

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
Факультет инфокоммуникационных технологий

**ОТЧЕТ
О ЛАБОРАТОРНОЙ РАБОТЕ № 2**

по теме:

*«Запросы на выборку и модификацию данных, представления и
индексы в PostgreSQL»*

по дисциплине: Проектирование и реализация баз данных

Специальность:

45.03.04 Интеллектуальные системы в гуманитарной сфере

Проверила:

Говорова М.М.

Дата: «..» ... 2023 г.

Оценка _____

Выполнил:

студент группы К32422

Малаев С.Г.

Санкт-Петербург 2022/2023

Цель работы: овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

Практическое задание:

1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию, часть 2 и 3).
2. Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.
3. Изучить графическое представление запросов и посмотреть историю запросов.
4. Создать простой и составной индексы для двух произвольных запросов и сравнить время выполнения запросов без индексов и с индексами. Для получения плана запроса использовать команду EXPLAIN.

Выполнение работы:

Предметная область – автомастерская (вариант 11)

Наименование БД – auto_repair_shop

Схемы логической модели базы данных

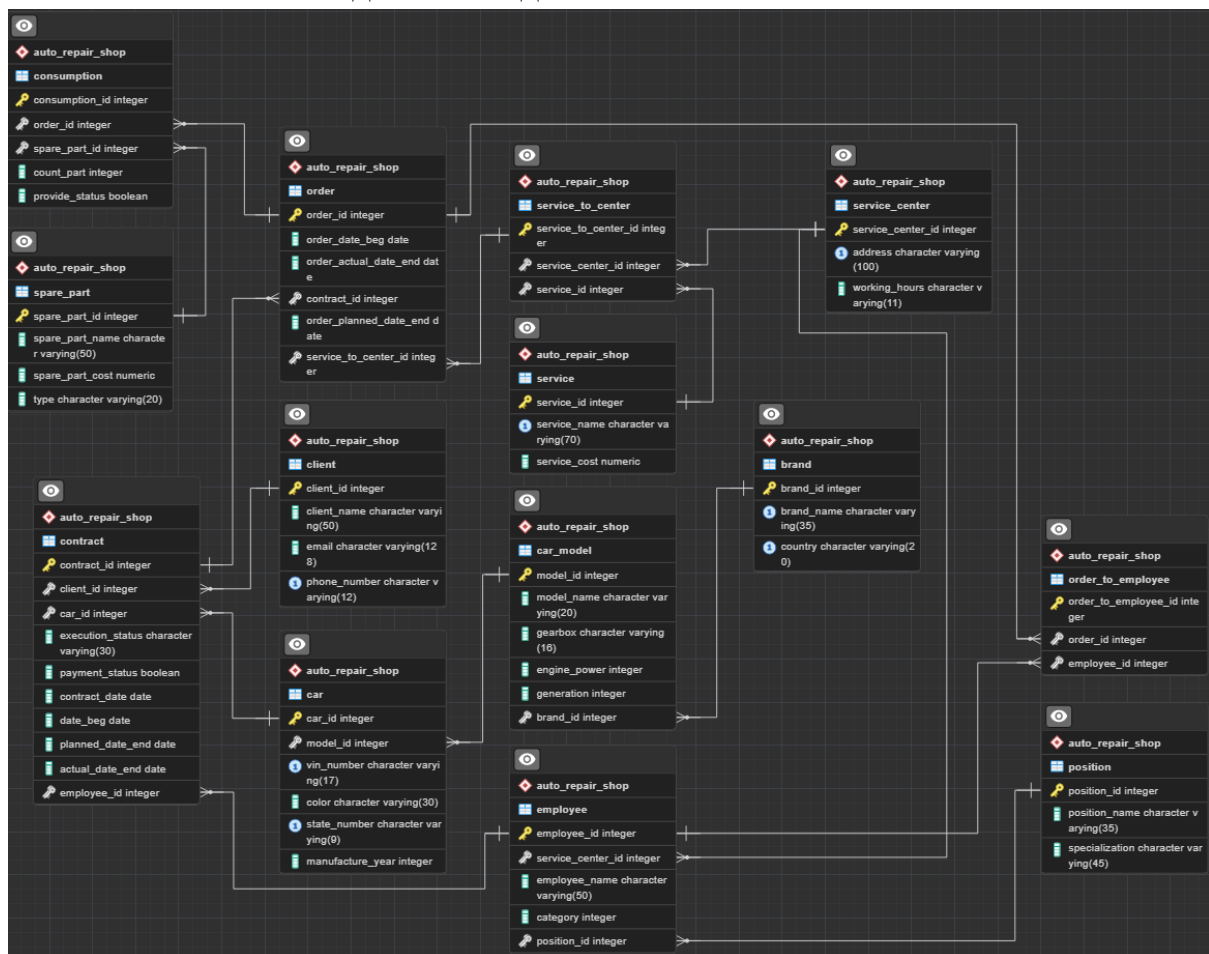


Рисунок 1 – ERD базы данных 1

Предустановки

```
SET search_path = "auto_repair_shop";
```

Запросы на выборку

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

Запрос:

```
WITH employee AS (  
    SELECT employee_name, COUNT(order_id) order_count  
    FROM employee e  
        JOIN order_to_employee USING(employee_id)  
        JOIN "order" USING(order_id)  
        JOIN contract USING(contract_id)  
        JOIN car USING(car_id)  
        JOIN car_model USING(model_id)  
        JOIN brand USING(brand_id)  
    WHERE brand_name = 'Toyota'  
    GROUP BY e.employee_id  
)  
  
SELECT employee_name, order_count  
FROM employee  
WHERE order_count = (  
    SELECT MAX(order_count)  
    FROM employee  
