

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

**ФГБОУ ВО «ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»**

БЕЗОПАСНОСТЬ СИСТЕМ БАЗ ДАННЫХ

ПРАКТИЧЕСКАЯ РАБОТА №4

**ВЫПОЛНИЛ: АИСТОВА ЕЛИЗАВЕТА
ГРУППА: УБ-02**

**ВОРОНЕЖ
2023**

Используем представление для скрытия столбцов. Следующая команда будет содержать только имена и телефон сотрудников.

```
obmen=# CREATE VIEW employe AS SELECT fio, phone FROM employer;
CREATE VIEW
obmen=# SELECT * FROM employe;
      fio      |      phone
-----+-----
Аистова Елизавета | +79124780524
Иванов Иван      | +79217408542
Петров Пётр      | +79564567890
Романов Роман    | +79159876543
(4 строки)
```

Теперь скроем строки. Следующая команда будет содержать только ID и стоимость валюты.

```
obmen=# CREATE VIEW valuta AS SELECT id_value, price FROM course WHERE id_course = 2;
CREATE VIEW
obmen=# SELECT * FROM valuta;
 id_value | price
-----+-----
        1 |  5.75
(1 строка)
```

Используем представление для отображения вычисляемых столбцов. Представление объединит столбцы id_employer и email.

```
obmen=# CREATE VIEW mail AS SELECT fio, ('(' || id_employer || ')') || email as mail FROM employer;
CREATE VIEW
obmen=# SELECT * FROM mail;
      fio      |      mail
-----+-----
Аистова Елизавета | (1)aisotva@mail.ru
Иванов Иван      | (2)ivanov@mail.ru
Петров Пётр      | (3)petrov@mail.ru
Романов Роман    | (4)roma@mail.ru
(4 строки)
```

Используем представление для скрытия сложного синтаксиса. Отобразим сведения о том, какой работник с какой валютой провёл операцию.

```
obmen=# CREATE VIEW obmen AS SELECT e.fio AS Employer, v.name AS Valuta FROM Employer E JOIN operation o ON E.id_employer=o.id_employer JOIN vid v ON v.id_value=o.id_value;
CREATE VIEW
obmen=# SELECT * FROM obmen;
      employer      |      valuta
-----+-----
Аистова Елизавета | Рубль
Петров Пётр      | Фунт
Иванов Иван      | Доллар
Романов Роман    | Йен
(4 строки)
```

Создадим хранимую процедуру, которая будет выводит ФИО работника, сумму и даты связанные с вводимой валютой.

```
obmen=# CREATE OR REPLACE FUNCTION operation(
obmen(#   IN currency_name TEXT,
obmen(#   OUT employer_name TEXT,
obmen(#   OUT operation_sum FLOAT,
obmen(#   OUT operation_date DATE
obmen(# )
obmen-# AS $operation$
obmen$$# DECLARE op RECORD;
obmen$$# BEGIN
obmen$$# FOR op IN SELECT employer.fio, vid.name, operation.sum, operation.date FROM operation JOIN employer
ON operation.id_employer = employer.id_employer JOIN vid ON operation.id_value = vid.id_value WHERE vid.name =
currency_name
obmen$$# LOOP
obmen$$#     employer_name := op.fio;
obmen$$#     operation_sum := op.sum;
obmen$$#     operation_date := op.date;
obmen$$# END LOOP;
obmen$$# END;
obmen$$# $operation$ LANGUAGE plpgsql;
CREATE FUNCTION
obmen=# SELECT operation ('Рубль');
operation
-----
("Аистова Елизавета",11456,2021-06-20)
(1 строка)
```

