

Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное образовательное учреждение
высшего образования

«Национальный исследовательский университет ИТМО»

Факультет инфокоммуникационных технологий

Лабораторная работа №2

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

Выполнила:

студентка 2 курса ИКТ группа K3242

Крамаренко И.С.

Проверила:

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.

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:

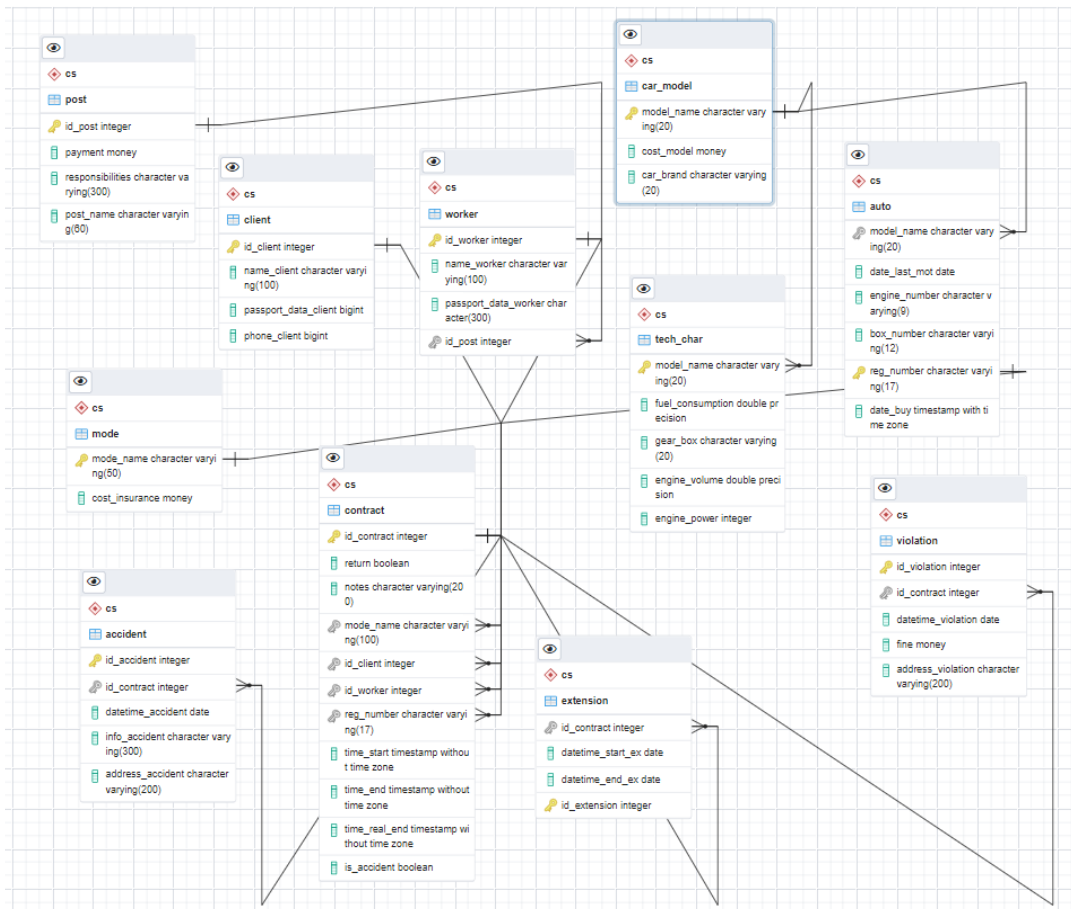



Рисунок 1 - Схема логической модели базы данных

Выполнение:

Запросы:

- Какой автомобиль находился в прокате максимальное количество часов?


```
select reg_number
from cs.contract
where time_real_end - time_start = (
    select max(time_real_end - time_start)
    from cs.contract)
```

	reg_number character varying (17) 
1	CH3KDHRYU73892173

- Автомобили какой марки чаще всего брались в прокат?

```
with brand_count as (  
    select car_brand, count(*)  
    from (  
        select * from cs.contract  
        inner join cs.auto on contract.reg_number=auto.reg_number  
        inner          join          cs.car_model          on  
        auto.model_name=car_model.model_name  
    ) as car_contract  
    group by car_brand  
)
```

```
select car_brand from brand_count  
where count = (  
    select max(count) from brand_count)
```

	car_brand character varying (20) 
1	Kia

- Определить убытки от простоя автомобилей за вчерашний день.

```

with lose_money_table as(
with noworktime as (select ('24:00:00'-sum(diff)) time_int from (
    select *, (time_real_end - time_start) diff
    from cs.contract where time_start::date = '2022-02-27' and
time_real_end::date = '2022-02-27'
    union select *, (time_real_end - '2022-02-27 00:00:00') diff
    from cs.contract where time_start::date < '2022-02-27' and
time_real_end::date = '2022-02-27'
    union select *, ('2022-02-27 23:59:59' - time_start) diff
    from cs.contract where time_start::date = '2022-02-27' and
time_real_end is null
    union select *, null as diff from cs.contract where time_start::date <
'2022-02-27' and time_real_end is null) big_table
group by reg_number)

