|  |  |
| --- | --- |
| Modern Data Management  &  Business Intelligence  Assignment 1 | Students  Eftychia Gkini (p2822108)  Maria Skoli (p2822131)  Class  Part Time |

Table of Contents

[Introduction 2](#_Toc87729733)

[Entity-Relationship Model 2](#_Toc87729734)

[Create Tables 2](#_Toc87729735)

[Relational Schema 4](#_Toc87729736)

[Insert Values in Tables 5](#_Toc87729737)

[Retrieve Data from E-Properties based on certain cases. 6](#_Toc87729738)

[Case 1: *Properties from Regions with average income > 40.000 and estimation period between 24.12.20 and 31.12.20.* 6](#_Toc87729739)

[Case 2: *Number of Estimations for the year 2020 per Agent* 7](#_Toc87729740)

[Case 3: *Number of Properties that have been estimated more than two times during 2020* 8](#_Toc87729741)

[Case 4: *Number of estimations in regions where the average income > 25.000€* 9](#_Toc87729742)

[Case 5: *Number of properties estimations in regions with population > 50.000 in 2020* 10](#_Toc87729743)

[Case 6: *Average estimation per square meter for each Region.* 11](#_Toc87729744)

[Case 7: *Number of Apartments and Number of Offices each Agent has estimated.* 12](#_Toc87729745)

[Case 8: *The change in the average Price of estimation per m2 from the year 2019 to the year 2020* 14](#_Toc87729746)

[Case 9: *The percentage of estimations and the percentage of population per Region during 2020* 16](#_Toc87729747)

[Connecting R with the database in SQL Server Management Studio 18](#_Toc87729748)

[Case 9: *The percentage of estimations and the percentage of population per region during 2020 (using R)* 18](#_Toc87729749)

Introduction

For the purpose of this Assignment, we are going to delve into to a real estate dataset based on the site E-Properties.

The line we will follow to perform our analysis is the following. First of all, we are going to design the Entity- Relationship Diagram which will be a great indicator of the existing relationship between the database. After that, we are going to Create our Tables in SQL and we will insert our data in each table.

Afterwards, we are going to present how to extract the information we want from our database using the appropriate queries based on specific cases.

Finally, we will present how to connect our database in SQL Server Management Studio with the programming language R in order to perform certain processes.

Entity-Relationship Model

The first step of our analysis is to design the Entity-Relationship Diagram of our data. This will be the guide we will follow to perform our analysis. The E-R Diagram consists of entities, relationships, attributes and constraints. Due to the nature of our assignment, we observed that our data can be divided in 5 main entities: Agent, Estimation, Property and Region. Each entity has several attributes as we can see in the E-R Diagram below.

More specifically, the attributes of the entity Agent are: Name, Last Name, Gender, Age, Address and the ID of the Agent. The attributes for the entity Estimation are the ID, the Date, and the Price of the real estate estimation. The attributes for the entity Property are the ID, Address, Floor, Size, Year of Construction and the Type of the apartment. The attribute Type has also the AFM and the Identity Number as attributes. Finally, Region entity has as attributes: the ID of the Region, the Name of the Region, the population and the average income.

To create related tables, we first define a relationship between two tables. The relationship that connects our entities is the following:

An Agent calculates one or more Estimations. An estimation is based on a specific property, but a property can have multiples estimations. Finally, one or more properties are located in a specific Region.

Therefore, the connections between the tables are “one to many” ( 1🡪\*) . Because relationships work both ways, there are also “many-to-one” relationships.

The E-R Diagram is shown below. The Drawio (app.diagrams.net) is used for the design of the Diagram.

