

Подзапросы (вложенные запросы, подчинённые запросы)

***SELECT* Студент.Фамилия
FROM Студент *LEFT JOIN* Оценки *ON*
Студент.[код студента] =
Оценки.[Код студента]
WHERE [Код_ дисциплины] *IS NULL*;**

Или:

***SELECT* Студент.Фамилия
FROM Студент *WHERE* [код студента]
NOT IN (*SELECT* [код студента] *FROM*
оценки)**

Подзапрос создаёт временную таблицу, содержимое которой извлекается и обрабатывается внешним оператором (обычно предикатом внешнего запроса в предложении **WHERE** или **HAVING**).

Текст подзапроса должен быть заключён в круглые скобки.

Подзапрос может вернуть следующее число значений:

- **одно значение,**
- **столбец значений,**
- **таблицу значений (несколько столбцов):**

1. При возврате **одного** значения обычно используются операторы сравнения.

Например, вывести данные об объектах недвижимости, цены которых не превышают средней цены объектов.

```
SELECT * FROM PROPERTY
```

```
WHERE Selling_price < =
```

```
(SELECT AVG(selling_price) FROM  
PROPERTY)
```

SELECT [код студента]

FROM оценки

GROUP BY [код студента]

***HAVING AVG(оценка) > (SELECT
AVG(оценка) FROM оценки)***

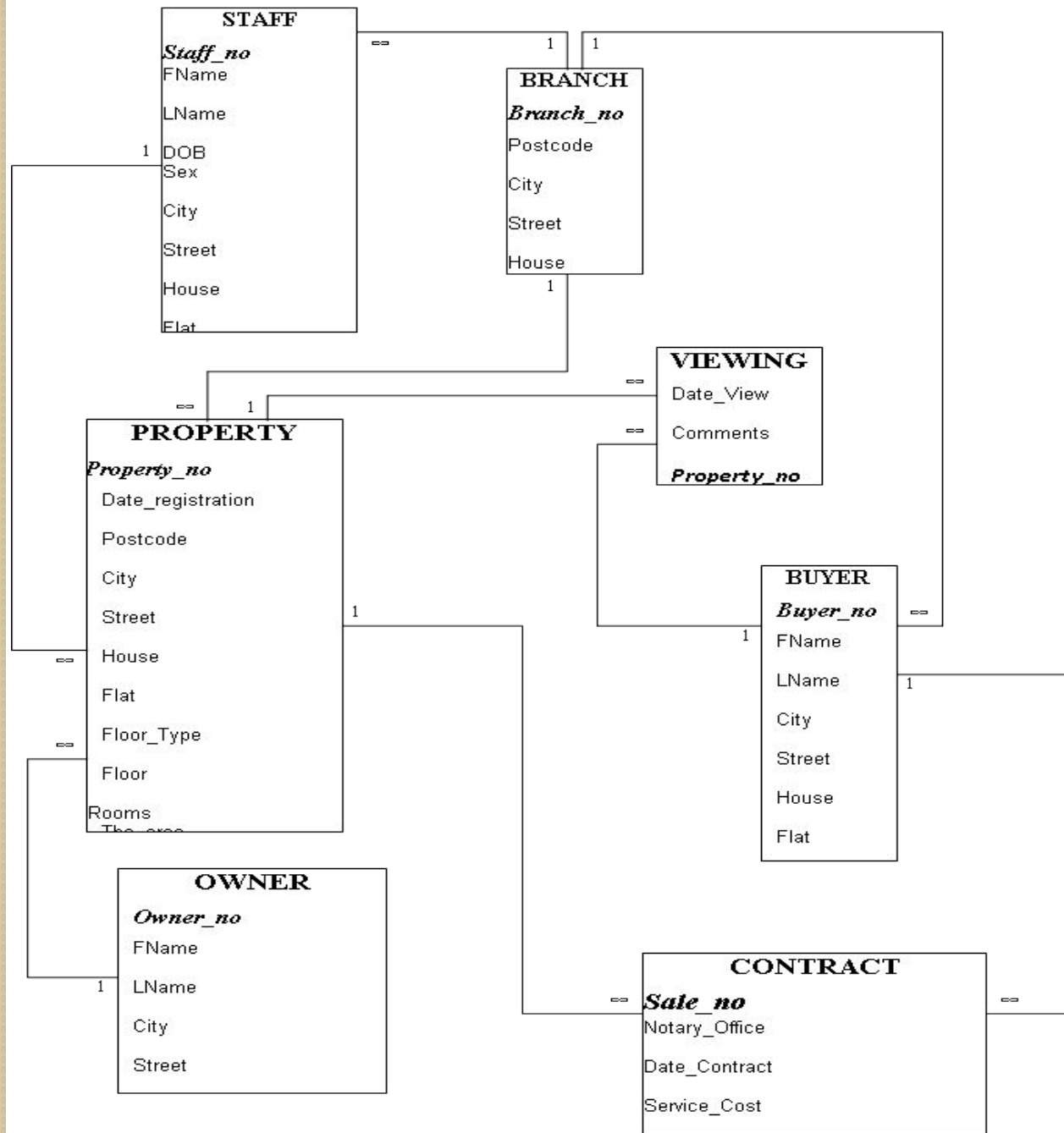
2. При возврате множества значений (одного столбца) используется проверка на принадлежность к множеству (***IN***), а также операторы **ANY** и **ALL**, которые используются совместно с операторами сравнения.

ANY(SOME) – условие верно, если хотя бы одно значение, которое вернул подзапрос, удовлетворяет заданному условию;

ALL – условие верно, если все значения, которые вернул подзапрос, удовлетворяют заданному условию.

Например, следующий запрос
вернёт сведения об объектах
собственности, **осмотренных**
покупателями.

```
SELECT * FROM PROPERTY  
WHERE Property_No IN (SELECT  
Property_No FROM Viewing);
```

Или

```
SELECT * FROM PROPERTY  
WHERE Property_No = ANY  
(SELECT Property_No FROM  
Viewing);
```

Однако, оператор **ANY** может использовать другие операторы сравнения кроме равенства, и таким образом делать сравнения которые являются выше возможностей **IN**.

