

**Министерство науки и высшего образования Российской Федерации**  
федеральное государственное автономное образовательное учреждение высшего  
образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Отчет**

по лабораторной работе №4 «ЗАПРОСЫ НА ВЫБОРКУ И МОДИФИКАЦИЮ ДАННЫХ.  
ПРЕДСТАВЛЕНИЯ. РАБОТА С ИНДЕКСАМИ»

по дисциплине «Проектирование и реализация баз данных»

Автор: Шалунов Андрей Ильич

Факультет: ИКТ

Группа: К3240 Преподаватель:

Горова М.М.



Санкт-Петербург 2023

## Оглавление

Цель работы.....	3
Практическое задание .....	3
Вариант 11. БД «Автомастерская».....	3
Выполнение .....	3
Вывод.....	19

## Цель работы

Овладеть практическими навыками создания таблиц базы данных PostgreSQL 1X, заполнения их рабочими данными, резервного копирования и восстановления БД.

## Практическое задание

Создать базу данных с использованием pgAdmin 4 (согласно индивидуальному заданию).

1. Создать схему в составе базы данных.
2. Создать таблицы базы данных.
3. Установить ограничения на данные: *Primary Key, Unique, Check, Foreign Key*.
4. Заполнить таблицы БД рабочими данными.
5. Создать резервную копию БД. *Указание:*  
*Создать две резервные копии:*
  - с расширением *CUSTOM* для восстановления БД;
  - с расширением *PLAIN* для листинга (в отчете);
  - при создании резервных копий БД настроить параметры *Dump options* для *Type of objects* и *Queries*.
7. Восстановить БД.

## Вариант 11. БД «Автомастерская»

### Описание предметной области:

Описание предметной области: Сеть автомастерских осуществляет ремонт автомобилей, используя для этих целей штат мастеров и свои мастерские. Стоимость ремонта включает цену деталей и стоимость работы.

Заработная плата мастеров составляет 50% стоимости работы.

С клиентом заключается договор на выполнение авторемонтных и профилактических работ, который сопровождается администратором. В каждом договоре может быть несколько видов услуг. Для выполнения видов работ могут требоваться детали или расходные материалы, которые предоставляет либо клиент, либо автомастерская. Если детали предоставляет автомастерская, то их стоимость включается в смету по договору.

Каждый вид работ могут выполнять разные мастера, в зависимости от их специализации. Распределение мастеров выполняет администратор.

БД должна содержать следующий минимальный набор сведений: Табельный номер сотрудника. ФИО сотрудника. Должность. Разряд мастера. Специализация. Адрес автомастерской. Дата заказа. Гос. Номер автомобиля. Марка. Мощность автомобиля. Год выпуска. Цвет автомобиля. Дата принятия в ремонт. Плановая дата окончания ремонта. Фактическая дата окончания ремонта. Вид ремонта. Стоимость вида ремонта. Название детали. Цена детали. Марка и модель автомобиля. Страна производителя. Госномер автомобиля. ФИО владельца. Номер телефона владельца. E-mail владельца.

## Выполнение

Название создаваемой БД – «Автомастерская» («Autorepair Shop»)

Схема логической модели базы данных, которая сгенерирована в Generate ERD, находится на рисунке 1.

