

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

Отчет

по Лабораторной Работе № 2

по дисциплине «Проектирование и реализация баз данных»

Вариант 12

Автор: Мазеин Никита Олегович

Факультет: ФИКТ

Группа: K32402

Преподаватель: Говорова Марина Михайловна

itmo

Санкт-Петербург

2023

1. Цель работы:

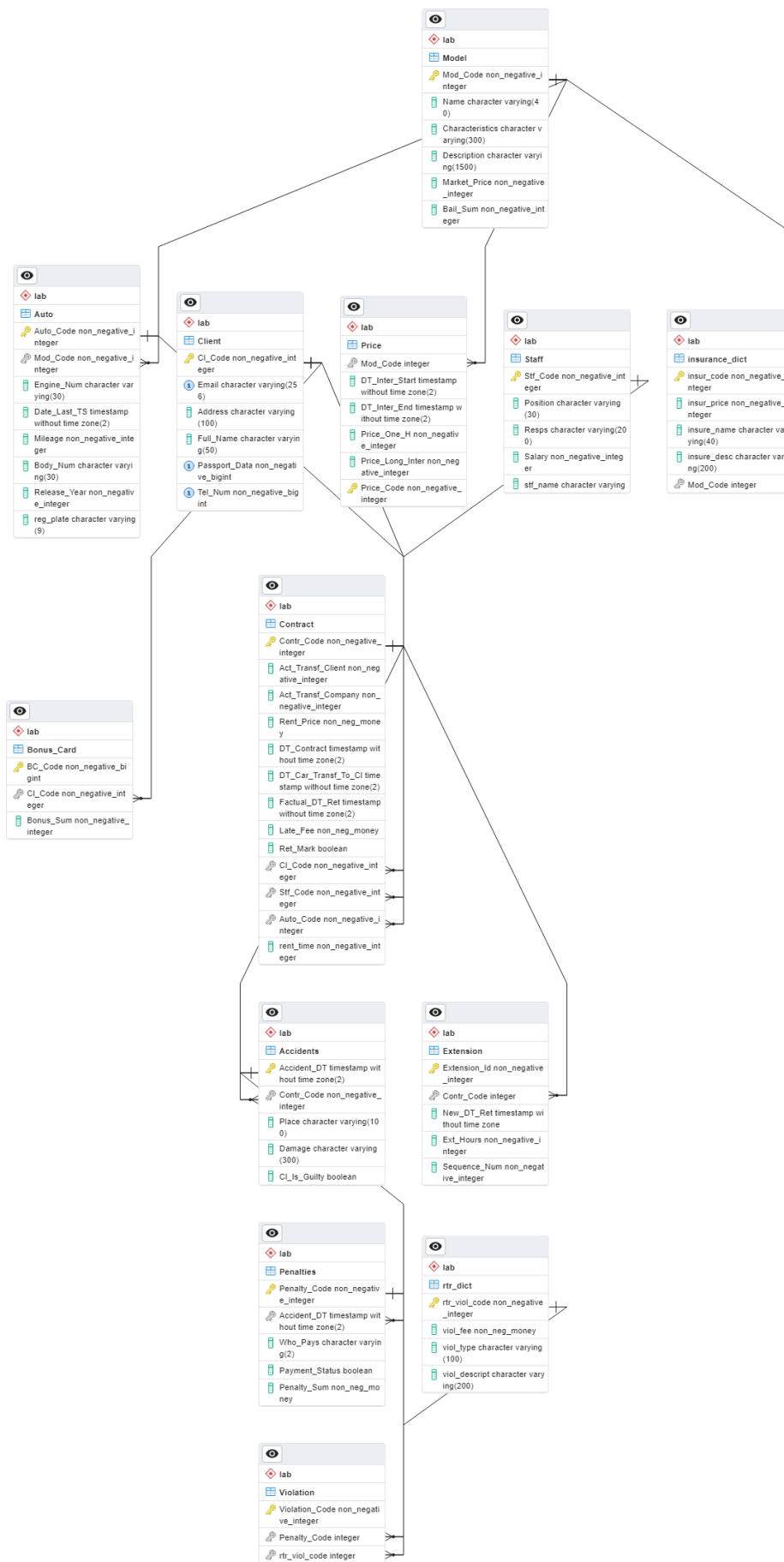
овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

2. Практическое задание:

БД «Прокат автомобилей»

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.