Следующий запрос возвращает фамилии тех сотрудников отделения 2, которые получают более высокую заработную плату, чем какой-либо из сотрудников отделения 1.

```
SELECT Fname  
FROM STAFF  
WHERE Branch_No = 2 AND  
SALARY > ANY  
(SELECT SALARY FROM STAFF  
Where Branch_no=1);
```

FNAME Branch_no SALARY

Батуркин	3	2500000,00
Чубаро	1	5060000,00
Коваленко	3	2500000,00
Логинов	1	2000000,00
Суворов	3	3800000,00
Ганущенко	2	1800000,00
Жарков	2	4200000,00
Сотникова	3	7000000,00
Янчиленко	2	1500000,00

FNAME

Жарков

ALL:

предикат является верным,
если каждое значение
выбранное подзапросом
удовлетворяет условию в
предикате внешнего запроса.

Следующий запрос возвращает фамилии тех сотрудников отделения 2, которые получают более высокую заработную плату, чем любой из сотрудников отделения 1.

1)

SELECT Fname

FROM STAFF

WHERE Branch_No = 2 AND SALARY

> ALL

(SELECT SALARY FROM STAFF

Where Branch_no=1);

2)

```
SELECT Fname  
FROM STAFF  
WHERE Branch_No = 2 AND  
SALARY >  
(SELECT MAX(SALARY)FROM  
STAFF Where Branch_no=1);
```


ALL используется в основном с неравенствами так как значение может быть "равным для всех" результатом подзапроса только если все результаты идентичны.

3. При возврате подзапросом таблицы (множество столбцов) можно проверить только факт наличия данных с помощью оператора **EXISTS** (если подзапрос ничего не возвращает, то результат - ложь)

```
SELECT *  
FROM T1  
WHERE EXISTS (SELECT * FROM T2);
```

Нет смысла использовать
EXISTS, если подзапрос
построен с помощью
обобщающей функции,
которая всегда возвращает
значение.

В предложении **HAVING** также могут использоваться подзапросы.

Например: вывести список **отделений** компании, в которых средняя заработная плата сотрудников превышает среднюю заработную плату всех сотрудников компании.

```
SELECT branch_no, avg(salary)
```

```
from staff
```

```
group by branch_no
```

```
having avg(salary) > (select
```

```
avg(salary) from staff )
```

Вывести список сотрудников, зарплата которых выше средней но ниже зарплаты директора компании

SELECT Staff_no, SALARY

FROM STAFF

WHERE SALARY BETWEEN

(SELECT AVG(SALARY) FROM STAFF)

AND

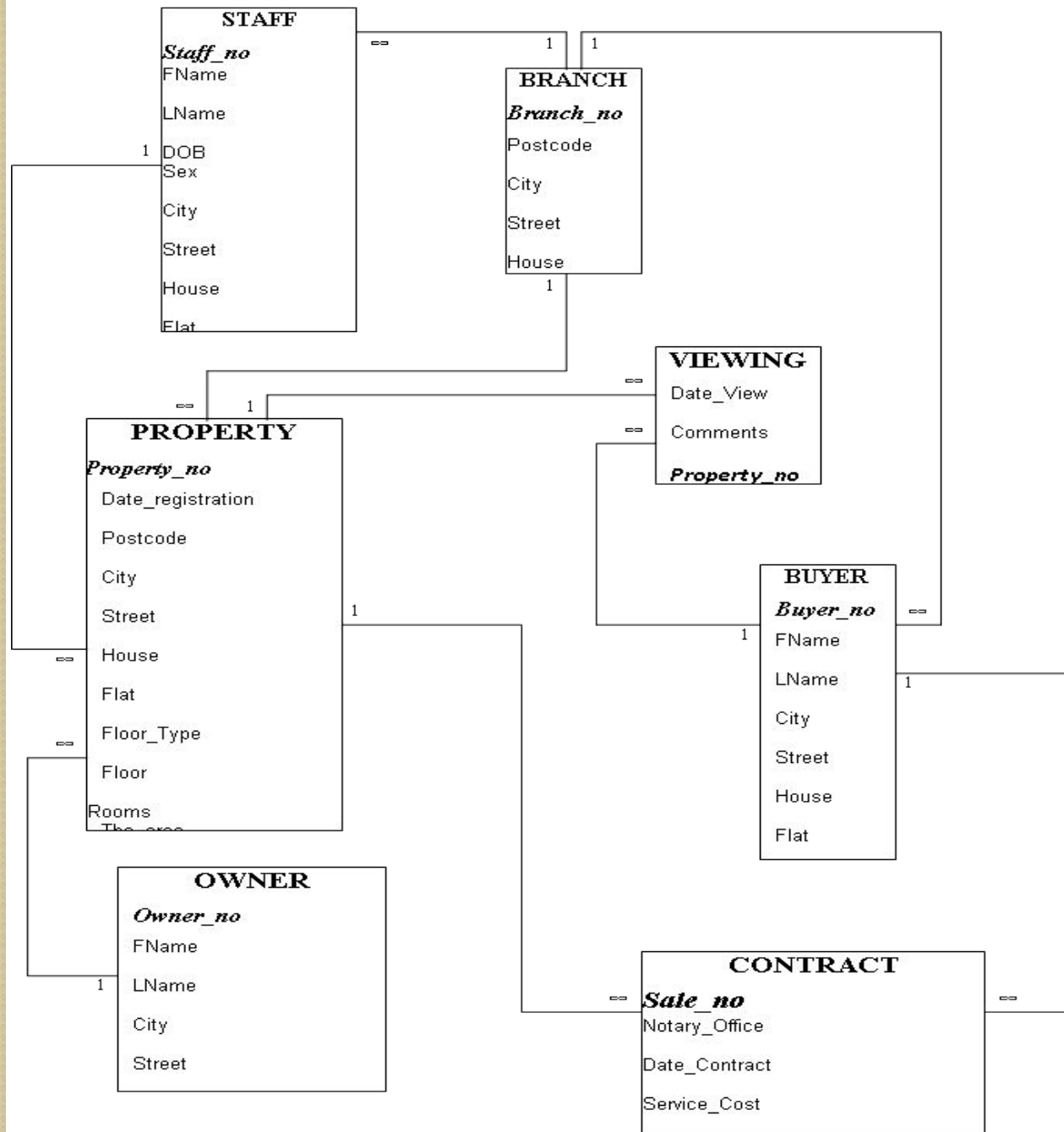
(SELECT SALARY FROM STAFF WHERE

Position='директор')

Связанные (соотнесенные) подзапросы

Возможны случаи, когда подзапрос должен использовать данные из внешнего запроса.

