

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: Шаповалов Илья Андреевич

Факультет: ИКТ

Группа: K3239

Преподаватель: Говорова М.М.



Оглавление

| | |
|---------------------------------|----|
| Цель работы | 3 |
| Практическое задание | 3 |
| Вариант 11. БД «Автомастерская» | 3 |
| Выполнение | 3 |
| Вывод | 23 |

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.

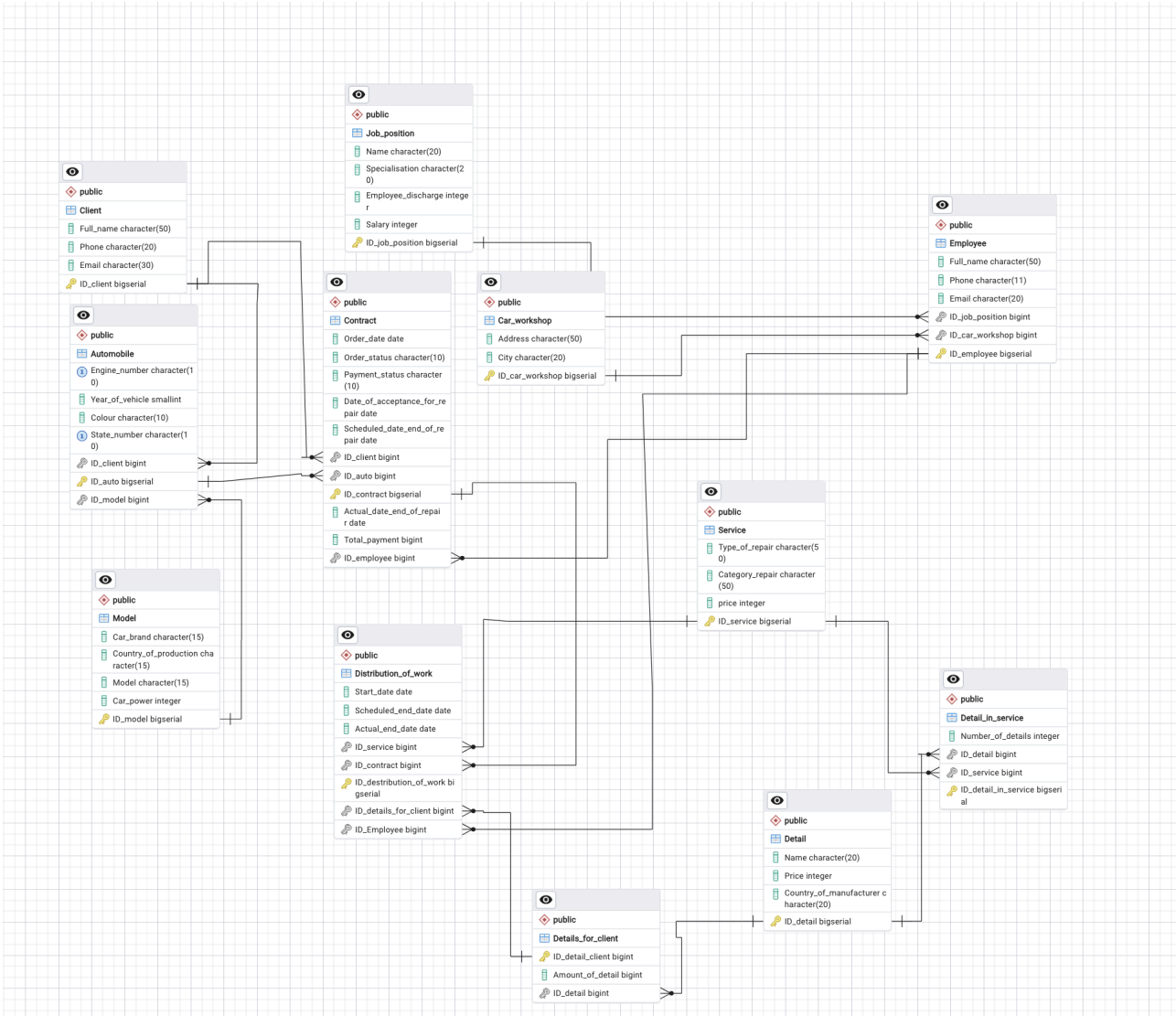


Рисунок 1 — ERD-схема базы данных

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  
    );
```

```
1 WITH RepairCounts AS (  
2     SELECT  
3         e."ID_employee" AS "Mechanic_ID",  
4         e."Full_name" AS "Mechanic_Name",  
5         COUNT(*) AS "Number_of_Repairs"  
6     FROM  
7         public."Employee" e  
8     JOIN public."Contract" c ON e."ID_employee" = c."ID_employee"  
9     JOIN public."Automobile" a ON c."ID_auto" = a."ID_auto"  
10    JOIN public."Model" m ON a."ID_model" = m."ID_model"  
11    JOIN public."Job_position" jp ON e."ID_job_position" = jp."ID_job_position"  
12    WHERE  
13        m."Car_brand" = 'Toyota' AND jp."Name" = 'Мастер-механик'  
14    GROUP BY  
15        e."ID_employee", e."Full_name"  
16 )  
17 SELECT  
18     "Mechanic_ID",  
19     "Mechanic_Name",  
20     "Number_of_Repairs"  
21 FROM  
22     RepairCounts  
23 WHERE  
24     "Number_of_Repairs" = (  
25         SELECT MAX("Number_of_Repairs") FROM RepairCounts  
26     );
```

Data Output Сообщения Notifications

| | Mechanic_ID bigint | Mechanic_Name character | Number_of_Repairs bigint |
|---|-----------------------|----------------------------|-----------------------------|
| 1 | 2 | Иван Петров | 1 |

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";
```

