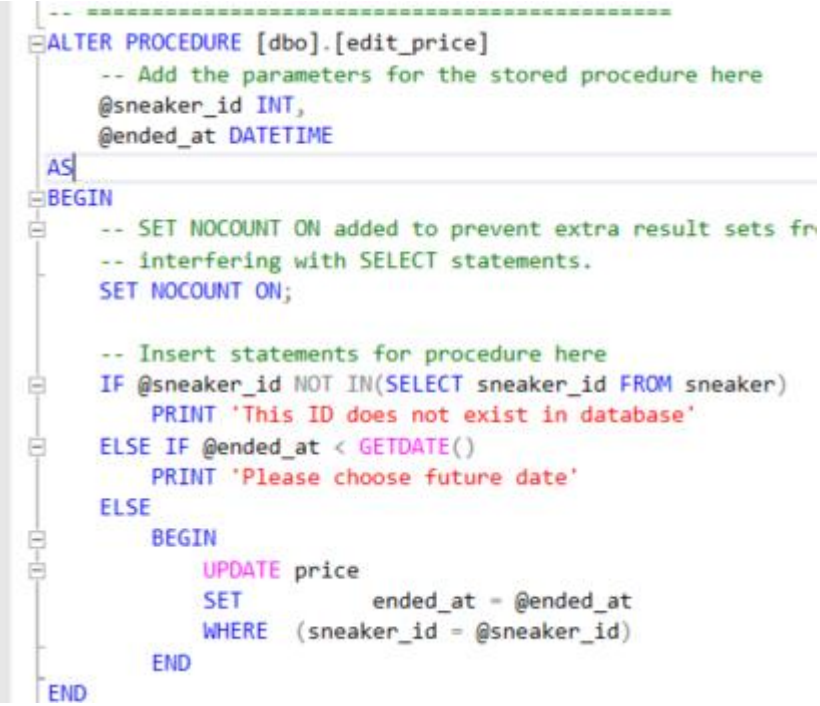
Koristi se za brisanje proizvoda.

```
ALTER PROCEDURE [dbo].[delete_product]
    -- Add the parameters for the stored procedure her
    @id int
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result se
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    IF (@id NOT IN (SELECT sneaker_id from sneaker))
        PRINT 'This model was not exist in database'
    ELSE
        BEGIN
            DELETE FROM sneaker
            WHERE sneaker_id=@id
        END
END
```

edit_price

Koristi se za editovanje cene i parametar ulazni su id patika i datum za kraj cene.



```
-- ============================================
ALTER PROCEDURE [dbo].[edit_price]
    -- Add the parameters for the stored procedure here
    @sneaker_id INT,
    @ended_at DATETIME
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets fr
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    IF @sneaker_id NOT IN(SELECT sneaker_id FROM sneaker)
        PRINT 'This ID does not exist in database'
    ELSE IF @ended_at < GETDATE()
        PRINT 'Please choose future date'
    ELSE
        BEGIN
            UPDATE price
            SET          ended_at = @ended_at
            WHERE  (sneaker_id = @sneaker_id)
        END
END
```

edit_user_last_name

Promena prezimena korisnika,u slucaju da je zensko,pa se mozda udala u medjuvremnu i promenila prezima...

```
-- Description: <Description,,>
-- =============================================
ALTER PROCEDURE [dbo].[edit_user_last_name]
    -- Add the parameters for the stored procedure here
    @user_id int,
    @last_name nvarchar(255)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    IF @user_id NOT IN (SELECT user_id FROM "user")
        PRINT 'User with this specific ID does not exist in database'
    ELSE
        BEGIN
            UPDATE [user]
            SET         last_name = @last_name
            WHERE  (user_id = @user_id)
        END
END
```

model_data

Podaci za jedan model,parametar je ime modela.

```
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    DECLARE
    @sneaker_name NVARCHAR(255),
    @category_name NVARCHAR(255),
    @brand_name NVARCHAR(255),
    @gender_name NVARCHAR(255),
    @price DECIMAL(16,2),
    @cover_picture_src NVARCHAR(255)

    IF EXISTS (SELECT * FROM sneaker WHERE name=@name)
        BEGIN
        SELECT @brand_name=brand.brand_name, @sneaker_name=sneaker.name,
        @cover_picture_src=sneaker.cover_picture_src,@category_name=category.category_name,@gender_name=gender.gender_name,@price=price.price
        FROM     brand INNER JOIN
                  sneaker ON brand.brand_id = sneaker.brand_id INNER JOIN
                  category ON sneaker.category_id = category.category_id INNER JOIN
                  gender ON sneaker.gender_id = gender.gender_id INNER JOIN
                  price ON sneaker.sneaker_id = price.sneaker_id
        WHERE @name=sneaker.name AND price.ended_at IS NULL
        PRINT 'Sneaker model:' + @sneaker_name + '   '
        + 'Brand:' + @brand_name + '   ' + 'Category:' + @category_name + '   ' + 'Gender:'
        + @gender_name + '   ' + 'Cover picture:' + @cover_picture_src + '   ' +'Price:' + CAST(@price AS VARCHAR) + 'RSD'
        END
    ELSE
        PRINT 'The requested model does not exist in the database'
END
```

