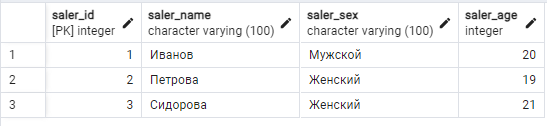
**Мелехин Александр Кс-30 Вариант 9 Лабораторная работа 4**

**Данные таблицы для лабораторной работы 4**

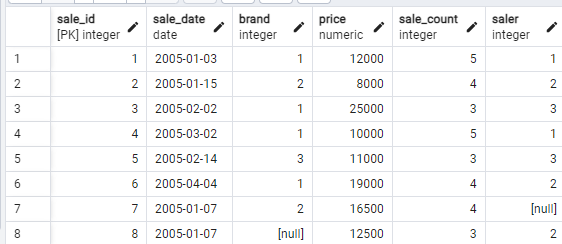
**Таблица salers**



**Таблица brands**



**Таблица sales**



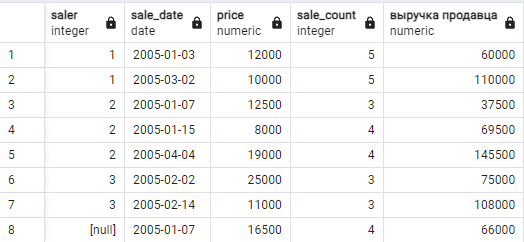
**Задание 1**

**Задание:** придумать запрос на применение агрегирующих оконных функций

**SQL код для задания:** SELECT saler, sale\_date, price, sale\_count, SUM(price \* sale\_count) OVER (PARTITION BY saler ORDER BY sale\_date) AS "выручка продавца" FROM sales;

**Пояснение:** Запрос вычисляет общую выручку для каждого продавца, используя оконную функцию с разбиением по каждому продавцу.

**Результат**



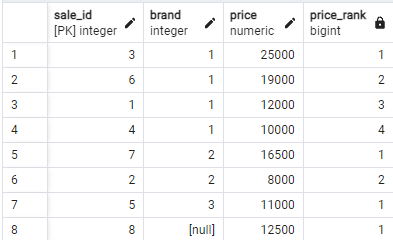
**Задание 2**

**Задание:** придумать запрос на применение ранжирующих оконных функций.

**SQL код для задания:** SELECT sale\_id, brand, price, RANK() OVER (PARTITION BY brand ORDER BY price DESC) AS price\_rank FROM sales;

**Пояснение:** запрос присваивает ранг каждому товару внутри группы марок на основе цены, от самой высокой к самой низкой.

**Результат**



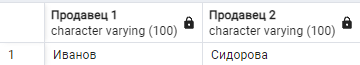
**Задание 3**

**Задание:** вывести пары атрибутов одной сущности при определенном условии.

**SQL код для задания:** SELECT DISTINCT LEAST(s1.saler\_name, s2.saler\_name) AS "Продавец 1", GREATEST(s1.saler\_name, s2.saler\_name) AS "Продавец 2" FROM salers s1 JOIN salers s2 ON s1.saler\_id < s2.saler\_id WHERE s1.saler\_age > 19 AND s2.saler\_age > 19;

**Пояснение:** запрос выводит пары продавцов с возрастом более 19 лет (*LEAST(s1.saler\_name, s2.saler\_name) AS "Продавец 1" и GREATEST(s1.saler\_name, s2.saler\_name) AS "Продавец 2"* нужны для того, чтобы избежать дубликатов пар, когда продавцы меняются местами или когда пара состоит из двух одинаковых продавцов).

**Результат**



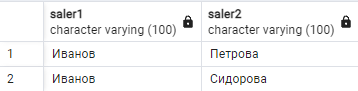
**Задание 4**

**Задание:** тоже, что и в пункте 3 с устранением избыточности без оператора DISTINCT.

**SQL код для задания:** SELECT s1.saler\_name AS saler1, s2.saler\_name AS saler2 FROM salers s1, salers s2 WHERE s1.saler\_id < s2.saler\_id AND s1.saler\_sex <> s2.saler\_sex;

