



Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»

Фізико Технічний Інститут  
Кафедра Математичних Методів Захисту Інформації

## БАЗИ ДАННИХ ТА ІНФОРМАЦІЙНІ СИСТЕМИ РОЗРАХУНКОВА РОБОТА

Виконав:  
студент групи ФІ-73  
Чіхладзе Вахтанг Зурабович  
Варіант 16

Викладач:  
Коломицев Михайло Володимирович

Київ 2020

**Мета:** Освоєння методів проектування баз даних і роботи з базами даних у середовищі СУБД MS SQL Server.

### **Варіант 16**

Предметна область - фірма по обміну та продажу нерухомості.

Задачі, що вирішує інформаційна система - видача довідок про характеристики нерухомості (наприклад, - район, площа квартири, кількість кімнат, поверховість, ціна, адреса і т.п.), і заявках клієнтів (обмін, купівля або продаж, адреса та телефон, дата заявки)

#### **Завдання на обробку даних:**

1.1 Створити впорядковані списки:

- ПІБ клієнтів, площа квартир;
- Квартири - поверхи (угруповання за кількістю кімнат і по поверхах);

1.2 обчислення:

- Середня ціна трикімнатної квартири;
- Сумарне число заявок по районах міста.

1.3 корекція:

- Видалення відомостей про виконані заявки;
- Зміна ціни квартири заданого клієнта з ХХ на УУ.

1.4 Звіти виду:

- Координати клієнта на прізвище, по телефону;
- Квартири площею від АА до ВВ в районі вулиці Миру;
- Список квартир; угруповання по районах та кількістю кімнат;
- Лист клієнтові про знайдений варіант угоди.

### **Хід роботи**

**Завдання 1.** Промодельовати функціональні вимоги до системи, що проектується за допомогою DFD-діаграми, для чого:

- Ідентифікувати зовнішні об'єкти, що є постачальниками або приймачами інформації, і види інформації, які циркулюють між системою і зовнішніми об'єктами;
- Визначити основні функціональні вимоги до системи і представити їх у вигляді процесів;
- Визначити основні види інформації і представити їх у вигляді потоків між процесами і зовнішніми об'єктами;
- Задати сховища інформації.

Для побудови DFD-діаграми використовувати інструментальний засіб CASE Studio 2.

Я проаналізував предметну область та завдання на обробку даних. Тепер розробляємо можливу діаграму потоків даних. Для побудови DFD замість CASE Studio 2 я скористуюсь веб сервісом draw.io. Також для коматнішого і структурованого проектування визначимо підсистеми 1,2,3

