

SQL-представления.

Использование представлений для скрывтия столбцов.

```
masterskaya=# create View OrderName AS select name, status, cost fr
om repair_order ORDER BY cost;
CREATE VIEW
masterskaya=# select * from OrderName;
  name                | status  | cost
-----+-----+-----
Repair Xiaomi Telephone | at work |  500
Repair Honor laptot   | at work | 1000
Repair office Computer | completed | 1000
Repair Gamer Computer  | completed | 2500
(4 rows)
```

Следующий оператор определяет представление, содержащее название, цену, начало и конец работы у клиент под номером 2.

```
masterskaya=# create view Order2 AS select name, cost, date_created, date_completed from repair_order where client_id='2';
CREATE VIEW
masterskaya=# select * from Order2;
  name                | cost | date_created | date_completed
-----+-----+-----+-----
Repair Xiaomi Telephone |  500 | 2022-10-10   | 2022-10-20
Repair office Computer  | 1000 | 2022-08-20   | 2022-08-30
(2 rows)
```

Использование представления для отображения вычисляемых столбцов

```
masterskaya=# create view Client2 AS select name, ((' || phone || ')) || email AS D
ate from client ORDER BY name;
CREATE VIEW
masterskaya=# select * from Client2;
  name                | date
-----+-----
Danil Nikitovich     | (+79204356633)danilian@mail.ru
Svetlana Pavlovna    | (79115672011)svetlana12521@mail.ru
Veronika Petrovna    | (+79205642296) veronika1243@gmail.com
(3 rows)
masterskaya=#
```

Использование представления для скрывтия сложного синтаксиса.

```
masterskaya=# CREATE VIEW contractor_typeWork AS
SELECT C.Name AS Contractor, TW.Name AS Type_of_work
FROM Contractor C
JOIN repair_order RO
ON C.id = RO.contractor_id
JOIN Type_of_work TW
ON RO.type_of_work_id = TW.id;
CREATE VIEW
masterskaya=# select * from contractor_typework
masterskaya-# ;
  contractor          | type_of_work
-----+-----
Ivan Ivanov          | laptop_repair
Ivan Ivanov          | telephone_repair
Misha Mihailovich    | computer_repair
Ivan Ivanov          | computer_repair
(4 rows)
masterskaya=#
```

Хранимая процедура.

```
masterskaya=# CREATE OR REPLACE FUNCTION NewOrder (
newname IN text,
neworder OUT text,
newcost OUT text,
newclient OUT char,
newcontractor OUT char)
AS $NewOrder$
DECLARE new_record RECORD;
BEGIN
for new_record in select type_of_work.type_name, repair_order.name, repair_order.cost, contractor.contractor_name, client.client_name FROM repair_order JOIN type_of_work ON type_of_work.id = type_of_work.id JOIN client on repair_order.client_id = client.id JOIN contractor on contractor_id=contractor.id WHERE type_of_work.type_name = newname
LOOP
neworder := new_record.type_name;
newcost := new_record.cost;
newclient := new_record.client_name;
newcontractor := new_record.contractor_name;
RAISE NOTICE 'Type of work % from % for % with name % is costs %', newname, newclient, newcontractor, neworder, newcost;
END LOOP;
END;
$NewOrder$ LANGUAGE plpgsql;
CREATE FUNCTION
masterskaya=# select NewOrder('computer_repair');
NOTICE: Type of work computer_repair from Danil Nikitovich for Misha Mihailovich with name computer_repair is costs 2500
NOTICE: Type of work computer_repair from Svetlana Pavlovna for Ivan Ivanov with name computer_repair is costs 1000
neworder
(1 row)
(1 row)
masterskaya=# select NewOrder('computer_repair');
```

