ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ

ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS

BUSINESS
ANALYTICS
Master of Science

Modern Data Management & Business Intelligence

Assignment #1: E-Properties

Despotis Spyridon: p2822111

Papailiou Thanasis: p2822128

2021

Contents

Introduction

For this assignment we used the MySQL Workbench environment in order to experiment with new functions and tools. Our main goal was to create an efficient model and database for the E-Properties Company with the highest possible performance.

To achieve this, first we defined the entities and the relationships between the tables and we designed the Entity Relationship Diagram. Then, we created the tables and inserted statements to test our model and database through queries. When designing the E-Properties Company database the areas we focused on the most were:

- Good naming conventions in order to reduce opportunities for error
- Building Relationships between tables
- Implementation of primary and secondary keys to ensure consistency in our data
- Preventing unwanted duplication
- Separated tables when needed for faster full table scans and shorter response times for queries
- Simple queries for fast execution

Finally, we exported the Relationship Shema Model which provided a structural visualisation of the database with the major entities and the inter-relationships among these entities.

Section 1: Deliverables of Assignment

The deliverables for the first assignment are the following:

- ✓ Report E-Properties Despotis - Papailiou.pdf
- ✓ ERD.png
- ✓ EER.pdf
- ✓ eproperties_tables.sql
- ✓ eproperties_insert_statements.sql
- ✓ Queries folder (query_a.sql, query_b.sql, query_c.sql, query_d.sql, query_e.sql, query_f.sql, query_g.sql, query_h.sql and query_i.sql)
- ✓ DM&BI Assignment #1 Q5.ipynb

Section 2: Entity Relationship Diagram & Cardinalities

2.1 Entity Relationship Diagram

The Entity Relationship Diagram (ERD) for the E-Properties Company, contains: six **Entities** (Valuer, Evaluation, Region, Property, Office and Residence), with **Attributes** and **four Relationships**.

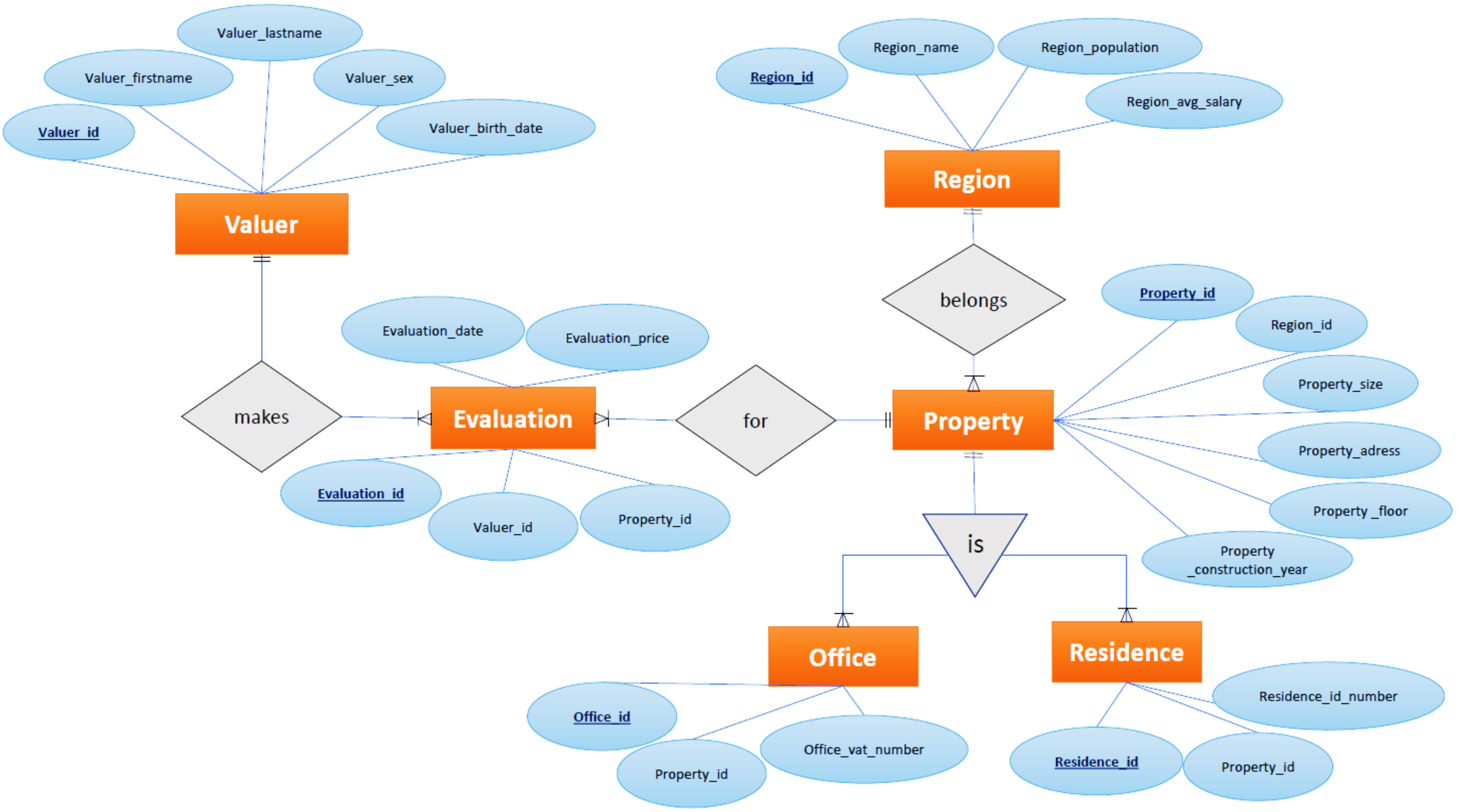
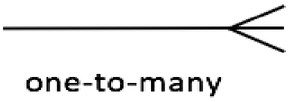


Figure 1: E-Properties ERD: Entities with orange color,

Relationships with grey color & Atrributes with blue color

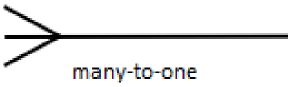
2.2 Cardinalities:

- ***Valuer – Evaluation***:



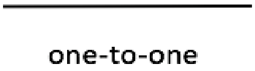
Assumption: It's evaluation has one valuer and its valuer can make many evaluations.

- ***Evaluation – Property***:



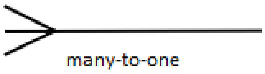
Assumption: Each evaluation refers to one property and one property can be evaluated many times.

- ***Property – Office:***
- ***Property – Residence:***



Assumption: A property can be either an office or a residence. But a property cannot be an office and a residence.

- ***Property - Region***:



Assumption: Each property belongs to one region and one region can have many properties.