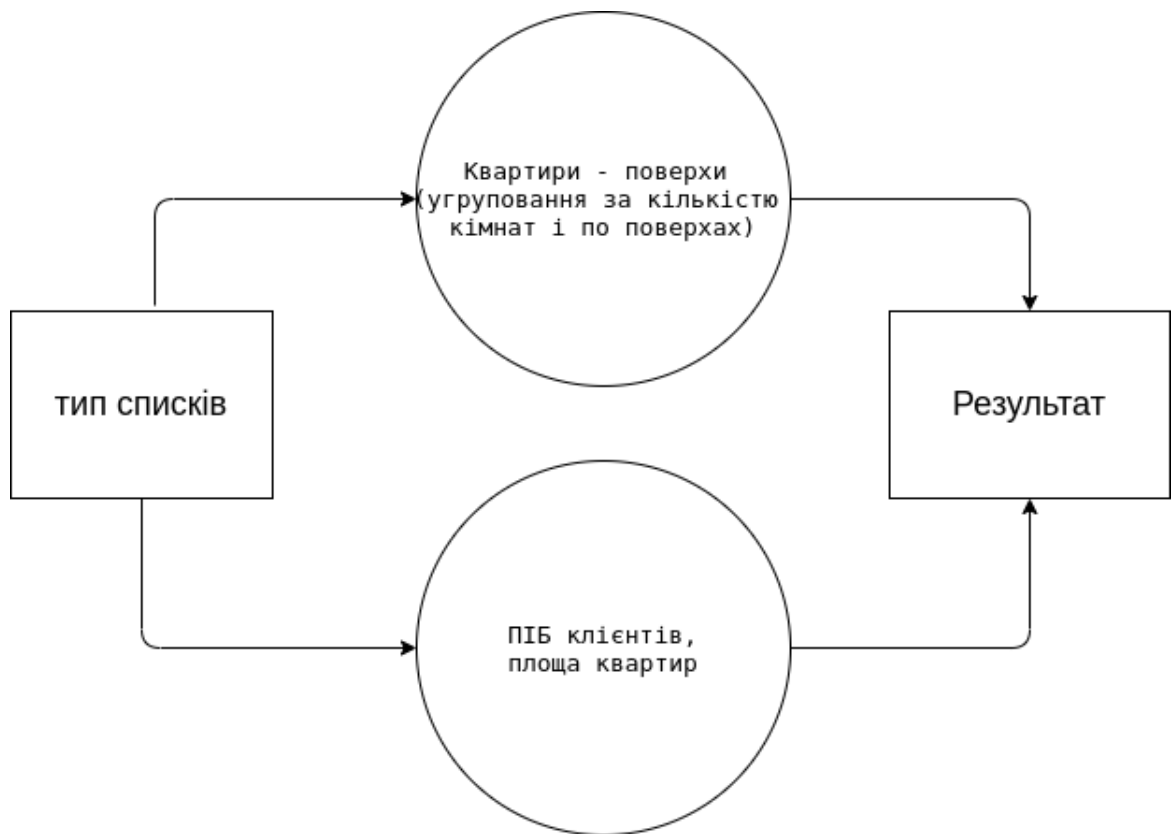


Рис 1. Підсистема "Визначення типу списку"

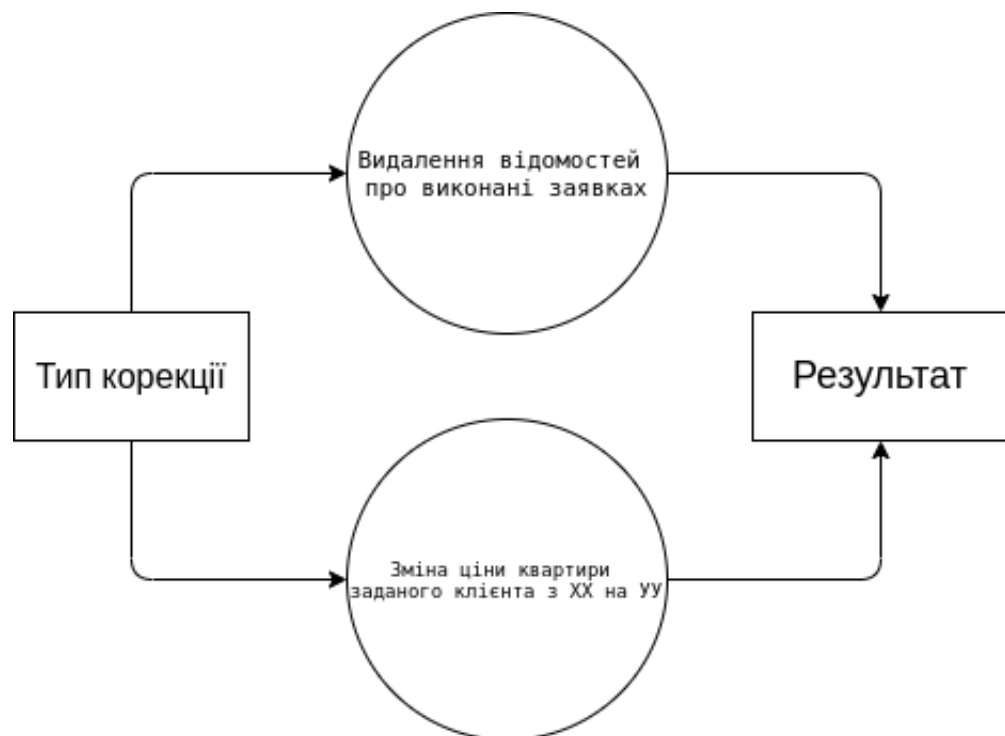


Рис 2. Підсистема "Визначення типу корекції"

