



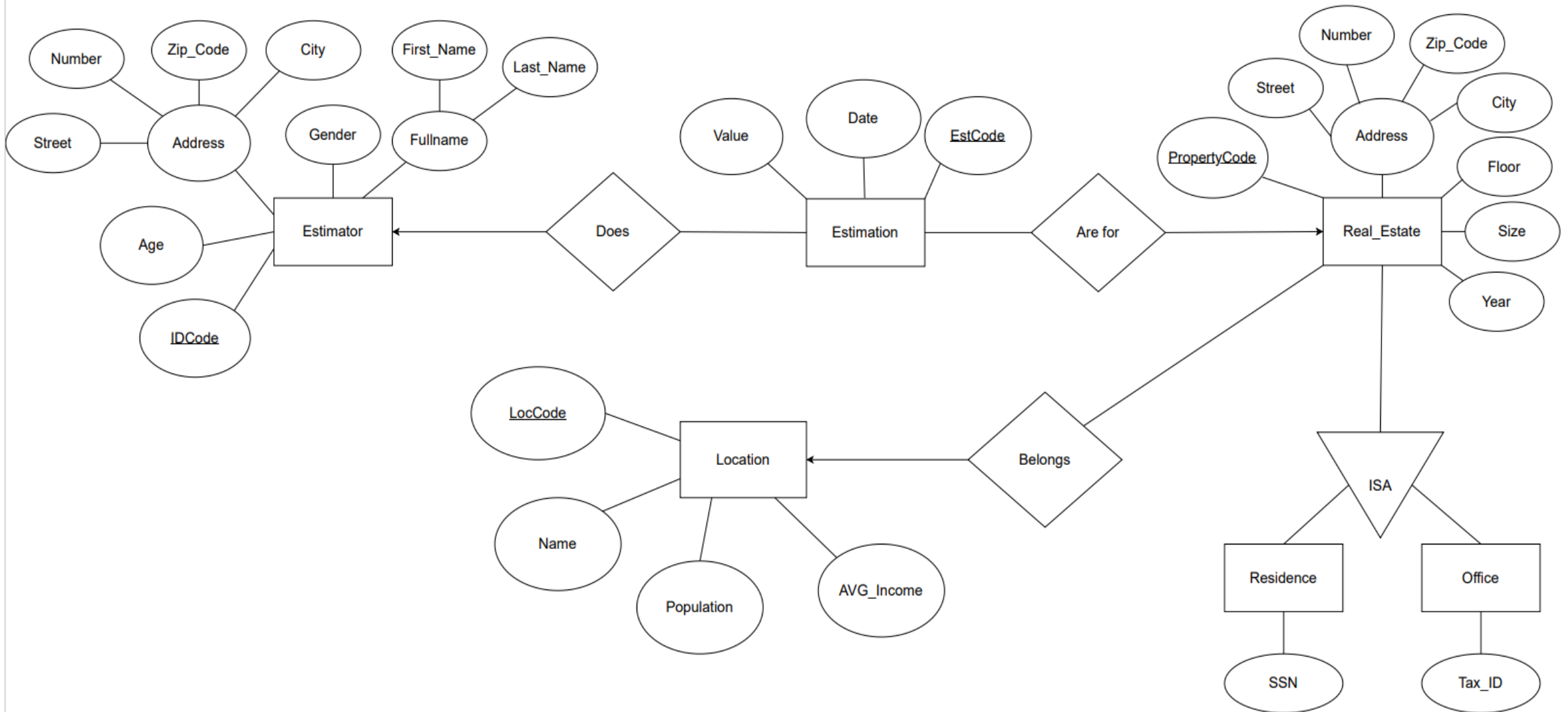
Data Management & Business Intelligence Assignment 1

Konstantinos Ninas

John Vaniotis



E-PROPERTIES ENTITY RELATIONSHIP DIAGRAM



Ερώτημα 2 - Relational Model

Estimator (IDCode, First_Name, Last_Name, Age, Gender, Street, Number, Zip_Code, City)