Запрос
История запросов
Scr

```

1  SELECT DISTINCT ON (e."ID_employee")
2    e."ID_employee" AS "ID_Механика",
3    e."Full_name" AS "Механик",
4    cl."ID_client" AS "ID_Постоянного_клиента",
5    cl."Full_name" AS "Постоянный_клиент"
6  FROM
7    public."Employee" e
8  JOIN
9    public."Contract" c ON e."ID_employee" = c."ID_employee"
10 JOIN
11   public."Client" cl ON c."ID_client" = cl."ID_client"
12 WHERE
13   e."ID_job_position" = (
14     SELECT "ID_job_position"
15     FROM public."Job_position"
16     WHERE "Name" = 'Мастер-механик'
17   )
18   AND c."ID_client" IN (
19     SELECT "ID_client"
20     FROM public."Contract"
21     GROUP BY "ID_client"
22     HAVING COUNT(DISTINCT "ID_employee") = 1
23   )
24 ORDER BY
25   e."ID_employee", cl."ID_client";

```

Data Output
Сообщения
Notifications

| | ID_Механика bigint | Механик character | ID_Постоянного_клиента bigint | Постоянный_клиент character |
|---|-----------------------|----------------------|----------------------------------|--------------------------------|
| 1 | 2 | Иван Петров | 2 | Ospelnikov Alexey |

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;
```

Запрос История запросов

```
1 SELECT
2     e."ID_employee" AS "ID_Механика",
3     e."Full_name" AS "Механик_Name",
4     SUM(CASE WHEN c."Actual_date_end_of_repair" > c."Scheduled_date_end_of_repair" THEN (c."Actual_date_end_of_repair" - c."Schedu
5 FROM
6     public."Employee" e
7 JOIN
8     public."Contract" c ON e."ID_employee" = c."ID_employee"
9 WHERE
10    e."ID_job_position" IN (SELECT "ID_job_position" FROM public."Job_position" WHERE "Name" = 'Мастер-механик')
11 GROUP BY
12     e."ID_employee", e."Full_name"
13 HAVING
14     SUM(CASE WHEN c."Actual_date_end_of_repair" > c."Scheduled_date_end_of_repair" THEN (c."Actual_date_end_of_repair" - c."Schedu
15
16
```

Data Output Сообщения Notifications

| ID_Механика bigint | Механик_Name character | Total_Delayed_Days bigint |
|-----------------------|---------------------------|------------------------------|
| 1 | Иван Петров | 1 |

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");
```



```
1
2 SELECT
3     e."ID_employee" AS "ID_Механика",
4     e."Full_name" AS "Механик",
5     SUM(c."Total_payment") as "Total_earnings"
6 FROM
7     public."Employee" e
8 INNER JOIN
9     public."Contract" c ON e."ID_employee" = c."ID_employee"
10 INNER JOIN
11     public."Job_position" j ON e."ID_job_position" = j."ID_job_position"
12 WHERE
13     c."Actual_date_end_of_repair" BETWEEN date_trunc('month', CURRENT_DATE) AND date_trunc('month', CURRENT_DATE) + INTERVAL '1 month'
14     AND j."Name" = 'Мастер-механик'
15 GROUP BY
16     e."ID_employee", e."Full_name";
17
18
```

Data Output

Сообщения

Notifications

ID_Механика

bigint

Механик

character

Total_earnings

numeric

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;
```

```

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;

```

ta Output Сообщения Notifications

ID_Владельца

bigint

Владелец

character

Телефон

character

Почта

character

Госномер автомобиля

character

Компания

character

Модель

character

Количество_посещений

bigint

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" =
'Mастер-механик'
GROUP BY
    e."ID_employee", e."Full_name", e."Email", e."Phone", e."ID_car_workshop";
```

```
1 SELECT
2     e."ID_employee" AS "ID_Механика",
3     e."Full_name" AS "Механик",
4     e."Email" AS "Почта",
5     e."Phone" AS "Телефон",
6     e."ID_car_workshop" AS "ID_car_workshop",
7     SUM(
8         CASE
9             WHEN c."Actual_date_end_of_repair" > c."Scheduled_date_end_of_repair"
10            THEN (c."Actual_date_end_of_repair" - c."Scheduled_date_end_of_repair") * 0.05
11            ELSE 0
12        END
13    ) AS "Сумма_штрафов_в_процентах"
14 FROM
15     public."Employee" e
16 JOIN
17     public."Contract" c ON e."ID_employee" = c."ID_employee"
18 JOIN
19     public."Job_position" jp ON e."ID_job_position" = jp."ID_job_position"
20 WHERE
21     c."Actual_date_end_of_repair" >= current_date - interval '1 month' AND jp."Name" = 'Мастер-механик'
22 GROUP BY
23     e."ID_employee", e."Full_name", e."Email", e."Phone", e."ID_car_workshop";
24
```

