Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики

Лабораторная работа№ 2
«Запросы на выборку данных к БД PostgreSQL. представления в PostrgreSQL»

Выполнил: Галиновский Роман Андреевич
Группа: К3240
Преподаватель: Говорова Марина Михайловна

Санкт-Петербург
2022

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

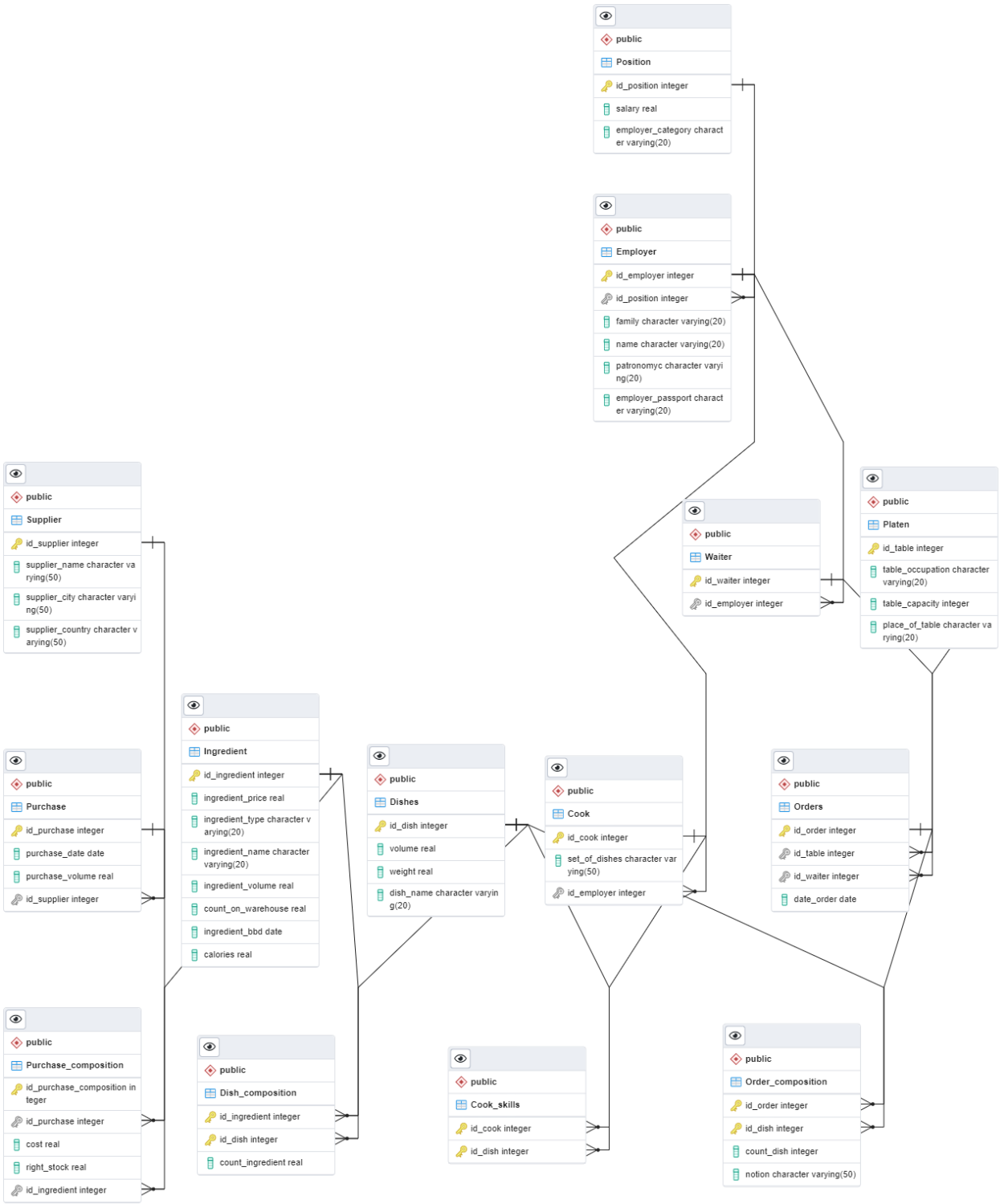
Оборудование: компьютерный класс.