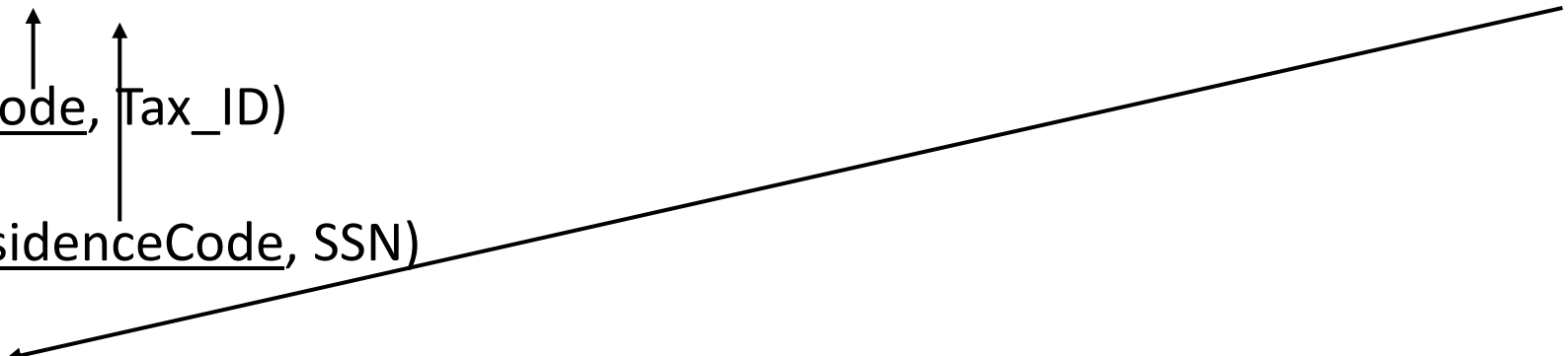
Estimation (EstCode, Value, Date, ID_Code, EstPropertyCode)

Real_Estate (PropertyCode, Street, Number, Zip_Code, City, Floor, Size, Year, LocCode)

Office (OfficeCode, Tax_ID)

Residence (ResidenceCode, SSN)

Location (LocCode, Name, Population, AVG_Income)



Ερώτημα 3 - Tables' Create & Insert Statements

```
CREATE TABLE `estimator` (  
  `IDCode` VARCHAR(10) NOT NULL  
  , `First_Name` VARCHAR(50) NULL  
  , `Last_Name` VARCHAR(50) NULL  
  , `Age` INT NULL  
  , `Gender` VARCHAR(10) NOT NULL CHECK (Gender in ('Male','Female'))  
  , `Street` VARCHAR(50) NULL  
  , `Number` INT NULL  
  , `Zip_Code` INT NULL  
  , `City` VARCHAR(50) NULL  
  , PRIMARY KEY (`IDCode`)  
  , UNIQUE INDEX `IDCode_UNIQUE` (`IDCode` ASC) VISIBLE);
```

IDCode	First_Name	Last_Name	Age	Gender	Street	Number	Zip_Code	City
EST-1	Leopold	Owens	61	Male	Doukissis Plakentias	12	41245	Athens
EST-10	Marta	Lawrence	32	Female	Ionion Nision	23	23476	Athens
EST-2	Amy	Whitehouse	53	Female	Grove Street	5	24123	New York
EST-3	Tom	Hardy	54	Male	Baker	24	12512	Berlin
EST-4	Harry	Potter	69	Male	Kifisias	21	26512	Milan
EST-5	Steve	Rodgers	28	Male	Victory Road	54	46356	Roma
EST-6	Youta	Molina	59	Female	Athinon	6	34321	Paris
EST-7	Misha	Mendez	56	Female	Liosion	89	32679	Athens
EST-8	Puck	Lloyd	27	Male	Faneromenis	86	90789	Athens
EST-9	Zoro	Roronoa	31	Male	Alkamenous	53	23467	Athens
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
INSERT INTO `estimator` (IDCode,First_Name,Last_Name,Age,Gender,Street,Number,Zip_Code,City)  
VALUES ('EST-1', 'Leopold', 'Owens', 61, 'Male', 'Doukissis Plakentias', 12, 41245, 'Athens');
```

Ερώτημα 3 - Tables' Create & Insert Statements

