

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФГБОУ ВО «ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»

БЕЗОПАСНОСТЬ СИСТЕМ БАЗ ДАННЫХ

СОЗДАНИЕ БАЗЫ ДАННЫХ В POSTGRESQL

Выполнил: Искорнев Дмитрий Романович

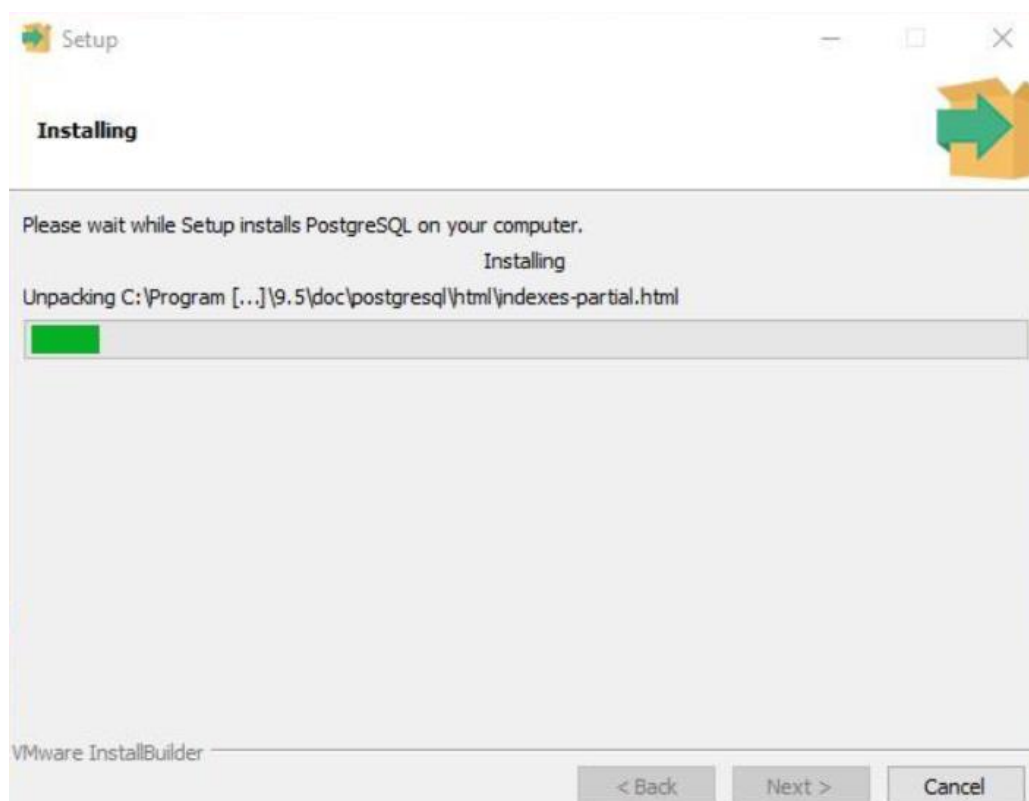
Группа: УБ-02

Номер зачетной книжки: 207418

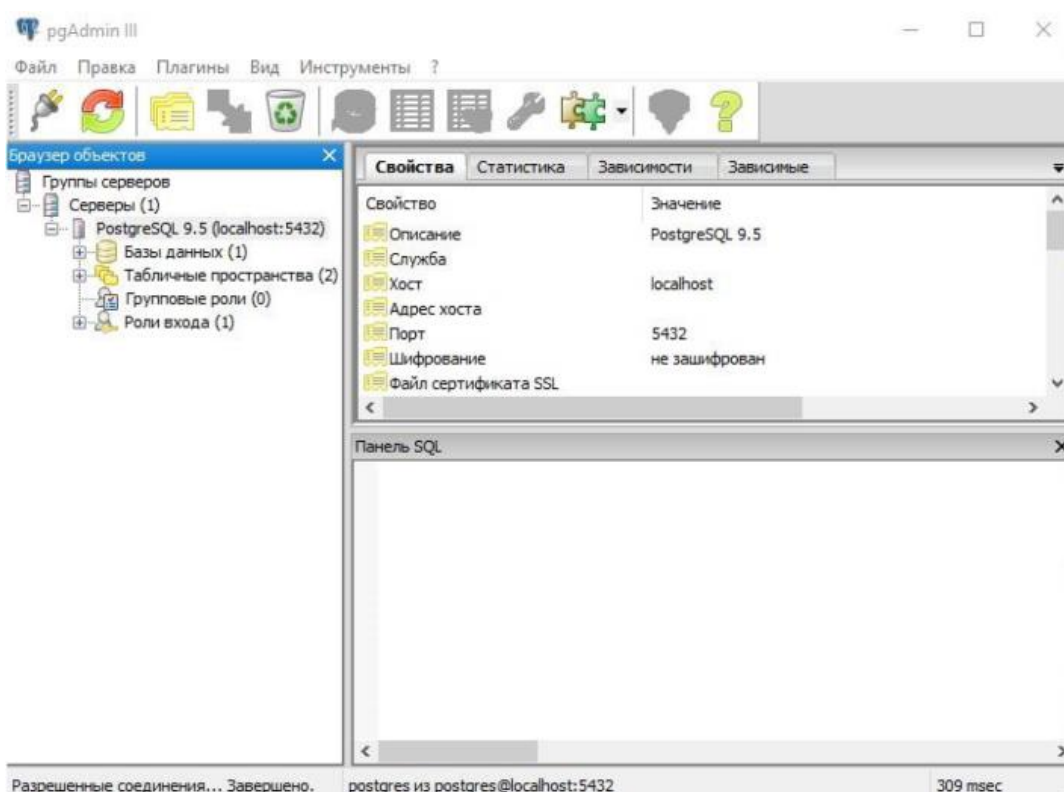
ВОРОНЕЖ

2024

## Установка PostgreSQL на Windows

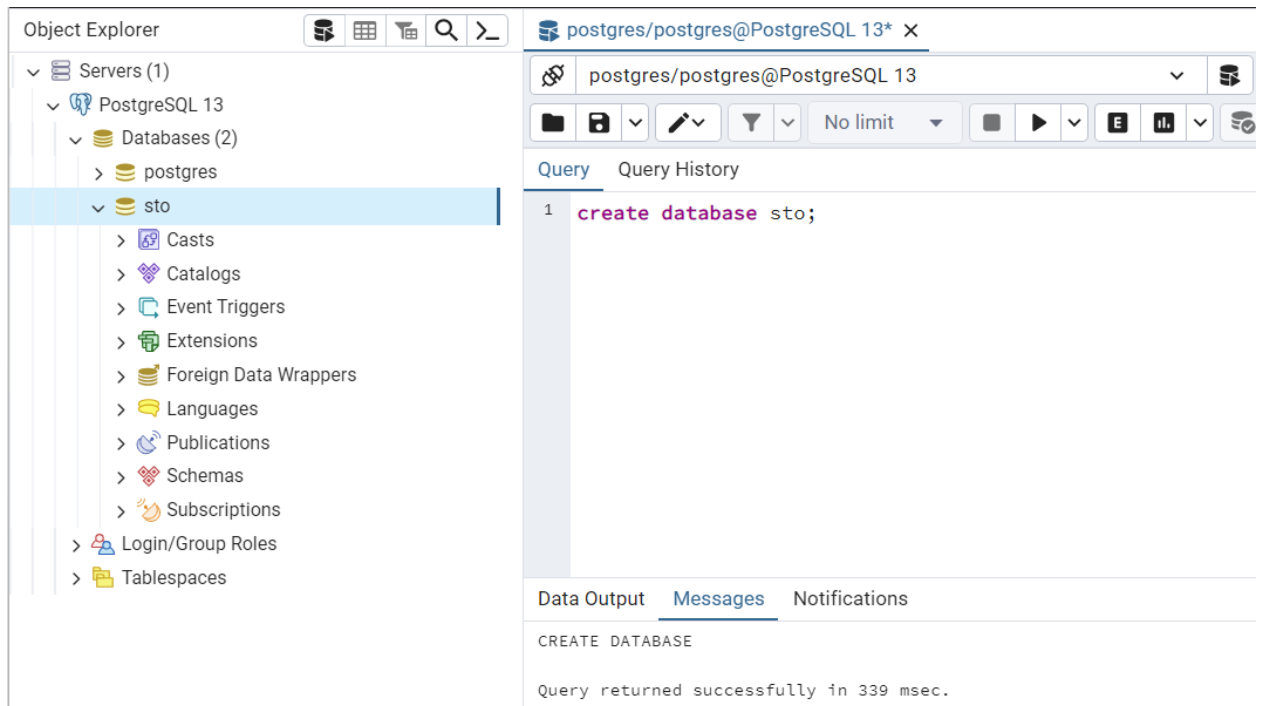


Заходим в PgAdmin и проверяем корректность установки, вместе с этим осуществляем авторизацию под пользователем postgres (вводим пароль, который указали при установке)



## Создание БД

Для создания новой базы данных используем оператор CREATE DATABASE. В браузере объектов pgAdmin можно увидеть, что база данных была создана корректно.