Программное обеспечение: СУБД PostgreSQL 1X, pgAdmin 4.

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 просмотреть 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.

Схема базы данных:



Выполнение:

1. Запросы к базе данных:

1.1. Вывести данные официанта, принявшего максимальное число заказов:

```
SELECT public."Waiter".id_waiter,
public."Employer".family,
public."Employer".name,
public."Employer".patronomyc,
public."Employer".employer_passport
FROM public."Waiter", public."Employer", public."Orders"
WHERE public."Waiter".id_employer = public."Employer".id_employer
AND public."Waiter".id_waiter = public."Orders".id_waiter
GROUP BY "Waiter".id_waiter,
public."Employer".family,
public."Employer".name,
public."Employer".patronomyc,
public."Employer".employer_passport
HAVING COUNT(public."Orders".id_waiter) = (select count(id_waiter) as name
                                           from public."Orders"
                                           group by id_waiter
                                           ORDER BY count(id_waiter) desc
                                           limit 1);
```

Результат:

	id_waiter integer	family character varying (20)	name character varying (20)	patronomyc character varying (20)	employer_passport character varying (20)
1	111132	Romanov	Mikhail	Yanovich	6666 133728

1.2. Вывести данные официанта, принявшего заказы на максимальную сумму:

```
SELECT public."Employer".id_employer,public."Employer".family,public."Employer".name,public."Employer".patronomyc,
public."Employer".employer_passport,
SUM(public."Ingredient".ingredient_price * public."Order_composition".count_dish * public."Dish_composition".count_ingredient) AS price
FROM public."Employer", public."Waiter", public."Ingredient",public."Order_composition",
public."Dish_composition",public."Dishes",public."Orders"
WHERE public."Ingredient".id_ingredient = public."Dish_composition".id_ingredient
AND public."Dish_composition".id_dish = public."Dishes".id_dish
AND public."Orders".id_order = public."Order_composition".id_order
AND public."Orders".id_waiter = public."Waiter".id_waiter
AND public."Waiter".id_employer = public."Employer".id_employer
GROUP BY public."Employer".id_employer, public."Employer".family,
public."Employer".name,public."Employer".patronomyc,public."Employer".employer_passport
HAVING SUM(public."Ingredient".ingredient_price) = (SELECT SUM(public."Ingredient".ingredient_price)
                                                    FROM public."Employer",
                                                    public."Ingredient",
                                                    public."Order_composition",
                                                    public."Waiter",
                                                    public."Dish_composition",
                                                    public."Orders",
                                                    public."Dishes"
                                                    WHERE public."Ingredient".id_ingredient = public."Dish_composition".id_ingredient
                                                     AND public."Dish_composition".id_dish = public."Dishes".id_dish
                                                     AND public."Orders".id_order = public."Order_composition".id_order
                                                     AND public."Orders".id_waiter = public."Waiter".id_waiter
                                                     AND public."Waiter".id_employer = public."Employer".id_employer
                                                    GROUP BY public."Employer".id_employer
                                                    ORDER BY SUM(public."Ingredient".ingredient_price) desc
                                                    LIMIT 1)
```

Результат:

	id_employer [PK] integer	family character varying (20)	name character varying (20)	patronomyc character varying (20)	employer_passport character varying (20)	price double precision
1	111127	Romanov	Mikhail	Yanovich	6666 133728	204600