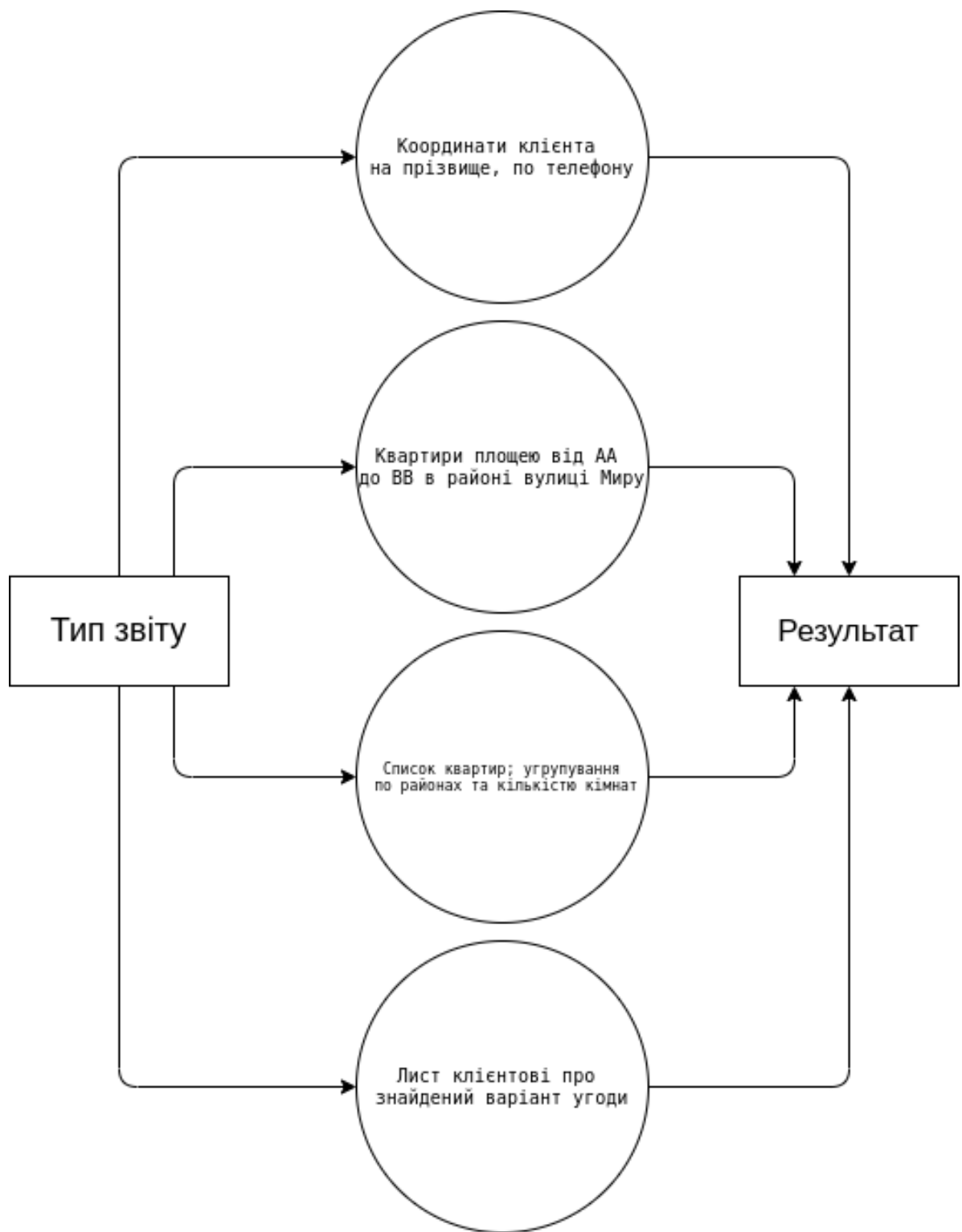


Рис 3. Підсистема "Визначення типу звіту"