);
```

Вывод данных:

#	employee_name	order_count
1	Демьянов Владислав Александрович	13

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

Изменения: Так как на один заказ приходится несколько работников, нету клиентов подходящих под подобные условия. Я изменил логику запроса: Определить механиков и их самого популярного клиента (если популярных клиентов несколько, выведите всех), необходимо вывести имя механика, имя клиента и количество заказов клиента в которых участвовал механик.

Запрос:

```
WITH t AS (
  SELECT employee_name, client_name, COUNT(order_id) AS order_count
  FROM client
    JOIN contract USING(client_id)
    JOIN "order" USING(contract_id)
    JOIN order_to_employee ote USING(order_id)
    JOIN employee e ON e.employee_id = ote.employee_id
  GROUP BY e.employee_id, client_id
),
rnked AS (
  SELECT
    employee_name,
    client_name,
    order_count,
    ROW_NUMBER() OVER (PARTITION BY employee_name ORDER BY order_count DESC) AS
row_num
  FROM t
)

SELECT employee_name, client_name, order_count
FROM rnked
WHERE row_num = 1
ORDER BY employee_name, order_count DESC, client_name;
```

Вывод данных:

#	employee_name	client_name	order_count
1	Абрамов Макар Тимофеевич	Леонов Владимир Маркович	2
2	Акимова Диана Михайловна	Титов Иван Максимович	2
3	Александров Руслан Александрович	Новиков Александр Владимирович	3
4	Алешин Павел Александрович	Румянцев Вячеслав Маркович	2
5	Андреева Любовь Артуровна	Щербаков Дмитрий Даниилович	3
...
68	Тихомиров Фёдор Ярославович	Коновалов Матвей Филиппович	2
69	Фролов Евгений Ильич	Исаев Дмитрий Матвеевич	3
70	Фролова Ульяна Дмитриевна	Новикова Екатерина Игоревна	2
71	Юдина Дарья Лукинична	Иванова Валерия Сергеевна	2

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

Изменения: Так как количество просроченных заказов довольно много (2703 записи), я добавил условие: задержка более чем на месяц (30 дней).

