



REPUBLIKA E SHQIPËRISË
UNIVERSITETI I TIRANËS
FAKULTETI I SHKENCAVE TË NATYRËS



Master i Shkencave në Informatikë

DETYRË

AlbaGame Data Warehouse

Sisteme Përpunim Informacioni

ABSTRAKT

Sistemet data warehouse integrojnë të dhëna nga burime heterogjene dhe përdoren nga vendimmarrësit për të analizuar statusin dhe zhvillimin e një biznesi. Të dhënat e papërpunuara që vijnë nga njësi dhe seksione të ndryshme të një biznesi shitjesh zakonisht vijnë nga burime heterogjene të tilla si tabelat e bazës së të dhënave të serverit SQL, të dhëna të llogaritura, ose baza të dhënash të thjeshta MS Access. Por, për të hyrë dhe analizuar të dhënat e papërpunuara nga burime heterogjene, duhet të vendoset në një format të përbashkët me të gjitha marrëdhëniet të paprekura. Kjo është premisa mbi të cilën bazohet koncepti i data warehouse.

Në këtë detyrë propozohet një model i thjeshtë për të ndërtuar një bazë të dhënash dhe një data warehouse për një kompani shitjesh duke përdorur SQL Server 2019. Duke përdorur ETL ne kemi nxjerrë të gjitha të dhënat heterogjene nga burime të ndryshme të të dhënave dhe kemi bërë transformimet e mundshme për t'i kthyer të gjitha në një format të përbashkët dhe në fund të ngarkojmë të gjitha të dhënat në një bazë të dhënash. Pasi të kemi të dhëna homogjene, është e lehtë të analizohen të dhënat. Kjo detyrë bëhet duke krijuar një bazë të dhënash shumë dimensionale nga baza e të dhënave relacionale dhe duke analizuar të dhënat nga këndvështrime të shumta. Pastaj në hapin e fundit bëjmë të mundur që ato të dhëna të pasqyrohen në një format të kuptueshëm nga një njeri i zakonshëm sipas raporteve ose tabelave etj.

Përmbajtja

1. Hyrje	4
2. Nxjerrja e kërkesave të biznesit	5
2.1 Kërkesat funksionale	5
2.2 Kërkesat jofunksionale	6
3. Baza e të dhënave	7
3.1 Skema e bazës së të dhënave relacionale	8
3.2 Përshkrimi i bazës së të dhënave relacionale	9
3.3 Script-et e bazës së të dhënave relacionale	10
4. Dizenjimi logjik i data warehouse	11
4.1 Dimensioni Klient	12
4.2 Dimensioni Dyqan	13
4.3 Dimensioni Kohë	14
4.4 Dimensioni Produkt	15
4.5 Fakti Shitjet	16
4.6 Fakti Shitjet2	17
5. Dizenjimi fizik i data warehouse	18
5.1 Script-et e data warehouse	19
6. Veprime me nivelet dhe dimensionet	20
6.1 Eleminimi i dimensionit Kohë	21
6.2 Eleminimi i nivelit të poshtëm të një dimensionit	22
6.3 Eleminimi i nivelit të poshtëm të një dimensionit që ka 2 hierarki	23
6.4 Eleminimi i nivelit të mesëm të një dimensionit	24
6.5 Shtimi i dimensionit Kohë	25
7. Testimi i të dhënave	26
7.1 Testi i limiteve	27
7.2 Testi spacial nr. 1	28
7.3 Testi spacial nr. 2	29
7.4 Testi i dispersionit	30
8. Konkluzione	31

1. Hyrje

Në këtë detyrë, ne diskutojmë për hartimin e një data warehouse për një biznes që tregton e-commerce (tregtia elektronike). Data warehouse dhe e-commerce janë dy fushat të cilat po zhvillohen me shpejtësi në teknologjinë e informacionit. Data warehouse është një bazë e integruar e të dhënave që përmban të dhëna historike të një biznesi për mbështetjen e proceseve të vendimmarrjes. Një data warehouse ofron një bazë për përpunim analitik të të dhënave për t'u përmirësuar inteligjenca e biznesit duke i kthyer të dhënat në informacion dhe njohuri. Meqenëse teknologjitë për e-commerce po zhvillohen me shpejtësi dhe bizneset elektronike po zgjerohen me shpejtësi, analizimi i mjediseve të e-commerce duke përdorur teknologjinë e data warehouse sjell si pasojë përmirësimin e inteligjencës së biznesit. Një data warehouse e mire dizajnuar “do të ushqente” biznesin me informacionin e duhur në kohën e duhur në mënyrë që të merren vendimet e duhura.

AlbaGame është një shitës argëtimi me pakicë i specializuar në videogames dhe platformave të videogames, aparateve elektronikë dhe aksesorëve të ndryshëm etj. Ka katër dyqane në internet që veprojnë në Shqipëri por jo vetëm. Klientët mund të blejnë produkte individuale si një videogame, një platformë për videogames, ose aksesorë. AlbaGame ka tre kanale kryesore të dorëzimit: Internet, telefon celular, dhe posta. Kompania blen produkte me shumicë, siç janë platformat videogames nga një kompani e tillë, e çfarëdo titulli, për një kosto të caktuar. Për transmetim në internet, kompania paguan një ofrues qendror.

2. Nxjerrja e kërkesave të biznesit

2.1 Nxjerrja e kërkesave funksionale

Kërkesat funksionale përcaktojnë funksionalitetin që ekipi i zhvillimit duhet të ndërtojë në sistemin e data warehouse për të mundësuar përdoruesit për të përmbushur kërkesat dhe nevojat e tyre, duke përmbushur kërkesat e biznesit. Kërkesat funksionale kapin qëllimin dhe sjelljen e sistemit të data warehouse. Kjo sjellje mund të shprehet si shërbime, detyra ose funksione nga të cilat kërkohet sistemi që të kryejë. Ato përshkruajnë atë që duhet të bëjë sistemi i analizës - një veprim që sistemi duhet të ndërmarrë nëse dëshiron të sigurojë funksionalitetet për përdoruesit e saj

Tabela 1.0: Kërkesat funksionale

Nr.	Kërkesat funksionale	Prioriteti
1	Përdoruesit e biznesit duhet të jenë në gjendje të analizojnë "shitjet e produkteve"	I lartë
2	Përdoruesit e biznesit duhet të jenë në gjendje të analizojnë "shitjet e planifikuara"	I lartë
3	Përdoruesit e biznesit do të jenë në gjendje të analizojnë "performancën e furnizuesit", mesatarja e ponderuar e totalit të shpenzuar dhe kostove.	I mesëm
4	Sistemi do t'u mundësojë menaxherëve të dyqaneve të shikojnë të dhënat e dyqaneve të tyre. Kjo sepse çdo menaxher dyqani është përgjegjës për dyqane të ndryshme.	I lartë
5	Në nivelin e dyqanit, aftësia për të parë të dhënat është e rëndësishme. Sistemi duhet t'u mundësojë menaxherëve të dyqaneve të shohin ngritjet dhe uljet e shitjeve, kostove dhe përfitimit.	I lartë

2.2 Nxjerrja e kërkesave jo funksionale

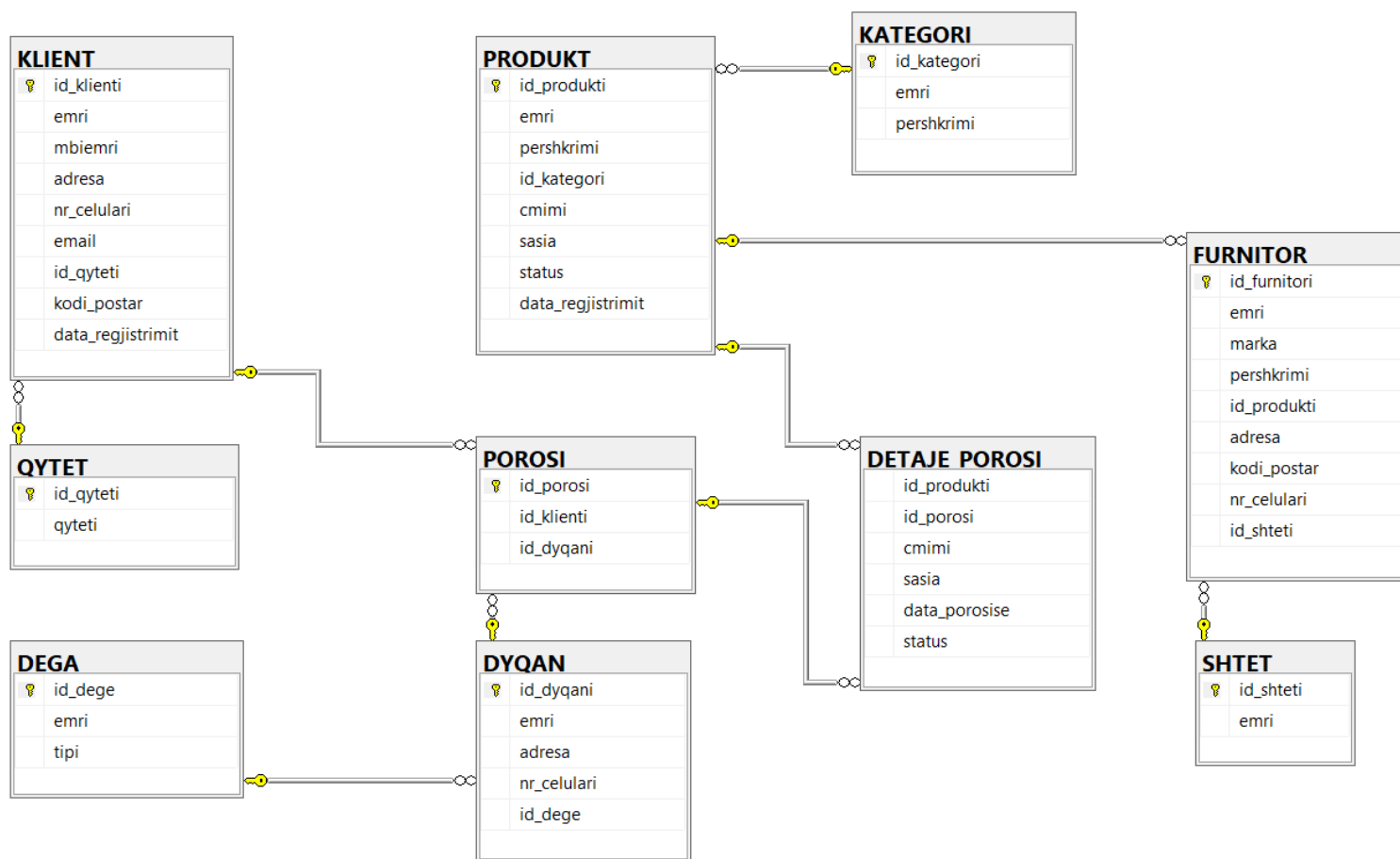
Tabela 1.1: Kërkesat jo funksionale

Nr.	Kërkesat jo funksionale
1	Aplikimet e të të dhënave të data warehouse nuk duhet të jenë të aksesueshme nga rrjeti jashtë kompanisë.
2	Të gjitha aplikacionet e web-based, të arritshme nga kudo brenda rrjetit të kompanisë.
3	Disa të dhëna të sensitive mund të aksesohen vetëm nga njerëz të caktuar.
4	Disa përdoruesve me të drejta shtesë u jepet mundësia për të kryer query SQL direkte