```

```

select (extract(hour from nwtime)*60*11 +
round(extract(minute from nwtime)*11)) lose_money from (select
sum(time_int) nwtime from noworktime) nwrk
union
select (count(reg_number)*24*60*11) lose_money
from cs.auto where reg_number not in (
    select reg_number from cs.contract
    where time_real_end is null or time_real_end::date='2022-02-27'))

```

```

select sum(lose_money) as “Убытки от простоя автомобилей в руб.
относительно поминутного тарифа” from lose_money_table

```

	Убытки от простоя автомобилей в ру.. double precision
1	107734

- Вывести данные автомобиля, имеющего максимальный пробег.

```

with auto_max as (
  with rent_time as (
    select reg_number, sum(time_real_end - time_start)
    from cs.contract group by reg_number
  ) select reg_number, sum
  from rent_time
  where sum = (select max(sum) from rent_time)
)
select * from cs.auto
where auto.reg_number=(
  select reg_number from auto_max
)

```

	model_name character varying (20)	date_last_mot date	engine_number character varying (9)	box_number character varying (12)	reg_number [PK] character varying (17)
1	Rio	2021-11-14	UF-378230	SHE78FU7DKRU	CH3KDHRYU73892173

- Какой автомобиль суммарно находился в прокате дольше всех.

```
with rent_time as (
    select reg_number, sum(time_real_end - time_start)
    from cs.contract
    group by reg_number
)
```

```
select reg_number, sum
from rent_time
where sum = (select max(sum)
from rent_time)
```

	reg_number character varying (17)	sum interval
1	CH3KDHRYU73892173	2 days 32:02:40

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

```
select auto.model_name as "Марка или модель машины",
count(auto.model_name) as "Количество"
from cs.auto
group by auto.model_name
union select car_model.car_brand, count(*)
from cs.auto
inner join cs.car_model on
auto.model_name=car_model.model_name
group by car_brand
```


	Марка или модель машины character varying (20)	Количество bigint
1	Volswagen	2
2	Kia	4
3	Rio X-Line	1
4	Kaptur	2
5	Solaris	2
6	Rio	3
7	Renault	2
8	Hyundai	2
9	Polo	2

- Определить средний “возраст” автомобилей компании.

```
select round(extract(day from age)/365) as "Средний возраст
автомобиля"
from (
    select avg(current_timestamp - date_buy) age
    from cs.auto
) aver_age
```

Средний возраст автомобиля double precision
5

Представления:

- Какой автомобиль ни разу не был в прокате?

```
create or replace view cs.car_not_used
as select reg_number
from cs.auto
where reg_number not in (
    select reg_number from cs.contract
)
```

	reg_number
	character varying (17)
1	CH4CHFKOD27364072
2	DV4KRMUOS37950264

- Вывести данные клиентов, не вернувших автомобиль вовремя.

create or replace view cs.bad_clients

as select * from cs.client

where id_client in (

select id_client from cs.contract

where time_real_end > time_end or time_real_end is null

)

	id_client	name_client	passport_data_client	phone_client
	integer	character varying (100)	bigint	bigint
1	1	Крамаренко Ирина Сергеевна	7516861853	9110065465
2	2	Хабибулин Илья Радикович	7516843567	9510637804
3	6	Тиунова Анна Васильевна	7516359260	9112685295
4	7	Боброва Мария Иосифовна	7614736542	9521348356
5	8	Третьякова Екатерина Сергеевна	7515678304	9423570472

Запросы на модификацию данных:

- INSERT






Добавить данные о клиентах, у которых 3 и более нарушений в “черный список”

```
insert into cs.blacklist_clients(id_client, name, passport_data, phone)
```

```
with client_count as (  
    select count(*) as cl_count, id_client  
    from cs.violation  
    left join cs.contract  
    on contract.id_contract=violation.id_contract  
    group by contract.id_client)
```

```
select id_client, name_client, passport_data_client, phone_client  
from cs.client  
where id_client in (  
    select id_client from client_count  
    where cl_count > 2)
```