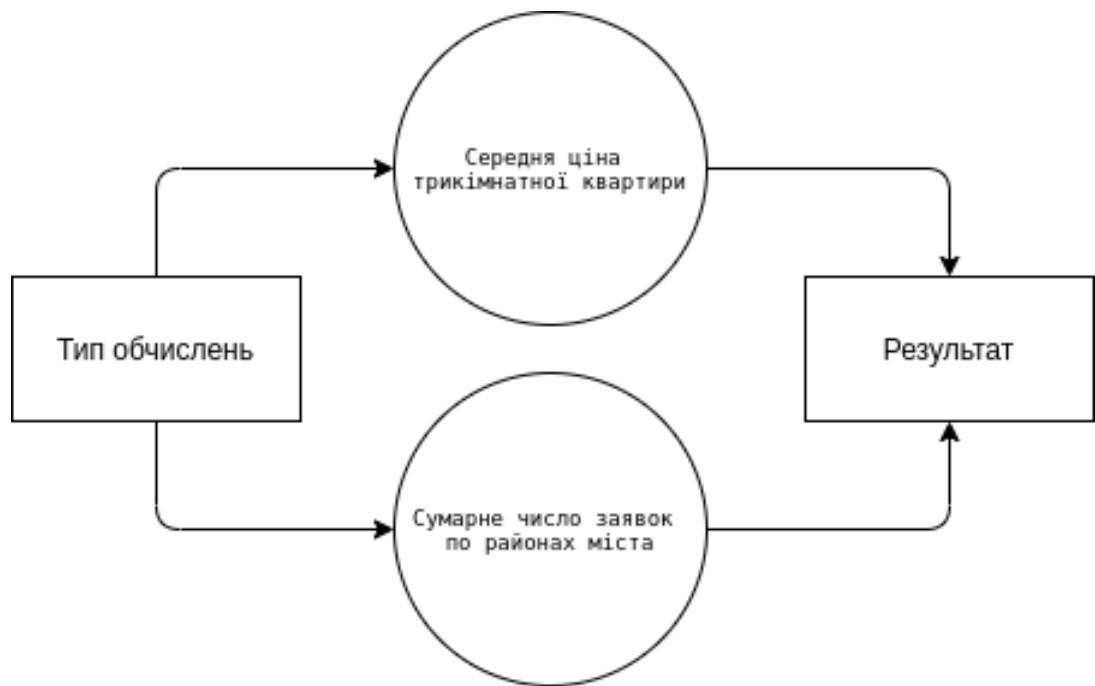


Рис 4. Підсистема "Визначення типу обчислень"

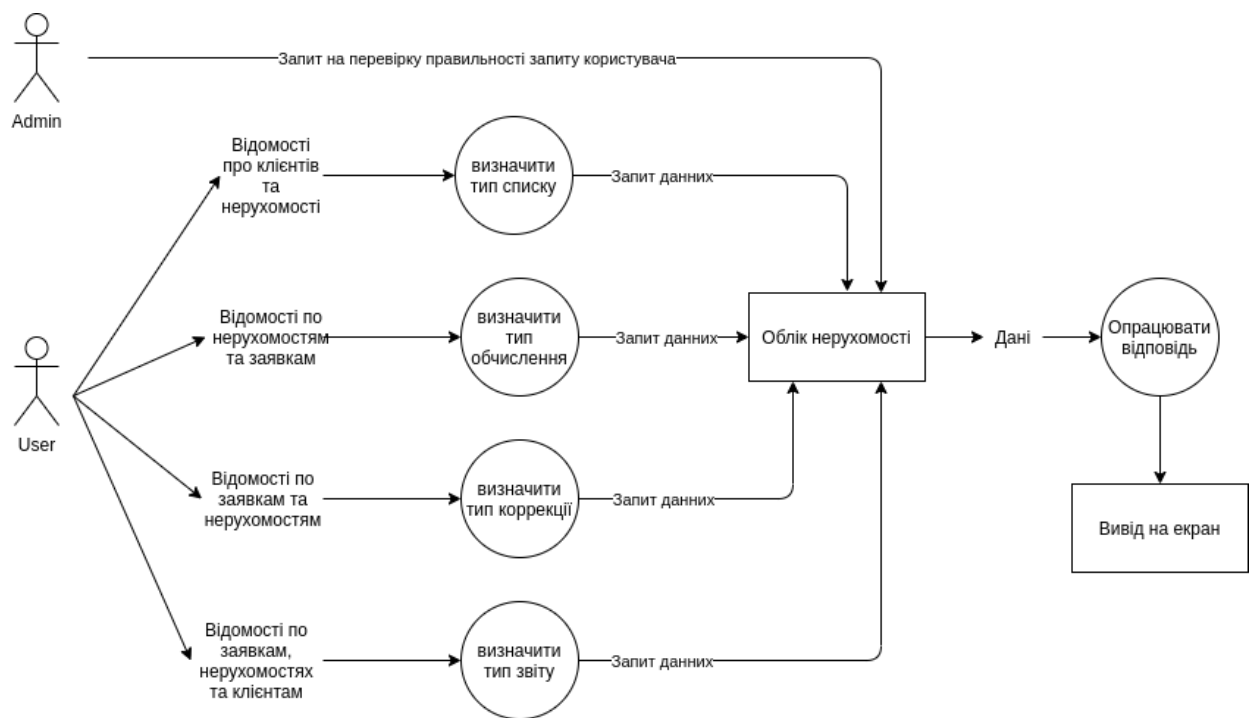


Рис 5. DFD діаграма

- Завдання 2. Побудувати концептуальну модель (ER-модель), для чого:
- За описом предметної області і поставленому завданню виявити необхідний набір сутностей, їх властивості і зв'язки між сутностями;
  - Визначити необхідний набір атрибутів для кожної сутності, виділивши атрибути що ідентифікують сутність;
  - Класифікувати сутності (стрижневі, асоціативні та інші);
  - Визначити зв'язки між об'єктами;
  - Задати обмеження цілісності;
  - Описати отриману ER-модель.

