#### МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

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

#### БЕЗОПАСНОСТЬ СИСТЕМ БАЗ ДАННЫХ

ПРАКТИЧЕСКАЯ РАБОТА №3

выполнил: пуговкин андрей николаевич

ГРУППА: УБ-01

НОМЕР ЗАЧЕТНОЙ КНИЖКИ: 207378

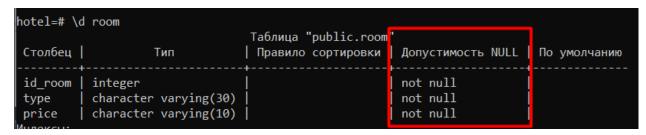
Оператор ALTER TABLE используется для изменения определения существующей таблицы, такой как изменение структуры таблицы, изменение типов данных столбцов, добавление ограничений.

Так, при помощи данного оператора поставим ограничение NOT NULL у нужных столбцов, используя команду:

## "ALTER TABLE 'Название таблицы' ALTER COLUMN 'Название столбца' SET NOT NULL".

hotel=# ALTER TABLE Room ALTER COLUMN type SET NOT NULL; ALTER TABLE

hotel=# ALTER TABLE Room ALTER COLUMN price SET NOT NULL; ALTER TABLE



hotel=# ALTER TABLE Guest ALTER COLUMN full\_name SET NOT NULL;
ALTER TABLE
hotel=# ALTER TABLE Guest ALTER COLUMN passport SET NOT NULL;
ALTER TABLE
hotel=# ALTER TABLE Guest ALTER COLUMN phone SET NOT NULL;
ALTER TABLE

hotel=# \d guest				
Таблица "public.guest"				
Столбец Тип	Правило сортировки	Допустимость NULL	По умолчанию	
	+			
id_guest   integer		not null	nextval('guest_id_guest_se	
q'::regclass)				
full_name   character varying(60)		not null		
passport   character varying(15)	1	not null		
phone   character varying(15)	1	not null		
id_room   integer	1			

hotel=# ALTER TABLE Employee ALTER COLUMN fio SET NOT NULL;
ALTER TABLE
hotel=# ALTER TABLE Employee ALTER COLUMN position SET NOT NULL;
ALTER TABLE
hotel=# ALTER TABLE Employee ALTER COLUMN phone SET NOT NULL;
IALTER TABLE

hotel=# \d emp	ployee			
Столбец анию	Тип		"public.employee"   Допустимость NULL	По умолч
	+    :-+	-+	   not null	
<pre>id_employee loyee_seq'::re fio</pre>		1	not null	nextval('employee_id_emp
position phone	character varying(20)   character varying(15)	į	not null not null	

Также, при помощи данного оператора можно добавлять (удалять) новые столбцы следующей командой:

# "ALTER TABLE 'Название таблицы' ADD (DROP) COLUMN 'Название столбца' 'Тип данных'".

hotel=# Al ALTER TABI hotel=# \c		MN test varchar; Таблица "public.room'		
Столбец	Тип	Правило сортировки		По умолчанию
	integer   character varying(30)   character varying(10)   character varying		not null not null not null	     
hotel=# Al ALTER TABI hotel=# \a		LUMN test;		

hotel=# AI ALTER TABI hotel=# \a		UMN test;		
Столбец		Таблица "public.room'   Правило сортировки		По умолчанию
id_room type price	   integer   character varying(30)   character varying(10)		not null not null not null	

Для того, чтобы вывести выполнить запрос для выборки используем оператор **SELECT**. Так, выведем ID гостя и его ФИО проживающего в комнате 121 из таблицы Guest.

Используя **ORDER BY 'Название колонки' ASC** (**DESC**), можно вывести результаты в порядке возрастания (убывания). Так, выведем ID гостей и дату их заезда в период с 2017 по 2019 года в порядке возрастания в таблице Registration.

```
hotel=# SELECT id_guest, checkin_date FROM Registration WHERE checkin_date BETWEEN '2017-01-01'AND '2019-12-31'
ORDER BY checkin_date ASC;
id_guest | checkin_date

1 | 2017-12-01
5 | 2018-05-24
3 | 2019-07-07
4 | 2019-07-07
(4 строки)
```

Чтобы узнать имена этих ID выполним следующую команду:

Для того, чтобы вывести всех гостей, чья фамилия начинается с определённой буквы, можно использовать следующую команду, где "%" обозначает последовательность из нескольких произвольных символов:

```
hotel=# SELECT * FROM GUEST WHERE full_name LIKE 'Ц%';
id_guest | full_name | passport | phone | id_room

3 | Цой Кирилл Владимирович | 42 18 030855 | +79807654321 | 121
4 | Цой Илья Владимирович | 42 18 030856 | +79801276345 | 121
(2 строки)
```

Также, используя функция **COUNT**, можно вывести количество строк из определённой таблицы, а функции **MIN** и **MAX** выводят минимальные и максимальные значения из таблицы.

```
hotel=# SELECT COUNT(*) FROM Employee;_
count
-----
4
(1 строка)
```

Для вставки данных в таблицу, можно использовать оператор **INSERT**, указав имена столбцов, которым будут присвоены значения:

```
hotel=# INSERT INTO Room (
hotel(# id room, type, price)
hotel-# VALUES ('01', 'Alone', '100$');
INSERT 0 1
hotel=# SELECT * FROM Room;
 id room | type
                  price
     101 | Single |
                    300$
      59
           Lux
                    700$
           Double
                    350$
     121
                    5aa¢
           Alone
                    100$
```

Используя **UPDATE** можно изменить данные не только одного столбца, но и сразу нескольких:

```
hotel=# UPDATE Room SET price ='300$', type = 'Lux' where id_room = 1;
UPDATE 1
hotel=# SELECT * FROM Room;
 id room
           type
                  price
          Single
    101
                   300$
      59
          Lux
                    700$
          Double
                    350$
     121
                  Lsaat
          Lux
                   300$
 5 строк)
```

### Для удаления данных из таблицы используется оператор **DELETE**: