

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. 431-3

_____ Бекиш Е.П.

« ____ » _____ 2023 г.

Проверил: ассистент каф. АСУ

_____ Яблонский Я. В.

« ____ » _____ 2023 г.

Томск 2023

1 Цель лабораторной работы

- ознакомиться с принципами работы консольной утилиты `psql`;
- научиться создавать макеты таблиц с использованием команд SQL.

2 Описание таблиц БД из индивидуального задания

На факультете существует группа студентов, обучаемых по индивидуальным планам. Все студенты проживают в общежитии. Некоторые совмещают учёбу с работой. Каждый студент в течение семестра изучает несколько учебных дисциплин. До окончания семестра студент должен отчитаться по каждой изучаемой дисциплине. Единственный предусмотренный вид отчётности — экзамен. Один из преподавателей факультета курирует группу. В его обязанности входит текущий и итоговый контроль успеваемости студентов. Для успешного выполнения обязанностей куратору нужно всегда иметь под рукой сведения о студентах и об их успеваемости.

На рисунке 2.1 представлена структура базы данных, соответствующая индивидуальному заданию.

СТРУКТУРА БАЗЫ ДАННЫХ

База данных состоит из следующих таблиц.

STUDENT – Сведения о студентах

Имя	Смысл	Тип	Длина	Свойства
Snum	Номер студенческого билета.	Строка	6	Только цифры. Первичный ключ таблицы.
Snam	Фамилия, имя, отчество студента.	Строка	40	Только буквы русского алфавита и пробелы.
Ann	Год обучения.	Целое		Допустимые значения {1, 2, 3, 4, 5}
Stel	Номер индивидуального телефона студента.	Строка	10	Только цифры. Значения уникальны.
Rnum	Номер комнаты, в которой проживает студент.	Строка	3	Ссылка на первичный ключ таблицы ROOM.

ROOM – Сведения о комнатах, в которых проживают студенты

Имя	Смысл	Тип	Длина	Свойства
Rnum	Номер комнаты.	Строка	3	Только цифры. Первичный ключ таблицы.
Tnum	Номер телефона, установленного в комнате.	Строка	6	Только цифры. Значения уникальны.

Рисунок 2.1 – структура базы данных

3 Описание процесса реализации таблиц

Первым делом для занесения данных в таблицы, необходимо их создать. Для этого используем команду `CREATE DATABASE`, которая позволяет создать новую базу данных.

В наших таблицах присутствуют свойства, где задаются правила проверки значений поля с помощью команды `CHECK`. При внесении данных СУБД отбросит все строки, где значения в столбцах не проходят проверку `CHECK`. Но некоторые свойства, которые имеют тип данных строка, могут содержать определенные символы, которые нужно проверить, поэтому мы можем использовать регулярные выражения. Они представлены в виде шаблона ... `CHECK(имя_поля SIMILAR TO 'регулярное_выражение')`.

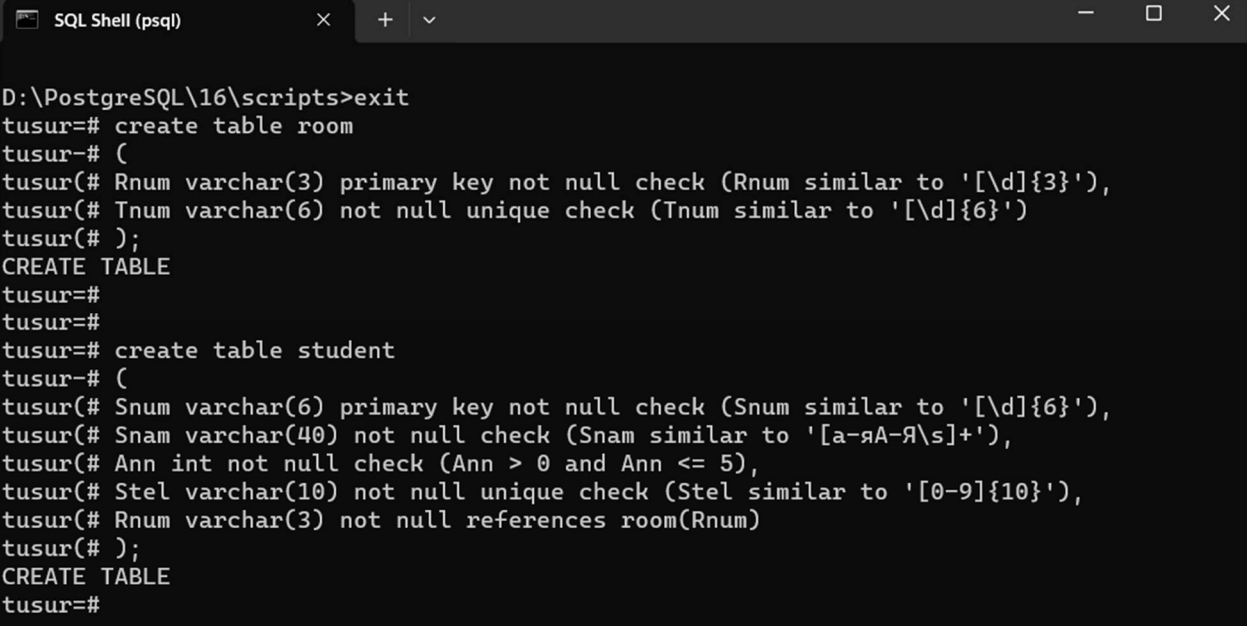
Примеры классов литералов:

- `[A-Z]` - символы верхнего регистра
- `[a-z]` - символы нижнего регистра
- `[A-Za-z]` - символы верхнего и нижнего регистра
- `[0-9]` – цифры (еще можно обозначать как `[\d]`)

Так же некоторые поля требуют свою уникальность, которую можно ограничить с помощью команды `UNIQUE`. Указывает на то, что все значения в поле должны быть уникальными (т.е. запрещает наличие дублирующихся значений в этом поле).

В таблице `STUDENT` присутствует ссылочность на первичный ключ поля из таблицы `ROOM`. Создается с помощью `REFERENCES имя_таблицы(имя_поля)`. Связывает поле с первичным ключом другой таблицы.

Таблицы, которые нужно создать – STUDENT, ROOM представлены на рисунке 3.1.



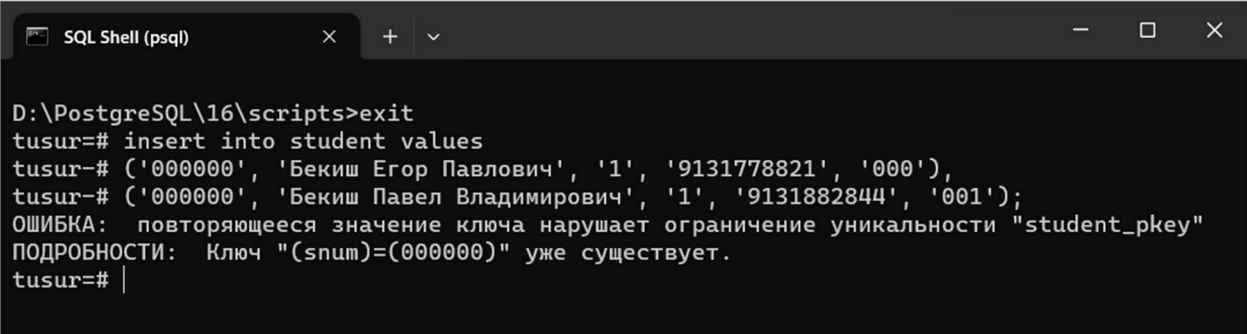
```
SQL Shell (psql)
D:\PostgreSQL\16\scripts>exit
tusur=# create table room
tusur-# (
tusur(# Rnum varchar(3) primary key not null check (Rnum similar to '[\d]{3}'),
tusur(# Tnum varchar(6) not null unique check (Tnum similar to '[\d]{6}'))
tusur(# );
CREATE TABLE
tusur=#
tusur=#
tusur=# create table student
tusur-# (
tusur(# Snum varchar(6) primary key not null check (Snum similar to '[\d]{6}'),
tusur(# Snam varchar(40) not null check (Snam similar to '[a-яA-Я\s]+'),
tusur(# Ann int not null check (Ann > 0 and Ann <= 5),
tusur(# Stel varchar(10) not null unique check (Stel similar to '[0-9]{10}'),
tusur(# Rnum varchar(3) not null references room(Rnum))
tusur(# );
CREATE TABLE
tusur=#
```

Рисунок 3.1 – создание таблиц STUDENT и ROOM

Теперь проверим корректность условий свойств имен для каждой таблицы.

Целостность свойств имен таблицы STUDENT:

- Snum - Номер студенческого билета. (рис. 3.2)
- Snam - Фамилия, имя, отчество студента. (рис. 3.3)
- Ann - Год обучения. (рис. 3.4)
- Stel - Номер индивидуального телефона студента. (рис. 3.5)



```
SQL Shell (psql)
D:\PostgreSQL\16\scripts>exit
tusur=# insert into student values
tusur-# ('000000', 'Бекиш Егор Павлович', '1', '9131778821', '000'),
tusur-# ('000000', 'Бекиш Павел Владимирович', '1', '9131882844', '001');
ОШИБКА: повторяющееся значение ключа нарушает ограничение уникальности "student_pkey"
ПОДРОБНОСТИ:  Ключ "(snum)=(000000)" уже существует.
tusur=# |
```

