

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

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

**БЕЗОПАСНОСТЬ СИСТЕМ БАЗ ДАННЫХ**

**ПРАКТИЧЕСКАЯ РАБОТА №4**

**ВЫПОЛНИЛ: ПУГОВКИН АНДРЕЙ НИКОЛАЕВИЧ**  
**ГРУППА: УБ-01**  
**НОМЕР ЗАЧЕТНОЙ КНИЖКИ: 207378**

**ВОРОНЕЖ**  
**2023**

Используем представление для скрытия столбцов. Следующая команда будет содержать только имена и паспорт гостей.

```
hotel=# CREATE VIEW Person AS SELECT full_name, passport FROM Guest;
CREATE VIEW
hotel=# SELECT * FROM Person;
      full_name      | passport
-----+-----
Пуговкин Андрей Николаевич | 42 16 010633
Зенищева Дарья Леонидовна | 42 17 020744
Цой Кирилл Владимирович | 42 18 030855
Цой Илья Владимирович | 42 18 030856
Хомутов Константин Зиновьевич | 42 19 040966
(5 строк)
```

Теперь скроем строки. Следующая команда будет содержать только имя и номер телефона Адмнистратора.

```
hotel=# CREATE VIEW Personal AS SELECT fio, phone FROM Employee WHERE position = 'Администратор';
CREATE VIEW
hotel=# SELECT * FROM Personal;
      fio      | phone
-----+-----
Денисенко Владимир Владимирович | +79012514563
(1 строка)
```

Используем представление для отображения вычисляемых столбцов. Представление объединит столбцы id\_guest и phone.

```
hotel=# CREATE VIEW People AS SELECT full_name, ((' || id_guest || ')) || phone AS phone FROM Guest;
CREATE VIEW
hotel=# SELECT * FROM People;
      full_name      | phone
-----+-----
Пуговкин Андрей Николаевич | (1)+79803569205
Зенищева Дарья Леонидовна | (2)+79801234567
Цой Кирилл Владимирович | (3)+79807654321
Цой Илья Владимирович | (4)+79801276345
Хомутов Константин Зиновьевич | (5)+79805461327
(5 строк)
```

Используем представление для скрытия сложного синтаксиса. Отобразим сведения о том, какой работник зарегистрировал гостя.

```
hotel=# CREATE VIEW Interest AS SELECT E.fio AS Employee, G.full_name AS Guest FROM Employee E JOIN Registration R ON E.id_employee=R.employee_id JOIN Guest G ON G.id_guest=R.id_guest;
CREATE VIEW
hotel=# SELECT * FROM Interest;
      employee      | guest
-----+-----
Маслов Александр Александрович | Хомутов Константин Зиновьевич
Денисенко Владимир Владимирович | Цой Кирилл Владимирович
Денисенко Владимир Владимирович | Цой Илья Владимирович
Саранов Игорь Александрович | Пуговкин Андрей Николаевич
Савченко Илона Игоревна | Зенищева Дарья Леонидовна
(5 строк)
```