¶ **Shënim:** Pas konsultimeve dhe sqarimeve gjatë orës së seminarit në lidhje me detyrën është parë e arsyeshme ndryshimi i disa elementeve në bazën e të dhënave relacionale burim. Janë bërë disa ndryshime të vogla krahasuar me bazën e të dhënave relacionale dorëzuar në pjesën e parë të detyrës. Në këtë document paraqiten vetëm ndryshimet krahasuar me dokumentin e parë si dhe pjesa e dytë e detyrës(kërkesave). Versioni i ri paraqitet si më poshtë.

3 Baza e të dhënave

3.1 Skema e bazës së të dhënave relacionale



3.2 Përshkrimi i bazës së të dhënave relacionale:

AlbaGame është një shitës argëtimi me pakicë i specializuar në videogames dhe platformave të videogames, aparateve elektronikë dhe aksesoreve të ndryshëm etj. Ka katër dyqane në internet që veprojnë në Shqipëri por jo vetëm. Klientët mund të blejnë produkte individuale si një videogames, një platformë për videogames, ose aksesore etj. Një klienti i ofrohet mundësia e regjistrimit dhe për të ruhen informacione si: emri, mbiemri, email, numri i telefonit, adresa, email, kodi postar dhe qyteti etj. Një klient mund të blejë disa produkte të kategorive të ndryshme. Për produktin ruhen informacione si: emri, përshkrimi, kategoria së cilës i përket si edhe tipi i produktit. Produkti i përket një kategorie të caktuar dhe për kategorinë ruhen informacione si: emri dhe përshkrimi. Këto produkte vijnë nga disa furnitorë të shteteve të ndryshme. Produktet klienti mund t'i porosisë nga 4 dyqane të ndryshme në zona (degë) të ndryshme dhe tipe të ndryshme dyqanesh psh. full outlet, distributor etj. Për dyqanin ruhen informacione si: emri, tipi dhe dega apo rajoni ku gjendet. Lidhjet mes tabelave të bazës së të dhënave variojnë nga lidhje 1 : 1 (psh. Dyqan – Dege), lidhje 1 : N (psh. Furnitor – Produkte) dhe lidhje M : N (psh. Klient – Porosi).

3.3 Script-et e bazës së të dhënave relacionale (krijimi dhe mbushja e tabelave):

```
--=====
-- Krijimi i bazes se te dhenave relacionale AlbaGame  --
--=====
-- AlbaGame Database
-- Sisteme Perpunim Informacioni (SPI)
-- Master i Shkencave ne Informatike
-- Mexhit Kurti
-- Dt. 10 Janar 2021

-- PERMBAJTJA
-- 1.   Tabela      QYTET
-- 2.   Tabela      KLIENT
-- 3.   Tabela      DEGA
-- 4.   Tabela      DYQAN
-- 5.   Tabela      KATEGORI
-- 6.   Tabela      SHTET
-- 7.   Tabela      FURNITOR
-- 8.   Tabela      PRODUKT
-- 9.   Tabela      POROSI
-- 10.  Tabela      DETAJE_POROSI

use AlbaGameRelationalDB
```

```

-----
--      1. Tabela   QYTET      --
-----

```

```

create table QYTET(
    id_qyteti      int    primary key identity(1, 1),
    qyteti         varchar(50) not null
)

```

```

insert into QYTET values('Berat')
insert into QYTET values('Diber')
insert into QYTET values('Durrës')
insert into QYTET values('Elbasan')
insert into QYTET values('Fier')
insert into QYTET values('Lushnjë')
insert into QYTET values('Gjirokastrë')
insert into QYTET values('Korçë')
insert into QYTET values('Pogradec')
insert into QYTET values('Kukës')
insert into QYTET values('Lezhë')
insert into QYTET values('Shkoder')
insert into QYTET values('Tiranë')
insert into QYTET values('Vlorë')
insert into QYTET values('Sarandë')

```

```

-----
--      2. Tabela   KLIENT      --
-----

```

```

create table KLIENT(
    id_klienti      int    primary key identity(1, 1),
    emri            varchar(50) not null,
    mbiemri        varchar(50) not null,
    adresa          varchar(100),
    nr_celulari     varchar(20),
    email           varchar(50),
    id_qyteti       int    foreign key references QYTET(id_qyteti),
    kodi_postar     varchar(4),
    data_regjistrimit date
)

```

```

insert into KLIENT(emri, mbiemri) values('Mexhit', 'Kurti')
insert into KLIENT(emri, mbiemri) values('Lorenc', 'Zhuka')
insert into KLIENT(emri, mbiemri) values('Marinel', 'Bektasha')
insert into KLIENT(emri, mbiemri) values('Naum', 'Todolli')
insert into KLIENT(emri, mbiemri) values('Kleina', 'Cika')
insert into KLIENT(emri, mbiemri) values('Marsiona', 'Stafa')
insert into KLIENT(emri, mbiemri) values('Kejsi', 'Asllanaj')
insert into KLIENT(emri, mbiemri) values('Ervin', 'Shehu')
insert into KLIENT(emri, mbiemri) values('Xhenaro', 'Ruci')

```



```

insert into KLIENT(emri, mbiemri) values('Jonida', 'Krraba')
insert into KLIENT(emri, mbiemri) values('Lorenc', 'Totri')
insert into KLIENT(emri, mbiemri) values('Alket', 'Kurti')
insert into KLIENT(emri, mbiemri) values('Redian', 'Kanani')
insert into KLIENT(emri, mbiemri) values('Ardit', 'Ademi')
insert into KLIENT(emri, mbiemri) values('Eraldo', 'Forgali')
insert into KLIENT(emri, mbiemri) values('Albion', 'Gjoni')
insert into KLIENT(emri, mbiemri) values('Erion', 'Isaku')
insert into KLIENT(emri, mbiemri) values('Aleksander', 'Deda')
insert into KLIENT(emri, mbiemri) values('Loresa', 'Hoxha')
insert into KLIENT(emri, mbiemri) values('Elsa', 'Imeraj')
insert into KLIENT(emri, mbiemri) values('Mario', 'Malja')
insert into KLIENT(emri, mbiemri) values('Anjeza', 'Sejdiu')
insert into KLIENT(emri, mbiemri) values('Serafin', 'Frroku')
insert into KLIENT(emri, mbiemri) values('Eva', 'Biba')
insert into KLIENT(emri, mbiemri) values('Amela', 'Beshiri')

update top(25) KLIENT set data_regjistrimit = GETDATE()
update top(25) KLIENT set kodi_postar = (
    select DISTINCT(ABS(CHECKSUM(NEWID())) % (1033 - 1001 - 1)) + 1001))
declare @i int, @n int
set @n = (select count(id_klienti) from KLIENT)
set @i = 1
while ( @i <= @n)
begin
    print @i
    update KLIENT set id_qyteti = (
        select top 1 id_qyteti from QYTET order by NEWID()) where id_klienti=@i
    update KLIENT set email = (
        select LOWER(emri+mbiemri+'@gmail.com') from KLIENT where id_klienti=@i
    )where id_klienti = @i
    update KLIENT set nr_celulari = (
        select top 1 Phone from Northwind.dbo.Customers order by NEWID())
    where id_klienti = @i
    update KLIENT set adresa = (
        select top 1 Address from Northwind.dbo.Customers order by NEWID())
    where id_klienti = @i
    print 'Updating...'
    if @i >= 25
    begin
        break
    end
    set @i = @i + 1
end
end

```

```

-----
--      3. Tabela   DEGA      --
-----

```

```

create table DEGA(
    id_dege      int primary key identity(1, 1),
    emri         varchar(100) not null,
    tipi         varchar(50)
)

insert into DEGA values('AlbaGame Selvia', 'Distributor')
insert into DEGA values('AlbaGame TEG', 'Full outlet')
insert into DEGA values('AlbaGame Blloku', 'Full outlet')
insert into DEGA values('AlbaGame Rruga e Kavajes', 'Distributor')

```

```

-----
--      4. Tabela   DYQAN     --
-----

```

```

create table DYQAN(
    id_dyqani    int primary key identity(1, 1),
    emri         varchar(100) not null,
    adresa       varchar(100),
    nr_celulari  varchar(20),
    id_dege      int foreign key references DEGA(id_dege)
)

insert into DYQAN values('AlbaGame', 'Rruga e Saracve Tirane, 1001,
Albania', '+355694052404', 1)
insert into DYQAN values('AlbaGame', 'Rruga Mujo Ulqinaku, Tiranë
Albania', '+355696023830', 4)
insert into DYQAN values('AlbaGame', '1001, Rruga Deshmoret e 4 Shkurtit,
Tirana, Albania', '+355682121526', 3)
insert into DYQAN values('AlbaGame', 'Tirana East Gate Rruga e Elbasanit
KM 5, Tiranë, Albania', '+355697035450', 2)

```

```

-----
--      5. Tabela   KATEGORI   --
-----

```

```

create table KATEGORI(
    id_kategori  int primary key identity(1, 1),
    emri         varchar(50) not null,
    pershkrimi   varchar(200)
)

insert into KATEGORI
values('Videogames', 'Videogames for all gaming platform')

