



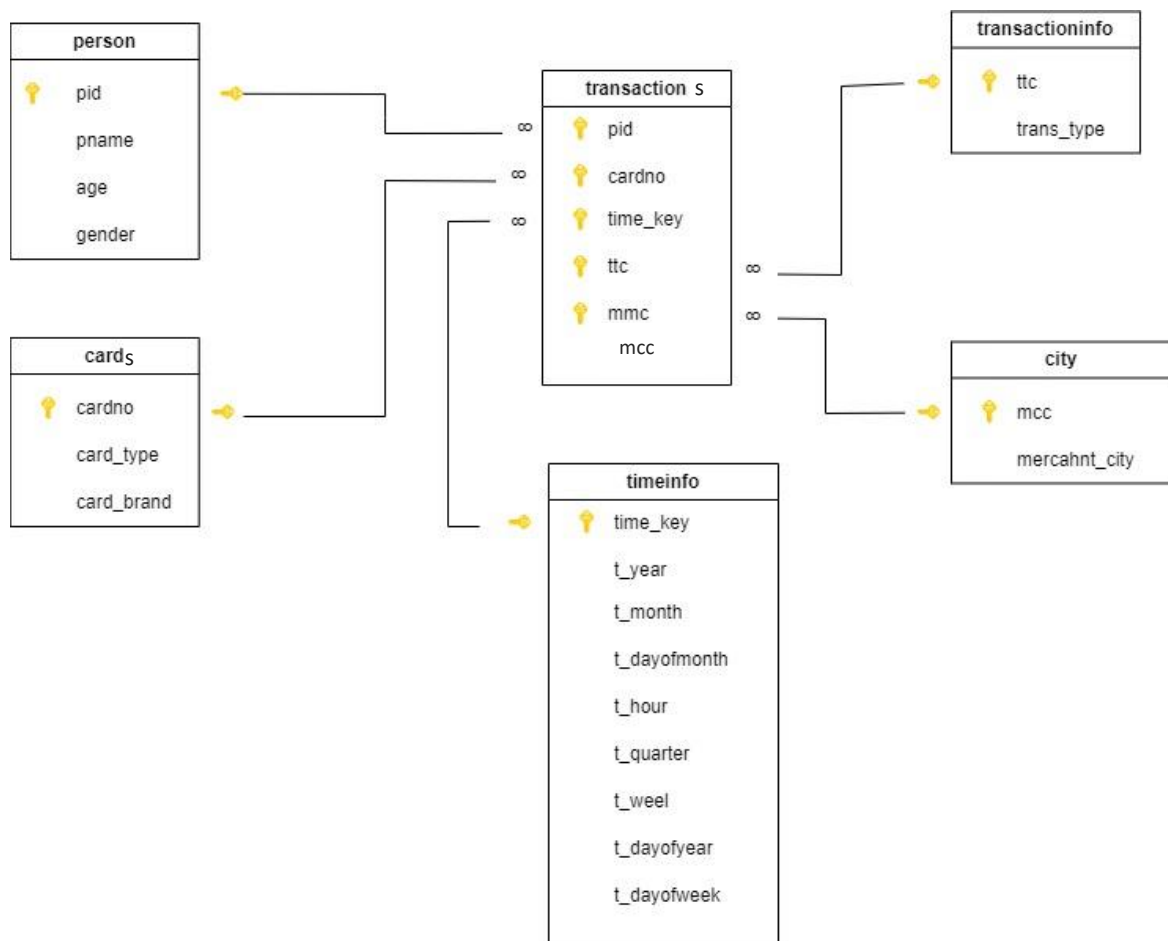
Ζήτημα 1ο

1) Για το πρώτο ερώτημα, αφού δημιουργήσαμε μια βάση δεδομένων με όνομα CTDW τρέξαμε τον εξής κώδικα:

```
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)
);
```

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

2) Το λογικό σχήμα της βάσης είναι της μορφής:





3) Οι εντολές στην sql για την δημιουργία του παραπάνω λογικού σχήματος και την τροφοδότηση των πινάκων με δεδομένα είναι οι εξής:

```
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_type varchar(20),
 card_brand varchar(30)
);
```

```
insert into cards select distinct cardno, card_type, card_brand from
CardsTransactions
```

```
create table transactioninfo
(ttc int primary key,
 trans_type varchar(30)
);
```

```
insert into transactioninfo select distinct ttc, trans_type from
CardsTransactions
```

```
create table timeinfo
(time_key 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 into timeinfo 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 order by tdate
```

```
create table city
(mcc int primary key,
 merchant_city varchar(50))
```



```
);
```

```
insert into city select distinct mcc, merchant_city from CardsTransactions
```

```
create table transactions
```

```
( pid int ,
  cardno char(16) ,
  time_key datetime,
  ttc int ,
  mcc int ,
  amount decimal(6,2)
```