Использование триггеров для проверки допустимости вводимых данных.

```
masterskaya=# CREATE OR REPLACE FUNCTION new_client() RETURNS trigger AS
$new_client$
BEGIN
IF EXISTS (SELECT * FROM client WHERE phone = NEW.phone) THEN
RAISE EXCEPTION 'There is already such a phone number %', NEW.phone;
END IF;
RETURN NEW;
END;
$new_client$ LANGUAGE plpgsql;
CREATE FUNCTION
masterskaya=#
```

```
masterskaya=# CREATE TRIGGER new_client
BEFORE INSERT ON client
FOR EACH ROW EXECUTE FUNCTION new_client();
CREATE TRIGGER
masterskaya=#
```

```
masterskaya=# select * from client;
 id | client_name | phone | email
----+-----+-----+-----
 1 | Veronika Petrovna | +79205642296 | veronika1243@gmail.com
 3 | Danil Nikitovich | +79204356633 | danilian@mail.ru
 2 | Svetlana Pavlovna | 79115672011 | svetlana12521@mail.ru
 4 | Veronika Petrova | +79256642296 | veroika1243@gmail.com
(4 rows)

masterskaya=# insert into client (id,client_name,phone,email) values ('5','Veronika Petrova','+79256642296','veroika1243@gmail.com');
ERROR: There is already such a phone number +79256642296
CONTEXT: PL/pgSQL function new_client() line 4 at RAISE
masterskaya=#
```

получение имени текущей базы данных:

```
masterskaya=# SELECT * FROM
information_schema.information_schema_catalog_name;
 catalog_name

masterskaya
(1 row)

masterskaya=#
```

получение списка ограничений:

```
You are now connected to database masterskaya as user postgres .
masterskaya=# SELECT * FROM information_schema.table_constraints;
```

constraint_catalog	constraint_schema	constraint_name	table_catalog	table_schema	table_name	constraint_type
masterskaya	pg_catalog	pg_proc_oid_index	masterskaya	pg_catalog	pg_proc	PRIMARY KEY
masterskaya	pg_catalog	pg_proc_proname_args_nsp_index	masterskaya	pg_catalog	pg_proc	UNIQUE
masterskaya	pg_catalog	pg_type_oid_index	masterskaya	pg_catalog	pg_type	PRIMARY KEY
masterskaya	pg_catalog	pg_type_typname_nsp_index	masterskaya	pg_catalog	pg_type	UNIQUE
masterskaya	pg_catalog	pg_attribute_relid_attnam_index	masterskaya	pg_catalog	pg_attribute	UNIQUE
masterskaya	pg_catalog	pg_attribute_relid_attnum_index	masterskaya	pg_catalog	pg_attribute	PRIMARY KEY
masterskaya	pg_catalog	pg_class_oid_index	masterskaya	pg_catalog	pg_class	PRIMARY KEY
masterskaya	pg_catalog	pg_class_relname_nsp_index	masterskaya	pg_catalog	pg_class	UNIQUE
masterskaya	pg_catalog	pg_attrdef_adrelid_adnum_index	masterskaya	pg_catalog	pg_attrdef	UNIQUE
masterskaya	pg_catalog	pg_attrdef_oid_index	masterskaya	pg_catalog	pg_attrdef	PRIMARY KEY
masterskaya	pg_catalog	pg_constraint_conrelid_contypid_conname_index	masterskaya	pg_catalog	pg_constraint	UNIQUE
masterskaya	pg_catalog	pg_constraint_oid_index	masterskaya	pg_catalog	pg_constraint	PRIMARY KEY
masterskaya	pg_catalog	pg_inherits_relid_seqno_index	masterskaya	pg_catalog	pg_inherits	PRIMARY KEY
masterskaya	pg_catalog	pg_index_indexrelid_index	masterskaya	pg_catalog	pg_index	PRIMARY KEY