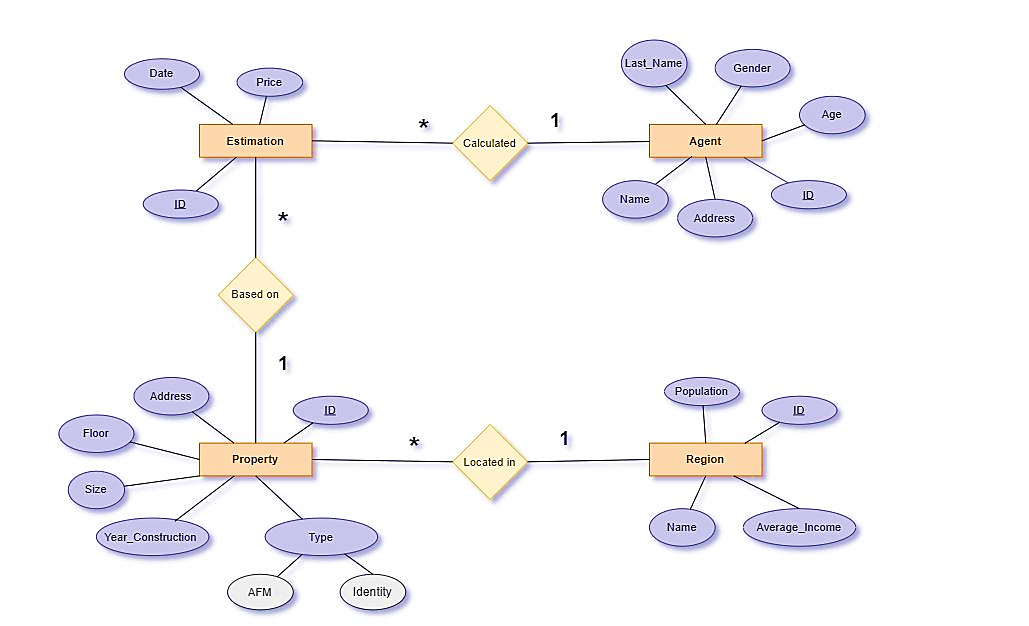
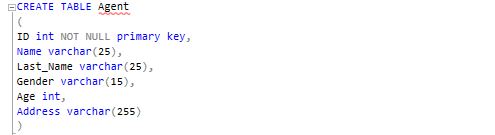


Figure 1: Entity- relationship model

Create Tables

The first step is to create a new database named *“E-Property”* in Microsoft SQL Server Management Studio. Then, we are going to create a table for each entity. These tables have as columns the attributes of the corresponding entity.

* The following SQL query creates a new table called **Agent**.



The columns ID and gender are of type int and will hold an integer. Also, ID has been declared as not null since it is a primary key and it must be not null. The columns: Name, Last\_Name, Gender and address are of type varchar and will hold characters.

* The following SQL query creates a new table called **Region**

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

We declare the column Name as varchar, the column Population as integer and the average income as decimal. The ID is the primary key of this table.

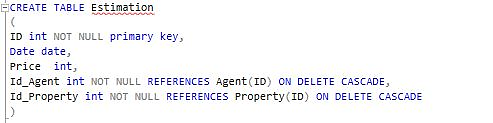
* The following SQL query creates a new table called **Property**

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

We declare the columns: ID, Floor property, Size, AFM and Year of Construction as integers and the columns Address, Type, Identity\_Number as varchar because they will hold characters. Also, we add a new variable: Id\_Region which is going to be the foreign key that connects the table Property with the table Region. We used the function “References … on delete cascade” in order to delete the referencing rows in the Property table when the referenced row is deleted in the Region table which has a primary key.

* The following SQL query creates a new table called **Estimation.**



We declare the columns ID and Price as integers and the column Date as date type. Also, we insert two more attributes the Id\_Agent and the Id\_Property which are going to be the foreign keys that connect the table Estimation with the tables Agent and Property respectively.

Query 1[[1]](#footnote-1)



Relational Schema

In order to have a better perspective of the connection of the tables we created in Microsoft SQL Server Management Studio the relational diagram below. In this diagram we can understand better the one-to-many connection. For instance, the primary key in the Property table, ID is designed to contain unique values. The foreign key in the Estimation table, Property\_ID is designed to allow multiple instances of the same value.

