

# E-properties

---

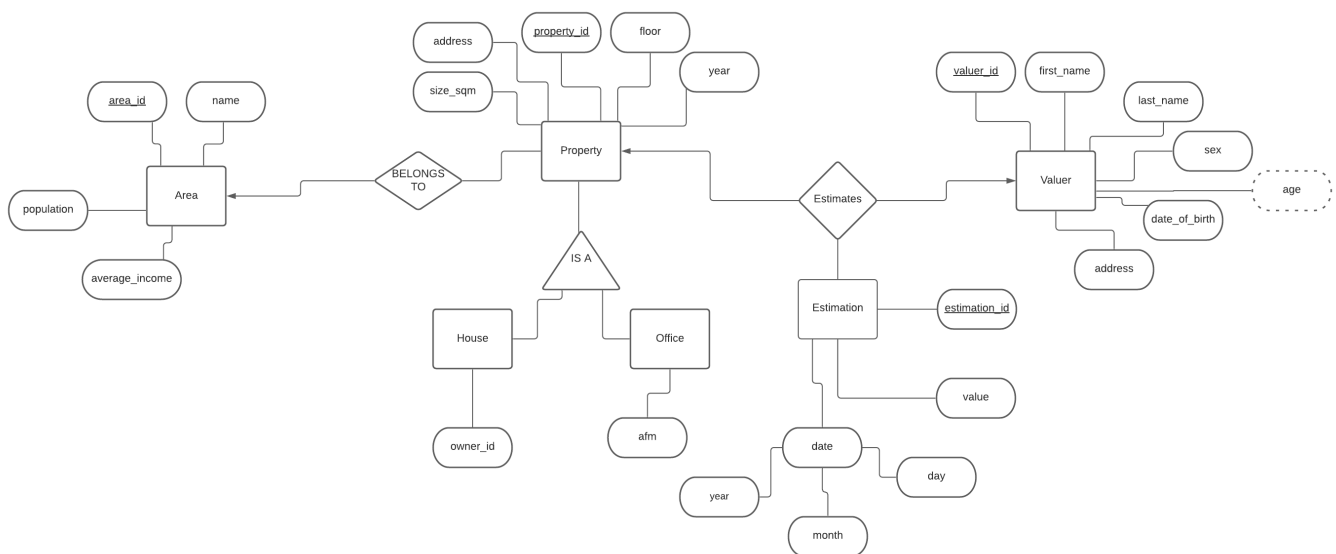
## Authors:

---

Ioannis Dekoulakos - p2822110@aueb.gr

Anastasios Moraitis - p2822124@aueb.gr

2)



Brief E-R diagram description:

- A property is associated with at most one area via belongs to, an area is associated with several (including 0) properties via belongs to.
- Entities House and Office are specializations of the Property entity. They derive the attributes of Property entity.
- An Estimation is associated with at most one Property via estimates, a Property is associated with several (including 0) Estimations via estimates.
- An Estimation is associated with at most one Valuer via estimates, a Valuer is associated with several (including 0) Estimations via estimates.

3)

Valuer(valuer\_id, first\_name, last\_name, sex, age, date\_of\_birth, address)

Estimation(estimation\_id, property\_id, valuer\_id, value, date)

Property(property\_id, address, floor, size\_sqm, constructed\_year, area\_id)

Office(property\_id, afm)

Home(property\_id, owner\_id)

Area(area\_id, name, population, average\_income)

Brief description of schema:

- Area has a primary key named area\_id. An area can be identified by its area\_id.
- Property has a primary key named property\_id. A property can be identified by its property\_id. It also has a foreign key on table area on area\_id.
- Home has a primary key named property\_id, which is also foreign key in table property. Office has a primary key named property\_id, which is also foreign key in table property.
- Valuer has a primary key named valuer\_id. A valuer can be identified by its valuer\_id.
- Estimation has a primary key named estimation\_id. An estimation can be identified by its estimation\_id. It also has a foreign key on table property on property\_id and on table valuer on valuer\_id.

