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:

Говорова М.М.

Санкт-Петербург 2023

Оглавление

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.

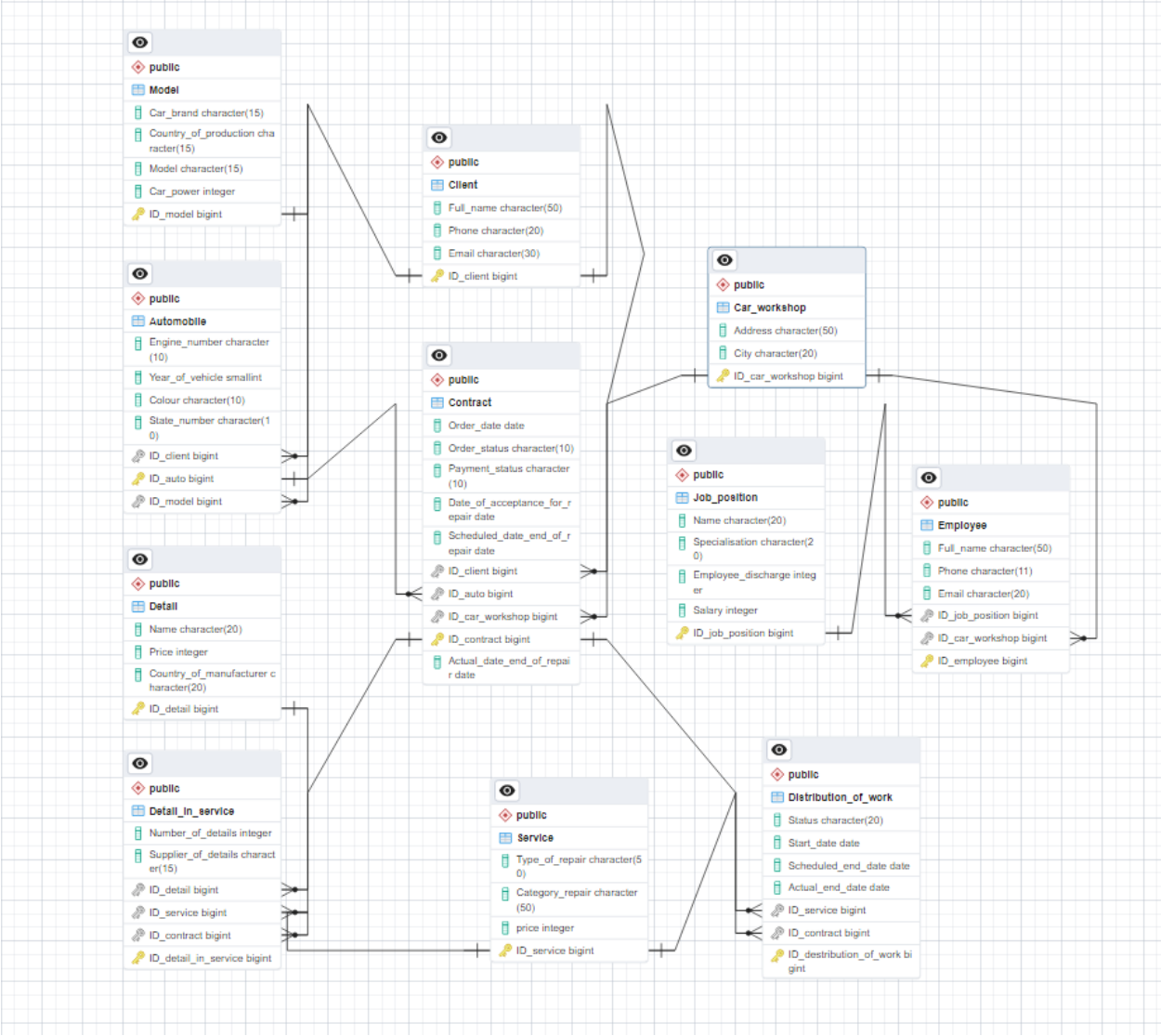


Рисунок 1 — ERD-схема базы данных

Запросы

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

```
WITH RepairCounts AS (

  SELECT

    e."ID_employee" AS "Mechanic_ID",

    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."ID_employee", e."Full_name"
)
SELECT
  "Mechanic_ID",
  "Mechanic_Name",
  "Number_of_Repairs"
FROM
  RepairCounts
```

WHERE

"Number_of_Repairs" = (

SELECT MAX("Number_of_Repairs") FROM RepairCounts

);

Query Query History

```
        e."ID_employee" AS "Mechanic_ID",
        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."ID_employee", e."Full_name"
)
SELECT
    "Mechanic_ID",
    "Mechanic_Name",
    "Number_of_Repairs"
FROM
    RepairCounts
WHERE
    "Number_of_Repairs" = (
        SELECT MAX("Number_of_Repairs") FROM RepairCounts
    );
```

