ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

ΜΕΤΑΠΤΥΧΙΑΚΟ ΤΜΗΜΑ : ΕΠΙΧΕΙΡΗΜΑΤΙΚΗ ΑΝΑΛΥΤΙΚΗ

ΜΑΘΗΜΑ : Data Management And Business Intelligence

ΔΙΔΑΣΚΟΝΤΕΣ : Δ.ΧΑΤΖΗΑΝΤΩΝΙΟΥ , Σ.ΣΑΦΡΑΣ

ΦΟΙΤΗΤΕΣ : Χαρίλαος Πετρακόγιαννης(F282211) , Ηλίας Δήμος (F2822102)

Περιεχόμενα

ΣΚΟΠΟΣ ΤΗΣ ΕΡΓΑΣΙΑΣ :

Σκοπός της εργασίας είναι η δημιουργία μιας βάσης δεδομένων και η εισαγωγή πινάκων και δεδομένων σε αυτή.

Στη εργασία μας δημιουργούμε μια βάση δεδομένων ονόματι «dbmi» η οποία έχει ως σκοπό τη απάντηση ερωτημάτων όσον αφορά την εταιρεία εκτίμησης ακίνητων "E-Properties". Αναλυτικότερα δημιουργούνται οι πίνακες που αφορούν:

Με παρενθέσεις αναφέρονται τα ονόματα των Columns του κάθε Table.

1. **Εκτιμητές (Table: Auditor)**

   Ο πίνακας των εκτιμητών αποτελείται από τον μοναδικό κωδικό τους (AU_ID) , το όνομα τους (FIRST_NAME) , το επίθετο τους (LAST_NAME), το φύλο (SEX) , ηλικία (AGE) και διεύθυνση(A_ADDRESS).

2. **Εκτιμήσεις (Table: Assessment)**

   Ο πίνακας των εκτιμήσεων αποτελείται από μοναδικό κωδικό τους (A_CODE) , τιμή εκτίμησης(PRICE) και ημερομηνία της εκτίμησης(DATE).

3. **Ακίνητα (Table: RealEestate)**

   Ο πίνακας των ακινήτων περιλαμβάνει τον μοναδικό κωδικό τους (RE_ID) ,
   τη διεύθυνση τους RE_ADDRESS) , όροφο (RE_FLOOR) , μέγεθος (RE_SIZE) σε τ.μ. και έτος κατασκευής (RE_YEAR).

4. **Περιοχή που ανήκει το ακίνητο (Table : Location)**

   Ο πίνακας της περιοχής των ακίνητων περιλαμβάνει τον κωδικό της περιοχής (L_CODE) ,
   τη ονομασία της (TITLE), τον πληθυσμό της (POPULATION) , και το μέσο εισόδημα των κατοίκων της (INCOME).

5. **Κατοικίες (Table : Houses)**

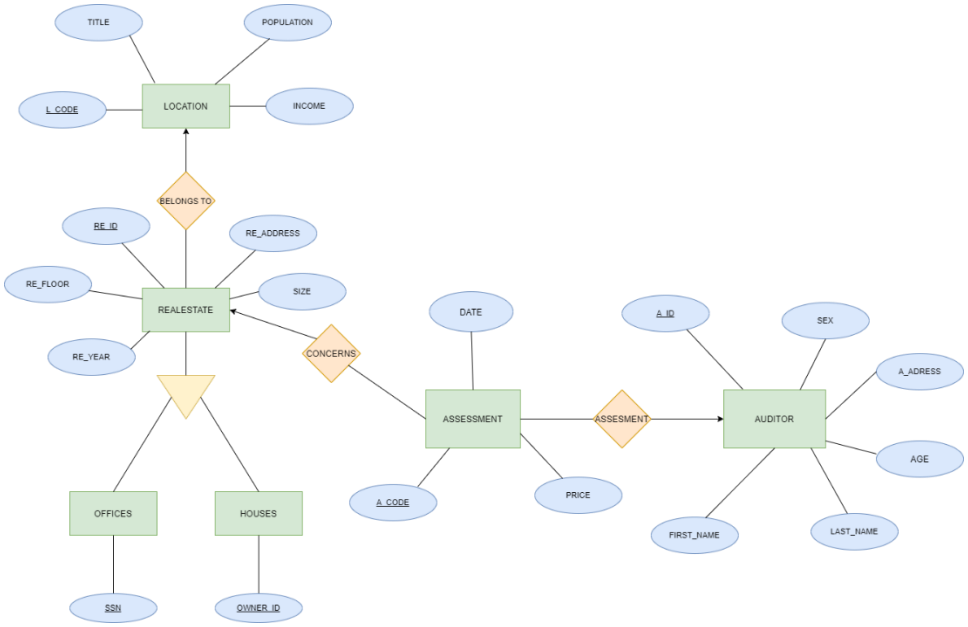
   Ο πίνακας των κατοικιών αποτελείται από τον αριθμό ταυτότητας του ιδιοκτήτη (OWNER_ID).