Запрос:

```
WITH delays AS (  
    SELECT employee_name,  
           order_id,  
           (order_actual_date_end - order_planned_date_end) AS delay  
    FROM auto_repair_shop.employee  
    JOIN auto_repair_shop.order_to_employee USING(employee_id)  
    JOIN auto_repair_shop."order" USING(order_id)  
    WHERE order_actual_date_end IS NOT NULL  
          AND order_actual_date_end > order_planned_date_end  
)  
  
SELECT employee_name, order_id, delay  
FROM delays  
WHERE delay > 30  
ORDER BY 3 DESC, 1, 2
```

Вывод данных:

#	employee_name	order_id	delay
1	Волошина Василиса Степановна	57	31
2	Полякова Виктория Андреевна	57	31
3	Синицын Василий Владимирович	57	31
4	Панкратова Анастасия Денисовна	157	37
5	Кузьмина Вероника Артемьевна	157	37
53	Павлов Александр Даниилович	3363	47
...
54	Еремина Дарья Артемьевна	3363	47
55	Смирнов Дамир Артёмович	3363	47
56	Дмитриева Кира Платоновна	3698	32
57	Баранова Екатерина Саввична	3893	33

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

Изменения: Так как все данные были записаны зарегистрированы за период с 2020 по 2022, требуемый запрос не релевантен. Кроме того каждый работник имеет свою специализацию и он не способен выполнить все виды работ. Я изменил запрос следующим образом: Вывести данные специалистов, который выполнили больше всего различных видов работ, за апрель месяц 2022 года.

Запрос:

```
WITH t AS (  
  SELECT employee_id, COUNT(DISTINCT service_id) service_count  
  FROM auto_repair_shop.order_to_employee  
    JOIN auto_repair_shop."order" USING(order_id)  
    JOIN auto_repair_shop.service_to_center USING(service_to_center_id)  
  WHERE order_actual_date_end IS NOT NULL  
    AND order_actual_date_end BETWEEN '2022-04-01' AND '2022-04-30'  
  GROUP BY employee_id  
)  
  
SELECT  
  employee_name,  
  position_name,  
  specialization,  
  "category",  
  service_count  
FROM t  
  JOIN auto_repair_shop.employee USING(employee_id)  
  JOIN auto_repair_shop.position USING(position_id)  
WHERE service_count = (  
  SELECT MAX(service_count)  
  FROM t  
)  
ORDER BY employee_name, category DESC
```

Вывод данных:

#	employee_name	position_name	specialization	category	service_count
1	Абрамов Макар Тимофеевич	Автослесарь	Легковые автомобили	8	8
2	Зиновьев Павел Кириллович	Арматурщик	Разборщик салона	5	8

5. Сколько заработал каждый мастер за прошедший месяц?

Изменения: Так как все данные были записаны зарегистрированы за период с 2020 по 2022, я отобрал за апрель месяц 2022 года

Запрос:

```
WITH order_employee_counts AS (  
  SELECT  
    employee_id,  
    employee_name,  
    order_id,  
    service_cost,  
    COUNT(employee_id) OVER (PARTITION BY order_id) employees_count  
  FROM "order"  
  JOIN order_to_employee USING(order_id)  
  JOIN employee USING(employee_id)  
  JOIN service_to_center USING(service_to_center_id)  
  JOIN service USING(service_id)  
  WHERE order_actual_date_end IS NOT NULL  
        AND order_actual_date_end BETWEEN '2022-04-01' AND '2022-04-30'  
)  
  
SELECT  
  employee_name,  
  COUNT(order_id) orders_taken,  
  ROUND(SUM(service_cost / employees_count)) salary  
FROM order_employee_counts  
GROUP BY employee_id, employee_name  
ORDER BY salary DESC, orders_taken DESC, employee_name;
```

Вывод данных:

#	employee_name	orders_taken	salary
1	Фролова Ульяна Дмитриевна	8	30708
2	Абрамов Макар Тимофеевич	8	29167
3	Семенов Илья Александрович	7	27667
4	Зиновьев Павел Кириллович	8	24625
5	Костин Фёдор Георгиевич	3	22125
...
64	Дмитриева Кира Платоновна	1	1333
65	Юдина Дарья Лукинична	1	1250
66	Семенова Анастасия Сергеевна	1	1000
67	Савельева Мирослава Тимофеевна	1	750

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

Изменения: Добавил условие для меньшей выборки: Количество обращений больше 10

Запрос:

```
WITH t AS (  
    SELECT client_id, count(contract_id) count_contracts  
    FROM auto_repair_shop.contract  
    GROUP BY client_id  
)  
  
SELECT client_name, email, phone_number, count_contracts  
FROM t  
    JOIN auto_repair_shop.client USING(client_id)  
WHERE count_contracts > 10  
ORDER BY count_contracts DESC, client_name;
```