Data Output Messages Notifications

	Mechanic_ID bigint	Mechanic_Name character	Number_of_Repairs bigint
1	9	Евгения Никитина	4

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

SELECT DISTINCT ON (e."ID_employee")

e."ID_employee" AS "ID_Механика",

e."Full_name" AS "Механик",

cl."ID_client" AS "ID_Постоянного_клиента",

```
    cl."Full_name" AS "Постоянный_клиент"
FROM
    public."Employee" e
JOIN
    public."Contract" c ON e."ID_employee" = c."ID_employee"
JOIN
    public."Client" cl ON c."ID_client" = cl."ID_client"
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
    )
ORDER BY
    e."ID_employee", cl."ID_client";
```

Query | Query History

```
SELECT DISTINCT ON (e."ID_employee")
    e."ID_employee" AS "ID_Механика",
    e."Full_name" AS "Механик",
    cl."ID_client" AS "ID_Постоянного_клиента",
    cl."Full_name" AS "Постоянный_клиент"
FROM
    public."Employee" e
JOIN
    public."Contract" c ON e."ID_employee" = c."ID_employee"
JOIN
    public."Client" cl ON c."ID_client" = cl."ID_client"
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
    )
ORDER BY
    e."ID_employee", cl."ID_client";
```

Data Output | Messages | Notifications

	ID_Механика bigint	Механик character	ID_Постоянного_клиента bigint	Постоянный_клиент character
1	2	Иван Петров	2	Ospelnikov Alexey
2	3	Самолимов Иван	6	Наталья Козлова
3	4	Алексей Иванов	5	Андрей Иванов
4	5	Ольга Козлова	50	Angel Olsen
5	7	Анна Павлова	122	Karen Cohen
6	8	Петр Михайлов	9	Павел Морозов
7	9	Евгения Никитина	54	Cindy Martinez
8	11	Наталья Кузнецова	131	Steven Little

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

SELECT

e."ID_employee" AS "ID_Механика",

e."Full_name" AS "Механик_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

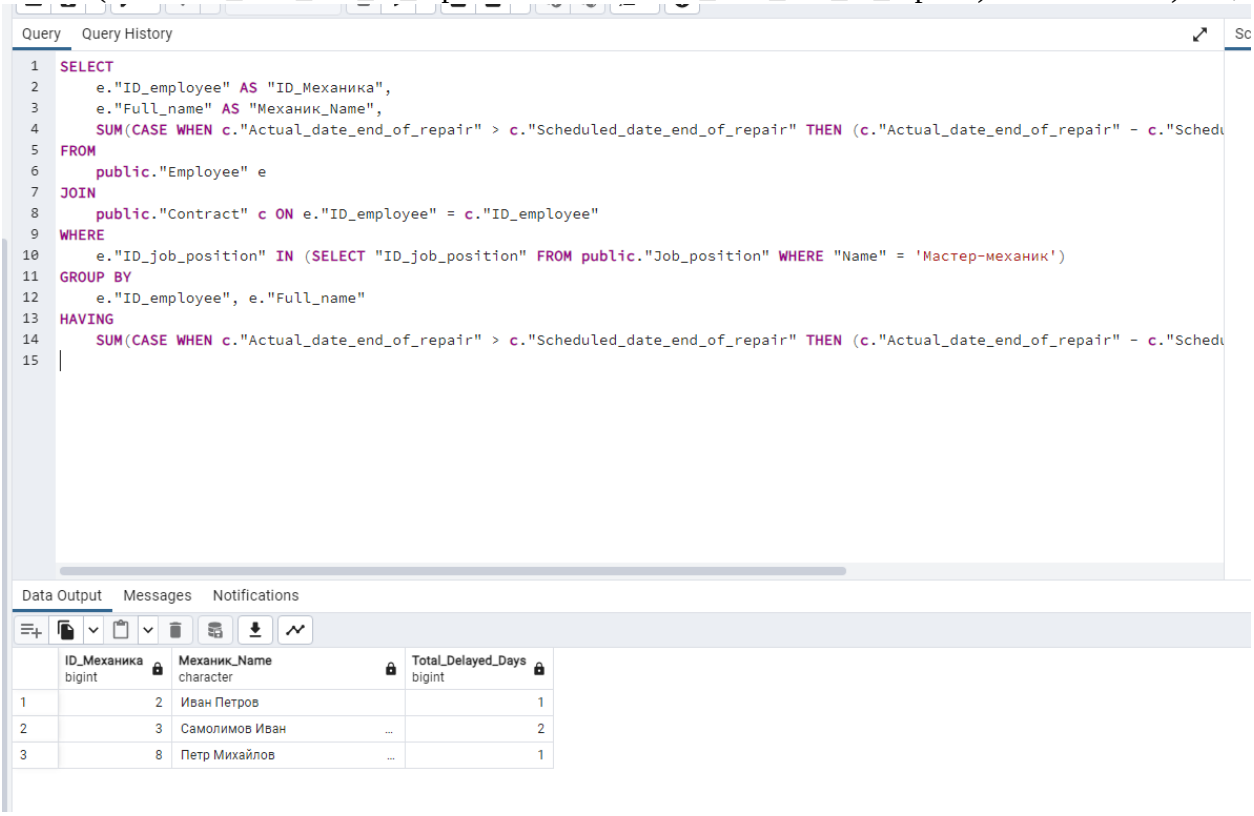
e."ID_job_position" IN (SELECT "ID_job_position" FROM public."Job_position" WHERE "Name" = 'Мастер-механик')