Для создания таблиц для базы данных используется оператор CREATE TABLE. Создадим следующие таблицы: Maintenance (Обслуживание), Customer (Клиент), Vehicle (Транспортное средство), Auto\_master (Мастер).

```
create table auto_master(  
    id SERIAL PRIMARY KEY,  
    name VARCHAR(20),  
    surname VARCHAR(20),  
    qualification VARCHAR(20)  
);  
  
create table customer(  
    id SERIAL PRIMARY KEY,  
    name VARCHAR(20),  
    surname VARCHAR(20),  
    phone_number VARCHAR(25)  
);
```

```

create table vehicle (
    id SERIAL PRIMARY KEY,
    customer_id int REFERENCES customer(id),
    model VARCHAR(20),
    brand VARCHAR(20),
    reg_number VARCHAR(20)
);

create table maintenance(
    id SERIAL PRIMARY KEY,
    auto_master_id int REFERENCES auto_master(id),
    vehicle_id int REFERENCES vehicle(id),
    scope_list VARCHAR(300),
    registration_date DATE,
    registration_time TIME,
    work_cost int,
    part_list VARCHAR(300)
);

```

Подключимся к БД с помощью утилиты psql и отобразим в ней список созданных сущностей:

\dt – список созданных таблиц

```

PS E:\PostgreSQL\pgAdmin 4\runtime> .\psql -h localhost -U postgres -p 5432 -d sto
Password for user postgres:

psql (13.14)
WARNING: Console code page (866) differs from Windows code page (1251)
        8-bit characters might not work correctly. See psql reference
        page "Notes for Windows users" for details.
Type "help" for help.

sto=# \dt

```

| List of relations |             |       |          |
|-------------------|-------------|-------|----------|
| Schema            | Name        | Type  | Owner    |
| public            | auto_master | table | postgres |
| public            | customer    | table | postgres |
| public            | maintenance | table | postgres |
| public            | vehicle     | table | postgres |

```

(4 rows)

```

\di – список созданных индексов

```
sto=# \di
```

| List of relations |                  |       |          |             |
|-------------------|------------------|-------|----------|-------------|
| Schema            | Name             | Type  | Owner    | Table       |
| public            | auto_master_pkey | index | postgres | auto_master |
| public            | customer_pkey    | index | postgres | customer    |
| public            | maintenance_pkey | index | postgres | maintenance |
| public            | vehicle_pkey     | index | postgres | vehicle     |

(4 rows)

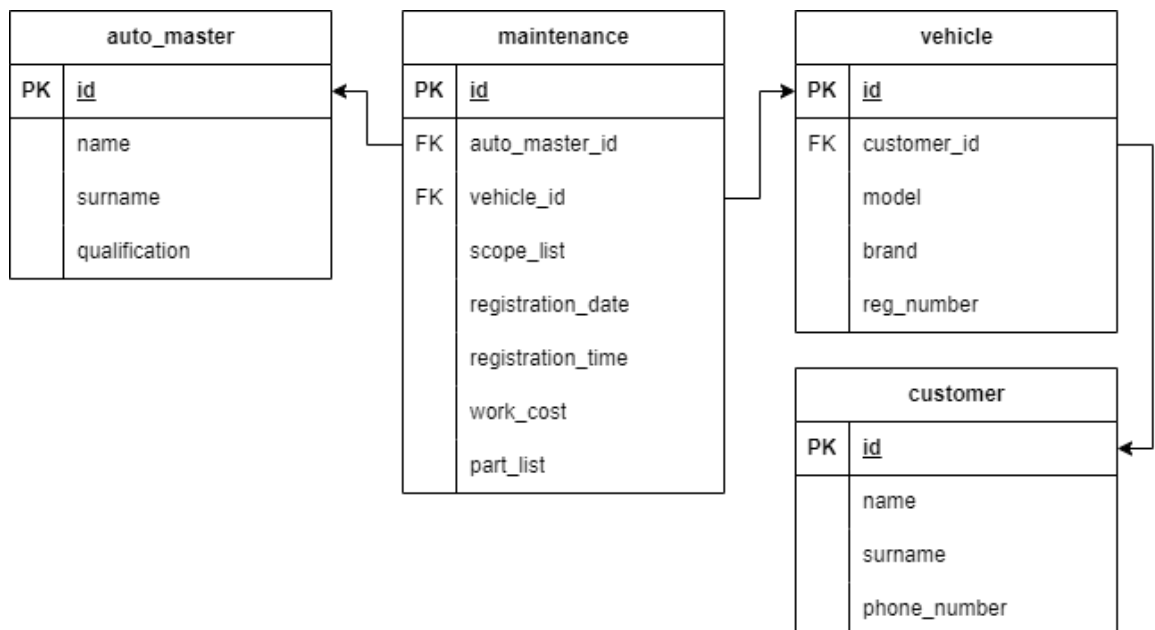
\ds – список созданных последовательностей

```
sto=# \ds
```

| List of relations |                    |          |          |
|-------------------|--------------------|----------|----------|
| Schema            | Name               | Type     | Owner    |
| public            | auto_master_id_seq | sequence | postgres |
| public            | customer_id_seq    | sequence | postgres |
| public            | maintenance_id_seq | sequence | postgres |
| public            | vehicle_id_seq     | sequence | postgres |

(4 rows)

Схема базы данных:



Для вставки данных в таблицы служит оператор INSERT. Для того, чтобы вывести данные таблицы используем оператор SELECT.

Таблица auto\_master:

sto/postgres@PostgreSQL 13

Query

Query History

```
1 insert into auto_master(name, surname, qualification)
2 values ('Иван','Ручкин','Механик');
3
4 insert into auto_master(name, surname, qualification)
5 values ('Василий','Тупия','Приемщик');
6
7 insert into auto_master(name, surname, qualification)
8 values ('Григорий','Карташов','Электрик');
9
10 insert into auto_master(name, surname, qualification)
11 values ('Татьяна','Хамова','Механик');
12
13 insert into auto_master(name, surname, qualification)
14 values ('Константин','Балашов','Ст.механик');
15
```

```
16 select * from auto_master;
17
18
```

Data Output

Messages

Notifications

|   | id<br>[PK] integer | name<br>character varying (20) | surname<br>character varying (20) | qualification<br>character varying (20) |
|---|--------------------|--------------------------------|-----------------------------------|---|
| 1 | 1                  | Иван                           | Ручкин                            | Механик                                 |
| 2 | 2                  | Василий                        | Тупия                             | Приемщик                                |
| 3 | 3                  | Григорий                       | Карташов                          | Электрик                                |
| 4 | 4                  | Татьяна                        | Хамова                            | Механик                                 |
| 5 | 5                  | Константин                     | Балашов                           | Ст.механик                              |

Таблица customer:

sto/postgres@PostgreSQL 13

No limit

Query Query History

```
1 insert into customer(name, surname, phone_number)
2 values ('Дмитрий', 'Искорнев', '+79204118282');
3
4 insert into customer(name, surname, phone_number)
5 values ('Иван', 'Иванов', '+79504318682');
6
7 insert into customer(name, surname, phone_number)
8 values ('Денис', 'Воробьев', '+79514416882');
9
10 insert into customer(name, surname, phone_number)
11 values ('Егор', 'Слепаков', '+79806551281');
12
13 insert into customer(name, surname, phone_number)
14 values ('Дмитрий', 'Бабочкин', '+79350444858');
15
```

```
16 select * from customer;
17
```

Data Output Messages Notifications

|   | id<br>[PK] integer | name<br>character varying (20) | surname<br>character varying (20) | phone_number<br>character varying (25) |
|---|--------------------|--------------------------------|-----------------------------------|--|
| 1 | 1                  | Дмитрий                        | Искорнев                          | +79204118282                           |
| 2 | 2                  | Иван                           | Иванов                            | +79504318682                           |
| 3 | 3                  | Денис                          | Воробьев                          | +79514416882                           |
| 4 | 4                  | Егор                           | Слепаков                          | +79806551281                           |
| 5 | 5                  | Дмитрий                        | Бабочкин                          | +79350444858                           |

Таблица vehicle:

Query
Query History

```

1 insert into vehicle(customer_id, model, brand, reg_number)
2 values(3,'Focus','Ford','Y214KP36');
3
4 insert into vehicle(customer_id, model, brand, reg_number)
5 values(2,'Granta','Lada','B555TY136');
6
7 insert into vehicle(customer_id, model, brand, reg_number)
8 values(5,'Priora','Lada','B125YK36');
9
10 insert into vehicle(customer_id, model, brand, reg_number)
11 values(1,'Lacetti','Chevrolet','H563YK36');
12
13 insert into vehicle(customer_id, model, brand, reg_number)
14 values(4,'C300','Mercedes','A111AA36');
15
16 select * from vehicle;
17