**Сутності:** {Заявки, Клієнти, Нерухомості(квартири), Операції, Регіони}

**Стрижневі:** Заявки

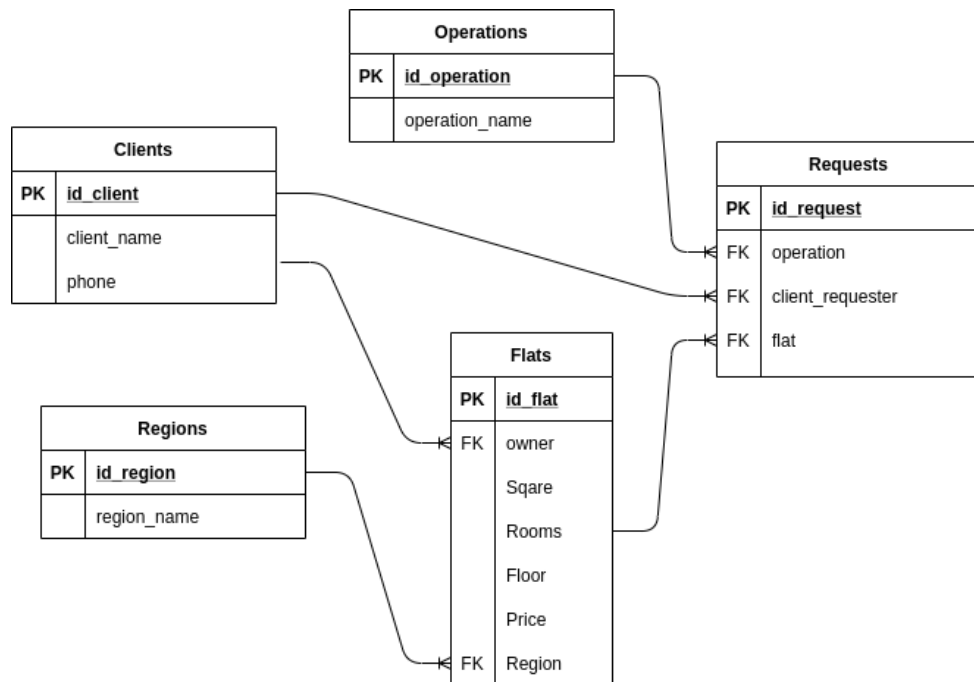


Рис 6. ER діаграма

**Обмеження:** phone має тільки цифри, Square>0, Rooms>0, Price>0

Завдання 3. Отримати реляційну схему з ER-моделі, для чого:

- Побудувати набір необхідних таблиць бази даних;
- Виділити первинні та зовнішні ключі певних таблиць;
- Привести отримані таблиці до третьої нормальної форми.
- Дати математичний опис отриманої моделі з використанням апарату реляційної алгебри.

Будемо працювати з ER-діаграмою вказаною на Рис.6. На данній діаграмі первинні ключі вказані як РК, а зовнішні ключі вказані як FK.

Модель належить до третьої нормальної форми тоді і тільки тоді коли модель належить першій і другій нормальній формі та не містить транзитивних зв'язків. Тож, треба перевірити модель на першу та другу нормальну форми.

Перевірка спроектованої ER-діаграми на першу нормальну форму:

Спроектвана мною модель належить до першої нормальної форми, оскільки усі атрибути атомарні і відсутні групи які повторюються.

Перевірка спроектованої ER-діаграми на другу нормальну форму:

Спроектвана мною модель належить до першої нормальної форми, оскільки вона належить першій нормальній формі, що ми довели у раніше, та відсутні неповні функціональні залежності не ключових атрибутів первинного ключа.

Перевірка спроектованої ER-діаграми на третю нормальну форму:

Спроектвана мною модель належить до першої нормальної форми, оскільки вона належить першій та другій нормальній формі, що ми довели у раніше, та кожне не ключове поле безпосередньо залежить від ключового поля, тобто нема транзитивних зв'язків.