Создадим триггер, который вызывается перед выполнением операции вставки таблицу employer, и выполняет функцию update_mail(), которая заменяет символ "@" на "_at_" в поле email новой записи. Затем триггер возвращает измененную запись для выполнения операции INSERT.

```
obmen=# CREATE OR REPLACE FUNCTION update_mail() RETURNS TRIGGER AS $update_mail$
obmen$$# BEGIN
obmen$$#     NEW.email := REPLACE(NEW.email, '@', '_at_');
obmen$$#     RETURN NEW;
obmen$$# END;
obmen$$# $update_mail$ LANGUAGE plpgsql;
CREATE FUNCTION
obmen=#
obmen=# CREATE TRIGGER update_mail
obmen-# BEFORE INSERT ON employer
obmen-# FOR EACH ROW
obmen-# EXECUTE FUNCTION update_mail();
CREATE TRIGGER
obmen=# INSERT INTO employer(
obmen(# fio, email, phone)
obmen-# VALUES ('Егоров Егор', 'egor@mail.ru', '+79103498675');
INSERT 0 1
obmen=# SELECT * FROM employer;
 id_employer |      fio      |      email      |      phone
-----+-----+-----+-----
          1 | Аистова Елизавета | aisotva@mail.ru | +79124780524
          2 | Иванов Иван      | ivanov@mail.ru  | +79217408542
          3 | Петров Пётр     | petrov@mail.ru  | +79564567890
          4 | Романов Роман    | roma@mail.ru    | +79159876543
          9 | Егоров Егор      | egor_at_mail.ru | +79103498675
(5 строк)
```

Получим имя текущей базы данных.

```
obmen=# SELECT * FROM
obmen=# information_schema.information_schema_catalog_name;
catalog_name
-----
obmen
(1 строка)
```

Получим список ограничений.

```
obmen=# SELECT * FROM information_schema.table_constraints;
```

constraint_catalog	constraint_schema	constraint_name	table_catalog	table_schema	table_name	constraint_type	is_deferrable	initially_deferred	enforced
obmen	pg_catalog	pg_proc_oid_index	obmen	pg_catalog	pg_proc	PRIMARY KEY	NO	NO	YES
obmen	pg_catalog	pg_proc_proname_args_nsp_index	obmen	pg_catalog	pg_proc	UNIQUE	NO	NO	YES
obmen	pg_catalog	pg_type_oid_index	obmen	pg_catalog	pg_type	PRIMARY KEY	NO	NO	YES
obmen	pg_catalog	pg_type_typname_nsp_index	obmen	pg_catalog	pg_type	UNIQUE	NO	NO	YES
obmen	pg_catalog	pg_attribute_relid_attnam_index	obmen	pg_catalog	pg_attribute	UNIQUE	NO	NO	YES
obmen	pg_catalog	pg_attribute_relid_attnum_index	obmen	pg_catalog	pg_attribute	PRIMARY KEY	NO	NO	YES
obmen	pg_catalog	pg_class_oid_index	obmen	pg_catalog	pg_class	PRIMARY KEY	NO	NO	YES
obmen	pg_catalog	pg_class_relnname_nsp_index	obmen	pg_catalog	pg_class	UNIQUE	NO	NO	YES
obmen	pg_catalog	pg_attrdef_adrelid_adnum_index	obmen	pg_catalog	pg_attrdef	UNIQUE	NO	NO	YES
obmen	pg_catalog	pg_attrdef_oid_index	obmen	pg_catalog	pg_attrdef	PRIMARY KEY	NO	NO	YES
obmen	pg_catalog	pg_constraint_conrelid_contypid_conname_index	obmen	pg_catalog	pg_constraint	UNIQUE	NO	NO	YES
obmen	pg_catalog	pg_constraint_oid_index	obmen	pg_catalog	pg_constraint	PRIMARY KEY	NO	NO	YES
obmen	pg_catalog	pg_inherits_relid_seqno_index	obmen	pg_catalog	pg_inherits	PRIMARY KEY	NO	NO	YES

Получим список внешних ключей.

```
obmen=# information_schema.referential_constraints;
```

constraint_catalog	constraint_schema	constraint_name	unique_constraint_catalog	unique_constraint_schema	unique_constraint_name	match_option	update_rule	delete_rule
obmen	public	operation_id_employer_fkey	obmen	public	employer_pkey	NONE	NO ACTION	NO ACTION
obmen	public	operation_id_value_fkey	obmen	public	vid_pkey	NONE	NO ACTION	NO ACTION
obmen	public	source_id_value_fkey	obmen	public	vid_pkey	NONE	NO ACTION	NO ACTION

(3 строки)