```

```

insert into KATEGORI
values('Playstation 5', 'Videogames for all gaming platform')
insert into KATEGORI
values('XBox Series X', 'Videogames for all gaming platform')
insert into KATEGORI
values('XBox One', 'Videogames for all gaming platform')
insert into KATEGORI
values('Switch', 'Videogames for all gaming platform')
insert into KATEGORI
values('PC Gaming', 'Videogames for all gaming platform')
insert into KATEGORI
values('Mobile Gaming', 'Videogames for all gaming platform')
insert into KATEGORI
values('Digital & Downloads', 'Videogames for all gaming platform')
insert into KATEGORI values('Storage', 'Electronics & Computers')
insert into KATEGORI values('PC Peripheral', 'Electronics & Computers')
insert into KATEGORI values('Network', 'Electronics & Computers')
insert into KATEGORI
values('Charger & Adapters', 'Electronics & Computers')
insert into KATEGORI values('XBox Series X', 'Electronics & Computers')
insert into KATEGORI values('Audio', 'Electronics & Computers')
insert into KATEGORI values('Smart Devices', 'Electronics & Computers')
insert into KATEGORI values('Sports & Outdoor', 'Electronics & Computers')
insert into KATEGORI values('Action Figures', 'Hobby & Toys')
insert into KATEGORI values('Sport Toys', 'Hobby & Toys')
insert into KATEGORI values('Board Games', 'Hobby & Toys')
insert into KATEGORI values('Electornic Toys', 'Hobby & Toys')

```

```

-----
--          6. Tabela   SHTET          --
-----

```

```

create table SHTET(
    id_shteti    int primary key identity(1, 1),
    emri         varchar(100) not null,
)

insert into SHTET values('Japan')
insert into SHTET values('United States of America')
insert into SHTET values('Germany')
insert into SHTET values('Great Britain')
insert into SHTET values('Singapore')
insert into SHTET values('Netherlands')
insert into SHTET values('France')
insert into SHTET values('Italy')
insert into SHTET values('China')

```

```

-----
--          7. Tabela   FURNITOR          --
-----

create table FURNITOR(
    id_furnitori    int primary key identity(1, 1),
    emri            varchar(50) not null,
    marka           varchar(50),
    pershkrimi      varchar(100),
    id_produkti     int foreign key references PRODUKT(id_produkti),
    adresa          varchar(50),
    kodi_postar     varchar(10),
    nr_celulari     varchar(20),
    id_shteti       int foreign key references SHTET(id_shteti)
)

insert into FURNITOR values('Play Station', 'PS', 'Gaming Platform', 1, ' ', ' ', ' ', 1)
insert into FURNITOR values('Nintendo Switch', 'NS', 'Gaming Platform', 1, ' ', ' ', ' ', 1)
insert into FURNITOR values('XBox', 'XBox', 'Gaming Platform', 1, ' ', ' ', ' ', 1)
insert into FURNITOR values('Razer', 'Razer', 'Devices', 1, ' ', ' ', ' ', 1)
insert into FURNITOR values('Trust Gaming', 'Trust Gaming', 'Gaming Platform', 1, ' ', ' ', ' ', 1)
insert into FURNITOR values('Steam', 'Steam', 'Gaming Platform', 1, ' ', ' ', ' ', 1)
insert into FURNITOR values('EA', 'EA Sports', 'Video Games', 1, ' ', ' ', ' ', 1)
insert into FURNITOR values('Fortnite', 'Fortnite', 'Video Game', 1, ' ', ' ', ' ', 1)
insert into FURNITOR values('Marvel', 'Marvel', 'Video Games', 1, ' ', ' ', ' ', 1)
insert into FURNITOR values('Konami', 'Konami', 'Video Games', 1, ' ', ' ', ' ', 1)

update top(10) FURNITOR set kodi_postar = (
    select DISTINCT(ABS(CHECKSUM(NEWID())) % (1033 - 1001 - 1)) + 1001))
declare @j int, @m int
set @m = (select count(id_produkti) from PRODUKT)
set @j = 1
while ( @j <= @m)
begin
    print @j
    update FURNITOR set adresa = (
        select top 1 Address from Northwind.dbo.Customers order by NEWID())
    where id_furnitori = @j
    update FURNITOR set id_shteti = (
        select top 1 id_shteti from SHTET order by NEWID())
    where id_furnitori = @j
end

```

```

update FURNITOR set id_produkti = (
select top 1 id_produkti from PRODUKT order by NEWID())
where id_furnitori = @j
    if @j >= @m
        begin
            break
        end
    set @j = @j + 1
end

```

```

-----
--          8. Tabela   PRODUKT      --
-----

```

```

create table PRODUKT(
    id_produkti        int primary key identity(1, 1),
    emri               varchar(50) not null,
    pershkrimi         varchar(100),
    id_kategori        int foreign key references KATEGORI(id_kategori),
    cmimi              money,
    sasia              int,
    status             int
)

insert into PRODUKT values('Console Nintendo Switch', 'Console Nintendo
Switch Mario Red & Blue Special Edition', 5, 397, 100, 1)
insert into PRODUKT
values('U-PlayStation VR', 'U-PlayStation VR', 1, 245, 50, 1)
insert into PRODUKT values('Gaming Chair Trust', 'Gaming Chair Trust GXT
712 Resto PRO', 11, 327, 20, 1)
insert into PRODUKT values('Keyboard Gaming Razer', 'Keyboard Gaming Razer
BlackWidow V3 Mechanical (Green Switch)', 10, 151, 30, 1)
insert into PRODUKT values('Cooling Fan Gaming Redragon', 'Cooling Fan
Gaming Redragon Ivy GCP500', 10, 28, 50, 1)
insert into PRODUKT values('Headset Gaming Redragon', 'Headset Gaming
Redragon Pandora H350 RGB', 10, 32, 50, 1)
insert into PRODUKT values('Microphone Gaming Redragon', 'Microphone
Gaming Redragon Quasar 2 GM200-1', 10, 43, 50, 1)
insert into PRODUKT
values('PS4 Puyo', 'PS4 Puyo Puyo Tetris 2', 5, 31, 50, 1)
insert into PRODUKT
values('XBox Puyo', 'XBox Puyo Puyo Tetris 2', 3, 35, 50, 1)
insert into PRODUKT
values('PS5 Puyo', 'PS5 Puyo Puyo Tetris 2', 5, 40, 50, 1)
insert into PRODUKT values('Colorfilm Instax Square', 'Colorfilm Instax
Square Rainbow WW1', 15, 15, 50, 1)
insert into PRODUKT values('Camera Instax Mini', 'Camera Instax Mini 11
Sky Blue Bundle Box', 9, 127, 50, 1)

```

```

insert into PRODUKT values('Gaming Chair Nacon', 'Gaming Chair Nacon PCCH-310', 8, 147, 50, 1)
insert into PRODUKT values('Game Capture Card Razer Ripsaw HD', 'Game Capture Card Razer Ripsaw HD', 6, 176, 50, 1)
insert into PRODUKT values('Controller Xbox Series X Wireless Shock Black + USB Cable', 3, 77, 50, 1)
insert into PRODUKT values('Usb Charger Ldnio', 'Usb Charger Ldnio 4 Ports 5V/4.4A 22W with LED Lamp White', 12, 16, 50, 1)
insert into PRODUKT values('PS4 Resident Evil 6 PlayStation Hits', 'PS4 Resident Evil 6 PlayStation Hits', 1, 20, 50, 1)
insert into PRODUKT values('Tablet Amazon Fire HD', 'Tablet Amazon Fire HD 8" 64GB B07TMJ1R3X Black', 7, 151, 50, 1)
insert into PRODUKT values('Kindle Amazon Touch 6', 'Kindle Amazon Touch 6" 8GB B07DLPWYB7 White', 7, 131, 50, 1)
insert into PRODUKT values('Electric Scooter Razor', 'Electric Scooter Razor Power Core E90 Black/Pink', 15, 155, 50, 1)
insert into PRODUKT values('Playdoh Kitchen', 'Playdoh Kitchen Creations Grocery Goodies', 17, 22, 50, 1)
insert into PRODUKT values('Lego Storage Minifigure', 'Lego Storage Minifigure Display Case Black 4066', 19, 25, 50, 1)
insert into PRODUKT values('Lego Marvel Super Heroes Avengers', 'Lego Marvel Super Heroes Avengers Helicarrier 76153', 19, 139, 50, 1)
insert into PRODUKT values('Lego Marvel Super Heroes Avengers', 'Lego Marvel Super Heroes Avengers Helicarrier 76153', 19, 139, 50, 1)
insert into PRODUKT values('PS5 Fortnite The Last Laugh Bundle', 'PS5 Fortnite The Last Laugh Bundle', 1, 39, 50, 1)
insert into PRODUKT values('Console PlayStation 5', 'Console PlayStation 5', 1, 508, 50, 1)

```

```

update top(25) PRODUKT set sasia = (
    select DISTINCT(ABS(CHECKSUM(NEWID())) % (100 - 50 - 1)) + 50))

--nqs eshte bere furnizmi atehere produkti eshte i disponueshem per porosi
update PRODUKT set status = 1 where id_produkti IN(
    select id_produkti from FURNITOR)
alter table PRODUKT add data_regjistrimit datetime
update PRODUKT set data_regjistrimit = '01-01-2020'

```

```

-----
--          9. Tabela   POROSI          --
-----

```

```

create table POROSI(
    id_porosi          int primary key identity(1,1),
    id_klienti         int foreign key references KLIENT(id_klienti),
    id_dyqani          int foreign key references DYQAN(id_dyqani),
)