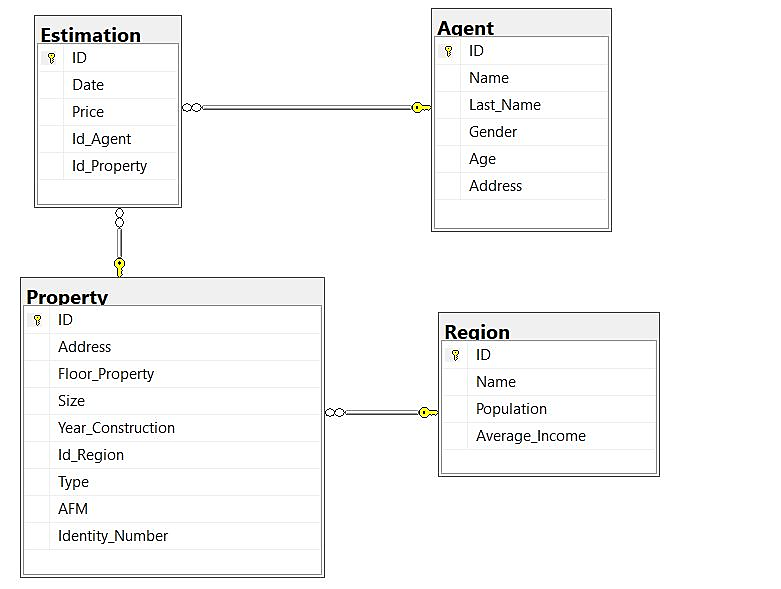


Figure 2: Relational Schema

Insert Values in Tables

The next step is to insert our data to the previous created tables. We use the statement *“INSERT INTO”* to insert a new record in the corresponding table.

It is important to note that we should follow the order we used when we created the tables because of the existing relations between them. For example, before we insert values into the Estimation table, we should first insert values into the Property table because the Estimation table contains the Property ID as a foreign key.

Following this reasoning path, we firstly inserted values into the tables Agent and Region, then into the Property table and last in the Estimation table.

Our database contains 20 Agents, 35 Properties in 20 Regions and the number of Estimations is 50.

An example of this process for the table Region is shown below.

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

The selection of the first 5 rows from Region Table will now look like this:

Εικόνα που περιέχει πίνακας

Περιγραφή που δημιουργήθηκε αυτόματα

Retrieve Data from E-Properties based on certain cases.

### **Case 1: *Properties from Regions with average income > 40.000 and estimation period between 24.12.20 and 31.12.20.***

For this problem we used the following query:

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Figure 3: Question 1



The INNER JOIN keyword selects all rows from both tables when there is a match between the columns. For instance, as far for the first inner join if we have rows in the table “Region” that do not have matches in the "Property" table, these rows will not be shown.

The WHERE keyword filters the rows and displays only those which satisfy the given conditions.

The combination of *Inner joins* and *Where* clause gives us the requested output which is shown below:

Εικόνα που περιέχει πίνακας

Περιγραφή που δημιουργήθηκε αυτόματα

Figure 4: Output Question 1

As we can see the properties that fulfill those conditions are the property with ID: 29 and Address Platwnos 276 and the property with ID: 32 and address Axarnwn 26.

### **Case 2: *Number of Estimations for the year 2020 per Agent***

For this case we used the following query.

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Figure 5: Question 2



The aggregate function COUNT() returns the number of rows that matches specified criteria of the where clause.

GROUP BY statement groups the estimations for each Agent and displays them as a summary number in a single row.

This query would produce the following result:

Εικόνα που περιέχει πίνακας

Περιγραφή που δημιουργήθηκε αυτόματα

Figure 6: Output Question 2

### **Case 3: *Number of Properties that have been estimated more than two times during 2020***

For this case we used the following query.