```

Data Output
Messages
Notifications

|   | id<br>[PK] integer | customer_id<br>integer | model<br>character varying (20) | brand<br>character varying (20) | reg_number<br>character varying (20) |
|---|--------------------|------------------------|---------------------------------|---------------------------------|--------------------------------------|
| 1 | 1                  | 3                      | Focus                           | Ford                            | Y214KP36                             |
| 2 | 2                  | 2                      | Granta                          | Lada                            | B555TY136                            |
| 3 | 3                  | 5                      | Priora                          | Lada                            | B125YK36                             |
| 4 | 4                  | 1                      | Lacetti                         | Chevrolet                       | H563YK36                             |
| 5 | 5                  | 4                      | C300                            | Mercedes                        | A111AA36                             |



Таблица maintenance:

```
sto/postgres@PostgreSQL 13
Query History
1 insert into maintenance(auto_master_id, vehicle_id, scope_list, registration_date,
2 registration_time, work_cost, part_list)
3 values(1,2,'Замена масла', '2024-01-13', '08:00:00',
4 5000, 'Моторное масло, масляный фильтр');
5
6 insert into maintenance(auto_master_id, vehicle_id, scope_list, registration_date,
7 registration_time, work_cost, part_list)
8 values(2,1,'Замена тормозных колодок', '2024-01-13', '08:30:00',
9 3400, 'Тормозные колодки');
10
11 insert into maintenance(auto_master_id, vehicle_id, scope_list, registration_date,
12 registration_time, work_cost, part_list)
13 values(5,3,'Замена свечей зажигания', '2024-01-13', '09:30:00',
14 5000, 'Свечи зажигания');
15
16 insert into maintenance(auto_master_id, vehicle_id, scope_list, registration_date,
17 registration_time, work_cost, part_list)
18 values(1,4,'Замена АКБ', '2024-01-14', '15:30:00',
19 5000, 'АКБ');
20
21 insert into maintenance(auto_master_id, vehicle_id, scope_list, registration_date,
22 registration_time, work_cost, part_list)
23 values(5,5,'Замена ремня ГРМ', '2024-01-14', '10:00:00',
24 5000, 'Ремень ГРМ, Натяжитель, Ролики ГРМ');
```

16 **select** \* **from** maintenance;

17

18

Data Output

Messages

Notifications

|   | <b>id</b><br>[PK] integer | <b>auto_master_id</b><br>integer | <b>vehicle_id</b><br>integer | <b>scope_list</b><br>character varying (300) | <b>registration_date</b><br>date | <b>registration_time</b><br>time without time zone | <b>work_cost</b><br>integer | <b>part_list</b><br>character varying (300) |
|---|---------------------------|----------------------------------|------------------------------|--|----------------------------------|--|-----------------------------|---|
| 1 | 1                         | 1                                | 2                            | Замена масла                                 | 2024-01-13                       | 08:00:00   | 5000                        | Моторное масло, масляны                     |
| 2 | 2                         | 2                                | 1                            | Замена тормозных колодок                     | 2024-01-13                       | 08:30:00   | 3400                        | Тормозные колодки                           |
| 3 | 3                         | 5                                | 3                            | Замена свечей зажигания                      | 2024-01-13                       | 09:30:00   | 5000                        | Свечи зажигания                             |
| 4 | 4                         | 1                                | 4                            | Замена АКБ                                   | 2024-01-14                       | 15:30:00   | 5000                        | АКБ   |
| 5 | 5                         | 5                                | 5                            | Замена ремня ГРМ                             | 2024-01-14                       | 10:00:00   | 5000                        | Ремень ГРМ, Натяжитель                      |

Для чтения заданных столбцов из одиночной таблицы используется оператор **SELECT** <название столбца>, ... **FROM** <название таблицы>;

Например, чтобы вывести дату, время и список работ по записи в СТО, необходимо выполнить следующий запрос:

```

16 select registration_date, registration_time, scope_list
17 from maintenance;
18

```

| Data Output Messages Notifications |                           |   |                                       |
|------------------------------------|---------------------------|---|---------------------------------------|
|                                    | registration_date<br>date | registration_time<br>time without time zone | scope_list<br>character varying (300) |
| 1                                  | 2024-01-13                | 08:00:00                                    | Замена масла                          |
| 2                                  | 2024-01-13                | 08:30:00                                    | Замена тормозных колодок              |
| 3                                  | 2024-01-13                | 09:30:00                                    | Замена свечей зажигания               |
| 4                                  | 2024-01-14                | 15:30:00                                    | Замена АКБ                            |
| 5                                  | 2024-01-14                | 10:00:00                                    | Замена ремня ГРМ                      |

Для того, чтобы вывести список клиентов с сортировкой по фамилии по убыванию номера первой буквы, необходимо использовать запрос:

```

16 select name, surname
17 from customer
18 order by surname asc;
19

```

| Data Output Messages Notifications |                                |                                   |
|------------------------------------|--------------------------------|-----------------------------------|
|                                    | name<br>character varying (20) | surname<br>character varying (20) |
| 1                                  | Дмитрий                        | Бабочкин                          |
| 2                                  | Денис                          | Воробьев                          |
| 3                                  | Иван                           | Иванов                            |
| 4                                  | Дмитрий                        | Искорнев                          |
| 5                                  | Егор                           | Слепаков                          |

Для того, чтобы из общего списка автомобилей выбрать только автомобили марки Lada, необходимо выполнить следующий запрос:

```

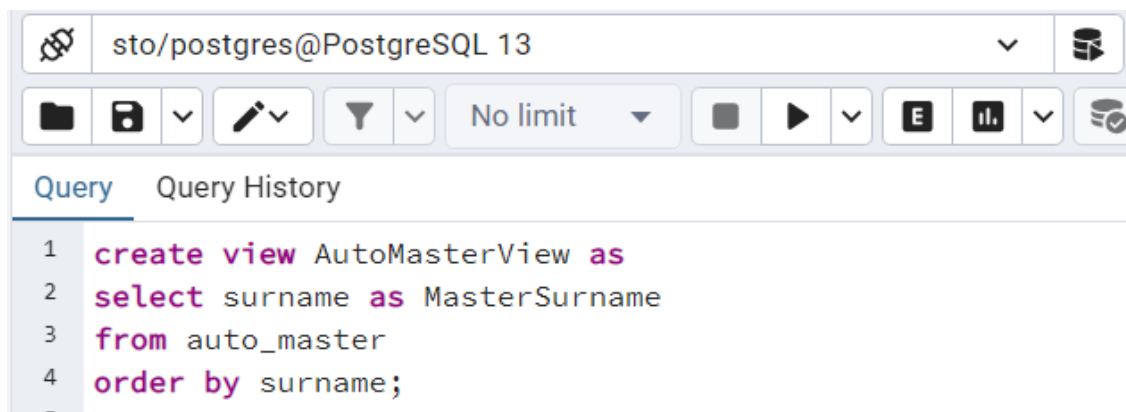
16 select * from vehicle
17 where brand = 'Lada';
18

```

| Data Output Messages Notifications |                    |                        |                                 |                                 |                                      |
|------------------------------------|--------------------|------------------------|---------------------------------|---------------------------------|--------------------------------------|
|                                    | id<br>[PK] integer | customer_id<br>integer | model<br>character varying (20) | brand<br>character varying (20) | reg_number<br>character varying (20) |
| 1                                  | 2                  | 2                      | Granta                          | Lada                            | B555TY136                            |
| 2                                  | 3                  | 5                      | Priora                          | Lada                            | B125YK36                             |

## SQL Представления

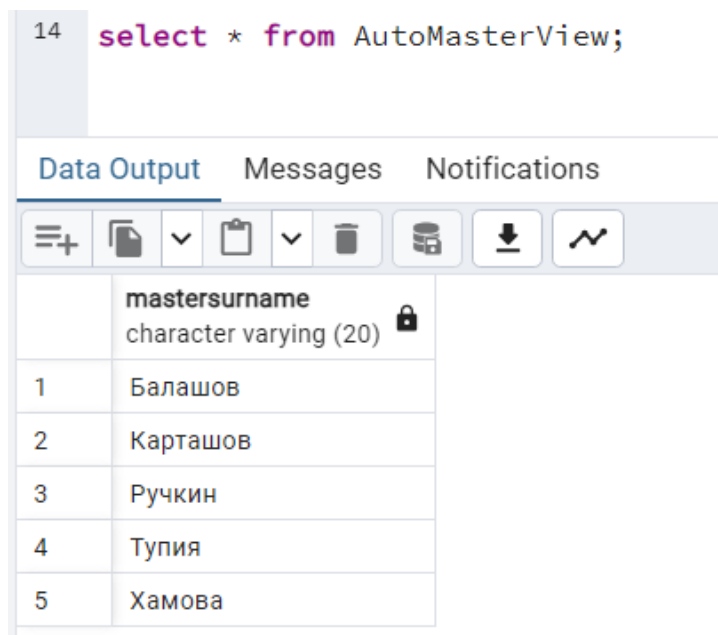
Следующий оператор определяет представление под названием AutoMasterView, базирующийся на таблице auto\_master.