1.3. Рассчитать премию каждого официанта за последние 10 дней (5% от стоимости каждого заказа:

```
SELECT public."Employer".id_employer,
public."Employer".family,
public."Employer".name,
public."Employer".patronomyc,
public."Employer".employer_passport,
SUM(public."Ingredient".ingredient_price *
    public."Order_composition".count_dish *
    public."Dish_composition".count_ingredient * 0.05) AS price
FROM public."Employer",
public."Waiter",
public."Ingredient",
public."Order_composition",
public."Orders",
public."Dish_composition",
public."Dishes"
WHERE public."Ingredient".id_ingredient = public."Dish_composition".id_ingredient
AND public."Dish_composition".id_dish = public."Dishes".id_dish
AND public."Dishes".id_dish = public."Order_composition".id_dish
AND public."Order_composition".id_order = public."Orders".id_order
AND public."Orders".id_waiter = public."Waiter".id_waiter
AND public."Waiter".id_employer = public."Employer".id_employer
AND public."Orders".date_order BETWEEN ((CURRENT_DATE)-10) AND (CURRENT_DATE)
GROUP BY public."Employer".id_employer,
public."Employer".family,
public."Employer".name,
public."Employer".patronomyc,
public."Employer".employer_passport
```

Результат:

	id_employer [PK] integer	family character varying (20)	name character varying (20)	patronomyc character varying (20)	employer_passport character varying (20)	price double precision
1	111126	Bezrukova	Kseniya	Sergeevna	11 124566	88.5
2	111127	Romanov	Mikhail	Yanovich	6666 133728	145.25
3	111128	Roseman	Eugeny	Yanovich	6636 135628	112.5

1.4. Подсчитать, сколько ингредиентов содержит каждое блюдо:

```
SELECT public."Dishes".dish_name, COUNT(public."Dish_composition".count_ingredient) AS count_ingredient
FROM public."Dishes",
public."Dish_composition"
WHERE public."Dishes".id_dish = public."Dish_composition".id_dish
GROUP BY public."Dishes".id_dish
```

Результат:

	dish_name character varying (20)	count_ingredient bigint
1	vkusno doner	3
2	obed krutoi	2
3	oaaoaoa mmm	3

1.5. Вывести название блюда, содержащее максимальное число ингредиентов:

```
SELECT public."Dishes".id_dish,
public."Dishes".dish_name,
COUNT(public."Dish_composition".count_ingredient)
FROM public."Dishes"
INNER JOIN public."Dish_composition"
ON public."Dishes".id_dish = public."Dish_composition".id_dish
GROUP BY public."Dishes".id_dish, public."Dishes".dish_name
HAVING COUNT(public."Dish_composition".count_ingredient) = (SELECT COUNT(public."Dish_composition".count_ingredient)
                                                           FROM public."Dish_composition"
                                                           GROUP BY public."Dish_composition".id_dish
                                                           ORDER BY COUNT(public."Dish_composition".count_ingredient)
                                                           DESC LIMIT 1)
```

Результат:

	id_dish [PK] integer	dish_name character varying (20)	count bigint
1	111177	pizza navernoe	4

1.6. Какой повар может приготовить максимальное число видов блюд?

```
SELECT public."Employer".family,
public."Employer".name,
public."Employer".patronomyc,
SUM(1 + LENGTH(public."Cook".set_of_dishes) - LENGTH(REPLACE(public."Cook".set_of_dishes, '/','')))
FROM public."Employer", public."Cook"
WHERE public."Employer".id_employer = public."Cook".id_employer
GROUP BY public."Employer".family,
public."Employer".name,
public."Employer".patronomyc
HAVING SUM(1 + LENGTH(public."Cook".set_of_dishes) - LENGTH(REPLACE(public."Cook".set_of_dishes, '/',''))) =
          (SELECT SUM(1 + LENGTH(public."Cook".set_of_dishes) - LENGTH(REPLACE(public."Cook".set_of_dishes, '/','')))
          FROM public."Employer", public."Cook"
          WHERE public."Employer".id_employer = public."Cook".id_employer
          GROUP BY public."Employer".family,
          public."Employer".name,
          public."Employer".patronomyc
          ORDER BY SUM DESC LIMIT 1)
```