Funkcije

Tabelarne funkcije

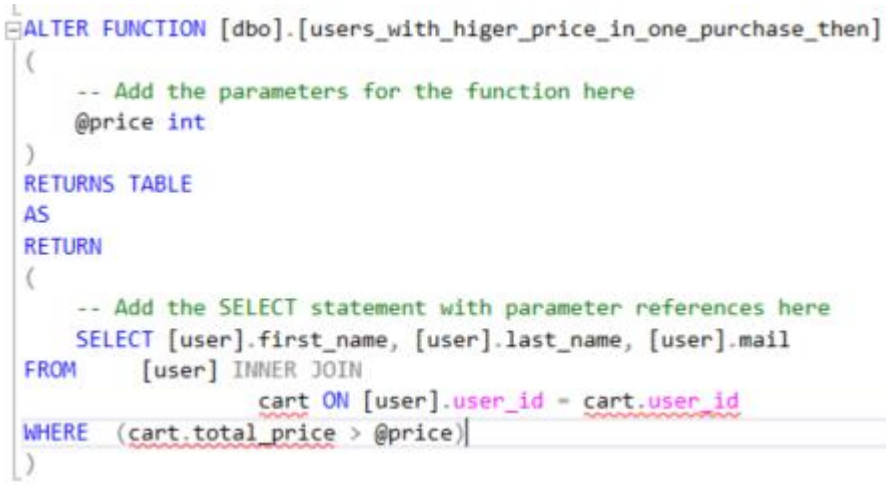
all_models_for_specific_store

Funckija ciji je ulazni parametar naziv prodavnice i ona vraca spisak svih modela dostupnih u toj prodavnici.

```
ALTER FUNCTION [dbo].[all_models_for_specific_store]
(
    -- Add the parameters for the function here
    @store NVARCHAR(255)
)
RETURNS TABLE
AS
RETURN
(
    -- Add the SELECT statement with parameter references here
    SELECT sneaker.name, brand.brand_name, category.category_name, gender.gender_name,
    sneaker_availability.quantity,
    store.store_name, size.size_eu, size.size_cm
    FROM
    store INNER JOIN sneaker_availability ON store.store_id = sneaker_availability.store_id
    INNER JOIN category INNER JOIN brand INNER JOIN
    sneaker_size INNER JOIN
    size ON sneaker_size.size_id = size.size_id INNER JOIN
    sneaker ON sneaker_size.sneaker_id = sneaker.sneaker_id ON brand.brand_id = sneaker.brand_id ON category.category_id = sneaker.category_id
    INNER JOIN
    gender ON sneaker.gender_id = gender.gender_id ON sneaker_availability.sneaker_size_id = sneaker_size.sneaker_size_id
    WHERE  (store.store_name = @store)
)
```

users_with_higher_price_in_one_purchase_than

Funckkija ciji je ulazni parametar cifra i u odnosu na nju se vracaju svi korisnici(ime,prezime,mejl) ciji su racuni bar jednom bili veci od zadate cifre.



```
ALTER FUNCTION [dbo].[users_with_higer_price_in_one_purchase_then]
(
    -- Add the parameters for the function here
    @price int
)
RETURNS TABLE
AS
RETURN
(
    -- Add the SELECT statement with parameter references here
    SELECT [user].first_name, [user].last_name, [user].mail
FROM     [user] INNER JOIN
                  cart ON [user].user_id = cart.user_id
WHERE  (cart.total_price > @price)
)
```

Skalarne funkcije

find_number_of_sneakers_based_on_category_and_gender

Funkcija ciji su ulazni parametni kategorija i pol i na osnovu toga se vraca broj proizvoda za zadati kriterijum,

```
ALTER FUNCTION [dbo].[find_number_of_sneakers_based_on_category_and_gender]
(
    -- Add the parameters for the function here
    @category NVARCHAR(255),
    @gender_type        NVARCHAR(255)
)
RETURNS INT
AS
BEGIN
    -- Declare the return variable here
    DECLARE @result INT

    -- Add the T-SQL statements to compute the return value here
    SET @result=(SELECT COUNT(sneaker.name) AS number_of_models
            FROM     brand INNER JOIN
                  sneaker ON brand.brand_id = sneaker.brand_id INNER JOIN
                  gender ON sneaker.gender_id = gender.gender_id INNER JOIN
                  category ON sneaker.category_id = category.category_id
        WHERE  (category.category_name = @category) AND (gender.gender_name = @gender_type))
    -- Return the result of the function
    RETURN @result

END
```