The screenshot shows a PostgreSQL client window titled 'sto/postgres@PostgreSQL 13'. The 'Query' tab is active, displaying the following SQL code:

```
1 create view AutoMasterView as
2 select surname as MasterSurname
3 from auto_master
4 order by surname;
```

Теперь для получения отсортированного списка фамилий авто-мастеров можно использовать созданное представление:



The screenshot shows the same PostgreSQL client window. The 'Query' tab is active, displaying the following SQL code:

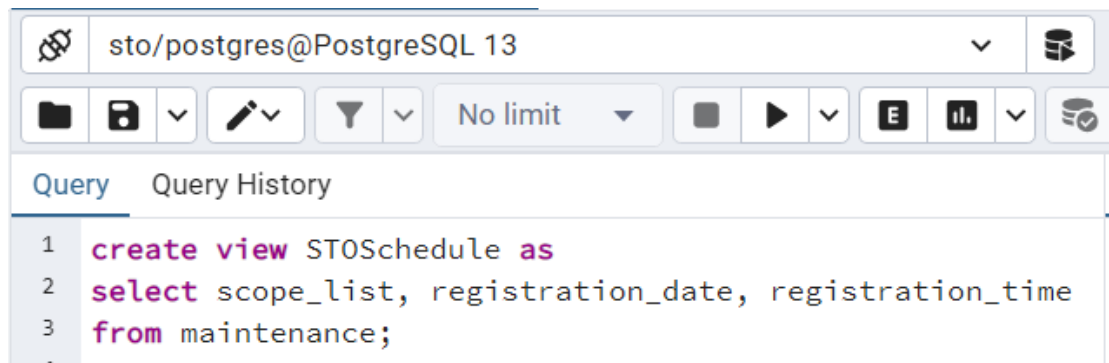
```
14 select * from AutoMasterView;
```

Below the query editor, the 'Data Output' tab is active, showing the results of the query. The results are displayed in a table with two columns: 'mastersurname' (character varying (20)) and an index column. The data is as follows:

|   | mastersurname |
|---|---------------|
| 1 | Балашов       |
| 2 | Карташов      |
| 3 | Ручкин        |
| 4 | Тупия         |
| 5 | Хамова        |

### *Использование представлений для скрытия столбцов и строк*

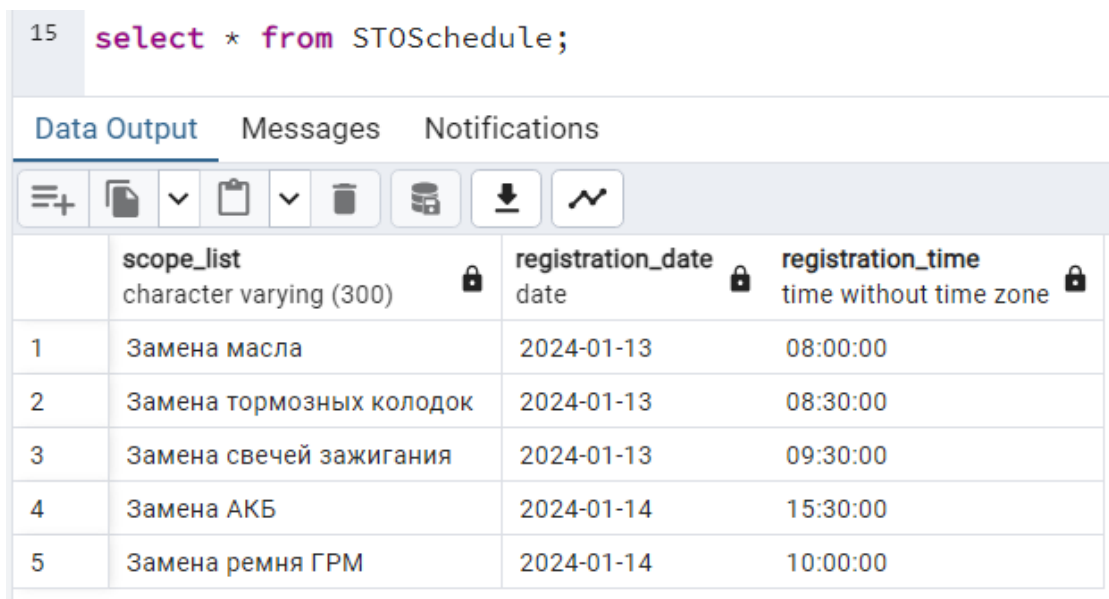
Необходимо получить расписание работ СТО. В таблице maintenance помимо списка работ, количества деталей и времени записи, присутствуют также id транспортного средства и id исполняющего работ, для получения расписания скроем эти столбцы и строки (не будем отображать их в представлении).



The screenshot shows a PostgreSQL client window titled 'sto/postgres@PostgreSQL 13'. The interface includes a toolbar with icons for file operations, query execution, and settings. Below the toolbar, there are tabs for 'Query' and 'Query History'. The 'Query' tab is active, displaying a SQL query:

```
1 create view STOSchedule as
2 select scope_list, registration_date, registration_time
3 from maintenance;
```

Представление STOSchedule будет отображать три столбца: список работ, дата и время.

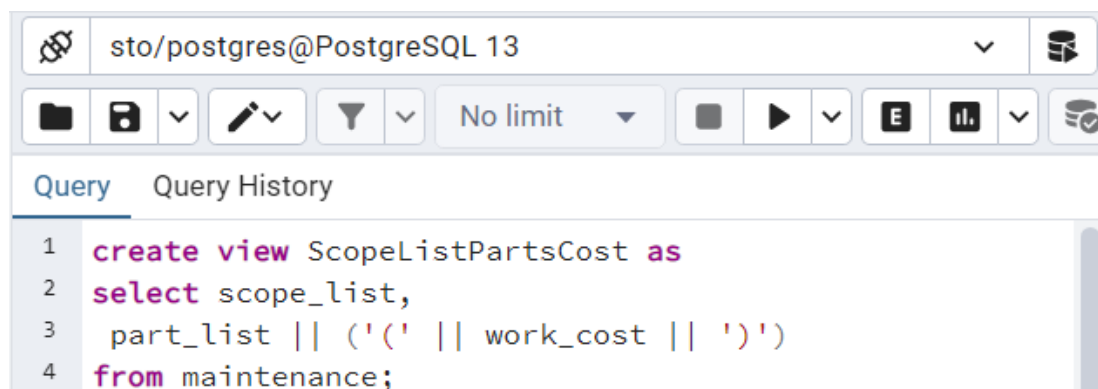


The screenshot shows the same PostgreSQL client window, but now the 'Data Output' tab is active. It displays the result of the query 'select \* from STOSchedule;'. The output is a table with three columns: 'scope\_list', 'registration\_date', and 'registration\_time'. The data is as follows:

|   | scope_list<br>character varying (300) | registration_date<br>date | registration_time<br>time without time zone |
|---|---------------------------------------|---------------------------|---|
| 1 | Замена масла                          | 2024-01-13                | 08:00:00                                    |
| 2 | Замена тормозных колодок              | 2024-01-13                | 08:30:00                                    |
| 3 | Замена свечей зажигания               | 2024-01-13                | 09:30:00                                    |
| 4 | Замена АКБ                            | 2024-01-14                | 15:30:00                                    |
| 5 | Замена ремня ГРМ                      | 2024-01-14                | 10:00:00                                    |

### ***Использование представлений для отображения вычисляемых столбцов***

Необходимо объединить в один столбец список запчастей part\_list и стоимость работ work\_cost. Это можно сделать, создав следующее представление:



The screenshot shows the PostgreSQL client window with the 'Query' tab active. The SQL query is:

```
1 create view ScopeListPartsCost as
2 select scope_list,
3 part_list || ('(' || work_cost || ')')
4 from maintenance;
```

Представление ScopeListPartsCost будет отображать три столбца, один из которых будет состоять из двух:

```
15 select * from ScopeListPartsCost;
16
```

| Data Output Messages Notifications |                                       |  |
|------------------------------------|---------------------------------------|--|
|                                    | scope_list<br>character varying (300) | ?column?<br>text                         |
| 1                                  | Замена масла                          | Моторное масло, масляный фильтр(5000)    |
| 2                                  | Замена тормозных колодок              | Тормозные колодки(3400)                  |
| 3                                  | Замена свечей зажигания               | Свечи зажигания(5000)                    |
| 4                                  | Замена АКБ                            | АКБ(5000)                                |
| 5                                  | Замена ремня ГРМ                      | Ремень ГРМ, Натяжитель, Ролики ГРМ(5000) |