Section 3: Relationship Shema Model

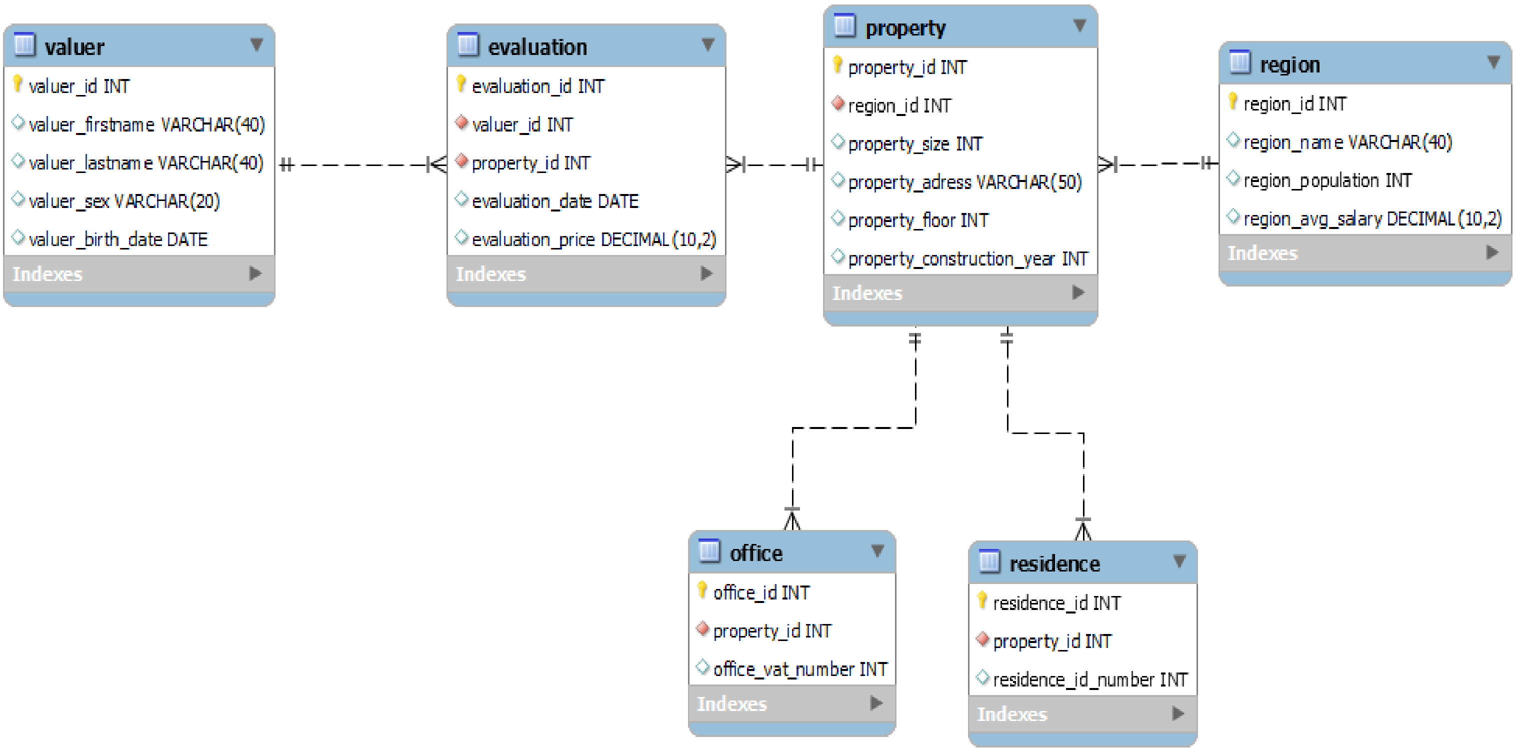


Figure 2: Relation Shema in MySQL.
Primary Keys are indicated with yellow color.
Foreign Keys are indicated with red color.

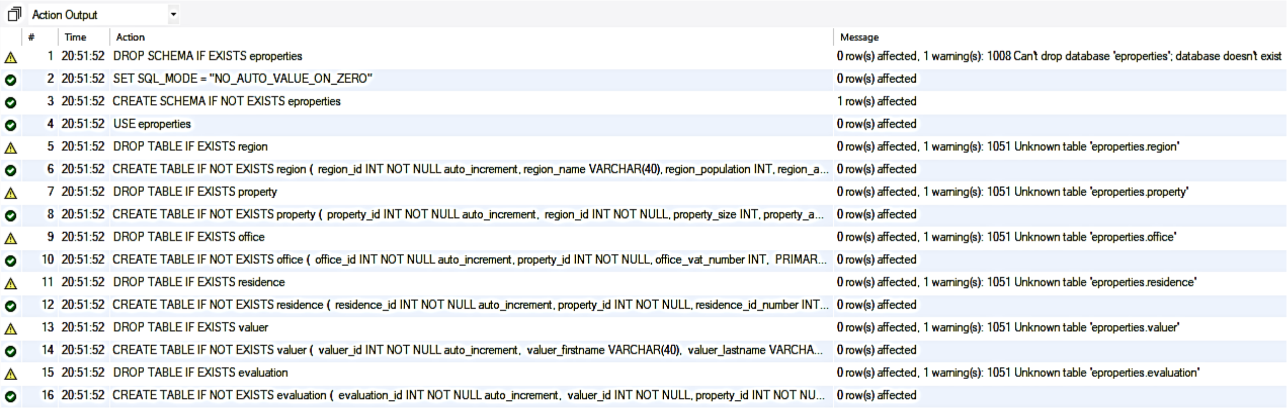
Section 4: Table Creation & Action Output

The following code is used for table creation in MySQL:

```
DROP SCHEMA IF EXISTS eproperties;
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
CREATE SCHEMA IF NOT EXISTS eproperties;
USE eproperties;

-- --- Table eproperties.region -- ---
DROP TABLE IF EXISTS region;
CREATE TABLE IF NOT EXISTS region (
region_id INT NOT NULL auto_increment,
region_name VARCHAR(40),
region_population INT,
region_avg_salary DECIMAL(10,2),
PRIMARY KEY (region_id)
);

-- --- Table eproperties.property -- ---
DROP TABLE IF EXISTS property;
CREATE TABLE IF NOT EXISTS property (
property_id INT NOT NULL auto_increment,
region_id INT NOT NULL,
property_size INT,
property_adress VARCHAR(50),
property_floor INT,
property_construction_year INT,
PRIMARY KEY (property_id),
FOREIGN KEY (region_id) REFERENCES region(region_id)
);

-- --- Table eproperties.office -- ---
DROP TABLE IF EXISTS office;
CREATE TABLE IF NOT EXISTS office (
office_id INT NOT NULL auto_increment,
property_id INT NOT NULL,
office_vat_number INT,
PRIMARY KEY (office_id),
FOREIGN KEY (property_id) REFERENCES property(property_id)
);
```

```
-- --- Table eproperties.residence -- ---
DROP TABLE IF EXISTS residence;
CREATE TABLE IF NOT EXISTS residence (
residence_id INT NOT NULL auto_increment,
property_id INT NOT NULL,
residence_id_number INT ,
PRIMARY KEY (residence_id),
FOREIGN KEY (property_id) REFERENCES property(property_id)
);

-- --- Table eproperties.valuer -- ---
DROP TABLE IF EXISTS valuer;
CREATE TABLE IF NOT EXISTS valuer (
valuer_id INT NOT NULL auto_increment,
valuer_firstname VARCHAR(40),
valuer_lastname VARCHAR(40),
valuer_sex VARCHAR(20),
valuer_birth_date DATE,
PRIMARY KEY (valuer_id)
);

-- --- Table eproperties.evaluation -- ---
DROP TABLE IF EXISTS evaluation;
CREATE TABLE IF NOT EXISTS evaluation (
evaluation_id INT NOT NULL auto_increment,
valuer_id INT NOT NULL,
property_id INT NOT NULL,
evaluation_date DATE,
evaluation_price DECIMAL(10,2),
PRIMARY KEY (evaluation_id),
FOREIGN KEY (valuer_id) REFERENCES valuer(valuer_id),
FOREIGN KEY (property_id) REFERENCES property(property_id)
);
```

