

Санкт-Петербургский Политехнический Университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Базы данных

Отчет по лабораторной работе №2
Язык SQL-DDL

Работу
выполнил:
Графов Д.И.
Группа: 33531/2
Преподаватель:
Мяснов А.В.

Санкт-Петербург
2019

Содержание

1. Цель работы	2
2. Программа работы	2
3. Теоретическая информация	2
3.1. Язык SQL	2
3.2. Типы данных	2
4. Выполнение работы	3
4.1. Инициализация	3
4.2. Изменение	5
4.3. Заполнение	6
5. Выводы	8

1. Цель работы

Познакомиться с основами проектирования схемы БД, языком описания сущностей и ограничений БД SQL-DDL.

2. Программа работы

1. Самостоятельное изучение SQL-DDL.
2. Создание скрипта БД в соответствии с согласованной схемой. Должны присутствовать первичные и внешние ключи, ограничения на диапазоны значений. Демонстрация скрипта преподавателю.
3. Создание скрипта, заполняющего все таблицы БД данными.
4. Выполнение SQL-запросов, изменяющих схему созданной БД по заданию преподавателя. Демонстрация их работы преподавателю.

3. Теоретическая информация

3.1. Язык SQL

Язык SQL (Structured Query Language) - язык структурированных запросов. Он позволяет формировать весьма сложные запросы к базам данных. В SQL определены два подмножества языка:

- SQL-DDL (Data Definition Language) - язык определения структур и ограничений целостности баз данных. Сюда относятся команды создания и удаления баз данных; создания, изменения и удаления таблиц; управления пользователями и т.д.
- SQL-DML (Data Manipulation Language) - язык манипулирования данными: добавление, изменение, удаление и извлечение данных, управления транзакциями

3.2. Типы данных

Символьные типы данных:

- CHAR(n) - символьные строки фиксированной длины. Длина строки определяется параметром n. CHAR без параметра соответствует CHAR(1). Для хранения таких данных всегда отводится n байт вне зависимости от реальной длины строки.
- VARCHAR(n) - символьная строка переменной длины. Для хранения данных этого типа отводится число байт, соответствующее реальной длине строки.

Целые типы данных:

- SMALLINT - короткое целое (2 байта)
- INTEGER - обычное целое (4 байта)
- BIGINT - длинное целое (8 байт)

Вещественные типы данных:

- REAL - числа с плавающей точкой (4 байта).
- DOUBLE PRECISION - числа с плавающей точкой (8 байт).
- NUMERIC(p,n) - тип данных аналогичный FLOAT с числом значащих цифр p и точностью n.

Дата и время - используются для хранения даты, времени и их комбинаций:

- DATE - тип данных для хранения даты.
- TIME - тип данных для хранения времени.
- TIMESTAMP - тип данных для хранения моментов времени (год + месяц + день + часы + минуты + секунды + доли секунд).

Двоичные типы данных - позволяют хранить данные любого объема в двоичном коде (оцифрованные изображения, исполняемые файлы и т.д.):

- BYTEA

4. Выполнение работы

4.1. Инициализация

```

1 DROP SCHEMA IF EXISTS public CASCADE;
2 CREATE SCHEMA public;
3
4 CREATE TYPE drink_type AS ENUM (
5     'пиво и сидр',
6     'водка и настойки',
7     'ром',
8     'текила',
9     'виски',
10    'биттеры',
11    'вино',
12    'бренди и коньяк',
13    'шоты',
14    'коктейли',
15    'безалкогольные'
16 );
17
18 CREATE TABLE components
19 (
20     component_id serial NOT NULL,
21     title character varying(50) NOT NULL,
22     alcohol double precision,
23     primary key (component_id)
24 );
25
26 CREATE TABLE drinks
27 (

```

```

28     drink_id      serial              NOT NULL,
29     title         character varying(50) NOT NULL,
30     rating        double precision,
31     volume        double precision,
32     alcohol       double precision,
33     average_price double precision,
34     drink_type    drink_type,
35     primary key (drink_id)
36 );
37
38 CREATE TABLE components_drinks
39 (
40     component_id int NOT NULL references components (component_id),
41     drink_id     int NOT NULL references drinks (drink_id),
42     quantity     double precision
43 );
44
45 CREATE TABLE places
46 (
47     place_id      serial              NOT NULL,
48     title         character varying(50) NOT NULL,
49     address       text,
50     rating        double precision,
51     average_bill double precision,
52     primary key (place_id)
53 );
54
55 CREATE TABLE places_drinks
56 (
57     place_id integer NOT NULL references places (place_id),
58     drink_id integer NOT NULL references drinks (drink_id)
59 );
60
61 CREATE TABLE discounts
62 (
63     discount_id serial              NOT NULL,
64     place_id    integer            NOT NULL references places (place_id),
65     drink_type  drink_type         NOT NULL,
66     amount      double precision NOT NULL,
67     description text,
68     weekday     integer,
69     time_start  time,
70     time_end    time,
71     primary key (discount_id)
72 );
73
74 CREATE TABLE food
75 (
76     food_id      serial              NOT NULL,
77     title         character varying(50) NOT NULL,