Результат:

	family character varying (20)	name character varying (20)	patronomyc character varying (20)	count bigint
1	Dobroslave	Shiryaev	Nurzupaevicg	4

1.7. Сколько закреплено столов за каждым из официантов?

```
SELECT public."Employer".family,
public."Employer".name,
public."Employer".patronomyc,
COUNT(public."Orders".id_waiter) as amount
FROM public."Employer", public."Platen",  public."Orders", public."Waiter"
WHERE public."Employer".id_employer = public."Waiter".id_employer
AND public."Waiter".id_waiter = public."Orders".id_waiter
AND public."Orders".id_table = public."Platen".id_table
GROUP BY public."Employer".family,
public."Employer".name,
public."Employer".patronomyc
```

Результат:

	family character varying (20)	name character varying (20)	patronomyc character varying (20)	amount bigint
1	Roseman	Eugeny	Yanovich	3
2	Romanov	Mikhail	Yanovich	4
3	Bezrukova	Kseniya	Sergeevna	3

1.8. Какой из ингредиентов используется в максимальном количестве блюд?

```
SELECT public."Ingredient".ingredient_name, COUNT(public."Dish_composition".id_dish)
FROM public."Ingredient", public."Dish_composition"
WHERE public."Ingredient".id_ingredient = public."Dish_composition".id_ingredient
GROUP BY public."Ingredient".ingredient_name
HAVING COUNT(public."Dish_composition".id_dish) = ANY(SELECT COUNT(public."Dish_composition".id_dish)
                                                      FROM public."Ingredient", public."Dish_composition"
                                                      WHERE public."Ingredient".id_ingredient = public."Dish_composition".id_ingredient
                                                      GROUP BY public."Ingredient".ingredient_name
                                                      ORDER BY COUNT(public."Dish_composition".id_dish) desc LIMIT 1)
```

Результат:

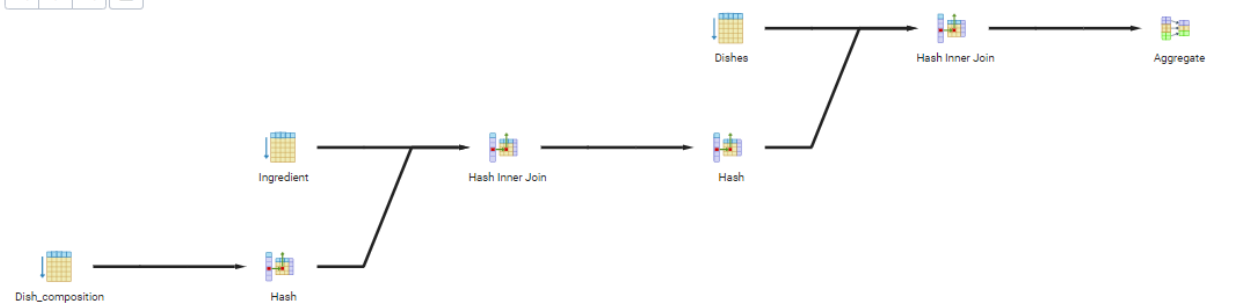
	ingredient_name character varying (20)	count bigint
1	eggs	5

2. *Представление:*

2.1. Для расчета стоимости ингредиентов для заданного блюда:

```
SELECT public."Dishes".dish_name, SUM(public."Ingredient".ingredient_price)
FROM public."Dishes", public."Ingredient", public."Dish_composition"
WHERE public."Ingredient".id_ingredient = public."Dish_composition".id_ingredient
AND public."Dish_composition".id_dish = public."Dishes".id_dish
GROUP BY public."Dishes".dish_name
```