6. **Γραφεία (Table : Offices)**

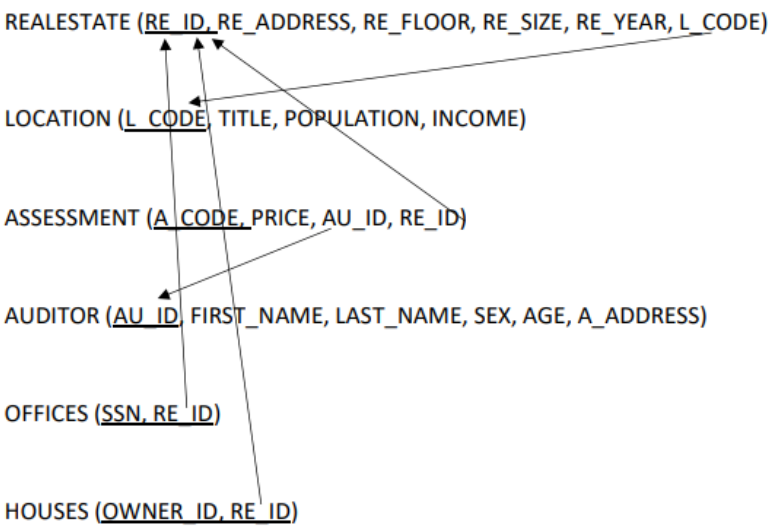
   Ο πίνακας τον γραφείων περιλαμβάνει τον αριθμό φορολογικού μητρώου του ιδιοκτήτη (OWNER_SSN).

ΕΡΩΤΗΜΑΤΑ ΠΡΟΣ ΑΠΑΝΤΗΣΗ:

2) [10%] Περιγράψτε την εφαρμογή σας χρησιμοποιώντας το μοντέλο οντοτήτων-συσχετίσεων. Χρησιμοποιήστε οποιοδήποτε πρόγραμμα θέλετε για να σχεδιάσετε το διάγραμμα.