Пример. Вывести данные тех агентов из таблицы ***Staff***, в ведении которых находится ровно два объекта.



Запрос

```
SELECT *  
FROM STAFF  
WHERE 2 IN (SELECT  
COUNT(*) FROM Property  
GROUP BY Staff_no)
```

вернет список **всех** агентов (**Staff_no**), имеющих в таблице **Property**, если хотя бы у одного из агентов имеется в ведении два объекта.


```
SELECT *  
FROM STAFF  
WHERE 2 = (SELECT  
COUNT(*) FROM Property  
GROUP BY Staff_no)
```

Для того чтобы получить требуемый ответ на запрос необходимо использовать связанный подзапрос:

```
SELECT *  
FROM Staff a  
WHERE 2 = (SELECT  
COUNT(*) FROM Property  
WHERE  
staff_no=a.staff_no)
```

Просматривается таблица **Staff**, из нее берется одна очередная запись и переписывается в таблицу с именем **a**.

Для этой записи выполняется подзапрос – подсчитывается количество записей таблицы **Property**, приходящихся на агента с данным **Staff_no**. Далее проверяется, равно ли это количество 2 и, если это условие выполняется, то запись кандидат заносится в выходной набор. Затем берется следующая запись из таблицы **Staff**.

а

ВМ0550262

Семенов

...

Property

3000	1/18/10 Б	Т	60000,0000	1	ВМ0550262	1
3001	37/21/7 Бз		35000,0000	1	ВМ0550262	5
3002	29/9 Лз	Т	92000,0000	2	ВМ0550266	7
3003	40/7,6 2Бз	-	14850,0000	1	ВМ0550260	5

Этот же результат может быть получен при помощи более простого запроса:

```
SELECT Staff_no FROM  
Property  
GROUP BY Staff_no  
HAVING COUNT(*)=2
```

Вывести список сотрудников, зарплата которых выше средней зарплаты сотрудников своего отделения

```
SELECT FNAME, Branch_no  
SALARY  
FROM STAFF S  
WHERE SALARY >  
(SELECT AVG(SALARY) FROM  
STAFF  
WHERE  
Staff.Branch_no=S.Branch_no)
```

S

FNAME	Branch_no	SALARY
Батуркин	3	2500000,00

Staff

FNAME	Branch_no	SALARY
Батуркин	3	2500000,00
Чубаро	1	5060000,00
Коваленко	3	2500000,00
Логинов	1	2000000,00
Суворов	3	3800000,00
Жарков	2	4200000,00
Ганущенко	2	1800000,00
Сотникова	3	7000000,00
Янчиленко	2	1500000,00

Пример связанного запроса:

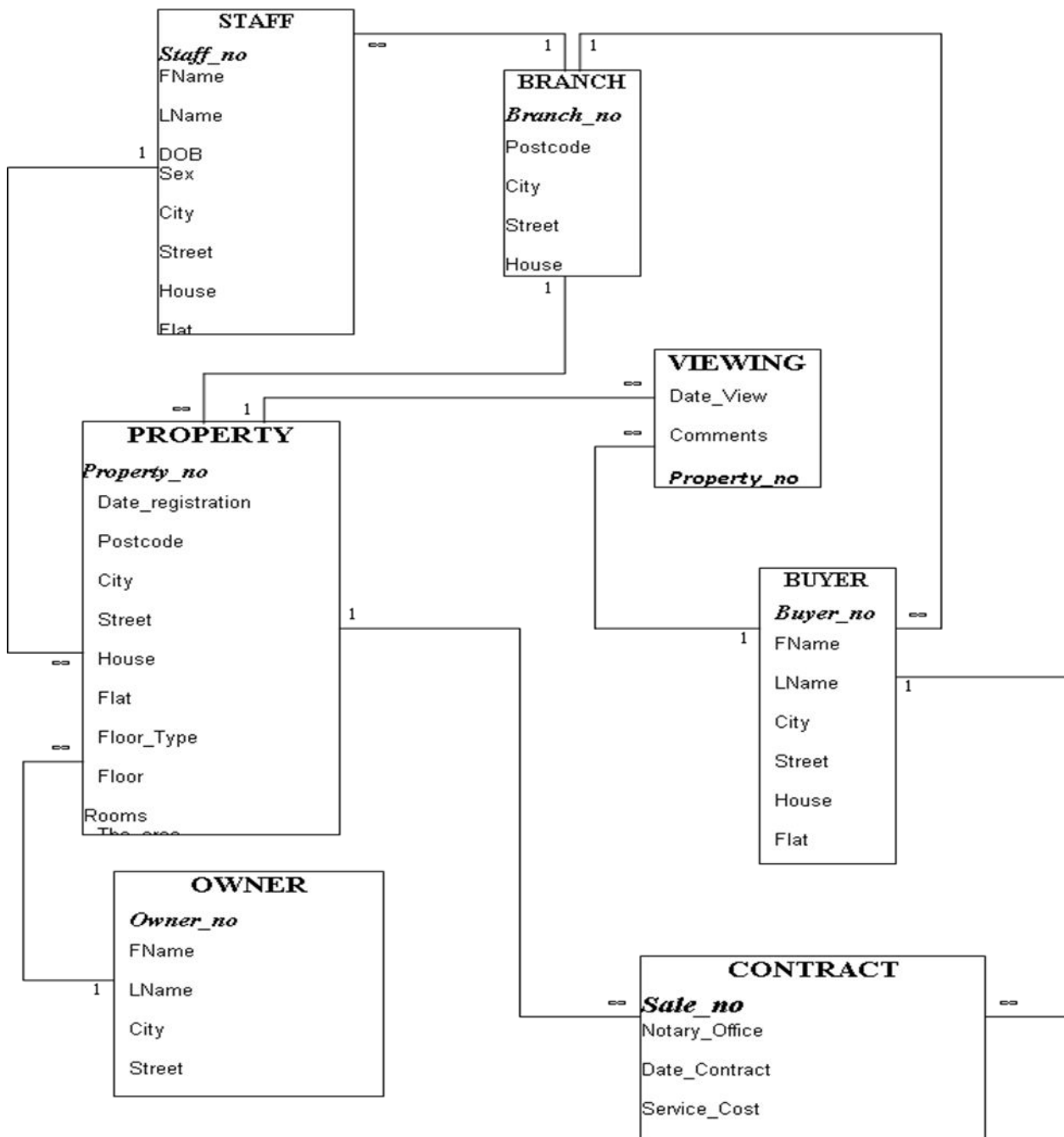
Вывести список продавцов, имеющих объекты собственности в Витебске

```
SELECT *  
FROM Owner  
WHERE 'Витебск' IN  
(SELECT City FROM Property WHERE  
Property.Owner_no =  
Owner.Owner_no)
```