Graphical Analysis Statistics

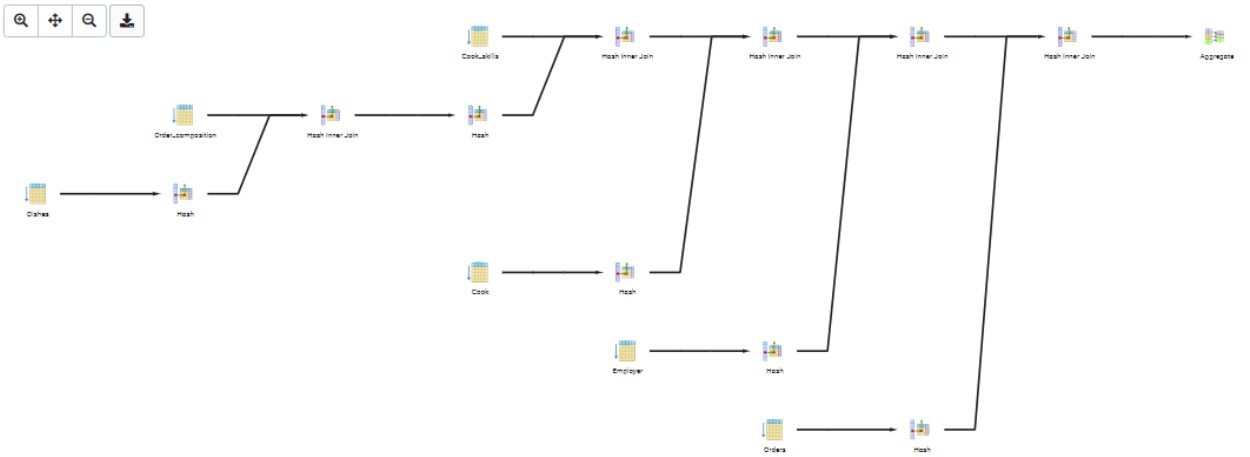


Результат:

	dish_name character varying (20)	sum real
1	vkusno doner	295
2	chapper	420
3	Borsh	215
4	pizza navernoe	230
5	oaaoaoa mmm	480
6	Porsh	450
7	Trap chappa	295
8	obed krutoi	145
9	obed ne krutoy	100
10	kakie-to sushi maybe	160

2.2 Для всех поваров количество приготовленных блюд по каждому блюду за определенную дату:

```
SELECT public."Orders".date_order,
public."Employer".family,
public."Employer".name,
public."Employer".patronomyc,
COUNT(public."Cook_skills".id_dish)
FROM public."Orders", public."Employer", public."Cook_skills",
public."Cook", public."Dishes",public."Order_composition"
WHERE public."Employer".id_employer = public."Cook".id_employer
AND public."Cook".id_cook = public."Cook_skills".id_cook
AND public."Cook_skills".id_dish = public."Dishes".id_dish
AND public."Dishes".id_dish = public."Order_composition".id_dish
AND public."Order_composition".id_order = public."Orders".id_order
GROUP BY public."Employer".family,
public."Employer".name,
public."Employer".patronomyc,
public."Orders".date_order
```



Результат:

	date_order date	family character varying (20)	name character varying (20)	patronomyc character varying (20)	count bigint
1	2022-06-05	Dobroslave	Shiryaev	Nurzupaevicg	2
2	2022-06-09	Ismail	Darzaev	Umarshapaevich	1
3	2022-06-08	Ismail	Darzaev	Umarshapaevich	1
4	2022-06-01	Maxim	Prihodko	Tatianovich	1
5	2022-06-01	Dobroslave	Shiryaev	Nurzupaevicg	1
6	2022-06-10	Maxim	Prihodko	Tatianovich	1
7	2022-06-04	Maxim	Prihodko	Tatianovich	2
8	2022-06-07	Ismail	Darzaev	Umarshapaevich	1
9	2022-06-07	Dobroslave	Shiryaev	Nurzupaevicg	1
10	2022-05-31	Dobroslave	Shiryaev	Nurzupaevicg	2
11	2022-06-02	Ismail	Darzaev	Umarshapaevich	1
12	2022-06-02	Dobroslave	Shiryaev	Nurzupaevicg	1
13	2022-06-08	Dobroslave	Shiryaev	Nurzupaevicg	1
14	2022-06-10	Ismail	Darzaev	Umarshapaevich	1
15	2022-06-03	Ismail	Darzaev	Umarshapaevich	2
16	2022-06-09	Maxim	Prihodko	Tatianovich	1