The action output is the following:



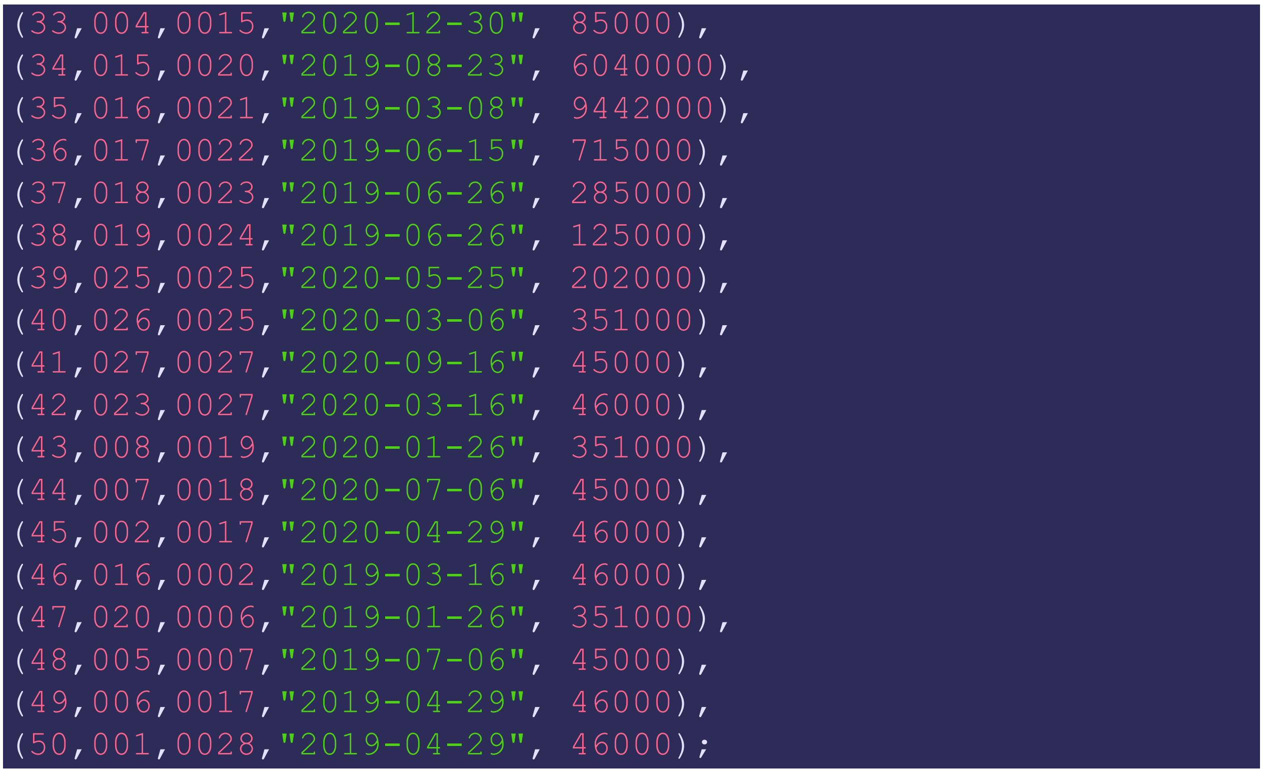
	#	Time	Action	Message
⚠	1	20:51:52	DROP SCHEMA IF EXISTS eproperties	0 row(s) affected, 1 warning(s): 1008 Can't drop database 'eproperties'; database doesn't exist
✓	2	20:51:52	SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO"	0 row(s) affected
✓	3	20:51:52	CREATE SCHEMA IF NOT EXISTS eproperties	1 row(s) affected
✓	4	20:51:52	USE eproperties	0 row(s) affected
⚠	5	20:51:52	DROP TABLE IF EXISTS region	0 row(s) affected, 1 warning(s): 1051 Unknown table 'eproperties.region'
✓	6	20:51:52	CREATE TABLE IF NOT EXISTS region (region_id INT NOT NULL auto_increment, region_name VARCHAR(40), region_population INT, region_a...	0 row(s) affected
⚠	7	20:51:52	DROP TABLE IF EXISTS property	0 row(s) affected, 1 warning(s): 1051 Unknown table 'eproperties.property'
✓	8	20:51:52	CREATE TABLE IF NOT EXISTS property (property_id INT NOT NULL auto_increment, region_id INT NOT NULL, property_size INT, property_a...	0 row(s) affected
⚠	9	20:51:52	DROP TABLE IF EXISTS office	0 row(s) affected, 1 warning(s): 1051 Unknown table 'eproperties.office'
✓	10	20:51:52	CREATE TABLE IF NOT EXISTS office (office_id INT NOT NULL auto_increment, property_id INT NOT NULL, office_vat_number INT, PRIMAR...	0 row(s) affected
⚠	11	20:51:52	DROP TABLE IF EXISTS residence	0 row(s) affected, 1 warning(s): 1051 Unknown table 'eproperties.residence'
✓	12	20:51:52	CREATE TABLE IF NOT EXISTS residence (residence_id INT NOT NULL auto_increment, property_id INT NOT NULL, residence_id_number INT...	0 row(s) affected
⚠	13	20:51:52	DROP TABLE IF EXISTS valuer	0 row(s) affected, 1 warning(s): 1051 Unknown table 'eproperties.valuer'
✓	14	20:51:52	CREATE TABLE IF NOT EXISTS valuer (valuer_id INT NOT NULL auto_increment, valuer_firstname VARCHAR(40), valuer_lastname VARCHA...	0 row(s) affected
⚠	15	20:51:52	DROP TABLE IF EXISTS evaluation	0 row(s) affected, 1 warning(s): 1051 Unknown table 'eproperties.evaluation'
✓	16	20:51:52	CREATE TABLE IF NOT EXISTS evaluation (evaluation_id INT NOT NULL auto_increment, valuer_id INT NOT NULL, property_id INT NOT NU...	0 row(s) affected

Section 5: Inserting Statements & Action Output

The following code is used to insert statements in MySQL:

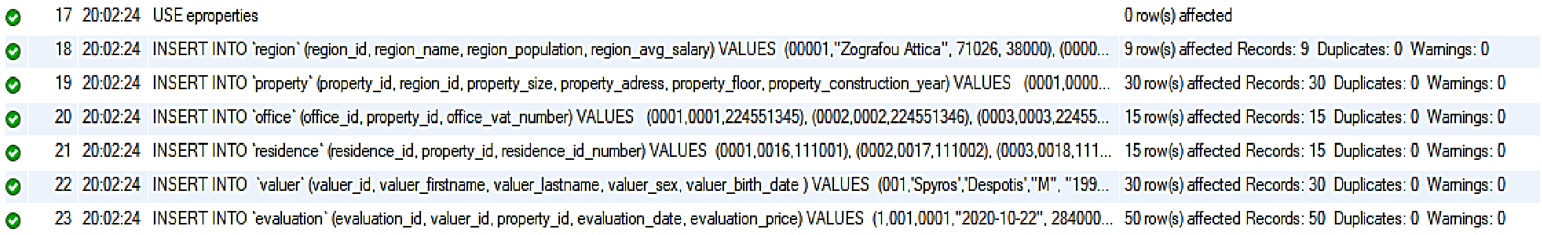
```
USE eproperties;

INSERT INTO `region` (region_id, region_name,
region_population, region_avg_salary) VALUES

(00001,"Zografou Attica", 71026, 38000),
(00002,"Kalamata Peloponnese", 54100, 55000),
(00003,"Athina Attica", 664046, 47657),
(00004,"Heraklion Crete", 144449, 25698),
(00005,"Thessaloniki Central Macedonia", 315196, 30607),
(00006,"Xanthi East Macedonia", 70873, 28986),
(00007,"Volos Thessaly", 144449, 27896),
(00008,"Kastoria West Macedonia", 20147, 32051),
(00009,"Chania Crete", 108642, 27617);

------------------------------------------------------------
INSERT INTO `property` (property_id, region_id,
property_size, property_adress, property_floor,
property_construction_year) VALUES

(0001,00001,70,"Georgiou Papandreou 19 Zografou Attica",
"2", 2020),
(0002,00002,100,"Policharous 33 Kalamata Peloponnese", "2",
2000),
(0003,00003,150,"Ermou 39 Athina Attica", "5", 1998),
(0004,00003,40,"Platia Varnava 11 Athina Attica", "2",
2018),
(0005,00003,87,"Ermou 39 Athina Attica","2", 2004),
(0006,00004,43,"15 Epimenidou st. Heraklion Crete", "5",
2002),
(0007,00006,90,"Stratou 2 Xanthi East Macedonia", "2",
2001),
(0008,00007,140,"Almirou 6 Volos Thessaly", "1", 2020),
(0009,00003,167,"Voutadon 42 Athina Attica", "7", 2000),
(0010,00003,190,"Sofokleous 19 Athina Attica", "4", 1999),
(0011,00005,200,"Sofouli 89 Kalamaria Central Macedonia",
"3", 2011),
(0012,00005,50,"Stratou 2 Thessaloniki Central Macedonia",
"8", 2014),
(0013,00003,56,"Sokratous 50 Athina Attica", "7", 2010),
(0014,00003,87,"Massalias 22 Athina Attica", "4", 2018),
(0015,00003,67,"Rovertou Galli 23 Athina Attica", "5",
2012),
(0016,00003,97,"Leoforos Kifisias 95 Athina", "4", 2015),
(0017,00008,69,"Grammou 237 Kastoria West Macedonia", "6",
2011),
(0018,00003,120,"Leoforos Mesogion 154 Athina Attica", "4",
1995),
(0019,00007,160,"Iolkou 383 Volos Thessaly", "4", 1994),
(0020,00003,134,"Vasilissis Sofias 46 Athina Attica", "2",
1965),
(0021,00003,76,"Pireos 206 Athina Attica", "9", 2021),
(0022,00001,88,"Grigoriou Afxentiou 142 Zografou Attica",
"3", 1992),
(0023,00003,89,"Ethnikis Antistaseos 23 Athina Attica",
"4", 1990),
(0024,00005,69,"Parou 6 Thessaloniki Central Macedonia",
"4", 2014),
(0025,00003,68,"Averof 36 Nea Ionia Attica", "5", 2013),
(0026,00003,132,"Akropolis 105 Athens Attica", "8", 2019),
(0027,00003,190,"Ellis 3 Kallithea Attica", "8", 2001),
(0028,00009,210,"Akti Kountourioti 19 Chania Crete", "2",
1991),
(0029,00003,88,"Ethnikis Antistaseos 1 Peristeri Attica",
"3", 2008),
(0030,00003,90,"Acharnon 340 Athina Attica", "6", 2006);
INSERT INTO `office` (office_id, property_id,
office_vat_number) VALUES

(0001,0001,224551345),
(0002,0002,224551346),
(0003,0003,224551347),
(0004,0004,224551348),
(0005,0005,224551349),
(0006,0006,224551350),
(0007,0007,224551351),
(0008,0008,224551352),
(0009,0009,224551353),
(0010,0010,224551354),
(0011,0011,224551355),
(0012,0012,224551356),
(0013,0013,224551357),
(0014,0014,224551358),
(0015,0015,224551359);
```

```
-------------------------------------------------------------
INSERT INTO `residence` (residence_id, property_id,
residence_id_number) VALUES

(0001,0016,111001),
(0002,0017,111002),
(0003,0018,111003),
(0004,0019,111004),
(0005,0020,111005),
(0006,0021,111006),
(0007,0022,111007),
(0008,0023,111008),
(0009,0024,111009),
(0010,0025,111010),
(0011,0026,111011),
(0012,0027,111012),
(0013,0028,111013),
(0014,0029,111014),
(0015,0030,111015);
-------------------------------------------------------------
INSERT INTO  `valuer` (valuer_id, valuer_firstname,
valuer_lastname, valuer_sex, valuer_birth_date ) VALUES

(001,'Spyros','Despotis',"M", "1995-07-11"),
(002,'Thanasis','Papailiou',"M", "1992-05-30"),
(003,'Dimitris','Papaioannou',"M", "1972-01-02"),
(004,'Vassilis','Mitroglou',"M", "1989-03-23"),
(005,'Christos','Karagiannis',"M", "1963-07-25"),
(006,'Athanasios','Antoniou',"M", "1976-07-30"),
(007,'Michalis','Makris',"M", "1954-12-16"),
(008,'Maria','Christou',"F", "1970-05-13"),
(009,'Eleni','Laskari',"F", "1987-11-20"),
(010,'Christina','Malla',"F", "1990-07-19"),
(011,'Georgia','Bakoyanni',"F", "1972-02-29"),
(012,'Vasiliki','Bountrogianni',"F", "1987-03-06"),
(013,'Konstandina','Carra',"F", "1949-02-13"),
(014,'Nikos','Bernardakis',"M", "1957-08-26"),
(015,'Paraskevi','Hatziioannou',"F", "1976-07-30"),
(016,'Michalis','Hatzimichalis',"M", "1954-12-16"),
(017,'Dimitris','Giannakopoulos',"M", "1970-05-13"),
(018,'Athanasios','Canellakis',"M", "1987-11-20"),
(019,'Maria','Chalkia',"F", "1990-07-19"),
(020,'Pandora','Fotiou',"F", "1972-02-29"),
(021,'Akis','Galanis',"M", "1987-03-06"),
(022,'Giorgos','Daskalakis',"M", "1949-02-13"),
(023,'Sophia','Dimitriou',"F", "1964-07-24"),
(024,'Dimitris','Anagnostou',"M", "1970-05-13"),
(025,'Markos','Canellakis',"M", "1996-03-01"),
(026,'Anastasia','Andreadi',"F", "1957-02-11"),
(027,'Angelina','Amara',"F", "1966-08-20"),
(028,'Minas','Anastasiadis',"M", "1988-12-19"),
(029,'Giorgos','Doxiadis',"M", "1979-05-20"),
(030,'Andromeda','Angelidou',"F", "1995-10-22");

-----------------------------------------------------------
INSERT INTO `evaluation` (evaluation_id, valuer_id,
property_id, evaluation_date, evaluation_price) VALUES

(1,001,0001,"2020-10-22", 284000),
(2,002,0002,"2020-10-10", 276000),
(3,003,0003,"2020-09-06", 250000),
(4,004,0004,"2019-11-28", 128000),
(5,005,0005,"2019-12-08", 52000),
(6,006,0006,"2020-08-13", 185000),
(7,007,0007,"2020-05-20", 87000),
(8,008,0008,"2019-12-12", 110000),
(9,009,0009,"2020-12-25", 113000),
(10,010,0010,"2020-12-28", 250000),
(11,011,0011,"2020-07-28", 190000),
(12,012,0012,"2020-01-06", 89000),
(13,013,0013,"2020-07-29", 39000),
(14,014,0014,"2019-11-02", 110000),
(15,015,0015,"2019-11-26", 65000),
(16,016,0016,"2020-12-08", 100000),
(17,017,0017,"2020-09-23", 99000),
(18,018,0018,"2020-04-02", 125000),
(19,019,0019,"2019-11-07", 780000),
(20,020,0020,"2020-08-23", 1040000),
(21,021,0021,"2020-03-08", 442000),
(22,022,0022,"2020-06-15", 715000),
(23,023,0023,"2020-06-26", 85000),
(24,024,0024,"2020-06-26", 325000),
(25,025,0025,"2019-12-05", 202000),
(26,026,0026,"2020-10-16", 351000),
(27,027,0027,"2020-07-26", 45000),
(28,028,0028,"2020-10-23", 514000),
(29,029,0029,"2020-01-23", 494000),
(30,030,0030,"2020-12-27", 1235000),
(31,002,0015,"2020-12-28", 75000),
(32,003,0015,"2020-12-29", 76000),
```



```
(33,004,0015,"2020-12-30", 85000),
(34,015,0020,"2019-08-23", 6040000),
(35,016,0021,"2019-03-08", 9442000),
(36,017,0022,"2019-06-15", 715000),
(37,018,0023,"2019-06-26", 285000),
(38,019,0024,"2019-06-26", 125000),
(39,025,0025,"2020-05-25", 202000),
(40,026,0025,"2020-03-06", 351000),
(41,027,0027,"2020-09-16", 45000),
(42,023,0027,"2020-03-16", 46000),
(43,008,0019,"2020-01-26", 351000),
(44,007,0018,"2020-07-06", 45000),
(45,002,0017,"2020-04-29", 46000),
(46,016,0002,"2019-03-16", 46000),
(47,020,0006,"2019-01-26", 351000),
(48,005,0007,"2019-07-06", 45000),
(49,006,0017,"2019-04-29", 46000),
(50,001,0028,"2019-04-29", 46000);
```

