Федеральное государственное бюджетное образовательное учреждение высшего образования
"Уфимский государственный авиационный технический университет"

Кафедра Высокопроизводительных вычислительных технологий и систем

Дисциплина: Базы данных

Отчет по лабораторной работе № 5

Тема: «Модификация данных.Транзакции»

Группа ПМ-353	Фамилия И.О.	Подпись	Дата	Оценка
Студент	Шамаев И.Р.			
Принял	Ямилева А.М.			

Уфа 2022

Теоритическая часть

Для подключения базы данных используется команда

psql -d demo -U postgres

Для создания таблиц в языке SQL служит команда CREATE TABLE.

Для удаления таблицы служит команда DROP TABLE;

Для выполнения вводу данных в таблицу служит команда INSERT.Ее упрощенный формат таков:

INSERT INTO имя-таблицы [(имя-атрибута, имя-атрибута, ...)] VALUES (значение-атрибута, значение-атрибута, ...);

Для выборки информации из таблиц базы данных служит команда SELECT. Ее синтаксис, упрощенный до предела, таков:

SELECT имя-атрибута, имя-атрибута, ... FROM имя-таблицы;

Теперь мы ознакомимся с командой UPDATE, предназначенной для обновления данных в таблицах. Ее упрощенный синтаксис таков:

UPDATE имя-таблицы SET имя-атрибута1 = значение-атрибута1, имя-атрибута2 = значение-атрибута2, ... WHERE условие;

Типы данных СУБД PostgreSQL

Группа числовых типов данных включает в себя целый ряд разновидностей: целочисленные типы, числа фиксированной точности, типы данных с плавающей точкой, последовательные типы (serial). В составе целочисленных типов находятся следующие представители: smallint, integer, bigint. Если атрибут таблицы имеет один из этих типов, то он позволяет хранить только целочисленные данные. При этом перечисленные типы различаются по количеству байтов, выделяемых для хранения данных.

Числа фиксированной точности представлены двумя типами — numeric и decimal. Однако они являются идентичными по своим возможностям. Для задания значения этого типа используются два базовых понятия: масштаб (scale) и точность (precision). Масштаб показывает число значащих цифр, стоящих справа от десятичной точки (запятой). Точность указывает общее число цифр как до десятичной точки, так и после нее.

Стандартные представители строковых типов — это типы character varying(n) и character(n), где параметр указывает максимальное число символов в строке, которую можно сохранить в столбце такого типа. При работе с многобайтовыми кодировками символов, например UTF-8, нужно учитывать, что речь идет о символах, а не о байтах. Если сохраняемая строка символов будет короче, чем указано в определении типа, то значение типа

character будет дополнено пробелами до требуемой длины, а значение типа character varying будет сохранено так, как есть.

PostgreSQL поддерживает все типы данных, предусмотренные стандартом SQL для даты и времени. Даты обрабатываются в соответствии с григорианским календарем, причем это делается даже в тех случаях, когда дата относится к тому моменту времени, когда этот календарь в данной стране еще не был принят. Для этих типов данных предусмотрены определенные форматы для ввода значений и для вывода. Причем эти форматы могут не совпадать.

В результате объединения типов даты и времени получается интегральный тип — временная отметка. Этот тип существует в двух вариантах: с учетом часового пояса — timestamp with time zone, либо без учета часового пояса — timestamp. Для первого варианта существует сокращенное наименование — timestamptz, которое является расширением PostgreSQL. При вводе и выводе значений этого типа данных используются соответствующие форматы ввода и вывода даты и времени.

Для получения значения текущей временной отметки (т. е. даты и времени в одном ′ значении) служит функция current_timestamp.

Последним типом является interval, который представляет продолжительность отрезка времени между двумя моментами времени. Его формат ввода таков: quantity unit [quantity unit ...] direction

Логический (boolean) тип может принимать три состояния: истина и ложь, а так же неопределенное состояние, которое можно представить значением NULL. Таким образом, тип boolean реализует трехзначную логику.

Практическая часть

Задание 4.14

```
demo=# INSERT INTO seats (aircraft_code, seat_no, fare_conditions)
demo-# VALUES ('CN1','7A','Economy');
INSERT 0 1
demo=# INSERT INTO seats (aircraft_code, seat_no, fare_conditions)
demo-# VALUES ('CN1','7B','Economy');
INSERT 0 1
demo=# select * from seats WHERE seats.aircraft_code ='CN1';
 aircraft_code | seat_no | fare_conditions
---------------+---------+-----------------
 CN1           | 1A      | Economy
 CN1           | 1B      | Economy
 CN1           | 2A      | Economy
 CN1           | 2B      | Economy
 CN1           | 3A      | Economy
 CN1           | 3B      | Economy
 CN1           | 4A      | Economy
 CN1           | 4B      | Economy
 CN1           | 5A      | Economy
 CN1           | 5B      | Economy
 CN1           | 6A      | Economy
 CN1           | 6B      | Economy
 CN1           | 7A      | Economy
 CN1           | 7B      | Economy
(14 строк)
```

