Лабораторная работа №3 Группировка данных посредством SQL-запросов. Подзапросы. Предложение HAVING, CASE в SQL-запросах

Предложение **GROUP BY** используется для определения групп выходных строк, к которым могут применяться агрегатные функции (**COUNT**, **MIN**, **MAX**, **AVG** и **SUM**). Если это предложение отсутствует, и используются агрегатные функции, то все столбцы с именами, упомянутыми в **SELECT**, должны быть включены в агрегатные функции, и эти функции будут применяться ко всему набору строк, которые удовлетворяют предикату запроса.

Создайте SQL-запросы:

1. Вывести информацию о количестве объектов недвижимости по каждому этажу

Вх. данные			Вых. данные	!
	этаж	_	количество	объектов
	недви	жим	ости	

В данном запросе необходимо использовать группировку данных. Для этого используется ключевое слово **GROUP BY**, которое указывает по какому полю группировать данные.

Подсчитать сколько однокомнатных и двухкомнатных объектов недвижимости находится в продаже

Пример

SELECT Количество комнат, COUNT(*) FROM Объекты недвижимости GROUP BY Количество комнат

2. Вывести информацию о количестве объектов недвижимости по каждому району

Вх. данные	Вых. данные	
	название района – количество	
	объектов недвижимости	

3. Вывести информацию о количестве двухкомнатных объектах недвижимости по каждому типу

Вх. данные	Вых. данные
количество комнат	тип объекта – количество
<u>Пример:</u>	объектов недвижимости
количество комнат = 2	

4. Вывести информацию о средней стоимости объектов недвижимости, расположенных на 2 этаже по каждому материалу здания

Вх. данные	Вых. данные
этаж	материал здания – средняя
<u>Пример:</u>	стоимость объекта недвижимости
этаж=2	

5. Вывести информацию о максимальной стоимости квартир, расположенных в каждом районе

Вх. данные	Вых. данные
тип объекта	название района – максимальная
Пример:	стоимость квартиры
тип объекта = квартира	

6. Вывести информацию о количестве квартир, проданных каждым риэлтором

Вх. данные	Вых. данные		
тип объекта	ФИО риэлтора – количество		
<u>Пример:</u>	квартир		
тип объекта = квартира			

7. Вывести информацию об общей стоимости апартаментов, расположенных в каждом районе

Вх. данные	Вых. данные
тип объекта	название район – общая стоимость
<u>Пример:</u>	апартаментов
тип объекта = апартаменты	

8. Вывести информацию о средней стоимости объектов недвижимости с площадью «ОТ» и «ДО» по каждому типу объекта

_	_
Вх. данные	Вых. данные

площадь	тип объекта- средняя стоимость объектов
<u>Пример:</u>	недвижимости
площадь >30 И <50	

9. Вывести информацию о средней оценке объектов недвижимости по каждому району

Вх. данные	Вых. данные
	название района – средняя оценка

10. Вывести информацию об общей продажной стоимости апартаментов, проданных в диапазоне дат «ОТ» и «ДО» по каждому риэлтору

Вх. данные	Вых. данные
тип объекта, дата продажи	ФИО риэлтора – общая продажная
<u>Пример:</u>	стоимость апартаментов
тип объекта = апартаменты	
дата продажи =	
>20.09.2017 <i>W</i> <20.09.2018	

11. Вывести информацию о средней оценке по каждому критерию для объекта недвижимости

Вх. данные	Вых. данные
адрес <u>Пример:</u> адрес = Победы 10 кв. 15	название критерия – средняя оценка

12. Вывести информацию о средней площади квартир по каждому району.

Вх. данные	Вых. данные
тип объекта	название района – средняя площадь
<u>Пример:</u>	
тип объекта = квартира	

13. Вывести информацию о максимальной и минимальной оценке по каждому критерию для объекта недвижимости

Вх. данные	Вых. данные		
адрес	название критерия – максимальная		
<u>Пример:</u>	оценка, минимальная оценка		
адрес = Победы 10 кв. 15			

14. Вывести информацию о количестве объектах недвижимости по количеству комнат, у которых разница между продажной и заявленной стоимостью больше 10000.

Вх. данные	Вых. данные
------------	-------------

количество	комнат	_	количество
объектов недвижимости			

Подзапросы. Предложение HAVING, CASE в SQL-запросах

- 1. Вывести адреса объектов недвижимости, у которых стоимость 1 м² меньше средней стоимости по району.
- 2. Вывести название районов, в которых количество проданных квартир больше 5.

Зачастую мы используем условие **HAVING** совместно с условием **GROUP BY**, чтобы отфильтровать строки, которые не удовлетворяют указанному условию.

Следующий запрос отображает синтаксис вызова условия HAVING:

```
SELECT
column_1,
aggregate_function (column_2)
FROM
tbl_name
GROUP BY
column_1
HAVING
condition;
```

Предложение **HAVING** устанавливает условие для групп, состоящих из строк, созданных предложением **GROUP BY** после его [предложения] применения, тогда как предложение **WHERE** устанавливает условие для отдельных строк перед применением условия **GROUP BY**. В этом и состоит главное отличие условия **HAVING** от условия **WHERE**.