```
CREATE TABLE `location` (  
  `LocCode` varchar(10) NOT NULL,  
  `Name` varchar(50) DEFAULT NULL,  
  `Population` int DEFAULT NULL,  
  `AVG_Income` float DEFAULT NULL,  
  PRIMARY KEY (`LocCode`)  
);
```

```
INSERT INTO `location`(LocCode, Name, Population, AVG_Income)  
VALUES ('Loc-1', 'Ampelokipoi', 55000,45000);
```

LocCode	Name	Population	AVG_Inc
Loc-1	Ampelokipoi	55000	45000
Loc-10	Kiato	50000	25000
Loc-2	Stomio	40000	54000
Loc-3	Doliana	70000	20000
Loc-4	Kipseli	100000	15000
Loc-5	Halkias	52000	35000
Loc-6	Filiatra	35000	30000
Loc-7	Zoniana	50000	40000
Loc-8	Kalamaria	70000	42000
Loc-9	Domokos	30000	31500

Ερώτημα 3 - Tables' Create & Insert Statements

```
CREATE TABLE `real_estate` (  
  `PropertyCode` VARCHAR(10) NOT NULL  
  , `Street` VARCHAR(50) NULL  
  , `Number` INT NULL  
  , `Zip_Code` INT NULL  
  , `City` VARCHAR(50) NULL  
  , `Floor` INT NULL  
  , `Size` FLOAT NULL  
  , `Year` INT NULL  
  , `LocCode` VARCHAR(10) NULL  
  , PRIMARY KEY (`PropertyCode`)  
  , INDEX `LocCode_idx` (`LocCode` ASC) VISIBLE  
  , CONSTRAINT `LocCode` FOREIGN KEY (`LocCode`) REFERENCES `location` (`LocCode`)  
  ON DELETE NO ACTION ON UPDATE NO ACTION);  
  
INSERT INTO `real_estate` (PropertyCode,Street,Number,Zip_Code,City,Floor,Size,Year,LocCode)  
VALUES ('Prop-1', 'Agias Theklas', 152,12315, 'Larissa',2,100, 2001,'Loc-2');
```

PropertyCode	Street	Number	Zip_Code	City	Floor	Size	Year	LocCode
Prop-1	Agias Theklas	152	12315	Larissa	2	100	2001	Loc-2
Prop-10	Georgias	75	97822	Thessaloniki	4	96	1990	Loc-8
Prop-11	Evelpidon	7	49771	Athens	2	80	1962	Loc-4
Prop-12	Kipselis	32	28978	Athens	3	96	1975	Loc-4
Prop-13	Kifisias	22	15612	Athens	3	107	2012	Loc-4
Prop-14	Ifaistou	21	21576	Lamia	1	115	2009	Loc-9
Prop-15	Marathonos	51	67322	Korinthos	6	130	1987	Loc-10
Prop-16	Doukissis Plakentias	10	26782	Kalamata	2	50	1999	Loc-6
Prop-17	Anapauseos	11	51246	Kalamata	5	60	2006	Loc-5
Prop-18	Geraka	51	21561	Ioannina	1	73	2011	Loc-3
Prop-19	Lokridos	51	58633	Korinthos	0	76	2021	Loc-10
Prop-2	Agion Anarguron	21	15461	Ioannina	5	80	1999	Loc-3
Prop-20	Alkamenous	25	72098	Thessaloniki	3	45	2014	Loc-1
Prop-3	Aleksandras	65	12615	Athens	2	120	2015	Loc-4
Prop-4	Kifisias	8	15612	Athens	2	67	1985	Loc-4
Prop-5	Patission	9	15092	Thessaloniki	3	51	1997	Loc-1
Prop-6	Panepistimiou	25	51093	Larissa	0	160	2013	Loc-2
Prop-7	Athinon	73	51987	Kalamata	1	88	2004	Loc-5
Prop-8	Aigaiou	6	52396	Crete	2	75	2001	Loc-7

Ερώτημα 3 - Tables' Create & Insert Statements