*Использование представлений для скрытия сложного синтаксиса*

Необходимо отобразить в одной таблице список работ, информацию о мастере, а также о машине клиента. Для этого каждый раз при выборке необходимо выполнять запрос с двумя join, или создать представление, реализующее данный сложный запрос:

```
sto/postgres@PostgreSQL 13
```

Query Query History

```
1 create view MaintenanceData as
2 select mt.scope_list, mt.registration_date, mt.registration_time,
3        am.name as master_name, am.surname as master_surname,
4        ve.brand as vehicle_brand, ve.model as vehicle_model, ve.reg_number as vehicle_reg_number
5 from maintenance mt
6 join auto_master am on mt.auto_master_id = am.id
7 join vehicle ve on mt.vehicle_id = ve.id;
```

Представление MaintenanceData будет отображать все необходимые нам данные о записи в СТО.

```
17 select * from MaintenanceData;
18
```

| scope_list<br>character varying (300) | registration_date<br>date | registration_time<br>time without time zone | master_name<br>character varying (20) | master_surname<br>character varying (20) | vehicle_brand<br>character varying (20) | vehicle_model<br>character varying (20) | vehicle_reg_number<br>character varying (20) |
|---------------------------------------|---------------------------|---|---------------------------------------|--|---|---|--|
| 1 Замена АКБ                          | 2024-01-14                | 15:30:00                                    | Иван                                  | Ручкин                                   | Chevrolet                               | Lacetti                                 | H563YK36                                     |
| 2 Замена масла                        | 2024-01-13                | 08:00:00                                    | Иван                                  | Ручкин                                   | Lada                                    | Granta                                  | B555TY136                                    |
| 3 Замена тормозных колодок            | 2024-01-13                | 08:30:00                                    | Василий                               | Тупия                                    | Ford                                    | Focus                                   | Y214KP36                                     |
| 4 Замена ремня ГРМ                    | 2024-01-14                | 10:00:00                                    | Константин                            | Балашов                                  | Mercedes                                | C300                                    | A111AA36                                     |
| 5 Замена свечей зажигания             | 2024-01-13                | 09:30:00                                    | Константин                            | Балашов                                  | Lada                                    | Priora                                  | B125YK36                                     |

Для получения количества записей в СТО (количество строк в таблице maintenance) можно использовать следующий код:

```

16 select count(*)
17 from maintenance;

```

| Data Output |        |   |
|-------------|--------|---|
|             | count  |   |
|             | bigint |   |
| 1           |        | 5 |

## Хранимые процедуры

Создадим процедуру, которая в виде аргументов будет принимать данные записи на СТО и добавлять эти данные в таблицу maintenance.

```

1 create or replace function insert_maintenance(
2     aauto_master_id int,
3     avehicle_id int,
4     ascope_list varchar(300),
5     aregistration_date date,
6     aregistration_time time,
7     awork_cost int,
8     apart_list varchar(300)
9 )
10 returns void as $$
11 begin
12     insert into maintenance (auto_master_id, vehicle_id, scope_list, registration_date,
13                             registration_time, work_cost, part_list)
14         values(aauto_master_id, avehicle_id, ascope_list, aregistration_date,
15              aregistration_time, awork_cost, apart_list);
16 end;
17 $$ language plpgsql;
18

```

## Пример вызова процедуры:

```

19
20 select insert_maintenance(3, 4, 'Замена фар', '2024-01-21', '09:00:00', 2500, 'Фары');
21
22

```

## Результат выполнения процедуры:

```

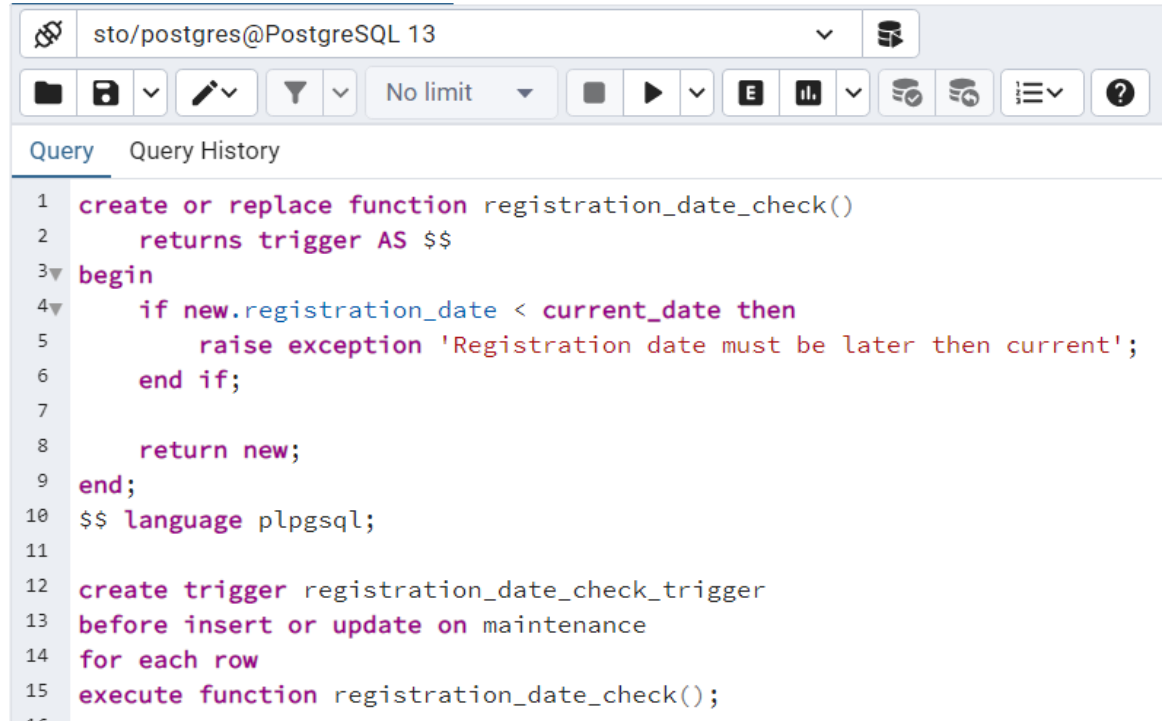
24 select * from MaintenanceData;

```

|   | scope_list<br>character varying (300) | registration_date<br>date | registration_time<br>time without time zone | master_name<br>character varying (20) | master_surname<br>character varying (20) | vehicle_brand<br>character varying (20) | vehicle_model<br>character varying (20) | vehicle_reg_number<br>character varying (20) |
|---|---------------------------------------|---------------------------|---|---------------------------------------|--|---|---|--|
| 1 | Замена АКБ                            | 2024-01-14                | 15:30:00                                    | Иван                                  | Ручкин                                   | Chevrolet                               | Lacetti                                 | H563YK36                                     |
| 2 | Замена масла                          | 2024-01-13                | 08:00:00                                    | Иван                                  | Ручкин                                   | Lada                                    | Granta                                  | B555TY136                                    |
| 3 | Замена тормозных колодок              | 2024-01-13                | 08:30:00                                    | Василий                               | Тупия                                    | Ford                                    | Focus                                   | Y214KP36                                     |
| 4 | Замена фар                            | 2024-01-21                | 09:00:00                                    | Григорий                              | Карташов                                 | Chevrolet                               | Lacetti                                 | H563YK36                                     |
| 5 | Замена ремня ГРМ                      | 2024-01-14                | 10:00:00                                    | Константин                            | Балашов                                  | Mercedes                                | C300                                    | A111AA36                                     |
| 6 | Замена свечей зажигания               | 2024-01-13                | 09:30:00                                    | Константин                            | Балашов                                  | Lada                                    | Priora                                  | B125YK36                                     |

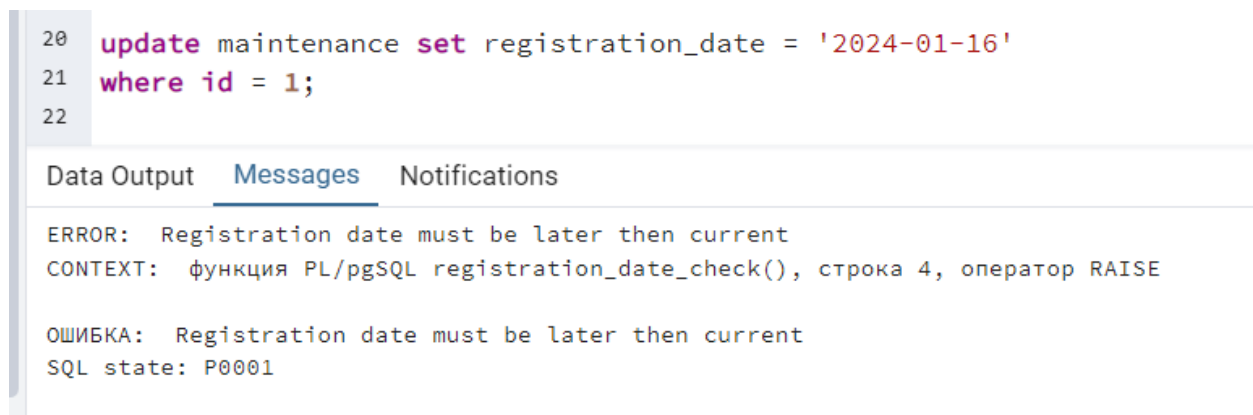
## Триггеры

### *Использование триггеров для проверки допустимости вводимых данных*



```
1 create or replace function registration_date_check()
2     returns trigger AS $$
3 begin
4     if new.registration_date < current_date then
5         raise exception 'Registration date must be later then current';
6     end if;
7
8     return new;
9 end;
10 $$ language plpgsql;
11
12 create trigger registration_date_check_trigger
13 before insert or update on maintenance
14 for each row
15 execute function registration_date_check();
16
```