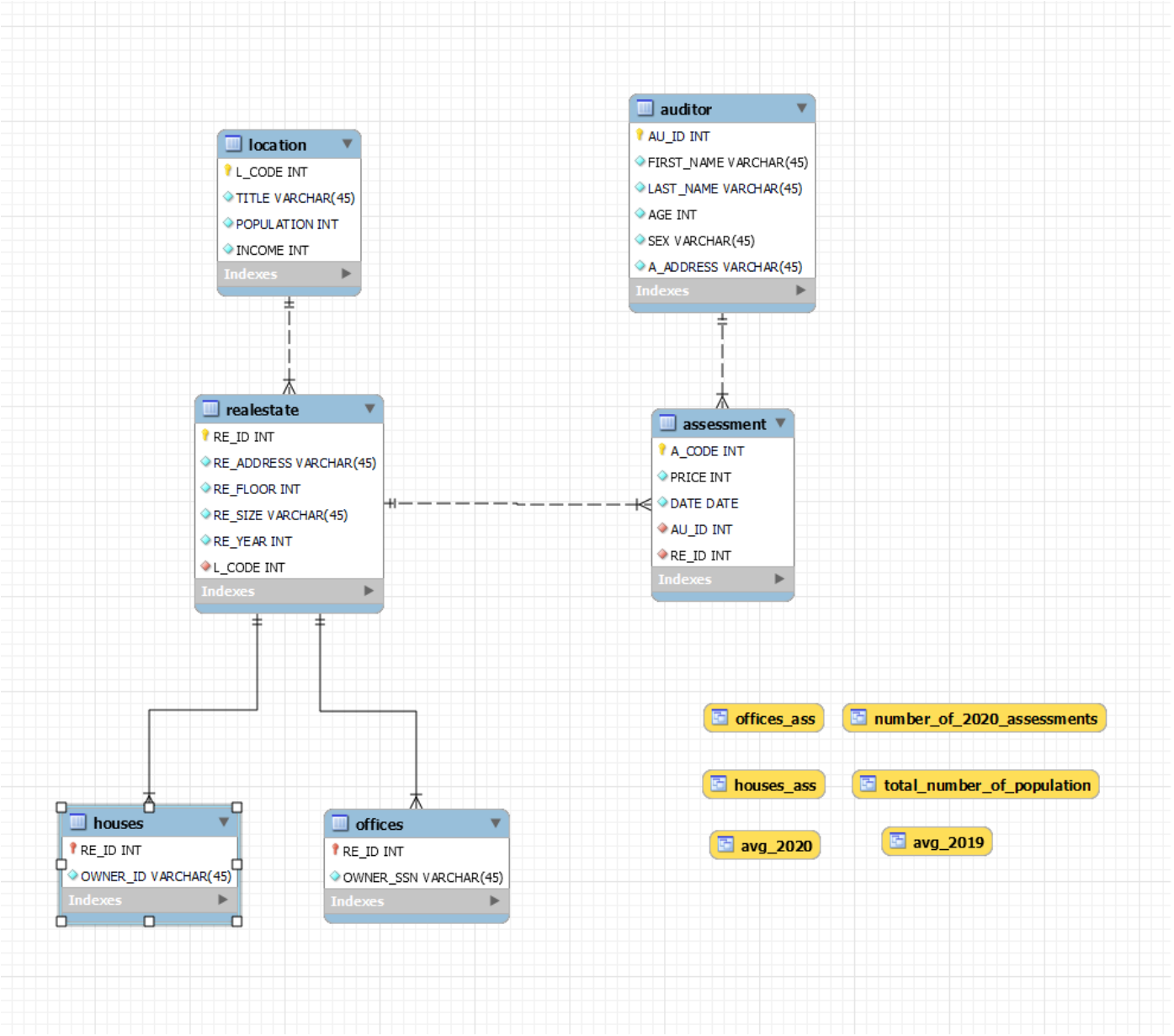


3) [10%] Μεταφέρετε τη σχεδίαση σας στο σχεσιακό μοντέλο και κατόπιν δημιουργήστε τους πίνακες, γνωρίσματα, και περιορισμούς στο SQL Server, χρησιμοποιώντας SQL (δλδ. CREATE TABLE). Εισάγετε γραμμές στους πίνακες με INSERT INTO.



ΔΗΜΙΟΥΡΓΙΑ ΠΙΝΑΚΩΝ ΚΑΙ ΕΙΣΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ ΣΤΗ ΒΑΣΗ «dmbi»