```

```

78     rating          double precision,
79     volume          double precision,
80     average_price double precision,
81     primary key (food_id)
82 );
83
84 CREATE TABLE places_food
85 (
86     place_id integer NOT NULL references places (place_id),
87     snack_id integer NOT NULL references food (food_id)
88 );

```

4.2. Изменение

```

1  --ISSUE
2  alter table places_drinks
3      add column price double precision;
4
5  alter table places_food
6      add column price double precision;
7
8  alter table places_drinks
9      add column amount int;
10
11 alter table places_food
12     add column amount int;
13
14 create table supplies_food
15 (
16     supplies_snacks_id serial not null,
17     place_id           int references places (place_id),
18     food_id            int references food (food_id),
19     amount              int,
20     price_per_item     double precision,
21     date                timestamp,
22     primary key (supplies_snacks_id)
23 );
24
25 create table supplies_drinks
26 (
27     supplies_drinks_id serial not null,
28     place_id           int references places (place_id),
29     drink_id           int references drinks (drink_id),
30     amount              int,
31     price_per_item     double precision,
32     date                timestamp,
33     primary key (supplies_drinks_id)
34 );

```

4.3. Заполнение

```
1  --FILL
2  insert into components(title, alcohol
3  )
4  values ('водка', 40),
5         ('гренадин', 0),
6         ('соус табаско', 0);
7
8  insert into drinks(title, volume, alcohol, drink_type
9  )
10 values ('боярский', 50, 40 / 2, 'шоты');
11
12 insert into components_drinks(component_id, drink_id, quantity
13 )
14 values ((select component_id from components where title = 'водка'),
15         (select drink_id from drinks where title = 'боярский'), 1),
16        ((select component_id from components where title = 'гренадин'),
17         (select drink_id from drinks where title = 'боярский'), 1),
18        ((select component_id from components where title = 'соус табаско'),
19         (select drink_id from drinks where title = 'боярский'), 1);
20
21
22 insert into components(title, alcohol
23 )
24 values ('томатный сок', 0);
25
26 insert into drinks(title, volume, alcohol, drink_type
27 )
28 values ('кровавая мэри', 50, 40 / 2, 'шоты');
29
30 insert into components_drinks(component_id, drink_id
31 )
32 values ((select component_id from components where title = 'водка'),
33         (select drink_id from drinks where title = 'кровавая мэри')),
34        ((select component_id from components where title = 'томатный сок'),
35         (select drink_id from drinks where title = 'кровавая мэри')),
36        ((select component_id from components where title = 'соус табаско'),
37         (select drink_id from drinks where title = 'кровавая мэри'));
38
39 insert into places(title, address
40 )
41 values ('Контакт бар', 'пр. Просвещения, 25'),
42        ('Контакт бар', 'ул. Садовая, 35'),
43        ('Контакт бар', 'пр. Владимирский, 17'),
44        ('Контакт бар', 'пр. Коломяжский, 15, к.2'),
45        ('Контакт бар', 'пл. Стачек, 7, лит. А'),
46        ('Контакт бар', 'ул. Гаккелевская, 34'),
47        ('Контакт бар', 'ул. Марата, 7'),
48        ('Контакт бар', 'Средний пр-т. ВО, 28'),
```

```

49         ('Контакт бар', 'пр.Чернышевского 11/57'),
50         ('Контакт бар', 'ул. Бухарестская, 74'),
51         ('СПБ бар', 'Каменноостровский пр., 37');
52
53 insert into food(title
54 )
55 values ('стрипсы'),
56         ('начос');
57
58 insert into places_drinks(place_id, drink_id
59 )
60 select distinct place_id, drink_id
61 from places,
62      drinks
63 where drinks.title = 'боярский'
64        and places.title = 'Контакт бар';
65
66 insert into places_food(place_id, snack_id
67 )
68 select distinct place_id, food_id
69 from places,
70      food
71 where food.title = 'стрипсы'
72        and places.title = 'Контакт бар';
73
74 insert into places_food(place_id, snack_id
75 )
76 select distinct place_id, food_id
77 from places,
78      food
79 where food.title = 'начос'
80        and places.title = 'Контакт бар';
81
82 insert into supplies_food(place_id, food_id
83 )
84 select distinct place_id, food_id
85 from places,
86      food
87 where food.title = 'начос'
88        and places.title = 'Контакт бар';
89
90 insert into supplies_drinks(place_id, drink_id
91 )
92 select distinct place_id, drink_id
93 from places,
94      drinks
95 where drinks.title = 'боярский'
96        and places.title = 'Контакт бар';
97
98 insert into discounts(place_id, drink_type, amount, description

```



```
99 )
100 select distinct place_id, 'пиво и сидр'::drink_type, 0.15, 'Цены для друзей на пиво!'
101 from places
102 where places.title = 'Контакт бар';
```

5. Выводы

В ходе выполнения данной работы были написаны 3 скрипта на языке SQL: создающий таблицы; изменяющий таблицы и добавляющий новые; скрипт, наполняющий таблицы данными. Таким образом, было осуществлено моё знакомство с основами проектирования схемы БД SQL.