**ΜΠΙΛΙΑΣ ΓΕΩΡΓΙΟΣ 3200278**

**ΕΡΓΑΣΙΑ 2**

**ΖΗΤΗΜΑ 1**

**Ερώτημα 1**

Αρχικά, για την δημιουργία του database πάτησα δεξί κλικ στο Databases στο Object Explorer και Πάτησα New Database… και έβαλα το όνομα CTDW στο πεδίο Database name:

Για την δημιουργία του πίνακα CardsTransactions εκτέλεσα το παρακάτω query:

CREATE TABLE CardsTransactions (

pid int,

pname varchar(50),

age int,

gender char(1),

cardno char(16),

card\_brand varchar(30),

card\_type varchar(20),

tdate datetime,

amount decimal(6,2),

ttc int,

trans\_type varchar(30),

mcc int,

merchant\_city varchar(50)

);

Και στην συνέχεια για να φορτώσω τα δεδομένα στον πίνακα CardsTransactions εκτέλεσα το παρακάτω query:

BULK INSERT CardsTransactions

FROM 'C:\Users\georg\OneDrive\Desktop\CardsTransactions.txt'

WITH (FIRSTROW =2, FIELDTERMINATOR='|', ROWTERMINATOR = '\n');

**Ερώτημα 2**

Για την δημιουργία του λογικού σχήματος της αποθήκης δεδομένων σε μορφή αστέρα (Star Schema) εκτέλεσα τον παρακάτω κώδικα:

CREATE TABLE customers(

pid int primary key,

pname varchar(50),

age int,

gender char(1)

);

CREATE TABLE cards(

cardno char(16) primary key,

card\_brand varchar(30),

card\_type varchar(20)

);

CREATE TABLE timeinfo(

tdate datetime primary key,

t\_year int,

t\_month int,

t\_dayofmonth int,

t\_hour int,

t\_quarter int,

t\_week int,

t\_dayofyear int,

t\_dayofweek int

);

CREATE TABLE cities(

mcc int primary key,

merchant\_city varchar(50)

);

CREATE TABLE transaction\_types(

ttc int primary key,

trans\_type varchar(30)

);

CREATE TABLE transactions(

ttc int,

pid int,

cardno char(16),

tdate datetime,

mcc int,

amount decimal(6,2),

primary key(ttc,pid,cardno,tdate,mcc),

foreign key (ttc) references transaction\_types(ttc),

foreign key (pid) references customers(pid),

foreign key (cardno) references cards(cardno),

foreign key (tdate) references timeinfo(tdate),

foreign key (mcc) references cities(mcc)

);

Όπως και φαίνεται και από τον κώδικα ουσιαστικά δημιούργησα 5 dimension tables (customers,cards,timeinfo,cities,transaction\_types) και ένα fact table (transactions)

**Ερώτημα 3**

Για την τροφοδότηση του σχήματος της αποθήκης με τα απαραίτητα στοιχεία από τον πίνακα CardsTransactions εκτέλεσα τον παρακάτω κώδικα:

INSERT INTO customers(pid,pname,age,gender)

SELECT DISTINCT pid,pname,age,gender

FROM CardsTransactions;

INSERT INTO cards(cardno,card\_brand,card\_type)

SELECT DISTINCT cardno,card\_brand,card\_type

FROM CardsTransactions;

SET DATEFIRST 1;

INSERT INTO timeinfo(tdate,t\_year,t\_month,t\_dayofmonth,t\_hour,t\_quarter,t\_week,t\_dayofyear,t\_dayofweek)

SELECT DISTINCT tdate, datepart(year, tdate), datepart(month, tdate),

datepart(day,tdate),datepart(hour, tdate),

datepart(quarter,tdate), datepart(week,tdate),

datepart(dayofyear,tdate),datepart(dw,tdate)

FROM CardsTransactions;

INSERT INTO cities(mcc,merchant\_city)

SELECT DISTINCT mcc,merchant\_city

FROM CardsTransactions;

INSERT INTO transaction\_types(ttc,trans\_type)

SELECT DISTINCT ttc,trans\_type

FROM CardsTransactions;