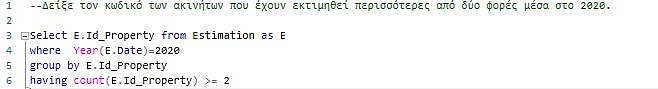


Figure 7: Question 3



We used the HAVING () statement because we wanted to add an aggregated function in it. This statement can be used instead of the WHERE after a GROUP BY statement.

In this code first we used the WHERE statement to select the estimations of the properties that have been executed in 2020. Then, the HAVING statement filters the previous result and displays only the ID of the properties that have been estimated 2 or more times.

This query would produce the following output:

Εικόνα που περιέχει πίνακας

Περιγραφή που δημιουργήθηκε αυτόματα

Figure 8: output question 3

The properties with ID: 24, 27 and 32 have been estimated two or more times during 2020.

### **Case 4: *Number of estimations in regions where the average income > 25.000€***

For this case we used the following query.

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Figure 9: Question 4



In this query, WHERE IN clause is used to filter records and returns list of all estimations where their average income is more than 25000.

At first, we create a selection from the table Region of all Regions’ IDs where the average income of those is greater than 25000. After that we select the properties that their ID\_Region is into the previous selection. At the final selection we display the Estimations’ IDs which are referenced to properties’ IDs into the second selection.

Εικόνα που περιέχει πίνακας

Περιγραφή που δημιουργήθηκε αυτόματαThis query would produce the following output:

Εικόνα που περιέχει πίνακας

Περιγραφή που δημιουργήθηκε αυτόματα

Figure 10: Output Question 4

### **Case 5: *Number of properties estimations in regions with population > 50.000 in 2020***

For this case we used the following query.

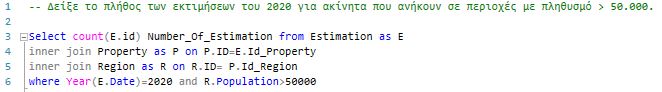


Figure 11: Question 5



In this query, WHERE clause is used to filter records and extract only rows where the year is equal to 2020 and population greater than 50000. The aggregated function COUNT is used to return the number of these rows.

This query would produce the following output:



Figure 12: Output Question 5

13 estimations have been calculated in 2020 for properties that are located into regions with population greater than 50.000.

### **Case 6: *Average estimation per square meter for each Region.***

For this case we used the following query.

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Figure 13: Question 6



The main purpose of this query is to estimate the average of the division of price with the meters squared (m²) of each property. We used the cast function to convert the outcome. Specifically, the outcome of final division:

is decimal with two digits after the decimal place.

The statement ORDER BY is used to display in ascending order the average estimations we calculated before. This query would produce the following output:

Εικόνα που περιέχει πίνακας

Περιγραφή που δημιουργήθηκε αυτόματα

Figure 14: Output question 6

For example, the average estimation price for a property in the Region with ID 4 is 2350€ per 1m2

### **Case 7: *Number of Apartments and Number of Offices each Agent has estimated.***

For this case we used the following query.

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Figure 15: Question 7



In this query we calculate the number of offices and apartments per assessor. We use two different functions to achieve this (SUM function and CASE function).

The SUM function returns the total sum of Apartments and Offices that have been estimated by a specific agent. By this way we can have only one line per assessor.

The CASE function is used to create two different columns. The first column counts the number of apartments and second the number of offices. When an Agent has estimated an Apartment, we add 1 apartment to the sum function that counts the Apartments. In the same way when an Agent estimates an Office, we add 1 office to the sum function that counts the Offices.

The query above would produce the following output:

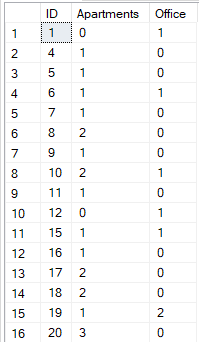


Figure 16: Output question 7

For example, the agent with ID 8 has estimated 2 apartments and 0 offices.