Примеры.

1. Вывести список владельцев собственности, чьи объекты были осмотрены в определенный день:

```
SELECT OWNER.Owner_no, FName, LName  
FROM OWNER INNER JOIN PROPERTY  
ON  
PROPERTY.Owner_no=OWNER.Owner_no  
WHERE PROPERTY.Property_no=(SELECT  
Property_no  
FROM VIEWING  
WHERE Date_View='18.01.11');
```



В таблице **VIEWING** будет найдена соответствующая дата и передана в предложение **WHERE**. После определения даты в основном запросе из таблицы **PROPERTY** будут отображены записи, удовлетворяющие заданному условию.

(В данном примере предполагается, что подзапрос должен вернуть только одно значение).

Использование оператора ***EXISTS***

Оператор ***EXISTS*** проверяет, возвращает ли подчиненный запрос хотя бы одну строку. Для проверки противоположного значения используется предикат ***NOT EXISTS***.

Пример.

Вывести данные об объектах собственности из таблицы ***PROPERTY*** только в том случае, если хотя бы один из них был осмотрен покупателями, и было получено согласие на приобретение:

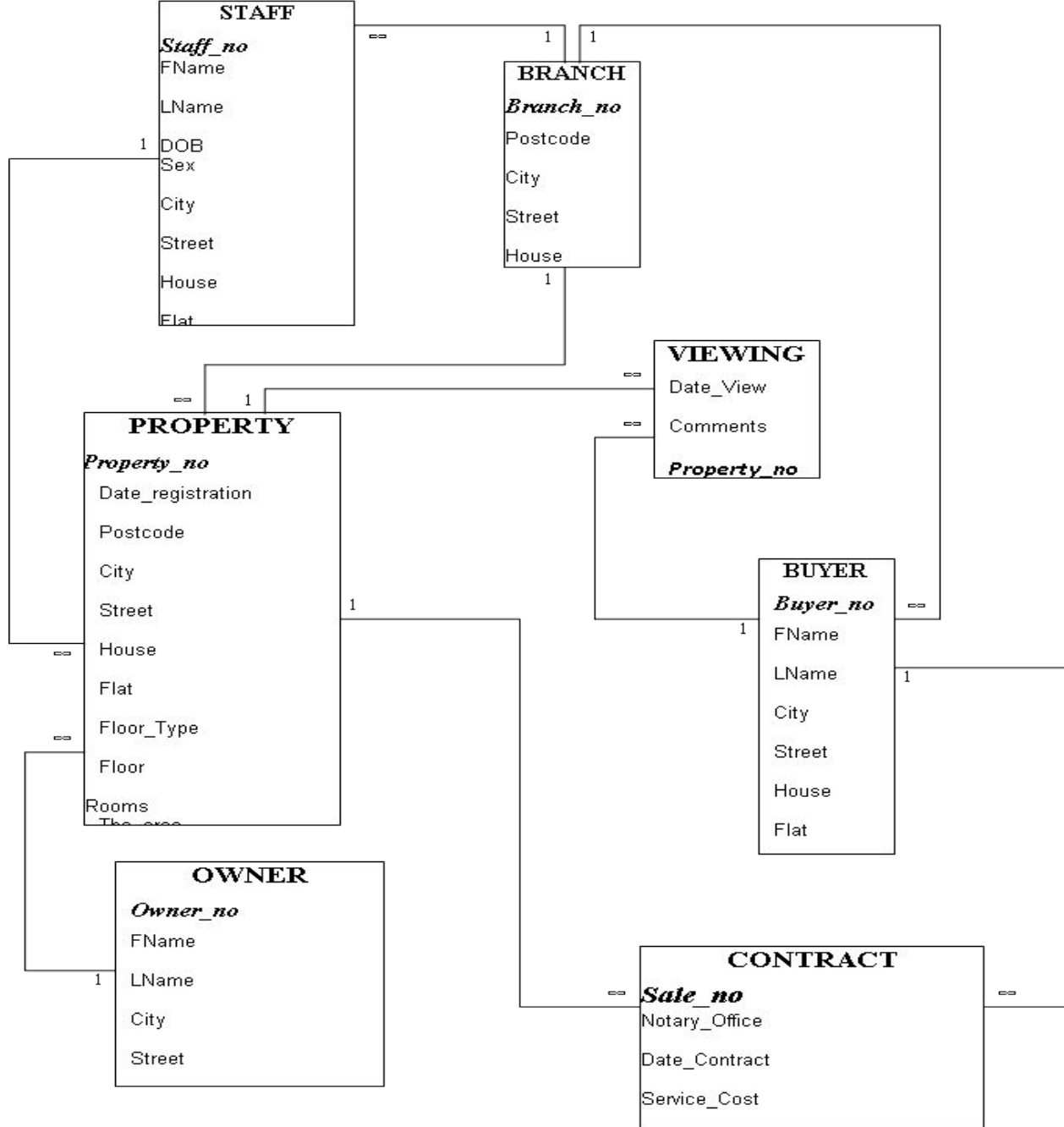
```
SELECT *  
FROM PROPERTY  
WHERE EXISTS (SELECT  
Property_no FROM VIEWING  
WHERE Comments='согласен');
```