**Пояснение:** в запросе выводятся пары продавцов разного пола.

**Результат**



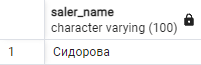
**Задание 5**

**Задание:** придумать однотабличный запрос, использующий подзапрос в условии отбора строк поле фразы WHERE.

**SQL код для задания:** SELECT saler\_name FROM salers WHERE saler\_age > (SELECT AVG(saler\_age) FROM salers);

**Пояснение:** запрос выбирает продавцов, чей возраст выше среднего.

**Результат**



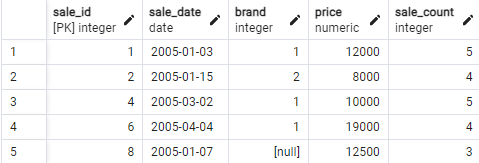
**Задание 6**

**Задание:** придумать многотабличный запрос, использующий подзапрос в условии отбора строк поле фразы WHERE.

**SQL код для задания:** SELECT sale\_id, sale\_date, brand, price, sale\_count FROM sales WHERE saler IN (SELECT saler\_id FROM salers WHERE saler\_age < 21);

**Пояснение:** Подзапрос выбирает ID продавцов моложе 21 года, и эти продажи отображаются.

**Результат**



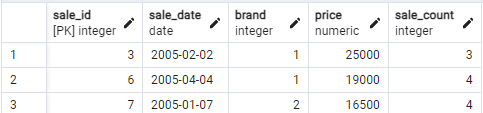
**Задание 7**

**Задание:** придумать запрос, использующий подзапрос с агрегатной функцией в условии отбора строк поле фразы WHERE.

**SQL код для задания:** SELECT sale\_id, sale\_date, brand, price, sale\_count FROM sales WHERE price > (SELECT AVG(price) FROM sales);

**Пояснение:** Запрос выбирает продажи, у которых цена выше средней по всем продажам.

**Результат**



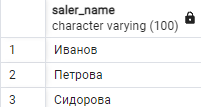
**Задание 8**

**Задание:** придумать запрос на использование подзапросов, которые выдают много строк с помощью оператора IN.

**SQL код для задания:** SELECT saler\_name FROM salers WHERE saler\_id IN (SELECT saler FROM sales WHERE brand IN (1, 2));

**Пояснение:** Запрос ищет продавцов, которые продали товары марок "Самсунг" или "Леново".

**Результат**



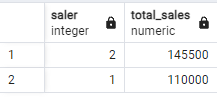
**Задание 9**

**Задание:** придумать запрос, использующий подзапрос в предложении HAVING.

**SQL код для задания:** SELECT saler, SUM(price \* sale\_count) AS total\_sales FROM sales GROUP BY saler HAVING SUM(price \* sale\_count) >= 110000;

**Пояснение:** запрос, выполняющий группировку по продавцу с ограничением по общей сумме продаж, превышающей 110000.

**Результат**



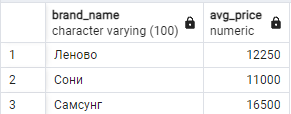
**Задание 10**

**Задание:** придумать запрос, использующий подзапрос в предложении FROM.

**SQL код для задания:** SELECT brand\_name, ROUND(AVG(price), 0) AS avg\_price FROM (SELECT brand, price FROM sales) AS sub JOIN brands ON brands.brand\_id = sub.brand GROUP BY brand\_name;

**Пояснение:** используется подзапрос в *FROM*, чтобы получить среднюю цену для каждой марки.

**Результат**



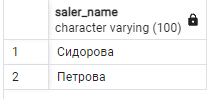
**Задание 11**

**Задание:** придумать запрос на использование соотнесенного подзапроса, который выдает много строк с помощью оператора IN

**SQL код для задания:** SELECT saler\_name FROM salers WHERE salers.saler\_id IN (SELECT saler FROM sales WHERE price > (SELECT AVG(price) FROM sales AS inner\_sales WHERE inner\_sales.brand = sales.brand));

**Пояснение:** подзапрос находит продавцов, которые продали товары дороже средней по конкретной марке.

**Результат**



**Задание 12**

**Задание:** придумать запрос на сравнение таблицы с собой.

**SQL код для задания:** SELECT s1.saler\_name FROM salers s1 LEFT JOIN salers s2 ON s1.saler\_age = s2.saler\_age AND s1.saler\_id <> s2.saler\_id WHERE s2.saler\_id IS NULL;

**Пояснение:** сравнение таблицы *salers* с самой собой для поиска продавцов, возраст которых уникален.

**Результат**

