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 № 2

Тема: «Основы языка определения данных»

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 1

```
demo=# CREATE TABLE students
demo-# ( record_book numeric( 5 ) NOT NULL,
demo(# name text NOT NULL,
demo(# doc_ser numeric( 4 ),
demo(# doc_num numeric( 6 ),
demo(# who_adds_row text DEFAULT current_user,
demo(# upd_time text DEFAULT current_time,
demo(# PRIMARY KEY ( record_book )
demo(# );
CREATE TABLE
demo=# INSERT INTO students ( record_book, name, doc_ser, doc_num )
demo-# VALUES ( 12300, 'Иванов Иван Иванович', 0402, 543281 );
INSERT 0 1
demo=# select * from students
demo-# ;
 record_book |         name         | doc_ser | doc_num | who_adds_row |     upd_time
-------------+----------------------+---------+---------+--------------+-------------------
       12300 | Иванов Иван Иванович |     402 |  543281 | postgres     | 07:40:17.57524+03
(1 строка)
```

Задание 2

```
demo=# \d progress
                        Таблица "bookings.progress"
   Столбец   |     Тип      | Правило сортировки | Допустимость NULL | По умолчанию
-------------+--------------+--------------------+-------------------+--------------
 record_book | numeric(5,0) |                    | not null          |
 subject     | text         |                    | not null          |
 acad_year   | text         |                    | not null          |
 term        | numeric(1,0) |                    | not null          |
 mark        | numeric(1,0) |                    | not null          | 5
Ограничения-проверки:
    "progress_mark_check" CHECK (mark >= 3::numeric AND mark <= 5::numeric)
    "progress_term_check" CHECK (term = 1::numeric OR term = 2::numeric)
```

```
demo=# CREATE TABLE progress
demo-# ( record_book numeric( 5 ) NOT NULL,
demo(# subject text NOT NULL,
demo(# acad_year text NOT NULL,
demo(# term numeric( 1 ) NOT NULL CHECK ( term = 1 OR term = 2 ),
demo(# mark numeric( 1 ) NOT NULL CHECK ( mark >= 3 AND mark <= 5 )
demo(# DEFAULT 5,
demo(# test_form text NOT NULL, CHECK(test_form ='экзамен' OR test_form = 'зачет'),
demo(# FOREIGN KEY ( record_book )
demo(# REFERENCES students ( record_book )
demo(# ON DELETE CASCADE
demo(# ON UPDATE CASCADE
demo(# );
CREATE TABLE
demo=# select * from progress
demo-# ;
 record_book | subject | acad_year | term | mark | test_form
-------------+---------+-----------+------+------+-----------
(0 строк)
```

```
demo-# VALUES ( 12300, 'физика', 'ыыыы',1,5,'экзамен' );
INSERT 0 1
demo=# select * from students
demo-# ;
 record_book |         name         | doc_ser | doc_num | who_adds_row
-------------+----------------------+---------+---------+--------------
       12300 | Иванов Иван Иванович |     402 |  543281 | postgres
(1 строка)


demo=# select * from progress
demo-# ;
 record_book | subject | acad_year | term | mark | test_form
-------------+---------+-----------+------+------+-----------
       12300 | физика  | ыыыы      |    1 |    5 | экзамен
(1 строка)


demo=# VALUES ( 12300, 'физика', 'ыыыы',1,5,'диффзачет' );
 column1 | column2 | column3 | column4 | column5 |  column6
---------+---------+---------+---------+---------+-----------
   12300 | физика  | ыыыы    |       1 |       5 | диффзачет
(1 строка)


demo=# INSERT INTO progress (record_book,subject,acad_year,term,mark,test_form)
demo-# VALUES ( 12300, 'физика', 'ыыыы',1,5,'диффзачет' );
ОШИБКА:  новая строка в отношении "progress" нарушает ограничение-проверку "progress_check"
ПОДРОБНОСТИ:  Ошибочная строка содержит (12300, физика, ыыыы, 1, 5, диффзачет).
demo=#
```

Задание 3

```
demo=# INSERT INTO progress (record_book,subject,acad_year,term,mark,test_form)
demo-# VALUES ( 12300, 'физика', 'ыыыы',NULL ,NULL ,'диффзачет' );
ОШИБКА:  новая строка в отношении "progress" нарушает ограничение-проверку "progress_test_form_check"
ПОДРОБНОСТИ:  Ошибочная строка содержит (12300, физика, ыыыы, null, null, диффзачет).
demo=#
```