<

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";
```

Запрос

История запросов

1

SELECT * FROM public."CustomerMechanicView"

Data Output

Сообщения

Notifications

☰

+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈












| | Mechanic_ID bigint | Mechanic_Name character | ... | Most_Repaired_Model character | Repair_Count bigint |
|---|-----------------------|----------------------------|-----|----------------------------------|------------------------|
| 1 | 2 | Иван Петров | | Camry | 1 |

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"
    );
```

```
1 SELECT * FROM public."ManagerBonusView"
```

Data Output Сообщения Notifications

|          | | | |
|---|----------------------------|---|---|
| | Mechanic_Name character |  | Bonus_Amount numeric  |
| 1 | Иван Петров | ... | 6000.0 |

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');
```

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  
    c."Amount_of_detail",  
    c."ID_detail",  
    c."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  
);
```

Индексы:

1
2
3
4

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

Data Output

Сообщения

Notifications

+

+

📄

▼

📋

▼

🗑

🗄

⬇

📈

| | QUERY PLAN | |
|---|--|---|
| | text | 🔒 |
| 1 | Seq Scan on "Contract" (cost=0.00..17.05 rows=2 width=144) (actual time=0.048..0.048 rows=0 loops... | |
| 2 | Filter: (("Order_date" >= '2023-01-01'::date) AND ("Order_date" < '2023-02-01'::date)) | |
| 3 | Rows Removed by Filter: 11 | |
| 4 | Planning Time: 0.164 ms | |
| 5 | Execution Time: 0.085 ms | |

1

CREATE INDEX idx_order_date ON public."Contract" ("Order_date");

Data Output

Сообщения

Notifications

CREATE INDEX

+

+

📄

▼

📋

▼

🗑

🗄

⬇

📈

| | QUERY PLAN | |
|---|---|---|
| | text | 🔒 |
| 1 | Seq Scan on "Contract" (cost=0.00..1.17 rows=1 width=144) (actual time=0.016..0.016 rows=0 loops... | |
| 2 | Filter: (("Order_date" >= '2023-01-01'::date) AND ("Order_date" < '2023-02-01'::date)) | |
| 3 | Rows Removed by Filter: 11 | |
| 4 | Planning Time: 0.709 ms | |
| 5 | Execution Time: 0.030 ms | |

```
1 DROP INDEX idx_order_date;
```

Data Output Сообщения Notifications

DROP INDEX

Запрос завершён успешно, время выполнения: 25 msec.

Составной индекс:

Запрос История запросов

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

Data Output Сообщения Notifications



QUERY PLAN

text



| | |
|---|--|
| 1 | Seq Scan on "Employee" (cost=0.00..13.15 rows=1 width=360) (actual time=0.024..0.025 rows=0 loops... |
| 2 | Filter: (("ID_job_position" = 1) AND ("ID_car_workshop" = 3)) |
| 3 | Rows Removed by Filter: 10 |
| 4 | Planning Time: 0.281 ms |
| 5 | Execution Time: 0.055 ms |

```

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

```

Data Output Сообщения Notifications



| | QUERY PLAN | |
|---|--|--|
| | text | |
| 1 | Seq Scan on "Employee" (cost=0.00..12.62 rows=1 width=360) (actual time=0.019..0.022 rows=1 loops... | |
| 2 | Filter: ("Email" = 'alex@example.com'::bpchar) | |
| 3 | Rows Removed by Filter: 9 | |
| 4 | Planning Time: 0.101 ms | |
| 5 | Execution Time: 0.039 ms | |

**CREATE INDEX idx_employee_email_job_position_workshop
ON public."Employee" ("Email", "ID_job_position", "ID_car_workshop")**

| | QUERY PLAN | |
|---|---|--|
| | text | |
| 1 | Seq Scan on "Employee" (cost=0.00..1.12 rows=1 width=360) (actual time=0.008..0.009 rows=1 loops... | |
| 2 | Filter: ("Email" = 'alex@example.com'::bpchar) | |
| 3 | Rows Removed by Filter: 9 | |
| 4 | Planning Time: 0.433 ms | |
| 5 | Execution Time: 0.019 ms | |

```

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

```

Data Output Сообщения Notifications



| | QUERY PLAN | |
|---|---|--|
| | text | |
| 1 | Seq Scan on "Employee" (cost=0.00..1.15 rows=1 width=360) (actual time=0.030..0.031 rows=0 loops... | |
| 2 | Filter: (("ID_job_position" = 1) AND ("ID_car_workshop" = 3)) | |
| 3 | Rows Removed by Filter: 10 | |
| 4 | Planning Time: 0.171 ms | |
| 5 | Execution Time: 0.047 ms | |

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.