Задание 4.15

```
demo=# UPDATE seats SET fare_conditions = 'Business' WHERE aircraft_code ='319' AND seat_no~'^6|^7|^8' RETURNING seat_no, fare_conditions;
 seat_no | fare_conditions
---------+-----------------
 6A      | Business
 6B      | Business
 6C      | Business
 6D      | Business
 6E      | Business
 6F      | Business
 7A      | Business
 7B      | Business
 7C      | Business
 7D      | Business
 7E      | Business
 7F      | Business
 8A      | Business
 8B      | Business
 8C      | Business
 8D      | Business
 8F      | Business
 8E      | Business
(18 строк)
```

Задание 4.16

```
demo=# Insert into flights (flight_id,flight_no,scheduled_departure,scheduled_arrival,departure_airport,arrival_airport,status,aircraft_code)
demo-# VALUES(1131,12345,current_timestamp+make_interval(days=>7),current_timestamp+make_interval(days=>8),'VKO','VVO','Scheduled','CN1');
ОШИБКА:  повторяющееся значение ключа нарушает ограничение уникальности "flights_pkey"
ПОДРОБНОСТИ:  Ключ "(flight_id)=(1131)" уже существует.
demo=# Insert into flights (flight_id,flight_no,scheduled_departure,scheduled_arrival,departure_airport,arrival_airport,status,aircraft_code)
demo-# VALUES(10113,12345,current_timestamp+make_interval(days=>7),current_timestamp+make_interval(days=>8),'VKO','VVO','Scheduled','CN1');
ОШИБКА:  повторяющееся значение ключа нарушает ограничение уникальности "flights_pkey"
ПОДРОБНОСТИ:  Ключ "(flight_id)=(10113)" уже существует.
demo=# Insert into flights (flight_id,flight_no,scheduled_departure,scheduled_arrival,departure_airport,arrival_airport,status,aircraft_code)
demo-# VALUES(555555,12345,current_timestamp+make_interval(days=>7),current_timestamp+make_interval(days=>8),'VKO','VVO','Scheduled','CN1');
INSERT 0 1
demo=# VALUES(555556,12345,current_timestamp+make_interval(days=>14),current_timestamp+make_interval(days=>15),'VVO','VKO','Scheduled','CN1');
 column1 | column2 |            column3            |            column4            | column5 | column6 |  column7  | column8
---------+---------+-------------------------------+-------------------------------+---------+---------+-----------+---------
  555556 |   12345 | 2022-05-30 11:41:46.694319+03 | 2022-05-31 11:41:46.694319+03 | VVO     | VKO     | Scheduled | CN1
(1 строка)


demo=# Insert into flights (flight_id,flight_no,scheduled_departure,scheduled_arrival,departure_airport,arrival_airport,status,aircraft_code)
demo-# VALUES(555556,12345,current_timestamp+make_interval(days=>14),current_timestamp+make_interval(days=>15),'VVO','VKO','Scheduled','CN1');
INSERT 0 1
demo=#
```

Задание 6.1

```
demo=# BEGIN;
BEGIN
demo=*# insert into bookings (book_ref,book_date,total_amount)
demo-*# VALUES('_12347',current_timestamp,123456);
INSERT 0 1
demo=*# INSERT into tickets (ticket_no,book_ref,passenger_id,passenger_name)
demo-*# VALUES(1348,'_12347',228,'Я');
INSERT 0 1
demo=*# INSERT into tickets (ticket_no,book_ref,passenger_id,passenger_name)
demo-*# VALUES(1349,'_12347',228,'не Я');
INSERT 0 1
demo=*# ROLLBACK;
ROLLBACK
demo=# SELECT * from tickets WHERE book_ref = '_12347';
 ticket_no | book_ref | passenger_id | passenger_name | contact_data
-----------+----------+--------------+----------------+--------------
(0 строк)


demo=#
```

Задание 6.4

```
Введите "help", чтобы получить справку.

demo=# \! chcp 1251
Текущая кодовая страница: 1251
demo=# BEGIN;
BEGIN
demo=*# END;
COMMIT
demo=# set search_path to bookings;
SET
demo=# BEGIN;
BEGIN
demo=*# INSERT INTO bookings(book_ref,book_date,total_amount)
demo-*# VALUES('_13756',current_timestamp,1000);
INSERT 0 1
demo=*# END;
COMMIT
demo=#
```

```
demo=# SELECT * from bookings WHERE total_amount = 1000;
 book_ref | book_date | total_amount
----------+-----------+--------------
(0 строк)


demo=# BEGIN;
BEGIN
demo=*# UPDATE bookings SET total_amount = 2*total_amount where total_amount =1000;
UPDATE 0
demo=*# UPDATE bookings SET total_amount = 2*total_amount where total_amount =1000;
UPDATE 0
demo=*# UPDATE bookings SET total_amount = 2*total_amount where total_amount =1000;
UPDATE 1
demo=*# END;
COMMIT
demo=#
```