INSERT INTO transactions(ttc,pid,cardno,tdate,mcc,amount)

SELECT ttc,pid,cardno,tdate,mcc,amount

FROM CardsTransactions;

**Ερώτημα 4**

Για την δημιουργία του database diagram πάτησα δεξί κλικ στο Database Diagrams στο Object Explorer > CTDW και Πάτησα New Database Diagram, μετά επέλεξα τα 6 Tables (customers,cards,timeinfo,cities,transaction\_types,transactions) και δημιουργήθηκε το παρακάτω Diagram:

**Εικόνα που περιέχει κείμενο, διάγραμμα, στιγμιότυπο οθόνης, Σχέδιο

Περιγραφή που δημιουργήθηκε αυτόματα**

**ΖΗΤΗΜΑ 2**

Οι επερωτήσεις που έγραψα και εκτέλεσα για τα ερωτήματα του ζητήματος 2 είναι οι εξής:

1.

/\*

Εμφανίστε έναν κατάλογο με την αξία των συναλλαγών ανά πόλη. Ο κατάλογος

πρέπει να είναι ταξινομημένος με βάση την πόλη σε αύξουσα διάταξη.

Results: 9.928 rows

\*/

SELECT merchant\_city,sum(amount) AS total\_transactions

FROM cities

INNER JOIN transactions

ON cities.mcc = transactions.mcc

GROUP BY merchant\_city

ORDER BY merchant\_city ASC

2.

/\*

Εμφανίστε έναν κατάλογο με την αξία των συναλλαγών ανά έτος και φύλο. Ο

κατάλογος πρέπει να είναι ταξινομημένος με βάση το έτος σε φθίνουσα διάταξη.

Results: 12 rows

\*/

SELECT t\_year,gender,sum(amount) as total\_transactions

FROM timeinfo

INNER JOIN transactions

ON timeinfo.tdate = transactions.tdate

INNER JOIN customers

ON transactions.pid = customers.pid

GROUP BY t\_year,gender

ORDER BY t\_year DESC

3.

/\*

Εμφανίστε έναν κατάλογο με τον αριθμό και την αξία των συναλλαγών ανά

επωνυμία (card\_brand) είδος (card\_type) κάρτας.

Results: 8 rows

\*/

SELECT card\_brand,card\_type,COUNT(amount) AS amount\_of\_transactions,SUM(amount) AS total\_transactions

FROM cards

INNER JOIN transactions

ON cards.cardno = transactions.cardno

GROUP BY card\_brand,card\_type

4.

/\*

Εμφανίστε έναν κατάλογο με ανάλυση της αξίας των συναλλαγών ανά τύπο

συναλλαγής (trans\_type) σε τριμηνιαία βάση για το έτος 2019.

Results: 12 rows

\*/

SELECT trans\_type,t\_quarter,SUM(amount) as total\_transactions

FROM transactions

INNER JOIN timeinfo

ON timeinfo.tdate = transactions.tdate

INNER JOIN transaction\_types

ON transaction\_types.ttc = transactions.ttc

WHERE t\_year = 2019

GROUP BY trans\_type,t\_quarter

5.

/\*

Η διοίκηση της τράπεζας θέλει μία αναφορά που θα περιέχει τις ακόλουθες

πληροφορίες για τις online συναλλαγές (trans\_type=’Online Transaction’).

a. Την συνολική αξία των online συναλλαγών.

b. Την αξία των online συναλλαγών ανά έτος.

c. Την αξία των online συναλλαγών ανά έτος και φύλο.

d. Την αξία των online συναλλαγών ανά έτος, φύλο και ηλικία.

Results: 851 rows

\*/

SELECT t\_year,gender,age,SUM(amount) as online\_transactions

FROM transactions

INNER JOIN timeinfo

ON timeinfo.tdate = transactions.tdate

INNER JOIN customers

ON customers.pid = transactions.pid

INNER JOIN transaction\_types

ON transaction\_types.ttc = transactions.ttc

WHERE trans\_type='Online Transaction'

GROUP BY ROLLUP (t\_year,gender,age)

**ΖΗΤΗΜΑ 3**

Οι επερωτήσεις που έγραψα και εκτέλεσα για τα ερωτήματα του ζητήματος 2 είναι οι εξής:

1.