- Property

<i>Property_no</i>	<i>Selling_price</i>	<i>Rooms</i>	<i>Staff_no</i>	<i>Branch_no</i>
<i>3000</i>	<i>60000,0000</i>	<i>1</i>	<i>BMO550262</i>	<i>1</i>
<i>3001</i>	<i>35000,0000</i>	<i>1</i>	<i>BMO550262</i>	<i>5</i>
<i>3002</i>	<i>92000,0000</i>	<i>2</i>	<i>BMO550266</i>	<i>7</i>
<i>3003</i>	<i>15000,0000</i>	<i>1</i>	<i>BMO550262</i>	<i>5</i>
<i>3005</i>	<i>75000,0000</i>	<i>3</i>	<i>BMO550260</i>	<i>6</i>
<i>3006</i>	<i>2000,0000</i>	<i>3</i>	<i>BMO550264</i>	<i>3</i>
<i>3007</i>	<i>2000,0000</i>	<i>3</i>	<i>BMO550267</i>	<i>2</i>

- VIEWING

<i>Date_view</i>	<i>Comments</i>	<i>Property_no</i>
<i>2003-03-31</i>	<i>согласен</i>	<i>3000 4</i>
<i>2012-03-25</i>	<i>требует ремонта</i>	<i>3001 7</i>
<i>2012-03-25</i>	<i>согласен</i>	<i>3002 1</i>
<i>2012-01-18</i>	<i>не согласен</i>	<i>3002 4</i>
<i>2012-01-17</i>	<i>согласен</i>	<i>3003 1</i>
<i>2012-01-19</i>	<i>согласен</i>	<i>3005 2</i>




Вывести данные об объектах
собственности из таблицы ***PROPERTY***
проданных покупателям


```
SELECT *  
FROM PROPERTY  
WHERE EXISTS (SELECT  
Property_no FROM VIEWING  
WHERE Comments='согласен'  
AND  
property.property_no=viewing.  
property_no)
```

```
SELECT *  
FROM PROPERTY INNER JOIN  
VIEWING ON  
Property.Property_no =  
VIEWING.Property_no  
WHERE Comments = 'согласен'
```

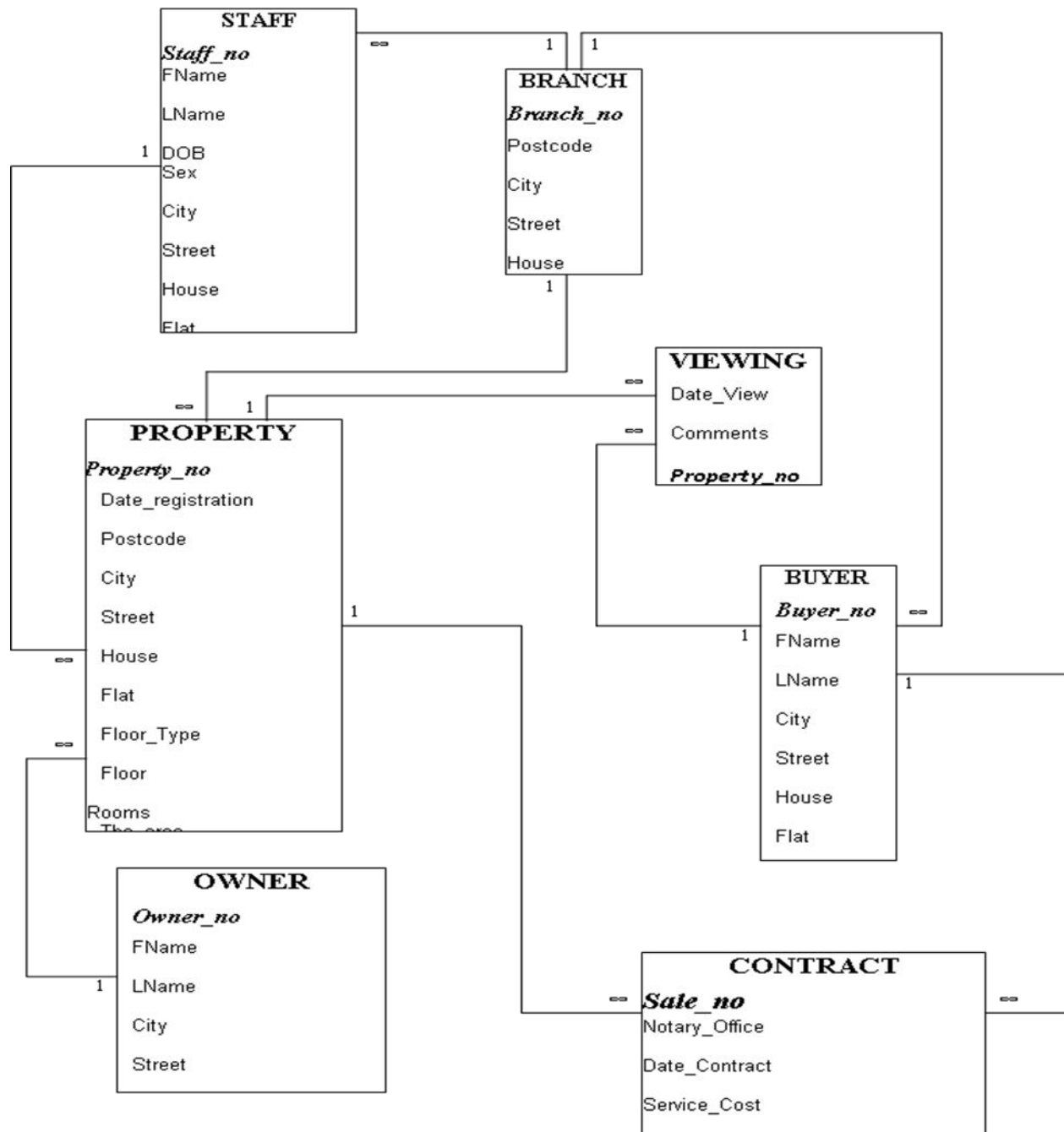
SELECT *
FROM PROPERTY WHERE
Property_no in (SELECT
Property_no FROM VIEWING
WHERE
Comments='согласен')