3. ERD Cxema:



4. Выполнение:

Задание 2. Создать запросы:

- Какой автомобиль находился в прокате максимальное количество часов?

Query		Query History	
1	SELECT	"Auto_Code",	rent_time
2	FROM	lab."Contract"	
3	ORDER BY	rent_time	DESC
4	LIMIT	1	

Data Output		Messages	Notifications
+	📄	📋	🗑️
+	📄	🗑️	📊
	Auto_Code	rent_time	
	integer	integer	
1	186549	72	

- Автомобили какой марки чаще всего брались в прокат?

Query		Query History	
1	SELECT	m."Name",	COUNT(*) AS rent_count
2	FROM	lab."Model" m	
3	JOIN	lab."Auto" USING	("Mod_Code")
4	JOIN	lab."Contract" USING	("Auto_Code")
5	GROUP BY	m."Mod_Code"	
6	ORDER BY	rent_count	DESC
7	LIMIT	1	

Data Output		Messages	Notifications
+	📄	📋	🗑️
+	📄	🗑️	📊
	Name	rent_count	
	character varying (40)	bigint	
1	BMW X5	7	

- Определить убытки от простоя автомобилей за вчерашний день.

```

1 SELECT SUM(p.current_price * h.hours_wo_rent) AS total_loss
2 FROM (
3     SELECT a."Auto_Code",
4           (24 - LEAST(COALESCE(hours_rented, 0), 24)) AS hours_wo_rent
5 FROM lab."Auto" a
6 LEFT JOIN (
7     SELECT "Auto_Code",
8           COALESCE(CAST(SUM(EXTRACT(EPOCH FROM ("Factual_DT_Ret" - "DT_Car_Transf_To_Cl")) / 3600 AS INTEGER), 0)
9 FROM lab."Contract"
10 WHERE DATE("DT_Car_Transf_To_Cl") = DATE(CURRENT_DATE - 3)
11 GROUP BY "Auto_Code"
12 ) AS c ON a."Auto_Code" = c."Auto_Code"
13 ORDER BY a."Auto_Code"
14 ) AS h
15 JOIN (
16     SELECT a."Auto_Code", p."Price_One_H" AS current_price
17 FROM lab."Auto" a
18 JOIN lab."Price" p ON a."Mod_Code" = p."Mod_Code"
19 WHERE p."DT_Inter_End" IS NULL
20 ORDER BY a."Auto_Code"
21 ) AS p ON h."Auto_Code" = p."Auto_Code";

```

Data Output Messages Notifications



	total_loss bigint
1	53504

- Вывести данные автомобиля, имеющего максимальный пробег.

```

1 SELECT "Auto_Code"
2 FROM lab."Auto"
3 WHERE "Mileage" = (
4     SELECT MAX("Mileage")
5     FROM lab."Auto");

```

Data Output Messages Notifications



	Auto_Code [PK] integer
1	539481

- Какой автомобиль суммарно находился в прокате дольше всех.

```

1 SELECT lab."Auto"."Auto_Code", SUM("rent_time")
2 FROM lab."Auto"
3 JOIN lab."Contract" ON lab."Auto"."Auto_Code" = lab."Contract"."Auto_Code"
4 GROUP BY lab."Auto"."Auto_Code"
5 HAVING SUM("rent_time") = (
6     SELECT MAX(total_rent_time)
7     FROM (
8         SELECT "Auto_Code", SUM("rent_time") AS total_rent_time
9         FROM lab."Contract"
10        GROUP BY "Auto_Code"
11     ) AS subquery
12 );
13

```

Data Output Messages Notifications



	Auto_Code [PK] integer	sum bigint
1	186549	168

- Определить, каким количеством автомобилей каждой марки и модели владеет компания.

```

1 SELECT COUNT (*)
2 FROM lab."Auto"
3 GROUP BY "Mod_Code"

```

Data Output Messages Notifications



	count bigint	
1	3	
2	3	
3	7	
4	6	
5	3	
6	4	
7	3	
8	3	

- Определить средний “возраст” автомобилей компании.

```

1 SELECT ROUND(AVG(EXTRACT(YEAR FROM CURRENT_DATE) - "Release_Year"), 3) AS avg_age
2 FROM lab."Auto";
3

