2^{Ο} ΠΡΟΤΖΕΚΤ ΣΤΟ ΜΑΘΗΜΑ Σ.Δ.Α.Δ.

ΤΣΙΝΤΖΟΣ ΙΩΑΝΝΗΣ

Ρ3200211

ΖΗΤΗΜΑ 1^{Ο}

Αφού δημιουργήσαμε τη βάση με όνομα CTDW κατασκευάζουμε τον πίνακα CardsTransactions με τα ζητούμενα ορίσματα:

```
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 τον φορτώνουμε με τα δεδομένα του δοθέντος txt file:

```
BULK INSERT CardsTransactions
FROM 'C:\temp\CardsTransactions.txt'
WITH (FIRSTROW =2, FIELDTERMINATOR='|', ROWTERMINATOR = '\n');
```

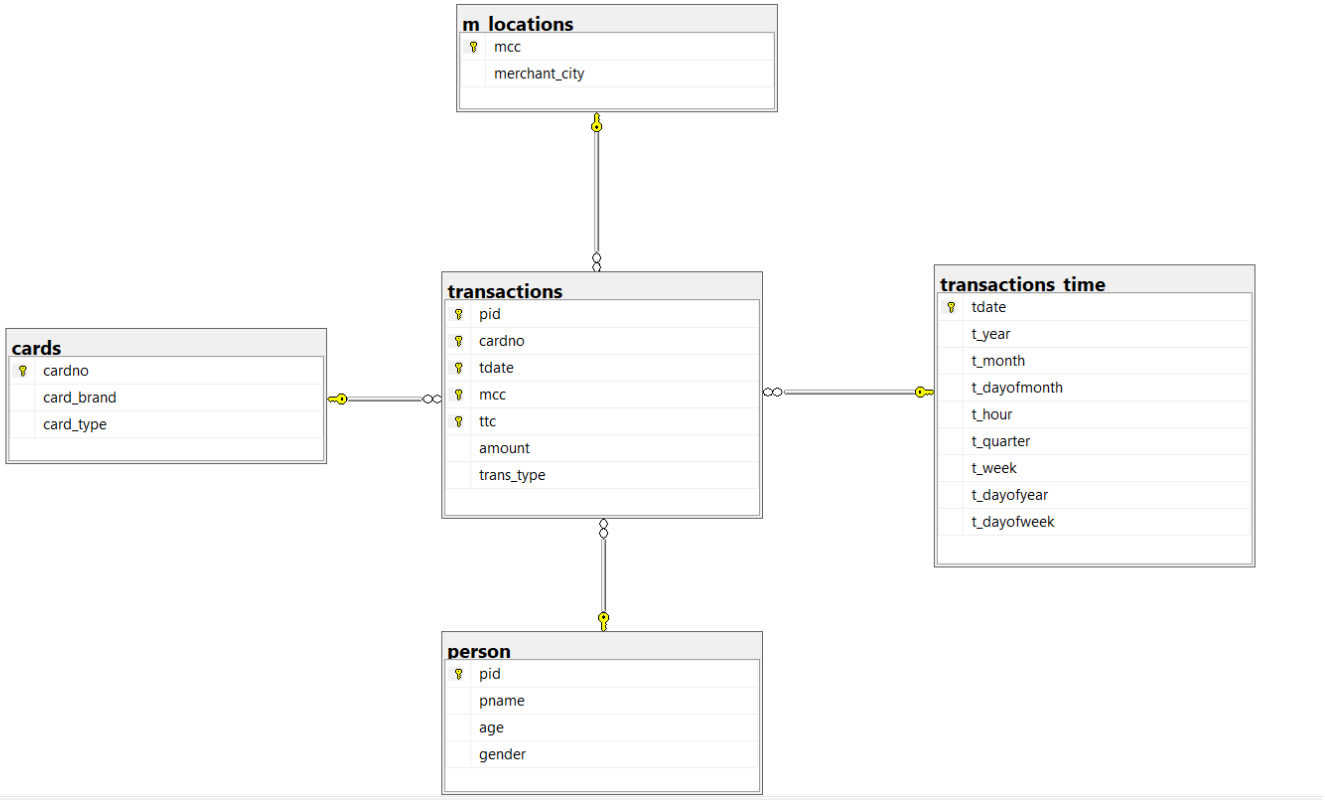
Επειτα φτιαχνουμε τους πινακες και τους τροφοδοτουμε με δεδομενα από τον CardsTransactions :

```
create table person
(pid int primary key,
pname varchar(50),
age int ,
gender char(1)
);
insert into person
	select distinct pid , pname , age , gender
		from CardsTransactions

create table cards
(cardno char(16) primary key,
card_brand varchar(30) ,
card_type varchar(20)
);
insert into cards
	select distinct cardno, card_brand , card_type
		from CardsTransactions
```

```
create table transactions_time
(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,
);

set datefirst 1;
insert transactions_time
       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


create table m_locations
(mcc int primary key,
merchant_city varchar(50)
);
insert m_locations
       select distinct mcc , merchant_city
              from CardsTransactions

create table transactions
( pid int,
cardno char(16),
tdate datetime,
mcc int,
ttc int,
amount decimal(6,2),
trans_type varchar(30),

primary key(pid,cardno,tdate,mcc,ttc),
foreign key (pid) references person(pid),
foreign key (cardno) references cards(cardno),
foreign key (tdate) references transactions_time(tdate),
foreign key (mcc) references m_locations(mcc),