### **Case 8: *The change in the average Price of estimation per m2 from the year 2019 to the year 2020***

For this case we used the following query.

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Figure 17: Question 8

****

**In this query we have created** WITH **clauses.** SQL WITH clause allow us to create a temporary query block, which can be referenced in several places within our main SQL query. The first WITH query is equal to second query and the only different of them is the year of reference.

**It is also necessary to mention the use of** CASE WHEN **function, which is used in the Main Query. In this question we estimate the average change in price per region between years 2019 and 2020. If we have available data for both 2020 and 2019 the query returns the appropriate result. However, if there is an absence of information about 2019 or 2020 estimations, the code returns a default message declaring the absence of information.**

This query would produce the following output:

Εικόνα που περιέχει πίνακας

Περιγραφή που δημιουργήθηκε αυτόματα

Figure 18: Output Question 8

**For Example, as we can see for Voula Region the price decreased per 187€ per m2.**

**For the Argiroupoli Region we have no estimations during 2019 so we can not calculate the difference in price, and the “no information 2019” note is shown in the results.**

### **Case 9: *The percentage of estimations and the percentage of population per Region during 2020***

For this case we calculate the estimations as a percentage of the total estimations calculated from the regions that have been estimated during 2020. The second thing to calculate is the population as a percentage of the total population calculated from all the available regions in the dataset. Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Figure : Question 9

****

For this query we used the OVER () and the PARTITION BY statements in order to create a new

column where we need to perform an aggregation. Specifically, the first OVER() statement

creates a new column in which the count aggregation is performed in order to calculate the

number of total estimations happened in 2020. The second OVER PARTITION BY statement

performs the count aggregation on estimation IDs which are grouped by (use of partition by)

the Region ID.

**The** SELECT DISTINCT **statement is used to avoid having multiple rows for a single region.**

**The** CAST **statement is used in order to convert the calculation into the form of**

**percentage (%).**

**The query above would produce the following output.**

Εικόνα που περιέχει πίνακας

Περιγραφή που δημιουργήθηκε αυτόματα

Figure : Output Question 9

Connecting R with the database in SQL Server Management Studio

### **Case 9: *The percentage of estimations and the percentage of population per region during 2020 (using R)***

For the last case, we establish a connection between the programming language R and our database in Microsoft SQL Server Management Studio (SSMS).

Firstly, the installation of RODBC package and the odbcDriverConnect function allow us to plug in our database “E-Properties”.

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Figure 21: Connection Sql with R

Secondly, by using the function sqlSave we create a new table in our database named “TemporaryTable “. Before the use of function sqlQuery this table did not include data, it was a “blank” table with three columns and zero rows.

After the use of function sqlQuery data was inserted in our table, by using our first SQL query. To be more specific, we use the while loop in SQL to insert our records in “TemporaryTable” table.

We declare variables @codeRegion, @EstPer and @EstPop and initialize them with 0 or 1 based on the use of each one in our query. Then, a while loop is executed until the value of the @codeRegion becomes twenty (Twenty is the number of regions). In the body of the while loop, the insert query is being used to insert one record into the “TemporaryTable” table. In the ID column the @codeRegion variable is appended and for the ValueRegion and ValuePop, correspondingly, the @EstPer and @EstPop variables are appended.

Finally, by using, again, the sqlSave function we select all the records from “TemporaryTable” to extract our result.

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Figure 22: Question 9 Script in r

The argument “tableofpercentages” produces the following output:

Εικόνα που περιέχει πίνακας

Περιγραφή που δημιουργήθηκε αυτόματα

Figure : Output in R

** **

1. By clicking the below image, you will automatically open the query in the SQL Server Management Studio [↑](#footnote-ref-1)