

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра автоматизированных систем управления (АСУ)

РАБОТА С УТИЛИТОЙ PSQL

Отчёт о лабораторной работе № 1
по дисциплине «Базы данных»

Выполнил: студент гр. 430-2

_____ Лузинсан А.А.

«___»_____ 2022 г.

Проверил: ассистент каф. АСУ

_____ Яблонский Я. В.

«___»_____ 2022 г.

Томск 2022

Оглавление

1 ВВЕДЕНИЕ.....	2
2 ОПИСАНИЕ БАЗЫ ДАННЫХ.....	3
3 ХОД РАБОТЫ.....	5
4 ЗАКЛЮЧЕНИЕ.....	9

1 ВВЕДЕНИЕ

Цель работы: используя psql создать таблицы T_Address, T_Pod, T_PubHouse и ввести в таблицы тестовые данные (1-2 строки). Убедиться в том, что поддерживаются ограничения целостности атрибута, сущности и ссылочные.

2 ОПИСАНИЕ БАЗЫ ДАННЫХ

Базой данных, исходя из индивидуального задания, является «Подписка на периодические издания», которая представляет собой автоматизированную информационную систему «Подписка» предназначенную для автоматизации деятельности почтового служащего, оформляющего подписку на периодические издания.

АИС поддерживает следующие функции:

- накопление первичной информации о периодических изданиях, подписчиках и подписке;
- оформление подписки;
- генерация отчетов заданных форматов.

Структура базы данных «Подписка» в данной лабораторной работе включает таблицы T_Address, T_Pod и T_PubHouse, свойства которых описаны в таблицах 2.1 — 2.3 настоящего отчёта.

Таблица 2.1 — T_Address - Справочник адресов

Имя	Смысл	Тип	Длина	Свойства
ID	Идентификатор записи	Счётчик		Первичный ключ таблицы
Post_code	Индекс	Строка	6	«xxxxxx» - 6 цифр
Region	Область/Край	Строка	30	Обязательное значение.
City	Город	Строка	30	Обязательное значение.
Street	Улица	Строка	50	Обязательное значение.
House	Дом	Целое		Больше нуля
Apartment	Квартира/Офис	Целое		Больше нуля или NULL.

Таблица 2.2 — T_Pod - Таблица подписки

Имя	Смысл	Тип	Длина	Свойства
ID	Уникальный индекс подписчика	Счётчик		Первичный ключ
FIO	ФИО	Строка	85	Обязательное значение.
IDAddr	Адрес подписчика	Целое		Ссылка на первичный ключ таблицы T_Address

Таблица 2.3 — T_PubHouse - Сведения об издательстве

Имя	Смысл	Тип	Длина	Свойства
ID	Индекс издательства	Строка	7	От 2 до 7 цифр, первичный ключ таблицы
Name	Наименование	Строка	50	Обязательное значение.
IDAddr	Адрес издательства	Целое		Ссылка на первичный ключ таблицы T_Address
Phone	Контактный телефон	Строка	15	Вводится в формате «(xxxx) xx-xx-xx», где x- цифра

3 ХОД РАБОТЫ

Чтобы пользоваться утилитой `psql` в терминале `linux` первым делом необходимо подключить сервер `postgres` командой **`sudo -i -u postgres`**. Вписываем пароль текущего пользователя и далее указываем команду **`psql`** для входа в утилиту.

Создаём базу данных командой: **`CREATE DATABASE subscription`** - и переключаемся на данную базу данных с помощью команды: `\c subscription`

Теперь создаём первую таблицу согласно заданию — `T_Address`. SQL-код данного запроса представлен далее:

```
CREATE TABLE T_Address(  
ID SERIAL PRIMARY KEY,  
Post_code VARCHAR(6) CHECK(post_code SIMILAR TO '\d{6}'),  
Region VARCHAR(30) NOT NULL,  
City VARCHAR(30) NOT NULL,  
Street VARCHAR(50) NOT NULL,  
House INTEGER CHECK(House > 0),  
Apartment INTEGER CHECK(Apartment > 0 OR NULL)  
);
```

В данном отношении:

- первичным ключом является атрибут `ID`, свойство которого указывается с помощью ограничения `PRIMARY KEY`;
- атрибуты `Region`, `City` и `Street` должны быть обязательно заполнены пользователем, поэтому они обладают ограничением `NOT NULL`;
- атрибут `House` принимает натуральные значения, что ограничивается условием `CHECK(House > 0)`, тогда как атрибут `Apartment` может не указываться явно, что добавляет ему дополнительное условие. В совокупности с предыдущим атрибут `Apartment` имеет ограничение целостности `CHECK(Apartment > 0 OR NULL)`;
- атрибут `Post_code` должен иметь в своей строке 6 цифр. Это ограничение легко задать с помощью регулярного выражения `'\d{6}'`. Тогда конечное условие будет выглядеть так: `CHECK(post_code SIMILAR TO '\d{6}')`.

На следующем этапе была создана таблица T_Pod, SQL-код запроса которой представлен далее:

```
CREATE TABLE T_Pod(  
ID SERIAL PRIMARY KEY UNIQUE,  
FIO VARCHAR(85) NOT NULL,  
IDAddr INTEGER REFERENCES t_address(ID)  
);
```

Таблица «Подписчики» представляет собой совокупность атрибутов:

- атрибут ID выступает первичным ключом таблицы и задаётся также, как и в предыдущей таблице, ограничением PRIMARY KEY. Помимо этого в описании свойств атрибута фигурирует слово уникальность, которое можно задать ограничением UNIQUE;
- атрибут FIO представляет собой символьную строку длиной 85, являющейся обязательной, что указывается ограничением NOT NULL;
- атрибут IDAddr предназначен для указания адреса подписчика, являющийся целым значением и ссылкой на первичный ключ таблицы T_Address. Зависимость от внешнего отношения задаётся с помощью REFERENCES t_address(ID).