```
CREATE TABLE `office` (  
  `OfficeCode` VARCHAR(10) NOT NULL  
  , `TAX_ID` VARCHAR(20) NULL  
  , PRIMARY KEY (`OfficeCode`)  
  , CONSTRAINT `OfficeCode` FOREIGN KEY (`OfficeCode`)  
    REFERENCES `real_estate` (`PropertyCode`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION);  
  
INSERT INTO `office` (OfficeCode,TAX_ID)  
  
VALUES ('Prop-2', 1512512);
```

OfficeCode	TAX_ID
Prop-10	6321241
Prop-15	4500912
Prop-16	8243092
Prop-17	5009214
Prop-19	5093451
Prop-2	1512512
Prop-20	2955214
Prop-5	1257671
Prop-6	5120956
Prop-7	5192215

Ερώτημα 3 - Tables' Create & Insert Statements

```
CREATE TABLE `residence` (  
  `ResidenceCode` varchar(10) NOT NULL  
  , `SSN` varchar(20) DEFAULT NULL  
  , PRIMARY KEY (`ResidenceCode`)  
  , CONSTRAINT `ResidenceCode` FOREIGN KEY (`ResidenceCode`)  
  REFERENCES `real_estate` (`PropertyCode`)) ;
```

```
INSERT INTO `residence` (ResidenceCode,SSN)  
VALUES ('Prop-1', 821348541);
```

ResidenceCode	SSN
Prop-1	821348541
Prop-11	219301589
Prop-12	412301891
Prop-13	959128541
Prop-14	213941785
Prop-18	491238154
Prop-3	123914120
Prop-4	123509124
Prop-8	401238191
Prop-9	550129814

Ερώτημα 3 - Tables' Create & Insert Statements

```
CREATE TABLE `estimation` (  
  `EstCode` varchar(10) NOT NULL  
  , `Value` float DEFAULT NULL  
  , `Date` date NOT NULL  
  , `ID_Code` varchar(10) DEFAULT NULL  
  , `EstPropertyCode` varchar(10) DEFAULT NULL  
  , PRIMARY KEY (`EstCode`)  
  , KEY `PropertyCode_idx` (`EstPropertyCode`)  
  , KEY `IDCode_idx` (`ID_Code`)  
  , CONSTRAINT `EstPropertyCode` FOREIGN KEY (`EstPropertyCode`)  
    REFERENCES `real_estate` (`PropertyCode`)  
  , CONSTRAINT `ID_Code` FOREIGN KEY (`ID_Code`) REFERENCES `estimator` (`IDCode`));
```

```
INSERT INTO `estimation`(EstCode, Value, Date, ID_Code, EstPropertyCode)  
VALUES ('E-1', 100000, 20201227, 'EST-1', 'Prop-2');
```

EstCode	Value	Date	ID_Code	EstPropertyCode
E-1	100000	2020-12-27	EST-1	Prop-2
E-10	110000	2019-05-05	EST-1	Prop-19
E-11	65000	2020-01-06	EST-5	Prop-4
E-12	72000	2021-09-02	EST-6	Prop-7
E-13	60000	2020-12-28	EST-2	Prop-4
E-14	96000	2020-12-30	EST-7	Prop-7
E-15	48000	2020-12-29	EST-4	Prop-8
E-16	90000	2021-03-06	EST-1	Prop-13
E-17	56000	2019-02-07	EST-2	Prop-9
E-18	83000	2021-07-19	EST-8	Prop-10
E-19	92000	2020-12-25	EST-9	Prop-11
E-2	120000	2019-01-05	EST-2	Prop-1
E-20	75000	2020-09-12	EST-9	Prop-15
E-21	58000	2020-12-28	EST-8	Prop-14
E-22	187000	2020-12-26	EST-1	Prop-20
E-23	81000	2021-12-05	EST-10	Prop-17
E-24	166000	2020-09-02	EST-8	Prop-12
E-25	42000	2018-11-12	EST-7	Prop-3
E-26	62000	2005-12-01	EST-5	Prop-16

Ερώτημα 4A