GROUP BY

e."ID_employee", 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;



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

SELECT

e."ID_employee" AS "ID_Механика",

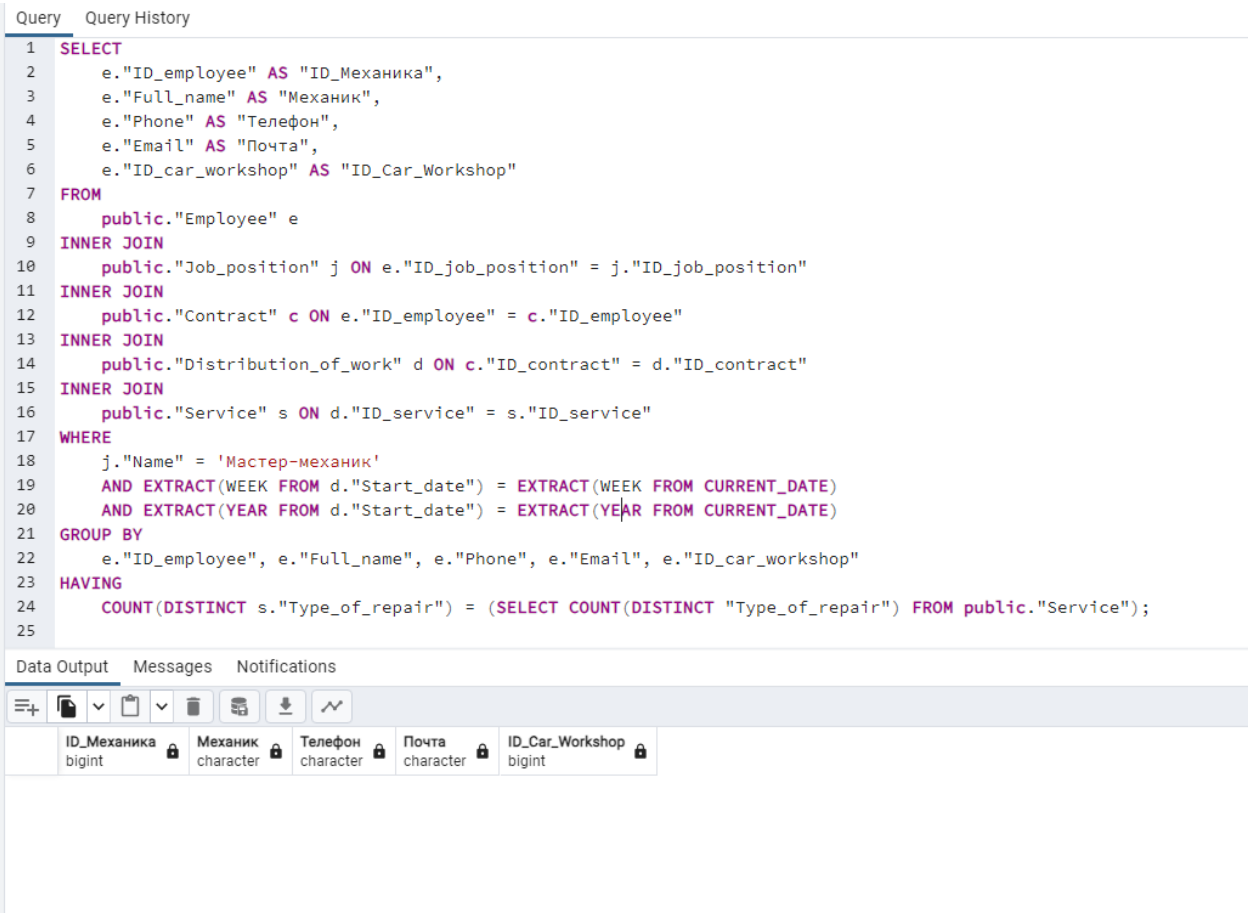
e."Full_name" AS "Механик",

e."Phone" AS "Телефон",

e."Email" AS "Почта",

e."ID_car_workshop" AS "ID_Car_Workshop"

```
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."ID_employee", e."Full_name", e."Phone", e."Email", e."ID_car_workshop"
HAVING
    COUNT(DISTINCT s."Type_of_repair") = (SELECT COUNT(DISTINCT "Type_of_repair")
FROM public."Service");
```



Сколько заработал каждый мастер за прошедший месяц?