Последняя таблица представляет собой сведения об издательстве, расположенные в T_PubHouse, создание которой реализуется следующим SQL-запросом:

```
CREATE TABLE T_PubHouse(  
ID VARCHAR(7) PRIMARY KEY CHECK(ID SIMILAR TO '\d{2,7}'),  
Name VARCHAR(50) NOT NULL,  
IDAddr INTEGER REFERENCES t_address(id),  
Phone VARCHAR(15) CHECK(Phone SIMILAR TO '(\d{4})\d{2}-\d{2}-\d{2}'))  
);
```

Атрибуты таблицы описаны далее:

- атрибут ID указывает на индекс издательства, который представляет собой строку от 2 до 7 цифр первичного ключа. Данное условие легко задать с помощью регулярного выражения и получить следующее: CHECK(ID

SIMILAR TO '\d{2,7}'), а также указывается ограничение первичного ключа: PRIMARY KEY;

- атрибут Name определяет наименование издательства и является обязательным значением, которое указывается ограничением NOT NULL;
- атрибут IDAddr позволяет определить местоположение издательства как ссылку на первичный ключ таблицы T_Address, реализуемую с помощью REFERENCES t_address(id);
- атрибут Phone представляет собой строку формата «(xxxx) xx-xx-xx», вид которой легко задать с помощью регулярного выражения и получить следующее ограничение CHECK(Phone SIMILAR TO '\(\d{4}\) \d{2}-\d{2}-\d{2}'). Экранирование символов „(“ и „)“ является обязательным на ОС Linux.

Таким образом, в результате выполнения оных SQL-запросов, получаем таблицы, свойства которых изображены на рисунках 3.1 — 3.3.

Таблица "public.t_address"				
Столбец	Тип	Правило сортировки	Допустимость NULL	По умолчанию
id	integer		not null	nextval('t_address_id_seq'::regclass)
post_code	character varying(6)			
region	character(30)		not null	
city	character(30)		not null	
street	character(50)		not null	
house	integer			
apartment	integer			
Индексы:				
"t_address_pkey" PRIMARY KEY, btree (id)				
Ограничения-проверки:				
"t_address_apartment_check" CHECK (apartment > 0 OR NULL::boolean)				
"t_address_house_check" CHECK (house > 0)				
Ссылки извне:				
TABLE "t_pod" CONSTRAINT "t_pod_idaddr_fkey" FOREIGN KEY (idaddr) REFERENCES t_address(id)				
TABLE "t_pubhouse" CONSTRAINT "t_pubhouse_idaddr_fkey" FOREIGN KEY (idaddr) REFERENCES t_address(id)				

Рисунок 3.1 — Свойства таблицы T_Address

Таблица "public.t_pod"				
Столбец	Тип	Правило сортировки	Допустимость NULL	По умолчанию
id	integer		not null	nextval('t_pod_id_seq'::regclass)
fio	character(85)		not null	
idaddr	integer			
Индексы:				
"t_pod_pkey" PRIMARY KEY, btree (id)				
Ограничения внешнего ключа:				
"t_pod_idaddr_fkey" FOREIGN KEY (idaddr) REFERENCES t_address(id)				

Рисунок 3.2 — Свойства таблицы T_Pod

Таблица "public.t_pubhouse"				
Столбец	Тип	Правило сортировки	Допустимость NULL	По умолчанию
id	character(7)		not null	
name	character(50)		not null	
idaddr	integer			
phone	character(15)			

Индексы:
 "t_pubhouse_pkey" PRIMARY KEY, btree (id)

Ограничения-проверки:
 "t_pubhouse_id_check" CHECK (char_length(id) > 2)
 "t_pubhouse_phone_check" CHECK (phone ~ similar_to_escape('(\d{4}) \d{2}-\d{2}-\d{2}'))

Ограничения внешнего ключа:
 "t_pubhouse_idaddr_fkey" FOREIGN KEY (idaddr) REFERENCES t_address(id)

Рисунок 3.3 — Свойства таблицы T_PubHouse

Чтобы проверить правильность составленных таблиц, были выполнены следующие запросы:

```
INSERT INTO t_address(post_code, region, city, street, house, apartment)
VALUES('634045', 'Томская Область', 'Томск', 'Нахимова', 20, 52);
INSERT INTO t_address(post_code, region, city, street, house, apartment)
VALUES('634043', 'Томская Область', 'Томск', 'Вершинина', 74, 438);
INSERT INTO t_pod(fio, idaddr) VALUES('Иван Иванович', 2);
INSERT INTO t_pubhouse(id, name, idaddr, phone) VALUES('29402',
'Habr', 1, '(8322) 20-28-10');
```

id	post_code	region	city	street	house	apartment
1	634045	Томская Область	Томск	Нахимова	20	52
2	634043	Томская Область	Томск	Вершинина	74	438

(2 строки)

Рисунок 3.4 — Содержимое таблицы T_Address

id	fio	idaddr
1	Иван Иванович	2

(1 строка)

Рисунок 3.5 — Содержимое таблицы T_Pod

id	name	idaddr	phone
29402	Habr	1	(8322) 20-28-10

(1 строка)

Рисунок 3.6 — Содержимое таблицы T_PubHouse

4 ЗАКЛЮЧЕНИЕ

В результате выполнения данной лабораторной работы я научилась пользоваться утилитой `psql` и создавать таблицы баз данных, указывая ограничения целостности с помощью `PRIMARY KEY`, `NULL\NOT NULL`, `UNIQUE`, `REFERENCES` и `CHECK`, а также проверила правильность созданных таблиц на примере тестовых данных.