Задание 6.5

```
demo=!# END;
ROLLBACK
demo=# BEGIN;
BEGIN
demo=*# INSERT INTO bookings(book_ref,book_date,total_amount)
demo-*# VALUES('_13757',current_timestamp,1000);
INSERT 0 1
demo=*# END;
COMMIT
demo=#
```

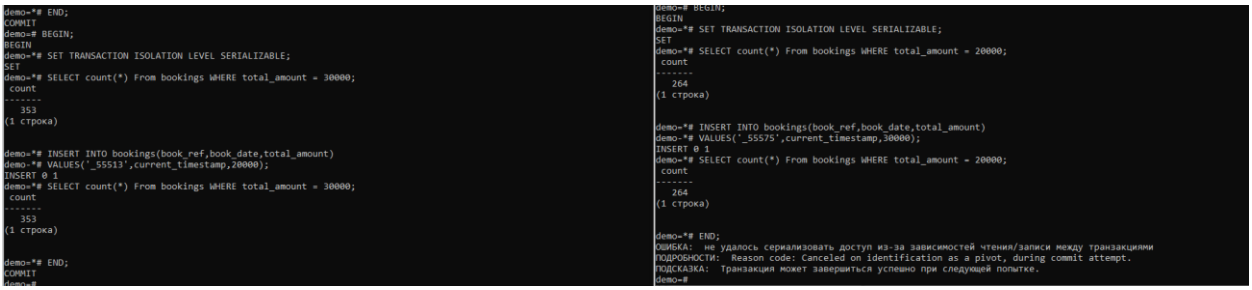
```
BEGIN
demo=*# SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SET
demo=*# UPDATE bookings SET total_amount = 2*total_amount where total_amount =1000;
UPDATE 0
demo=*# UPDATE bookings SET total_amount = 2*total_amount where total_amount =1000;
UPDATE 0
demo=*# END;
COMMIT
demo=#
```

Задание 6.6

```
BEGIN
demo=*# INSERT INTO bookings(book_ref,book_date,total_amount)
demo-*# VALUES('_13757',current_timestamp,1000);
INSERT 0 1
demo=*# END;
COMMIT
demo=# BEGIN;
BEGIN
demo=*# SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SET
demo=*# SELECT count(*) From bookings WHERE total_amount = 30000;
 count
-------
   352
(1 строка)


demo=*# INSERT INTO bookings(book_ref,book_date,total_amount)
demo-*# VALUES('_55556',current_timestamp,20000);
INSERT 0 1
demo=*# SELECT count(*) From bookings WHERE total_amount = 30000;
 count
-------
   352
(1 строка)


demo=*# END;
COMMIT
demo=#
```

```
demo=*# UPDATE bookings SET total_amount = 2*total_amount where total_amount =1000;
UPDATE 0
demo=*# UPDATE bookings SET total_amount = 2*total_amount where total_amount =1000;
UPDATE 0
demo=*# END;
COMMIT
demo=# BEGIN;
BEGIN
demo=*# SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SET
demo=*# SELECT count(*) From bookings WHERE total_amount = 20000;
 count
-------
   263
(1 строка)


demo=*# INSERT INTO bookings(book_ref,book_date,total_amount)
demo-*# VALUES('_55555',current_timestamp,30000);
INSERT 0 1
demo=*# SELECT count(*) From bookings WHERE total_amount = 20000;
 count
-------
   263
(1 строка)


demo=*# END;
COMMIT
demo=#
```

Задание 6.7



```
demo=*# END;
COMMIT
demo=# BEGIN;
BEGIN
demo=*# SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
SET
demo=*# SELECT count(*) From bookings WHERE total_amount = 30000;
 count
-------
   353
(1 строка)


demo=*# INSERT INTO bookings(book_ref,book_date,total_amount)
demo=*# VALUES('_55513',current_timestamp,20000);
INSERT 0 1
demo=*# SELECT count(*) From bookings WHERE total_amount = 30000;
 count
-------
   353
(1 строка)


demo=*# END;
COMMIT
demo=#
```

```
demo=# BEGIN;
BEGIN
demo=*# SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
SET
demo=*# SELECT count(*) From bookings WHERE total_amount = 20000;
 count
-------
   264
(1 строка)


demo=*# INSERT INTO bookings(book_ref,book_date,total_amount)
demo=*# VALUES('_55575',current_timestamp,30000);
INSERT 0 1
demo=*# SELECT count(*) From bookings WHERE total_amount = 20000;
 count
-------
   264
(1 строка)


demo=*# END;
ОШИБКА:  не удалось сериализовать доступ из-за зависимостей чтения/записи между транзакциями
ПОДРОБНОСТИ:  Reason code: Canceled on identification as a pivot, during commit attempt.
ПОДСКАЗКА:  Транзакция может завершиться успешно при следующей попытке.
demo=#
```

Список литературы

1. PostgreSQL. Основы языка SQL: учеб. пособие / Е. П. Моргунов; под ред. Е. В. Рогова, П. В. Лузанова. — СПб.: БХВ-Петербург, 2018. — 336 с.