The action output is the following:



```
17 20:02:24 USE eproperties                                                                                                              0 row(s) affected
18 20:02:24 INSERT INTO `region` (region_id, region_name, region_population, region_avg_salary) VALUES  (00001,"Zografou Attica", 71026, 38000), (0000...  9 row(s) affected Records: 9  Duplicates: 0  Warnings: 0
19 20:02:24 INSERT INTO `property` (property_id, region_id, property_size, property_adress, property_floor, property_construction_year) VALUES   (0001,0000...  30 row(s) affected Records: 30  Duplicates: 0  Warnings: 0
20 20:02:24 INSERT INTO `office` (office_id, property_id, office_vat_number) VALUES   (0001,0001,224551345), (0002,0002,224551346), (0003,0003,22455...  15 row(s) affected Records: 15  Duplicates: 0  Warnings: 0
21 20:02:24 INSERT INTO `residence` (residence_id, property_id, residence_id_number) VALUES  (0001,0016,111001), (0002,0017,111002), (0003,0018,111...  15 row(s) affected Records: 15  Duplicates: 0  Warnings: 0
22 20:02:24 INSERT INTO  `valuer` (valuer_id, valuer_firstname, valuer_lastname, valuer_sex, valuer_birth_date ) VALUES  (001,'Spyros','Despotis',"M", "199...  30 row(s) affected Records: 30  Duplicates: 0  Warnings: 0
23 20:02:24 INSERT INTO `evaluation` (evaluation_id, valuer_id, property_id, evaluation_date, evaluation_price) VALUES  (1,001,0001,"2020-10-22", 284000...  50 row(s) affected Records: 50  Duplicates: 0  Warnings: 0
```

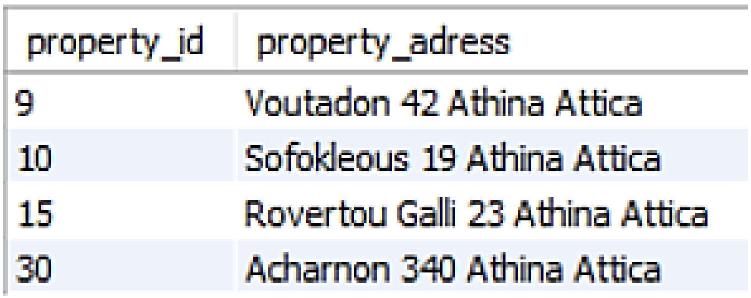
Section 6: SQL Queries

a) Δείξε τον κωδικό και τη διεύθυνση των ακινήτων που ανήκουν σε περιοχή με μέσο εισόδημα μεγαλύτερο των 40.000€ και έχουν εκτιμηθεί μεταξύ 24/12/2020 και 31/12/2020.

- *Query*

```
SELECT property.property_id, property.property_adress
FROM property
INNER JOIN region
ON property.region_id = region.region_id
INNER  JOIN evaluation
ON property.property_id = evaluation.property_id
WHERE region_avg_salary > 40000 AND evaluation_date BETWEEN
'2020-12-24' AND '2020-12-31'
GROUP BY property.property_adress;
```

- *Result*



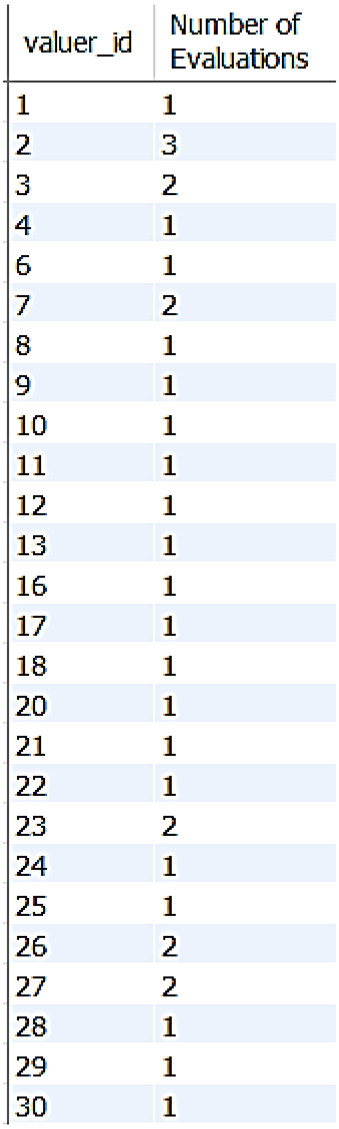
property_id	property_adress
9	Voutadon 42 Athina Attica
10	Sofokleous 19 Athina Attica
15	Rovertou Galli 23 Athina Attica
30	Acharnon 340 Athina Attica

b) Για κάθε εκτιμητή δείξε το πλήθος των εκτιμήσεων που έχει πραγματοποιήσει το 2020.