```

Data Output Messages Notifications



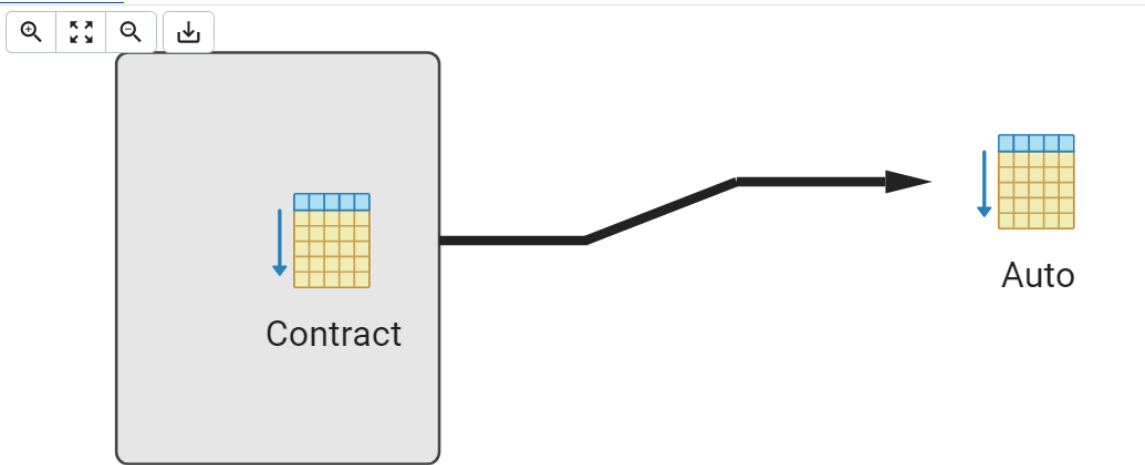
	avg_age numeric	
1	5.313	

Задание 3. Создать представление:

- Какой автомобиль ни разу не был в прокате?

```
1 SELECT "Auto_Code" FROM lab."Auto"
2 WHERE "Auto_Code" NOT IN
3 (SELECT "Auto_Code"
4 FROM lab."Contract")
```

	Auto_Code [PK] integer
1	205635
2	759321
3	324516
4	362718
5	237389
6	184729
7	837287
8	276172
9	367932
10	452145
11	872364
12	539481
13	287461
14	721930
15	972841

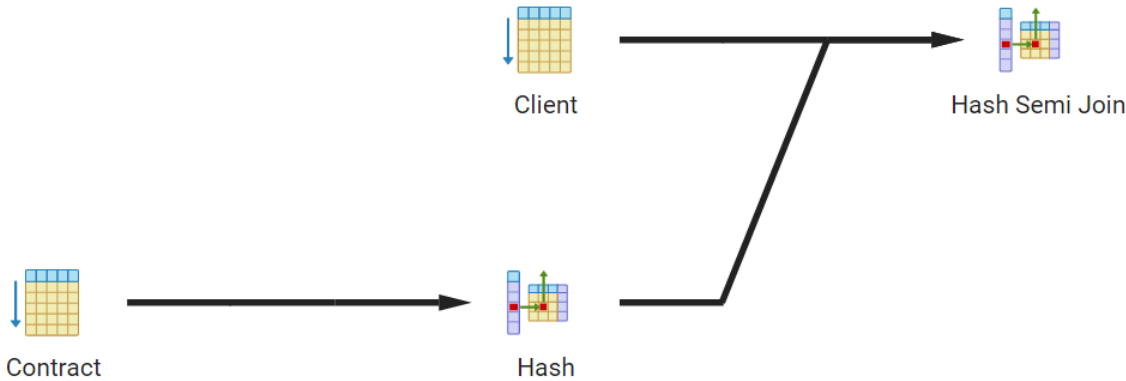


#	Node	Rows	
		Actual	Loops
1.	→ Seq Scan on Auto as Auto (rows=15 loops=1) Filter: (NOT (hashed SubPlan 1)) Rows Removed by Filter: 17	15	1
2.	→ Seq Scan on Contract as Contract (rows=33 loops=1)	33	1

- Вывести данные клиентов, не вернувших автомобиль вовремя.

```
1 SELECT * FROM lab."Client"
2 WHERE "Cl_Code" IN
3 (SELECT "Cl_Code"
4 FROM lab."Contract"
5 WHERE "Late_Fee" IS NOT NULL);
```