Завдання 4. Використовуючи СУБД MS SQL Server створити спроектовану базу даних.

```
01 | CREATE DATABASE RieltorsDatabase
02 |
03 | GO
04 | CREATE TABLE Operations(
05 |     id_operation INT IDENTITY PRIMARY KEY,
06 |     operation_name NVARCHAR(50) NOT NULL
07 | );
08 |
09 | CREATE TABLE Regions(
10 |     id_region INT IDENTITY PRIMARY KEY,
11 |     region_name NVARCHAR(255) NOT NULL
12 | );
13 |
14 | CREATE TABLE Clients(
15 |     id_client INT IDENTITY PRIMARY KEY,
16 |     client_name NVARCHAR(255) NOT NULL,
17 |     phone NVARCHAR(15) NOT NULL,
18 |     CONSTRAINT phone CHECK(phone LIKE '[0123456789]%')
19 |
20 | );
21 |
22 | CREATE TABLE Flats(
23 |     id_flat INT IDENTITY PRIMARY KEY,
24 |     client_owner INT FOREIGN KEY REFERENCES Clients(id_client),
25 |     region INT FOREIGN KEY REFERENCES Regions(id_region),
26 |     square FLOAT NOT NULL,
27 |     rooms INT NOT NULL,
28 |     floor INT NOT NULL,
29 |     price FLOAT NOT NULL,
30 |     CONSTRAINT square CHECK(square>0.0),
31 |     CONSTRAINT rooms CHECK(rooms>=0),
32 |     CONSTRAINT price CHECK(price>0.0)
33 | );
34 |
35 | CREATE TABLE Requests(
36 |     id_request INT IDENTITY PRIMARY KEY,
37 |     operation INT FOREIGN KEY REFERENCES Operations(id_operation),
38 |     client_requester INT FOREIGN KEY REFERENCES Clients(id_client),
39 |     flat INT FOREIGN KEY REFERENCES Flats(id_flat)
40 | );
```



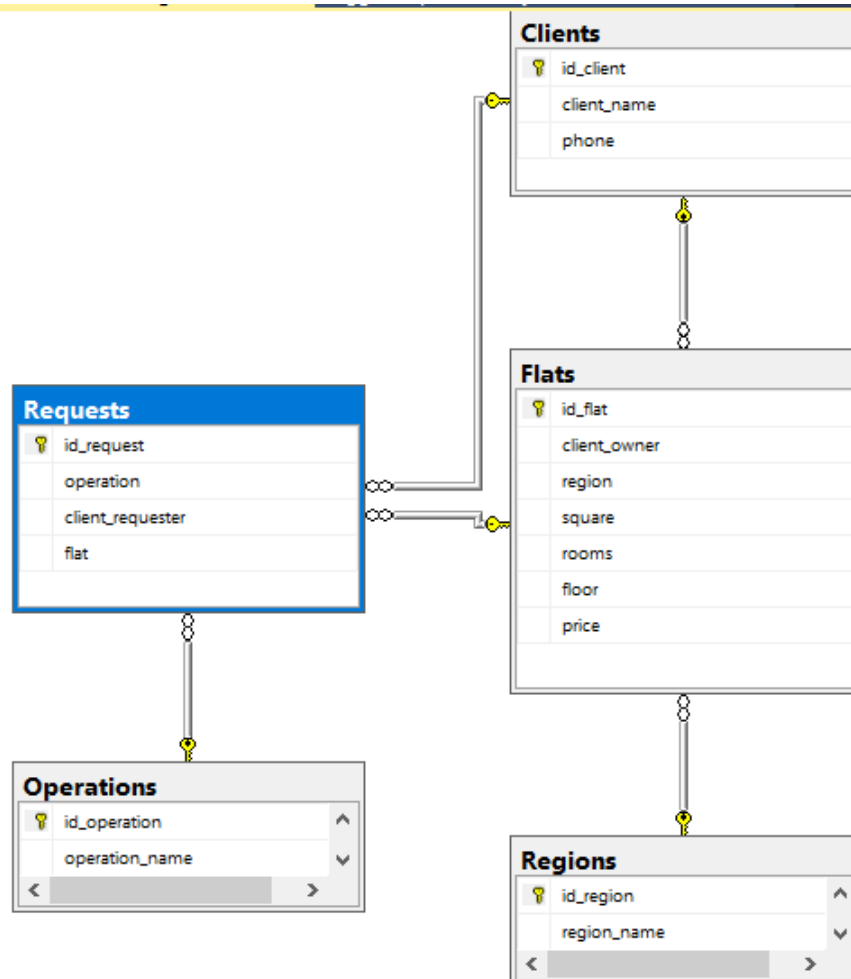


Рис 7. Побудовна ER діаграма у середовищі MSSQL

Завдання 5. На мові SQL записати вирази для зазначених у варіанті завдання типів запитів. Перевірити працездатність написаних запитів в інтерактивному режимі.