Вывод данных:

#	client_name	email	phone_number	count_contracts
1	Аникина Лидия Львовна	tyramini@mail.ru	+79434961499	13
2	Григорьев Марк Васильевич	Hack_War@yandex.ru	+79281715485	13
3	Кузьмин Тимофей Александрович	GtRussian@msn.com	+79213411263	13
4	Кошелев Глеб Артёмович	ghosthack@gmail.com	+79371631453	12
5	Крюков Михаил Ильич	SushiShoqapik@hotmail.com	+79831413748	12
...
16	Попова Арина Артемьевна	hadessbeta-574897@yandex.ru	+74532858461	11
17	Смирнова София Павловна	CrazyChicken_4251@yandex.ru	+79399747188	11
18	Смирнова Таисия Максимовна	wait_gold@list.ru	+74475725226	11
19	Щербаков Дмитрий Даниилович	NULL	+79934943213	11

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

Изменения: Так как все данные были записаны зарегистрированы за период с 2020 по 2022, я отобрал за апрель месяц 2022 года

Запрос:

```
WITH t AS (  
    SELECT employee_name, SUM(penalty) penalty  
    FROM (  
        SELECT  
            employee_name,  
            0.05 * (order_actual_date_end - order_planned_date_end) penalty  
        FROM auto_repair_shop.employee  
        JOIN auto_repair_shop.order_to_employee USING(employee_id)  
        JOIN auto_repair_shop."order" USING(order_id)  
        WHERE order_actual_date_end IS NOT NULL  
        AND order_actual_date_end > order_planned_date_end  
        AND order_actual_date_end BETWEEN '2022-04-01' AND '2022-04-30'  
    ) t_1  
    GROUP BY 1  
)  
  
SELECT  
    employee_name,  
    CASE  
        WHEN penalty > 1 THEN 1  
        ELSE penalty  
    END penalty  
FROM t  
ORDER BY 2 DESC, 1;
```

Вывод данных:

#	employee_name	penalty
1	Абрамов Макар Тимофеевич	1
2	Александров Руслан Александрович	1
3	Волошина Василиса Степановна	1
4	Емельянов Иван Егорович	1
5	Казаков Олег Александрович	1
...
44	Васильев Тимур Георгиевич	0.10
45	Еремина Дарья Артемьевна	0.10
46	Кузьмина Вероника Артемьевна	0.10
47	Баранова Екатерина Саввична	0.05

Создание представлений

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

Запрос:

```
CREATE VIEW clients_view AS (  
  WITH t1 AS (  
    SELECT  
      ote.employee_id,  
      model_id,  
      COUNT(order_id) orders_count  
    FROM order_to_employee ote  
    JOIN "order" USING(order_id)  
    JOIN contract USING(contract_id)  
    JOIN car USING(car_id)  
    GROUP BY ote.employee_id, model_id  
  ),  
  t2 AS (  
    SELECT  
      employee_id,  
      model_id,  
      orders_count,  
      ROW_NUMBER() OVER (PARTITION BY employee_id ORDER BY orders_count DESC)  
row_number  
    FROM t1  
  )  
  
  SELECT  
    employee_name,  
    brand_name,  
    model_name  
  FROM t2  
    JOIN employee USING(employee_id)  
    JOIN car_model USING(model_id)  
    JOIN brand USING(brand_id)  
  WHERE row_number = 1  
  ORDER BY COALESCE(category, 0) DESC, 1  
);
```

Вывод:

```
SELECT * FROM clients_view;
```

#	employee_name	brand_name	model_name
1	Андреева Любовь Артуровна	Toyota	Prius
2	Казаков Олег Александрович	BMW	5-Series
3	Королев Степан Арсентьевич	Ford	Fusion
4	Костин Фёдор Георгиевич	Suzuki	Grand Vitara
5	Павлов Александр Даниилович	Citroen	Berlingo First
...
68	Романова Амина Семёновна	Ford	Explorer
69	Савельев Егор Эмильевич	Toyota	Highlander
70	Савельева Мирослава Тимофеевна	Hyundai	Tucson
71	Юдина Дарья Лукинична	Citroen	C2

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

Изменения: Так как все данные были записаны зарегистрированы за период с 2020 по 2022, я отобрал за последний год