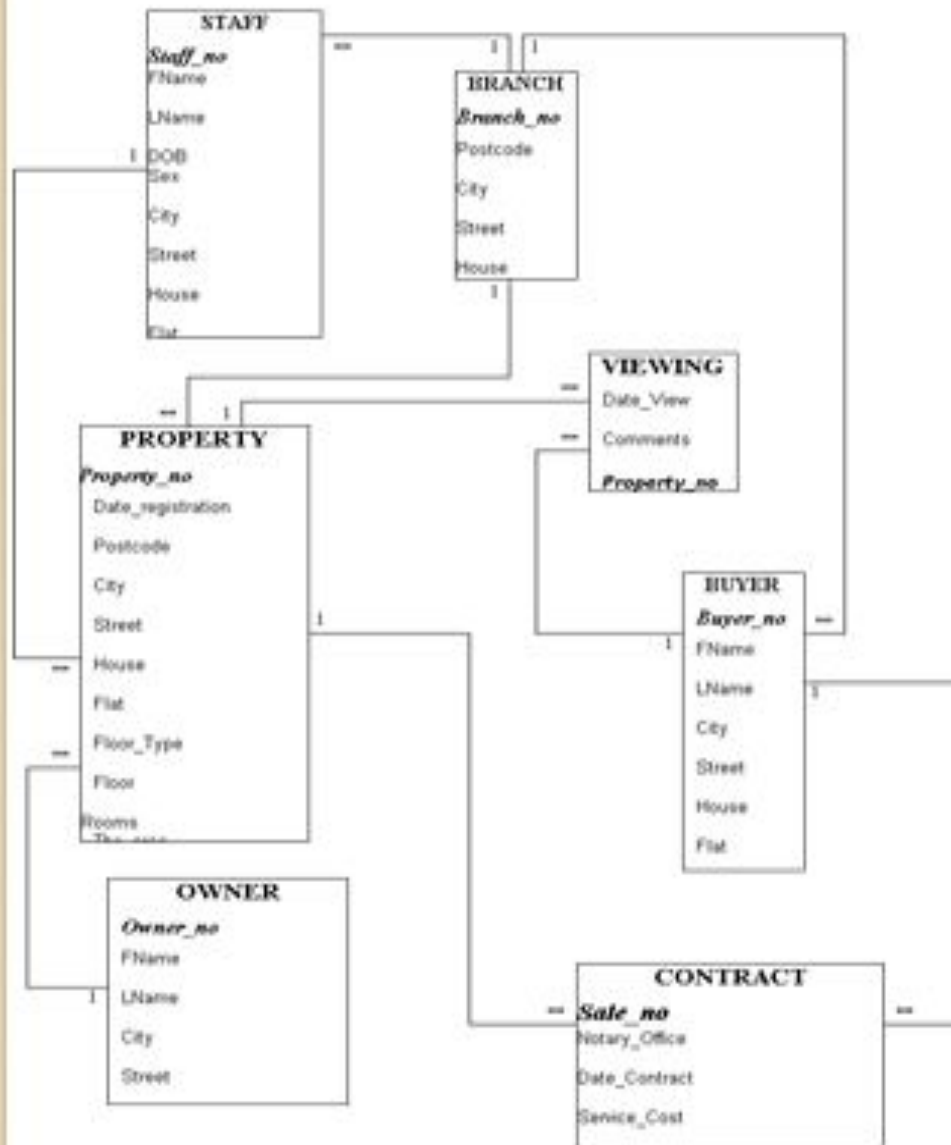


```
SELECT *  
FROM PROPERTY WHERE  
Property_no = ANY(SELECT  
Property_no FROM VIEWING  
WHERE Comments='согласен')
```

Найти номера телефонов отделений, среди служащих которых имеются менеджеры.

```
SELECT Btel_no  
FROM Branch  
WHERE EXISTS  
(SELECT Staff_no  
FROM Staff  
WHERE position='менеджер'  
AND Staff.Branch_No =  
Branch.Branch_No )
```





Запросы с предикатом **EXISTS** можно переформулировать в виде запросов с предикатом сравнения

```
SELECT *  
FROM Branch  
WHERE  
(SELECT count(*)  
FROM Staff  
WHERE (position =  
'менеджер') AND  
(Staff.Branch_No =  
Branch.Branch_No)) > 0
```


Порождаемые таблицы

Найти максимальную из средних
заработных плат отделений
компании

```
SELECT  
max(avg(SALARY) FROM  
STAFF GROUP BY  
branch_no)
```

Staff_no

FName

LName

DOB

Sex

City

Street

House

Flat

Stel_no

Date_Joined

Position

Salary

STAFF

Branch_no

Branch_no	Avgsal
1	3250000,00
2	2600000,00
3	7575000,00
4	1500000,00

```
SELECT max(avgсал)  
FROM (SELECT  
Branch_no,  
AVG(salary)AS avgсал  
FROM staff GROUP BY  
branch_no)...
```

```
SELECT max(avgsal)  
FROM (SELECT Branch_no,  
AVG(salary)AS avgsal FROM  
staff GROUP BY branch_no)  
AS tbl
```

```
SELECT max(avgsal)  
FROM (SELECT Branch_no,  
AVG(salary)FROM staff  
GROUP BY branch_no)  
AS tbl(br,avgsal)
```

Существует возможность указывать в разделе **FROM** не только ссылки на таблицы, но и запросы.