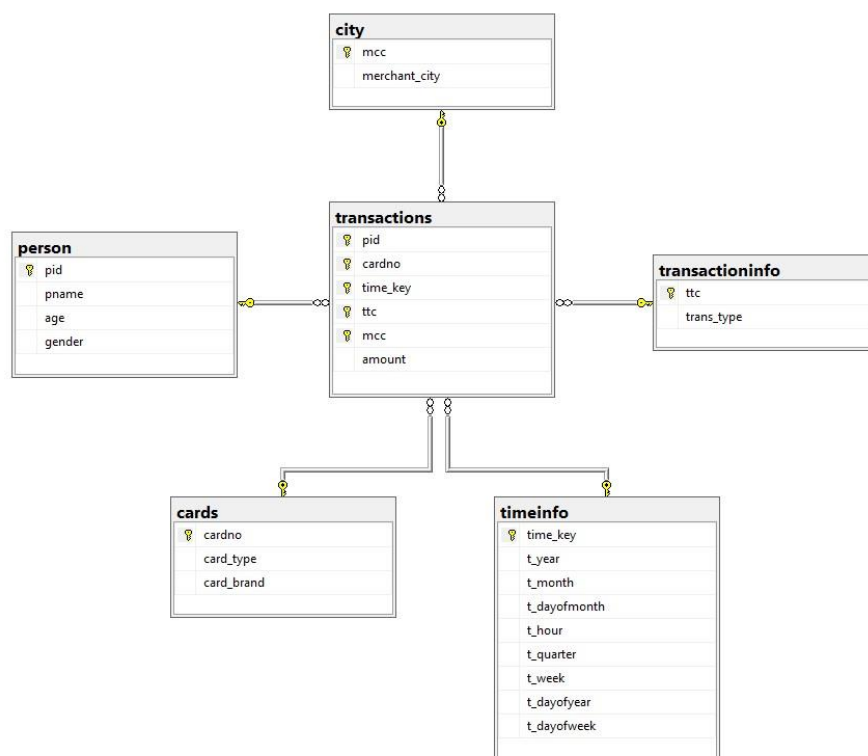
```
primary key(pid, cardno, time_key, ttc, mcc),
foreign key (pid) references person(pid),
foreign key (time_key) references timeinfo(time_key),
foreign key (cardno) references cards(cardno),
foreign key (ttc) references transactioninfo(ttc),
foreign key (mcc) references city(mcc)
);
```

```
INSERT INTO transactions
```

```
SELECT pid, cardno, tdate, ttc, mcc, sum(amount)
  from CardsTransactions
```

```
GROUP BY pid, cardno, tdate, ttc, mcc
```

4) Το σχήμα της αποθήκης χρησιμοποιώντας την επιλογή "Database diagrams" του SQL Server Management Studio είναι το εξής:





Ζήτημα 2ο

1)

Εμφανίστε έναν κατάλογο με την αξία των συναλλαγών ανά πόλη. Ο κατάλογος πρέπει να είναι ταξινομημένος με βάση την πόλη σε αύξουσα διάταξη.

```
SELECT c.merchant_city AS City, SUM(t.amount) AS TotalTransactionValue
FROM transactions t
JOIN city c ON t.mcc = c.mcc
GROUP BY c.merchant_city
ORDER BY c.merchant_city ;
```

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.

```
SELECT ti.t_year AS "Year", p.gender AS Gender, SUM(t.amount) AS
TotalTransactionValue
FROM transactions t
JOIN person p ON t.pid = p.pid
JOIN timeinfo ti ON ti.time_key = t.time_key
GROUP BY ti.t_year, p.gender
ORDER BY ti.t_year DESC;
```

3)

Εμφανίστε έναν κατάλογο με τον αριθμό και την αξία των συναλλαγών ανά επωνυμία (card_brand) είδος (card_type) κάρτας

```
SELECT c.card_brand AS Card_Brand, c.card_type AS Card_Type, COUNT(t.amount) AS
Number_of_Transactions, SUM(t.amount) AS Total_Transactions_Value
FROM
    transactions t
JOIN
    cards c ON t.cardno = c.cardno
GROUP BY
    c.card_brand,
    c.card_type;
```

4)