```

```

declare @k int, @x int
set @x = (select count(id_klienti) from KLIENT)
set @k = 1
while ( @k <= @x)
begin
    print @k
    insert into POROSI values(1, 1)
    update POROSI set id_klienti = (
        select top 1 id_klienti from KLIENT order by NEWID())
    update POROSI set id_dyqani = (
        select top 1 id_dyqani from DYQAN order by NEWID())
    if @k >= @x
    begin
        break
    end
    set @k = @k + 1
end

-----
--          10. Tabela  DETAJE_POROSI          --
-----

create table DETAJE_POROSI(
    id_produkti        int foreign key references PRODUKT(id_produkti),
    id_porosi          int foreign key references POROSI(id_porosi),
    cmimi              money,
    sasia              int,
    data_porosise      date,
    status             int
)
declare @a int, @b int, @data_fillestare Date = '2020-01-01'
set @b = (select count(id_porosi) from POROSI)
set @a = 1
while ( @a <= @b)
begin
    print @a
    insert into DETAJE_POROSI values(1, 1, 1, 1, '2021-01-18', 1)
    update DETAJE_POROSI set id_produkti = (
        select top 1 id_produkti from PRODUKT order by NEWID())
    update DETAJE_POROSI set id_porosi = (
        select top 1 id_porosi from POROSI order by NEWID())
    update DETAJE_POROSI set data_porosise =
        DATEADD(DAY, ABS(CHECKSUM(NEWID())) % 364 ), @data_fillestare)
    if @a >= @b
    begin
        break
    end
    set @a = @a + 1
end

```

```

update DETAJE_POROSI set cmimi =
p.cmimi - p.cmimi * (0.01 * ABS(DATEDIFF(month, d.data_porosise,
p.data_regjistrimit))) from DETAJE_POROSI d INNER JOIN PRODUKT p on
d.id_produkti = p.id_produkti;
update DETAJE_POROSI set sasia = (
select (ABS(CHECKSUM(NEWID())) % (10 - 1 - 1)) + 1))
update DETAJE_POROSI set status =
p.status from DETAJE_POROSI d INNER JOIN PRODUKT p
on d.id_produkti = p.id_produkti

```

4 Dizenjimi logjik i data warehouse

4.1 Dimensioni Klient

Për çdo dimension kemi paraqitur tabelën e dimensionit dhe script-in e tij në SQL

Tabela 2.0: Dimensioni Klient

Dimensioni	Nr	Nivelet	Fushat	Burimi			Hirearkitë
				Databaza	Tabela	Fusha	
KLIENT	①	Total	id_total	vlerë fikse (konstante)			<div>①</div> <div>↓</div> <div>②</div> <div>↓</div> <div>③</div>
			total				
	②	Qytet	id_qyteti	AlbaGame	Qytet	id_qyteti	
			qyteti			qyteti	
	③	Klient	id_klient	AlbaGame	Klient	id_klient	
			emer			emer	
			mbiemer			mbiemer	
			email			email	
			adrese			adrese	
			nr_celulari			nr_celulari	
			kodi_postar			kodi_postar	
			data_regjistrimit			data_regjistrimit	

4.2 Dimensioni Dyqan

Dimensioni	Nr	Nivelet	Fushat	Burimi			Hirearkitë
				Databaza	Tabela	Fusha	
DYQAN	①	Total	id_total	vlerë fikse (konstante)			① ↓ ② ↓ ③
			total				
	②	Dega	id_dege	AlbaGame	Dega	id_dege	
			emer			emer	
			tipi			tipi	
	③	Dyqan	id_dyqani	AlbaGame	Dyqan	id_dyqani	
			emer			emer	
			adresa			adresa	
			nr_celulari			nr_celulari	

4.3 Dimensioni Kohw

Dimensioni	Nr	Nivelet	Fushat	Burimi			Hirearkitë
				Databaza	Tabela	Fusha	
KOHË	①	Total	id_total	vlerë fikse (konstante)			① ↓ ② ↓ ③ ↓ ④
			total				
	②	Vit	id_viti	gjenerohet nga procedura			
			viti				
			perskrim_viti				
	③	Muaji	id_muaji	gjenerohet nga procedura			
			muaji				
			perskrim_muaji				
	④	Dita	id_dite	gjenerohet nga procedura			
			dita				
			perskrim_dite				

4.4 Dimensioni Produkt

Tabela 2.3: Dimensioni Produkt

Dimensioni	Nr	Nivelet	Fushat	Burimi			Hirearkitë
				Databaza	Tabela	Fusha	
PRODUKT	①	Total	id_total	vlerë fikse (konstante)			<pre> graph TD 1((1)) --> 4((4)) 1 --> 2((2)) 4 --> 5((5)) 2 --> 3((3)) 3 --> 5 </pre>
			total				
	②	Shtet	id_shtet	AlbaGame	Shtet	id_shteti	
			shteti			emri	
	③	Furnitor	id_furnitori	AlbaGame	Furnitor	id_furnitori	
			emer			emer	
			adrese			adrese	
			marka			marka	
			pershkrimi			pershkrimi	
			kodi_postar			kodi_postar	
			nr_celulari			nr_celulari	
	④	Kategori	id_kategori	AlbaGame	Kategori	id_kategori	
			emer			emer	
			pershkrim			pershkrim	
	⑤	Produkt	id_produkti	AlbaGame	Produkt	id_produkti	
			emer			emer	
			pershkrimi			pershkrimi	
			cmimi			cmimi	
			sasia			sasi	
			status			status	

4.5 Fakti Shitjet

Tabela 3.0: Fakti Shitjet

Fakti	Dimensionet	Të matshmet		Burimet		
		Emërtimi	Funks. agregat	Databaza	Tabela	Kolona
SHITJET	Klient	sasia	Sum	AlbaGame	Detaje_Porosi	sasia
	Produkt	cmimi_max	max			cmimi
	Dyqan	cmimi_min	min			cmimi
	Kohe	vlera	sum			$cmim \times sasi$

4.6 Fakti Shitjet2

Tabela 3.1: Fakti Shitjet2

Fakti	Dimensionet	Të matshmet		Burimet		
		Emërtimi	Funks. agregat	Databaza	Tabela	Kolona
SHITJET2	Klient	sasia	sum	AlbaGame	Porosi	sasia
	Dyqan					
		vlera	sum		Detaje_Porosi	$cmim \times sasi$
	Kohe					

5 Dizenjimi fizik i data warehouse

```

-----
-- Krijimi i data warehouse per AlbaGame --
-----

-- AlbaGame Data Warehouse
-- Sisteme Perpunim Informacioni (SPI)
-- Master i Shkencave ne Informatike
-- Mexhit Kurti
-- Dt. 15 Janar 2021

-- PERMBAJTJA
-- 1. Konfigurimi i DW
-- 2. Dimensioni KLIENT
-- 3. Dimensioni DYQAN
-- 4. Dimensioni PRODUKT
-- 5. Dimensioni KOHE
-- 6. Fakti SHITJET
-- 7. Fakti SHITJET2
-- 8. Celja Fillestare

-----
--          1. Konfigurimi i DW          --
-----
go alter database AlbaGame set recovery simple
go alter database AlbaGame set auto_shrink off
go alter database AlbaGame set auto_create_statistics on
go alter database AlbaGame set auto_update_statistics on
go

use AlbaGameDW

-----
--          2. Dimensioni KLIENT          --
-----
go if exists( select * from sys.tables where name = 'dim_klient' )
drop table dim_klient go
create table dim_klient(
    id_dim_klienti          int primary key identity(1, 1),
    id_totali               int,
    totali                  varchar(100),
    id_qyteti               int,
    qyteti                  varchar(50),
    id_klienti              int,
    emri                    varchar(50),
    mbiemri                 varchar(50),
    adresa                  varchar(100),
    nr_celulari             varchar(20),
    email                   varchar(50),
    kodi_postar             varchar(4),
    data_regjistrimit       date,
    klienti_burim           int
)

```

```
--mbushja e dimensionit Klient
create or alter procedure mbushKlienteFillestare
as
begin
    delete dim_klient
    --mbushja e nivelit Total
    insert into dim_klient(totali) values('Totali i klienteve')
    update dim_klient set id_totali = id_dim_klienti

    --mbushja e nivelit Qytet
    insert into dim_klient(id_totali, totali, id_qyteti, qyteti)
    select dk.id_totali, dk.totali, q.id_qyteti, q.qyteti
    from AlbagameDW.dbo.dim_klient dk, AlbaGameRelationalDB.dbo.QYTET q
    join AlbaGameRelationalDB.dbo.KLIENT k
    on q.id_qyteti = k.id_qyteti
    order by q.id_qyteti, q.qyteti
    --update dim_klient set id_qyteti = id_dim_klienti where qyteti is not null

    --mbushja e nivelit Klient
    insert into dim_klient(id_totali, totali, id_qyteti, qyteti, emri, mbiemri,
    adresa, nr_celulari, email, kodi_postar, data_regjistrimit, klienti_burim)
    select dk.id_totali, dk.totali, dk.id_qyteti, dk.qyteti, k.emri, k.mbiemri,
    k.adresa, k.nr_celulari, k.email, k.kodi_postar, k.data_regjistrimit,
    k.id_klienti
    from dim_klient dk join AlbaGameRelationalDB.dbo.KLIENT k
    on dk.id_qyteti = k.id_qyteti
    update dim_klient set id_klienti = id_dim_klienti
    where klienti_burim is not null
end
```

```
-----
--          3. Dimensioni DYQAN          --
-----
go
if exists(
    select * from sys.tables where name = 'dim_dyqan'
)
drop table dim_dyqan
go
create table dim_dyqan(
    id_dim_dyqan          int primary key identity(1, 1),
    id_totali             int,
    totali                varchar(100),
    id_dege               int,
    emer_dege             varchar(100),
    tipi                 varchar(50),
    dege_burim            int,
    id_dyqani             int,
    emer_dyqani           varchar(100),
    adresa                varchar(100),
    nr_celulari           varchar(20),
    dyqan_burim           int
)
```

```

--mbushja e dimensionit Dyqan
create or alter procedure mbushDyqaneFillestare
as
begin
    delete dim_dyqan
    --mbushja e nivelit Total
    insert into dim_dyqan(totali) values('Totali i dyqaneve')
    update dim_dyqan set id_totali = id_dim_dyqan

    --mbushja e nivelit Dege
    insert into dim_dyqan(id_totali, totali, emer_dege, tipi, dege_burim)
    select dd.id_totali, dd.totali, dg.emri, dg.tipi, dg.id_dege
    from AlbagameDW.dbo.dim_dyqan dd, AlbaGameRelationalDB.dbo.DEGA dg
    join AlbaGameRelationalDB.dbo.DYQAN dq
    on dg.id_dege = dq.id_dege
    order by dg.id_dege, dg.emri

    update dim_dyqan set id_dege = id_dim_dyqan where dege_burim is not null

    --mbushja e nivelit Dyqan
    insert into dim_dyqan(id_totali, totali, id_dege, emer_dege, tipi, dege_burim,
    emer_dyqani, adresa, nr_celulari, dyqan_burim)
    select dd.id_totali, dd.totali, dd.id_dege, dd.emer_dege, dd.tipi,
    dd.dege_burim, dq.emri, dq.adresa, dq.nr_celulari, dq.id_dyqani
    from AlbagameDW.dbo.dim_dyqan dd
    join AlbaGameRelationalDB.dbo.DYQAN dq
    on dd.id_dege = dq.id_dege
    order by dd.id_dege, dd.emer_dege
    update dim_dyqan set id_dyqani = id_dim_dyqan where dyqan_burim is not null
end