- *Query*

```
SELECT valuer_id,COUNT(valuer_id) AS 'Number of
Evaluations'
FROM evaluation
WHERE YEAR(evaluation_date) = 2020
GROUP BY valuer_id;
```

➢ *Result*



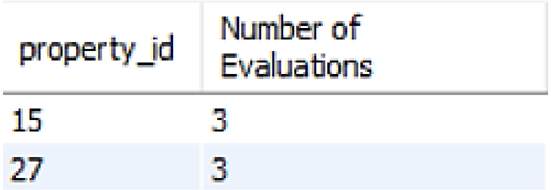
valuer_id	Number of Evaluations
1	1
2	3
3	2
4	1
6	1
7	2
8	1
9	1
10	1
11	1
12	1
13	1
16	1
17	1
18	1
20	1
21	1
22	1
23	2
24	1
25	1
26	2
27	2
28	1
29	1
30	1

c) Δείξε τον κωδικό των ακινήτων που έχουν εκτιμηθεί περισσότερες από δύο φορές μέσα στο 2020.

➢ *Query*

```
SELECT property_id,COUNT(property_id) AS 'Number of
Evaluations'
FROM evaluation
WHERE YEAR(evaluation_date) = 2020
GROUP BY property_id
HAVING COUNT(property_id) > 2;
```

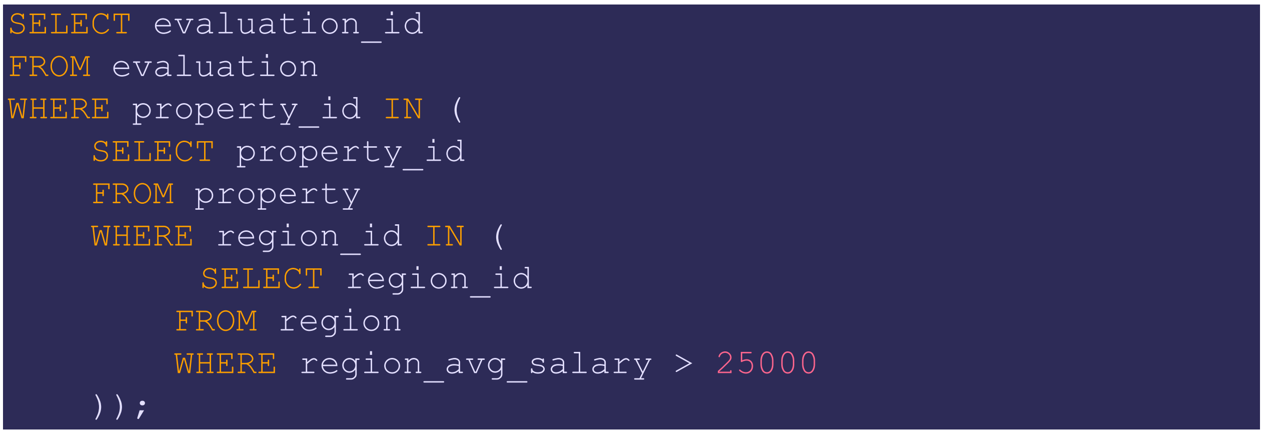
➢ *Result*



property_id	Number of Evaluations
15	3
27	3

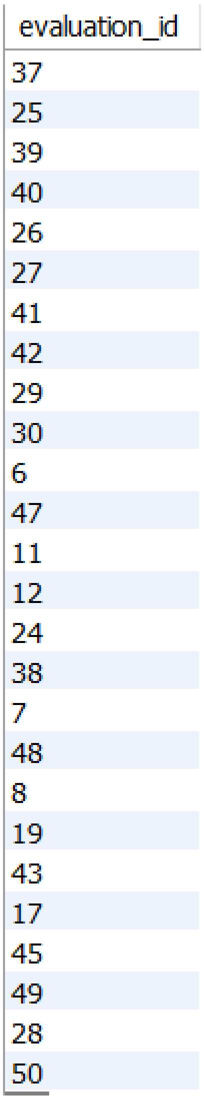
d) Χρησιμοποιώντας εμφωλευμένα ερωτήματα, δείξε τον κωδικό των εκτιμήσεων που έχουν πραγματοποιηθεί σε περιοχές με μέσο εισόδημα μεγαλύτερο των 25.000€.

➢ *Query*



```
SELECT evaluation_id
FROM evaluation
WHERE property_id IN (
    SELECT property_id
    FROM property
    WHERE region_id IN (
         SELECT region_id
        FROM region
        WHERE region_avg_salary > 25000
    ));
```

➢ *Result*



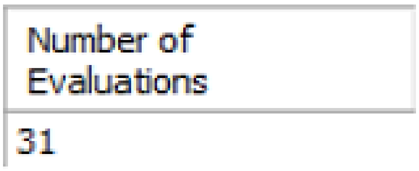
evaluation_id
37
25
39
40
26
27
41
42
29
30
6
47
11
12
24
38
7
48
8
19
43
17
45
49
28
50

e) Δείξε το πλήθος των εκτιμήσεων του 2020 για ακίνητα που ανήκουν σε περιοχές με πληθυσμό > 50.000.

- *Query*

```
SELECT Count(evaluation.evaluation_id) AS 'Number of
Evaluations'
FROM evaluation
INNER JOIN property
ON evaluation.property_id = property.property_id
INNER JOIN region
ON property.region_id = region.region_id
WHERE region_population > 50000 and YEAR(evaluation_date)=
2020;
```

- *Result*



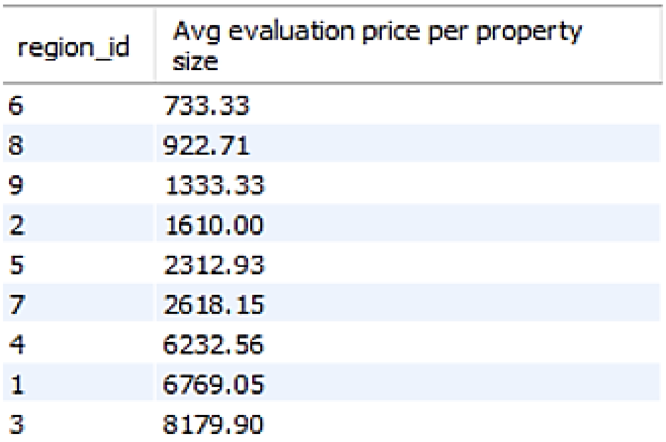
Number of Evaluations
31

f) Για κάθε κωδικό περιοχής, δείξε τον κωδικό της περιοχής και τη μέση τιμή εκτίμησης ανά τμ της περιοχής, σε αύξουσα σειρά της μέσης τιμής εκτίμησης.