SELECT

e."ID_employee" AS "ID_Механика",

e."Full_name" AS "Механик",

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."ID_employee", e."Full_name";

Query Query History

```
SELECT
    e."ID_employee" AS "ID_Механика",
    e."Full_name" AS "Механик",
    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 mo
    AND j."Name" = 'Мастер-механик'
GROUP BY
    e."ID_employee", e."Full_name";
```

Data Output Messages Notifications

	ID_Механика bigint	Механик character	Total_earnings numeric
1	2	Иван Петров	94443
2	3	Самолимов Иван	90834
3	4	Алексей Иванов	110237
4	5	Ольга Козлова	79036
5	7	Анна Павлова	94616
6	8	Петр Михайлов	95135
7	9	Евгения Никитина	41474
8	11	Наталья Кузнецова	50530

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

SELECT

c."ID_client" AS "ID_Владельца",

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_auto" = co."ID_auto"
GROUP BY
    c."ID_client", c."Full_name", c."Phone", c."Email", a."State_number", m."Car_brand", m."Model"
HAVING
    COUNT(co."ID_client") > 1;
```

Query | Query History | Scratch Pad

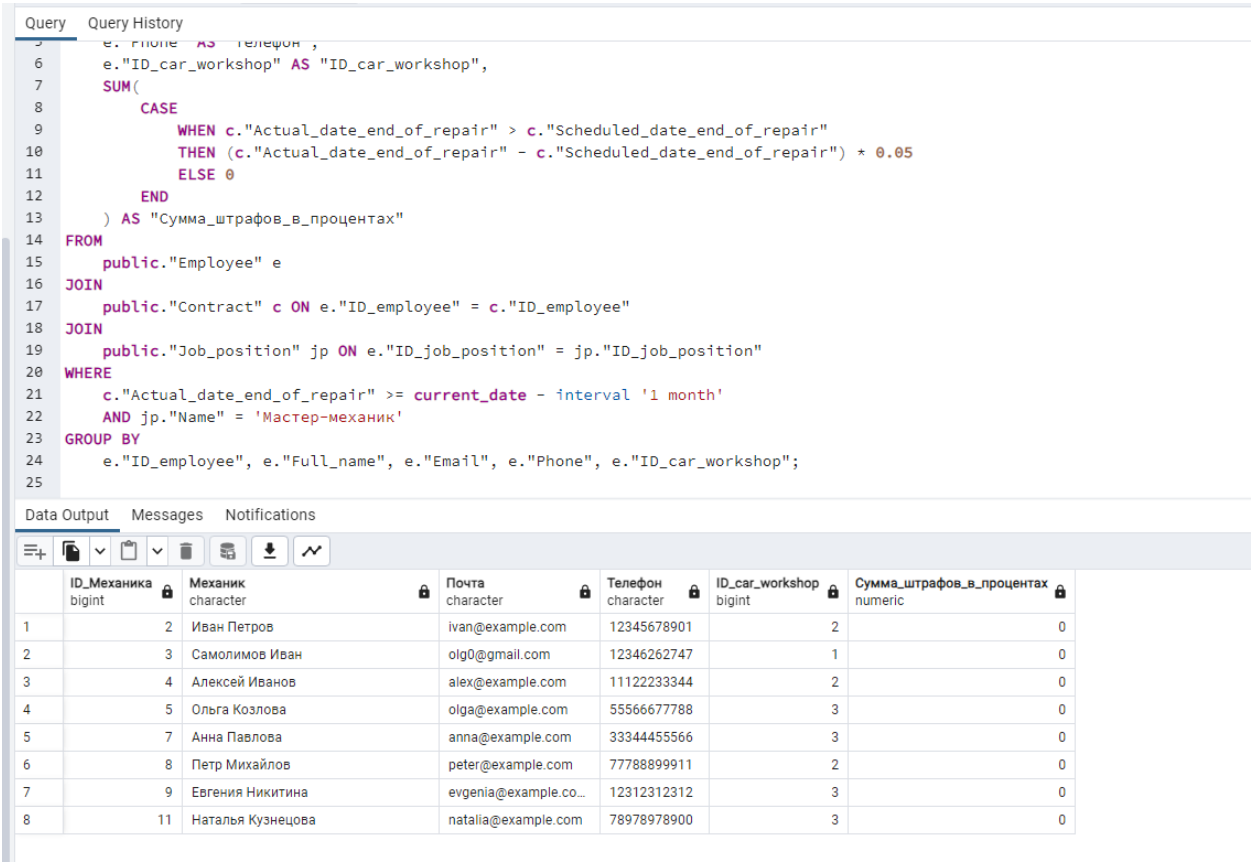
```
SELECT
    c."ID_client" AS "ID_Владельца",
    c."Full_name" AS "Владелец",
    c."Phone" AS "Телефон",
    c."Email" AS "Почта",
    a."State_number" AS "Госномер автомобиля",
    m."Car_brand" AS "Компания",
    m."Model" AS "Модель",
    COUNT(co."ID_client") AS "Количество_посещений"
FROM
    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_auto" = co."ID_auto"
GROUP BY
    c."ID_client", c."Full_name", c."Phone", c."Email", a."State_number", m."Car_brand", m."Model"
HAVING
```

Data Output | Messages | Notifications

	ID_Владельца bigint	Владелец character	Телефон character	Почта character	Госномер автомобиля character	Компания character	Модель character
1	636	Calvin Griffin	(744)559-0222x8385	kathy27@example.com	L 794149	Jackson-Byrd	same
2	623	Michael Ramirez	001-663-461-8933	wmcguire@example.net	G53-VSO	Noble-Thompson	us
3	929	Aaron Atkinson	(683)667-5842	lawsoncathy@example.com	A91 7VZ	Parker PLC	effect
4	257	Tiffany May	212.467.5989x7954	bellcurtis@example.net	638 DVB	Miller and Sons	national
5	542	Thomas White	(457)551-1167	jamesburton@example.net	615 AZW	Torres, Dunn and Mays	hold
6	627	Tiffany Williams	880-651-1692	johnshaw@example.net	660 XWU	Ortiz, Simmons and Craig	court
7	1126	Richard Taylor	+1-357-686-5070x890	monica41@example.net	YMO 708	Greene and Sons	just
8	785	Tyler Torres	(689)295-3619x3735	brittanychristensen@example.com	CTT7334	Hunter PLC	edge
9	1190	Patricia Johnson	(522)377-3973x026	ymyers@example.org	656 NGN	Ingram, Stevens and Patton	by
10	43	Amanda Burns	001-799-271-2613x44250	jessica21@example.com	46E 444	Mcneil-Graham	our
11	66	Danielle Rivas	001-706-410-0850x172	amanda02@example.org	289X	Rodriguez-Ramos	investment
12	332	Deanna Gardner	(632)785-3250x9955	fboyd@example.net	6A732	Bush LLC	plan

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

