**Отчет по Лаб. работе 3. Бабкин Иван КС-33**

1. Сформулировать на русском языке (в виде текста) осмысленные запросы. Для сформулированного запроса написать код на языке SQL. Выполнить запросы в СУБД PostgreSQL, приложить результаты в виде картинки.

1.1 Создать запрос на простейшее агрегирование с использованием какой-либо функции агрегирования. **Запрос:**

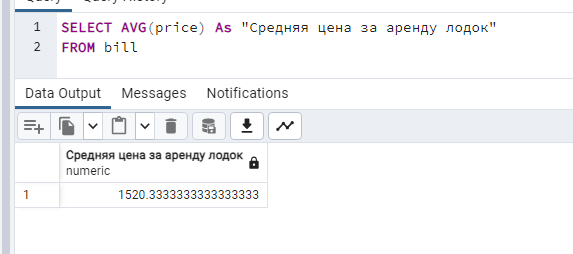
Вывести среднюю цену за аренду лодок

**Код:**

SELECT AVG(price) As "Средняя цена за аренду лодок"

FROM bill

**Скриншот выполнения работы программы:**

****

1.2Создать запрос на агрегирование с группировкой по значению функции (например, на функцию выделения года из данных типа date)

**Запрос:**

Посчитать кол-во заказов за каждый месяц в 2005 году

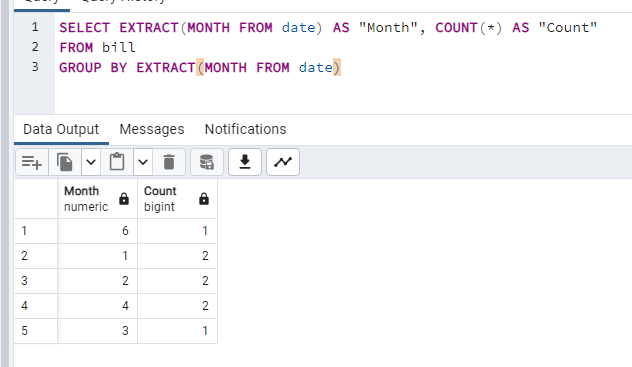
**Код:**

SELECT EXTRACT(MONTH FROM date) AS "Month", COUNT(\*) AS "Count"

FROM bill

GROUP BY EXTRACT(MONTH FROM date)

**Скриншот выполнения работы программы:**

****

1.3 Создать запрос на группировку данных по одному полю с использованием функций группировки.

**Запрос:**

Найти кол-во мужчин и женщин, делавших покупки

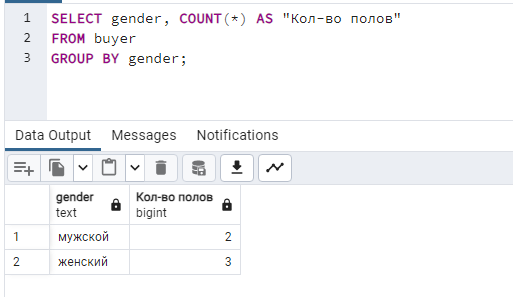
**Код:**

SELECT gender, COUNT(\*) AS "Кол-во полов"

FROM buyer

GROUP BY gender;

**Скриншот выполнения работы программы:**

****

1.4 Создать запрос на группировку данных по двум полям с использованием функций группировки. И объяснить в чем разница результатов по сравнению с п.3.

**Запрос:**

Вывести кол-во совпадений по гендеру и возрасту

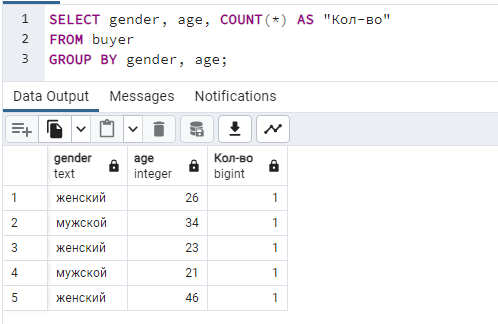
**Код:**

SELECT gender, age, COUNT(\*) AS "Кол-во"

FROM buyer

GROUP BY gender, age;

**Скриншот выполнения работы программы:**

****

1.5 Создать запрос на группировку данных по одному полю с использованием функций группировки и условием отбора (HAVING по датам).

**Запрос:**

Сгруппировать покупки, вывести все ID, дату, цену и время использования, которые были сделаны до 2005.04.07

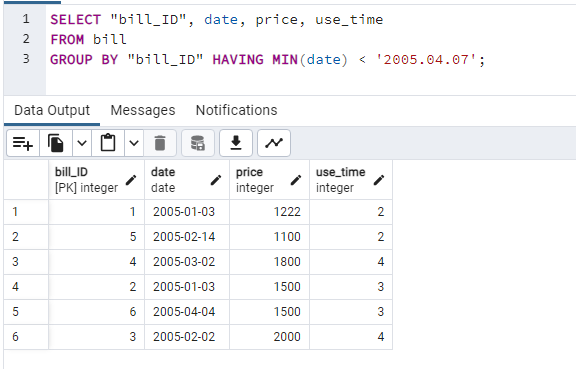
**Код:**

SELECT "bill\_ID", date, price, use\_time

FROM bill

GROUP BY "bill\_ID" HAVING MIN(date) < '2005.04.07';

**Скриншот выполнения работы программы:**

****

1.6 Создать запрос на группировку данных по одному полю с использованием функций группировки и условием отбора (HAVING) по числовому полю.

