



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

**О Т Ч Е Т**

по лабораторной работе № 1


Название: Введение в SQL

Дисциплина: Базы данных

Студент

ИУ6-33Б

(Группа)

  
(Подпись, дата)

Д. И. Мироненко

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

Д. А. Миков

(И.О. Фамилия)

Москва, 2023

## Задание

Первое практическое задание заключается в знакомстве со средой pgAdmin и написании SQL-запросов с использованием оператора SELECT.

Для модельной базы данных должны быть составлены 4 произвольных SELECT-запроса, демонстрирующие полученные знания. Запросы должны охватывать проработанные темы. После составления запросов следует убедиться в их правильности при помощи более простых запросов.

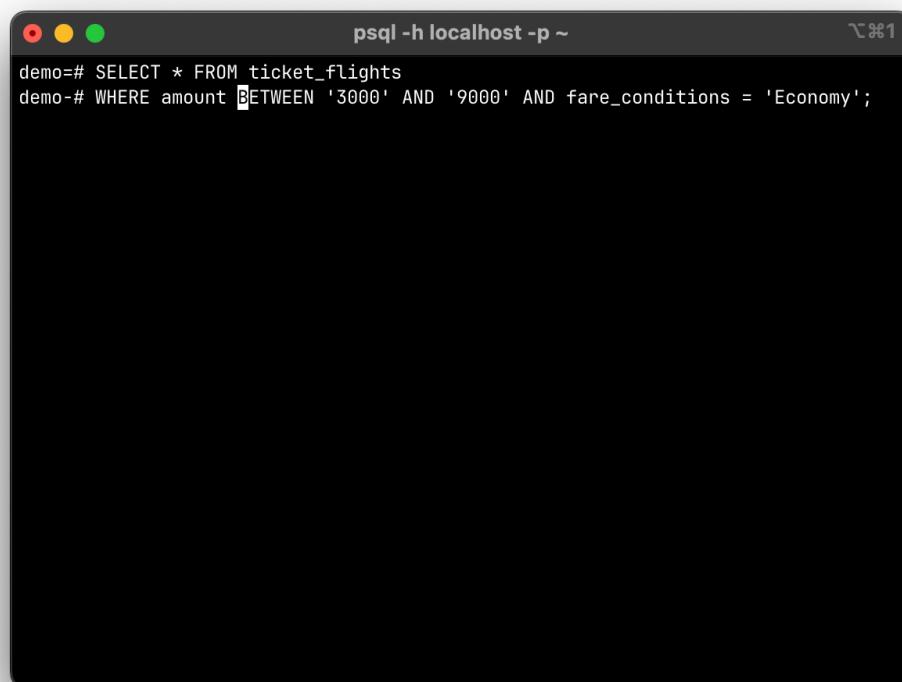
Дополнительные вопросы могут заключаться в построении более сложных запросов или объяснении работы подготовленных заданий.

## Загрузка базы данных



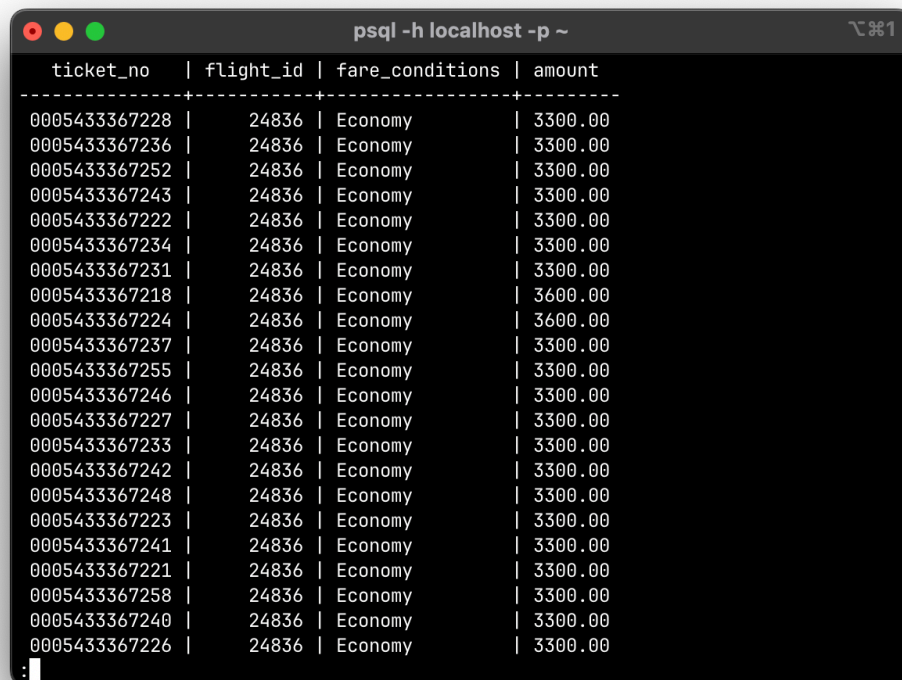
**Рисунок 1** – загрузка сторонней базы данных