```
SELECT
   e."ID_employee" AS "ID_Механика",
   e."Full_name" AS "Механик",
   e."Email" AS "Почта",
   e."Phone" AS "Телефон",
   e."ID_car_workshop" AS "ID_car_workshop",
   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") * 0.05
         ELSE 0
      END
   ) AS "Сумма_штрафов_в_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."ID_employee", e."Full_name", e."Email", e."Phone", e."ID_car_workshop";
```



Query | Query History

```
    e."Phone" AS "Телефон",
    e."ID_car_workshop" AS "ID_car_workshop",
    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") * 0.05
            ELSE 0
        END
    ) AS "Сумма_штрафов_в_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."ID_employee", e."Full_name", e."Email", e."Phone", e."ID_car_workshop";
```

Data Output | Messages | Notifications

	ID_Механика bigint	Механик character	Почта character	Телефон character	ID_car_workshop bigint	Сумма_штрафов_в_процентах numeric
1	2	Иван Петров	ivan@example.com	12345678901	2	0
2	3	Самолимов Иван	olg0@gmail.com	12346262747	1	0
3	4	Алексей Иванов	alex@example.com	11122233344	2	0
4	5	Ольга Козлова	olga@example.com	55566677788	3	0
5	7	Анна Павлова	anna@example.com	33344455566	3	0
6	8	Петр Михайлов	peter@example.com	77788899911	2	0
7	9	Евгения Никитина	evgenia@example.co...	12312312312	3	0
8	11	Наталья Кузнецова	natalia@example.com	78978978900	3	0

Задание 3. Создать представление:

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