Получим список хранимых процедур.

```
postgres=# \c обмен;
Вы подключены к базе данных "обмен" как пользователь "postgres".
obmen=# SELECT * FROM
obmen=# information_schema.referential_constraints;
```

constraint_catalog	constraint_schema	constraint_name	unique_constraint_catalog	unique_constraint_schema	unique_constraint_name	match_option	update_rule	delete_rule
obmen	public	operation_id_employer_fkey	obmen	public	employer_pkey	NONE	NO ACTION	NO ACTION
obmen	public	operation_id_value_fkey	obmen	public	vid_pkey	NONE	NO ACTION	NO ACTION
obmen	public	source_id_value_fkey	obmen	public	vid_pkey	NONE	NO ACTION	NO ACTION

(3 строки)

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

specific_catalog	specific_schema	specific_name	routine_catalog	routine_schema	routine_name	routine_type	module_catalog	module_schema	module_name	udt_catalog	udt_schema	udt_name	data_type	character_maximum_length	character_octet_length	character_set_catalog	character_set_schema	character_set_name	collation_catalog	collation_schema	collation_name	numeric_precision	numeric_precision_radix	numeric_scale	datetime_precision	interval_type	interval_precision	type_udt_catalog	type_udt_schema	type_udt_name	scope_catalog	scope_schema	scope_name	maximum_cardinality	dtd_identifier	routine_body	external_name	external_security_invoker
obmen	public	operation_id_employer_fkey	obmen	public	employer_pkey	NONE	NO ACTION	NO ACTION																														
obmen	public	operation_id_value_fkey	obmen	public	vid_pkey	NONE	NO ACTION	NO ACTION																														
obmen	public	source_id_value_fkey	obmen	public	vid_pkey	NONE	NO ACTION	NO ACTION																														

(3 строки)

Получим список последовательностей.

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

sequence_catalog	sequence_schema	sequence_name	data_type	numeric_precision	numeric_precision_radix	numeric_scale	start_value	minimum_value	maximum_value	increment	cycle_option
obmen	public	employer_id_employer_seq	integer	32	2	0	1	1	2147483647	1	NO
obmen	public	vid_id_value_seq	integer	32	2	0	1	1	2147483647	1	NO
obmen	public	source_id_source_seq	integer	32	2	0	1	1	2147483647	1	NO
obmen	public	operation_id_operation_seq	integer	32	2	0	1	1	2147483647	1	NO

(4 строки)

Получим список таблиц.

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

table_catalog	table_schema	table_name	table_type	self_referencing_column_name	reference_generation	user_defined_type_catalog	user_defined_type_schema	user_defined_type_name
table_name	is_insertable_into	is_typed	commit_action					
obmen	public	work	VIEW					
obmen	YES	NO						
obmen	public	mail	VIEW					
obmen	YES	NO						
obmen	public	course	BASE TABLE					
obmen	YES	NO						
obmen	public	employer	BASE TABLE					
obmen	YES	NO						
obmen	public	obmen	VIEW					
obmen	NO	NO						
obmen	public	operation	BASE TABLE					
obmen	YES	NO						
obmen	public	vid	BASE TABLE					
obmen	YES	NO						
obmen	public	employee	VIEW					
obmen	YES	NO						
obmen	public	valuta	VIEW					
obmen	YES	NO						
obmen	pg_catalog	pg_statistic	BASE TABLE					
obmen	YES	NO						

Получим список триггеров.

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

trigger_catalog	trigger_schema	trigger_name	event_manipulation	event_object_catalog	event_object_schema	event_object_table	action_order	action_condition	action_statement	action_orientation	action_timing	action_reference_old_table	action_reference_new_table	action_reference_old_row	action_reference_new_row	created
obmen	public	update_mail	INSERT	obmen	public	employer	1		EXECUTE FUNCTION update_mail()	ROW	BEFORE					

(1 строка)

Получим список представлений.

```
obmen=# SELECT * FROM information_schema.views;
```

table_catalog	table_schema	table_name	view_definition	check_option	is_updatable	is_insertable_into	is_trigger_updatable	is_trigger_deletable	is_trigger_insertable_into
obmen	public	work	SELECT employer.fio,						
			+ NONE	YES	YES	NO	NO	NO	NO
			(((('::text employer.id_employer) '::text) (employer.phone)::text) AS phone						
			+						
			FROM employer;						
obmen	public	mail	SELECT employer.fio,						
			+ NONE	YES	YES	NO	NO	NO	NO

-- Далее --