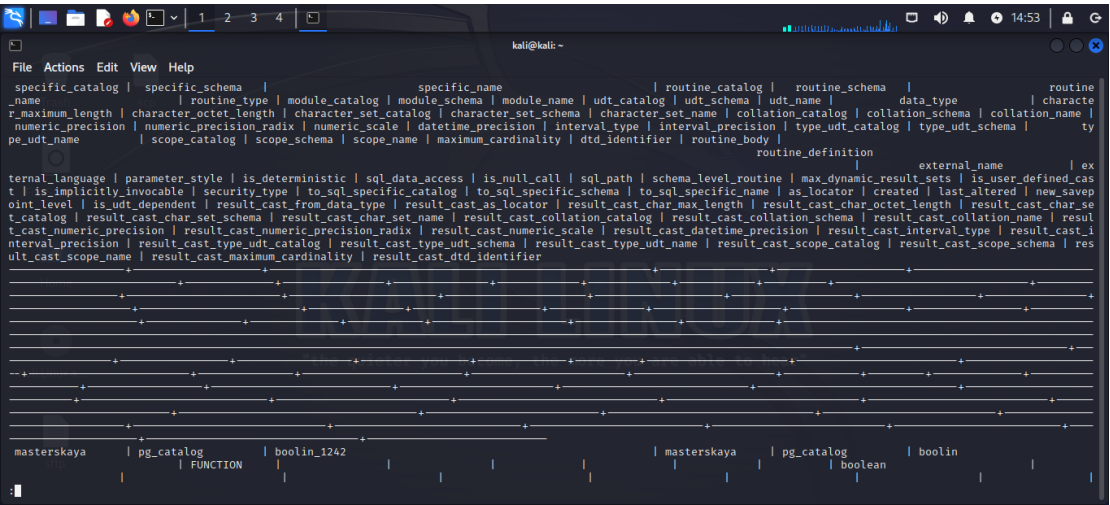
получение списка внешних ключей:

```
masterskaya=# SELECT * FROM
information_schema.referential_constraints;
```

constraint_catalog	constraint_schema	constraint_name	unique_constraint_catalog	unique_constraint_schema	unique_constraint_name	match_opt
masterskaya	public	repair_order_client_id_fkey	masterskaya	public	client_pkey	NONE
masterskaya	public	repair_order_contractor_id_fkey	masterskaya	public	contractor_pkey	NONE
masterskaya	public	repair_order_type_of_work_id_fkey	masterskaya	public	type_of_work_pkey	NONE

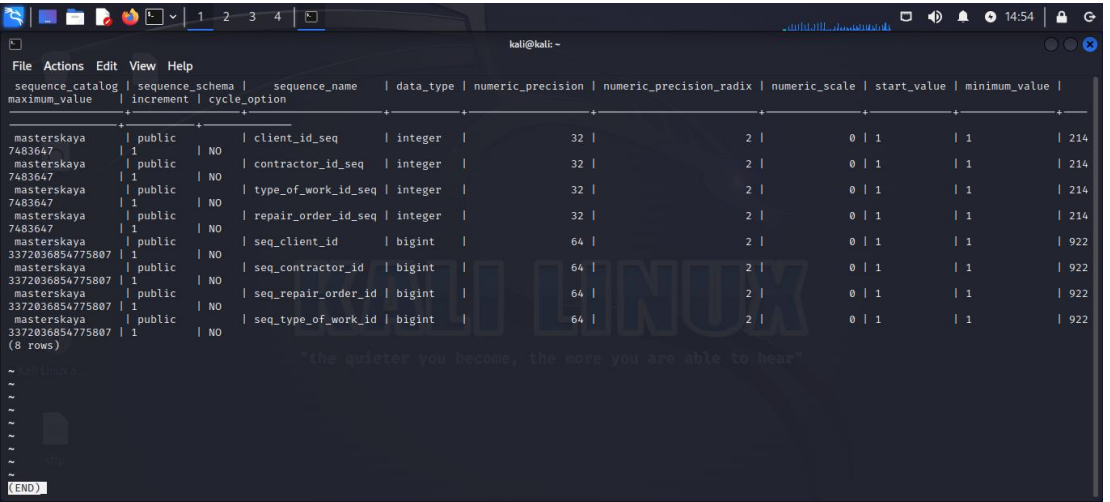
получение списка хранимых процедур:

```
masterskaya=# SELECT * FROM information_schema.routines;
```