```

```

-----
--          4. Dimensioni PRODUKT          --
-----
go
if exists(
    select * from sys.tables where name = 'dim_produkt'
)
drop table dim_produkt
go
create table dim_produkt(
    id_dim_produkt          int primary key identity(1, 1),
    id_totali               int,
    totali                  varchar(100),
    id_shteti               int,
    shteti                  varchar(100),
    id_furnitori            int,
    emer_furnitori          varchar(50),
    marka                   varchar(50),
    adresa                  varchar(50),
    kodi_postar             varchar(10),
    nr_celulari             varchar(20),
    furnitor_burim          int,
    id_kategori             int,
    emer_kategorie          varchar(50),
    pershkrimi_k            varchar(100),
    kategori_burim          int,
    id_produkti             int,
    emer_produkti           varchar(50),
    pershkrimi_p            varchar(100),
    cmimi                   int,
    sasia                   int,
    status                  int,
    produkt_burim           int
)

--mbushja e diemnsionit Produkt
create or alter procedure mbushProdukteFillestare
as
begin
    delete dim_produkt
    --mbushja e nivelit Total
    insert into dim_produkt(totali) values('Totali i produkteve')
    update dim_produkt set id_totali = id_dim_produkt

    --mbushja e nivelit Shtet
    insert into dim_produkt(id_totali, totali, id_shteti, shteti)
    select dp.id_totali, dp.totali, s.id_shteti, s.emri
    from AlbagameDW.dbo.dim_produkt dp, AlbaGameRelationalDB.dbo.SHTET s
    join AlbaGameRelationalDB.dbo.FURNITOR f
    on s.id_shteti = f.id_shteti
    order by s.id_shteti, s.emri
    --update dim_klient set id_qyteti = id_dim_klienti where qyteti is not null

```

```

--mbushja e nivelit Furnitor
insert into dim_produkt(id_totali, totali, id_shteti, shteti, emer_furnitori,
marka, adresa, kodi_postar, nr_celulari, furnitor_burim)
select dp.id_totali, dp.totali, dp.id_shteti, dp.shteti, f.emri, f.marka,
f.adresa, f.kodi_postar, f.nr_celulari, f.id_furnitori
from AlbagameDW.dbo.dim_produkt dp
join AlbaGameRelationalDB.dbo.FURNITOR f
on dp.id_shteti = f.id_shteti
order by dp.id_shteti, dp.shteti

update dim_produkt set id_furnitori = id_dim_produkt
where furnitor_burim is not null

--mbushja e nivelit Kategori
insert into dim_produkt(id_totali, totali, emer_kategorie, pershkrimi_k,
kategori_burim)
select dp.id_totali, dp.totali, k.emri, k.pershkrimi, k.id_kategori
from AlbagameDW.dbo.dim_produkt dp
join AlbaGameRelationalDB.dbo.FURNITOR f
on dp.furnitor_burim = f.id_furnitori
join AlbaGameRelationalDB.dbo.PRODUKT p
on f.id_produkti = p.id_produkti
join AlbaGameRelationalDB.dbo.KATEGORI k
on p.id_kategori = k.id_kategori
order by dp.kategori_burim, emer_kategorie

update dim_produkt set id_kategori = id_dim_produkt
where kategori_burim is not null

--mbushja e nivelit Produkt
insert into dim_produkt(id_totali, totali, emer_kategorie, pershkrimi_k,
kategori_burim, emer_produkti, pershkrimi_p, cmimi, sasia, status,
produkt_burim)
select dp.id_totali, dp.totali, dp.emer_kategorie, dp.pershkrimi_p,
dp.kategori_burim, p.emri, p.pershkrimi, p.cmimi, p.sasia, p.status,
p.id_produkti
from AlbagameDW.dbo.dim_produkt dp
join AlbaGameRelationalDB.dbo.PRODUKT p
on dp.kategori_burim = p.id_kategori
order by dp.kategori_burim, emer_kategorie

update dim_produkt set id_produkti = id_dim_produkt
where produkt_burim is not null
end

-----
--          6. Fakti SHITJET          --
-----

go
if exists(
    select * from sys.tables where name = 'fakt_shitjet'
)
drop table fakt_shitjet

go

```



```

create table fakt_shitjet(
    klient        int foreign key references dim_klient(id_dim_klienti),
    produkt       int foreign key references dim_produkt(id_dim_produkt),
    dyqan         int foreign key references dim_dyqan(id_dim_dyqan),
    kohe          int foreign key references dim_kohe(id_dim_kohe),
    sasia         int,
    cmimi_mes     money,
    cmimi_min     money,
    cmimi_max     money,
    vlera         money,
    primary key(klient, produkt, dyqan, kohe)
)

--mbushja e faktit Shitjet
create or alter procedure mbushShitjetFilestare
as
begin
    insert into fakt_shitjet
    select distinct dk.id_dim_klienti, dp.id_dim_produkt, dd.id_dim_dyqan,
    dkh.id_dim_kohe,
    sum(d.sasia) sasia,
    avg(d.cmimi) cmimi_mes,
    min(d.cmimi) cmimi_min,
    max(d.cmimi) cmimi_max,
    sum(d.sasia * d.cmimi) vlera
    from AlbaGameRelationalDB.dbo.POROSI p
    join AlbaGameRelationalDB.dbo.DETAJE_POROSI d on p.id_porosi = d.id_porosi
    join dim_klient dk on p.id_klienti = dk.klienti_burim
    join dim_dyqan dd on p.id_dyqani = dd.dyqan_burim
    join dim_produkt dp on dp.produkt_burim = d.id_produkti
    join dim_kohe dkh on dkh.muaji = month(d.data_porosise) and dkh.viti =
    year(d.data_porosise) and dkh.dita = day(d.data_porosise)
    group by dk.id_dim_klienti, dd.id_dim_dyqan, dp.id_dim_produkt,
    dkh.id_dim_kohe
end

-----
--          7. Fakti SHITJET2          --
-----

go
if exists(
    select * from sys.tables where name = 'fakt_shitjet2'
)
drop table fakt_shitjet2
go
create table fakt_shitjet2(
    klient        int foreign key references dim_klient(id_dim_klienti),
    dyqan         int foreign key references dim_dyqan(id_dim_dyqan),
    kohe          int foreign key references dim_kohe(id_dim_kohe),
    sasia         int,
    vlera         money,
    primary key(klient, dyqan, kohe)
)

```

```
--mbushja e faktit Shitjet2
create or alter procedure mbushShitjetFilestare2
as
begin
    insert into fakt_shitjet2
    select dk.id_dim_klienti, dd.id_dim_dyqan, dkh.id_dim_kohe,
    sum(d.sasia) sasia,
    sum(d.sasia * d.cmimi) vlera
    from AlbaGameRelationalDB.dbo.POROSI p
    join AlbaGameRelationalDB.dbo.DETAJE_POROSI d on p.id_porosi = d.id_porosi
    join dim_klient dk on p.id_klienti = dk.klienti_burim
    join dim_dyqan dd on p.id_dyqani = dd.dyqan_burim
    join dim_kohe dkh on dkh.muaji = month(d.data_porosise) and dkh.viti =
    year(d.data_porosise) and dkh.dita = day(d.data_porosise)
    group by dk.id_dim_klienti, dd.id_dim_dyqan, dkh.id_dim_kohe
end
```

```
-----
--          8. Celja Fillestare          --
-----
--procedura e pergjithshme
create or alter procedure mbushShitjet
as
begin
    exec dbo.mbashKlienteFillestare
    exec dbo.mbashProdukteFillestare
    exec dbo.mbashDyqaneFillestare
    exec dbo.mbashKohaFillestare
    exec dbo.mbashShitjetFilestare
    exec dbo.mbashShitjetFilestare2
end

select * from fakt_shitjet
select * from fakt_shitjet2
```