## Create Schema

```
CREATE SCHEMA 'e_properties';
```

```
-- Create the various tables based on the schema.  
-- We have used auto increment for the primary keys on each table. We don't need  
-- to know their value when we insert a new entry to them. They are also not null.  
create table area  
(area_id int not null primary key auto_increment,  
 name varchar(50),
```

```

    population int,
    -- defining average income of an area as a decimal with 2 digits on the decimal
    part.
    -- that level of granularity is enough for our application.
    average_income decimal(10,2)
);

```

```

CREATE TABLE property (
    property_id int not null primary key auto_increment,
    address varchar(60),
    floor int,
    size_sqm int,
    constructed_year int,
    area_id int not null,

    FOREIGN KEY (area_id)
        REFERENCES area(area_id)
);

```

```

CREATE TABLE office (
    property_id int not null primary key,
    afm int,

    foreign key (property_id)
        references property(property_id)
);

```

```

CREATE TABLE home (
    property_id int not null primary key,
    -- owner_id has type varchar, because a person's identity can have
    alphanumeric symbols.
    owner_id varchar(15),

    foreign key (property_id)
        references property(property_id)
);

```

```

CREATE TABLE valuer (
    valuer_id int not null primary key auto_increment,
    first_name varchar(50) not null,
    last_name varchar(50) not null,
    sex char,
    -- represents the age of the valuer/it can be calculated from this field.
    date_of_birth date not null,
    address varchar(100)
);

```

```

create table Estimation
(estimation_id int not null primary key auto_increment,
    date date,

    property_id int not null,

    valuer_id int not null,

```

```

value int not null,
foreign key (property_id) references property(property_id),
foreign key (valuer_id) references valuer(valuer_id)
);

```

- Fill Data

```

INSERT INTO area (name,population,average_income) VALUES("New
York",18713220,20000.00), ("Los Angeles",12750807,70000.00),
("Chicago",8604203,10000.00), ("Miami",1000000,11000.00),
("Dallas",15000,120000.00);

INSERT INTO property (address,floor,size_sqm,constructed_year,area_id)
VALUES("7793 Old York St.?",2,100,2000,1), ("83 Bishop Circle?",1,86,2001,2),
("9257 Elizabeth Rd.?",4,150,2005,3), ("398 South Illinois Court?",6,77,1998,4),
("142 King Street?",2,52,2010,5), ("927 Golden Star Street?",3,97,2015,3), ("8199
S. Center Street",1,103,1999,2), ("7293 NW. Indian Summer Dr",7,150,2003,5), ("978
Morris St",4,180,2008,5), ("881 N. Coffee Road?",5,195,1997,1);

INSERT INTO valuer (first_name,last_name,sex,date_of_birth,address)
VALUES("Constance","Mcdowell", "M","1970-01-01","463B Southampton Dr.?"),
("Nataniel","Hanson","F","1980-01-01","9124 Princess Drive"),
("Tazmin","Walls","F","1965-01-01","684 E. Prince Ave"),
("Joe","Espinosa","F","1975-01-01","416 East Circle Court?"),
("Nazim","Villegas","M","1983-01-01","56 Lafayette Street"), ("Randal","l
Hayden","M","1956-01-01","8488 Edgemont St.?"), ("Geraldine","O Moore","F","1977-
01-01","440 New Road?"), ("Arjun","Mccall","M","1995-01-01","5 Cambridge Dr."),
("Pia","Steele","F","1994-01-01","9043 Cross St.?"),
("Gordon","Sharples","M","1981-01-01","9635 E. Sycamore Street?"),
("Rachelle","Mill","F","1974-01-01","41 E. Brickyard St.?");

INSERT INTO Estimation (date,property_id,valuer_id,value) VALUES("2008-11-
01",1,1,150000), ("2009-01-02",2,2,70000), ("2010-03-03",3,3,100000), ("2011-03-
04",4,4,1000000), ("2020-12-04",5,5,85000), ("2013-03-05",6,6,105000), ("2014-05-
06",7,7,300000), ("2015-03-07",8,8,200000), ("2016-07-07",9,9,1500000), ("2019-10-
08",10,10,50000), ("2020-12-30",4,1,87000), ("2020-12-18",5,3,104000), ("2020-12-
27",2,5,344000), ("2021-04-28",10,7,900000), ("2019-12-28",8,9,98000);

INSERT INTO home (property_id,owner_id) VALUES(2,"A0458793"), (4,"AI787456"),
(6,"AT748596"), (8,"AM456789"), (10,"AM656554");

INSERT INTO office (property_id,afm) VALUES(1,56203756), (3,62141545),
(5,165956896), (7,159119659), (9,159119660);

```

4)