	Cl_Code [PK] integer	Email character varying (256)	Address character varying (100)	Full_Name character varying (50)	Passport_Data bigint	Tel_Num bigint
1	23456	jane.smith@yahoo.com	456 Elm St, Anytown, USA	Jane Smith	2345678901	71234567891
2	34567	bob.jones@hotmail.com	789 Maple Ave, Anytown, USA	Bob Jones	3456789012	71234567892
3	56789	jim.smith@gmail.com	456 Cedar Ave, Anytown, USA	Jim Smith	5678901234	71234567894
4	78901	mike.jones@hotmail.com	123 Elm St, Anytown, USA	Mike Jones	7890123456	71234567896
5	34568	amy.brown@hotmail.com	321 Maple Ave, Anytown, USA	Amy Brown	6666666666	71234567917
6	11234	peter.parker@hotmail.com	444 Main St, New York City	Peter Parker	6789678967	71234567971



#	Node	Rows		Loops
		Actual		
1.	→ Hash Semi Join (rows=6 loops=1) Hash Cond: (("Client"."Cl_Code")::integer = ("Contract"."Cl_Code")::integer)	6		1
2.	→ Seq Scan on Client as Client (rows=27 loops=1)	27		1
3.	→ Hash (rows=6 loops=1) Buckets: 1024 Batches: 1 Memory Usage: 9 kB	6		1
4.	→ Seq Scan on Contract as Contract (rows=6 loops=1) Filter: ("Late_Fee" IS NOT NULL) Rows Removed by Filter: 27	6		1

Задание ?. Запросы на модификацию данных

- INSERT

```
1 INSERT INTO lab."Staff" ("Stf_Code", "Position", "Resps", "Salary", "stf_name")
2 SELECT subquery.next_code, subquery."Position", subquery."Resps", subquery."Salary", 'Johnny Depth'
3 FROM (
4     SELECT MAX("Stf_Code") + 1 AS next_code, "Position", "Resps", "Salary"
5     FROM lab."Staff"
6     WHERE "Position" IN (
7         SELECT "Position"
8         FROM lab."Staff"
9         GROUP BY "Position"
10        HAVING COUNT(*) = 1
11    ) AND "Salary" = (
12        SELECT MIN("Salary")
13        FROM lab."Staff"
14        WHERE "Position" IN (
15            SELECT "Position"
16            FROM lab."Staff"
17            GROUP BY "Position"
18            HAVING COUNT(*) = 1
19        )
20    )
21    GROUP BY "Position", "Resps", "Salary"
22 ) AS subquery;
```

🔍

🔍

🔍

📄

```
graph LR
    S1[Staff] --> A1[Aggregate]
    S1 --> H1[Hash]
    S1 --> S2[Sort]
    A1 --> H1
    H1 --> HIJ[Hash Inner Join]
    S2 --> NLIJ[Nested Loop Inner Join]
    HIJ --> A2[Aggregate]
    NLIJ --> A2
    A2 --> A3[Aggregate]
    A3 --> SQS[SubQuery Scan]
    SQS --> I[Insert]
```

11	678902	Customer Serv Representative	Assists customers with inquiries and issues. Provides customer support.	35000	Johnny Depth
----	--------	------------------------------	---	-------	--------------

- UPDATE

```
1 UPDATE lab."Auto"
2 SET "Date_Last_TS" = CURRENT_DATE
3 WHERE EXTRACT(YEAR FROM "Date_Last_TS") <= EXTRACT(YEAR FROM CURRENT_DATE) - 5;
```

Data Output Messages Explain × Notifications

UPDATE 9

Query returned successfully in 107 msec.

