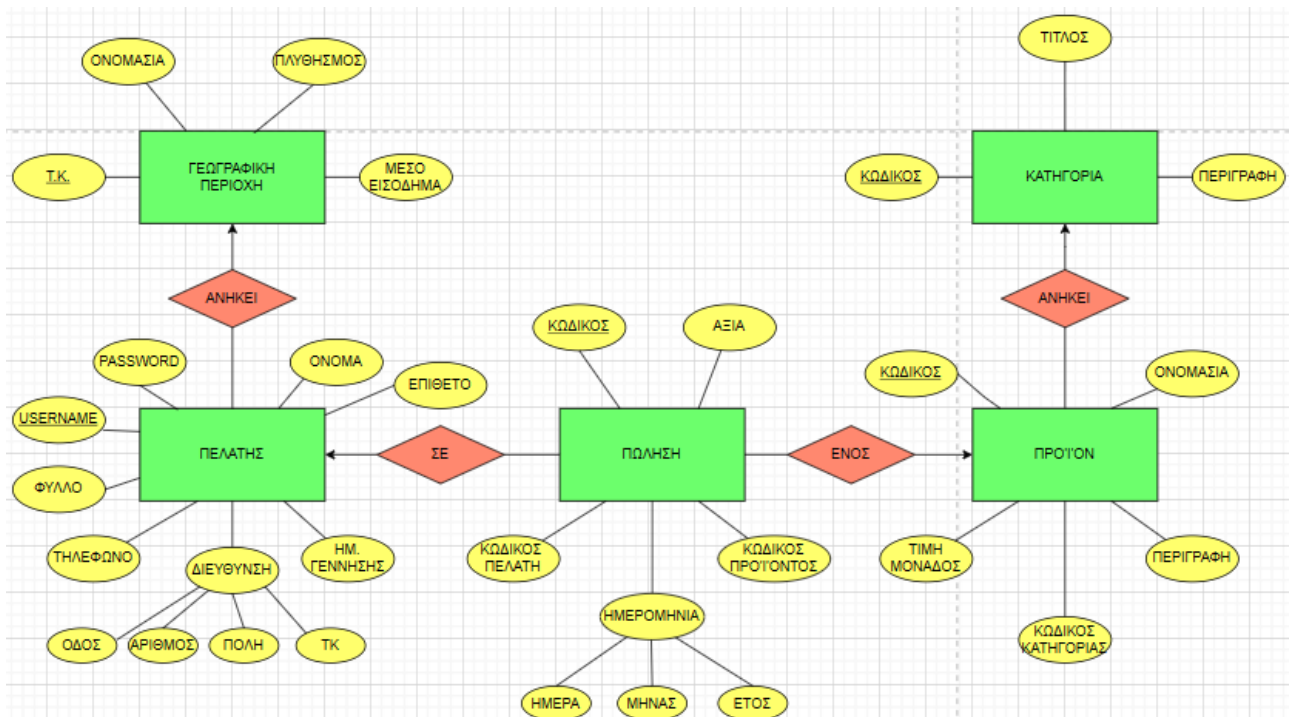


**ΕΡΓΑΣΙΑ 1 ΓΙΑ ΤΟ ΜΑΘΗΜΑ
ΔΙΑΧΕΙΡΙΣΗ ΔΕΔΟΜΕΝΩΝ, ΕΠΙΧΕΙΡΗΜΑΤΙΚΗ ΕΥΦΥΪΑ
και ΟΠΤΙΚΟΠΟΙΗΣΗ**

Παναγιώτης Πετρόχειλος

Ερώτημα 2



Ερώτημα 3

Σχεσιακό μοντέλο :

CUSTOMER (USERNAME, PASSWORD, NAME, SURNAME, BIRTHDATE, PHONE, SEX, CITY, STREET, NUMBER, ZIP)

GEOAREA (ZIP, NAME, POPULATION, AVINCOME)

SALE (CODE, VALUE, DATE, PCODE, USERNAME)

PRODUCT (PCODE, NAME, DESCRIPTION, UNIT PRICE, CATCODE)

CATEGORY (CATCODE, NAME, DESCRIPTION)

Δημιουργία βάσης δεδομένων :

```
create database etairia1;
```

```
<execute>
```

```
use etairia1;
```

```
CREATE TABLE geoarea (
    ZIP int not null primary key,
    name nvarchar(60) not null,
    population int not null CHECK (population > 0),
    avincome money not null CHECK (avincome > 0),
);
CREATE TABLE customer (
    username nvarchar(60) not null primary key,
    password nvarchar(60) not null unique,
    name nvarchar(60) not null,
    surname nvarchar(60) not null,
    sex nvarchar(10) not null,
    birthdate date,
    phone nvarchar(60) not null,
    city nvarchar(60),
    street nvarchar(60),
    number int,
    ZIP int foreign key references geoarea,
);
CREATE TABLE category (
    catcode int IDENTITY(1,1) not null primary key,
    name nvarchar(60) not null,
    description nvarchar(255),
);
CREATE TABLE product (
    pcode int IDENTITY(1,1) not null primary key,
    name nvarchar(60) not null,
    description nvarchar(255),
    unit_price money not null CHECK (unit_price > 0),
    catcode int foreign key references category,
);
CREATE TABLE sale (
    code int IDENTITY(1,1) primary key,
    value money not null,
    [date] date not null,
    pcode int not null foreign key references product,
    username nvarchar(60) not null foreign key references customer,
);
```

Ερώτημα 4

a.

```
select code, date
from sale
where [date] >= '2019-1-12' and [date] <= '2019-2-12'
```

b.

```
select code
from sale JOIN customer ON sale.username = customer.username
where DATEDIFF(year, customer.birthdate, GETDATE ()) > 40
```

C.

```
select c.name, surname, phone, s.date
from customer as c join geoarea as g on c.ZIP = g.ZIP join sale as s on c.username =
s.username
where g.avincome > 40000 and s.[date] = '2019-12-25'
```

D.

```
select month(date) as [Month], sum(value) as Value
from sale
where year(date) = 2019
group by month(date)
order by Month
```

E.

```
select product.name, sum(value) as Value
from sale join product on product.pcode = sale.pcode
where year(date) = 2019
group by product.name
order by product.name
```

F.

```
select customer.sex, customer.ZIP, avg(value) as Value
from sale join customer on customer.username = sale.username
group by customer.sex, customer.ZIP
order by customer.ZIP, customer.sex
```

G.

```
select sum(value) as Value
from sale join customer on customer.username = sale.username join geoarea on customer.ZIP =
geoarea.ZIP
where geoarea.avincome > 50000
```

H.

```
select m2019.Month, m2018.sales2018, m2019.sales2019, (sales2019/sales2018)*100 as
percentage
from
    (select DATEPART(month, date) as [Month], sum(value) as sales2019 from sale where
    year(date) = 2019
    group by DATEPART(month, date)) as m2019
,
    (select DATEPART(month, date) as [Month], sum(value) as sales2018 from sale where
    year(date) = 2018
    group by DATEPART(month, date)) as m2018
where m2018.Month = m2019.Month
order by m2019.Month
```

I.

1.

```
create view salespercatergory(category, name, sales) as
select c.catcode, c.name, sum(value) as sales
from category as c join product as p on c.catcode = p.catcode join sale as s on s.pcode =
p.pcode
group by c.catcode, c.name
```

<execute>

2.

```
create view salestotal(total) as
select sum(value) as total
from sale
```

<execute>

3.

```
select category, name, (sales/total)*100 as percentage
from salespercatery, salestotal
order by category
```