Рисунок 3.2 – проверка Snum на первичный ключ таблицы

```
SQL Shell (psql)
D:\PostgreSQL\16\scripts>exit
tusur=# insert into student values
tusur-# ('000000', 'Бекиш Егор Павлович', '1', '9131778821', '000')
tusur=# ;
ОШИБКА: новая строка в отношении "student" нарушает ограничение-проверку "student_snam_check"
ПОДРОБНОСТИ: Ошибочная строка содержит (000000, Бекиш Егор Павлович, 1, 9131778821, 000).
tusur=# |
```

Рисунок 3.3 – проверка Snam на ограничение русских символов

```
SQL Shell (psql)
D:\PostgreSQL\16\scripts>exit
tusur=# insert into student values
tusur-# ('000000', 'Бекиш Егор Павлович', '0', '9131778821', '000');
ОШИБКА: новая строка в отношении "student" нарушает ограничение-проверку "student_ann_check"
ПОДРОБНОСТИ: Ошибочная строка содержит (000000, Бекиш Егор Павлович, 0, 9131778821, 000).
tusur=# insert into student values
tusur-# ('000000', 'Бекиш Егор Павлович', '6', '9131778821', '000');
ОШИБКА: новая строка в отношении "student" нарушает ограничение-проверку "student_ann_check"
ПОДРОБНОСТИ: Ошибочная строка содержит (000000, Бекиш Егор Павлович, 6, 9131778821, 000).
tusur=# |
```

Рисунок 3.4 – проверка Ann на ограничение года от 1 до 5

```
tusur=# insert into student values
tusur-# ('000000', 'Бекиш Егор Павлович', '1', '9131778821', '000'),
tusur-# ('000001', 'Бекиш Павел Владимирович', '1', '9131778821', '001');
ОШИБКА: повторяющееся значение ключа нарушает ограничение уникальности "student_stel_key"
ПОДРОБНОСТИ: Ключ "(stel)=(9131778821)" уже существует.
tusur=# |
```

Рисунок 3.5 – проверка Stel на уникальность номера телефона

Целостность свойств имен таблицы ROOM:

- Rnum - Номер комнаты. (рис. 3.6)
- Tnum - Номер телефона, установленного в комнате. (рис. 3.7)

```
tusur=# insert into room values
tusur-# ('000', '000000'),
tusur-# ('000', '000001');
ОШИБКА: повторяющееся значение ключа нарушает ограничение уникальности "room_pkey"
ПОДРОБНОСТИ: Ключ "(rnum)=(000)" уже существует.
tusur=#
```

Рисунок 3.6 – проверка Rnum на первичный ключ

```
tusur=# insert into room values
tusur-# ('000', '000000'),
tusur-# ('001', '000000');
ОШИБКА: повторяющееся значение ключа нарушает ограничение уникальности "room_tnum_key"
ПОДРОБНОСТИ: Ключ "(tnum)=(000000)" уже существует.
tusur=# |
```

Рисунок 3.7 – проверка Tnum на уникальность телефона в комнате

Ссылочность свойств имен таблицы STUDENT с первичным ключом таблицы ROOM представлена на рисунке 3.8.

```
tusur=# insert into room values
tusur-# ('000', '000000'),
tusur-# ('001', '000001');
INSERT 0 2
tusur=#
tusur=#
tusur=# insert into student values
tusur-# ('000000', 'Бекиш Егор Павлович', '1', '9131778821', '000'),
tusur-# ('000001', 'Бекиш Павел Владимирович', '1', '9131882844', '002');
ОШИБКА: INSERT или UPDATE в таблице "student" нарушает ограничение внешнего ключа "student_rnum_fkey"
ПОДРОБНОСТИ: Ключ (rnum)=(002) отсутствует в таблице "room".
tusur=#
```

Рисунок 3.8 – проверка Rnum на ссылочность первичного ключа из ROOM

Успешное добавление данных в таблицу представлено на рисунке 3.9.

```
D:\PostgreSQL\16\scripts>exit
tusur=# insert into student values
tusur-# ('000000', 'Бекиш Егор Павлович', '1', '9131778821', '000'),
tusur-# ('000001', 'Бекиш Павел Владимирович', '1', '9131882844', '000');
INSERT 0 2
tusur=#
tusur=#
tusur=# select * from room join student on
tusur-# room.rnum = student.rnum;
 rnum | tnum | snum |          sname          | ann |   stel   | rnum
-----+-----+-----+-----+-----+-----+-----
  000 | 000000 | 000000 | Бекиш Егор Павлович    |   1 | 9131778821 | 000
  000 | 000000 | 000001 | Бекиш Павел Владимирович |   1 | 9131882844 | 000
(2 строки)

tusur=#
```

Рисунок 3.9 – успешное добавление данных

4 Вывод

В ходе выполнения лабораторной работы я научился пользоваться утилитой psql, создавать в ней таблицы, ограничивать поля классов требуемыми свойствами, используя в них регулярные выражения при необходимости.