Создадим хранимую процедура, которая будет выводить всех пользователей, которые жили в заданном типе комнаты.

```
hotel=# CREATE OR REPLACE FUNCTION GuestsType(  
hotel(# IN roomtype TEXT,  
hotel(# OUT guestname TEXT,  
hotel(# OUT guestpassport TEXT,  
hotel(# OUT guestphone TEXT,  
hotel(# OUT checkindate DATE,  
hotel(# OUT checkoutdate DATE  
hotel(# )  
hotel=# AS $GuestsType$  
hotel$# DECLARE guest_record RECORD;  
hotel$# BEGIN  
hotel$#   FOR guest_record IN SELECT Guest.full_name, Guest.passport, Guest.phone, Room.type, Registration.checkin_date, Registration.checkout_date FROM Guest JOIN Registration ON Guest.id_guest = Registration.id_guest JOIN Room ON Registration.id_room = Room.id_room WHERE Room.type = roomtype  
hotel$#   LOOP  
hotel$#     guestname := guest_record.full_name;  
hotel$#     guestpassport := guest_record.passport;  
hotel$#     guestphone := guest_record.phone;  
hotel$#     checkindate := guest_record.checkin_date;  
hotel$#     checkoutdate := guest_record.checkout_date;  
hotel$#     RAISE NOTICE 'Гость % с паспортом % и телефоном % остановился в % комнате с % по %', guestname, guestpassport, guestphone, roomtype, checkindate, checkoutdate;  
hotel$#   END LOOP;  
hotel$# END;  
hotel$# $GuestsType$ LANGUAGE plpgsql;  
CREATE FUNCTION
```

```
hotel=# SELECT gueststype ('Lux');  
NOTICE: Гость Хомутов Константин Зиновьевич с паспортом 42 19 040966 и телефоном +79805461327 остановился в Lux комнате с 2018-05-24 по 2018-06-02  
gueststype  
-----  
("Хомутов Константин Зиновьевич", "42 19 040966", +79805461327, 2018-05-24, 2018-06-02)  
(1 строка)  
  
hotel=# SELECT gueststype ('Single');  
NOTICE: Гость Пуговкин Андрей Николаевич с паспортом 42 16 010633 и телефоном +79803569205 остановился в Single комнате с 2017-12-01 по 2017-12-25  
gueststype  
-----  
("Пуговкин Андрей Николаевич", "42 16 010633", +79803569205, 2017-12-01, 2017-12-25)  
(1 строка)
```

Создадим триггер, который относится к типу "Использование триггеров для проверки допустимости вводимых данных". Проверяет, зарегистрирована ли комната на указанный период времени, прежде чем добавит новую запись в таблицу Registration. Если комната уже забронирована на указанный период времени, триггер вызывает исключение с сообщением об ошибке.

```
hotel=# CREATE OR REPLACE FUNCTION occupied() RETURNS TRIGGER AS $occupied$  
hotel$# BEGIN  
hotel$#   IF EXISTS ( SELECT * FROM Registration WHERE id_room = NEW.id_room AND ((NEW.checkin_date, NEW.checkout_date) OVERLAPS (checkin_date, checkout_date) OR NEW.checkin_date = checkin_date OR NEW.checkout_date = checkout_date))  
hotel$#   THEN  
hotel$#     RAISE EXCEPTION 'Комната % уже забронирован на запрошенный период', NEW.id_room;  
hotel$#   END IF;  
hotel$#   RETURN NEW;  
hotel$# END;  
hotel$# $occupied$ LANGUAGE plpgsql;  
CREATE FUNCTION  
hotel=# CREATE TRIGGER occupied  
hotel=# BEFORE INSERT ON Registration  
hotel=# FOR EACH ROW EXECUTE FUNCTION occupied();  
CREATE TRIGGER
```

```
hotel=# SELECT * FROM Guest;  
id_guest | full_name | passport | phone | id_room  
-----+-----+-----+-----+-----  
1 | Пуговкин Андрей Николаевич | 42 16 010633 | +79803569205 | 101  
2 | Зенищева Дарья Леонидовна | 42 17 020744 | +79801234567 | 24  
3 | Цой Кирилл Владимирович | 42 18 030855 | +79807654321 | 121  
4 | Цой Илья Владимирович | 42 18 030856 | +79801276345 | 121  
5 | Хомутов Константин Зиновьевич | 42 19 040966 | +79805461327 | 59  
6 | Колаев Кирилл Павлович | 4217 123456 | +79801234567 | 59  
(6 строк)  
  
hotel=# INSERT INTO Registration (  
hotel=# id_guest, id_room, checkin_date, checkout_date, employee_id)  
hotel=# VALUES ('6', '59', '2018-05-27', '2018-06-01', '3');  
ERROR: Комната 59 уже забронирован на запрошенный период
```