Триггер `registration_date_check_trigger` вызывает функцию `registration_date_check()` при операции добавления или изменения таблицы `maintenance`, которая проверяет вводимую дату так, чтобы она не была раньше текущей. В случае отрицательного результата функция вызывает исключение:



```
20 update maintenance set registration_date = '2024-01-16'
21 where id = 1;
22
```

Data Output   **Messages**   Notifications

```
ERROR:  Registration date must be later then current
CONTEXT:  функция PL/pgSQL registration_date_check(), строка 4, оператор RAISE

ОШИБКА:  Registration date must be later then current
SQL state: P0001
```

## Использование триггеров для присвоения значений по умолчанию

```
sto/postgres@PostgreSQL 13
Query Query History
1 create or replace function maintenance_default_values()
2 returns trigger as $$
3 begin
4     if new.scope_list is null then
5         new.scope_list := 'Нет данных о списке работ';
6     end if;
7
8     if new.part_list is null then
9         new.part_list := 'Нет данных о списке запчастей';
10    end if;
11
12    return new;
13 end;
14 $$ language plpgsql;
15
16 create trigger maintenance_default_values_trigger
17 before insert on maintenance
18 for each row
19 execute function maintenance_default_values();
20
```

Триггер `maintenance_default_values_trigger` вызывает функцию `maintenance_default_values()` при операции добавления в таблицу `maintenance`, которая проверяет вводимые данные `scope_list` и `part_list` на `null`, в случае, если вставляемые в таблицу данные равны `null` – добавляет вместо `null` значение по умолчанию:

```
21 select insert_maintenance(5, 5, NULL, '2024-04-06', '09:00:00', 3500, NULL);
22 select * from maintenance;
```

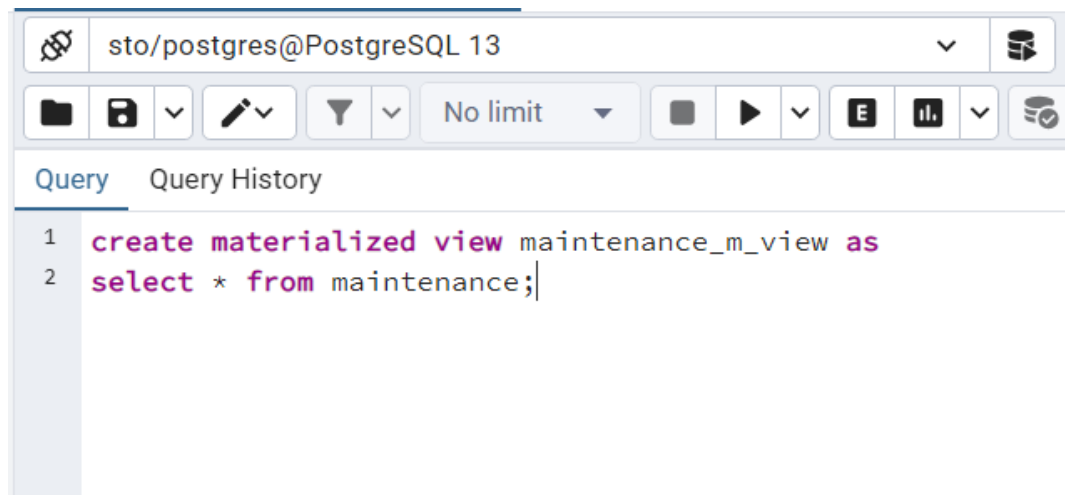
| id | auto_master_id | vehicle_id | scope_list                | registration_date | registration_time | work_cost | part_list                          |
|----|----------------|------------|---------------------------|-------------------|-------------------|-----------|------------------------------------|
| 4  | 5              | 5          | Замена ремня ГРМ          | 2024-01-14        | 10:00:00          | 5000      | Ремень ГРМ, Натяжитель, Ролики ГРМ |
| 5  | 6              | 3          | Замена фар                | 2024-01-21        | 09:00:00          | 2500      | Фары                               |
| 6  | 1              | 1          | Замена масла              | 2024-01-16        | 08:00:00          | 5000      | Моторное масло, масляный фильтр    |
| 7  | 7              | 5          | Нет данных о списке работ | 2024-04-06        | 09:00:00          | 3500      | Нет данных о списке запчастей      |

Total rows: 7 of 7    Query complete 00:00:00.038    Rows selected: 1

## Триггер, обновляющий представление

Для создания такого триггера для начала создадим материализованное представление, данные в котором не обновляются при изменении таблиц, на основе которых они созданы:

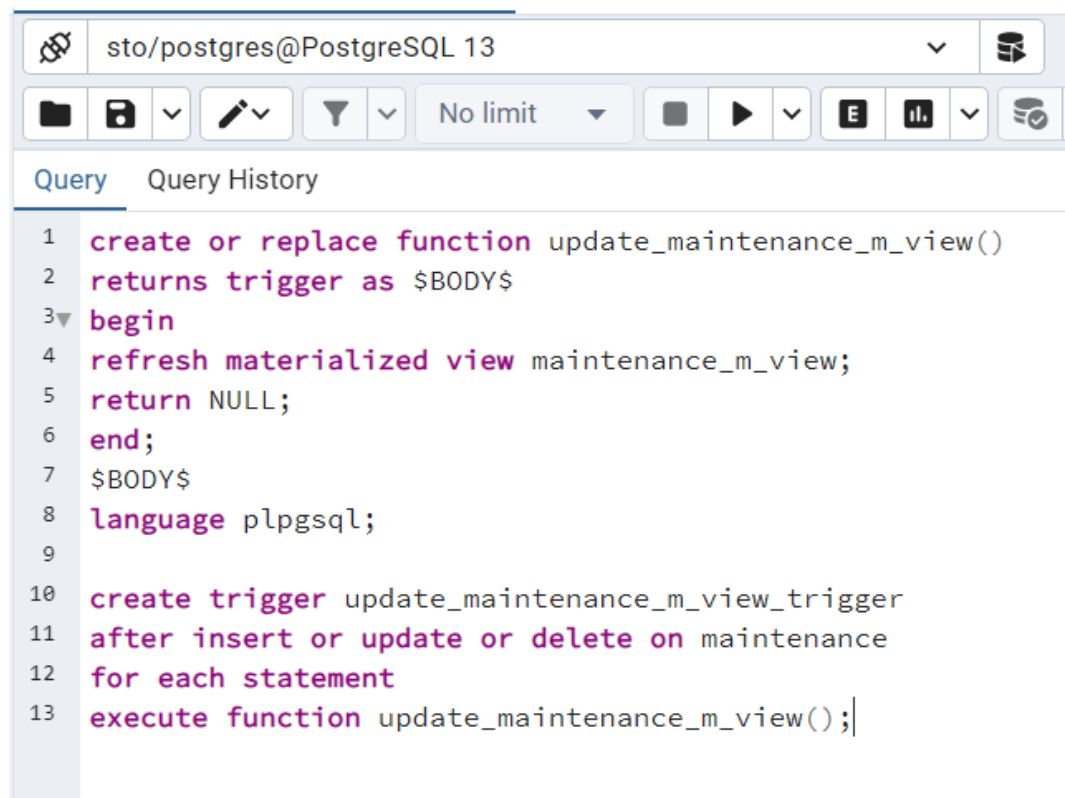




The screenshot shows a PostgreSQL query editor interface. At the top, the connection is 'sto/postgres@PostgreSQL 13'. Below the connection bar is a toolbar with icons for file operations, query execution, and other database functions. The main area is divided into 'Query' and 'Query History' tabs. The 'Query' tab is active, displaying a SQL query to create a materialized view named 'maintenance\_m\_view' from the 'maintenance' table.

```
1 create materialized view maintenance_m_view as
2 select * from maintenance;
```