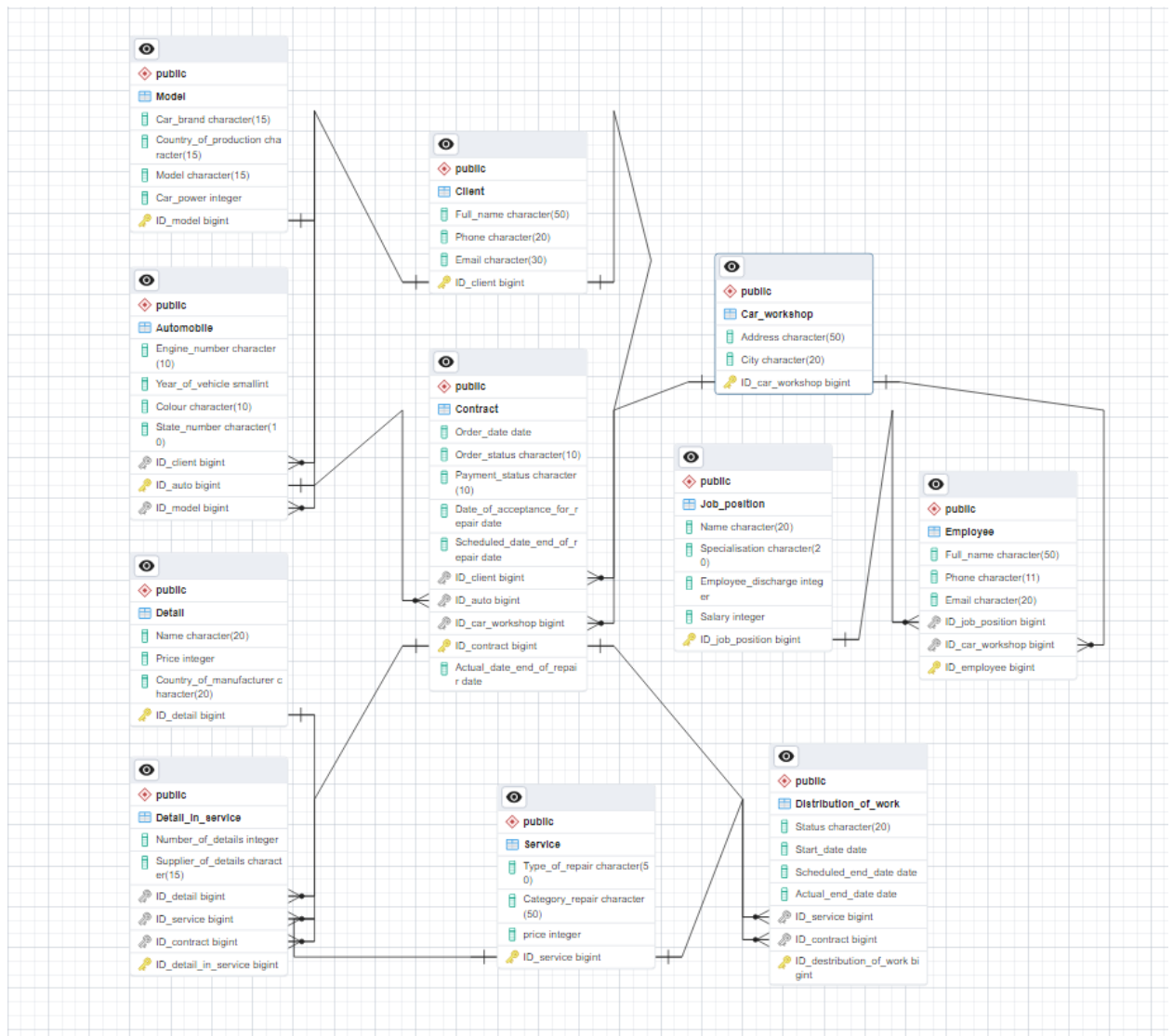


Рисунок 1 — ERD-схема базы данных

## Запросы

**Выбрать фамилию того механика, который чаще всех работает с автомобилями марки "Тойота".**

```
SELECT "Mechanic_Name", "Number_of_Repairs"
```

```
FROM (
```

```
SELECT e."Full_name" AS "Mechanic_Name", COUNT(*) AS "Number_of_Repairs"
```

```
FROM public."Employee" e

JOIN public."Contract" c ON e."ID_employee" = c."ID_employee"

JOIN public."Automobile" a ON c."ID_auto" = a."ID_auto"

JOIN public."Model" m ON a."ID_model" = m."ID_model"

JOIN public."Job_position" jp ON e."ID_job_position" = jp."ID_job_position"

WHERE m."Car_brand" = 'Toyota' AND jp."Name" = 'Мастер-механик'

GROUP BY e."Full_name"

) AS subquery

ORDER BY "Number_of_Repairs" DESC

LIMIT 1;
```

```

1 SELECT "Mechanic_Name", "Number_of_Repairs"
2 FROM (
3     SELECT e."Full_name" AS "Mechanic_Name", COUNT(*) AS "Number_of_Repairs"
4     FROM public."Employee" e
5     JOIN public."Contract" c ON e."ID_employee" = c."ID_employee"
6     JOIN public."Automobile" a ON c."ID_auto" = a."ID_auto"
7     JOIN public."Model" m ON a."ID_model" = m."ID_model"
8     JOIN public."Job_position" jp ON e."ID_job_position" = jp."ID_job_position"
9     WHERE m."Car_brand" = 'Toyota' AND jp."Name" = 'Мастер-механик'
10    GROUP BY e."Full_name"
11 ) AS subquery
12 ORDER BY "Number_of_Repairs" DESC
13 LIMIT 1;
14 |

```

Data Output Messages Notifications



	Mechanic_Name character	Number_of_Repairs bigint
1	Евгения Никитина	4

**Определить тех владельцев автомобилей, которых всегда обслуживает один и тот же механик. Вывести фамилии механика и его постоянного клиента.**

SELECT

e."Full\_name" AS "Механик",

c."ID\_client" AS "Постоянный клиент"

FROM public."Employee" e

JOIN public."Contract" c ON e."ID\_employee" = c."ID\_employee"

WHERE e."ID\_job\_position" = (SELECT "ID\_job\_position" FROM public."Job\_position"  
WHERE "Name" = 'Мастер-механик')

AND c."ID\_client" IN (

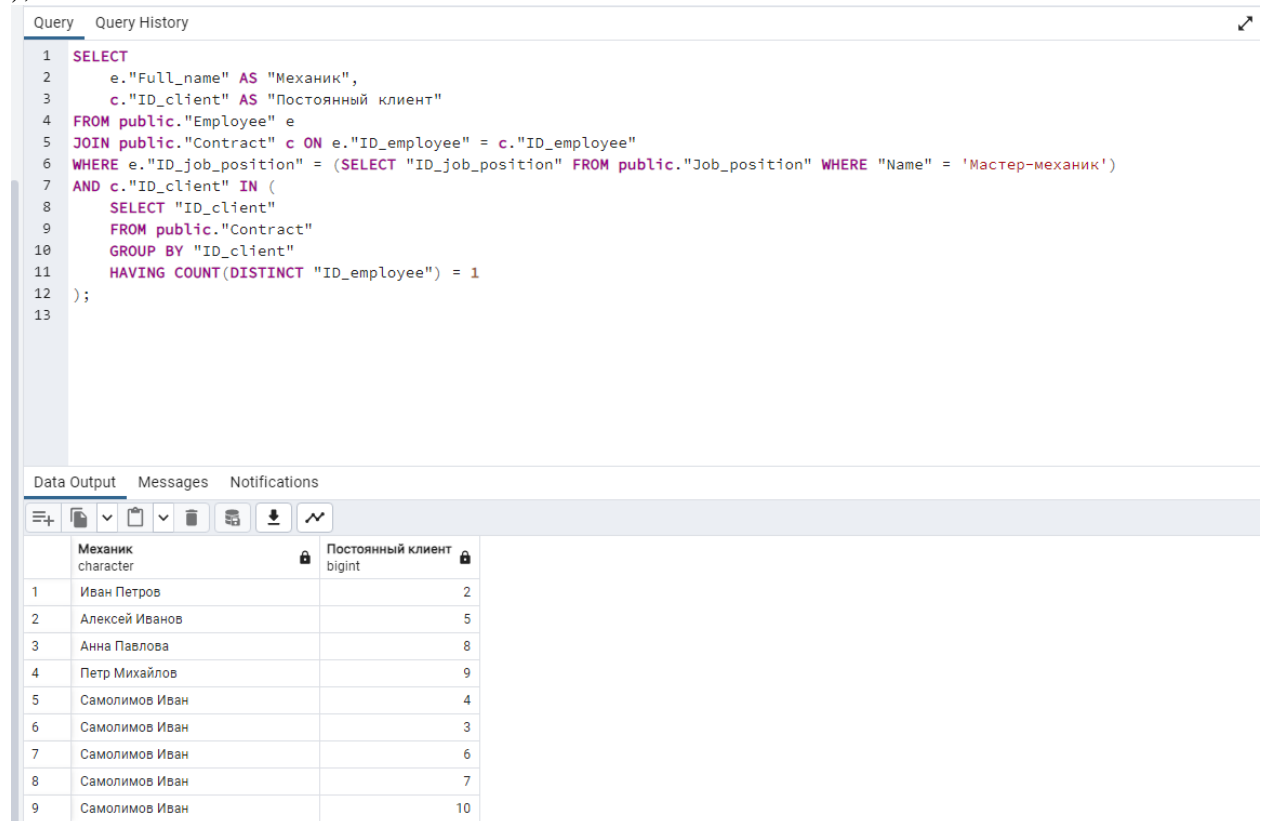
SELECT "ID\_client"