EER DIAGRAM FOR THE «dmbi» DATABASE



1. **ASSESSMENT**

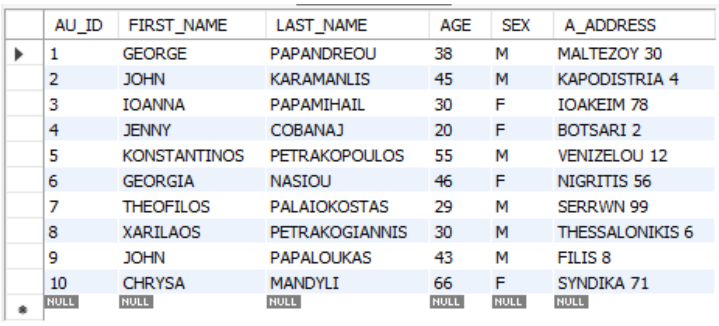
```
CREATE TABLE `assessment` (
  `A_CODE` int NOT NULL,
  `PRICE` int NOT NULL,
  `DATE` date NOT NULL,
  `AU_ID` int NOT NULL,
  `RE_ID` int NOT NULL,
  PRIMARY KEY (`A_CODE`),
  KEY `AU_ID_idx` (`AU_ID`),
  KEY `RE_ID_idx` (`RE_ID`),
  CONSTRAINT `AU_ID` FOREIGN KEY (`AU_ID`) REFERENCES `auditor` (`AU_ID`),
  CONSTRAINT `RE_ID` FOREIGN KEY (`RE_ID`) REFERENCES `realestate` (`RE_ID`)
);

INSERT INTO assessment
VALUES
(10,25000,'2020-10-25',1,1001),
(11,30000,'2020-12-20',2,1002),
(12,37200,'2020-12-31',3,1003),
(13,40000,'2019-12-27',4,1004),
(14,86400,'2016-12-25',5,1005),
(15,23400,'2020-12-01',6,1006),
(16,76300,'2019-11-21',7,1007),
(17,29999,'2019-12-02',8,1008),
(18,39200,'2020-12-24',9,1009),
(19,89000,'2014-09-13',10,1010),
(20,25000,'2020-10-26',1,1001),
(21,25000,'2020-10-27',1,1001),
(22,30000,'2019-05-19',6,1011);
```

A_CODE	PRICE	DATE	AU_ID	RE_ID
10	25000	2020-10-25	1	1001
11	30000	2020-12-20	2	1002
12	37200	2020-12-31	3	1003
13	40000	2019-12-27	4	1004
14	86400	2016-12-25	5	1005
15	23400	2020-12-01	6	1006
16	76300	2019-11-21	7	1007
17	29999	2019-12-02	8	1008
18	39200	2020-12-24	9	1009
19	89000	2014-09-13	10	1010
20	25000	2020-10-26	1	1001
21	25000	2020-10-27	1	1001
22	30000	2019-05-19	6	1011
NULL	NULL	NULL	NULL	NULL

2. **AUDITOR**

```
CREATE TABLE `auditor` (
  `AU_ID` int NOT NULL,
  `FIRST_NAME` varchar(45) NOT NULL,
  `LAST_NAME` varchar(45) NOT NULL,
  `AGE` int NOT NULL,
  `SEX` varchar(45) NOT NULL,
  `A_ADDRESS` varchar(45) NOT NULL,
  PRIMARY KEY (`AU_ID`)
);
```

```
INSERT INTO auditor
VALUES
(1, 'GEORGE', 'PAPANDREOU', 38, 'M', 'MALTEZOY 30'),
(2, 'JOHN', 'KARAMANLIS', 45, 'M', 'KAPODISTRIA 4'),
(3, 'IOANNA', 'PAPAMIHAIL', 30, 'F', 'IOAKEIM 78'),
(4, 'JENNY', 'COBANAJ', 20, 'F', 'BOTSARI 2'),
(5, 'KONSTANTINOS', 'PETRAKOPOULOS', 55, 'M', 'VENIZELOU 12'),
(6, 'GEORGIA', 'NASIOU', 46, 'F', 'NIGRITIS 56'),
(7, 'THEOFILOS', 'PALAIOKOSTAS', 29, 'M', 'SERRWN 99'),
(8, 'XARILAOS', 'PETRAKOGIANNIS', 30, 'M', 'THESSALONIKIS 6'),
(9, 'JOHN', 'PAPALOUKAS', 43, 'M', 'FILIS 8'),
(10, 'CHRYSA', 'MANDYLI', 66, 'F', 'SYNDIKA 71');
```



AU_ID	FIRST_NAME	LAST_NAME	AGE	SEX	A_ADDRESS
1	GEORGE	PAPANDREOU	38	M	MALTEZOY 30
2	JOHN	KARAMANLIS	45	M	KAPODISTRIA 4
3	IOANNA	PAPAMIHAIL	30	F	IOAKEIM 78
4	JENNY	COBANAJ	20	F	BOTSARI 2
5	KONSTANTINOS	PETRAKOPOULOS	55	M	VENIZELOU 12
6	GEORGIA	NASIOU	46	F	NIGRITIS 56
7	THEOFILOS	PALAIOKOSTAS	29	M	SERRWN 99
8	XARILAOS	PETRAKOGIANNIS	30	M	THESSALONIKIS 6
9	JOHN	PAPALOUKAS	43	M	FILIS 8
10	CHRYSA	MANDYLI	66	F	SYNDIKA 71
NULL	NULL	NULL	NULL	NULL	NULL