После:

	 id_client integer	 name character varying	 passport_data bigint	 phone bigint	 id [PK] integer
1	5	Кукрякова Рената Радиковна	7715789352	9524178625	1

- UPDATE

Обновить статус аварии в контракте.

update cs.contract

set is_accident = true

where id_contract in (

select id_contract from cs.accident

)

	mode_name character varying (100)	id_client integer	id_worker integer	reg_number character varying (17)	time_start timestamp without time zone	time_end timestamp without time zone	time_real_end timestamp without time zone	is_accident boolean
	Поминутно	7	3	BH3UFJFWP67809732	2022-02-23 17:04:06	2022-02-23 17:58:35	2022-02-23 17:58:35	false
	Фикс	2	6	CH3RBHJLI73904752	2022-02-24 10:06:49	2022-02-24 13:27:38	2022-02-24 13:27:38	false
	Поминутно	1	3	DH7DGMIS62045273	2022-02-24 15:46:14	2022-02-24 16:00:31	2022-02-24 16:00:31	false
	Дни	3	7	CH3KDHRYU73892173	2022-02-24 18:22:45	2022-02-27 18:22:45	2022-02-27 18:01:53	false
	Фикс	7	6	EH2SHRTID37483502	2022-02-25 17:36:49	2022-02-25 20:40:28	2022-02-25 20:40:28	false
	Часы	1	3	HJ4SHYCKR37809484	2022-02-25 15:03:42	2022-02-25 21:03:42	2022-02-25 22:38:27	false
	Часы	2	7	EH6LDTTRIF26370451	2022-02-26 09:56:01	2022-02-26 11:56:01	[null]	false
	Фикс	5	6	CH3KDHRYU73892173	2022-02-26 12:35:51	2022-02-26 19:29:55	2022-02-26 19:29:55	false
	Поминутно	7	3	DH7DGMIS62045273	2022-02-26 16:27:20	2022-02-26 16:45:28	2022-02-26 16:45:28	false
	Фикс	6	3	VF7LCNFUC74747544	2022-02-26 20:11:32	2022-02-26 23:15:39	[null]	false

	mode_name character varying (100)	id_client integer	id_worker integer	reg_number character varying (17)	time_start timestamp without time zone	time_end timestamp without time zone	time_real_end timestamp without time zone	is_accident boolean
	Поминутно	7	3	BH3UFJFWP67809732	2022-02-23 17:04:06	2022-02-23 17:58:35	2022-02-23 17:58:35	false
	Фикс	2	6	CH3RBHJLI73904752	2022-02-24 10:06:49	2022-02-24 13:27:38	2022-02-24 13:27:38	false
	Поминутно	1	3	DH7DGMIS62045273	2022-02-24 15:46:14	2022-02-24 16:00:31	2022-02-24 16:00:31	false
	Дни	3	7	CH3KDHRYU73892173	2022-02-24 18:22:45	2022-02-27 18:22:45	2022-02-27 18:01:53	true
	Фикс	7	6	EH2SHRTID37483502	2022-02-25 17:36:49	2022-02-25 20:40:28	2022-02-25 20:40:28	false
	Часы	1	3	HJ4SHYCKR37809484	2022-02-25 15:03:42	2022-02-25 21:03:42	2022-02-25 22:38:27	false
	Часы	2	7	EH6LDTTRIF26370451	2022-02-26 09:56:01	2022-02-26 11:56:01	[null]	false
	Фикс	5	6	CH3KDHRYU73892173	2022-02-26 12:35:51	2022-02-26 19:29:55	2022-02-26 19:29:55	false
	Поминутно	7	3	DH7DGMIS62045273	2022-02-26 16:27:20	2022-02-26 16:45:28	2022-02-26 16:45:28	false
	Фикс	6	3	VF7LCNFUC74747544	2022-02-26 20:11:32	2022-02-26 23:15:39	[null]	false
	Часы	11	6	EH2SHRTID37483502	2022-02-26 22:08:44	2022-02-27 05:27:10	2022-02-27 05:20:18	false
	Поминутно	9	7	CH3RBHJLI73904752	2022-02-27 13:52:05	2022-02-27 14:37:50	2022-02-27 14:37:50	true
	Фикс	1	6	CH3KDHRYU73892173	2022-02-27 14:19:57	2022-02-27 15:41:36	2022-02-27 15:49:25	false
	Фикс	8	7	EH2SHRTID37483502	2022-02-27 20:51:32	2022-02-27 21:59:03	[null]	false
	Часы	7	3	DH6KJDHUI89032873	2022-02-28 22:04:38	2022-03-01 22:04:38	2022-03-02 04:29:53	false
	Фикс	2	6	EH2SHRTID37483502	2022-02-28 10:47:39	2022-02-28 15:38:20	[null]	false