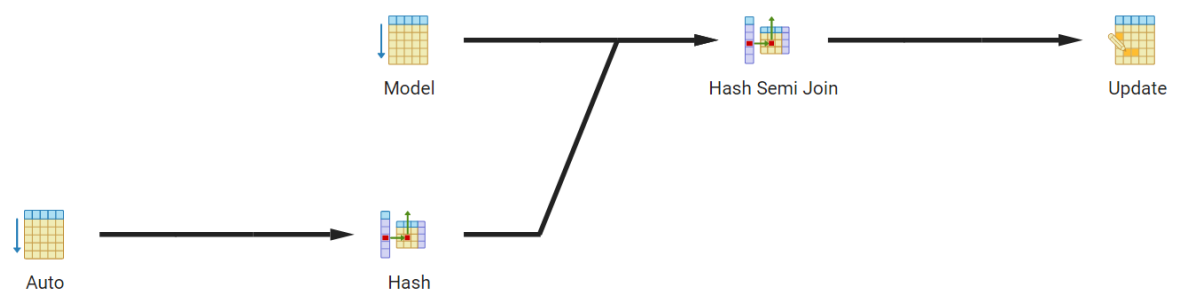


Query	Query History
1	UPDATE lab."Model"
2	SET "Bail_Sum" = "Bail_Sum" + 100
3	WHERE "Mod_Code" IN (
4	SELECT "Mod_Code"
5	FROM lab."Auto"
6	WHERE "Date_Last_TS" = CURRENT_DATE);

Data Output	Messages	Explain	×	Notifications
-------------	----------	---------	---	---------------

UPDATE 5

Query returned successfully in 77 msec.



- DELETE

Query Query History

1

DELETE FROM lab."Price"

2

WHERE "Mod_Code" IN (

3

SELECT "Mod_Code"

4

FROM lab."Auto"

5

WHERE "Auto_Code" NOT IN (

6

SELECT DISTINCT "Auto_Code"

7

FROM lab."Contract"

8

)

9

) AND "Price_One_H" <

10

(SELECT AVG("Price_One_H")

11

FROM lab."Price")

Data Output Messages Explain × Notifications

DELETE 601

Query returned successfully in 120 msec.



Задание ?. Создание индексов

Используем код из предыдущего задания:

```
1 SELECT SUM(p.current_price * h.hours_wo_rent) AS total_loss
2 FROM (
3 SELECT a."Auto_Code",
4 (24 - LEAST(COALESCE(hours_rented, 0), 24)) AS hours_wo_rent
5 FROM lab."Auto" a
6 LEFT JOIN (
7 SELECT "Auto_Code",
8 COALESCE(CAST(SUM(EXTRACT(EPOCH FROM ("Factual_DT_Ret" - "DT_Car_Transf_To_Cl")) / 3600 AS INTEGER), 0) AS hours_rented
9 FROM lab."Contract"
10 WHERE DATE("DT_Car_Transf_To_Cl") = DATE(CURRENT_DATE - 3)
11 GROUP BY "Auto_Code"
12 ) AS c ON a."Auto_Code" = c."Auto_Code"
13 ORDER BY a."Auto_Code"
14 ) AS h
15 JOIN (
16 SELECT a."Auto_Code", p."Price_One_H" AS current_price
17 FROM lab."Auto" a
18 JOIN lab."Price" p ON a."Mod_Code" = p."Mod_Code"
19 WHERE p."DT_Inter_End" IS NULL
20 ORDER BY a."Auto_Code"
21 ) AS p ON h."Auto_Code" = p."Auto_Code";
```

Data Output Messages Notifications

Successfully run. Total query runtime: 78 msec.
1 rows affected.

Без Индексации: время выполнения 78 мс

```
1 CREATE INDEX idx_auto_auto_code ON lab."Auto" ("Auto_Code");
2
3 CREATE INDEX idx_contract_auto_code ON lab."Contract" ("Auto_Code");
4 CREATE INDEX idx_contract_dt_car_transf_to_cl ON lab."Contract" ("DT_Car_Transf_To_Cl");
5 CREATE INDEX idx_contract_factual_dt_ret ON lab."Contract" ("Factual_DT_Ret");
6
7 CREATE INDEX idx_price_mod_code ON lab."Price" ("Mod_Code");
8 CREATE INDEX idx_price_dt_inter_end ON lab."Price" ("DT_Inter_End");
```