Результатом вычисления выражения запросов в SQL является таблица. Следовательно, в любой конструкции языка, где может присутствовать ссылка на таблицу SQL, допускается присутствие выражения запросов.

Вывести список объектов собственности, которые не осматривались покупателями

Property

<i>Property_no</i>	<i>Selling_price</i>	<i>Rooms</i>	<i>Staff_no</i>	<i>Branch_no</i>
<i>3000</i>	<i>60000,0000</i>	<i>1</i>	<i>ВМО550262</i>	<i>1</i>
<i>3001</i>	<i>35000,0000</i>	<i>1</i>	<i>ВМО550262</i>	<i>5</i>
<i>3002</i>	<i>92000,0000</i>	<i>2</i>	<i>ВМО550266</i>	<i>7</i>
<i>3003</i>	<i>15000,0000</i>	<i>1</i>	<i>ВМО550262</i>	<i>5</i>
<i>3005</i>	<i>75000,0000</i>	<i>3</i>	<i>ВМО550260</i>	<i>6</i>
<i>3006</i>	<i>2000,0000</i>	<i>3</i>	<i>ВМО550264</i>	<i>3</i>
<i>3007</i>	<i>2000,0000</i>	<i>3</i>	<i>ВМО550267</i>	<i>2</i>

<i>Date_view</i>	<i>Comments</i>	<i>Property_no</i>	
<i>2003-03-31</i>	<i>согласен</i>	<i>3000</i>	<i>4</i>
<i>2012-03-25</i>	<i>требует ремонта</i>	<i>3001</i>	<i>7</i>
<i>2012-03-25</i>	<i>согласен</i>	<i>3002</i>	<i>1</i>
<i>2012-01-18</i>	<i>не согласен</i>	<i>3002</i>	<i>4</i>
<i>2012-01-17</i>	<i>согласен</i>	<i>3003</i>	<i>1</i>
<i>2012-01-19</i>	<i>согласен</i>	<i>3005</i>	<i>2</i>

I)

SELECT Property_no

FROM PROPERTY

WHERE PROPERTY_No NOT IN

(SELECT Property_no FROM

VIEWING)

2)

```
SELECT PROPERTY.Property_no  
FROM PROPERTY LEFT JOIN VIEWING  
ON PROPERTY.Property_no=  
VIEWING.Property_no  
WHERE VIEWING.Property_No IS NULL
```

3)

SELECT Property_no

FROM PROPERTY

WHERE NOT EXISTS

(SELECT PROPERTY_no

FROM VIEWING)

3)

SELECT Property_no

FROM PROPERTY

WHERE NOT EXISTS (SELECT

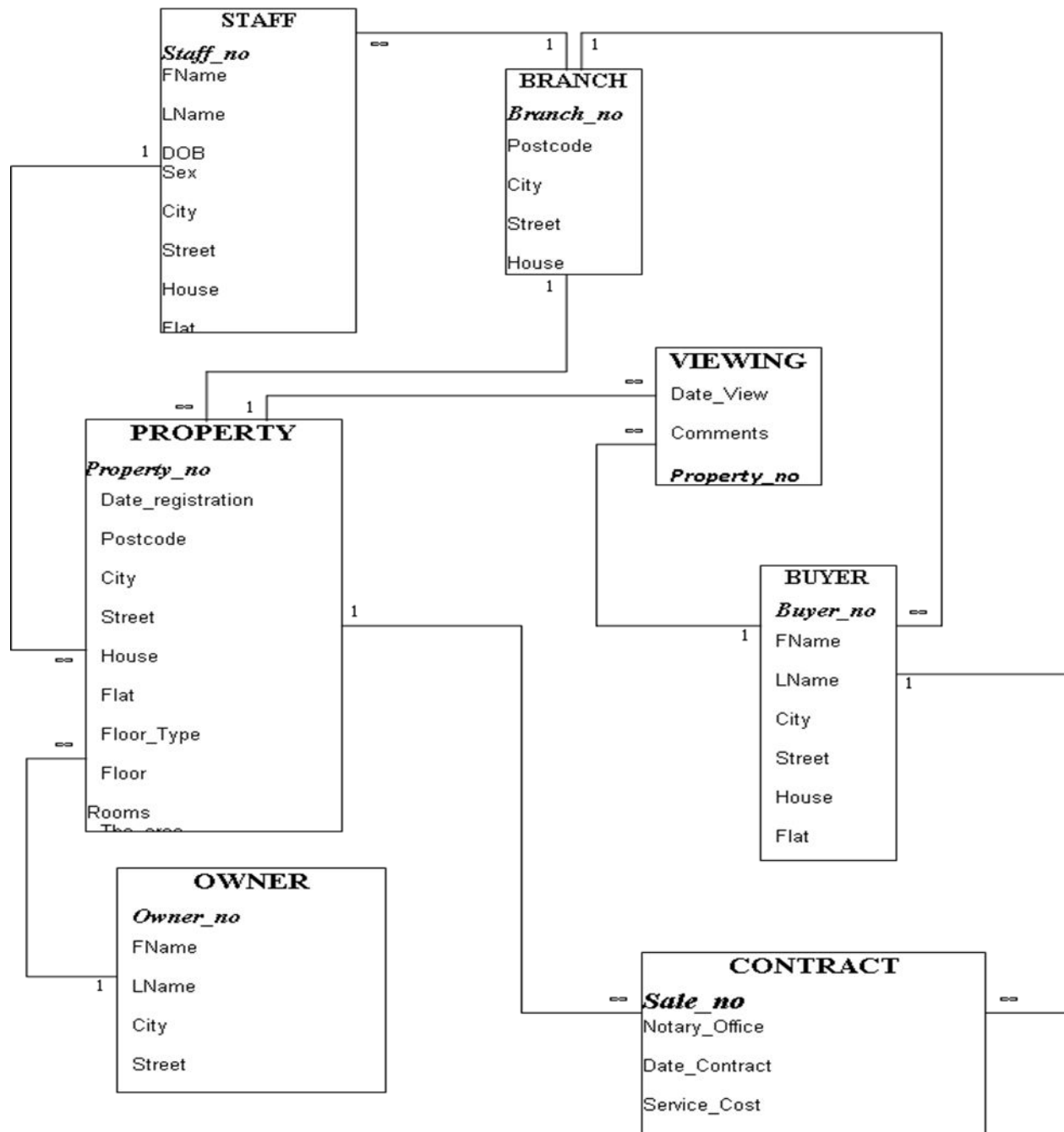
PROPERTY_no FROM VIEWING

WHERE

VIEWING.Property_no=PROPERTY

Property_no)

Вывести список объектов, цена которых выше средней.