```
CREATE VIEW public."CustomerMechanicView" AS

WITH RepairCounts AS (

    SELECT

        e."ID_employee",

        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."ID_employee", m."Model"
),
MaxRepairs AS (
    SELECT
        "ID_employee",
        MAX("Repair_Count") AS "Max_Repair_Count"
    FROM
        RepairCounts
    GROUP BY
        "ID_employee"
)
SELECT
    e."ID_employee" AS "Mechanic_ID",
    e."Full_name" AS "Mechanic_Name",
    rc."Most_Repaired_Model",
    rc."Repair_Count"
FROM
```

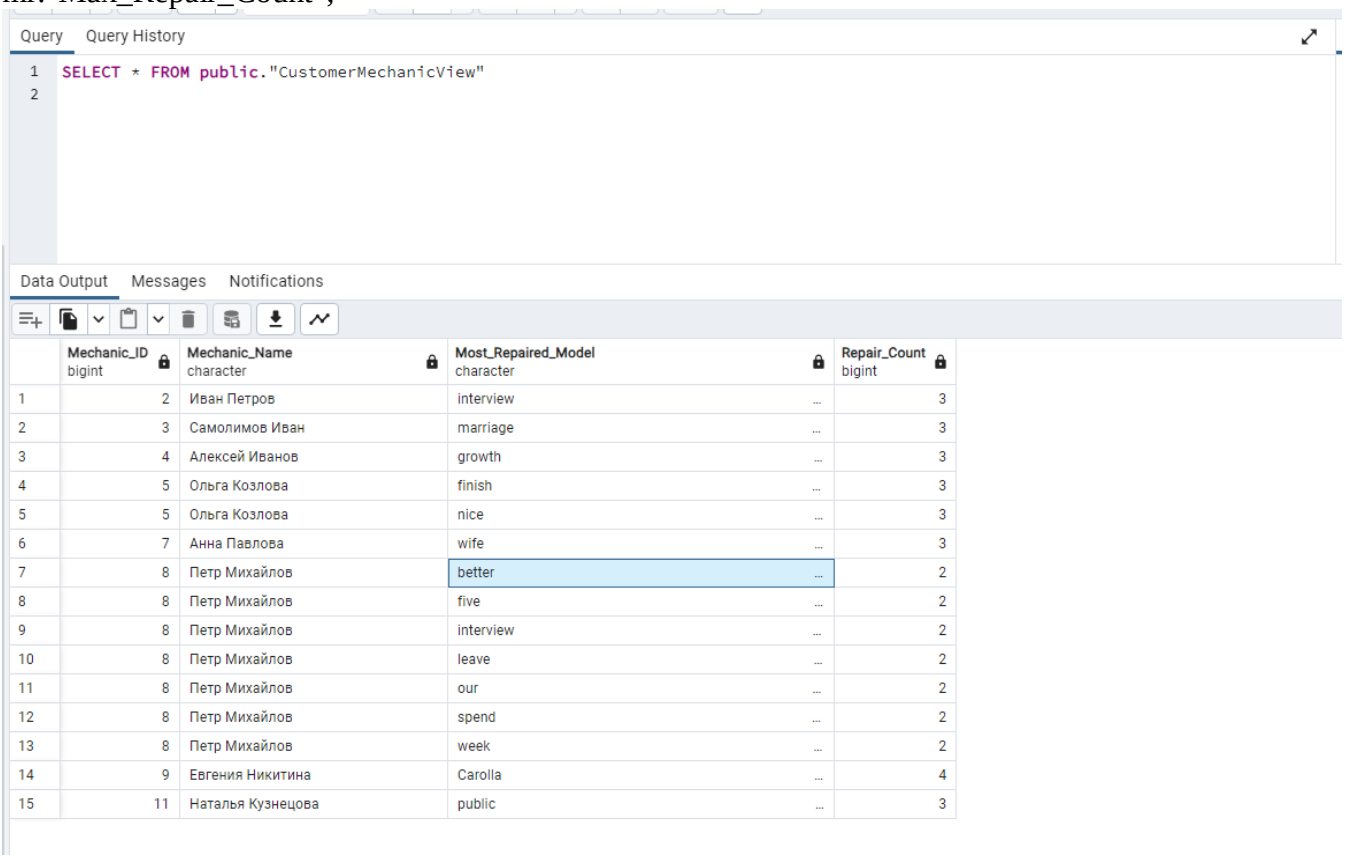
public."Employee" e

JOIN

RepairCounts rc ON e."ID_employee" = rc."ID_employee"

JOIN

MaxRepairs mr ON rc."ID_employee" = mr."ID_employee" AND rc."Repair_Count" = mr."Max_Repair_Count";



Query | Query History