Создадим сводную таблицу, которая выведет сколько комнат было забронировано в каждом месяце каждого года.

```
hotel=# SELECT YEAR,
hotel=# SUM(JAN) AS JAN, SUM(FEB) AS FEB, SUM(MAR) AS MAR, SUM(APR) AS APR, SUM(MAY) AS MAY, SUM(JUN) AS JUN, SUM(JUL) AS JUL, SUM(AUG) AS AUG, SUM(SEP) AS SEP, SUM(OCT) AS OCT, SUM(NOV) AS NOV, SUM(DEC) AS DEC
hotel=# FROM crosstab(
hotel=# 'SELECT EXTRACT(YEAR FROM checkin_date) AS YEAR,
hotel=# EXTRACT(MONTH FROM checkin_date) AS MONTH,
hotel=# COUNT(*) AS COUNT
hotel=# FROM Registration
hotel=# GROUP BY YEAR, MONTH',
hotel=# 'SELECT GENERATE_SERIES(1, 12)'
hotel=# ) AS (
hotel=# YEAR int,
hotel=# JAN int, FEB int, MAR int, APR int, MAY int, JUN int,
hotel=# JUL int, AUG int, SEP int, OCT int, NOV int, DEC int
hotel=# )
hotel=# GROUP BY YEAR
hotel=# ORDER BY YEAR;
 year | jan | feb | mar | apr | may | jun | jul | aug | sep | oct | nov | dec
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
2017 |    |    |    |    |    |    |    |    |    |    |    |    |
2018 |    |    |    |    |    |    |    |    |    |    |    |    |    |
2019 |    |    |    |    |    |    |    |    |    |    |    |    |    |
2020 |    |    |    |    |    |    |    |    |    |    |    |    |    |
(4 строки)
```

Получим имя текущей базы данных.

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

Получим список ограничений.

```
hotel=# 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
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
 hotel | pg_catalog | pg_proc_oid_index | hotel | pg_catalog | pg_proc | PRIMARY KEY | NO | NO | YES
 hotel | pg_catalog | pg_proc_proname_args_nsp_index | hotel | pg_catalog | pg_proc | UNIQUE | NO | NO | YES
 hotel | pg_catalog | pg_type_oid_index | hotel | pg_catalog | pg_type | PRIMARY KEY | NO | NO | YES
 hotel | pg_catalog | pg_type_typname_nsp_index | hotel | pg_catalog | pg_type | UNIQUE | NO | NO | YES
 hotel | pg_catalog | pg_attribute_relid_attnam_index | hotel | pg_catalog | pg_attribute | UNIQUE | NO | NO | YES
 hotel | pg_catalog | pg_attribute_relid_attnum_index | hotel | pg_catalog | pg_attribute | PRIMARY KEY | NO | NO | YES
 hotel | pg_catalog | pg_class_oid_index | hotel | pg_catalog | pg_class | PRIMARY KEY | NO | NO | YES
 hotel | pg_catalog | pg_class_relname_nsp_index | hotel | pg_catalog | pg_class | UNIQUE | NO | NO | YES
 hotel | pg_catalog | pg_attrdef_adrelid_adnum_index | hotel | pg_catalog | pg_attrdef | UNIQUE | NO | NO | YES
 hotel | pg_catalog | pg_attrdef_oid_index | hotel | pg_catalog | pg_attrdef | PRIMARY KEY | NO | NO | YES
 hotel | pg_catalog | pg_constraint_conrelid_contypid_conname_index | hotel | pg_catalog | pg_constraint | UNIQUE | NO | NO | YES
 hotel | pg_catalog | pg_constraint_oid_index | hotel | pg_catalog | pg_constraint | PRIMARY KEY | NO | NO | YES
 hotel | pg_catalog | pg_inherits_relid_seqno_index | hotel | pg_catalog | pg_inherits | PRIMARY KEY | NO | NO | YES
 hotel | pg_catalog | pg_index_indexrelid_index | hotel | pg_catalog | pg_index | PRIMARY KEY | NO | NO | YES
 hotel | pg_catalog | pg_operator_oid_index | hotel | pg_catalog | pg_operator | PRIMARY KEY | NO | NO | YES
 hotel | pg_catalog | pg_operator_oprname_l_r_n_index | hotel | pg_catalog | pg_operator | UNIQUE | NO | NO | YES
 hotel | pg_catalog | pg_opfamily_am_name_nsp_index | hotel | pg_catalog | pg_opfamily | UNIQUE | NO | NO | YES
 hotel | pg_catalog | pg_opfamily_oid_index | hotel | pg_catalog | pg_opfamily | PRIMARY KEY | NO | NO | YES
 hotel | pg_catalog | pg_opclass_am_name_nsp_index | hotel | pg_catalog | pg_opclass | UNIQUE | NO | NO | YES
 hotel | pg_catalog | pg_opclass_oid_index | hotel | pg_catalog | pg_opclass | PRIMARY KEY | NO | NO | YES
```