- *Query*

```
SELECT region.region_id, ROUND(AVG(evaluation_price /
property_size),2 )AS 'Avg evaluation price per property
size'
FROM evaluation
INNER JOIN property
ON evaluation.property_id = property.property_id
INNER JOIN region
ON property.region_id = region.region_id
GROUP BY region_id
ORDER BY 2 ASC;
```

- *Result*



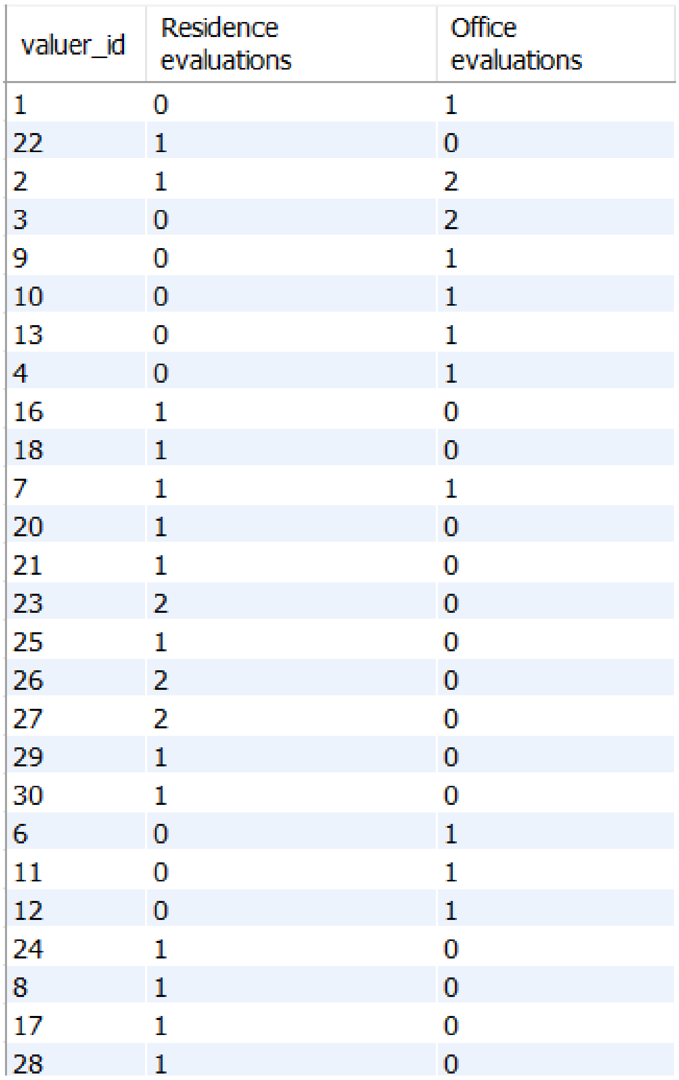
region_id	Avg evaluation price per property size
6	733.33
8	922.71
9	1333.33
2	1610.00
5	2312.93
7	2618.15
4	6232.56
1	6769.05
3	8179.90

g) Για κάθε εκτιμητή και για το 2020, δείξε τον κωδικό του εκτιμητή, το πλήθος των εκτιμήσεων κατοικιών που έχει πραγματοποιήσει, και το πλήθος των εκτιμήσεων γραφείων που έχει πραγματοποιήσει (3 στήλες).

- *Query*

```
SELECT valuer_id,COUNT(residence.residence_id) AS
'Residence evaluations' ,COUNT(office.office_id) AS 'Office
evaluations'
FROM evaluation
INNER JOIN property
ON evaluation.property_id = property.property_id
LEFT JOIN residence
ON property.property_id = residence.property_id
LEFT JOIN office
ON property.property_id = office.property_id
WHERE YEAR(evaluation_date) = 2020
GROUP BY valuer_id;
```

- *Result*



valuer_id	Residence evaluations	Office evaluations
1	0	1
22	1	0
2	1	2
3	0	2
9	0	1
10	0	1
13	0	1
4	0	1
16	1	0
18	1	0
7	1	1
20	1	0
21	1	0
23	2	0
25	1	0
26	2	0
27	2	0
29	1	0
30	1	0
6	0	1
11	0	1
12	0	1
24	1	0
8	1	0
17	1	0
28	1	0

h) Για κάθε κωδικό περιοχής, δείξε τη μεταβολή της μέσης τιμής εκτίμησης ανά τμ μεταξύ 2020 και 2019.

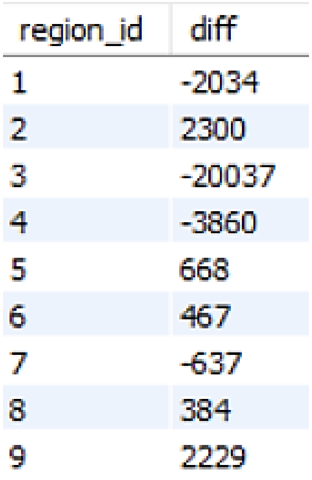
- *Query*

```
SELECT T.region_id, ROUND(sum(case when T.year = 2020 then
T.avg end) - sum(case when T.year = 2019 then T.avg end))
as diff
FROM (
     SELECT
region.region_id,YEAR(evaluation.evaluation_date) AS
'year', AVG(evaluation_price / property_size) AS 'avg'
     FROM evaluation
     INNER JOIN property
     ON evaluation.property_id = property.property_id
     INNER JOIN region
     ON property.region_id = region.region_id
     WHERE YEAR(evaluation.evaluation_date) = 2019  OR
YEAR(evaluation.evaluation_date) = 2020
     GROUP BY region_id, YEAR(evaluation.evaluation_date)
    )  AS T
GROUP BY T.region_id;
```

- *Alternative Query*

```
SELECT  r.region_id, ROUND(AVG(CASE WHEN evaluation_date
LIKE "%2020%" THEN (evaluation_price/property_size) END) -
AVG(CASE WHEN evaluation_date LIKE "%2019%" THEN
(evaluation_price/property_size) END)) AS  diff
FROM region r INNER JOIN property p ON r.region_id =
p.region_id INNER JOIN evaluation e ON p.property_id =
e.property_id
GROUP BY r.region_id;
```

- *Result*



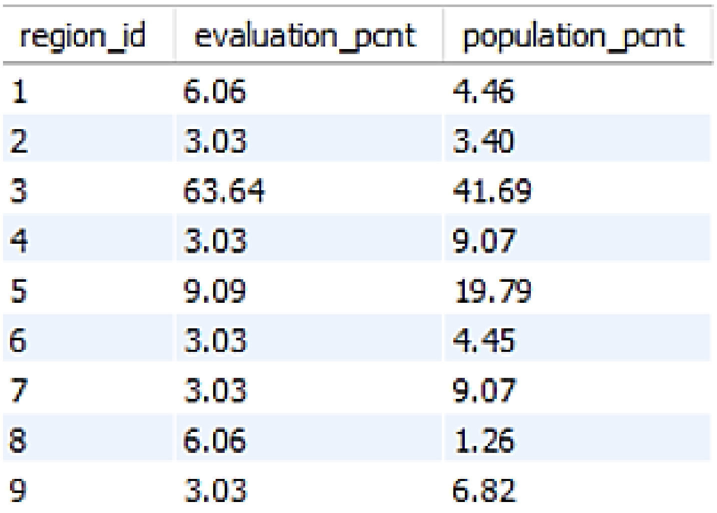
region_id	diff
1	-2034
2	2300
3	-20037
4	-3860
5	668
6	467
7	-637
8	384
9	2229

i) Για κάθε κωδικό περιοχής και για το 2020, δείξε το πλήθος των εκτιμήσεων της περιοχής σαν ποσοστό του συνολικού πλήθος εκτιμήσεων του 2020 (μία στήλη), και τον πληθυσμό της περιοχής σαν ποσοστό του συνολικού πληθυσμού όλων των περιοχών.