3. *Запросы на модификацию данных*

3.1 Добавить новое блюдо – INSERT

До:

	id_dish [PK] integer	volume real	weight real	dish_name character varying (20)
1	111171	150.5	420	Borsh
2	111172	90.3	115.2	Porsh
3	111173	40.6	50	Trap chappa
4	111174	45.2	55.7	chapper
5	111175	75.5	90.1	oaaoaoa mmm
6	111176	80	100.2	vkusno doner
7	111177	85.3	125.2	pizza navernoe
8	111178	160.3	330.7	kakie-to sushi maybe
9	111179	36.6	78.8	obed krutoi
10	111800	455.5	774.7	obed ne krutoy

После:

```
INSERT INTO public."Dishes"(
    id_dish, volume, weight, dish_name)
    VALUES (111181, 10, 10, 'DAYTE BALLI PLEASE');
```

	id_dish [PK] integer	volume real	weight real	dish_name character varying (20)
1	111171	150.5	420	Borsh
2	111172	90.3	115.2	Porsh
3	111173	40.6	50	Trap chappa
4	111174	45.2	55.7	chapper
5	111175	75.5	90.1	oaaoaoa mmm
6	111176	80	100.2	vkusno doner
7	111177	85.3	125.2	pizza navernoe
8	111178	160.3	330.7	kakie-to sushi maybe
9	111179	36.6	78.8	obed krutoi
10	111800	455.5	774.7	obed ne krutoy
11	111181	10	10	DAYTE BALLI PLEASE

3.2 UPDATE - Повысить зарплату на 2000 всем работникам, у которых она меньше 30000

До:

	id_position [PK] integer	salary real	employer_category character varying (20)
1	111111	30000	manager
2	111112	25000	waiter
3	111113	60000	cook
4	111114	75000	director
5	111115	15000	cleaner
6	111116	20000	barmen

После:

```
UPDATE public."Position"
    SET salary=salary+2000
    WHERE salary IN(SELECT salary FROM public."Position"
                    WHERE salary < 30000);
```

	id_position [PK] integer	salary real	employer_category character varying (20)
1	111111	30000	manager
2	111113	60000	cook
3	111114	75000	director
4	111112	27000	waiter
5	111115	17000	cleaner
6	111116	22000	barmen

3.3 DELETE – Удалить просто так всех тех сотрудников, что зарабатывают ниже положенного(25000 руб)

До:

	id_employer [PK] integer	id_position integer	family character varying (20)	name character varying (20)	patronomyc character varying (20)	employer_passport character varying (20)
1	111121	111114	Nagiev	Dmitriy	Vladimirovich	6716 53355
2	111122	111111	Victoryia	Lazareva	Sergeevna	6728 22837
3	111123	111113	Dobroslave	Shiryaev	Nurzupaevicg	6111 11111
4	111124	111113	Ismail	Darzaev	Umarshapaevich	6111 195851
5	111125	111113	Maxim	Prihodko	Tatianovich	6111 133337
6	111126	111112	Bezrukova	Kseniya	Sergeevna	11 124566
7	111127	111112	Romanov	Mikhail	Yanovich	6666 133728
8	111128	111112	Roseman	Eugeny	Yanovich	6636 135628
9	111129	111116	Kadirov	Ramzan	Ahmatovich	666 666666
10	111300	111115	Sosamov	Tigran	Karimovich	1111 22848