3. **HOUSES**

```
CREATE TABLE `houses` (
  `RE_ID` int NOT NULL,
  `OWNER_ID` varchar(45) NOT NULL,
  PRIMARY KEY (`RE_ID`,`OWNER_ID`),
  CONSTRAINT `houses_ibfk_1` FOREIGN KEY (`RE_ID`) REFERENCES `realestate` (`RE_ID`)
) ;
```

```
INSERT INTO houses
VALUES
(1001,987123),
(1002,456753),
(1003,123951),
(1004,179352),
(1005,783549);
```

RE_ID	OWNER_ID
1001	987123
1002	456753
1003	123951
1004	179352
1005	783549
NULL	NULL

4. LOCATION

```
CREATE TABLE `location` (
  `L_CODE` int NOT NULL,
  `TITLE` varchar(45) NOT NULL,
  `POPULATION` int NOT NULL,
  `INCOME` int NOT NULL,
  PRIMARY KEY (`L_CODE`)
) ;

INSERT INTO location
VALUES
(101,'N.IONIA', 66000,15000),
(102,'XOLARGOS',90000,41000),
(103,'PATHSIA',34000,100000),
(104,'PAPAGOU',20000,80000),
(105,'PERISTERI',50000,20000),
(106,'METAKSOURGIO',10000,42000),
(107,'KIFISIA',70000,97000),
(108,'ILION',43900,10000),
(109,'VIKTORIA',50003,7000),
(110,'KIPSELI',28400,23000);
```

L_CODE	TITLE	POPULATION	INCOME
101	N.IONIA	66000	15000
102	XOLARGOS	90000	41000
103	PATHSIA	34000	100000
104	PAPAGOU	20000	80000
105	PERISTERI	50000	20000
106	METAKSOURGIO	10000	42000
107	KIFISIA	70000	97000
108	ILION	43900	10000
109	VIKTORIA	50003	7000
110	KIPSELI	28400	23000
NULL	NULL	NULL	NULL

5. OFFICES

```
CREATE TABLE `offices` (
  `RE_ID` int NOT NULL,
  `OWNER_SSN` varchar(45) NOT NULL,
  PRIMARY KEY (`RE_ID`,`OWNER_SSN`),
  CONSTRAINT `offices_ibfk_1` FOREIGN KEY (`RE_ID`) REFERENCES `realestate` (`RE_ID`)
) ;
```

```
INSERT INTO offices
VALUES
(1006,937293),
(1007,938110),
(1008,382610),
(1009,398127),
(1010,293003),
(1011,131225);
```

RE_ID	OWNER_SSN
1006	937293
1007	938110
1008	382610
1009	398127
1010	293003
1011	131225
NULL	NULL

6. REALESTATE

```
CREATE TABLE `realestate` (
  `RE_ID` int NOT NULL,
  `RE_ADDRESS` varchar(45) NOT NULL,
  `RE_FLOOR` int NOT NULL,
  `RE_SIZE` varchar(45) NOT NULL,
  `RE_YEAR` int NOT NULL,
  `L_CODE` int NOT NULL,
  PRIMARY KEY (`RE_ID`),
  KEY `L_CODE_idx` (`L_CODE`),
  CONSTRAINT `L_CODE` FOREIGN KEY (`L_CODE`) REFERENCES `location` (`L_CODE`)
) ;
```

```
INSERT INTO realestate
VALUES
(1001,'XOREMI 73',2,110,2020,101),
(1002,'KREONTOS 29',3,90,1867,102),
(1003,'ANTIGONIS 29',6,120,2001,103),
(1004,'AMFIARAOU 34',3,130,2020,104),
(1005,'GERAKIOU 39',2,70,2019,105),
(1006,'APSOU 30',8,110,2020,106),
(1007,'THEOFANOUS 12',9,130,2015,107),
(1008,'DRAGOUMI 29',2,100,1978,108),
(1009,'KIFISIAS 39',5,170,2020,109),
(1010,'TSALDARI 20',2,50,1847,110),
(1011,'APSOU 34',3,110,1950,106);
```