Получим список внешних ключей.

```
hotel=# SELECT * FROM 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
-----+-----+-----+-----+-----+-----+-----+-----+-----
 hotel | public | registration_id_room_fkey | hotel | public | room_pkey | NONE | NO ACTION | NO ACTION
 hotel | public | guest_id_room_fkey | hotel | public | room_pkey | NONE | NO ACTION | NO ACTION
 hotel | public | registration_id_guest_fkey | hotel | public | guest_pkey | NONE | NO ACTION | NO ACTION
 hotel | public | registration_employee_id_fkey | hotel | public | employee_pkey | NONE | NO ACTION | NO ACTION
(4 строки)
```

Получим список хранимых процедур.

[illegible]

Получим список последовательностей.

option	sequence_catalog	sequence_schema	sequence_name	data_type	numeric_precision	numeric_precision_radix	numeric_scale	start_value	minimum_value	maximum_value	increment	cycle
hotel	public	guest_id_guest_seq	integer	32	2	0	1	1	2147483647	1	NO	
hotel	public	employee_id_employee_seq	integer	32	2	0	1	1	2147483647	1	NO	
hotel	public	registration_id_registration_seq	integer	32	2	0	1	1	2147483647	1	NO	

(3 строки)

Получим список таблиц.

hotel=# 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	user_defined_type_base_name
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	user_defined_type_base_name
hotel	public	guest	BASE TABLE						
hotel	public	registration	BASE TABLE						
hotel	public	employee	BASE TABLE						
hotel	public	room	BASE TABLE						
hotel	public	person	VIEW						
hotel	public	personal	VIEW						
hotel	public	people	VIEW						
hotel	public	interest	VIEW						
hotel	pg_catalog	pg_statistic	BASE TABLE						
hotel	pg_catalog	pg_type	BASE TABLE						
hotel	pg_catalog	pg_foreign_table	BASE TABLE						
hotel	pg_catalog	pg_authid	BASE TABLE						
hotel	pg_catalog	pg_shadow	VIEW						
hotel	pg_catalog	pg_statistic_ext_data	BASE TABLE						

## Получим список триггеров.

[illegible]

Получим список представлений.

```
hotel=# SELECT * FROM information\_schema.views;
неверная команда \_schema.views;
Введите \? для получения справки.
hotel=# SELECT * FROM information\_schema.views;
ERROR: syntax error at or near "SELECT"
СТРОКА 2: SELECT * FROM information\_schema.views;
^
hotel=# 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
+-----+
hotel | public | person | SELECT guest.full_name,
| | | +| NONE | YES | YES | NO | NO | NO
| | | | guest.passport
| | | +| | FROM guest;
hotel | public | personal | SELECT employee.fio,
```