Для того, чтобы при выполнении операций SELECT, UPDATE и DELETE на таблице maintenance данные обновлялись в представлении maintenance\_m\_view, создадим триггер update\_maintenance\_m\_view\_trigger, который вызывает функцию update\_maintenance\_m\_view():



The screenshot shows the same PostgreSQL query editor interface. The 'Query' tab is active, displaying a SQL query to create a function 'update\_maintenance\_m\_view()' and a trigger 'update\_maintenance\_m\_view\_trigger' that fires on INSERT, UPDATE, or DELETE operations on the 'maintenance' table.

```
1 create or replace function update_maintenance_m_view()
2 returns trigger as $BODY$
3 begin
4 refresh materialized view maintenance_m_view;
5 return NULL;
6 end;
7 $BODY$
8 language plpgsql;
9
10 create trigger update_maintenance_m_view_trigger
11 after insert or update or delete on maintenance
12 for each statement
13 execute function update_maintenance_m_view();
```

## Триггер, обеспечивающий ссылочную целостность

Для обеспечения ссылочной целостности создадим триггер `check_customer_vehicle_id_trigger`, который срабатывает при изменении или добавлении новой строки в `maintenance`, и вызывает процедуру `check_customer_vehicle_id`. Данная процедура проверяет наличие `id` авто-мастера и транспортного средства в соответствующих таблицах:

```
Query Query History
1 create or replace function check_customer_vehicle_id()
2 returns trigger as $$
3 begin
4     if new.auto_master_id is not null and not exists (select 1 from auto_master where id = new.auto_master_id) then
5         raise exception 'Отсутствует auto_master с указанным id';
6     end if;
7
8     if new.vehicle_id is not null and not exists (select 1 from vehicle where id = new.vehicle_id) then
9         raise exception 'Отсутствует vehicle с указанным id';
10    end if;
11
12    return new;
13 end;
14 $$ language plpgsql;
15
16 create trigger check_customer_vehicle_id_trigger
17 before insert or update on maintenance
18 for each row
19 execute function check_customer_vehicle_id();
20
```

В случае отсутствия в них указанного `id` выбрасывается исключение:

```
28
21 select insert_maintenance(8,9,NULL, '2024-04-08', '09:00:00', 3700, NULL);

```

| Data Output  | Messages | Notifications |
|--|----------|---------------|
| <p>ERROR: Отсутствует auto_master с указанным id<br/>CONTEXT: функция PL/pgSQL check_customer_vehicle_id(), строка 4, оператор RAISE<br/>SQL-оператор: "insert into maintenance (auto_master_id, vehicle_id, scope_list, registration_date, registration_time, work_cost, part_list)<br/>values(aauto_master_id, avehicle_id, ascope_list, aregistration_date, aregistration_time, awork_cost, apart_list)"<br/>функция PL/pgSQL insert_maintenance(integer,integer,character varying,date,time without time zone,integer,character varying), строка 3, оператор SQL-оператор</p> <p>ОШИБКА: Отсутствует auto_master с указанным id<br/>SQL state: P0001</p> |          |               |

## Модуль TABLEFUNC

Создадим процедуру, которая может быть использована для создания сводной таблицы значений стоимости работ, которые осуществляются для определенных автомобилей:

```

1 create or replace function calculate_average_work_cost_by_vehicle()
2 returns table (vehicle_reg_number varchar(20),
3               avg_work_cost numeric)
4 as $$
5 begin
6     return query
7     select vc.reg_number, avg(mt.work_cost) as avg_work_cost
8     from maintenance mt
9     join vehicle vc on mt.vehicle_id = vc.id
10    group by vc.reg_number;
11 end;
12 $$ language plpgsql;
13

```

Вывод процедуры:

```

14
15 select * from calculate_average_work_cost_by_vehicle();
16

```

Data Output
Messages
Notifications

|   | vehicle_reg_number<br>character varying | avg_work_cost<br>numeric |
|---|---|--------------------------|
| 1 | H563YK36                                | 3750.0000000000000000    |
| 2 | Y214KP36                                | 3400.0000000000000000    |
| 3 | A111AA36                                | 5000.0000000000000000    |
| 4 | B555TY136                               | 5000.0000000000000000    |
| 5 | B125YK36                                | 5000.0000000000000000    |

## Функция ROW\_NUMBER

```

13
14 select row_number() over (order by surname) as num, surname
15 from auto_master;

```

Data Output
Messages
Notifications

|   | num<br>bigint | surname<br>character varying (20) |
|---|---------------|-----------------------------------|
| 1 | 1             | Балашов                           |
| 2 | 2             | Карташов                          |
| 3 | 3             | Ручкин                            |
| 4 | 4             | Тупия                             |
| 5 | 5             | Хамова                            |

Данная функция используется для нумерации строк таблицы auto\_master в порядке сортировки по столбцу surname.

Также данную функцию можно применять для ограничения количества обрабатываемых строк. Так, данный запрос извлекает первые три строки данных:

```
12 select * from (select row_number() over (order by surname) as num, surname
13 from auto_master) subquery
14 where num <=3;
15
16
```

Data OutputMessagesNotifications

|   | num<br>bigint | surname<br>character varying (20) |
|---|---------------|-----------------------------------|
| 1 | 1             | Балашов                           |
| 2 | 2             | Карташов                          |
| 3 | 3             | Ручкин                            |

В случае, если при группировке необходимо нумеровать строки для каждой группы отдельно, можно использовать оператор partition by:

```
11 select row_number() over (partition by qualification) num,
12 qualification, count(*)
13 from auto_master
14 group by qualification;
15
16
```

| Data Output |               | Messages                                | Notifications   |
|-------------|---------------|---|-----------------|
|             |               |   |                 |
|             | num<br>bigint | qualification<br>character varying (20) | count<br>bigint |
| 1           | 1             | Механик                                 | 2               |
| 2           | 1             | Приемщик                                | 1               |
| 3           | 1             | Ст.механик                              | 1               |
| 4           | 1             | Электрик                                | 1               |

## Функция COALESCE

Функция возвращает значение первого аргумента, значение которого не равно NULL.

```
14 select scope_list, work_cost, coalesce(work_cost, 0) coalesce_work_cost
15 from maintenance;
16
```

Data Output Messages Notifications

|   | scope_list<br>character varying (300) | work_cost<br>integer | coalesce_work_cost<br>integer |
|---|---------------------------------------|----------------------|-------------------------------|
| 1 | Замена свечей зажигания               | 5000                 | 5000                          |
| 2 | Замена АКБ                            | 5000                 | 5000                          |
| 3 | Замена ремня ГРМ                      | 5000                 | 5000                          |
| 4 | Замена фар                            | 2500                 | 2500                          |
| 5 | Замена масла                          | 5000                 | 5000                          |
| 6 | Замена тормозных колодок              | [null]               | 0                             |

В данном случае функция используется для замены всех значений NULL в столбце work\_cost на 0.

## Числовые функции

```
16 SELECT ABS(100) X1, ABS(-100) X2, ABS(-100.2) X3;
```

Data Output Messages Notifications

|   | x1<br>integer | x2<br>integer | x3<br>numeric |
|---|---------------|---------------|---------------|
| 1 | 100           | 100           | 100.2         |

```

15 SELECT FLOOR(100.22) X1, FLOOR(-100.22) X2,
16 FLOOR(100.99) X3, FLOOR(100.01) X4;

```

Data Output Messages Notifications

|   | x1<br>numeric | x2<br>numeric | x3<br>numeric | x4<br>numeric |
|---|---------------|---------------|---------------|---------------|
| 1 | 100           | -101          | 100           | 100           |

```

15 SELECT TRUNC(100.25678) X1, TRUNC(-100.25678) X2,
16 TRUNC(100.99) X3, TRUNC(100.25678, 2) X4;

```

Data Output Messages Notifications

|   | x1<br>numeric | x2<br>numeric | x3<br>numeric | x4<br>numeric |
|---|---------------|---------------|---------------|---------------|
| 1 | 100           | -100          | 100           | 100.25        |

```

15 SELECT ROUND(100.25678) X1, ROUND(100.5) X2,
16 ROUND(100.99) X3, ROUND(100.25678, 2) X4;

```

Data Output Messages Notifications

|   | x1<br>numeric | x2<br>numeric | x3<br>numeric | x4<br>numeric |
|---|---------------|---------------|---------------|---------------|
| 1 | 100           | 101           | 101           | 100.26        |

```

16 SELECT SIGN(100.22) X1, SIGN(-100.22) X2, SIGN(0) X3;

```

Data Output Messages Notifications

|   | x1<br>numeric | x2<br>numeric | x3<br>double precision |
|---|---------------|---------------|------------------------|
| 1 | 1             | -1            | 0                      |

```
16 SELECT MOD(10, 3) X1, MOD(10, 2) X2, MOD(100, 98) X3;
```