RE_ID	RE_ADDRESS	RE_FLOOR	RE_SIZE	RE_YEAR	L_CODE
1001	XOREMI 73	2	110	2020	101
1002	KREONTOS 29	3	90	1867	102
1003	ANTIGONIS 29	6	120	2001	103
1004	AMFIARAOU 34	3	130	2020	104
1005	GERAKIOU 39	2	70	2019	105
1006	APSOU 30	8	110	2020	106
1007	THEOFANOUS 12	9	130	2015	107
1008	DRAGOUMI 29	2	100	1978	108
1009	KIFISIAS 39	5	170	2020	109
1010	TSALDARI 20	2	50	1847	110
1011	APSOU 34	3	110	1950	106
NULL	NULL	NULL	NULL	NULL	NULL

ΥΛΟΠΟΙΗΣΗ ΕΡΩΤΗΜΑΤΩΝ ΜΕ SQL QUERIES

4) [60%] Γράψτε SQL και εκτελέστε τα παρακάτω ερωτήματα:

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.

SQL QUERY

```
SELECT realestate.RE_ID, realestate.RE_ADDRESS
FROM realestate, location, assessment
WHERE realestate.L_CODE=location.L_CODE AND realestate.RE_ID=assessment.RE_ID AND
location.INCOME > 40000 AND assessment.DATE BETWEEN '2020-12-24' AND '2020-12-31' ;
```

Outcome:

RE_ID	RE_ADDRESS
1003	ANTIGONIS 29

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

SQL QUERY

```
SELECT count(assessment.A_CODE) as COUNT_OF_ASSESSEMENTS_IN_2020, auditor.AU_ID, auditor.FIRST_NAME,
auditor.LAST_NAME
FROM assessment
right JOIN AUDITOR
ON assessment.AU_ID=auditor.AU_ID and year(assessment.date)=2020
GROUP BY auditor.AU_ID ;
```

Outcome:

COUNT_OF_ASSESSEMENTS_IN_2020	AU_ID	FIRST_NAME	LAST_NAME
3	1	GEORGE	PAPANDREOU
1	2	JOHN	KARAMANLIS
1	3	IOANNA	PAPAMIHAIL
0	4	JENNY	COBANAJ
0	5	KONSTANTINOS	PETRAKOPOULOS
1	6	GEORGIA	NASIOU
0	7	THEOFILOS	PALAIOKOSTAS
0	8	XARILAOS	PETRAKOGIANNIS
1	9	JOHN	PAPALOUKAS
0	10	CHRYSA	MANDYLI

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

SQL QUERY

```
SELECT DISTINCT(assessment.RE_ID)
FROM assessment
WHERE EXTRACT(YEAR FROM assessment.DATE)=2020
GROUP BY assessment.RE_ID
HAVING COUNT(assessment.RE_ID)>2 ;
```

Outcome:

RE_ID
1001

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

SQL QUERY

```
SELECT assessment.A_CODE AS ASSESSMENT_CODE
FROM assessment
WHERE assessment.RE_ID IN (SELECT realestate.RE_ID
                           FROM realestate
                           WHERE realestate.L_CODE IN (SELECT location.L_CODE
                                                       FROM location
                                                       WHERE location.INCOME > 25000));
```

Outcome:

ASSESSMENT_CODE
11
12
13
15
22
16

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

SQL QUERY

```
SELECT COUNT(A_CODE)
FROM assessment
INNER JOIN REALESTATE
ON assessment.RE_ID=realestate.RE_ID
INNER JOIN location
ON realestate.L_CODE=location.L_CODE
WHERE location.POPULATION > 50000 AND YEAR(assessment.date)=2020;
```

Outcome:

	TOTAL_ASSESSMENTS_2020
▶	5

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

SQL QUERY

```
SELECT location.L_CODE AS LOCATION_CODE, AVG(assessment.PRICE)/realestate.RE_SIZE as
AVERAGE_PRICE_PER_REALESTATE_SIZE
FROM location, assessment, realestate
WHERE location.L_CODE=realestate.L_CODE AND assessment.RE_ID=realestate.RE_ID
GROUP BY location.L_CODE
ORDER BY AVERAGE_PRICE_PER_REALESTATE_SIZE ASC;
```

Outcome:

	LOCATION_CODE	AVERAGE_PRICE_PER_REALESTATE_SIZE
▶	101	227.27272727272728
	109	230.58823529411765
	106	242.72727272727272
	108	299.99
	104	307.6923076923077
	103	310
	102	333.3333333333333
	107	586.9230769230769
	105	1234.2857142857142
	110	1780

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).

SQL QUERY

```
CREATE VIEW OFFICES_ASS AS
SELECT auditor.AU_ID AS AUDITOR_ID, assessment.DATE AS DATE_OF_ASSESSMENT, COUNT(offices.RE_ID) AS
TOTAL_OFFICE_ASSESSMENTS_2020
FROM offices,assessment,auditor
WHERE offices.RE_ID=assessment.RE_ID
AND YEAR(assessment.DATE)=2020
AND assessment.AU_ID=auditor.AU_ID
GROUP BY auditor.AU_ID ;

CREATE VIEW HOUSES_ASS AS
SELECT auditor.AU_ID AS AUDITOR_ID,assessment.DATE AS DATE_OF_ASSESSMENT, COUNT(houses.RE_ID) AS
TOTAL_HOUSE_ASSESSMENTS_2020
FROM houses,assessment,auditor
WHERE houses.RE_ID=assessment.RE_ID
AND YEAR(assessment.DATE)=2020
AND assessment.AU_ID=auditor.AU_ID
GROUP BY auditor.AU_ID ;

SELECT auditor.AU_ID AS AUDITOR_ID, TOTAL_OFFICE_ASSESSMENTS_2020, TOTAL_HOUSE_ASSESSMENTS_2020
FROM auditor
LEFT JOIN HOUSES_ASS ON (auditor.AU_ID=houses_ass.AUDITOR_ID)
LEFT JOIN OFFICES_ASS ON (auditor.AU_ID=offices_ass.AUDITOR_ID)
GROUP BY auditor.AU_ID;
```

Outcome:

AUDITOR_ID	TOTAL_OFFICE_ASSESSMENTS_2020	TOTAL_HOUSE_ASSESSMENTS_2020
1	0	3
2	0	1
3	0	1
4	0	0
5	0	0
6	1	0
7	0	0
8	0	0
9	1	0
10	0	0

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

SQL QUERY

```
CREATE VIEW AVG_2020 AS
SELECT LOCATION.L_CODE,AVG(assessment.Price)/realestate.RE_SIZE AS AV_2020
from location
inner join realestate
on location.L_CODE=realestate.L_CODE
inner join assessment
on realestate.RE_ID=assessment.RE_ID
WHERE YEAR(assessment.DATE)=2020
GROUP BY location.L_CODE;

CREATE VIEW AVG_2019 AS
SELECT LOCATION.L_CODE,AVG(assessment.Price)/realestate.RE_SIZE  AS AV_2019
from location
inner join realestate
on location.L_CODE=realestate.L_CODE
inner join assessment
on realestate.RE_ID=assessment.RE_ID
WHERE YEAR(assessment.DATE)=2019
GROUP BY location.L_CODE;

SELECT LOCATION.L_CODE,(av_2020-av_2019)/av_2019 AS CHANGE_OF_PRICE
FROM location
LEFT JOIN AVG_2020 ON (location.L_CODE=AVG_2020.L_CODE)
LEFT JOIN AVG_2019 ON (location.L_CODE=AVG_2019.L_CODE)
GROUP BY location.L_CODE;
```

Outcome:

	L_CODE	CHANGE_OF_PRICE
▶	101	0
	102	0
	103	0
	104	0
	105	0
	106	-0.22000000000000008
	107	0
	108	0
	109	0
	110	0

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.