```
Select DISTINCT
T0.PropertyCode AS 'Property Code'
, T0.Street
, T0.Number
, T0.Zip_Code
, T0.City
FROM real_estate T0, location T1, estimation T2
WHERE T1.LocCode = T0.LocCode AND T0.PropertyCode = T2.EstPropertyCode
AND T1.AVG_Income > 40000 AND (T2.Date >= '20201224'
AND T2.Date <= '20201231')
```

Property Code	Street	Number	Zip_Code	City
Prop-20	Alkamenous	25	72098	Thessaloniki

Ερώτημα 4B

```
Select T0.IDCode
, T0.First_Name
, T0.Last_Name
, ifnull(T1.EstCode,0) AS 'Estimation_Count'
#We use ifnull to show the estimators with 0 estimations
FROM estimator AS T0
LEFT JOIN
(SELECT COUNT(EstCode) AS 'EstCode'
, T1.ID_Code
FROM estimation AS T1
WHERE (T1.Date >= '20200101' AND T1.Date <= '20201231')) #We find all estimators that have at least one estimations
GROUP BY T1.ID_Code) AS T1 ON T0.IDCode = T1.ID_Code;
```

IDCode	First_Name	Last_Name	Estimation_Count
EST-1	Leopold	Owens	3
EST-10	Marta	Lawrence	0
EST-2	Amy	Whitehouse	2
EST-3	Tom	Hardy	2
EST-4	Harry	Potter	4
EST-5	Steve	Rodgers	1
EST-6	Youta	Molina	0
EST-7	Misha	Mendez	1
EST-8	Puck	Lloyd	2
EST-9	Zoro	Roronoa	2

Ερώτημα 4C

```
Select EstPropertyCode  
, COUNT(*)  
FROM estimation  
WHERE (date >= '20200101' AND date <= '20201231')  
GROUP BY EstPropertyCode  
HAVING COUNT(*) > 2 ;
```

EstPropertyCode	COUNT(*)
Prop-4	3

Ερώτημα 4D

```
Select EstCode
FROM estimation
WHERE EstPropertyCode IN
(SELECT PropertyCode
FROM location, real_estate
WHERE AVG_Income > 25000
AND real_estate.LocCode = location.LocCode);
```

EstCode
E-22
E-30
E-6
E-2
E-9
E-8
E-23
E-12
E-14
E-26
E-17
E-15
E-29
E-18
E-21

Ερώτημα 4Ε

```
Select COUNT(EstCode)
FROM ESTIMATION T0
INNER JOIN REAL_ESTATE T1 ON T0.EstPropertyCode =
T1.PropertyCode
INNER JOIN Location T2 ON T1.LocCode = T2.LocCode
WHERE (T0.Date >= '20200101' AND T0.Date <= '20201231')
AND T2.Population > 50000;
```

COUNT(EstCode)
12

Ερώτημα 4F

```
SELECT T0.LocCode
, T0.Average_Value_per_sq_m
FROM
(Select T0.LocCode
, ROUND(sum(T1.Value)/sum(T0.size),3)
AS 'Average_Value_per_sq_m'
FROM real_estate T0, estimation T1
WHERE T0.PropertyCode = T1.EstPropertyCode
GROUP BY T0.LocCode) T0 #Nested query that calculates the average value
per square for each location, we use a nested a query to order the results
based on the calculated value
ORDER BY T0.Average_Value_per_sq_m;
```

LocCode	Average_Value_per_sq_m
Loc-9	504.348
Loc-6	742.138
Loc-4	833.245
Loc-2	852.778
Loc-8	864.583
Loc-10	898.058
Loc-7	930
Loc-5	1055.085
Loc-3	1274.678
Loc-1	1816.327

Ερώτημα 4G

