

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное учреждение высшего
образования

“Московский авиационный институт”

(национальный исследовательский университет)

Факультет №3 «Системы управления, информатика и электроэнергетика»

Отчет по домашней работе №2

по курсу «**Управление базами данных**»

Выполнили:

студенты группы ЗО-218М-19:

Пономарев Роман

Принял:

Доцент, к.т.н. Моргунов Е.П.

Москва, 2020

- 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**

Ответ: SELECT count(*) FROM seats WHERE aircraft_code = 'SU9';

При попытке добавить в таблицу **aircrafts** строку со значение атрибута **aircraft_code = 'SU9'** мы получаем указанную выше ошибку. Это происходит из-за того, что значения атрибута **aircraft_code** должны быть уникальными (мы объявляли его как первичный ключ), а запись с таким же значением уже имеется.

```
demo=# INSERT INTO aircrafts VALUES ('SU9', 'Sukhoi SuperJet-100', 3000);
ОШИБКА: повторяющееся значение ключа нарушает ограничение уникальности "aircrafts_pkey"
ПОДРОБНОСТИ: Ключ "(aircraft_code)=(SU9)" уже существует.
demo=#
```

- 2) 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**).

Ответ: SELECT * FROM aircrafts ORDER BY range DESC;

```
demo=# SELECT * FROM aircrafts 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
 CN1           | Cessna 208 Caravan | 1200
(9 строк)
demo=#
```

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.

Ответ:

UPDATE aircrafts SET range = range * 2 WHERE aircraft_code = 'SU9';

```
demo=# SELECT * FROM aircrafts 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
CN1	Cessna 208 Caravan	1200

(9 строк)

```
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
733	Boeing 737-300	4200
320	Airbus A320-200	5700
321	Airbus A321-200	5600
319	Airbus A319-100	6700
CN1	Cessna 208 Caravan	1200
CR2	Bombardier CRJ-200	2700
SU9	Sukhoi SuperJet-100	6000

(9 строк)

4) 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**).

Ответ: DELETE FROM aircrafts WHERE range < 0;

```
demo=# DELETE FROM aircrafts WHERE range < 0;
```

```
DELETE 0
```

```
demo=#
```