Задание 4

```
строка содержит (12300, Физика, 2016/2017, 1, 6, null).
```

Задание 5

```
demo=# INSERT INTO students ( record_book, name, doc_ser, doc_num )
demo-# VALUES ( 12301, 'Иванов Иван Иванович', NULL, NULL );
INSERT 0 1
demo=# INSERT INTO students ( record_book, name, doc_ser, doc_num )
demo-# VALUES ( 12302, 'Иванов Иван Иванович', NULL, NULL );
INSERT 0 1
demo=# select * from students;
 record_book |         name         | doc_ser | doc_num | who_adds_row |      upd_time
-------------+----------------------+---------+---------+--------------+--------------------
       12300 | Иванов Иван Иванович |     402 |  543281 | postgres     | 08:39:09.616648+03
       12301 | Иванов Иван Иванович |         |         | postgres     | 08:40:48.824694+03
       12302 | Иванов Иван Иванович |         |         | postgres     | 08:41:02.632655+03
(3 строки)


demo=# INSERT INTO students ( record_book, name, doc_ser, doc_num )
demo-# VALUES ( 12303, 'Иванов Иван Иванович', 402, 543281 );
ОШИБКА:  повторяющееся значение ключа нарушает ограничение уникальности "students_doc_ser_key"
ПОДРОБНОСТИ:  Ключ "(doc_ser)=(402)" уже существует.
demo=#
```

Задание 6

```
demo-# VALUES ( 12301, 'Иванов Иван Петрович', 0403, 543282 );
INSERT 0 1
demo=# INSERT INTO progress ( record_book, subject, acad_year, term )
demo-# VALUES ( 12300, 'Физика', '2016/2017',1 );
ОШИБКА:  столбец "record_book" в таблице "progress" не существует
СТРОКА 1: INSERT INTO progress ( record_book, subject, acad_year, term...
                                 ^
demo=# INSERT INTO progress ( doc_ser,doc_num, subject, acad_year, term,mark )
demo-# VALUES ( 0402,543281, 'Физика', '2016/2017',1,5);
INSERT 0 1
demo=# INSERT INTO progress ( doc_ser,doc_num, subject, acad_year, term,mark )
demo-# VALUES ( 0403,543282, 'Физика', '2016/2017',1,4);
INSERT 0 1
demo=# select * from students;
 record_book |         name         | doc_ser | doc_num
-------------+----------------------+---------+---------
       12300 | Иванов Иван Иванович |     402 |  543281
       12301 | Иванов Иван Петрович |     403 |  543282
(2 строки)


demo=# select * from progress;
 doc_ser | doc_num | subject | acad_year | term | mark
---------+---------+---------+-----------+------+------
     402 |  543281 | Физика  | 2016/2017 |    1 |    5
     403 |  543282 | Физика  | 2016/2017 |    1 |    4
(2 строки)
```

Задание 8

```
demo=#
demo=# INSERT INTO progress ( doc_ser,doc_num, subject_id, acad_year, term,mark )
demo-# VALUES(402,543281,1,'2017',1,5);
INSERT 0 1
demo=# INSERT INTO progress ( doc_ser,doc_num, subject_id, acad_year, term,mark )
demo-# VALUES(403,543281,2,'2017',1,5);
ОШИБКА:  INSERT или UPDATE в таблице "progress" нарушает ограничение внешнего ключа "progress_doc_ser_doc_num_fkey"
ПОДРОБНОСТИ:  Ключ (doc_ser, doc_num)=(403, 543281) отсутствует в таблице "students".
demo=#
```

Задание 9

```
demo=# ALTER TABLE students ADD CHECK(trim(both ' ' from name) <> '');
ALTER TABLE
demo=# INSERT INTO students ( record_book, name, doc_ser, doc_num )
demo-# VALUES ( 12302, '        ', 0404, 543283 );
ОШИБКА:  новая строка в отношении "students" нарушает ограничение-проверку "students_name_check1"
ПОДРОБНОСТИ:  Ошибочная строка содержит (12302,         , 404, 543283).
```

Задание 10

```
demo=# ALTER TABLE students ALTER COLUMN doc_ser SET DATA TYPE character;
ОШИБКА:  ограничение внешнего ключа "progress_doc_ser_doc_num_fkey" нельзя реализовать
ПОДРОБНОСТИ:  Столбцы ключа "doc_ser" и "doc_ser" имеют несовместимые типы: numeric и character.
demo=#
```

Задание 17

```
demo=# CREATE VIEW long_range AS
demo-# SELECT DISTINCT * FROM aircrafts
demo-# WHERE range > 5999;
CREATE VIEW
demo=# SELECT * FROM long_range;
 aircraft_code |      model      | range
---------------+-----------------+-------
 319           | Airbus A319-100 |  6700
 763           | Boeing 767-300  |  7900
 773           | Boeing 777-300  | 11100
(3 строки)
```

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.