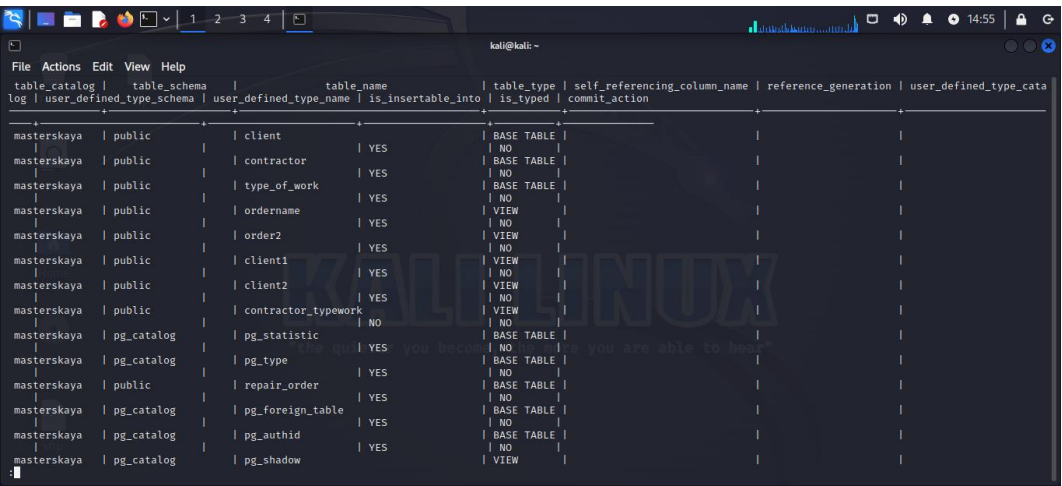
получение списка последовательностей:

```
masterskaya=# SELECT * FROM information_schema.sequences;
masterskaya=#
```



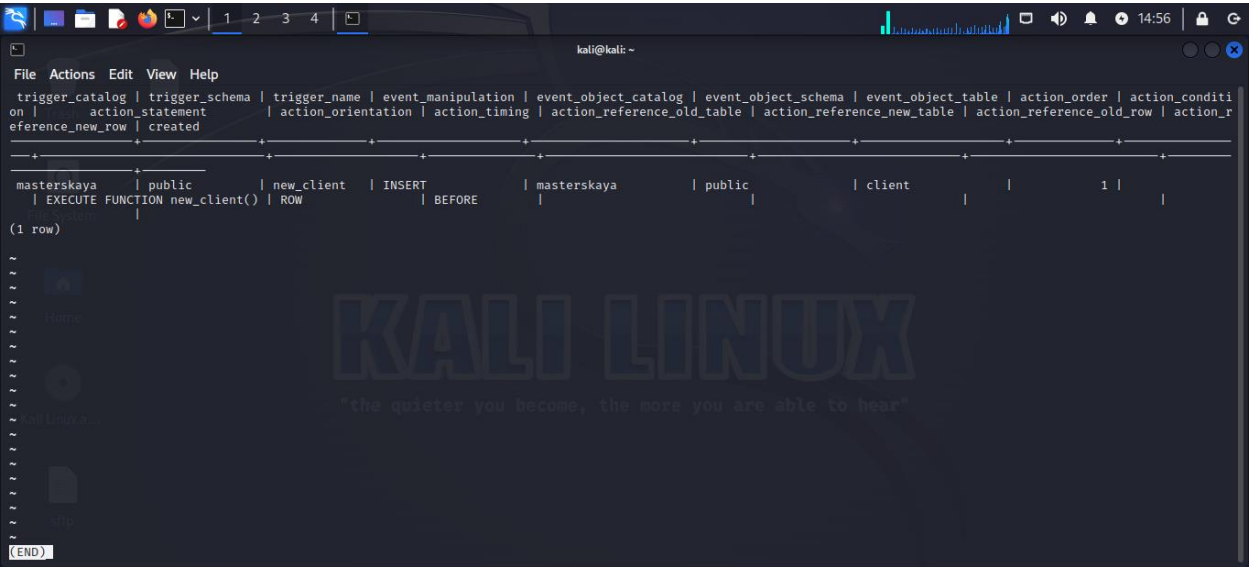
получение списка таблиц:

```
masterskaya=# SELECT * FROM information_schema.tables;
```



получение списка триггеров:

```
masterskaya=# SELECT * FROM information_schema.triggers;
masterskaya=#
```



получение списка представлений:

```
masterskaya=# SELECT * FROM information_schema.triggers;
masterskaya=# SELECT * FROM information\_schema.views;
```