```
SELECT DISTINCT T0.ID_Code
,IFNULL(T1.Office_Count,0) AS 'Office_Count'
,IFNULL(T1.Residence_Count,0) AS 'Residence_Count'
#We use ifnull to show in the results estimators with 0 estimations
FROM estimation T0
LEFT JOIN
(SELECT T0.IDCode AS 'Estimator_ID'
, COUNT(T1.TAX_ID) AS 'Office_Count'
, COUNT(T2.SSN) AS 'Residence_Count'
FROM Estimator AS T0
INNER JOIN estimation T3 ON T3.ID_Code = T0.IDCode
LEFT JOIN Office T1 ON T1.OfficeCode = T3.EstPropertyCode
LEFT JOIN Residence T2 ON T2.ResidenceCode = T3.EstPropertyCode
WHERE (T3.Date >= '20200101' AND T3.Date <= '20201231')) #Nested query that finds the estimators that have more than 0 calculations
GROUP BY T0.IDCode) T1 ON T0.ID_Code = T1.Estimator_ID;
```

ID_Code	Office_Count	Residence_Count
EST-1	3	0
EST-10	0	0
EST-2	0	2
EST-3	0	2
EST-4	1	3
EST-5	0	1
EST-6	0	0
EST-7	1	0
EST-8	0	2
EST-9	1	1

Ερώτημα 4Η

```
SELECT T0.LocCode
```

```
,T1.Perc_Value_Change_2019_2020
```

```
FROM Location T0
```

LEFT JOIN #left join in order to also show the locations without a change in the below years

```
(SELECT T0.LocCode
```

```
, Round((T1.Average_Value_per_sq_m - T0.Average_Value_per_sq_m) /T0.Average_Value_per_sq_m,3) AS  
'Perc_Value_Change_2019_2020'
```

#Calculate the change of average value per sq m in years 2019-2020

```
FROM (Select T0.LocCode
```

```
, ROUND(sum(T1.Value)/sum(T0.size),3) AS 'Average_Value_per_sq_m'
```

```
FROM real_estate T0
```

```
, estimation T1
```

```
WHERE T0.PropertyCode = T1.EstPropertyCode
```

```
AND (T1.Date >= '20190101' AND T1.Date <= '20191231')
```

```
GROUP BY T0.LocCode) T0 # Nested query to find average value per square meter for locations in 2019,
```

LocCode	Perc_Value_Change_2019_2020
Loc-1	NULL
Loc-3	NULL
Loc-4	NULL
Loc-5	NULL
Loc-6	NULL
Loc-7	NULL
Loc-8	NULL
Loc-9	NULL
Loc-10	-0.601
Loc-2	-0.167

Συνέχεια Ερωτήματος 4Η

```
(Select T0.LocCode
, ROUND(sum(T1.Value)/sum(T0.size),3) AS 'Average_Value_per_sq_m'
FROM real_estate T0
, estimation T1
WHERE T0.PropertyCode = T1.EstPropertyCode
AND (T1.Date >= '20200101' AND T1.Date <= '20201231')
GROUP BY T0.LocCode) T1
# Nested query to find average value per square meter for locations in 2020
WHERE T0.LocCode = T1.LocCode) T1 ON T0.LocCode = T1.LocCode
ORDER BY T1.Perc_Value_Change_2019_2020;
```

Ερώτημα 4|

```
SELECT T0.LocCode
,T0.Count_per_Loc/ T1.All_estimations as 'Estimation Percentage'
,T0.Population / T2.Total_Population as 'Population Percentage'
FROM
(SELECT T0.LocCode
,IFNULL(T1.Count_per_Loc,0) AS 'Count_per_Loc'
,T0.Population
FROM location T0
LEFT JOIN (SELECT COUNT(*) AS 'Count_per_Loc'
,T0.LocCode
,T2.Population
FROM real_estate T0
```

Συνέχεια ερωτήματος 4I

INNER JOIN estimation T1 ON T0.PropertyCode = T1.EstPropertyCode

INNER JOIN location T2 ON T0.LocCode = T2.LocCode

WHERE (T1.Date >= '20200101' AND T1.Date <= '20201231')

GROUP BY T0.LocCode,T2.Population) T1 ON T0.LocCode = T1.LocCode) T0

#Nested Query that finds calculations per location

, (SELECT COUNT(*) AS 'All_estimations'

FROM estimation T1

WHERE (T1.Date >= '20200101' AND T1.Date <= '20201231')) T1

,(SELECT SUM(Population) AS 'Total_Population'

FROM Location T0) T2;

LocCode	Estimation Percentage	Population Percentage
Loc-1	0.1176	0.0996
Loc-10	0.0588	0.0906
Loc-2	0.0588	0.0725
Loc-3	0.1176	0.1268
Loc-4	0.4118	0.1812
Loc-5	0.0588	0.0942
Loc-6	0.0000	0.0634
Loc-7	0.1176	0.0906
Loc-8	0.0000	0.1268
Loc-9	0.0588	0.0543

Ερώτημα 5 (Percentile Population)