## Запрос №1



```
psql -h localhost -p ~
demo=# SELECT * FROM ticket_flights
demo=# WHERE amount BETWEEN '3000' AND '9000' AND fare_conditions = 'Economy';
```

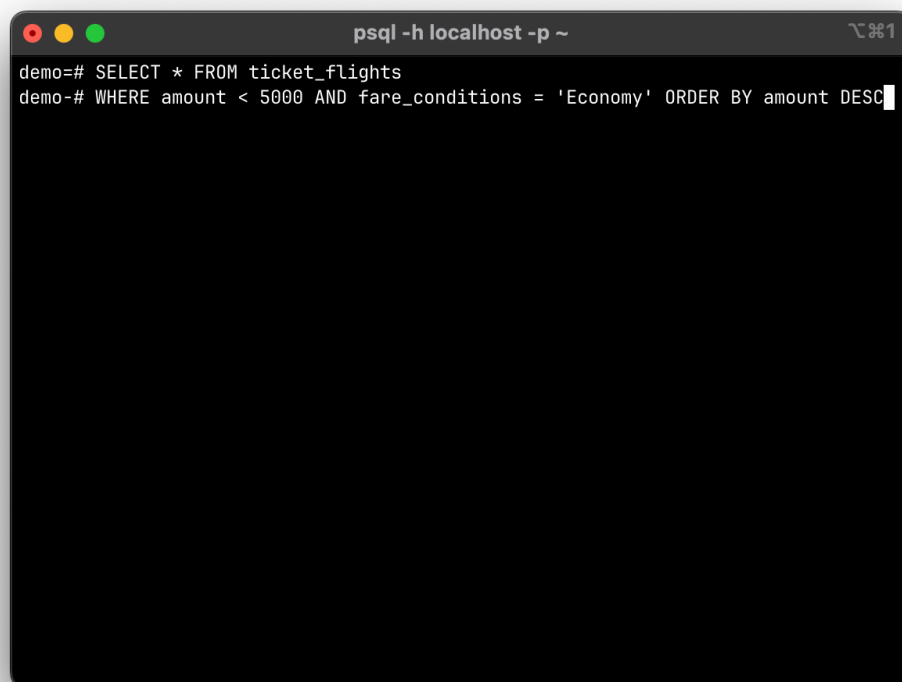
Рисунок 2 - запрос №1



ticket_no	flight_id	fare_conditions	amount
0005433367228	24836	Economy	3300.00
0005433367236	24836	Economy	3300.00
0005433367252	24836	Economy	3300.00
0005433367243	24836	Economy	3300.00
0005433367222	24836	Economy	3300.00
0005433367234	24836	Economy	3300.00
0005433367231	24836	Economy	3300.00
0005433367218	24836	Economy	3600.00
0005433367224	24836	Economy	3600.00
0005433367237	24836	Economy	3300.00
0005433367255	24836	Economy	3300.00
0005433367246	24836	Economy	3300.00
0005433367227	24836	Economy	3300.00
0005433367233	24836	Economy	3300.00
0005433367242	24836	Economy	3300.00
0005433367248	24836	Economy	3300.00
0005433367223	24836	Economy	3300.00
0005433367241	24836	Economy	3300.00
0005433367221	24836	Economy	3300.00
0005433367258	24836	Economy	3300.00
0005433367240	24836	Economy	3300.00
0005433367226	24836	Economy	3300.00