Εμφανίστε έναν κατάλογο με ανάλυση της αξίας των συναλλαγών ανά τύπο συναλλαγής (trans_type) σε τριμηνιαία βάση για το έτος 2019.

```
SELECT ti.t_quarter AS Quarter, tr.trans_type AS Transaction_Type,
SUM(t.amount) AS Total_TransactionValue
FROM transactions t
JOIN timeinfo ti ON t.time_key = ti.time_key
JOIN transactioninfo tr ON t.ttc = tr.ttc
WHERE ti.t_year = 2019
GROUP BY ti.t_quarter, tr.trans_type
ORDER BY ti.t_quarter, tr.trans_type;
```



5)

Η διοίκηση της τράπεζας θέλει μία αναφορά που θα περιέχει τις ακόλουθες πληροφορίες για τις online συναλλαγές (trans_type='Online Transaction').

- Την συνολική αξία των online συναλλαγών.
- Την αξία των online συναλλαγών ανά έτος.
- Την αξία των online συναλλαγών ανά έτος και φύλο.
- Την αξία των online συναλλαγών ανά έτος, φύλο και ηλικία. Γράψτε μια μόνο επερώτηση (δίχως την χρήση του τελεστή UNION) σε γλώσσα SQL η οποία να παράγει την παραπάνω αναφορά.

```
select ti.t_year as "YEAR", p.gender as GENDER, p.age as AGE, sum(t.amount) as  
Total_Amount  
from transactions t, transactioninfo tr, timeinfo ti, person p  
where tr.ttc = t.ttc AND ti.time_key = t.time_key and p.pid = t.pid and  
tr.trans_type='Online Transaction'  
  
group by ROLLUP (ti.t_year, p.gender, p.age)
```

Ζήτημα 3ο

1)

Γράψτε μια επερώτηση σε γλώσσα SQL το αποτέλεσμα της οποίας είναι η δημιουργία ενός κύβου (data cube), κάθε κελί του οποίου περιέχει τον αριθμό των συναλλαγών για έναν συγκεκριμένο συνδυασμό τιμών: έτος, επωνυμία κάρτας (card_brand) και φύλο.

```
select ti.t_year as "YEAR", c.card_brand as CARD_BRAND, p.gender as GENDER  
, COUNT(t.amount) as Number_of_Transactions  
from transactions t, timeinfo ti, cards c, person p  
where t.pid = p.pid AND ti.time_key = t.time_key AND t.cardno = c.cardno  
group by CUBE(ti.t_year, c.card_brand, p.gender)
```

2)

Θεωρείστε ότι το DBMS δεν υποστηρίζει τον τελεστή CUBE για την δημιουργία του παραπάνω κύβου ούτε την εντολή GROUP BY GROUPING SETS παρά μόνο την εντολή GROUP BY. Δημιουργήστε μια MATERIALIZED όψη (INDEXED VIEW στον SQL SERVER) η οποία θα περιέχει το αποτέλεσμα ενός μόνο GROUP BY του κύβου του ερωτήματος 1. Γράψτε κατάλληλες εντολές SQL ώστε να παράγετε τα υπόλοιπα GROUP BY του κύβου από την όψη που δημιουργήσατε.

Αρχικά, εφόσον οι διαστάσεις του κύβου είναι 3, και χρησιμοποιούμε ένα μόνο γνώρισμα για κάθε διάσταση, θα έχουμε $2^3=8$ group bys

Η όψη που θα δημιουργήσουμε θα περιέχει ένα μόνο group by και από αυτό θα μπορούν να υπολογιστούν και τα υπόλοιπα. Το συγκεκριμένο λοιπόν group by θα περιέχει το t_year, card_brand, gender.



Ο παρακάτω κώδικας δημιουργεί την materialized όψη:

```
CREATE VIEW cube_view_ask_3_
WITH SCHEMABINDING
AS
SELECT
    ti.t_year ,
    c.card_brand,
    p.gender,
    COUNT_BIG(*) AS TransactionCount
FROM
    dbo.transactions t
JOIN
    dbo.timeinfo ti ON t.time_key = ti.time_key
JOIN
    dbo.cards c ON t.cardno = c.cardno
JOIN
    dbo.person p ON t.pid = p.pid
JOIN
    dbo.transactioninfo tr ON tr.ttc = t.ttc
GROUP BY
    ti.t_year,
    c.card_brand,
    p.gender;
```

```
CREATE UNIQUE CLUSTERED INDEX index_on_view
ON cube_view_ask_3_ (t_year, card_brand, gender);
```

Έπειτα, εκτελούμε τα υπόλοιπα group bys (7 στο πλήθος) που έχουν μείνει ως εξής:

Για το t_year, card_brand

```
SELECT
    t_year as "YEAR",
    card_brand AS CardBrand,
    SUM(TransactionCount) AS TotalTransactionCount
FROM
    dbo.cube_View_ask_3_
GROUP BY
    t_year, card_brand;
```

Για το t_year, gender

```
SELECT
    t_year as "YEAR",
    gender AS Gender,
    SUM(TransactionCount) AS TotalTransactionCount
FROM
    dbo.cube_View_ask_3_
GROUP BY
    t_year, gender;
```



Για το card_brand, gender

```
SELECT
    card_brand as Card_Brand,
    gender as Gender,
    SUM(TransactionCount) AS TotalTransactionCount
FROM
    dbo.cube_View_ask_3_
GROUP BY
    card_brand, gender;
```

Για το t_year

```
SELECT
    t_year as "YEAR",
    SUM(TransactionCount) AS TotalTransactionCount
FROM
    dbo.cube_View_ask_3_
GROUP BY
    t_year
```

Για το card_brand

```
SELECT
    card_brand as Card_Brand,
    SUM(TransactionCount) AS TotalTransactionCount
FROM
    dbo.cube_View_ask_3_
GROUP BY
    card_brand
```

Για το gender

```
SELECT
    gender as Gender, SUM(TransactionCount) AS TotalTransactionCount
FROM
    dbo.cube_View_ask_3_
GROUP BY gender
```

none

```
SELECT
    SUM(TransactionCount) AS TotalTransactionCount
FROM
    dbo.cube_View_ask_3_
```

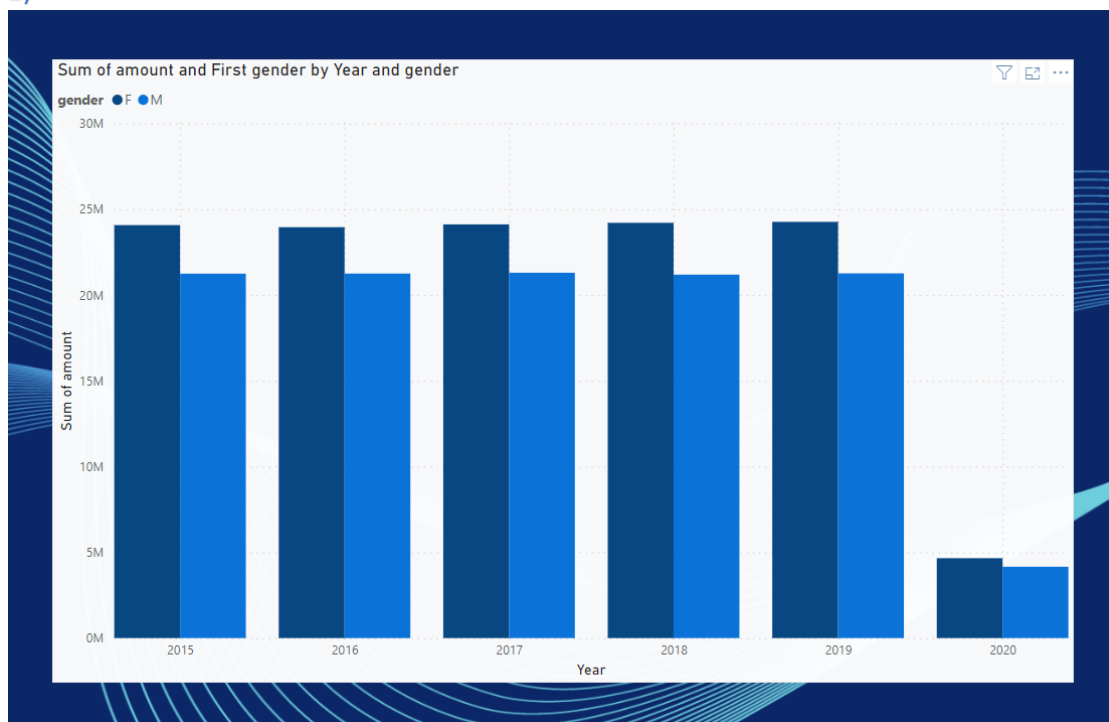


καθώς και το *group by* για την ίδια την όψη (που είναι ένα *select* ερώτημα), δηλαδή *t_year*, *card_brand*, *gender*

```
SELECT
  t_year as "YEAR",
  card_brand AS CardBrand,
  gender AS Gender,
  TransactionCount AS TotalTransactionCount
FROM
  dbo.cube_View_ask_3_
```

Ζήτημα 4°

1)





2)