FROM public."Contract"

GROUP BY "ID\_client"

HAVING COUNT(DISTINCT "ID\_employee") = 1

);



Query Query History

```
1 SELECT
2     e."Full_name" AS "Механик",
3     c."ID_client" AS "Постоянный клиент"
4 FROM public."Employee" e
5 JOIN public."Contract" c ON e."ID_employee" = c."ID_employee"
6 WHERE e."ID_job_position" = (SELECT "ID_job_position" FROM public."Job_position" WHERE "Name" = 'Мастер-механик')
7 AND c."ID_client" IN (
8     SELECT "ID_client"
9     FROM public."Contract"
10    GROUP BY "ID_client"
11   HAVING COUNT(DISTINCT "ID_employee") = 1
12 );
13
```

Data Output Messages Notifications

	Механик character	Постоянный клиент bigint
1	Иван Петров	2
2	Алексей Иванов	5
3	Анна Павлова	8
4	Петр Михайлов	9
5	Самолимов Иван	4
6	Самолимов Иван	3
7	Самолимов Иван	6
8	Самолимов Иван	7
9	Самолимов Иван	10

**Вывести фамилии механиков, которые не выполняли работы в срок и количество дней просрочки выполнения заказа.**

SELECT

e."Full\_name" AS "Mechanic\_Name",

SUM(CASE WHEN c."Actual\_date\_end\_of\_repair" > c."Scheduled\_date\_end\_of\_repair"  
THEN (c."Actual\_date\_end\_of\_repair" - c."Scheduled\_date\_end\_of\_repair") ELSE 0 END) AS  
"Total\_Delayed\_Days"

FROM

public."Employee" e

JOIN

public."Contract" c ON e."ID\_employee" = c."ID\_employee"

WHERE

e."ID\_job\_position" IN (SELECT "ID\_job\_position" FROM public."Job\_position" WHERE "Name" = 'Мастер-механик')

GROUP BY

e."Full\_name"

HAVING

SUM(CASE WHEN c."Actual\_date\_end\_of\_repair" > c."Scheduled\_date\_end\_of\_repair" THEN (c."Actual\_date\_end\_of\_repair" - c."Scheduled\_date\_end\_of\_repair") ELSE 0 END) > 0;

The screenshot shows a SQL query editor with a query window and a results window. The query window contains the following SQL code:

```
1 SELECT
2   e."Full_name" AS "Mechanic_Name",
3   SUM(CASE WHEN c."Actual_date_end_of_repair" > c."Scheduled_date_end_of_repair" THEN (c."Actual_date_end_of_repair" - c."Sch
4 FROM
5   public."Employee" e
6 JOIN
7   public."Contract" c ON e."ID_employee" = c."ID_employee"
8 WHERE
9   e."ID_job_position" IN (SELECT "ID_job_position" FROM public."Job_position" WHERE "Name" = 'Мастер-механик')
10 GROUP BY
11   e."Full_name"
12 HAVING
13   SUM(CASE WHEN c."Actual_date_end_of_repair" > c."Scheduled_date_end_of_repair" THEN (c."Actual_date_end_of_repair" - c."Sch
14
```

The results window shows the following data:

	Mechanic_Name character	Total_Delayed_Days bigint
1	Иван Петров	1
2	Петр Михайлов	1
3	Самолимов Иван	2

**Вывести данные механика, который выполнял все виды ремонта за прошедшую неделю.**

SELECT e."Full\_name"

FROM public."Employee" e

INNER JOIN public."Job\_position" j ON e."ID\_job\_position" = j."ID\_job\_position"



INNER JOIN public."Contract" c ON e."ID\_employee" = c."ID\_employee"

INNER JOIN public."Distribution\_of\_work" d ON c."ID\_contract" = d."ID\_contract"

INNER JOIN public."Service" s ON d."ID\_service" = s."ID\_service"

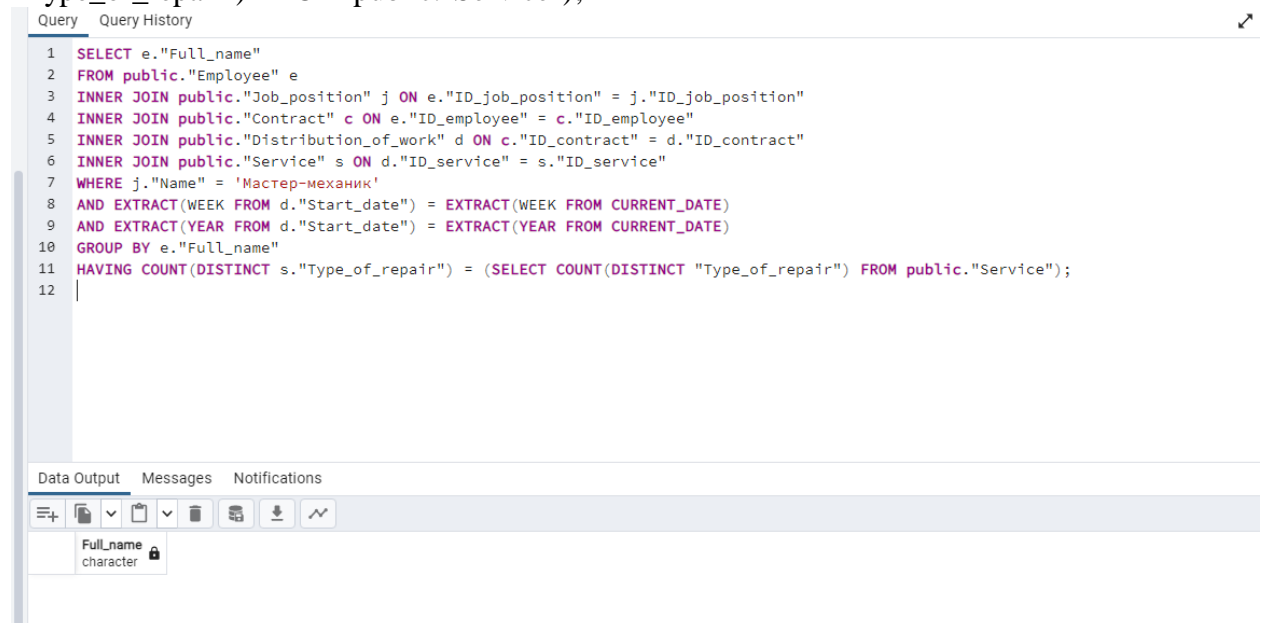
WHERE j."Name" = 'Мастер-механик'

AND EXTRACT(WEEK FROM d."Start\_date") = EXTRACT(WEEK FROM CURRENT\_DATE)

AND EXTRACT(YEAR FROM d."Start\_date") = EXTRACT(YEAR FROM CURRENT\_DATE)

GROUP BY e."Full\_name"

HAVING COUNT(DISTINCT s."Type\_of\_repair") = (SELECT COUNT(DISTINCT "Type\_of\_repair") FROM public."Service");



The screenshot shows a SQL query editor with a query window and a results window. The query window contains the following SQL code:

```
1 SELECT e."Full_name"
2 FROM public."Employee" e
3 INNER JOIN public."Job_position" j ON e."ID_job_position" = j."ID_job_position"
4 INNER JOIN public."Contract" c ON e."ID_employee" = c."ID_employee"
5 INNER JOIN public."Distribution_of_work" d ON c."ID_contract" = d."ID_contract"
6 INNER JOIN public."Service" s ON d."ID_service" = s."ID_service"
7 WHERE j."Name" = 'Мастер-механик'
8 AND EXTRACT(WEEK FROM d."Start_date") = EXTRACT(WEEK FROM CURRENT_DATE)
9 AND EXTRACT(YEAR FROM d."Start_date") = EXTRACT(YEAR FROM CURRENT_DATE)
10 GROUP BY e."Full_name"
11 HAVING COUNT(DISTINCT s."Type_of_repair") = (SELECT COUNT(DISTINCT "Type_of_repair") FROM public."Service");
12
```

The results window shows a table with one column, "Full\_name", and one row with the value "character".

**Сколько заработал каждый мастер за прошедший месяц?**

SELECT e."Full\_name", SUM(c."Total\_payment") as "Total\_earnings"

FROM public."Employee" e

INNER JOIN public."Contract" c ON e."ID\_employee" = c."ID\_employee"

INNER JOIN public."Job\_position" j ON e."ID\_job\_position" = j."ID\_job\_position"

WHERE c."Actual\_date\_end\_of\_repair" BETWEEN date\_trunc('month', CURRENT\_DATE)  
AND date\_trunc('month', CURRENT\_DATE) + INTERVAL '1 month - 1 day'

AND j."Name" = 'Мастер-механик'

GROUP BY e."Full\_name";

Query

Query History

```

1 SELECT e."Full_name", SUM(c."Total_payment") as "Total_earnings"
2 FROM public."Employee" e
3 INNER JOIN public."Contract" c ON e."ID_employee" = c."ID_employee"
4 INNER JOIN public."Job_position" j ON e."ID_job_position" = j."ID_job_position"
5 WHERE c."Actual_date_end_of_repair" BETWEEN date_trunc('month', CURRENT_DATE) AND date_trunc('month', CURRENT_DATE) + INTERVAL '1'
6 AND j."Name" = 'Мастер-механик'
7 GROUP BY e."Full_name";
8

```

Data Output

Messages

Notifications

+

📄

▼

🗑️

🔄

📥

📶

	Full_name character	Total_earnings numeric
1	Алексей Иванов	110237
2	Анна Павлова	94616
3	Евгения Никитина	41474
4	Иван Петров	94443
5	Наталья Кузнецова	50530
6	Ольга Козлова	79036
7	Петр Михайлов	95135
8	Самолимов Иван	90834

**Вывести данные владельцев автомобилей, которые обращались в ремонт больше одного раза.**

SELECT

c."Full\_name" AS "Владелец",  
  
c."Phone" AS "Телефон",  
  
c."Email" AS "Почта",  
  
a."State\_number" AS "Госномер автомобиля",  
  
m."Car\_brand" AS "Компания",  
  
m."Model" AS "Модель",  
  
COUNT(co."ID\_client") AS "Количество посещений"

FROM public."Client" c

JOIN public."Automobile" a ON c."ID\_client" = a."ID\_client"

JOIN public."Model" m ON a."ID\_model" = m."ID\_model"

JOIN public."Contract" co ON a."ID\_client" = co."ID\_client"

GROUP BY c."Full\_name", a."State\_number",c."Phone",c."Email",m."Car\_brand",m."Model"

HAVING COUNT(co."ID\_client") > 1;

Query Query History

```

1 SELECT
2   c."Full_name" AS "Владелец",
3   c."Phone" AS "Телефон",
4   c."Email" AS "Почта",
5   a."State_number" AS "Госномер автомобиля",
6   m."Car_brand" AS "Компания",
7   m."Model" AS "Модель",
8   COUNT(co."ID_client") AS "Количество посещений"
9 FROM public."Client" c
10 JOIN public."Automobile" a ON c."ID_client" = a."ID_client"
11 JOIN public."Model" m ON a."ID_model" = m."ID_model"
12 JOIN public."Contract" co ON a."ID_client" = co."ID_client"
13 GROUP BY c."Full_name", a."State_number",c."Phone",c."Email",m."Car_brand",m."Model"
14 HAVING COUNT(co."ID_client") > 1;
15

```

Data Output Messages Notifications

	Владелец character	Телефон character	Почта character	Госномер автомобиля character	Компания character	Модель character	Количество посещений bigint
1	Сергей Васильев	89218975555	sergey@email.com	DEF123	Lexus	RX	2

**За каждый день просрочки выполнения заказа механику назначается штраф в размере 5%. Рассчитать штраф каждого механика за прошедший месяц.**

SELECT

e."Full\_name" AS "Механик",

SUM(

CASE

WHEN c."Actual\_date\_end\_of\_repair" > c."Scheduled\_date\_end\_of\_repair"

THEN (c."Actual\_date\_end\_of\_repair" - c."Scheduled\_date\_end\_of\_repair") \* 5

ELSE 0

END

```
) || '%' AS "Сумма штрафов процентах"
```

```
FROM public."Employee" e
```

```
JOIN public."Contract" c ON e."ID_employee" = c."ID_employee"
```

```
JOIN public."Job_position" jp ON e."ID_job_position" = jp."ID_job_position"
```

```
WHERE c."Actual_date_end_of_repair" >= current_date - interval '1 month'
```

```
AND jp."Name" = 'Мастер-механик'
```

```
GROUP BY e."Full_name";
```

The screenshot shows a database query editor with a query window and a results window. The query window contains the following SQL code:

```
1 SELECT
2   e."Full_name" AS "Механик",
3   SUM(
4     CASE
5       WHEN c."Actual_date_end_of_repair" > c."Scheduled_date_end_of_repair"
6       THEN (c."Actual_date_end_of_repair" - c."Scheduled_date_end_of_repair") * 5
7       ELSE 0
8     END
9   ) || '%' AS "Сумма штрафов процентах"
10 FROM public."Employee" e
11 JOIN public."Contract" c ON e."ID_employee" = c."ID_employee"
12 JOIN public."Job_position" jp ON e."ID_job_position" = jp."ID_job_position"
13 WHERE c."Actual_date_end_of_repair" >= current_date - interval '1 month'
14 AND jp."Name" = 'Мастер-механик'
15 GROUP BY e."Full_name";
16
```

The results window shows the following data:

	Механик character	Сумма штрафов процентах text
1	Иван Петров	5%
2	Петр Михайлов	5%
3	Самолимов Иван	10%

### Задание 3. Создать представление:

Для заказчиков (фамилию механика и модель автомобиля, которую он ремонтирует чаще всего)

```
CREATE VIEW public."CustomerMechanicView" AS
```

```
SELECT DISTINCT ON (e."Full_name")
```

```
  e."Full_name" AS "Mechanic_Name",
```

```

    m."Model" AS "Most_Repaired_Model",

    COUNT(*) AS "Repair_Count"

FROM

    public."Employee" e

JOIN

    public."Contract" c ON e."ID_employee" = c."ID_employee"

JOIN

    public."Automobile" a ON c."ID_auto" = a."ID_auto"

JOIN

    public."Model" m ON a."ID_model" = m."ID_model"

JOIN

    public."Job_position" jp ON e."ID_job_position" = jp."ID_job_position"

WHERE

    jp."Name" = 'Мастер-механик'

    AND c."Actual_date_end_of_repair" IS NOT NULL

GROUP BY

    e."Full_name", m."Model"

ORDER BY

    e."Full_name", COUNT(*) DESC;

```

```

1 SELECT * FROM public."CustomerMechanicView"
2

```

Data Output Messages Notifications



	Mechanic_Name character	Most_Repaired_Model character	Repair_Count bigint
1	Алексей Иванов	growth	3
2	Анна Павлова	wife	3
3	Евгения Никитина	Carolla	4
4	Иван Петров	interview	3
5	Наталья Кузнецова	public	3
6	Ольга Козлова	finish	3
7	Петр Михайлов	our	2
8	Самолимов Иван	marriage	3

Для менеджеров (рассчитать премию все механикам, которые за прошедший месяц все свои заказы выполнили своевременно - 10% от зарплаты)

```
CREATE VIEW public."ManagerBonusView" AS
```

```
SELECT
```

```
e."Full_name" AS "Mechanic_Name",
```

```
jp."Salary" * 0.1 AS "Bonus_Amount"
```

```
FROM
```

```
public."Employee" e
```

JOIN

public."Job\_position" jp ON e."ID\_job\_position" = jp."ID\_job\_position"

WHERE

e."ID\_job\_position" IN (

SELECT "ID\_job\_position"

FROM public."Job\_position"

WHERE "Name" = 'Механик'

)

AND e."ID\_employee" NOT IN (

SELECT DISTINCT ON (c."ID\_employee")

c."ID\_employee"

FROM public."Contract" c

WHERE

c."Actual\_date\_end\_of\_repair" IS NOT NULL

AND c."Order\_date" >= (SELECT date\_trunc('month', CURRENT\_DATE))

AND c."Order\_date" < (SELECT date\_trunc('month', CURRENT\_DATE) + INTERVAL  
'1 month')

AND c."Actual\_date\_end\_of\_repair" <= c."Scheduled\_date\_end\_of\_repair"

);

The screenshot shows a database query editor with a 'Query' tab. The query is: `SELECT * FROM public.manager_bonus`. Below the query, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, displaying a table with 8 rows and 2 columns: 'Mechanic\_name' (character) and 'Bonus' (numeric). The data is as follows:

	Mechanic_name character	Bonus numeric
1	Алексей Иванов	6000.00
2	Анна Павлова	6000.00
3	Евгения Никитина	6000.00
4	Иван Петров	6000.00
5	Наталья Кузнецова	6000.00
6	Ольга Козлова	6000.00
7	Петр Михайлов	6000.00
8	Самолимов Иван	6000.00

## Запросы на модификацию данных

1) UPDATE запрос, обновляющий статус заказа для всех заказов, которые были приняты к ремонту и закончены в течение месяца.

UPDATE public."Contract" c1

SET "Order\_status" = 'Completed'

WHERE "Order\_status" = 'In progress'

AND "Actual\_date\_end\_of\_repair" BETWEEN (SELECT date\_trunc('month', CURRENT\_DATE))



AND (SELECT date\_trunc('month', CURRENT\_DATE) + INTERVAL '1 month - 1 day');

До

	Order_date date	Order_status character	Payment_status character	Date_of_acceptance_for_repair date	Scheduled_date_end_of_repair date	ID_client bigint	ID_auto bigint	ID_contract [PK] bigint	Actual_date_end_of_repair date	Total_payment bigint	ID_employee bigint
515	2023-09-22	Completed	Paid	2023-10-05	2023-10-12	597	806	515	2023-10-08	2262	7
516	2023-02-19	New	Paid	2023-08-31	2023-09-16	863	111	516	2023-09-04	5335	3
517	2023-07-19	New	Refunded	2023-08-26	2023-08-26	758	890	517	2023-08-26	2093	1
518	2023-05-10	Completed	Pending	2023-05-15	2023-10-22	503	111	518	2023-09-15	2336	5
519	2023-10-23	Completed	Pending	2023-11-03	2023-11-06	416	420	519	2023-11-04	8629	3
520	2023-09-20	In progress	Refunded	2023-10-23	2023-11-04	966	303	520	2023-10-24	5336	11
521	2023-07-07	New	Refunded	2023-08-20	2023-10-12	634	540	521	2023-09-11	8509	5
522	2023-10-11	In progress	Refunded	2023-10-20	2023-10-25	515	268	522	2023-10-23	7289	5
523	2023-04-20	New	Paid	2023-08-12	2023-09-26	871	852	523	2023-09-14	4829	1
524	2023-01-24	In progress	Pending	2023-02-08	2023-06-19	339	575	524	2023-04-27	4230	6
525	2023-02-02	Completed	Refunded	2023-06-24	2023-10-17	1208	109	525	2023-08-28	8806	2
526	2023-05-11	Completed	Refunded	2023-11-06	2023-11-08	1265	433	526	2023-11-07	8541	3
527	2023-07-19	In progress	Pending	2023-10-30	2023-10-31	1392	981	527	2023-10-30	1919	9
528	2023-09-19	Completed	Refunded	2023-10-03	2023-10-26	637	225	528	2023-10-10	1465	3
529	2023-04-26	In progress	Refunded	2023-10-07	2023-10-16	1298	62	529	2023-10-10	7666	6
530	2023-08-20	Completed	Paid	2023-08-30	2023-09-09	613	618	530	2023-09-07	4087	7
531	2023-01-16	In progress	Refunded	2023-08-27	2023-10-14	402	879	531	2023-09-01	8588	3
532	2023-04-24	In progress	Paid	2023-07-22	2023-08-12	1409	320	532	2023-08-05	8533	2
533	2023-03-07	New	Pending	2023-10-08	2023-11-02	1339	915	533	2023-10-26	9975	3
534	2023-04-21	New	Refunded	2023-08-16	2023-09-30	1371	86	534	2023-09-06	2712	5
535	2023-02-15	Completed	Paid	2023-10-26	2023-10-30	996	857	535	2023-10-28	9466	7
536	2023-02-17	Completed	Pending	2023-02-17	2023-04-17	140	426	536	2023-03-27	6231	11
537	2023-10-06	New	Paid	2023-10-14	2023-11-08	796	900	537	2023-11-06	2799	5
538	2023-09-03	New	Refunded	2023-10-26	2023-10-28	982	199	538	2023-10-26	2134	10
539	2023-10-29	New	Pending	2023-10-30	2023-10-30	278	476	539	2023-10-30	1208	6
540	2023-01-24	New	Refunded	2023-09-30	2023-10-20	943	996	540	2023-10-17	7942	3
541	2023-08-14	Completed	Paid	2023-09-02	2023-10-01	1132	218	541	2023-09-02	9312	10

После

515	2023-09-22	Completed	Paid	2023-10-05	2023-10-12	597	806	515	2023-10-08	2262	7
516	2023-02-19	New	Paid	2023-08-31	2023-09-16	863	111	516	2023-09-04	5335	3
517	2023-07-19	New	Refunded	2023-08-26	2023-08-26	758	890	517	2023-08-26	2093	1
518	2023-05-10	Completed	Pending	2023-05-15	2023-10-22	503	111	518	2023-09-15	2336	5
519	2023-10-23	Completed	Pending	2023-11-03	2023-11-06	416	420	519	2023-11-04	8629	3
520	2023-09-20	In progress	Refunded	2023-10-23	2023-11-04	966	303	520	2023-10-24	5336	11
521	2023-07-07	New	Refunded	2023-08-20	2023-10-12	634	540	521	2023-09-11	8509	5
522	2023-10-11	In progress	Refunded	2023-10-20	2023-10-25	515	268	522	2023-10-23	7289	5
523	2023-04-20	New	Paid	2023-08-12	2023-09-26	871	852	523	2023-09-14	4829	1
524	2023-01-24	In progress	Pending	2023-02-08	2023-06-19	339	575	524	2023-04-27	4230	6
525	2023-02-02	Completed	Refunded	2023-06-24	2023-10-17	1208	109	525	2023-08-28	8806	2
526	2023-05-11	Completed	Refunded	2023-11-06	2023-11-08	1265	433	526	2023-11-07	8541	3
527	2023-07-19	In progress	Pending	2023-10-30	2023-10-31	1392	981	527	2023-10-30	1919	9
528	2023-09-19	Completed	Refunded	2023-10-03	2023-10-26	637	225	528	2023-10-10	1465	3
529	2023-04-26	In progress	Refunded	2023-10-07	2023-10-16	1298	62	529	2023-10-10	7666	6
530	2023-08-20	Completed	Paid	2023-08-30	2023-09-09	613	618	530	2023-09-07	4087	7
531	2023-01-16	In progress	Refunded	2023-08-27	2023-10-14	402	879	531	2023-09-01	8588	3
532	2023-04-24	In progress	Paid	2023-07-22	2023-08-12	1409	320	532	2023-08-05	8533	2
533	2023-03-07	New	Pending	2023-10-08	2023-11-02	1339	915	533	2023-10-26	9975	3
534	2023-04-21	New	Refunded	2023-08-16	2023-09-30	1371	86	534	2023-09-06	2712	5
535	2023-02-15	Completed	Paid	2023-10-26	2023-10-30	996	857	535	2023-10-28	9466	7
536	2023-02-17	Completed	Pending	2023-02-17	2023-04-17	140	426	536	2023-03-27	6231	11
537	2023-10-06	New	Paid	2023-10-14	2023-11-08	796	900	537	2023-11-06	2799	5
538	2023-09-03	New	Refunded	2023-10-26	2023-10-28	982	199	538	2023-10-26	2134	10
539	2023-10-29	New	Pending	2023-10-30	2023-10-30	278	476	539	2023-10-30	1208	6
540	2023-01-24	New	Refunded	2023-09-30	2023-10-20	943	996	540	2023-10-17	7942	3
541	2023-08-14	Completed	Paid	2023-09-02	2023-10-01	1132	218	541	2023-09-02	9312	10

2) INSERT запрос для вставки в детали в таблицу детали от клиента, если у клиента статус заказа — не оплачен.

```
INSERT INTO public."Details_from_client" ("Amount_of_detail", "ID_detail",  
"ID_distribution")
```

```
SELECT
```

```
5 AS "Amount_of_detail",
```

```
1 AS "ID_detail",
```

```
1 AS "ID_distribution"
```

```
FROM
```

```
public."Contract" c
```

```
WHERE
```

```
c."Payment_status" = 'Pending';
```

```
1 INSERT INTO public."Details_from_client" ("Amount_of_detail", "ID_detail", "ID_distribution")  
2 SELECT  
3     5 AS "Amount_of_detail",  
4     1 AS "ID_detail",  
5     1 AS "ID_distribution"  
6 FROM  
7     public."Contract" c  
8 WHERE  
9     c."Payment_status" = 'Pending';  
10
```

3) DELETE запрос, который удаляет всех клиентов, не делавших заказ больше 3 лет.

```
DELETE FROM public."Client"
```

```
WHERE "ID_client" NOT IN (

SELECT c."ID_client"

FROM public."Client" c

LEFT JOIN public."Contract" ct ON c."ID_client" = ct."ID_client"

WHERE ct."Order_date" >= (CURRENT_DATE - INTERVAL '3 years') OR ct."Order_date"

IS NULL

);
```

13	2023-10-23	Completed	Paid		2023-11-01	2023-11-02	1052	596	13	2023-11-01	9196	8
14	2023-03-22	Completed	Paid		2023-03-23	2023-08-04	132	55	14	2023-06-15	8381	8
15	2023-10-25	New	Pending		2023-10-29	2023-10-30	93	600	15	2023-10-29	4420	5
16	2023-09-29	In progress	Paid		2023-11-06	2023-11-07	908	836	16	2023-11-06	1361	4
17	2023-10-17	New	Pending		2023-10-20	2023-11-05	913	481	17	2023-10-24	8839	7
18	2023-08-01	New	Paid		2023-10-08	2023-10-27	1342	767	18	2023-10-17	4033	8
19	2023-07-20	Completed	Paid		2023-11-08	2023-11-08	1060	436	19	2023-11-08	7892	10
20	2023-10-20	In progress	Paid		2023-11-04	2023-11-05	131	80	20	2023-11-04	7781	11
21	2023-08-18	Completed	Pending		2023-08-31	2023-09-30	890	511	21	2023-09-02	4648	5
22	2023-04-02	In progress	Refunded	...	2023-04-25	2023-09-15	1334	595	22	2023-08-30	1384	9
23	2023-10-09	New	Pending		2023-10-12	2023-10-25	854	10	23	2023-10-19	2775	6
24	2023-08-17	New	Paid		2023-09-21	2023-10-14	357	541	24	2023-09-30	4414	6
25	2023-03-04	Completed	Pending		2023-08-07	2023-08-24	530	684	25	2023-08-07	7547	2
26	2023-03-28	Completed	Paid		2023-07-30	2023-10-21	1321	795	26	2023-09-05	6685	1
27	2023-04-29	Completed	Refunded	...	2023-09-18	2023-10-19	1063	730	27	2023-10-14	6913	1
28	2023-04-09	New	Refunded	...	2023-07-20	2023-10-26	1025	746	28	2023-09-25	4595	7
29	2023-11-08	Completed	Paid		2023-11-08	2023-11-08	394	681	29	2023-11-08	9008	7
30	2023-01-07	Completed	Paid		2023-09-15	2023-09-27	801	226	30	2023-09-18	4724	9

12	2023-10-19	Заказан	Оплачен	...	2023-10-21	2023-10-24	11	11	12	[null]	[null]	5
13	2023-10-23	Completed	Paid		2023-11-01	2023-11-02	1052	596	13	2023-11-01	9196	8
14	2023-03-22	Completed	Paid		2023-03-23	2023-08-04	132	55	14	2023-06-15	8381	8
15	2023-10-25	New	Pending		2023-10-29	2023-10-30	93	600	15	2023-10-29	4420	5
16	2023-09-29	In progress	Paid		2023-11-06	2023-11-07	908	836	16	2023-11-06	1361	4
17	2023-10-17	New	Pending		2023-10-20	2023-11-05	913	481	17	2023-10-24	8839	7
18	2023-08-01	New	Paid		2023-10-08	2023-10-27	1342	767	18	2023-10-17	4033	8
19	2023-07-20	Completed	Paid		2023-11-08	2023-11-08	1060	436	19	2023-11-08	7892	10
20	2023-10-20	In progress	Paid		2023-11-04	2023-11-05	131	80	20	2023-11-04	7781	11
21	2023-08-18	Completed	Pending		2023-08-31	2023-09-30	890	511	21	2023-09-02	4648	5
22	2023-04-02	In progress	Refunded	...	2023-04-25	2023-09-15	1334	595	22	2023-08-30	1384	9
23	2023-10-09	New	Pending		2023-10-12	2023-10-25	854	10	23	2023-10-19	2775	6
24	2023-08-17	New	Paid		2023-09-21	2023-10-14	357	541	24	2023-09-30	4414	6
25	2023-03-04	Completed	Pending		2023-08-07	2023-08-24	530	684	25	2023-08-07	7547	2
26	2023-03-28	Completed	Paid		2023-07-30	2023-10-21	1321	795	26	2023-09-05	6685	1
27	2023-04-29	Completed	Refunded	...	2023-09-18	2023-10-19	1063	730	27	2023-10-14	6913	1
28	2023-04-09	New	Refunded	...	2023-07-20	2023-10-26	1025	746	28	2023-09-25	4595	7
29	2023-11-08	Completed	Paid		2023-11-08	2023-11-08	394	681	29	2023-11-08	9008	7

EXPLAIN ANALYZE

SELECT \*

FROM public."Contract"

WHERE "Order\_date" >= '2023-01-01' AND "Order\_date" < '2023-02-01';

Query

Query History

1

EXPLAIN ANALYZE

2

SELECT \*

3

FROM public."Contract"

4

WHERE "Order\_date" >= '2023-01-01' AND "Order\_date" < '2023-02-01';

5

Data Output

Messages

Notifications

≡+

▼

▼

	QUERY PLAN	
	text	
1	Seq Scan on "Contract" (cost=0.00..45.98 rows=101 width=118) (actual time=0.014..0.123 rows=103 loops...	
2	Filter: (("Order_date" >= '2023-01-01'::date) AND ("Order_date" < '2023-02-01'::date))	
3	Rows Removed by Filter: 896	
4	Planning Time: 0.095 ms	
5	Execution Time: 0.148 ms	

Query

Query History

1

CREATE INDEX idx\_order\_date ON public."Contract" ("Order\_date");

2

Data Output

Messages

Notifications

CREATE INDEX

Query returned successfully in 458 msec.

Query
Query History

```

1 EXPLAIN ANALYZE
2 SELECT *
3 FROM public."Contract"
4 WHERE "Order_date" >= '2023-01-01' AND "Order_date" < '2023-02-01';

```

Data Output
Messages
Notifications

+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

	QUERY PLAN	
	text	🔒
1	Bitmap Heap Scan on "Contract" (cost=5.31..37.83 rows=101 width=118) (actual time=0.016..0.050 rows=103 loops=1)	
2	Recheck Cond: (("Order_date" >= '2023-01-01'::date) AND ("Order_date" < '2023-02-01'::date))	
3	Heap Blocks: exact=22	
4	-> Bitmap Index Scan on idx_order_date (cost=0.00..5.29 rows=101 width=0) (actual time=0.011..0.011 rows=103 loop=1)	
5	Index Cond: (("Order_date" >= '2023-01-01'::date) AND ("Order_date" < '2023-02-01'::date))	
6	Planning Time: 1.940 ms	
7	Execution Time: 0.081 ms	

DROP INDEX idx\_order\_date;

Query
Query History

```

1 DROP INDEX idx_order_date;

```

```

EXPLAIN ANALYZE
SELECT *
FROM public."Automobile"
WHERE "Year_of_vehicle" = 2020;

```

QueryQuery History

1

EXPLAIN ANALYZE

2

SELECT \*

3

FROM public."Automobile"

4

WHERE "Year\_of\_vehicle" = 2020;

5

Data OutputMessagesNotifications

+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

	QUERY PLAN	
	text	🔒
1	Seq Scan on "Automobile" (cost=0.00..32.49 rows=25 width=119) (actual time=0.011..0.102 rows=25 loops...	
2	Filter: ("Year_of_vehicle" = 2020)	
3	Rows Removed by Filter: 974	
4	Planning Time: 0.141 ms	
5	Execution Time: 0.112 ms	

CREATE INDEX idx\_year\_of\_vehicle ON public."Automobile" ("Year\_of\_vehicle");

QueryQuery History

1

CREATE INDEX idx\_year\_of\_vehicle ON public."Automobile" ("Year\_of\_vehicle");

Data OutputMessagesNotifications

CREATE INDEX

Query returned successfully in 403 msec.

QueryQuery History

1

EXPLAIN ANALYZE

2

SELECT \*

3

FROM public."Automobile"

4

WHERE "Year\_of\_vehicle" = 2020;

5

Data OutputMessagesNotifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

	QUERY PLAN	
	text	🔒
1	Bitmap Heap Scan on "Automobile" (cost=4.34..25.72 rows=25 width=119) (actual time=0.021..0.038 rows=25 loops=1)	
2	Recheck Cond: ("Year_of_vehicle" = 2020)	
3	Heap Blocks: exact=16	
4	-> Bitmap Index Scan on idx_year_of_vehicle (cost=0.00..4.34 rows=25 width=0) (actual time=0.014..0.014 rows=25 loop...	
5	Index Cond: ("Year_of_vehicle" = 2020)	
6	Planning Time: 1.580 ms	
7	Execution Time: 0.057 ms	

DROP INDEX idx\_year\_of\_vehicle;

## Вывод

В ходе проведения данной лабораторной работы были осуществлены разнообразные запросы к базе данных "Автомастерская", включая использование соединений таблиц, подзапросов и других методов. Также были выполнены запросы на создание представлений и модификацию данных, включая операции вставки, изменения и удаления.