

Университет ИТМО
Факультет программной инженерии и компьютерной техники

Распределённые системы хранения данных. Лабораторная работа №1.

Группа: Р33131
Студент: Смирнов Виктор Игоревич
Преподаватель: Афанасьев Дмитрий Борисович
Вариант: 776

Ключевые слова

База данных, PostgreSQL, системный каталог.

Содержание

1	Цель работы	1
2	Текст задания	1
3	Создание локального окружения для тестирования	2
4	Реализация скрипта	2
5	Вывод	3
6	Вывод	4

1 Цель работы

Научиться проектировать базы данных, составлять инфологические и даталогические модели данных, реализовывать их в БД PostgreSQL, научиться выполнять запросы.

2 Текст задания

Используя сведения из системных каталогов получить информацию о любой таблице: Номер по порядку, Имя столбца, Атрибуты (в атрибуты столбца включить тип данных, ограничение типа CHECK).

Пример вывода:

Таблица: Н_ЛЮДИ

Но.	Имя столбца	Атрибуты
1	ИД	Type : NUMBER(9) NOT NULL Comment : 'Уникальный номер человека'
2	ФАМИЛИЯ	Type : VARCHAR2(25) NOT NULL Comment : 'Фамилия человека'
3	ИМЯ	Type : VARCHAR2(2000) NOT NULL Comment : 'Имя человека'
4	ОТЧЕСТВО	Type : VARCHAR2(20) Comment : 'Отчество человека'
5	ДАТА_РОЖДЕНИЯ	Type : DATE NOT NULL Comment : 'Дата рождения человека'
6	ПОЛ	Type : CHAR(1) NOT NULL Constr : "AVCON_378561_ПОЛ_000" CHECK (ПОЛ IN ('М', 'Ж')) Constr : "AVCON_388176_ПОЛ_000" CHECK (ПОЛ IN ('М', 'Ж')) Comment : 'Пол человека'
7	ИНОСТРАН	Type : VARCHAR2(3) NOT NULL
8	КТО СОЗДАЛ	Type : VARCHAR2(40) NOT NULL
9	КОГДА СОЗДАЛ	Type : DATE NOT NULL
10	КТО ИЗМЕНИЛ	Type : VARCHAR2(40) NOT NULL
11	КОГДА ИЗМЕНИ	Type : DATE NOT NULL
12	ДАТА СМЕРТИ	Type : DATE Comment : 'Дата смерти человека'
13	ПИН	Type : VARCHAR2(20)
14	ИНН	Type : VARCHAR2(20)

3 Создание локального окружения для тестирования

Для локального взаимодействия я решил поднять PostgreSQL в Docker контейнере.

```
1 version: '3.8'
2 services:
3   database:
4     container_name: database
5     image: postgres
6     restart: always
7     ports:
8       - 5432:5432
9     volumes:
10      - ../workspace
11     environment:
12       POSTGRES_USER: postgres
13       POSTGRES_PASSWORD: postgres
14       POSTGRES_DB: postgres
15     networks:
16       - common
17 networks:
18   common:
19     name: common
20     driver: bridge
```

Далее был написан SQL скрипт, создающий таблицу, аналогичную той, что в примере.

```
1 drop table person;
2 create table person (
3   id numeric(9, 2) primary key,
4   last_name varchar(25) not null,
5   first_name varchar(2000) not null,
6   patronymic varchar(20),
7   birth_date date not null,
8   gender char(1) not null,
9   foreigner varchar(3) not null,
10  created_who varchar(40) not null,
11  created_when date not null,
12  edited_who varchar(40) not null,
13  edited_when date not null,
14  death_date date,
15  pin varchar(20),
16  inn varchar(20),
17
18  check (gender in ('M', 'F')),
19  check (gender in ('M', 'F'))
20 );
21
22 comment on column person.id is 'The unique number of the person';
23 comment on column person.id is 'The unique number of the person';
24 comment on column person.last_name is 'Last name of the person';
25 comment on column person.first_name is 'The name of the person';
26 comment on column person.patronymic is 'The patronymic of the person';
27 comment on column person.birth_date is 'Date of birth of a person';
28 comment on column person.death_date is 'Date of death of a person';
```