- a)

```

select distinct(property.property_id), property.address, Estimation.estimate_id,
area.area_id
FROM property, Estimation, area
WHERE area.area_id = property.area_id and Estimation.property_id =
property.property_id
and area.average_income > 40000
and estimation.date BETWEEN '2020-12-24' AND '2020-12-31';

```

property_id	address	estimate_id	area_id
2	83 Bishop Circle?	13	2

- b)

```

CREATE VIEW Estimations2020 as
SELECT Estimation.estimate_id, Estimation.valuer_id FROM Estimation
WHERE YEAR(Estimation.date) = 2020;

select valuer.valuer_id, count(Estimations2020.estimate_id) AS 'Estimates in
2020'
from valuer
left join Estimations2020 on (valuer.valuer_id = Estimations2020.valuer_id)
GROUP BY valuer.valuer_id;

```

valuer_id	Estimates in 2020
1	1
2	0
3	1
4	0
5	2
6	0
7	0
8	0
9	0
10	0
11	0

- c)

```
select property.property_id
from property, Estimation
where property.property_id = Estimation.property_id
and YEAR(Estimation.date) = 2020
GROUP BY property.property_id
HAVING count(Estimation.estimate_id) >= 2
```

### property\_id

5

- d)

```
SELECT Estimation.estimate_id
FROM Estimation, property as p
WHERE Estimation.property_id = p.property_id
AND p.property_id in (SELECT property.property_id
                      FROM area, property
                      WHERE area.area_id = property.area_id
                      AND area.average_income > 25000);
```

### estimate\_id

2

13

7

5

12

8

15

9

- e)

```
SELECT COUNT(Estimation.estimate_id)
FROM Estimation, property, area
WHERE Estimation.property_id = property.property_id
AND property.area_id = area.area_id
AND YEAR(Estimation.date) = 2020
AND area.population > 50000;
```

## Total Estimations in 2020 for popul >50K

2

- f)

```
SELECT area.area_id, AVG(Estimation.value/property.size_sqm) AS AVERAGE_ESTIMATION
FROM area, property, Estimation
WHERE Estimation.property_id = property.property_id
AND property.area_id = area.area_id
GROUP BY area.area_id
ORDER BY AVERAGE_ESTIMATION;
```

area_id	AVERAGE_ESTIMATION
3	874.57044674
1	2123.93162393
2	2575.52494920
5	2790.92307692
4	7058.44155844

- g)

```
create view office_est2020 AS
SELECT estimation.valuer_id , count(Estimation.estimate_id) as ocount
FROM Estimation,office
where estimation.property_id=office.property_id and (YEAR(Estimation.date) = 2020)
group by estimation.valuer_id;

create view home_est2020 AS
SELECT estimation.valuer_id , count(Estimation.estimate_id) as hcount
FROM Estimation,home
where estimation.property_id=home.property_id and (YEAR(Estimation.date) = 2020)
group by estimation.valuer_id;

select valuer.valuer_id, office_est2020.ocount,home_est2020.hcount
from valuer
left join office_est2020 on (valuer.valuer_id = office_est2020.valuer_id)
left join home_est2020 on (valuer.valuer_id = home_est2020.valuer_id);
```

valuer_id	ocount	hcount
1	null	1
2	null	null

valuer_id	ocount	hcount
3	1	null
4	null	null
5	1	1
6	null	null
7	null	null
8	null	null
9	null	null
10	null	null
11	null	null

- h)

```

CREATE VIEW average19 AS
SELECT area.area_id, AVG(Estimation.value/property.size_sqm) AS AVERAGE_ESTIMATION
FROM area, property, Estimation
WHERE Estimation.property_id = property.property_id
AND property.area_id = area.area_id
AND YEAR(Estimation.date) = 2019
GROUP BY area.area_id;

CREATE VIEW average20 AS
SELECT area.area_id, AVG(Estimation.value/property.size_sqm) AS AVERAGE_ESTIMATION
FROM area, property, Estimation
WHERE Estimation.property_id = property.property_id
AND property.area_id = area.area_id
AND YEAR(Estimation.date) = 2020
GROUP BY area.area_id;

SELECT area.area_id,
(CASE
    WHEN average19.AVERAGE_ESTIMATION IS NULL AND average20.AVERAGE_ESTIMATION
IS NULL
        THEN "No Estimations for those 2 years"
    WHEN average19.AVERAGE_ESTIMATION IS NULL
        THEN "No Estimations for 2019"
    WHEN average20.AVERAGE_ESTIMATION IS NULL
        THEN 0
    ELSE
        ((average20.AVERAGE_ESTIMATION -
average19.AVERAGE_ESTIMATION)/average19.AVERAGE_ESTIMATION)
END) AS "Change"
FROM area
LEFT JOIN average19 ON (average19.area_id = area.area_id)
LEFT JOIN average20 ON (average20.area_id = area.area_id);