6. Veprime me nivelet dhe dimensioned (shtim / fshirje)

6.1 Eleminimi i nivelit te poshtem te nje dimensionit

```
-----
--      AlbaGame Data Warehouse
--      Sisteme Perpunim Informacioni      (SPI)
--      Master i Shkencave ne Informatike
--      Mexhit Kurti
--      Dt. 07 Shkurt 2021
--      PJESA 3

--      PERMBAJTJA
--      1. Eleminimi i nivelit te poshtem te nje dimensionit
--      2. Eleminimi i nivelit te poshtem ne dimensionin qe ka 2 hierarki
--      3. Eleminimi i nje niveli te mesem ne nje dimension
--      4. Eleminimi i dimensionit KOHE
```

```

--      5. Shtimi i dimensionit KOHE
--      6. Permbledhje e procedurave te krijuara
--      7. Restore

use AlbaGameDW
-----
--      1. Eleminimi i nivelit te poshtem ne nje dimensionin      --
-----
--Zgjedhim dimensionin KLIENT (dimension me 1 hirearki)
--Eleminojme nivelin me te poshtem (niveli KLIENT)
select * from dim_klient
select * from fakt_shitjet

delete dim_klient where id_dim_klienti is not null
--nk mund ta bejme dicka te tille(pse?)
--PROBLEMI: te dhenat qe jane ne dimensionin DIM_KLIENT ndodhen tek tabela
FAKT_SHITJET
--nje alternative zgjidhjeje eshte krijimi i nje tabele temporare me te dhenat e
faktit
--tabela temporare sherben si kopje e faktit shitje por me te dhena te transformuara
create or alter procedure eleminoDimensioninKlient
as
begin
    select dk.id_dim_klienti, fsh.produkt, fsh.dyqan, fsh.kohe,
    sum(fsh.sasia) sasia,
    sum(fsh.sasia * fsh.cmimi_mes)/sum(fsh.sasia) cmimi_mes,
    min(fsh.cmimi_min) cmimi_min,
    max(fsh.cmimi_max) cmimi_max,
    sum(fsh.vlera) vlera into shitjet_tmp
    from fakt_shitjet fsh
    join dim_klient dk on fsh.klient = dk.id_dim_klienti
    group by dk.id_dim_klienti, fsh.produkt, fsh.dyqan, fsh.kohe

    --fshijme tabelen FAKT_SHITJET
    delete fakt_shitjet

    --te dhenat e transformuara nga tabela SHITJET_TMP i kalojme tek tabela
    FAKT_SHITJET
    insert into fakt_shitjet
    select * from shitjet_tmp

    --fshijme tabelen temporare SHITJET_TMP
    drop table shitjet_tmp

    --niveli qe kerkojme te heqim (KLIENT) ben pjese ne disa fakte
    drop table fakt_shitjet2

    --riemertojme kolonen
    exec sp_rename 'fakt_shitjet.klient', 'klienti_t', 'COLUMN'

    --fshijme te dhenat nga dimensionin KLIENT
    delete from dim_klient where id_dim_klienti is not null
    alter table dim_klient drop id_klienti, emri, mbiemri, adresa, nr_celulari,
    email, kodi_postar, data_regjistrimit, klient_burim

```

```

--riemertojme dimensionin
exec sp_rename 'dim_klient', 'dim_klient_t'

end
--ndryshojme script-et tek procedurat fillestare
--    ...    ...    ...

```

6.2 Eleminimi i nivelit te poshtem ne dimensionin qe ka 2 hierarki

```

-----
--    2. Eleminimi i nivelit te poshtem ne dimensionin qe ka 2 hierarki    --
-----
--Zgjedhim dimensionin PRODUKT (dimension me 2 hirearki)
--Eleminojme nivelin me te poshtem (niveli PRODUKT)

--fshirja e nivelit PRODUKT sjell si pasoje ndryshimin e lidhjes se dimensionit
PRODUKT
--ne kete rast secila hirearki do perfaqesohet ne nje dimension te vetin
--krijimi i tabelave te reja te niveleve si dimensione me vete
create or alter procedure krijDimensioneTeReja
as
begin
    --krijimi i tabelës dim_kategori
    if exists( select * from sys.tables where name = 'dim_kategori' )
    drop table dim_kategori
    create table dim_kategori(
        id_dim_kategori          int primary key,
        id_totali                int,
        totali                   varchar(100),
        id_kategori              int,
        emer_kategorie           varchar(50),
        pershkrimi_k             varchar(100),
        kategori_burim           int
    )

    --krijimi i tabelës dim_furnitor
    if exists( select * from sys.tables where name = 'dim_furnitor' )
    drop table dim_furnitor
    create table dim_furnitor(
        id_dim_furnitor          int primary key,
        id_totali                int,
        totali                   varchar(100),
        id_shteti                int,
        shteti                   varchar(100),
        id_furnitori             int,
        emer_furnitori           varchar(50),
        marka                    varchar(50),
        adresa                    varchar(50),
        kodi_postar              varchar(10),
        nr_celulari              varchar(20),
        furnitor_burim           int
    )
end

```

```

--mbushja e dimensioneve te reja me vlera
create or alter procedure mbushDimensioneTeReja
as
begin
    --mbushja e dimensionit dim_kategori
    insert into dim_kategori
    select id_dim_produkt, id_totali, totali, id_kategori, emer_kategorie,
    pershkrimi_k, kategori_burim
    from dim_produkt
    where id_shteti is null and id_kategori is null and id_furnitori is null

    insert into dim_kategori
    select id_dim_produkt, id_totali, totali , id_kategori, emer_kategorie,
    pershkrimi_k, kategori_burim
    from dim_produkt
    where id_kategori is not null and id_dim_produkt is not null

    --mbushja e dimensionit dim_furnitor
    insert into dim_furnitor(id_dim_furnitor, id_totali, totali)
    select id_dim_produkt, id_totali, totali
    from dim_produkt
    where id_shteti is null and id_kategori is null and id_furnitori is null

    insert into dim_furnitor
    select id_dim_produkt, id_totali, totali, id_shteti, shteti, id_furnitori,
    emer_furnitori, marka, adresa, kodi_postar, nr_celulari, furnitor_burim
    from dim_produkt
    where id_shteti is not null and id_produkti is null
end

--krijojme tabelen fakt te re per keto dimensione
create or alter procedure krijoFaktTeRi
as
begin
    --krijimi i faktit te ri
    if exists(
        select * from sys.tables where name = 'fakt_shitje_te_reja'
    )
    drop table fakt_shitje_te_reja

    create table fakt_shitje_te_reja(
        klient        int foreign key references dim_klient(id_dim_klienti),
        dyqan          int foreign key references dim_dyqan(id_dim_dyqan),
        kategori       int foreign key references dim_kategori(id_dim_kategori),
        furnitor       int foreign key references dim_furnitor(id_dim_furnitor),
        kohe           int foreign key references dim_kohe(id_dim_kohe),
        sasia_re       int,
        cmimi_mes       money,
        cmimi_min       money,
        cmimi_max       money,
        vlera           money,
        primary key(klient, dyqan, kategori, furnitor, kohe)
    )

```

```

--mbushja e faktit te ri me te dhena
insert into fakt_shitje_te_reja
select fsh.klienti_t, fsh.dyqan , dp.id_kategori, dp.id_furnitori, fsh.kohe,
sum(fsh.sasia) sasia_re,
sum(vlera)/sum(fsh.sasia) cmimi_mes,
min(cmimi_min) cmimi_min,
max(cmimi_max) cmimi_max,
sum(vlera) vlera
from fakt_shitjet fsh
join dim_produkt dp on dp.id_dim_produkt = fsh.produkt
group by fsh.klienti_t, fsh.dyqan, dp.id_kategori, dp.id_furnitori, fsh.kohe
end

--procedura per eleminimin e nivelit te poshtem me me sh se 1 hirearki
create or alter procedure eleminoDimensioninProdukt
as
begin
    exec krijoDimensioneTeReja
    exec mbushDimensioneTeReja
    exec krijoFaktTeRi
end

exec eleminoDimensioninProdukt

```

6.3 Eleminimi i nje niveli te mesem ne nje dimension

```

-----
--          3. Eleminimi i nje niveli te mesem ne nje dimension          --
-----
--Zgjedhim dimensionin DYQAN (dimension me 1 hirearki)
--Eleminojme nivelin e mesem (niveli DEGA)

delete dim_dyqan where id_dim_dyqan is not null
--nk mund ta bejme dicka te tille(pse?)
--PROBLEMI: te dhenat qe jane ne dimensionin DIM_DYQAN ndodhen tek tabela
    FAKT_SHITJET
--nje alternative zgjidhjeje eshte krijimi i nje tabele temporare me te dhenat e
    faktit
--tabela temporare sherben si kopje e faktit shitje por me te dhena te transformuara
create or alter procedure eleminoNivelinDege
as
begin
    select fsh.klienti_t, fsh.produkt, dd.id_dim_dyqan, fsh.kohe,
    sum(fsh.sasia) sasia,
    sum(fsh.sasia * fsh.cmimi_mes)/sum(fsh.sasia) cmimi_mes,
    min(fsh.cmimi_min) cmimi_min,
    max(fsh.cmimi_max) cmimi_max,
    sum(fsh.vlera) vlera into shitjet_tmp
    from fakt_shitjet fsh
    join dim_dyqan dd on fsh.dyqan = dd.id_dim_dyqan
    group by fsh.klienti_t, fsh.produkt, dd.id_dim_dyqan, fsh.kohe

```

```

--fshijme tabelen FAKT_SHITJET
delete fakt_shitjet

--te dhenat e transformuara nga tabela SHITJET_TMP i kalojme tek tabela
  FAKT_SHITJET
insert into fakt_shitjet
select * from shitjet_tmp

--fshijme tabelen temporare SHITJET_TMP
drop table shitjet_tmp

--niveli qe kerkojme te heqim (KLIENT) ben pjese ne disa fakte
drop table fakt_shitjet2

--riemertojme kolonen
exec sp_rename 'fakt_shitjet.dyqan', 'dyqani_t', 'COLUMN'

--fshijme te dhenat nga dimensionin KLIENT
delete from dim_dyqan where id_dim_dyqan is not null
alter table dim_dyqan drop id_dege, emer_dege, tipi, dege_burim

--riemertojme dimensionin
exec sp_rename 'dim_dyqan', 'dim_dyqan_t'

end
--ndryshojme script-et tek procedurat fillestare
--      ...      ...      ...
exec eliminoNivelinDege

```

6.4 Eleminimi i dimensionit Kohe

```

-----
--      4. Eleminimi i dimensionit kohe      --
-----
--krijojme nje tabele temporare te faktit
create or alter procedure eliminoDimensioninKohe
as
begin
  select klienti_t, produkt, dyqani_t, kohe,
    sum(sasia) sasia,
    sum(sasia * cmimi_mes)/sum(sasia) cmimi_mes,
    min(cmimi_min) cmimi_min,
    max(cmimi_max) cmimi_max,
    sum(vlera) vlera into shitjet_temp
  from fakt_shitjet
  group by klienti_t, produkt, dyqani_t, kohe

  --fshijme tabelen e faktit
  delete fakt_shitjet

  --fshijme kolonat e dimensioneve te tjera(perjashtuar kolonen produkt)
  alter table shitjet_temp drop column klienti_t, dyqani_t, kohe

```

```

--kalojme te dhenat nga tabela temporare ne tabelen fakt
insert into fakt_shitjet
select * from shitjet_temp

--fshijme dimensionin kohe
drop table dim_kohe

--perditesojme scriptet
--... ..
end