4 Реализация скрипта

```
1 drop procedure IF EXISTS solution;
2 create or replace procedure solution(
3   table_name text
4 ) as $$
5 declare
6   table_schema text;
7 begin
8   select information_schema.tables.table_schema into table_schema
9   from information_schema.tables
10  where information_schema.tables.table_name = solution.table_name
11  limit 1;
12
13  call print_table_info(table_schema, table_name);
```

```

14 end;
15 $$ language plpgsql;
16
17 call solution('person');

```

5 ВЫВОД

```

1 psql:get_table_info.sql:249: INFO: |--- Table "public.person" Information
   |-----|
2 psql:get_table_info.sql:249: INFO: | N | Name | Attributes
   |
3 psql:get_table_info.sql:249: INFO: |-----|
   |
4 psql:get_table_info.sql:249: INFO: | 1 | id | Type : numeric (9, 2)
   |
5 psql:get_table_info.sql:249: INFO: | | | Comment : The unique number of
   | the person
6 psql:get_table_info.sql:249: INFO: |-----|
   |
7 psql:get_table_info.sql:249: INFO: | 2 | last_name | Type : character varying
   | (25)
8 psql:get_table_info.sql:249: INFO: | | | Comment : Last name of the
   | person
9 psql:get_table_info.sql:249: INFO: |-----|
   |
10 psql:get_table_info.sql:249: INFO: | 3 | first_name | Type : character varying
   | (2000)
11 psql:get_table_info.sql:249: INFO: | | | Comment : The name of the
   | person
12 psql:get_table_info.sql:249: INFO: |-----|
   |
13 psql:get_table_info.sql:249: INFO: | 4 | patronymic | Type : character varying
   | (20)
14 psql:get_table_info.sql:249: INFO: | | | Comment : The patronymic of
   | the person
15 psql:get_table_info.sql:249: INFO: |-----|
   |
16 psql:get_table_info.sql:249: INFO: | 5 | birth_date | Type : date
   |
17 psql:get_table_info.sql:249: INFO: | | | Comment : Date of birth of a
   | person
18 psql:get_table_info.sql:249: INFO: |-----|
   |
19 psql:get_table_info.sql:249: INFO: | 6 | gender | Type : character (1)
   |
20 psql:get_table_info.sql:249: INFO: | | | Constr : person_gender_check
   | ((gender = ANY (ARRAY['M'::bpchar, 'F'::bpchar])))
21 psql:get_table_info.sql:249: INFO: | | | Constr : person_gender_check1
   | ((gender = ANY (ARRAY['M'::bpchar, 'F'::bpchar])))
22 psql:get_table_info.sql:249: INFO: |-----|
   |
23 psql:get_table_info.sql:249: INFO: | 7 | foreigner | Type : character varying
   | (3)
24 psql:get_table_info.sql:249: INFO: |-----|
   |
25 psql:get_table_info.sql:249: INFO: | 8 | created_who | Type : character varying
   | (40)
26 psql:get_table_info.sql:249: INFO: |-----|
   |
27 psql:get_table_info.sql:249: INFO: | 9 | created_when | Type : date
   |
28 psql:get_table_info.sql:249: INFO: |-----|
   |

```

```

29 psql:get_table_info.sql:249: INFO: | 10 | edited_who | Type : character varying
    (40)
30 psql:get_table_info.sql:249: INFO:
    |-----|
31 psql:get_table_info.sql:249: INFO: | 11 | edited_when | Type : date
    |
32 psql:get_table_info.sql:249: INFO:
    |-----|
33 psql:get_table_info.sql:249: INFO: | 12 | death_date | Type : date
    |
34 psql:get_table_info.sql:249: INFO: | | | Comment : Date of death of a
    person
35 psql:get_table_info.sql:249: INFO:
    |-----|
36 psql:get_table_info.sql:249: INFO: | 13 | pin | Type : character varying
    (20)
37 psql:get_table_info.sql:249: INFO:
    |-----|
38 psql:get_table_info.sql:249: INFO: | 14 | inn | Type : character varying
    (20)
39 psql:get_table_info.sql:249: INFO:
    |-----|

```

6 Вывод

Данная лабораторная работа помогла мне изучить системный каталог PostgreSQL.

Список литературы