```



area_id	Change
1	0
2	No Estimations for 2019
3	No Estimations for those 2 years
4	No Estimations for 2019
5	1.781593406611

- i)

```

CREATE VIEW totalPopulation as
SELECT SUM(area.population) as totalPopulation from area;

-- 2nd column Views
CREATE VIEW areaEstimates20 as
SELECT area.area_id, Count(Estimation.estimate_id) as totalEstimates
from area, property, Estimation
WHERE Estimation.property_id = property.property_id
AND property.area_id = area.area_id
AND YEAR(Estimation.date) = 2020
group by area.area_id;

CREATE VIEW totalEstimations2020 as
SELECT COUNT(Estimation.estimate_id) as total from Estimation
WHERE YEAR(Estimation.date) = 2020;

CREATE VIEW popPercent as
(SELECT area.area_id, (area.population/tp.totalPopulation*100) as areaPopPercent
FROM area, totalPopulation as tp);

-- Answer

SELECT area.area_id, (areaEs2020.totalEstimates/totalEs2020.total*100) as
area_pertotalest2020, popPercent.areaPopPercent
FROM area
INNER JOIN totalEstimations2020 as totalEs2020
LEFT JOIN areaEstimates20 as areaEs2020 on (area.area_id = areaEs2020.area_id)
LEFT JOIN popPercent on (popPercent.area_id = area.area_id);

```

area_id	area_pertotalest2020	areaPopPercent
1	null	45.5495
2	25.0000	31.0365
3	null	20.9433
4	25.0000	2.4341

area_id	area_pertotalest2020	areaPopPercent
5	50.0000	0.0365

5)

Printing the result of 4)i) with the help of python and pandas library. Notice that we are using views, defined in 4)i), which do not make use of GROUP BY and other aggregate operations.

```
# pip3 install mysql-connector-python
# pip3 install pandas --no-build-isolation
import mysql.connector
import pandas as pd
from pandas.core.indexing import need_slice

mydb = mysql.connector.connect(
    host="localhost",
    user="eproperties_clientuser",
    password="eproperties_clientuser",
    database="e_properties"
)

##
## Total population
##
mycursor = mydb.cursor()

mycursor.execute("SELECT totalPopulation FROM totalPopulation")

myresult = mycursor.fetchone()
print(myresult[0])

mycursor.close()

##
## Total Estimations 2020
##

mycursor = mydb.cursor()

mycursor.execute("SELECT total FROM totalEstimations2020")

totalEstimations2020 = mycursor.fetchone()[0]
print(myresult[0])

mycursor.close()

##
## Result set of areas and estimations for 2020
##

mycursor = mydb.cursor()
```

```

mycursor.execute("""SELECT area.area_id, Estimation.estimate_id
from area, property, Estimation
WHERE Estimation.property_id = property.property_id
AND property.area_id = area.area_id
AND YEAR(Estimation.date) = 2020""")

myresult = mycursor.fetchall()

df = pd.DataFrame(myresult, columns=['area_id', 'totalEstimates2020'])

estimationsPerArea2020 = df.groupby(['area_id']).count()
print(estimationsPerArea2020)

mycursor.close()

##
## Result set of areas and population percentage
##

mycursor = mydb.cursor()

mycursor.execute("SELECT area_id, areaPopPercent FROM popPercent")

# get data and close connections to server
myresult = mycursor.fetchall()
mycursor.close()
mydb.close()

# import fetched data into a data frame and join the totalEstimates for 2020 per
area,
# then apply the calculation for each row of the merged column.
df = pd.DataFrame(myresult, columns=['area_id', 'areaPopPercent'])

resultset = pd.merge(df, estimationsPerArea2020, on=['area_id'], how='outer')
resultset["totalEstimates2020"] =
resultset["totalEstimates2020"]/totalEstimations2020* 100
print(resultset)

```