После:

```
DELETE FROM public."Employer"
    WHERE id_position IN(SELECT public."Employer".id_position FROM public."Position", public."Employer"
                         WHERE public."Employer".id_position = public."Position".id_position
                         AND public."Position".salary < 25000);
```

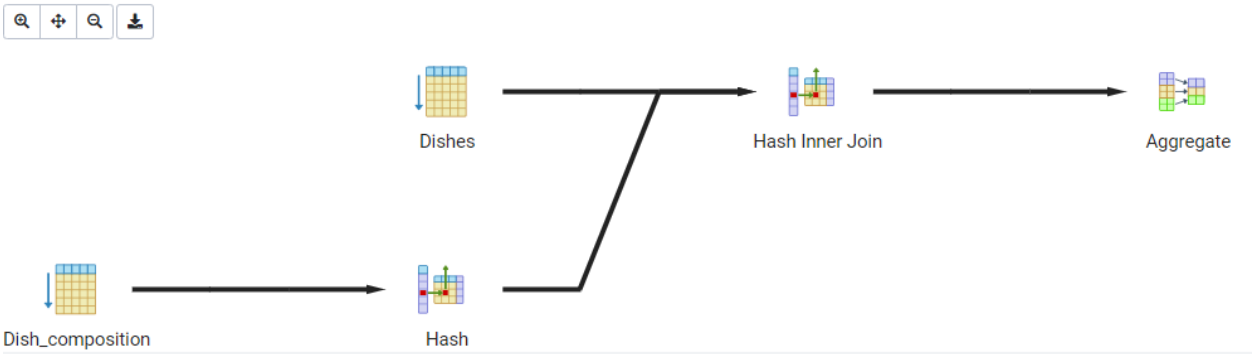
	id_employer [PK] integer	id_position integer	family character varying (20)	name character varying (20)	patronomyc character varying (20)	employer_passport character varying (20)
1	111121	111114	Nagiev	Dmitriy	Vladimirovich	6716 53355
2	111122	111111	Victoryia	Lazareva	Sergeevna	6728 22837
3	111123	111113	Dobroslave	Shiryaev	Nurzupaevicg	6111 11111
4	111124	111113	Ismail	Darzaev	Umarshapaevich	6111 195851
5	111125	111113	Maxim	Prihodko	Tatianovich	6111 133337
6	111126	111112	Bezrukova	Kseniya	Sergeevna	11 124566
7	111127	111112	Romanov	Mikhail	Yanovich	6666 133728
8	111128	111112	Roseman	Eugeny	Yanovich	6636 135628

4. *Создание Индексов:*

4.1 Четвёртый запрос без индекса:

```
SELECT public."Dishes".dish_name AS "БЛЮДО",
SUM(public."Dish_composition".count_ingredient) AS "КОЛЛИЧЕСТВО ИНГРЕДИЕНТОВ"
FROM public."Dishes",
public."Dish_composition"
WHERE public."Dishes".id_dish = public."Dish_composition".id_dish
GROUP BY public."Dishes".id_dish
```

```
Successfully run. Total query runtime: 63 msec.
10 rows affected.
```



	БЛЮДО character varying (20)	КОЛЛИЧЕСТВО ИНГРЕДИЕНТОВ real
1	kakie-to sushi maybe	8
2	pizza navernoe	12
3	obed krutoi	5
4	oaaoaoa mmm	7
5	Trap chappa	5
6	Porsh	8
7	obed ne krutoy	3
8	Borsh	5
9	chapper	5
10	vkusno doner	7

С Индексом:
создание индекса:

```
CREATE INDEX meal ON public."Dishes"(dish_name)
```

Data Output | Explain | Messages | Notifications

```
CREATE INDEX

Query returned successfully in 70 msec.
```

запрос:

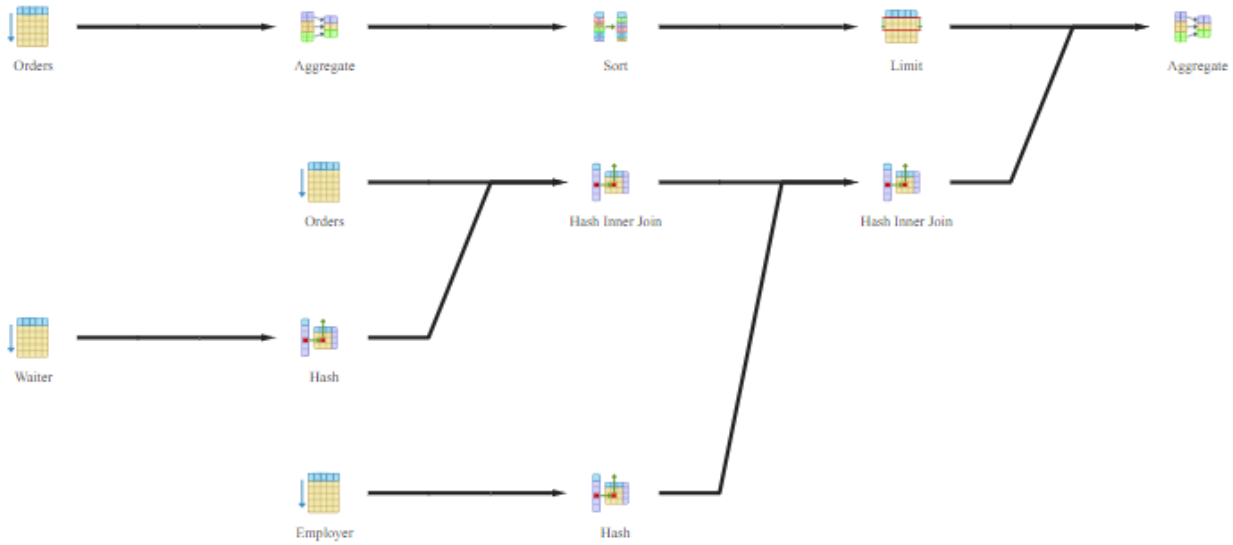
```
Successfully run. Total query runtime: 56 msec.
10 rows affected.
```

план выполнения запроса не изменился.

Первый запрос без индекса:

```
SELECT public."Waiter".id_waiter,
public."Employer".family,
public."Employer".name,
public."Employer".patronomyc,
public."Employer".employer_passport
FROM public."Waiter", public."Employer", public."Orders"
WHERE public."Waiter".id_employer = public."Employer".id_employer
AND public."Waiter".id_waiter = public."Orders".id_waiter
GROUP BY "Waiter".id_waiter,
public."Employer".family,
public."Employer".name,
public."Employer".patronomyc,
public."Employer".employer_passport
HAVING COUNT(public."Orders".id_waiter) = (select count(id_waiter) as name
                                           from public."Orders"
                                           group by id_waiter
                                           ORDER BY count(id_waiter) desc
                                           limit 1);
```

```
Successfully run. Total query runtime: 117 msec.
1 rows affected.
```



	id_waiter integer	family character varying (20)	name character varying (20)	patronomyc character varying (20)	employer_passport character varying (20)
1	111132	Romanov	Mikhail	Yanovich	6666 133728

Создание индекса:

```
CREATE UNIQUE INDEX unqindx ON
public."Waiter"(id_waiter, id_employer)
```

```
CREATE INDEX

Query returned successfully in 73 msec.
```

запрос:

```
Successfully run. Total query runtime: 46 msec.
1 rows affected.
```

4.2 Удаление индекса:

```
DROP INDEX

Query returned successfully in 50 msec.
```

Выводы:

В ходе данной лабораторной работы я овладел практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL. В рамках работы активно использовались подзапросы, в том числе для операций создания, редактирования и удаления данных. Также мной были освоены просты и составные индексы, которые я применил на практике.