```

6.5 Shtimi i dimensionit Kohe

```

-----
--          5. Shtimi i dimensionit kohe          --
-----
--shtimi i dimensionit kohe do te behet serish nga e para
--puna perfshin krijimin e tabeles
--          mbushjen e tabeles me te dhena
--          krijimin e tabeles fakt perseri nga e para
create or alter procedure shtoDimensioninKohe
as
begin
    --krijimi i tabeles dim_kohe
    if exists(
        select * from sys.tables where name = 'dim_kohe'
    )
    drop table dim_kohe
    create table dim_kohe(
        id_dim_kohe          int primary key identity(1, 1),
        id_totali            int,
        totali               varchar(100),
        id_viti              int,
        viti                 varchar(100),
        pershkrim_viti       varchar(100),
        id_muaji             int,
        muaji                varchar(100),
        pershkrim_muaji      varchar(100),
        id_dite              int,
        dita                 int,
        pershkrim_dite       varchar(100),
    )

    --mbushja me te dhena
    --ne kemi nje procedure qe e kemi ndertuar ne hapat e pare te ndertimit te DW
    --japim run procedures mbushKohaFillestare
    exec mbushKohaFillestare

    --krijimin e tabeles fakt_shitjet
    if exists(
        select * from sys.tables where name = 'fakt_shitjet'
    )

```



```

drop table fakt_shitjet
create table fakt_shitjet(
    klient        int foreign key references dim_klient(id_dim_klienti),
    produkt       int foreign key references dim_produkt(id_dim_produkt),
    dyqan         int foreign key references dim_dyqan_t(id_dim_dyqan),
    kohe          int foreign key references dim_kohe(id_dim_kohe),
    sasia         int,
    cmimi_mes     money,
    cmimi_min     money,
    cmimi_max     money,
    vlera         money,
    primary key(klient, produkt, dyqan, kohe)
)
--mbushja me te dhena
--ne kemi nje procedure qe e kemi ndertuar ne hapat e pare te ndertimit te DW
--japim run procedures mbushShitjetFillestare
exec mbushShitjetFilestare
end

-----
--          5. Permbledhja e detyres 3          --
-----
--duhet te ekzekutohen te gjitha procedurat mesiperme
--eliminimi i nivelit te poshtem me 1 hirearki KLIENT
exec eliminoDimensioninKlient
--eliminimi i nivelit te poshtem me me sh se 1 hirearki PRODUKT
exec eliminoDimensioninProdukt
--eliminimi i nivelit te mesem DEGE
exec eliminoNivelinDege
--eliminimi i dimensionit KOHE
exec eliminoDimensioninKohe
--shtimi i dimensionit KOHE
exec shtoDimensioninKohe

-----
--          6.          RESTORE          --
-----
--keto jane ndryshime te perkohshme
--japim RESTORE databazes per ta rikthyer ne gjendjen e meparshme

```

7. Testimi i te dhenave

Ne datamart te ndertuar me siper, zgjidhni nje prej dimensioneve dhe se bashku me dimensionin e kohes, krijoni nje seri kohore.

Mbi kete seri, qe do jete e formatit Dimension(variabli) Dimension_Kohe(data) Vlere (referohu tabelës [VleraTest] ne leksion) aplikoni testet sipas leksionit duke ndertuar procedurat perkatese.

- a) Test limitesh
- b) Testi spacial Nr.1
- c) Testi spacial Nr.2
- d) Testi dispersionit

Procedurat duhet te jene te ekzekutueshme dhe te japin si rezultat nje liste me vlera te dyshuara.

7.1 Testi i limiteve

```

-----
--          Testimi i te dhenave          --
-----

-----
--      AlbaGame Data Warehouse
--      Sisteme Perpunim Informacioni      (SPI)
--      Master i Shkencave ne Informatike
--      Mexhit Kurti
--      Dt. 20 Janar 2021

--      PERMBAJTJA
--      1.      Krijimi i tabelës seriteBruto
--      2.      Testi i limiteve
--      3.      Testi spacial nr. 1
--      4.      Testi spacial nr. 2
--      5.      Testi i dispersionit

use AlbaGameDW

-----
--      1. Krijimi i tabelës seriteBruto      --
-----
go
if exists(
    select * from sys.tables where name = 'seriteBruto'
)
drop table seriteBruto
go

```

```

create table seriteBruto(
    id            int identity(1, 1) not null,
    variabel       int                null,
    vendodhje     int                null,
    vlera         float              null,
    data          datetime           null
)

--unifikimi i vlerave
update seriteBruto set vlera = -55555 where vlera is null

--mbushim tabelen seriteBruto me te dhena
insert into seriteBruto
select fsh.produkt, fsh.klient, fsh.cmimi_mes, convert(datetime, right('0' +
cast(dh.dita as varchar(2)), 2) + '-' + right('0' + cast(dh.muaji as varchar(2)), 2)
+ '-' + cast(dh.viti as varchar), 103) from fakt_shitjet fsh
join dim_kohe dh on fsh.kohe = dh.id_dim_kohe

--shtojme 2 kolona ne tebele
alter table seriteBruto add diferenca float(3)
alter table seriteBruto add perqindje numeric(10, 3)

--krijojme nje procedure per te gjeneruar diferencat
go
create or alter procedure gjenero_diferencen
as
begin
    declare @variabli int
    set @variabli = 1
    while @variabli <= (select max(variabel) from seriteBruto)
    begin
        update seriteBruto set diferenca = abs(vlera - (
            select max(vlera) from seriteBruto sb2
            where sb2.variabel = @variabli
        ))
        where variabel = @variabli
        set @variabli = @variabli + 1
    end
end
go
--krijojme nje procedure per te gjeneruar perqindjet
create or alter procedure gjenero_perqindjen
as
begin
    declare @variabli int
    set @variabli = 1
    while @variabli <= (select max(variabel) from seriteBruto)
    begin
        update seriteBruto set perqindje = abs(vlera / (
            select max(vlera) from seriteBruto sb2
            where sb2.variabel = @variabli
        )) - 1
    end
end

```

```

        where variabel = @variabli
        set @variabli = @variabli + 1
    end
end

exec gjenero_diferencen
exec gjenero_perqindjen
select * from seriteBruto order by variabel desc

-----
--      2. Testi i limiteve      --
-----
--krijojme tabelen ku do te vendosim testet
go
if exists(
    select * from sys.tables where name = 'testet'
)
drop table testet
go
create table testet(
    id                int identity(1, 1) not null,
    testues           nvarchar(50)      null,
    kategori          nvarchar(200)     null,
    tabela            nvarchar(200)     null,
    kushti            nvarchar(200)     null,
    pershkrim         nvarchar(200)     null,
    fusha1            nvarchar(200)     null,
    fusha2            nvarchar(200)     null,
    limiti_poshtem    float              null,
    limiti_siperm     float              null
)

go
create or alter procedure gjenero_testet_limiteve @kategori varchar(200)
as
begin
    declare @id int, @vlera float, @limiti_poshtem float, @limiti_siperm float
    set @id = 1
    --vendosim limitet per cmimet e produkteve
    set @limiti_poshtem = 0.0
    set @limiti_siperm = 1000.0
    while @id <= (select max(id) from seriteBruto)
    begin
        set @vlera = (select vlera from seriteBruto where id = @id)
        if @vlera > 0 and @vlera <= 1000
        begin
            insert into testet(testues, kategori, tabela, kushti, pershkrim,
            fusha1, fusha2, limiti_poshtem, limiti_siperm)
            values('Testi i limiteve ' + cast(@id as varchar), 'Kategoria 1',
            'testet', 'vlera > 0 and vlera <= 1000', 'Vlera ploteson
            kushtin!', 'Plotesohet', 'Plotesohet', @limiti_poshtem,
            @limiti_siperm)
        end
        else
        begin
            --
        end
    end
end

```

```

        insert into testet(testues, kategori, tabela, kushti, pershkrim,
        fusha1, fusha2, limiti_poshtem, limiti_siperm)
        values('Testi i limiteve ' + cast(@id as varchar), 'Kategoria 1',
        'testet', 'vlera > 0 and vlera <= 1000', 'Vlera nuk ploteson
        kushtin!', 'Duhet te vleresohet', 'Duhet te vleresohet',
        @limiti_poshtem, @limiti_siperm)
    end
    set @id = @id + 1
end
select id, kategori, tabela, kushti, pershkrim, fusha1, fusha2,
limiti_poshtem, limiti_siperm
from testet where kategori = @kategori order by id

--kursori per listen e vlerave qe nk plotesojne kushtin
declare @test varchar, @tabela varchar, @kushti varchar, @pershkrim varchar,
@fusha1 varchar, @fusha2 varchar
declare cursor_vlera_test cursor
for
select * from testet where @fusha1 != 'Plotesohet' or @fusha2 != 'Plotesohet'
open cursor_vlera_test
fetch next from cursor_vlera_test into
    @id,
    @test,
    @kategori,
    @tabela,
    @kushti,
    @pershkrim,
    @fusha1,
    @fusha2,
    @limiti_poshtem,
    @limiti_siperm

while @@FETCH_STATUS = 0
begin
select @id, @test, @kategori, @tabela, @kushti, @pershkrim, @fusha1,
    @fusha2, @limiti_poshtem, @limiti_siperm
    FETCH NEXT FROM cursor_product INTO
        @id,
        @test,
        @kategori,
        @tabela,
        @kushti,
        @pershkrim,
        @fusha1,
        @fusha2,
        @limiti_poshtem,
        @limiti_siperm
end
end
go

exec gjenero_testet_limiteve @kategori = 'Kategoria 1'