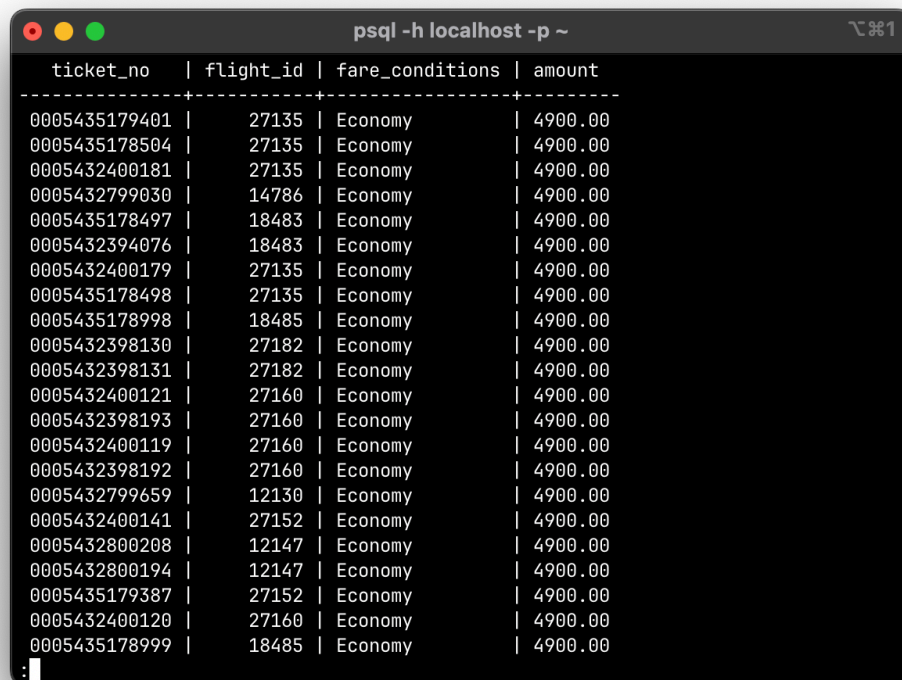
Рисунок 3 – результат запроса №1

## Запрос №2



```
psql -h localhost -p ~
demo=# SELECT * FROM ticket_flights
demo=# WHERE amount < 5000 AND fare_conditions = 'Economy' ORDER BY amount DESC
```

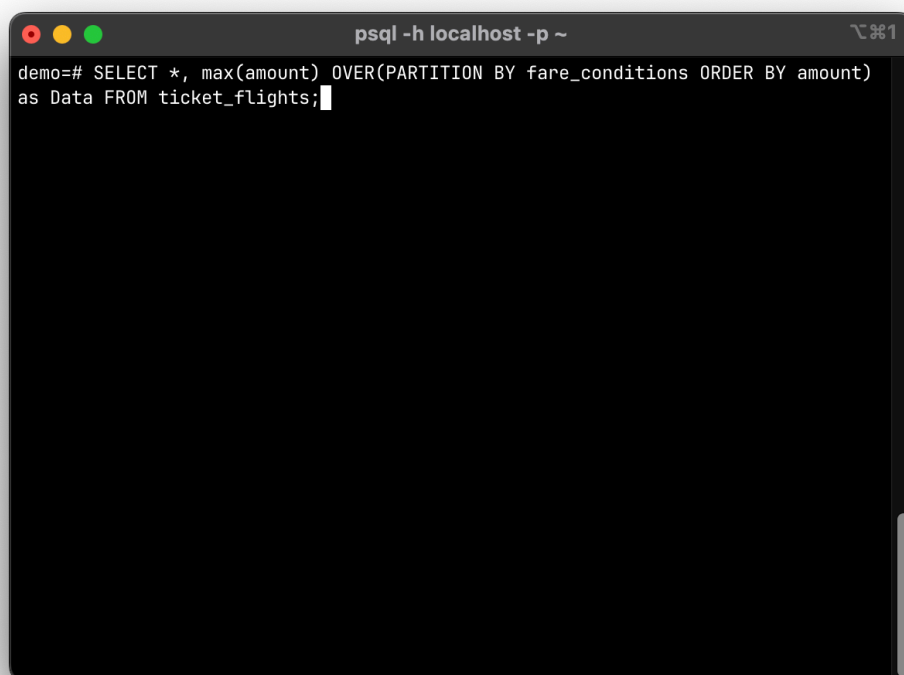
Рисунок 4 - запрос №2



ticket_no	flight_id	fare_conditions	amount
0005435179401	27135	Economy	4900.00
0005435178504	27135	Economy	4900.00
0005432400181	27135	Economy	4900.00
0005432799030	14786	Economy	4900.00
0005435178497	18483	Economy	4900.00
0005432394076	18483	Economy	4900.00
0005432400179	27135	Economy	4900.00
0005435178498	27135	Economy	4900.00
0005435178998	18485	Economy	4900.00
0005432398130	27182	Economy	4900.00
0005432398131	27182	Economy	4900.00
0005432400121	27160	Economy	4900.00
0005432398193	27160	Economy	4900.00
0005432400119	27160	Economy	4900.00
0005432398192	27160	Economy	4900.00
0005432799659	12130	Economy	4900.00
0005432400141	27152	Economy	4900.00
0005432800208	12147	Economy	4900.00
0005432800194	12147	Economy	4900.00
0005435179387	27152	Economy	4900.00
0005432400120	27160	Economy	4900.00
0005435178999	18485	Economy	4900.00

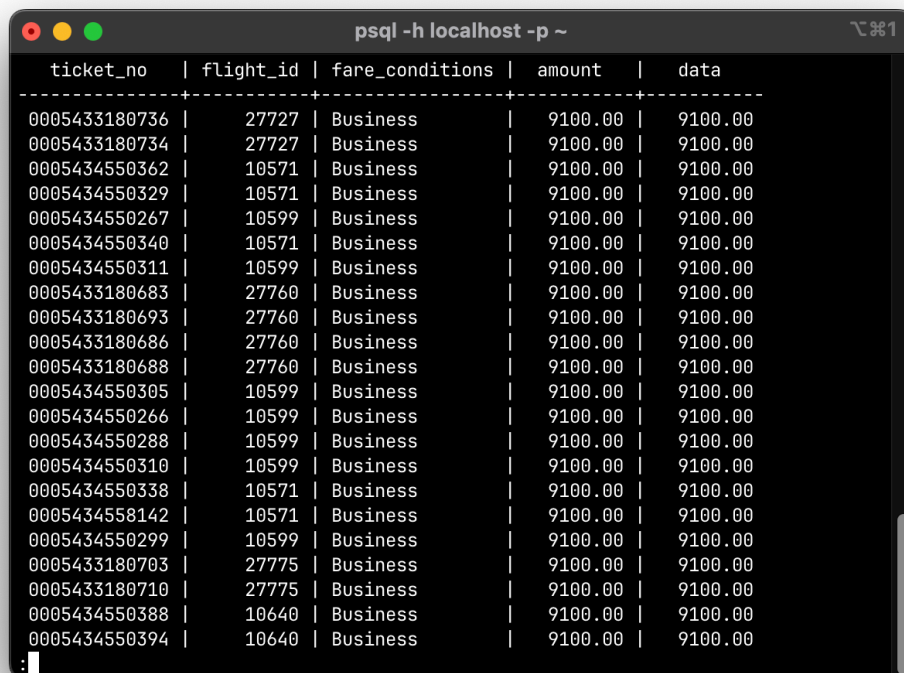
Рисунок 5 – результат запроса №2

## Запрос №3



```
psql -h localhost -p ~
demo=# SELECT *, max(amount) OVER(PARTITION BY fare_conditions ORDER BY amount)
as Data FROM ticket_flights;
```

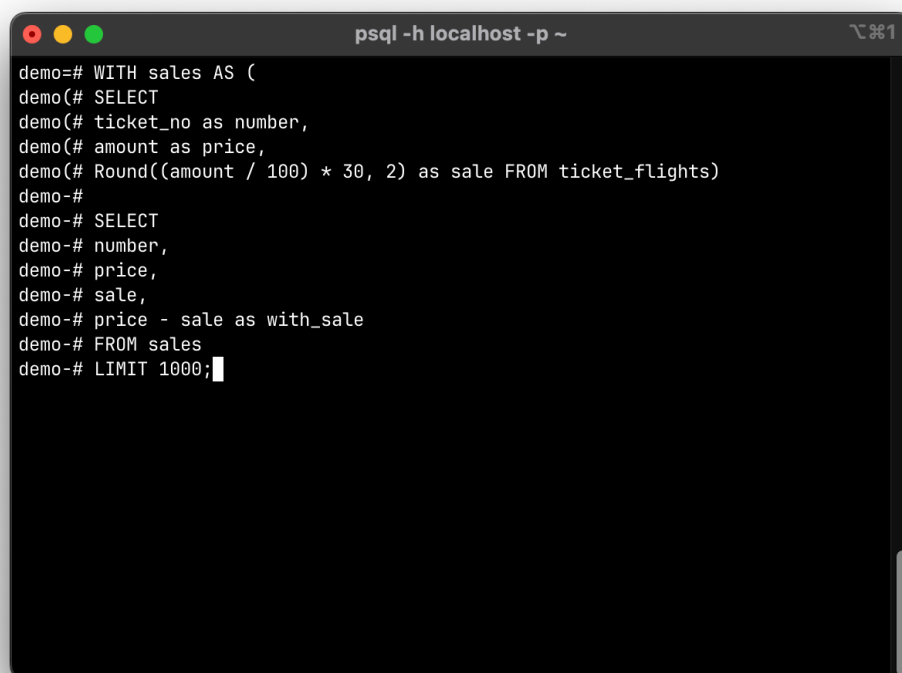
Рисунок 6 - запрос №3



ticket_no	flight_id	fare_conditions	amount	data
0005433180736	27727	Business	9100.00	9100.00
0005433180734	27727	Business	9100.00	9100.00
0005434550362	10571	Business	9100.00	9100.00
0005434550329	10571	Business	9100.00	9100.00
0005434550267	10599	Business	9100.00	9100.00
0005434550340	10571	Business	9100.00	9100.00
0005434550311	10599	Business	9100.00	9100.00
0005433180683	27760	Business	9100.00	9100.00
0005433180693	27760	Business	9100.00	9100.00
0005433180686	27760	Business	9100.00	9100.00
0005433180688	27760	Business	9100.00	9100.00
0005434550305	10599	Business	9100.00	9100.00
0005434550266	10599	Business	9100.00	9100.00
0005434550288	10599	Business	9100.00	9100.00
0005434550310	10599	Business	9100.00	9100.00
0005434550338	10571	Business	9100.00	9100.00
0005434558142	10571	Business	9100.00	9100.00
0005434550299	10599	Business	9100.00	9100.00
0005433180703	27775	Business	9100.00	9100.00
0005433180710	27775	Business	9100.00	9100.00
0005434550388	10640	Business	9100.00	9100.00
0005434550394	10640	Business	9100.00	9100.00

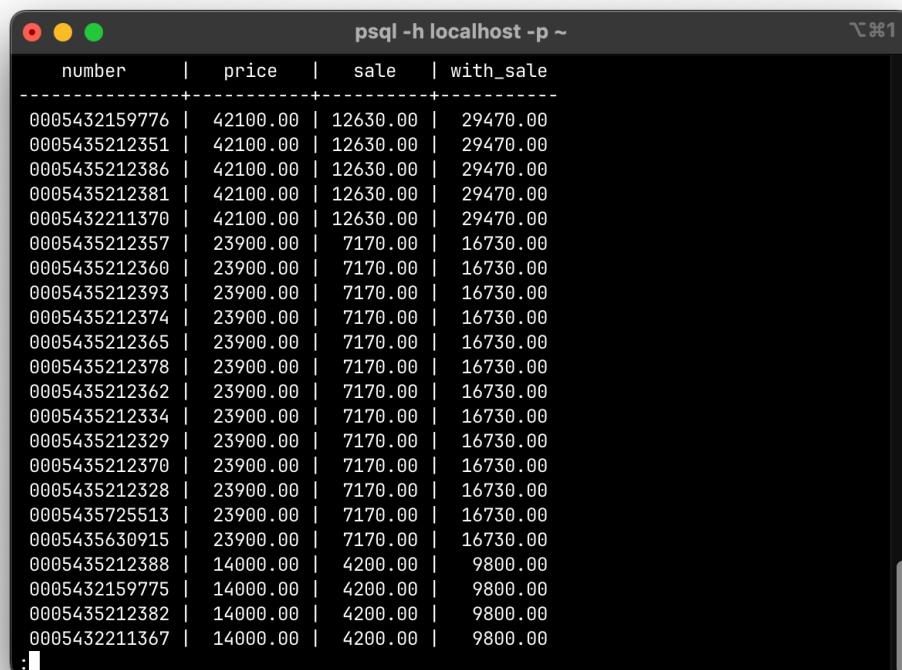
Рисунок 7 – результат запроса №3

## Запрос №4



```
psql -h localhost -p ~
demo=# WITH sales AS (
demo=# SELECT
demo=# ticket_no as number,
demo=# amount as price,
demo=# Round((amount / 100) * 30, 2) as sale FROM ticket_flights)
demo=#
demo=# SELECT
demo=# number,
demo=# price,
demo=# sale,
demo=# price - sale as with_sale
demo=# FROM sales
demo=# LIMIT 1000;
```

Рисунок 8 - запрос №4



number	price	sale	with_sale
0005432159776	42100.00	12630.00	29470.00
0005435212351	42100.00	12630.00	29470.00
0005435212386	42100.00	12630.00	29470.00
0005435212381	42100.00	12630.00	29470.00
0005432211370	42100.00	12630.00	29470.00
0005435212357	23900.00	7170.00	16730.00
0005435212360	23900.00	7170.00	16730.00
0005435212393	23900.00	7170.00	16730.00
0005435212374	23900.00	7170.00	16730.00
0005435212365	23900.00	7170.00	16730.00
0005435212378	23900.00	7170.00	16730.00
0005435212362	23900.00	7170.00	16730.00
0005435212334	23900.00	7170.00	16730.00
0005435212329	23900.00	7170.00	16730.00
0005435212370	23900.00	7170.00	16730.00
0005435212328	23900.00	7170.00	16730.00
0005435725513	23900.00	7170.00	16730.00
0005435630915	23900.00	7170.00	16730.00
0005435212388	14000.00	4200.00	9800.00
0005432159775	14000.00	4200.00	9800.00
0005435212382	14000.00	4200.00	9800.00
0005432211367	14000.00	4200.00	9800.00

Рисунок 9 – результат запроса №4

## Вывод

Получен опыт взаимодействия с консолью psql; изучены основы языка SQL.

## Контрольные вопросы

- 1. Рассказать про операцию соединения (JOIN) и различные её разновидности. JOIN объединяет несколько таблиц. Существуют следующие разновидности:*
  - LEFT JOIN (по умолчанию) – возвращает все строки из левой таблицы с добавлением правых;
  - RIGHT JOIN – возвращает все строки из правой таблицы с добавлением левых;
  - INNER JOIN – возвращает только строки, которые есть в обеих таблицах;
  - FULL JOIN – возвращает все строки из левой и правой таблиц.

Отсутствующие значения заменяются на NULL.
- 2. Рассказать про агрегатные функции, предложения GROUP BY и HAVING.*

Агрегатные функции выполняют обработку над несколькими значениями из одного столбца. GROUP BY группирует строки по одинаковым значениям одного или нескольких столбцов. HAVING фильтрует результат агрегатных функций по указанному условию.
- 3. Как выбрать только уникальные значения какого-либо столбца?*

SELECT DISTINCT <column> FROM <table>;
- 4. Как осуществить сортировку по возрастанию/убыванию по значению какого-либо столбца?*

По возрастанию: SELECT <columns> FROM <table> ORDER BY <column> [ASC].

По убыванию: SELECT <columns> FROM <table> ORDER BY <column> DESC.
- 5. Как агрегатные функции ведут себя по отношению к неопределённым значениям?*

Агрегатные функции игнорируют NULL, кроме COUNT.

6. *Чем отличаются UNION и UNION ALL?*

UNION – объединяет результаты двух запросов.

UNION ALL – объединяет результаты двух запросов, сохраняя все строки.

7. *Чем отличаются COUNT(\*) и COUNT(field)?*

COUNT(\*) учитывает NULL.

COUNT(field) не учитывает NULL.

8. *Как подсчитать количество уникальных значений столбца?*

Нужно использовать COUNT(DISTINCT field).

9. *Как можно осуществить проверку на неопределенное значение?*

Нужно использовать условие IS NULL.

10. *Рассказать про предикат LIKE.*

Предикат LIKE выполняет поиск по шаблону (LIKE '%gmail.com').

11. *Как можно выбрать только определенное количество строк?*

SELECT <columns> FROM <table> LIMIT <count>;

12. *Округлить результирующее значение до 3 знаков после точки.*

Нужно использовать ROUND(<column>, 3).

13. *Округлить вещественное число до целого без нулей после точки.*

Нужно использовать ROUND(<column>).

14. *Переписать запрос, не используя функцию MAX (MIN).*

Вместо MAX можно добавить: ORDER BY <column> DESC LIMIT 1;

Вместо MIN можно добавить: ORDER BY <column> [ASC] LIMIT 1;

15. *Изменить формат вывода данных (например, формат даты и времени).*

Нужно использовать функцию to\_char: to\_char(timestamp '2002-04-20 17:31:12.66', 'HH12:MI:SS') → 05:31:12.