Запрос:

```
CREATE VIEW manager_view AS (  
  WITH t AS (  
    SELECT  
      employee_name,  
      SUM(service_cost) salary,  
      COUNT(order_actual_date_end) -  
      SUM((order_actual_date_end <= order_planned_date_end)::INT) delays_count  
    FROM employee  
    JOIN order_to_employee USING (employee_id)  
    JOIN "order" USING (order_id)  
    JOIN service_to_center USING (service_to_center_id)  
    JOIN service USING (service_id)  
    WHERE order_actual_date_end IS NOT NULL  
      AND order_actual_date_end >= NOW() - INTERVAL '1 year'  
    GROUP BY employee_name  
  ),  
  avg_delays AS (  
    SELECT AVG(delays_count) average_delays  
    FROM t  
  )  
  
  SELECT employee_name,  
    salary,  
    salary * (  
      (delays_count < (SELECT average_delays FROM avg_delays))::INT / 10.0  
    ) bonus  
  FROM t  
  ORDER BY salary DESC, bonus DESC, employee_name  
)
```

Вывод:

```
SELECT * FROM manager_view;
```

#	employee_name	salary	bonus
1	Беляева Виктория Львовна	201500	0
2	Романова Василиса Максимовна	198000	0
3	Афанасьев Денис Андреевич	194500	0
4	Зиновьев Павел Кириллович	184500	0
5	Рябов Никита Александрович	181250	0
...
68	Алешин Павел Александрович	96750	9675
69	Никольский Максим Николаевич	91500	9150
70	Некрасов Александр Артёмович	88750	8875
71	Кузьмин Даниил Степанович	71500	7150

Запросы на модификацию данных

1. Моделирование сценария, в котором новый клиент привозит свой автомобиль на обслуживание. Автомобиль никогда раньше не был в сервисном центре, и клиент еще не зарегистрирован в базе данных.

Запрос:

```
BEGIN;

WITH new_brand AS (
  INSERT INTO brand(brand_name, country)
  VALUES ('Acura', 'Япония')
  RETURNING brand_id
),
new_model AS (
  INSERT INTO car_model(model_name, gearbox, engine_power, generation, brand_id)
  SELECT 'MDX', 'автоматическая', 300, 2, brand_id
  FROM new_brand
  RETURNING model_id
),
new_car AS (
  INSERT INTO car(model_id, vin_number, color, state_number, manufacture_year)
  SELECT model_id, 'OIPWFTHBFMH12H9X', 'Серебристый', 'X101X0124', 2008
  FROM new_model
  RETURNING car_id
),
new_client AS (
  INSERT INTO client(client_name, phone_number)
  VALUES ('Иванов Роман Сергеевич', '+79239902192')
  RETURNING client_id
),
new_contract AS (
  INSERT INTO contract(client_id, car_id, execution_status, payment_status)
  SELECT client_id, car_id, 'Ожидание оплаты', FALSE
  FROM new_client, new_car
  RETURNING *
)
SELECT * FROM new_contract;

COMMIT;
```

Вывод:

#	contract_id	client_id	car_id	execution_status	payment_status	contract_date	date_beg	planned_date_end	actual_date_end	employee_id
1	5251	1001	1000	Ожидание оплаты	FALSE					

2. Смоделируем ситуацию при которой нам нужно обновить статус тех контрактов, у которых были выполнены все заказы

Запрос:

```
WITH updated_contracts AS (  
  UPDATE contract  
  SET execution_status = 'Ожидание подтверждения клиента'  
  WHERE contract_id IN (  
    SELECT contract_id  
    FROM contract  
    JOIN "order" USING(contract_id)  
    WHERE execution_status = 'Выполнение'  
    GROUP BY contract_id  
    HAVING COUNT(order_id) = COUNT(order_actual_date_end)  
  )  
  RETURNING *  
)  
SELECT COUNT(contract_id) FROM updated_contracts;
```

Вывод:

#	count
1	412

3. Смоделируем ситуации при которой из-за технических ошибок были зарегистрированы машины которые не упоминались ни в одном из контрактов. Необходимо удалить ненужные машины.