Property_no	Selling_price	Rooms	Staff_no	Branch
3000	60000,0000	1	BMO550262	1
3001	35000,0000	1	BMO550262	5
3002	92000,0000	2	BMO550266	7
3003	15000,0000	1	BMO550262	5
3005	75000,0000	3	BMO550260	6
3006	2000,0000	3	BMO550264	3
3007	2000,0000	3	BMO550267	2

```
SELECT PROPERTY_no  
FROM PROPERTY  
WHERE  
selling_Price > (SELECT  
AVG(selling_price) FROM  
PROPERTY)
```

Вывести список объектов, цена которых выше средней в своем отделении.

```
SELECT PROPERTY_no  
FROM PROPERTY a  
WHERE selling_Price > (SELECT  
AVG(selling_price) FROM  
PROPERTY WHERE  
PROPERTY.Branch_no =  
a.Branch_no)
```

В каком городе продаётся
самая дешёвая квартира

SELECT City,


MIN(selling_price)

FROM property

GROUP BY City

Витебск	15000,00
Новопол	2000,00
Полоцк	2000,00

***SELECT City
FROM property WHERE
selling_price
=
(SELECT
MIN(selling_price) FROM
PROPERTY)***



**В каких городах средняя цена
квартиры не превышает 40000**

Property_no	Selling_price	Rooms	Staff_no	Branch
3000	60000,0000	1	BMO550262	1
3001	35000,0000	1	BMO550262	5
3002	92000,0000	2	BMO550266	7
3003	15000,0000	1	BMO550262	5
3005	75000,0000	3	BMO550260	6
3006	2000,0000	3	BMO550264	3
3007	2000,0000	3	BMO550267	2

***SELECT City,
FROM PROPERTY
GROUP BY CITY
HAVING AVG(selling_price)
<40000***

Без группировки

```
SELECT City  
FROM PROPERTY a  
WHERE  
(SELECT  
AVG(selling_price) FROM  
PROPERTY WHERE  
PROPERTY.City =  
a.CITY) < 40000
```

Найти номера отделений, в которых средний размер зарплаты равен максимальному размеру зарплаты сотрудников какого-либо другого отделения.

```
SELECT Branch_no, AVG(SALARY)  
FROM STAFF a  
GROUP BY Branch_no  
HAVING AVG(SALARY) =  
(SELECT MAX(SALARY)FROM  
STAFF  
WHERE a.branch_no<>  
STAFF.branch_no)
```

Найти номера отделов и минимальный и максимальный размер зарплаты сотрудников для тех отделов, в которых средний размер зарплаты не меньше среднего размера зарплаты сотрудников во всей компании

```
SELECT Branch_no, min(SALARY),  
max(SALARY)  
FROM STAFF  
GROUP BY Branch_no  
HAVING AVG(SALARY) >(SELECT  
AVG(SALARY) FROM STAFF)
```

Во всех отделах найти фамилии и число служащих, у которых в данном отделе имеются однофамильцы и фамилии которых совпадают с фамилиями руководителей их отделов.

Staff_no

FName

LName

DOB

Sex

City

Street

House

Flat

Stel_no

Date_Joined

Position


Salary

STAFF

```
SELECT Branch_no, FName, COUNT(*)  
FROM STAFF  
GROUP BY Branch_no, FName  
HAVING COUNT(*) > 1  
AND FName LIKE(SELECT FName FROM  
STAFF a  
WHERE Position='директор' AND  
STAff.Branch_no=a.Branch_no)
```

Найти самую дешевую квартиру

```
SELECT *  
FROM PROPERTY  
WHERE selling_price  
=(SELECT  
MIN(selling_price)FROM  
PROPERTY)
```



Найти номер телефона
отделения, продающего самую
дешёвую квартиру

```
SELECT Btel_no  
FROM PROPERTY INNER JOIN BRANCH  
ON PROPERTY.BRANCH_no=BRANCH.  
BRANCH_no  
WHERE selling_price =(SELECT  
MIN(selling_price)FROM PROPERTY)
```

```
SELECT Btel_no  
FROM Branch  
WHERE Branch_no IN(SELECT  
branch_no FROM PROPERTY  
WHERE Property_no IN(SELECT  
PROPERTY_no FROM PROPERTY  
WHERE selling_price=(SELECT  
MIN(selling_price)FROM  
PROPERTY)))
```

Создание таблицы из набора результатов

При помощи оператора ***SELECT INTO*** можно поместить набор результатов запроса в новую таблицу. Кроме того, этот оператор позволяет создавать временные таблицы. Запросы к временной таблице иногда оказываются проще тех, которые пришлось бы выполнять, обращаясь к нескольким таблицам или базам данных. Оператор ***SELECT INTO*** позволяет создать локальную или глобальную временную таблицу.

Например, создать таблицу, содержащую объекты собственности, находящиеся в городе Полоцке.

```
SELECT *  
INTO ##PROPERTY_POLOCK  
FROM PROPERTY  
WHERE City='Полоцк'
```


Найти максимальную из средних заработных плат отделений компании

1)

```
SELECT Branch_no,  
AVG(salary)avgсал  
INTO ##AVGTAB  
FROM staff  
GROUP BY branch_no
```

2)

```
SELECT MAX(avgсал)  
FROM ##AVGTAB
```