1.1 Створити впорядковані списки:

- ПІБ клієнтів, площа квартир;

```
01 | SELECT C.client_name,F.id_flat,F.square
02 | FROM Flats F INNER JOIN Clients C ON F.client_owner=C.id_client
```

RESULTS			
	client_name	id_flat	square
1	Василь Петре...	1	41,45
2	Василь Петре...	2	28,43
3	Ума Дурман	3	20,93
4	Микола Кунко	4	103,26
5	Василь Митець	5	49,81
6	Василь Митець	6	123,5
7	Семен Спасок...	7	68,3
8	Данило Галиц...	8	205,39
9	Данило Галиц...	9	48,88
10	Оксана Кновко	10	123,67
11	Захар Беркут	11	230,9
12	Захар Беркут	12	148,39
13	Катерина Гул	13	191,7
14	Катерина Гул	14	120,56

Результат виконання

- Квартири - поверхи (угруповання за кількістю кімнат і по поверхах);

```
01 | SELECT F.rooms,F.floor,count(*)[кількість таких квартир,що мають дану кількість к
    | імнат та знаходяться на даному поверсі]
02 | FROM Flats F
03 | GROUP BY F.rooms,F.floor
```

1.2 Обчислення:

- Середня ціна трикімнатної квартири;

```
01 | SELECT AVG(price)[середня ціна трикімнатної квартири]
02 | FROM Flats
03 | WHERE rooms=3
```

- Сумарне число заявок по районах міста.

```
01 | SELECT F.region,COUNT(F.region)[сумарне число заявок по районах міста]
02 | FROM Requests R INNER JOIN Flats F ON R.flat=F.id_flat
03 | GROUP BY F.region
```

RESULTS				CTRL+ALT
	rooms	floor	кількість таких квартир,що мають дану кількість кімнат та знаходяться на даному поверсі	
1	2	1	2	
2	3	1	5	
3	4	1	2	
4	5	1	1	
5	3	2	1	
6	1	4	1	
7	2	4	2	

Результат виконання

RESULTS	
	середня ціна трикімнатної квартири
1	50333,3333333333

Результат виконання

RESULTS		
	region	сумарне число заявок по районах міста
1	1	4
2	6	1

Результат виконання

1.3 Корекція:

- Видалення відомостей про виконані заявках;

Початковий стан таблиці:

01 | **SELECT \* FROM Requests**

RESULTS				
	id_request	operation	client_requester	flat
1	1	1	1	9
2	2	1	4	5
3	3	2	7	14
4	4	3	6	1
5	5	2	9	14

Результат виконання

Тепер видалимо 3й запит:

```

01 | CREATE PROC deleteRequest
02 |     @id_to_delete INT
03 | AS
04 |     DELETE FROM Requests
05 |     WHERE id_request = @id_to_delete
06 | GO

```

```

07 | GO
08 | EXEC deleteRequest 3
09 | GO
10 | SELECT * FROM Requests

```

RESULTS				
	id_request	operation	client_requester	flat
1	1	1	1	9
2	2	1	4	5
3	4	3	6	1
4	5	2	9	14

Результат виконання

- Зміна ціни квартири заданого клієнта з XX на YY.  
Початковий стан таблиці:

```

01 | SELECT * FROM Flats

```

RESULTS							CTRL+A
	id_flat	client_owner	region	square	rooms	floor	price
1	1	1	1	41,45	1	4	50000
2	2	1	2	28,43	2	1	15000
3	3	2	5	20,93	4	1	50000
4	4	3	3	103,26	3	1	30000
5	5	4	6	49,81	3	1	43000
6	6	4	2	123,5	3	1	98000
7	7	5	6	68,3	2	4	12000
8	8	6	7	205,39	2	4	67000
9	9	6	1	48,88	2	1	76000
10	10	7	2	123,67	3	1	50000
11	11	8	5	230,9	5	1	29000
12	12	8	8	148,39	3	1	45000
13	13	9	7	191,7	4	1	84000
14	14	9	1	120,56	3	2	36000

