

## Отчет по ДЗ 2. Глава 3 (упражнения 1–4).

### Упражнение 1

1. Попробуйте ввести в таблицу `aircrafts` строку с таким значением атрибута «Код самолета» (`aircraft_code`), которое вы уже вводили, например:

```
INSERT INTO aircrafts  
VALUES ( 'SU9', 'Sukhoi SuperJet-100', 3000 );
```

Обратите внимание, что в этой команде мы не привели список атрибутов, что вполне допустимо при задании значений атрибутов в том же порядке, в котором атрибуты следуют в определении таблицы. Но в ваших прикладных программах так поступать все же не следует, поскольку в случае возможной реструктуризации таблицы и изменения порядка следования атрибутов в ней ваши команды `INSERT` могут перестать работать корректно.

Вы получите сообщение об ошибке.

ОШИБКА: повторяющееся значение ключа нарушает ограничение уникальности "aircrafts\_pkey"

ПОДРОБНОСТИ: Ключ "(aircraft\_code)=(SU9)" уже существует.

Подумайте, почему появилось сообщение. Если вы забыли структуру таблицы `aircrafts`, то можно вывести ее определение на экран с помощью команды

```
\d aircrafts
```

```
postgres@DESKTOP-3C11GST: ~  
postgres@DESKTOP-3C11GST:~$ psql -d demo  
psql (14.9 (Ubuntu 14.9-0ubuntu0.22.04.1))  
Type "help" for help.  
  
demo=# set search_path=bookings  
demo=# ;  
SET  
demo=# select * from aircrafts;  
 aircraft_code |          model          | range  
-----+-----+-----  
773            | Boeing 777-300          | 11100  
763            | Boeing 767-300          | 7900  
SU9            | Sukhoi SuperJet-100     | 3000  
320            | Airbus A320-200         | 5700  
321            | Airbus A321-200         | 5600  
319            | Airbus A319-100         | 6700  
733            | Boeing 737-300          | 4200  
CN1            | Cessna 208 Caravan      | 1200  
CR2            | Bombardier CRJ-200      | 2700  
(9 rows)  
  
demo=#
```

```
demo=# INSERT INTO aircrafts VALUES ('MIG', 'MIG29', 1500);
INSERT 0 1
demo=# SELECT * FROM aircrafts;
```

aircraft_code	model	range
773	Boeing 777-300	11100
763	Boeing 767-300	7900
SU9	Sukhoi SuperJet-100	3000
320	Airbus A320-200	5700
321	Airbus A321-200	5600
319	Airbus A319-100	6700
733	Boeing 737-300	4200
CN1	Cessna 208 Caravan	1200
CR2	Bombardier CRJ-200	2700
MIG	MIG29	1500

```
(10 rows)

demo=#
```

- Предложение ORDER BY команды SELECT позволяет отсортировать данные при выводе. По умолчанию сортировка выполняется по возрастанию значений атрибута, указанного в этом предложении. Но можно упорядочить строки и по убыванию значения атрибута. Для этого нужно после имени атрибута в предложении ORDER BY добавить ключевое слово DESC (это сокращение от слова descendant — убывающий порядок). Самостоятельно напишите команду для выборки всех строк из таблицы aircrafts, чтобы строки были упорядочены по убыванию значения атрибута «Максимальная дальность полета, км» (range).

```
demo=# SELECT * FROM aircrafts
demo=# ORDER BY range DESC;
```

aircraft_code	model	range
773	Boeing 777-300	11100
763	Boeing 767-300	7900
319	Airbus A319-100	6700
320	Airbus A320-200	5700
321	Airbus A321-200	5600
733	Boeing 737-300	4200
SU9	Sukhoi SuperJet-100	3000
CR2	Bombardier CRJ-200	2700
MIG	MIG29	1500
CN1	Cessna 208 Caravan	1200

(10 rows)

3. Команда UPDATE позволяет в процессе обновления выполнять арифметические действия над значениями, находящимися в строках таблицы. Представим себе, что двигатели самолета Sukhoi SuperJet стали в два раза экономичнее, вследствие чего дальность полета этого лайнера возросла ровно в два раза. Команда UPDATE позволяет увеличить значение атрибута range в строке, хранящей информацию об этом самолете, даже не выполняя предварительно выборку с целью выяснения текущего значения этого атрибута. При присваивании нового

значения атрибуту range можно справа от знака «=» написать не только числовую константу, но и целое выражение. В нашем случае оно будет простым: `range = range * 2`. Самостоятельно напишите команду UPDATE полностью, при этом не забудьте, что увеличить дальность полета нужно только у одной модели — Sukhoi SuperJet, поэтому необходимо использовать условие WHERE. Затем с помощью команды SELECT проверьте полученный результат.

```

demo=# \h UPDATE
Command:      UPDATE
Description:  update rows of a table
Syntax:
[ WITH [ RECURSIVE ] with_query [, ...] ]
UPDATE [ ONLY ] table_name [ * ] [ [ AS ] alias ]
    SET { column_name = { expression | DEFAULT } |
        ( column_name [, ...] ) = [ ROW ] ( { expression | DEFAULT } [, ...] ) |
        ( column_name [, ...] ) = ( sub-SELECT )
    } [, ...]
    [ FROM from_item [, ...] ]
    [ WHERE condition | WHERE CURRENT OF cursor_name ]
    [ RETURNING * | output_expression [ [ AS ] output_name ] [, ...] ]

URL: https://www.postgresql.org/docs/14/sql-update.html

demo=# UPDATE aircrafts SET range=range*2 WHERE model='Sukhoi SuperJet-100';
UPDATE 1
demo=# SELECT * FROM aircrafts;
 aircraft_code |          model          | range
-----+-----+-----
773            | Boeing 777-300         | 11100
763            | Boeing 767-300         | 7900
320            | Airbus A320-200        | 5700
321            | Airbus A321-200        | 5600
319            | Airbus A319-100        | 6700
733            | Boeing 737-300         | 4200
CN1            | Cessna 208 Caravan     | 1200
CR2            | Bombardier CRJ-200     | 2700
MIG            | MIG29                  | 1500
SU9            | Sukhoi SuperJet-100    | 6000
(10 rows)

demo=#

```

- Если в предложении WHERE команды DELETE вы укажете логически и синтаксически корректное условие, но строк, удовлетворяющих этому условию, в таблице не окажется, то в ответ СУБД выведет сообщение

```
DELETE 0
```

Такая ситуация не является ошибкой или сбоем в работе СУБД. Например, если после удаления какой-то строки вы повторно попытаетесь удалить ее же, то получите именно такое сообщение.

Самостоятельно смоделируйте описанную ситуацию, подобрав условие, которому гарантированно не соответствует ни одна строка в таблице «Самолеты» (aircrafts).

```

demo=# DELETE FROM aircrafts WHERE aircraft_code='NaN';
DELETE 0
demo=#

```