```
1  SELECT * FROM public."CustomerMechanicView"
2
```

Data Output | Messages | Notifications

	Mechanic_ID bigint	Mechanic_Name character	Most_Repaired_Model character	Repair_Count bigint
1	2	Иван Петров	interview	3
2	3	Самолимов Иван	marriage	3
3	4	Алексей Иванов	growth	3
4	5	Ольга Козлова	finish	3
5	5	Ольга Козлова	nice	3
6	7	Анна Павлова	wife	3
7	8	Петр Михайлов	better	2
8	8	Петр Михайлов	five	2
9	8	Петр Михайлов	interview	2
10	8	Петр Михайлов	leave	2
11	8	Петр Михайлов	our	2
12	8	Петр Михайлов	spend	2
13	8	Петр Михайлов	week	2
14	9	Евгения Никитина	Carolla	4
15	11	Наталья Кузнецова	public	3

2)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",

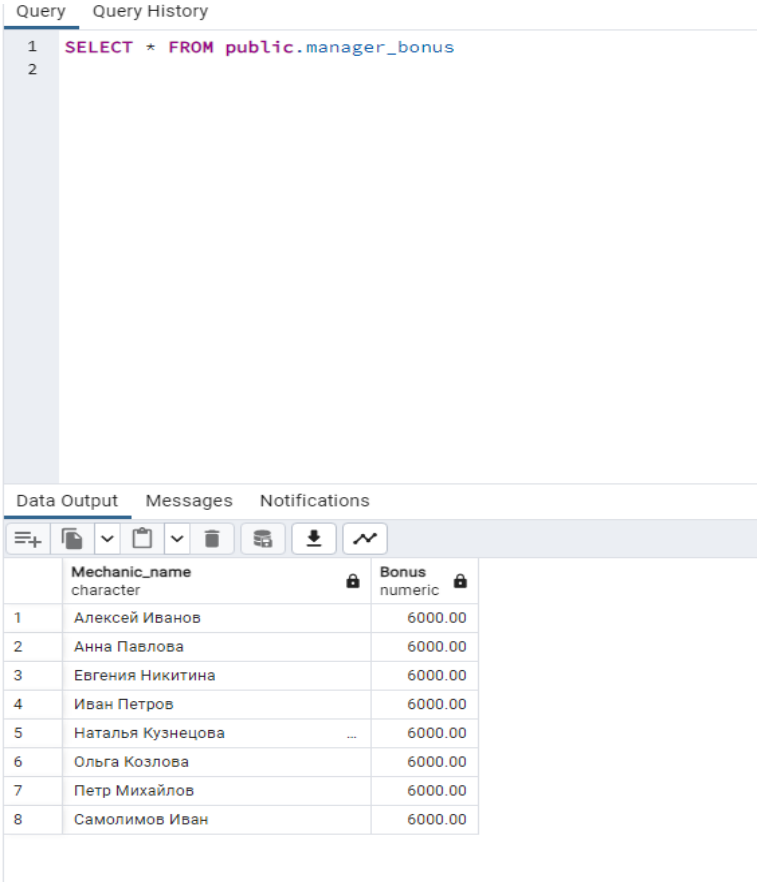
jp."Salary" * 0.1 AS "Bonus_Amount"

```
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" = 'Мастер-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"

);



	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 деталей 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';

```
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';
```