Query Query History

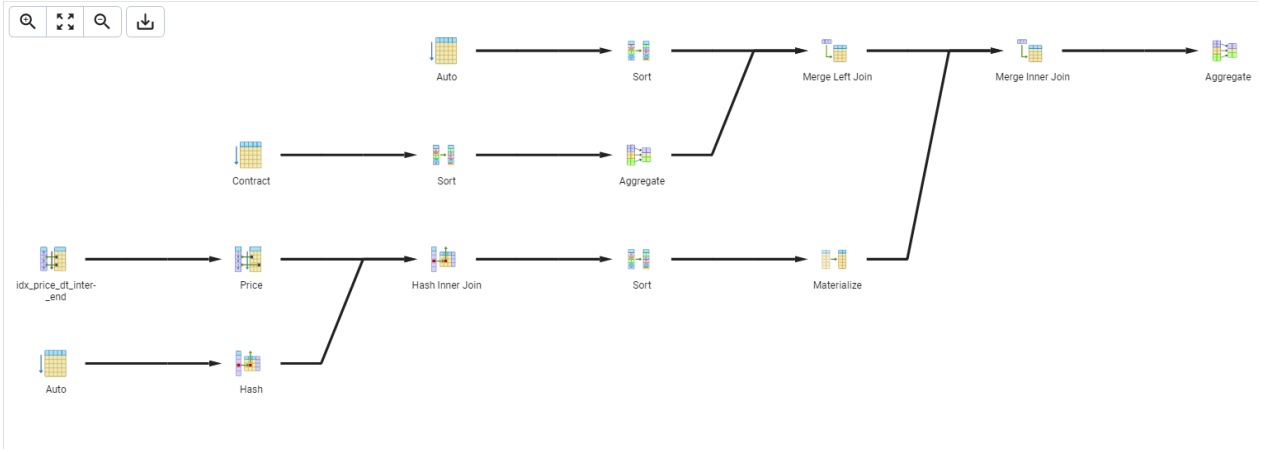
```
1 SELECT SUM(p.current_price * h.hours_wo_rent) AS total_loss
2 FROM (
3 SELECT a."Auto_Code",
4 (24 - LEAST(COALESCE(hours_rented, 0), 24)) AS hours_wo_rent
5 FROM lab."Auto" a
6 LEFT JOIN (
7 SELECT "Auto_Code",
8 COALESCE(CAST(SUM(EXTRACT(EPOCH FROM ("Factual_DT_Ret" - "DT_Car_Transf_To_Cl")) / 3600 AS INTEGER), 0) AS hours_rented
9 FROM lab."Contract"
10 WHERE DATE("DT_Car_Transf_To_Cl") = DATE(CURRENT_DATE - 3)
11 GROUP BY "Auto_Code"
```

Data Output Messages Notifications

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

После индексации: время выполнения 46 мс

Итог: при помощи индексации мы ускорили время выполнения запроса на 32мс (~25%)



Graphical Analysis Statistics				
#	Node	Rows Actual	Loops	
1.	→ Aggregate (rows=1 loops=1)		1	1
2.	→ Merge Inner Join (rows=32 loops=1)		32	1
3.	→ Merge Left Join (rows=32 loops=1)		32	1
4.	→ Sort (rows=32 loops=1)		32	1
5.	→ Seq Scan on Auto as a (rows=32 loops=1)		32	1
6.	→ Aggregate (rows=0 loops=1)		0	1
7.	→ Sort (rows=0 loops=1)		0	1
8.	→ Seq Scan on Contract as Contract (rows=0 loops=1) Filter: (date("DT_Car_Transf_To_CIT") < (CURRENT_DATE - 3)) Rows Removed by Filter: 33		0	1
9.	→ Materialize (rows=32 loops=1)		32	1
10.	→ Sort (rows=32 loops=1)		32	1
11.	→ Hash Inner Join (rows=32 loops=1) Hash Cond: (p."Mod_Code" = (a_1."Mod_Code")::integer)		32	1
12.	→ Bitmap Heap Scan on Price as p (rows=8 loops=1) Recheck Cond: ("DT_Inter_End" IS NULL) Heap Blocks: exact=1		8	1
13.	→ Bitmap Index Scan using idx_price_dt_inter_end (rows=8 loops=1) Index Cond: ("DT_Inter_End" IS NULL)		8	1
14.	→ Hash (rows=32 loops=1) Buckets: 1024 Batches: 1 Memory Usage: 10 kB		32	1
15.	→ Seq Scan on Auto as a_1 (rows=32 loops=1)		32	1

5. Выводы:

По итогам выполнения лабораторной работы были выполнены SELECT запросы к созданной базе данных, я ознакомился с созданием запросов INSERT, UPDATE и DELETE, а также с графическим представлением запросов. Я также изучил, как создавать простые и составные индексы, и заметил, что это позволяет сокращать количество этапов выполнения запросов и снижать время выполнения.