/\*

Results: 105 rows

\*/

SELECT t\_year,card\_brand,gender,COUNT(amount) AS transactions\_no

FROM transactions

INNER JOIN timeinfo

ON timeinfo.tdate = transactions.tdate

INNER JOIN cards

ON cards.cardno = transactions.cardno

INNER JOIN customers

ON customers.pid = transactions.pid

GROUP BY CUBE (t\_year,card\_brand,gender)

2.

Για την δημιουργία μιας MΑTERIALIZED όψης (INDEXED VIEW στον SQL SERVER) η οποία θα περιέχει το αποτέλεσμα ενός μόνο GROUP BY του κύβου του ερωτήματος 1 έγραψα τον παρακάτω κώδικα:

--Creating the Materialized view

CREATE VIEW materialized\_view\_cube

WITH SCHEMABINDING

AS

SELECT [ti].t\_year, [ca].card\_brand, [cu].gender, COUNT\_BIG(\*) AS transactions\_no

FROM dbo.transactions [t],dbo.timeinfo [ti],dbo.cards [ca],dbo.customers [cu]

WHERE [ti].tdate = [t].tdate AND [ca].cardno = [t].cardno AND [cu].pid = [t].pid

GROUP BY [ti].t\_year, [ca].card\_brand, [cu].gender

--and its clustered index

CREATE UNIQUE CLUSTERED INDEX idx\_view ON materialized\_view\_cube(t\_year,card\_brand,gender);

Οι κατάλληλες εντολές SQL ώστε να παράγω τα υπόλοιπα GROUP BY του κύβου από την όψη που δημιούργησα είναι οι εξής:

SET STATISTICS IO ON;

-- GROUP BY ( )

SELECT NULL AS t\_year, NULL AS card\_brand, NULL AS gender, SUM(transactions\_no) AS total\_transactions

FROM materialized\_view\_cube

UNION ALL

-- GROUP BY (t\_year)

SELECT t\_year, NULL AS card\_brand, NULL AS gender, SUM(transactions\_no) AS total\_transactions

FROM materialized\_view\_cube

GROUP BY t\_year

UNION ALL

-- GROUP BY (card\_brand)

SELECT NULL AS t\_year, card\_brand, NULL AS gender, SUM(transactions\_no) AS total\_transactions

FROM materialized\_view\_cube

GROUP BY card\_brand

UNION ALL

-- GROUP BY (gender)

SELECT NULL AS t\_year, NULL AS card\_brand, gender, SUM(transactions\_no) AS total\_transactions

FROM materialized\_view\_cube

GROUP BY gender

UNION ALL

-- GROUP BY (card\_brand,gender)

SELECT NULL AS t\_year, card\_brand, gender, SUM(transactions\_no) AS total\_transactions

FROM materialized\_view\_cube

GROUP BY card\_brand,gender

UNION ALL

-- GROUP BY (t\_year,gender)

SELECT t\_year, NULL AS card\_brand, gender, SUM(transactions\_no) AS total\_transactions

FROM materialized\_view\_cube

GROUP BY t\_year,gender

UNION ALL

-- GROUP BY (t\_year,card\_brand)

SELECT t\_year, card\_brand, NULL AS gender, SUM(transactions\_no) AS total\_transactions

FROM materialized\_view\_cube

GROUP BY t\_year,card\_brand

UNION ALL

-- GROUP BY (t\_year,card\_brand,gender)

SELECT t\_year, card\_brand, gender, SUM(transactions\_no) AS total\_transactions

FROM materialized\_view\_cube

GROUP BY t\_year,card\_brand,gender

Το παρακάτω query μας δίνει ακριβώς τα ίδια αποτελέσματα με το 1ο ερώτημα του ζητήματος χωρίς την χρήση κάποιου GROUP BY GROUPING SETS. (105 rows)

**ΖΗΤΗΜΑ 4**

1.

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, διάγραμμα, παράλληλα

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμμή, γραμματοσειρά

Περιγραφή που δημιουργήθηκε αυτόματα2.