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

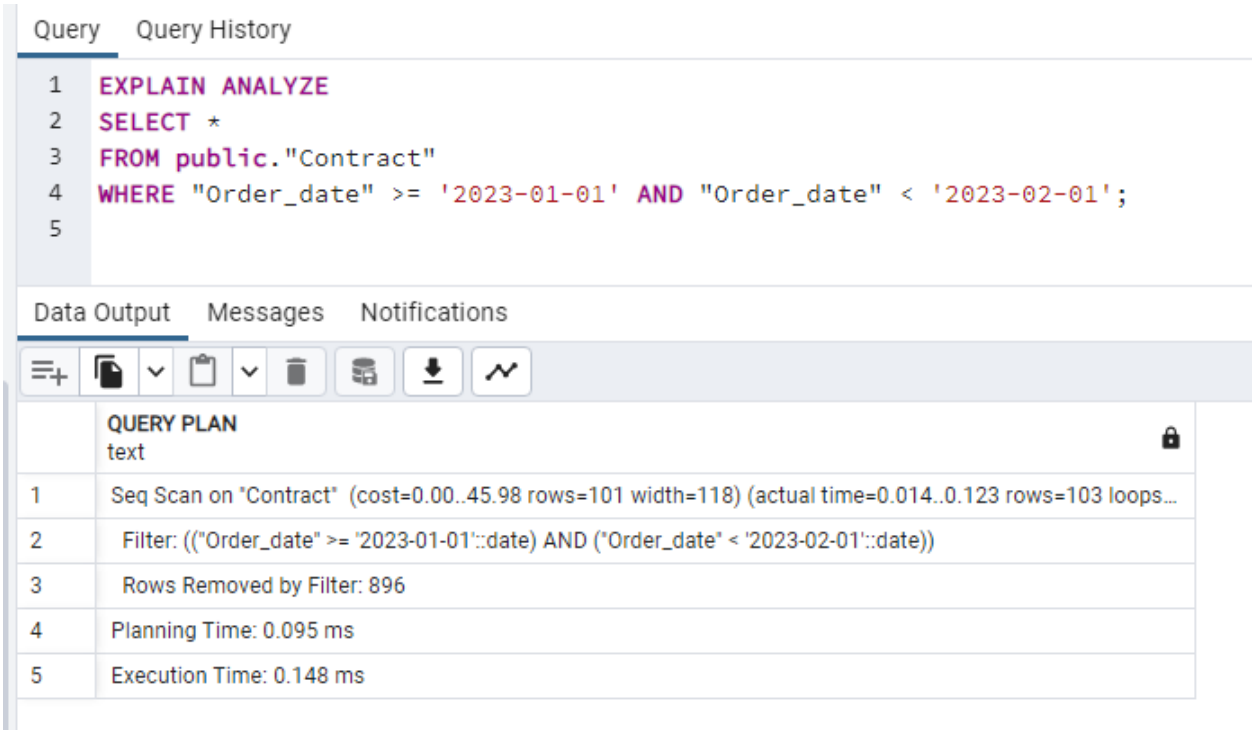
Индексы

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';
```

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...
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;

Query Query History

```
1  EXPLAIN ANALYZE
2  SELECT *
3  FROM public."Automobile"
4  WHERE "Year_of_vehicle" = 2020;
5
```

Data Output Messages Notifications

	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");

Query Query History

```
1  CREATE INDEX idx_year_of_vehicle ON public."Automobile" ("Year_of_vehicle");
```

Data Output Messages Notifications

```
CREATE INDEX

Query returned successfully in 403 msec.
```

Query Query History

```
1  EXPLAIN ANALYZE
2  SELECT *
3  FROM public."Automobile"
4  WHERE "Year_of_vehicle" = 2020;
5
```

Data Output Messages Notifications

	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;

Составной индекс:

Query Query History

```
1  EXPLAIN ANALYZE
2  SELECT *
3  FROM public."Employee"
4  WHERE "ID_job_position" = 1
5    AND "ID_car_workshop" = 3;
```

Data Output Messages Notifications

	QUERY PLAN text
1	Seq Scan on "Employee" (cost=0.00..13.15 rows=1 width=360) (actual time=0.011..0.011 rows=0 loops...
2	Filter: (("ID_job_position" = 1) AND ("ID_car_workshop" = 3))
3	Rows Removed by Filter: 11
4	Planning Time: 0.073 ms
5	Execution Time: 0.019 ms

Query Query History

```
1  EXPLAIN ANALYZE
2  SELECT *
3  FROM public."Employee"
4  WHERE "Email" = 'alex@example.com';
```

Data Output Messages Notifications

	QUERY PLAN text
1	Seq Scan on "Employee" (cost=0.00..12.62 rows=1 width=360) (actual time=0.037..0.038 rows=1 loops...
2	Filter: ("Email" = 'alex@example.com'::bpchar)
3	Rows Removed by Filter: 10
4	Planning Time: 0.073 ms
5	Execution Time: 0.048 ms

CREATE INDEX idx_employee_email_job_position_workshop

ON public."Employee" ("Email", "ID_job_position", "ID_car_workshop");

Query Query History

```
1  EXPLAIN ANALYZE
2  SELECT *
3  FROM public."Employee"
4  WHERE "Email" = 'alex@example.com';
```

Data Output Messages Notifications

	QUERY PLAN text
1	Seq Scan on "Employee" (cost=0.00..1.14 rows=1 width=360) (actual time=0.013..0.014 rows=1 loops...
2	Filter: ("Email" = 'alex@example.com'::bpchar)
3	Rows Removed by Filter: 10
4	Planning Time: 1.527 ms
5	Execution Time: 0.022 ms

Query Query History

```
1  EXPLAIN ANALYZE
2  SELECT *
3  FROM public."Employee"
4  WHERE "ID_job_position" = 1
5    AND "ID_car_workshop" = 3;
```

Data Output Messages Notifications

	QUERY PLAN text
1	Seq Scan on "Employee" (cost=0.00..1.17 rows=1 width=360) (actual time=0.020..0.020 rows=0 loops...
2	Filter: (("ID_job_position" = 1) AND ("ID_car_workshop" = 3))
3	Rows Removed by Filter: 11
4	Planning Time: 0.120 ms
5	Execution Time: 0.033 ms

DROP INDEX idx_employee_email_job_position_workshop

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.