money_spent_for_one_user

Funkcija koja dobija id korisnika i na osnovu njega sabira sav potrosen novac u nasim radnjama/porudzbinama.

```
-- ============================================
ALTER FUNCTION  [dbo].[money_spent_for_one_user]
(
    -- Add the parameters for the function here
    @id INT
)
RETURNS decimal(16,2)
AS
BEGIN
    -- Declare the return variable here
    DECLARE @R decimal(16,2)

    -- Add the T-SQL statements to compute the return value here
    SELECT @R=SUM(cart.total_price)
    FROM     [user] INNER JOIN cart ON [user].user_id = cart.user_Id
    WHERE  (cart.user_Id = @id)
    -- Return the result of the function
    RETURN @R
END
```

sum_prices_for_one_store_until_today

Funkcija ciji je ulazni parametar ime prodavnice i na osnovu toga sabira sve racune za tu prodavnicu do danasnjeg dana.

```
-- Description: <Description, ,>
-- =============================================
ALTER FUNCTION [dbo].[sum_prices_for_one_store_until_today]
(
    -- Add the parameters for the function here
    @store NVARCHAR(255)
)
RETURNS DECIMAL(16,2)
AS
BEGIN
    -- Declare the return variable here
    DECLARE @R DECIMAL(16,2)

    -- Add the T-SQL statements to compute the return value here
    SELECT @R=SUM(cart.total_price)
    FROM     sneaker_availability INNER JOIN
                  store ON sneaker_availability.store_id = store.store_id INNER JOIN
                  cart_sneaker ON sneaker_availability.sneaker_availability_id = cart_sneaker.sneaker__availability_id INNER JOIN
                  cart ON cart_sneaker.cart_id = cart.cart_id
    WHERE  (store.store_name = @store)

    -- Return the result of the function
    RETURN @R

END
```

Trigeri

Slozeni trigeri,za tabele cart i cart_sneakers nisu uradjeni i objasnjeno je u tekstu iznad,tako da su trigeri uradjeni za insert i update,za svaku tabelu koja postoji u bazi,jer se zadesilo tako da je negde i logicno da na svakom mestu postoje u tabelu aktivnost upisuju se isto sve izmene delete,insert,update za svaku tabelu tako da ce u prilogu ispod biti samo skrinsotovi za jednu tabelu,jer se samo razlikuje ime tabele u celoj skripti.


```
SET QUOTED_IDENTIFIER ON
GO
ALTER TRIGGER [dbo].[update_sneaker_activity]
ON [dbo].[sneaker]
AFTER UPDATE
AS
BEGIN
     SET NOCOUNT ON;
     DECLARE @data NVARCHAR(MAX);
     SET @data = (SELECT * FROM inserted FOR JSON PATH, ROOT('sneaker'));
     IF (@data IS NOT NULL)
         BEGIN
             INSERT INTO activity (action, data)
             VALUES ('The record from the table sneaker has been updated.', @data)
             PRINT 'The sneaker has been successfully updated.';
          END
     UPDATE sneaker
     SET updated_at = GETDATE()
     WHERE (sneaker_id = (SELECT sneaker_id FROM inserted))
END
```

```
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER TRIGGER [dbo].[delete_sneaker_activity]
ON [dbo].[sneaker]
AFTER DELETE
AS
BEGIN
     SET NOCOUNT ON;
     DECLARE @data NVARCHAR(MAX);
     SET @data = (SELECT * FROM deleted FOR JSON PATH, ROOT('sneaker'));
     IF (@data IS NOT NULL)
         BEGIN
             INSERT INTO activity (action, data)
             VALUES ('The record from the table sneaker has been deleted.', @data)
             PRINT 'The sneaker has been successfully deleted.';
         END
END
```

```
GO
ALTER TRIGGER [dbo].[insert_sneaker_activity]
ON [dbo].[sneaker]
AFTER INSERT
AS
BEGIN
     SET NOCOUNT ON;
     DECLARE @data NVARCHAR(MAX);
     SET @data = (SELECT * FROM inserted FOR JSON PATH, ROOT('sneaker'));
     IF (@data IS NOT NULL)
         BEGIN
             INSERT INTO activity (action, data)
             VALUES ('The new record has been inserted into table sneaker.', @data)
             PRINT 'New record has been successfully inserted into the table sneaker.';
         END
     ALTER TABLE sneaker DISABLE TRIGGER update_sneaker_activity;
     UPDATE sneaker
     SET inserted_at = GETDATE()
     WHERE (sneaker_id = (SELECT sneaker_id FROM inserted))
     ALTER TABLE sneaker ENABLE TRIGGER update_sneaker_activity;
END
```