- DELETE

Удалить автомобили, у которых “возраст” более 10 лет.

```
with count_age_car as (
    select      reg_number,      (extract(year      from
diff)*365*24*60*60*1000 +
    extract(day from diff)*24*60*60*1000 +
    extract(hour from diff)*60*60*1000 +
    extract(minute from diff)*60*1000 +
    round(extract(second from diff)*1000)) age
    from (select reg_number, (current_timestamp - date_buy)
diff from cs.auto) age_car
)
```

```
delete from cs.auto
where reg_number in (
    select reg_number from count_age_car
    where age > 3153600000000)
```

До:

	model_name character varying (20)	date_last_mot date	engine_number character varying (9)	box_number character varying (12)	reg_number [PK] character varying (17)	date_buy timestamp with time zone
1	Polo	2022-02-17	GR-378212	HSE27KE7BSTD	BH3UFJFWP67809732	2018-04-13 00:00:00+03
2	Rio	2021-11-14	UF-378230	SHE78FU7DKRU	CH3KDHRYU73892173	2016-09-25 00:00:00+03
3	Kaptur	2022-02-10	DY-789320	DNR73ND7JDNF	CH3RBHJLI73904752	2020-12-05 00:00:00+03
4	Solaris	2021-12-02	KI-678432	EHW67DH2JDHF	CH4CHFKOD27364072	2013-03-01 00:00:00+04
5	Rio	2022-01-15	LD-267832	LIS78TD1HJKF	DH6KJDHUI89032873	2017-07-10 00:00:00+03
6	Kaptur	2022-02-13	GR-378923	FHY26DK8BDGE	DH7DGVWMIS62045273	2016-05-30 00:00:00+03
7	Solaris	2022-01-31	KI-372892	DHU26DT2HUJF	EH2SHRTID37483502	2018-04-13 00:00:00+03
8	Rio X-Line	2021-12-12	FO-478923	GYD78LF2GYFI	EH6LDTRIF26370451	2017-12-12 00:00:00+03
9	Rio	2021-12-25	ZJ-789423	DHR26DJ2SHRT	HJ4SHYCKR37809484	2019-08-21 00:00:00+03
10	Polo	2022-02-23	ZJ-257652	NFU10FX7TPSA	VF7LCNFUC74747544	2020-02-13 00:00:00+03
11	Solaris	2022-01-14	DO-874023	DEP30VU8FJEL	DV4KRMUOS37950264	2011-02-03 00:00:00+03

После:

	model_name character varying (20)	date_last_mot date	engine_number character varying (9)	box_number character varying (12)	reg_number [PK] character varying (17)	date_buy timestamp with time zone
1	Polo	2022-02-17	GR-378212	HSE27KE7BSTD	BH3UFJFWP67809732	2018-04-13 00:00:00+03
2	Rio	2021-11-14	UF-378230	SHE78FU7DKRU	CH3KDHRYU73892173	2016-09-25 00:00:00+03
3	Kaptur	2022-02-10	DY-789320	DNR73ND7JDNF	CH3RBHJLI73904752	2020-12-05 00:00:00+03
4	Solaris	2021-12-02	KI-678432	EHW67DH2JDHF	CH4CHFOD27364072	2013-03-01 00:00:00+04
5	Rio	2022-01-15	LD-267832	LIS78TD1HJKF	DH6KJDHUI89032873	2017-07-10 00:00:00+03
6	Kaptur	2022-02-13	GR-378923	FHY26DK8BDGE	DH7DGMIS62045273	2016-05-30 00:00:00+03
7	Solaris	2022-01-31	KI-372892	DHU26DT2HUIF	EH2SHRTID37483502	2018-04-13 00:00:00+03
8	Rio X-Line	2021-12-12	FO-478923	GVD78LF2GYFI	EH6LDTRIF26370451	2017-12-12 00:00:00+03
9	Rio	2021-12-25	ZJ-789423	DHR26DJ2SHRT	HJ4SHYCKR37809484	2019-08-21 00:00:00+03
10	Polo	2022-02-23	ZJ-257652	NFU10FX7TPSA	VF7LCNFUC74747544	2020-02-13 00:00:00+03