```
import mysql.connector
```

```
MyDB = mysql.connector.connect(user="root", passwd="mypw", host="127.0.0.1", database="e-properties"  
, auth_plugin='mysql_native_password')
```

```
mycursor = MyDB.cursor()
```

```
mycursor.execute('SELECT T2.Population FROM location T2')
```

```
Pop_sum = mycursor.fetchall()
```

```
Population_sum = 0
```

```
for i in range(0, len(Pop_sum)):
```

```
    Population_sum += Pop_sum[i][0] #calculate population sum
```

```
print(Population_sum)
```

```
mycursor.execute('SELECT DISTINCT T0.LocCode ,T0.Population From location T0')
```

```
Population_per_Loc = mycursor.fetchall()
```

```
pop_perc = {}
```

```
for i in range(0, len(Population_per_Loc)):
```

```
    pop_perc[Population_per_Loc[i][0]] = int(Population_per_Loc[i][1])/Population_sum #calculate percentile population per loc
```

```
print(pop_perc)
```

```
C:\Users\ninas\AppData\Local\Programs\Python\Python39\python.exe "C:\Program Files\JetBrains\PyCharm Community Edition 2021.2.2\plugins\python-ce\helpers\pydev\pydevd.py" --multiproc --qt-support=auto --client 12  
Connected to pydev debugger (build 212.5284.44)  
552000  
{'Loc-1': 0.09963768115942029, 'Loc-10': 0.09057971014492754, 'Loc-2': 0.07246376811594203, 'Loc-3': 0.12681159420289856, 'Loc-4': 0.18115942028985507, 'Loc-5': 0.09420289855072464, 'Loc-6': 0.06340579710144928,
```

Συνέχεια ερωτήματος 5 (Percentile Estimations)

```
mycursor.execute('SELECT T0.LocCode FROM real_estate T0, estimation T1 WHERE T0.PropertyCode = T1.EstPropertyCode  
AND (T1.Date >= "20200101" AND T1.Date <= "20201231")')
```

```
Location_Estimated_2020 = mycursor.fetchall()
```

```
Total_locations_2020 = len(Location_Estimated_2020)
```

```
Estim_per_location = {}
```

```
for i in range(0, len(Location_Estimated_2020)): #count estimations per loc
```

```
    if Location_Estimated_2020[i][0] in Estim_per_location.keys():
```

```
        Estim_per_location[Location_Estimated_2020[i][0]] += 1
```

```
    else:
```

```
        Estim_per_location[Location_Estimated_2020[i][0]] = 1
```

```
for i in range(0, len(Population_per_Loc)): #add locations with 0 estimations
```

```
    if Population_per_Loc[i][0] not in Estim_per_location.keys():
```

```
        Estim_per_location[Population_per_Loc[i][0]] = 0
```


Συνέχεια ερωτήματος 5 (Percentile Estimations)

```
Perc_per_location = Estim_per_location
```

```
for i in Perc_per_location.keys():
```

```
    Perc_per_location[i] = Perc_per_location[i] / int(Total_locations_2020)
```

```
#count percentile estimations per loc
```

```
print(Perc_per_location)
```

```
C:\Users\ninas\AppData\Local\Programs\Python\Python39\python.exe "C:\Program Files\JetBrains\PyCharm Community Edition 2021.2.2\plugins\python-ce\helpers\pyd
Connected to pydev debugger (build 212.5284.44)
{'Loc-3': 0.006920415224913495, 'Loc-4': 0.02422145328719723, 'Loc-5': 0.0034602076124567475, 'Loc-7': 0.006920415224913495, 'Loc-10': 0.0034602076124567475,
...

```