В PostgreSQL вы можете использовать условие **HAVING** без условия **GROUP BY**. В таком случае, условие **HAVING** обратит запрос в одну группу. К тому же, список **SELECT** и условие **HAVING** могут ссылаться только на столбцы из агрегатных функций. Данный тип запроса возвращает одну строку, если условие в предложении **HAVING** принимает значение true, или нулевую строку, если оно равно false.

- 3. Вывести адреса квартир и название района, средняя оценка которых выше 3,5 баллов.
- 4. Вывести ФИО риэлторов, которые продали меньше 5 объектов недвижимости

- 5. Определить годы, в которых было размещено от 2 до 3 объектов недвижимости
- 6. Определить адреса квартир, стоимость 1м² которых меньше средней по району.
- 7. Определить ФИО риэлторов, которые ничего не продали в текущем году.
- 8. Вывести названия районов, в которых средняя площадь продаваемых квартир больше 30m^2 .
- 9. Вывести для указанного риэлтора (ФИО) года, в которых он продал больше 2 объектов недвижимости.
- 10. Вывести ФИО риэлторов, которые заработали премию в текущем месяце больше 40000 рублей. Премия рассчитываются по формуле: общая стоимость всех проданных квартир * 15%
- 11. Вывести количество однокомнатных и двухкомнатных квартир в указанном районе. Формат вывода:

Вид квартиры	Количество объектов недвижимости
Однокомнатных квартир	2
Двухкомнатных квартир	5

Выражение CASE в SQL представляет собой общее условное выражение, напоминающее операторы if/else в других языках программирования:

CASE WHEN условие THEN результат

[WHEN ...]
[ELSE результат]

END

Предложения CASE можно использовать везде, где допускаются выражения. Каждое условие в нём представляет собой выражение, возвращающее результат типа boolean. Если результатом выражения оказывается true, значением выражения CASE становится результатом, следующий за условием, а остальная часть выражения CASE не вычисляется. Если же условие не выполняется, за ним таким же образом проверяются все последующие предложения WHEN. Если не выполняется ни одно из условий WHEN, значением CASE становится результатом, записанный в предложении ELSE. Если при этом предложение ELSE отсутствует, результатом выражения будет NULL.

12. Определить индекс средней оценки по каждому критерию для указанного объекта недвижимости. Вывести среднюю оценку и эквивалентный текст согласно таблице:

Диапазон	Эквивалентный текст		
от 90% до 100%	превосходно		
от 80% до 90%	очень хорошо		
от 70% до 80%	хорошо		
от 60% до 70%	удовлетворительно		
до 60 %	неудовлетворительно		

Образец вывода результатов:

Критерий	Средняя оценка	Текст
Экология	5 из 5	превосходно
Чистота	4 из 5	очень хорошо
Соседи	3,5 из 5	хорошо

13. Добавить новую таблицу «Структура объекта недвижимости» с колонками: Объект недвижимости, Тип комнаты, Площадь. Установите ограничение-проверку **площади**, которая должна быть больше нуля и **типа комнаты** (1, 2, 3, 4), где 1 – кухня, 2 – зал, 3 – спальня, 4 – санузел.

Ограничение-проверка — наиболее общий тип ограничений. В его определении вы можете указать, что значение данной колонки должно удовлетворять логическому выражению (проверке истинности). Например, цену товара можно ограничить положительными значениями так:

```
CREATE TABLE products (

product_no integer,

name text,

price numeric CHECK (price > 0)
);
```

Как вы видите, ограничение определяется после типа данных, как и значение по умолчанию. Значения по умолчанию и ограничения могут указываться в любом порядке. Ограничение-проверка состоит из ключевого слова СНЕСК, за которым идёт выражение в скобках. Это выражение должно включать колонку, для которой задаётся ограничение, иначе оно не имеет большого смысла.

14. Вывести информацию о комнатах для объекта недвижимости.

Тип комнаты	Площадь

Зал	20
Кухня	5

- 15. Вывести количество объектов недвижимости по каждому району, общая площадь которых больше 40 м². Использовать таблицу «Структура объекта недвижимости».
- 16. Используя функции для работы с датой: extract(field from timestamp) и age(timestamp, timestamp), вывести квартиры, которые были проданы не позже 4 месяцев после размещения объявления о их продаже.
- 17. Вывести адреса объектов недвижимости, стоимость 1м² которых меньше средней всех объектов недвижимости по району, объявления о которых были размещены не более 4 месяцев назад. Формат вывода:

Адрес	Статус
ул. Победы 40	продано
Ул. Савушкина 20	в продаже

18. Вывести информацию о количество продаж в предыдущем и текущем годах по каждому району, а также процент изменения.

Название района	2018	2019	Разница в %
Кировский	20	40	100
Ленинский	30	20	-33,33