**Запрос:**

Сгруппировать все чеки по цене, так чтобы цена была больше 1500руб., а также вывести время использования и дату, совершения покупки

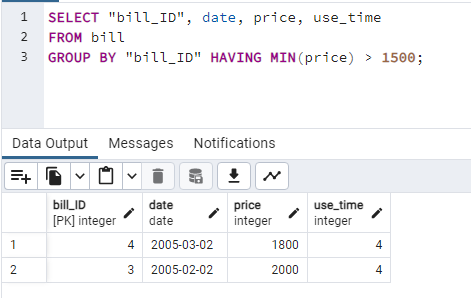
**Код:**

SELECT "bill\_ID", date, price, use\_time

FROM bill

GROUP BY "bill\_ID" HAVING MIN(price) > 1500;

**Скриншот выполнения работы программы:**

****

1.7 Создать запрос на группировку данных по одному полю с использованием функций группировки и условием отбора (HAVING с групповой функцией) по числовому полю.

**Запрос:**

Найти максимальную, минимальную и среднюю цену на аренду лодок при условии, что средняя цена не превышает 1600:

**Код:**

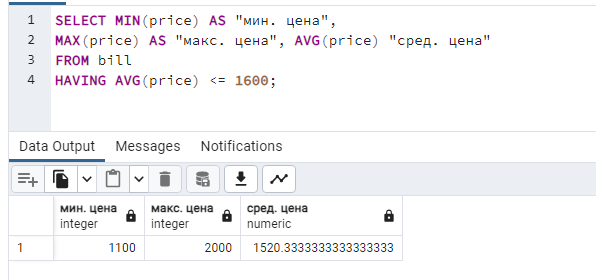
SELECT MIN(price) AS "мин. цена",

MAX(price) AS "макс. цена", AVG(price) "сред. цена"

FROM bill

HAVING AVG(price) <= 1600;

**Скриншот выполнения работы программы:**

****

1.8 Создать запрос на группировку данных по одному полю с использованием функций группировки с соединением 3-х таблиц. **Запрос:**

Сгруппировать данные по фамилии арендатора и по фамилии покупателя, а затем подсчитывает количество заказов для каждой уникальной комбинации.

**Код:**

SELECT

s.seller\_name,

buy.second\_name,

COUNT(b."bill\_ID") AS total\_bill

FROM

bill b

JOIN

seller s ON b."seller\_ID" = s."seller\_ID"

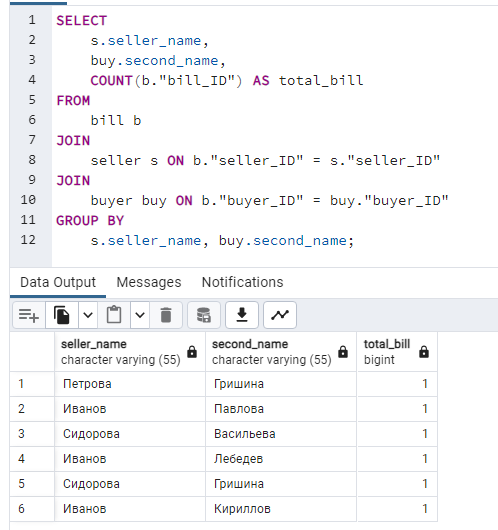
JOIN

buyer buy ON b."buyer\_ID" = buy."buyer\_ID"

GROUP BY

s.seller\_name, buy.second\_name;

**Скриншот выполнения работы программы:**

****

1.9 Создать вычисляемый запрос с соединением 3-х таблиц с сортировкой строк по возрастанию в вычисленном поле.

**Запрос:**

Выбрать информацию о заказах, включая идентификатор

заказа, фамилия продавца и фамилию покупателя. Результаты сортируются по фамилии покупателя.

**Код:**

SELECT

b."bill\_ID",

buy."second\_name",

s."seller\_name"

FROM

bill b

JOIN

buyer buy ON b."buyer\_ID" = buy."buyer\_ID"

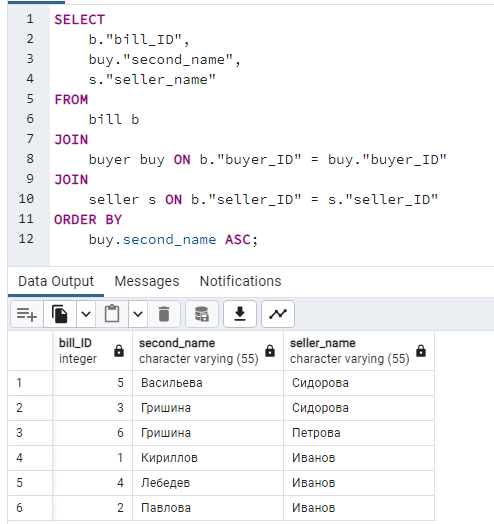
JOIN

seller s ON b."seller\_ID" = s."seller\_ID"

ORDER BY

buy.second\_name ASC;

**Скриншот выполнения работы программы:**

****

1.10 Создать вычисляемый запрос с соединением 3-х таблиц, с группировкой по одному полю. Сортировку результирующего набора строк выполнить по убыванию в вычисленном поле.

**Запрос:**

Сгруппировать покупки по фамилии покупателя, а также вывести кол-во совершенных покупок, отсортированных по убыванию.

**Код:**

SELECT

buy."second\_name",

COUNT(b."bill\_ID") AS total\_bills

FROM

bill b

JOIN

buyer buy ON b."buyer\_ID" = buy."buyer\_ID"

JOIN

seller s ON b."seller\_ID" = s."seller\_ID"

GROUP BY

buy."second\_name"

ORDER BY

total\_bills DESC;

**Скриншот выполнения работы программы:**