Data Output Messages Notifications

|   | x1<br>integer | x2<br>integer | x3<br>integer |
|---|---------------|---------------|---------------|
| 1 | 1             | 0             | 2             |

```
16 SELECT POWER(10, 2) X1, POWER(100, 0.5) X2,  
17 POWER(1000, 0.33333333) X3, POWER(1000, -0.33333333) X4;
```

Data Output Messages Notifications

|   | x1<br>double precision | x2<br>numeric       | x3<br>numeric      | x4<br>numeric       |
|---|------------------------|---------------------|--------------------|---------------------|
| 1 | 100                    | 10.0000000000000000 | 9.9999997697414934 | 0.10000000023025851 |

```
16 SELECT SQRT(100) X;
```

Data Output Messages Notifications

|   | x<br>double precision |
|---|-----------------------|
| 1 | 10                    |

```
16 SELECT EXP(1) X1, LN(1) X2, LN(EXP(2)) X3;
```

Data Output Messages Notifications

|   | x1<br>double precision | x2<br>double precision | x3<br>double precision |
|---|------------------------|------------------------|------------------------|
| 1 | 2.718281828459045      | 0                      | 2                      |










16

SELECT LOG(2, 8) X1, LOG(10, 100) X2;

Data Output

Messages

Notifications



|   | x1<br>numeric      | x2<br>numeric      |
|---|--------------------|--------------------|
| 1 | 3.0000000000000000 | 2.0000000000000000 |

## Тригонометрические функции

16

SELECT

SIN(0)

X1,

COS(0)

X2,

TAN(0)

X3,

COT(0)

;

Data Output

Messages

Notifications

</

## Строковые и символьные функции

16 **SELECT** CONCAT('У ', ' попа ', ' была ', ' собака') X1,  
17 CONCAT('Test', NULL) X2,  
18 CONCAT(NULL, 'Test') X3,  
19 CONCAT(NULL, NULL) X4;

Data OutputMessagesNotifications



```

16 SELECT CONCAT('У ', ' попа ', ' была ', ' собака') X1,
17 'У попа ' || ' была собака' X2;

```

Data Output Messages Notifications

|   | x1<br>text         | x2<br>text         |
|---|--------------------|--------------------|
| 1 | У попа была собака | У попа была собака |

```

16 SELECT LOWER('TeXt DATA') X;

```

Data Output Messages Notifications

|   | x<br>text |
|---|-----------|
| 1 | text data |

```

16 SELECT UPPER('TeXt DATA') X;

```

Data Output Messages Notifications

|   | x<br>text |
|---|-----------|
| 1 | TEXT DATA |

```

16 SELECT INITCAP('Иванов петр сидорович') X;

```

Data Output Messages Notifications

|   | x<br>text             |
|---|-----------------------|
| 1 | Иванов Петр Сидорович |

```

16 SELECT LTRIM('TeXt DATA') X1,
17 LTRIM(' _ # TeXt DATA', ' #_') X2,
18 LTRIM(' 1234567890 TeXt DATA', ' 1234567890') X3
19 UNION ALL
20 SELECT RTRIM('TeXt DATA') X1,
21 LTRIM(' _ # TeXt DATA', ' #_') X2,
22 LTRIM(' 1234567890 TeXt DATA', ' 1234567890') X3

```

Data Output Messages Notifications

|   | <div> <div>≡+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> </div> |   |   |
|---|---|---|---|
|   | <div> <div>x1</div> <div>text</div> <div>🔒</div> </div>   | <div> <div>x2</div> <div>text</div> <div>🔒</div> </div> | <div> <div>x3</div> <div>text</div> <div>🔒</div> </div> |
| 1 | TeXt DATA   | TeXt DATA   | TeXt DATA   |
| 2 | TeXt DATA   | TeXt DATA   | TeXt DATA   |

```

16 CREATE OR REPLACE FUNCTION is_password_correct(
17 password IN char)
18 RETURNS int AS $is_password_correct$
19 BEGIN
20 IF TRANSLATE(password, '0123456789', '*') = password THEN
21 RAISE WARNING 'Пароль должен содержать хотя бы одну цифру!';
22 RETURN 0;
23 END IF;
24 RAISE INFO 'Корректный пароль!';
25 RETURN 1;
26 END;
27 $is_password_correct$ LANGUAGE plpgsql;
28
29 select * from is_password_correct('vadf');

```

Data Output Messages Notifications

|   | <div> <div>≡+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> </div> |  |
|---|---|--|
|   | <div> <div>is_password_correct</div> <div>integer</div> <div>🔒</div> </div>   |  |
| 1 | 0   |  |

Функции работы с датой и временем

16

SELECT NOW();

Data Output

Messages

Notifications

≡+

▼

▼

|   | now<br>timestamp with time zone |  |
|---|---------------------------------|--|
| 1 | 2024-04-06 01:26:23.836357+03   |  |

16

SELECT NOW() D1,

17

NOW() + JUSTIFY\_INTERVAL('30 DAYS 1 HOUR 2 MINUTE') D2,

18

NOW() - JUSTIFY\_INTERVAL('30 DAYS 1 HOUR 2 MINUTE') D3;

Data Output

Messages

Notifications

≡+

▼

▼

|   | d1<br>timestamp with time zone |  | d2<br>timestamp with time zone |  | d3<br>timestamp with time zone |  |
|---|--------------------------------|--|--------------------------------|--|--------------------------------|--|
| 1 | 2024-04-06 01:26:49.211541+03  |  | 2024-05-06 02:28:49.211541+03  |  | 2024-03-06 00:24:49.211541+03  |  |

16

SELECT

17

DATE\_TRUNC('HOUR', NOW()) D1,

18

DATE\_TRUNC('DAY', NOW()) D2,

19

DATE\_TRUNC('MONTH', NOW()) D3;

Data Output

Messages

Notifications

≡+

▼

▼

|   | d1<br>timestamp with time zone |  | d2<br>timestamp with time zone |  | d3<br>timestamp with time zone |  |
|---|--------------------------------|--|--------------------------------|--|--------------------------------|--|
| 1 | 2024-04-06 01:00:00+03         |  | 2024-04-06 00:00:00+03         |  | 2024-04-01 00:00:00+03         |  |

```
16 SELECT
17 DATE_TRUNC('MONTH', NOW()) D1,
18 DATE_TRUNC('MONTH', NOW())
19 + JUSTIFY_INTERVAL('1 MONTH - 1 DAY') D2;
```

Data Output Messages Notifications

|   | d1<br>timestamp with time zone | d2<br>timestamp with time zone |
|---|--------------------------------|--------------------------------|
| 1 | 2024-04-01 00:00:00+03         | 2024-04-30 00:00:00+03         |

```
16 SELECT
17 CURRENT_DATE D1,
18 AGE(MAKE_TIMESTAMP(2013, 7, 15, 8, 15, 23.5)) D2,
19 AGE(MAKE_DATE(2016, 3, 3),
20 MAKE_TIMESTAMP(2013, 7, 15, 8, 15, 23.5)) D3;
```

Data Output Messages Notifications

|   | d1<br>date | d2<br>interval                     | d3<br>interval                    |
|---|------------|------------------------------------|-----------------------------------|
| 1 | 2024-04-06 | 10 years 8 mons 21 days 15:44:36.5 | 2 years 7 mons 18 days 15:44:36.5 |

```
16 SELECT
17 NOW() D1,
18 EXTRACT(MONTH FROM NOW()) D2,
19 EXTRACT(YEAR FROM NOW()) D3,
20 EXTRACT(MINUTE FROM NOW()) D4;
```

Data Output Messages Notifications

|   | d1<br>timestamp with time zone | d2<br>double precision | d3<br>double precision | d4<br>double precision |
|---|--------------------------------|------------------------|------------------------|------------------------|
| 1 | 2024-04-06 01:28:44.217833+03  | 4                      | 2024                   | 28                     |

```

16 SELECT
17 NOW() D1,
18 EXTRACT(MONTH FROM AGE(DATE_TRUNC('MONTH', NOW()),
19 DATE_TRUNC('MONTH', MAKE_DATE(2016, 3, 3)))) D2;

```

Data Output Messages Notifications

|   | d1<br>timestamp with time zone | d2<br>double precision |
|---|--------------------------------|------------------------|
| 1 | 2024-04-06 01:29:15.293465+03  | 1                      |

```

16 SELECT
17 TO_DATE('05 Dec 2000', 'DD Mon YYYY') D1,
18 TO_DATE('15.12.2000', 'dd.mm.yy') D2;

```

Data Output Messages Notifications

|   | d1<br>date | d2<br>date |
|---|------------|------------|
| 1 | 2000-12-05 | 2000-12-15 |

```

16 SELECT NOW() D1,
17 TO_CHAR(NOW(), 'DD.MM.YY HH24:MI') D2;

```

Data Output Messages Notifications

|   | d1<br>timestamp with time zone | d2<br>text     |
|---|--------------------------------|----------------|
| 1 | 2024-04-06 01:30:16.521088+03  | 06.04.24 01:30 |