SQL QUERY

```
CREATE VIEW NUMBER_OF_2020_ASSESSMENTS AS
SELECT COUNT(assessment.A_CODE) AS COUNT_2020
FROM assessment
WHERE YEAR(assessment.date)=2020;

CREATE VIEW TOTAL_NUMBER_OF_POPULATION AS
SELECT SUM(location.POPULATION) AS TOTAL_POPULATION
FROM location;

SELECT  location.L_CODE AS LOCATION_CODE, COUNT(assessment.A_CODE)/COUNT_2020*100 AS
PERCENT_ASSESSMENT, location.POPULATION/TOTAL_POPULATION*100 AS PERCENT_POPULATION
FROM NUMBER_OF_2020_ASSESSMENTS,TOTAL_NUMBER_OF_POPULATION,realestate
INNER JOIN location ON (location.L_CODE=realestate.L_CODE)
LEFT JOIN assessment ON (realestate.RE_ID = assessment.RE_ID AND YEAR(assessment.DATE)=2020)
GROUP BY location.L_CODE;
```

Outcome:

LOCATION_CODE	PERCENT_ASSESSMENT	PERCENT_POPULATION
101	42.8571	14.2764
102	14.2857	19.4678
103	14.2857	7.3545
104	0.0000	4.3262
105	0.0000	10.8154
106	14.2857	2.1631
107	0.0000	15.1416
108	0.0000	9.4959
109	14.2857	10.8161
110	0.0000	6.1432

ΣΥΝΔΕΣΗ ΣΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ «dmbi» ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΕΡΩΤΗΜΑΤΟΣ (i) ΧΩΡΙΣ GROUP BY

5) [20%] Χρησιμοποιώντας όποια γλώσσα προγραμματισμού επιθυμείτε, συνδεθείτε στη ΒΔ και υλοποιήστε το ερώτημα (i) παραπάνω χωρίς τη χρήση GROYP BY στο SQL Statement, δλδ μπορείτε να χρησιμοποιήσετε μόνο SELECT…FROM…WHERE.

Python Code

```
import mysql.connector
import pandas as pd


mydb = mysql.connector.connect(
  host="127.0.0.1",
  user="root",
  password="trav4-mk3",
  database="dmbi"
)
mycursor=mydb.cursor()

print("MySQL Database connection is successful")

sql=("SELECT L.L_CODE AS LOCATION_CODE, (count/COUNT_2020) AS PERCENT_ASSESSMENT,
loc_population/tot_population*100 AS PERCENT_POPULATION "
     "FROM (  "
         "SELECT L1.L_CODE "
     ",(SELECT COUNT(assessment.A_CODE)*10 FROM assessment,location,realestate WHERE
realestate.L_CODE=L1.L_CODE AND assessment.RE_ID=realestate.RE_ID AND
YEAR(assessment.DATE)=2020) AS count "
     ",(SELECT location.POPULATION FROM location WHERE location.L_CODE=L1.L_CODE) AS
loc_population "
     ",(SELECT COUNT_2020 FROM NUMBER_OF_2020_ASSESSMENTS) AS count_2020  "
     ",(SELECT TOTAL_POPULATION FROM TOTAL_NUMBER_OF_POPULATION) AS tot_population "
    "FROM (SELECT distinct location.L_CODE FROM location,realestate,assessment WHERE
location.L_CODE=realestate.L_CODE AND realestate.RE_ID=assessment.RE_ID) AS L1 "
     ") AS L " )


mycursor.execute(sql)
#Making the outcome of the sql query a data frame
df=pd.DataFrame(mycursor,columns=['LOCATION CODE','PERCENT ASSESSMENT','PERCENT POPULATION '])

print(df)
```

Python Outcome:

```
MySQL Database connection is successful
   LOCATION CODE PERCENT ASSESSMENT PERCENT POPULATION
0            101           42.8571            14.2764
1            102           14.2857            19.4678
2            103           14.2857             7.3545
3            104            0.0000             4.3262
4            105            0.0000            10.8154
5            106           14.2857             2.1631
6            107            0.0000            15.1416
7            108            0.0000             9.4959
8            109           14.2857            10.8161
9            110            0.0000             6.1432

Process finished with exit code 0
```