Запрос:

```
WITH deleted_cars AS (  
  DELETE FROM car  
  WHERE car_id IN (  
    SELECT car_id  
    FROM car  
    LEFT JOIN contract USING(car_id)  
    WHERE contract_id IS NULL  
  )  
  RETURNING *  
)  
SELECT * FROM deleted_cars;
```

Вывод:

#	car_id	model_id	vin_number	color	state_number	manufacture_year
1	518	623	IGP4E4O97QFHRJLNP	Серо-зелёный	M354AK138	2023
2	283	21	7NBLZ8OWDK0OWQH0H	Средне-стально й	K203XX66	2003
3	499	523	HRWFHPCA5L89BFWV1	Серо-чёрный	C491KT07	1999
4	314	533	SCLE8XL5AUPEPJ4A6Z	Серебристо-зелёно -голубой	K788AC32	1993
5	826	110	AL2Y1ZX81CHEH55BV	Ярко-синий	H734EO78	1994
6	215	409	H3NVP7EN2LCGWB3GR	Тёмно-зелёный	K964OH196	1993
7	659	225	RESU9DNOVFGSJ280Z	Тёмно-коричневый	O179CY75	2007
8	137	409	WYP92L88V6GY0T8QD	Тёмно-сине-зелён ый	O773CY76	2010
9	952	611	CQXZDW9JPMSJV8KJ6	Серебристо-серо-г рафитовый	T048PC45	2000
10	339	559	OFG9O7Z080ZAZTAZA	Серебристо-бежев ый	O937AA197	2001

Сравнение запросов с использованием индексирования

1. Сравнение простого индекса

Запрос:

```
EXPLAIN ANALYZE SELECT * FROM client WHERE client_name = 'Иванов Роман Сергеевич';
```

Создание индекса:

```
CREATE INDEX idx_client_client_name ON client(client_name);
```

Время выполнения без индекса:

```
Seq Scan on client (cost=0.00..27.50 rows=1 width=89) (actual time=0.170..0.170 rows=1 loops=1)
  Filter: ((client_name)::text = 'Иванов Роман Сергеевич'::text)
  Rows Removed by Filter: 1000
Planning Time: 0.108 ms
Execution Time: 0.186 ms
```

С индексом:

```
Index Scan using idx_client_client_name on client (cost=0.28..8.29 rows=1 width=89)
(actual time=0.019..0.020 rows=1 loops=1)
  Index Cond: ((client_name)::text = 'Иванов Роман Сергеевич'::text)
Planning Time: 0.056 ms
Execution Time: 0.030 ms
```

2. Сравнение составного индекса

Запрос:

```
EXPLAIN ANALYZE
SELECT *
FROM contract
WHERE execution_status = 'Ожидание оплаты'
      AND payment_status = TRUE;
```

Создание индекса:

```
CREATE INDEX idx_contract_execution_payment ON contract(execution_status,
payment_status);
```

Время выполнения без индекса:

```
Seq Scan on contract (cost=0.00..138.55 rows=1126 width=59) (actual time=0.821..0.821
rows=0 loops=1)
  Filter: (payment_status AND ((execution_status)::text = 'Ожидание оплаты'::text))
  Rows Removed by Filter: 5251
Planning Time: 0.085 ms
Execution Time: 0.835 ms
```

С индексом:

```
Bitmap Heap Scan on contract (cost=18.58..96.15 rows=1005 width=59) (actual
time=0.015..0.016 rows=0 loops=1)
  Recheck Cond: ((execution_status)::text = 'Ожидание оплаты'::text)
  Filter: payment_status
  -> Bitmap Index Scan on idx_contract_execution_payment (cost=0.00..18.33 rows=1005
width=0) (actual time=0.015..0.015 rows=0 loops=1)
        Index Cond: (((execution_status)::text = 'Ожидание оплаты'::text) AND
(payment_status = true))
Planning Time: 0.082 ms
Execution Time: 0.033 ms
```

Заключение:

В ходе выполнения данной лабораторной работе были реализованы запросы на выборку данных и представления к базе данных на PostgreSQL согласно индивидуальному заданию. Более того были смоделированы различные штатные ситуации и имплементированы 3 запроса на модификацию данных. Также был проведен анализ графического представления всех запросов. Созданы простые и составные индексы для различных запросов и проанализировано их время выполнения с использованием индексов.