- *Query*

```
DROP VIEW IF EXISTS T;
CREATE VIEW T AS
     SELECT region.region_id AS 'region_id',
COUNT(evaluation_id) AS 'count', region_population AS
'population'
     FROM region
     INNER JOIN property
     ON region.region_id = property.region_id
     INNER JOIN evaluation
     ON property.property_id = evaluation.property_id
     WHERE YEAR(evaluation.evaluation_date) = 2020
     GROUP BY region_id;

SELECT region_id, ROUND(count / sum_count * 100,2 )AS
'evaluation_pcnt', ROUND(population / sum_population *
100,2)AS 'population_pcnt'
FROM T, (
    SELECT SUM(count) sum_count, SUM(population)
sum_population
    FROM T
) T1;
```

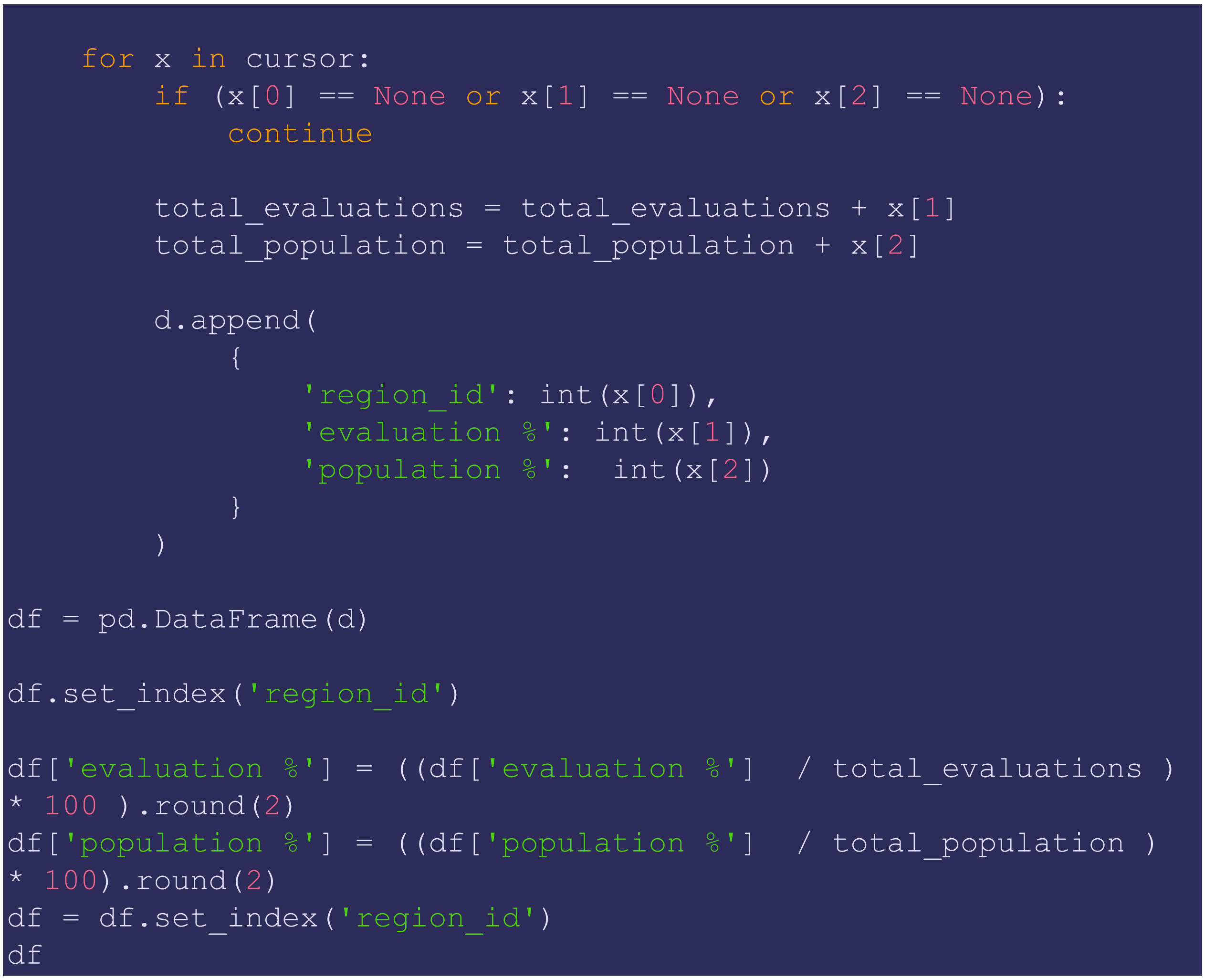
- *Result*



region_id	evaluation_pcnt	population_pcnt
1	6.06	4.46
2	3.03	3.40
3	63.64	41.69
4	3.03	9.07
5	9.09	19.79
6	3.03	4.45
7	3.03	9.07
8	6.06	1.26
9	3.03	6.82

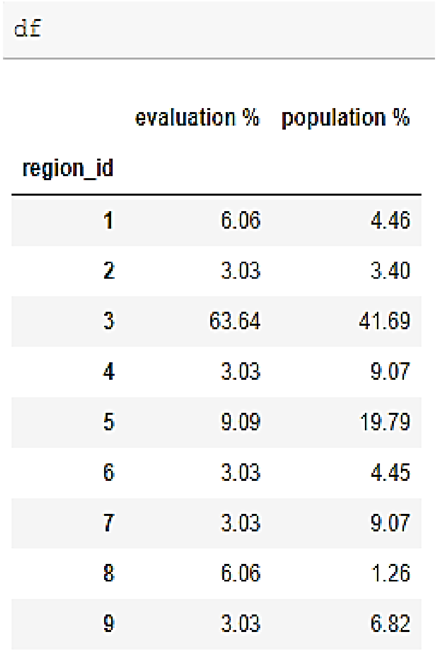
Section 7: Question 5 using Python (Connect to MySQL)

```
import pandas as pd
import datetime
import mysql.connector

# We connect to the database
cnx = mysql.connector.connect(user='root',passwd='thanoshmty7!',
database='eproperties')
cursor = cnx.cursor()

#We retrieve and store all region_ids in a list
query = """
SELECT region.region_id AS 'region_id'
FROM region
"""

cursor.execute(query)

region_ids=[]
for region_id in cursor:
    region_ids.append(region_id[0])


#We create a for loop, iterating in every region_id and storing
the evaluation and population per region_id
#in a dataframe. We also find the total number of evaluations
and population for all the region_ids, so we
# compute the percentages.
total_evaluations = 0
total_population = 0
d = []

for reg_id in region_ids:

    query = """ SELECT region.region_id AS 'region_id',
COUNT(evaluation_id) AS 'count', region_population AS
'population'
    FROM region
    INNER JOIN property ON region.region_id = property.region_id
    INNER JOIN evaluation ON property.property_id =
evaluation.property_id
    WHERE YEAR(evaluation.evaluation_date) = 2020 AND
region.region_id = %s
    """
    cursor.execute(query,(reg_id,))
```



```
    for x in cursor:
        if (x[0] == None or x[1] == None or x[2] == None):
            continue

        total_evaluations = total_evaluations + x[1]
        total_population = total_population + x[2]

        d.append(
            {
                'region_id': int(x[0]),
                'evaluation %': int(x[1]),
                'population %':  int(x[2])
            }
        )

df = pd.DataFrame(d)

df.set_index('region_id')

df['evaluation %'] = ((df['evaluation %']  / total_evaluations )
* 100 ).round(2)
df['population %'] = ((df['population %']  / total_population )
* 100).round(2)
df = df.set_index('region_id')
df
```

- ### *Result*



```
df
```

	evaluation %	population %
region_id		
1	6.06	4.46
2	3.03	3.40
3	63.64	41.69
4	3.03	9.07
5	9.09	19.79
6	3.03	4.45
7	3.03	9.07
8	6.06	1.26
9	3.03	6.82