);
insert transactions
       select distinct pid ,cardno ,tdate ,mcc ,amount ,ttc , trans_type
              from CardsTransactions
GROUP BY
pid , cardno , tdate , mcc , amount ,ttc, trans_type
```

ΔΙΑΓΡΑΜΜΑΤΙΚΗ ΑΝΑΠΑΡΑΣΤΑΣΗ:



ΖΗΤΗΜΑ 2Ο

1

```
SELECT m.merchant_city AS City, SUM(t.amount) AS TotalAmount
FROM transactions t
JOIN m_locations m ON t.mcc = m.mcc
GROUP BY m.merchant_city
ORDER BY City ASC;
```

2

```
SELECT DATEPART(year, t.tdate) AS Year, p.gender AS Gender, SUM(t.amount) AS TotalAmount
FROM transactions t
JOIN person p ON t.pid = p.pid
GROUP BY DATEPART(year, t.tdate), p.gender
ORDER BY Year DESC;
```

3

```
SELECT c.card_brand AS CardBrand, c.card_type AS CardType, COUNT(*) AS TransactionCount,
SUM(t.amount) AS TotalAmount
FROM transactions t
JOIN cards c ON t.cardno = c.cardno
GROUP BY c.card_brand, c.card_type;
```

4

```
SELECT DATEPART(QUARTER, tdate) AS Quarter, trans_type AS TransactionType, SUM(amount) AS
TotalAmount
FROM transactions
WHERE DATEPART(YEAR, tdate) = 2019
GROUP BY DATEPART(QUARTER, tdate), trans_type;
```

5

```sql
SELECT
    CASE
        WHEN grouping_id(t_year, gender, age_group) = 7 THEN 'Value of Online Transactions by Year,
Gender, and Age'
        WHEN grouping_id(t_year, gender) = 3 THEN 'Value of Online Transactions by Year and Gender'
        WHEN grouping_id(t_year) = 1 THEN 'Value of Online Transactions by Year'
        ELSE 'Total Value of Online Transactions'
    END AS ReportSection,
    SUM(amount) AS TotalValue,
    t_year,
    gender,
    age_group
FROM
    transactions
    LEFT JOIN person ON transactions.pid = person.pid
    LEFT JOIN transactions_time ON transactions.tdate = transactions_time.tdate
    LEFT JOIN (
        SELECT
            pid,
            CASE
                WHEN age < 18 THEN 'Under 18'
                WHEN age >= 18 AND age < 30 THEN '18-29'
                WHEN age >= 30 AND age < 40 THEN '30-39'
                WHEN age >= 40 AND age < 50 THEN '40-49'
                WHEN age >= 50 THEN '50+'
            END AS age_group
        FROM
            person
    ) AS age_table ON transactions.pid = age_table.pid
WHERE
    trans_type = 'Online Transaction'
GROUP BY
    GROUPING SETS (
        (),
        (t_year),
        (t_year, gender),
        (t_year, gender, age_group)
    )
HAVING
    GROUPING(t_year) = 1 OR (GROUPING(t_year) = 0 AND GROUPING(gender) = 1) OR (GROUPING(t_year) = 0
AND GROUPING(gender) = 0 AND GROUPING(age_group) = 0)
ORDER BY
    t_year DESC,
    gender,
    age_group
```

ZHTHMA 3Ο

1.

```
SELECT
    t_year,
    card_brand,
    gender,
    COUNT(*) AS TransactionCount
FROM
    transactions
    INNER JOIN cards ON transactions.cardno = cards.cardno
    INNER JOIN transactions_time ON transactions.tdate = transactions_time.tdate
    INNER JOIN person ON transactions.pid = person.pid
GROUP BY
    ROLLUP (t_year, card_brand, gender)
HAVING
    GROUPING(t_year) = 0
    AND GROUPING(card_brand) = 0
    AND GROUPING(gender) = 0
ORDER BY
    t_year,
    card_brand,
    gender
```

2.

```
CREATE VIEW OnlineTransactionsCubeView
AS
SELECT
    t_year,
    card_brand,
    gender,
    COUNT(*) AS TransactionCount
FROM
    transactions
    INNER JOIN cards ON transactions.cardno = cards.cardno
    INNER JOIN transactions_time ON transactions.tdate = transactions_time.tdate
    INNER JOIN person ON transactions.pid = person.pid
WHERE
    transactions.trans_type = 'Online Transaction'
GROUP BY
    t_year,
    card_brand,
    gender;
```

```
SELECT
    t_year,
    card_brand,
    gender,
    COUNT(*) AS TransactionCount
INTO #OnlineTransactionsCubeTemp
FROM
    transactions
    INNER JOIN cards ON transactions.cardno = cards.cardno
    INNER JOIN transactions_time ON transactions.tdate = transactions_time.tdate
    INNER JOIN person ON transactions.pid = person.pid
WHERE
```

```
    transactions.trans_type = 'Online Transaction'
GROUP BY
    t_year,
    card_brand,
    gender;



CREATE CLUSTERED INDEX IX_OnlineTransactionsCubeTemp
ON #OnlineTransactionsCubeTemp (t_year, card_brand, gender);
```