```

7.2 Testi special nr. 1

```

-----
--      3. Testi spacial nr. 1      --
-----
go
create or alter procedure gjenero_testet_spaciale
as
begin
    declare @id int, @limit_1 float, @limit_2 float, @vlera_1 float, @vlera_2
    float, @vlera_3 float, @variabel int
    set @id = 1
    while @id <= (select max(variabel) from seriteBruto)
    begin
        set @variabel = (select variabel from seriteBruto where id = @id)
        set @vlera_1 = (select vlera from seriteBruto
        where id = @id and variabel = @variabel)
        set @vlera_2 = (select vlera from seriteBruto
        where id = @id + 1 and variabel = @variabel)
        set @vlera_3 = (select vlera from seriteBruto
        where id = @id + 2 and variabel = @variabel)
        set @limit_1 = @vlera_1 * 0.3
        set @limit_2 = @vlera_1 * 0.01
        if abs(@vlera_1 - @vlera_2) > @limit_1
        begin
            if @id = 1
            begin
                if abs(@vlera_1 - @vlera_2) > @limit_2
                begin
                    insert into testet(testues, kategori, tabela, kushti,
                    pershkrim, fusha1, fusha2, limiti_poshtem,
                    limiti_siperm)
                    values('Testi spacial ' + cast(@id as varchar),
                    'Kategoria 2', 'testet', 'vlera < ' + cast(@limit_1
                    as varchar) + ' and vlera > ' + cast(@limit_2 as
                    varchar) + '1000', 'Vlere e dyshuar!', 'Nuk
                    plotesohet', 'Nuk Plotesohet', @limit_1, @limit_2)
                end
            end
            else if
            abs((@vlera_2 - @vlera_3) - (@vlera_1 - @vlera_2)) > @limit_2
            begin
                insert into testet(testues, kategori, tabela, kushti,
                pershkrim, fusha1, fusha2, limiti_poshtem, limiti_siperm)
                values('Testi spacial ' + cast(@id as varchar), 'Kategoria
                2', 'testet', 'vlera < ' + cast(@limit_1 as varchar) + '
                and vlera > ' + cast(@limit_2 as varchar) + '1000',
                'Vlere e dyshuar!', 'Nuk plotesohet', 'Nuk Plotesohet',
                @limit_1, @limit_2)
            end
            else if abs((@vlera_2 - @vlera_3) - (@vlera_1 - @vlera_2)) = 0
            begin
                insert into testet(testues, kategori, tabela, kushti,
                pershkrim, fusha1, fusha2, limiti_poshtem, limiti_siperm)

```

```

        values('Testi spacial ' + cast(@id as varchar), 'Kategoria
        2', 'testet', 'vlere < ' + cast(@limit_1 as varchar) + '
        and vlere > ' + cast(@limit_2 as varchar), 'Vlere e sakte!
        Nk ka ndryshim', 'Plotesohet', 'Plotesohet', @limit_1,
        @limit_2)
    end
    else if @vlere_1 is null or @vlere_2 is null or @vlere_3 is null
    begin
        print('Raste dyshimi qe duhen kontrolluar')
    end
end
else
begin
    insert into testet(testues, kategori, tabela, kushti, pershkrim,
    fusha1, fusha2, limiti_poshem, limiti_siperm)
    values('Testi spacial ' + cast(@id as varchar), 'Kategoria 2',
    'testet', 'vlere < ' + cast(@limit_1 as varchar) + ' and vlere >
    ' + cast(@limit_2 as varchar), 'Vlere e sakte! Nk ka ndryshim',
    'Plotesohet', 'Plotesohet', @limit_1, @limit_2)
end
set @id = @id + 1
end
end
go

exec gjenero_testet_spaciale

```

7.3 Testi special nr. 2

```

-----
--      4. Testi spacial nr. 2      --
-----
go
create or alter procedure gjenero_testet_spaciale_2
as
begin
    declare @id int, @limit_1 float, @vlere_1 float, @vlere_2 float, @vlere_3
    float, @variabel int
    set @id = 1
    while @id <= (select max(variabel) from seriteBruto)
    begin
        set @variabel = (select variabel from seriteBruto where id = @id)
        set @vlere_1 = (select vlere from seriteBruto
        where id = @id and variabel = @variabel)
        set @vlere_2 = (select vlere from seriteBruto
        where id = @id + 1 and variabel = @variabel)
        set @vlere_3 = (select vlere from seriteBruto
        where id = @id + 2 and variabel = @variabel)
        set @limit_1 = @vlere_1 * 0.3
        if abs(@vlere_1 - @vlere_2) > @limit_1 and abs(@vlere_2 - @vlere_3) >
        @limit_1 and abs(@vlere_1 - @vlere_3) > @limit_1
        begin

```

```

set @vlera_1 = (@vlera_1 + @vlera_2 + @vlera_3) / 3
insert into testet(testues, kategori, tabela, kushti, pershkrim,
fusha1, fusha2, limiti_poshtem, limiti_siperm)
values('Testi spacial nr. 2 - ' + cast(@id as varchar),
'Kategoria 2', 'testet', 'vlera < ' + cast(@limit_1 as varchar),
'Vlere e pasakte! Aplikohet interpoli gjeografik', 'Plotesohet',
'- ', @limit_1, 0)
update seriteBruto set vlera = @vlera_1 where id = @id
end
else
begin
insert into testet(testues, kategori, tabela, kushti,
pershkrim, fusha1, fusha2, limiti_poshtem, limiti_siperm)
values('Testi spacial nr. 2 - ' + cast(@id as varchar),
'Kategoria 2', 'testet', 'vlera < ' + cast(@limit_1 as varchar),
'Vlere e sakte!', 'Plotesohet', '- ', @limit_1, 0)
end
set @id = @id + 1
end
end
go

exec gjenero_testet_spaciale_2

```

7.4 Testi i dispersionit

```

-----
--      5. Testi i dispersionit      --
-----
--krijimi i tabelës vleraTemp
go if exists(
select * from sys.tables where name = 'vleraTemp'
)
drop table vleraTemp
go
create table vleraTemp(
id int,
vlera float
)
--krijimi i tabelës vleraTest
go
if exists(
select * from sys.tables where name = 'vleraTest'
)
drop table vleraTest
go
create table vleraTest(
variabli int,
stacioni int,
data date,
t float,
s float,
vlera float
)

```



```

go
create or alter procedure testiDispersionit
@variabli int
as
begin
    declare @i int, @vendodhje int, @data datetime, @data1 datetime, @data2
    datetime, @vlera float, @t float, @s float, @seri varchar, @stacioni varchar
    declare kursori_vendodhje cursor local for
    select vendodhje from seriteBruto
    --hapim kursorin e vendodhjes
    open kursori_vendodhje
    fetch next from kursori_vendodhje into @vendodhje
    while @@FETCH_STATUS = 0
    begin
        fetch next from kursori_vendodhje into @vendodhje
        set @i = 1
        delete vleraTemp

        declare kursori_date cursor local for
        select data, vlera from seriteBruto
            where vendodhje = @vendodhje and variabel = @variabli
            order by data
        open kursori_date
        fetch next from kursori_date into @data, @vlera
        while @@FETCH_STATUS = 0
        begin
            fetch next from kursori_date into @data, @vlera
            if @i = 1
                set @data1 = @data
            else
                set @data2 = @data

            if DATEADD(DAY, 1, @data1) != @data2
            begin
                delete vleraTemp
                set @data1 = @data
                set @i = 1
            end
            if @i < 31
            begin
                insert into vleraTemp values(@i, @vlera)
                set @i = @i + 1
            end
            else
            begin
                set @seri = (select variabel from seriteBruto
                    where id = @i)
                set @stacioni = (select vendodhje from seriteBruto
                    where id = @i)
                select @t = ROUND(SUM(vlera) / 30, 2) from vleraTemp
                select @s = ROUND(SUM(POWER(vlera - @t, 2)) / 30, 2)
                from vleraTemp
                if ABS(@t - @vlera) >
                    2.05 * ROUND(SQRT(@s), 2) * ROUND(SQRT(30), 2)
            end
        end
    end
end

```

```

        insert into vleraTest values(@seri, @stacioni,
        @data, @t, 2.05 * ROUND(SQRT(@s), 2) * ROUND(SQRT(31
        / 30), 2), @vlera)
        insert into vleraTemp values(@i, @vlera)
        delete vleraTemp where id = @i - 30
        set @i = @i + 1
    end
    set @data1 = @data
end
    end
    close kursori_date
    deallocate kursori_date
end
close kursori_vendodhje
deallocate kursori_vendodhje
go

go
create or alter procedure ekzekuto_testin_dispersionit
as
begin
    declare @j int
    set @j = 1
    while @j <= (select max(variabel) from seriteBruto)
    begin
        exec testiDispersionit @variabli = @j
        set @j = @j + 1
    end
end
go

exec ekzekuto_testin_dispersionit

select * from seriteBruto order by variabel asc
select * from vleraTemp
select * from vleraTest

```

9. Konkluzione

Ne ditet e sotme, bizneset, kryesisht ato ecommerce gjenerojnë vëllime të mëdha të të dhënave. Shumica e tyre kanë mësuar se të dhënat janë një mjet i rëndësishëm vendimmarrës, prandaj ata ruajnë çdo të dhënë të gjeneruar nga departamentet. Qëllimi i ruajtjes së të dhënave është që ato të mund të analizohen për të nxjerrë njohuri që mund të përdoren për vendimmarrje. Bizneset gjithashtu mund të shkojnë në një masë të mbledhjes së të dhënave nga burime të jashtme. Kjo ndodh veçanërisht kur një biznes dëshiron të dijë më shumë për atë që po ndodh në treg dhe t'i kuptojë më mirë nevojat e klienteve.

Me ane te ketij projekti ne mësuar më shumë rreth data warehouse dhe mënyrës së implementimit të SQL Server për Data Warehouse. I gjithë procesi është i gjatë dhe kërkon kohë. Mesuar perceptimin logjik te implementimit ten je data warehouse si nxjerrja e dimensioneve dhe fakteve bazuar ne nje baze te dhenash relacionale dhe me pas ndertimin fizik te data warehouse. Gjithashtu mesuar metodiken e ndryshimeve qe mund te behen ne data warehouse sic eshte ndryshimi ne te dhena, ndryshimi i niveleve ten je dimensionit apo dhe ndryshimi i vete dimensionit.

Ne fund por jo me pak e rendesishmja testimi i data warehouse me ane te testeve per te dhenat e data warehouse, perdorimi i algoritmeve per te implementuar dhe krijuar testet e te dhenave.

Referenca:

2008 Building a Data Warehouse With Examples in SQL Server Authors: Rainardi, Vincent

<https://sites.google.com/a/fshn.edu.al/denisi/home>

<https://sites.google.com/a/fshn.edu.al/course-26/>

<https://www.udemy.com/course/implementing-a-data-warehouse-with-sql-server-2012/>



Mexhit Kurti

Fakulteti i Shkencave të Natyrës

Master Shkencor në Informatikë

Sisteme Përpunim Informacioni

2021