Результат виконання

Тепер змінимо ціну 1ї квартири 1го хозяїна квартири з 50000 на 45000:

```

01 | CREATE PROC changePrice
02 |     @id_client INT,
03 |     @client_flat_id INT,
04 |     @new_price FLOAT
05 | AS
06 |     UPDATE Flats
07 |     SET price=@new_price
08 |     WHERE client_owner=@id_client AND id_flat=@client_flat_id
09 | GO
10 | EXEC changePrice 1,1,45000
11 | GO
12 | SELECT * FROM Flats

```



- Квартири площею від АА до ВВ в районі вулиці Миру;

```
01 | CREATE PROC getFlats
02 |     @low FLOAT,
03 |     @high FLOAT,
04 |     @region INT
05 | AS
06 |     SELECT id_flat
07 |     FROM Flats
08 |     WHERE region=@region AND (square BETWEEN @low AND @high)
09 | GO
10 | EXEC getFlats 45.0,200.0,1
```

RESULTS	
	id_flat
1	9
2	14

Результат виконання

- Список квартир; угруповання по районах та кількістю кімнат;

```
01 | SELECT region,rooms,COUNT(*) [кількість таких квартир,що мають дану кількість кімн
    |       at та знаходяться в одному регіоні]
02 | FROM Flats
03 | GROUP BY region,rooms
```

RESULTS				CTRL+ALT
	region	rooms	кількість таких квартир,що мають дану кількість кімнат та знаходяться в одному регіоні	
1	1	1	1	
2	1	2	1	
3	2	2	1	
4	6	2	1	
5	7	2	1	
6	1	3	1	
7	2	3	2	
8	3	3	1	
9	6	3	1	
10	8	3	1	
11	5	4	1	
12	7	4	1	
13	5	5	1	

Результат виконання

- Лист клієнтові про знайдений варіант угоди.

```
01 | SELECT R.client_requester,F.client_owner
02 | FROM Requests R INNER JOIN Flats F ON R.flat=F.id_flat
```

RESULTS		
	client_requester	client_owner
1	1	6
2	4	4
3	6	1
4	9	9

Результат виконання

В даній системі цим двом клієнтам одсилається лист,що відповідний юзер хоче зробити якусь операцію.

Завдання 6. Реалізувати закінчене додаток, що працює з створеною базою даних. Додаток має

- Заносити інформацію в таблиці створеної бази даних;
- Виконувати необхідні дії по модифікації і видалення даних в таблицях створеної бази даних;
- Підтримувати цілісність бази даних;
- Виконувати запити з варіанта завдання.

Ці процедури додають інформацію до створених таблиць бах даних:

```
01 | USE RieltorsDatabase
02 | GO
03 |
04 | CREATE PROC addClient
05 |     @client_name NVARCHAR(255),
06 |     @phone NVARCHAR(15)
07 | AS
08 |     INSERT INTO Clients(client_name, phone) VALUES
09 |     (@client_name,@phone)
10 | GO
11 |
12 | CREATE PROC addFlat
13 |     @client_id INT,
14 |     @region_id INT,
15 |     @square FLOAT,
16 |     @rooms INT,
17 |     @floor INT,
18 |     @price FLOAT
19 | AS
20 |     INSERT INTO Flats(client_owner,region,square,rooms,floor,price) VALUES
21 |     (@client_id,@region_id,@square,@rooms,@floor,@price)
22 | GO
23 |
24 | CREATE PROC addOperation
25 |     @operation_name NVARCHAR(50)
26 | AS
27 |     INSERT INTO Operations(operation_name) VALUES
```

```

28 |         (@operation_name)
29 | GO
30 |
31 | CREATE PROC addRegion
32 |     @region_name NVARCHAR(255)
33 | AS
34 |     INSERT INTO Regions(region_name) VALUES
35 |         (@region_name)
36 | GO
37 |
38 | CREATE PROC addRequest
39 |     @operation INT,
40 |     @client_id INT,
41 |     @flat INT
42 | AS
43 |     INSERT INTO Requests(operation,client_requester,flat) VALUES
44 |         (@operation,@client_id,@flat)
45 | GO

```

Необхідні дії по модифікації даних в базі даних було реалізовано в завданні 5.  
Цілісність даних була забезпечена на етапі створення таблиць в завданні 4.  
Виконання запитів було виконано в інтерактивному режимі в завданні 5.