Создание индексов:

1. Без индекса

```
1) explain select reg_number
from cs.contract
where time_real_end - time_start = (
    select max(time_real_end - time_start)
    from cs.contract
)
```

	QUERY PLAN
	text
1	Seq Scan on contract (cost=1.25..2.49 rows=1 width=18)
2	[...] Filter: ((time_real_end - time_start) = \$0)
3	[...] InitPlan 1 (returns \$0)
4	[...] -> Aggregate (cost=1.24..1.25 rows=1 width=16)
5	[...] -> Seq Scan on contract contract_1 (cost=0.00..1.16 rows=16 width=16)

Запрос выполнен успешно. Общее время выполнения: 224 msec.,
обработано строк: 5.

```

2) explain with auto_max as (
    with rent_time as (
        select reg_number, sum(time_real_end - time_start)
        from cs.contract group by reg_number
    ) select reg_number, sum
    from rent_time
    where sum = (select max(sum) from rent_time)
)
select * from cs.auto
where auto.reg_number=(
    select reg_number from auto_max
)

```

	QUERY PLAN	
	text	
1	Seq Scan on auto (cost=1.79..2.92 rows=1 width=200)	
2	[...] Filter: ((reg_number)::text = (\$2)::text)	
3	[...] InitPlan 3 (returns \$2)	
4	[...] -> Subquery Scan on auto_max (cost=1.58..1.79 rows=1 width=52)	
5	[...] -> CTE Scan on rent_time rent_time_1 (cost=1.58..1.78 rows=1 width=68)	
6	[...] Filter: (sum = \$1)	
7	[...] CTE rent_time	
8	[...] -> HashAggregate (cost=1.28..1.37 rows=9 width=34)	
9	[...] Group Key: contract.reg_number	
10	[...] -> Seq Scan on contract (cost=0.00..1.16 rows=16 width=34)	
11	[...] InitPlan 2 (returns \$1)	
12	[...] -> Aggregate (cost=0.20..0.21 rows=1 width=16)	
13	[...] -> CTE Scan on rent_time (cost=0.00..0.18 rows=9 width=16)	

Запрос выполнен успешно. Общее время выполнения: 205 msec.,
обработано строк: 13.

2. Создание индексов

1) Создание простого индекса:

```
create index id_cntr on cs.contract(id_contract)
```

2) Создание составного индекса:

```
create index id_car on cs.auto(model_name, date_last_mot,
engine_number, box_number, reg_number)
```

3. Запросы с индексами

- 1) Запрос выполнен успешно. Общее время выполнения: 92 msec., обработано строк: 5.

	QUERY PLAN	
	text	
1	Seq Scan on contract (cost=1.25..2.49 rows=1 width=18)	
2	[...] Filter: ((time_real_end - time_start) = \$0)	
3	[...] InitPlan 1 (returns \$0)	
4	[...] -> Aggregate (cost=1.24..1.25 rows=1 width=16)	
5	[...] -> Seq Scan on contract contract_1 (cost=0.00..1.16 rows=16 width=16)	

- 2) Запрос выполнен успешно. Общее время выполнения: 140 msec., обработано строк: 13.

	QUERY PLAN	
	text	
1	Seq Scan on auto (cost=1.79..2.92 rows=1 width=200)	
2	[...] Filter: ((reg_number)::text = (\$2)::text)	
3	[...] InitPlan 3 (returns \$2)	
4	[...] -> Subquery Scan on auto_max (cost=1.58..1.79 rows=1 width=52)	
5	[...] -> CTE Scan on rent_time rent_time_1 (cost=1.58..1.78 rows=1 width=68)	
6	[...] Filter: (sum = \$1)	
7	[...] CTE rent_time	
8	[...] -> HashAggregate (cost=1.28..1.37 rows=9 width=34)	
9	[...] Group Key: contract.reg_number	
10	[...] -> Seq Scan on contract (cost=0.00..1.16 rows=16 width=34)	
11	[...] InitPlan 2 (returns \$1)	
12	[...] -> Aggregate (cost=0.20..0.21 rows=1 width=16)	
13	[...] -> CTE Scan on rent_time (cost=0.00..0.18 rows=9 width=16)	

4. Удаление индексов

- 1) drop index id_cntr
- 2) drop index id_car

Вывод: в ходе выполнения лабораторной работы были изучены такие программные обеспечения, как СУБД PostgreSQL 1X, pgAdmin 4, созданы

представления и запросы на выборку данных к базе данных PostgreSQL, использованы подзапросы при модификации данных и индексов.