Федеральное государственное автономное

Образовательное учреждение

Высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и информационных технологий

институт

Кафедра «Информатика»

кафедра

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №3**

Основы языка определения данных

тема

Преподаватель \_\_\_\_\_\_\_\_\_ Е.П. Моргунов

подпись, дата инициалы, фамилия

Студент КИ16-16Б, №031620303 \_\_\_\_\_\_\_\_\_ В.В. Кокташев

Номер группы, зачетной книжки подпись, дата инициалы, фамилия

Красноярск 2018

# Цель работы

# Ознакомиться с основами языка определения данных СУБД PostgreSQL;

# выполнить задания в конце пятого раздела учебного пособия.

# Ход работы

1. При использовании значений по умолчанию с ключевым словом DEFAULT возможны и ситуации, когда типичным будет не конкретное значение данных, а способ его получения. Например, если мы захотим фиксировать в каждой строке таблицы «Студенты» (students) имя пользователя базы данных, добавившего эту строку в таблицу, тогда необходимо в определение таблицы добавить еще один столбец. Этот столбец по умолчанию будет получать значение, возвращаемое функцией current\_user. Эта функция — current\_user — будет вызываться не при создании таблицы, а при вставке каждой строки. При этом в команде INSERT не требуется указывать значение для столбца who\_adds\_row, поскольку функция current\_user будет вызываться самой СУБД PostgreSQL. Давайте пойдем дальше и пожелаем фиксировать не только имя пользователя базы данных, добавившего строку в таблицу, но также и момент времени, когда это было сделано. Самостоятельно внесите модификацию в определение таблицы students для решения этой задачи, а затем выполните команду INSERT для проверки полученного решения. Если до выполнения этого упражнения вы еще не ознакомились с командой ALTER TABLE, то вместо модифицирования определения таблицы сначала удалите ее, а затем создайте заново.

Результат выполнения задания представлен на рисунке 1.

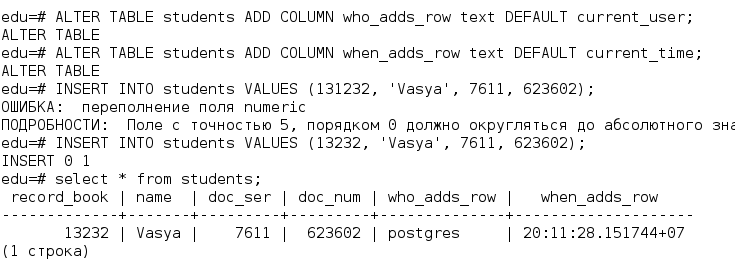


Рисунок 1 – Задание 1

1. В определении таблицы «Успеваемость» (progress) на атрибуты term и mark наложены как ограничения CHECK, так и ограничение NOT NULL. Возникает вопрос: не является ли ограничение NOT NULL избыточным? Ведь мы в ограничении CHECK явно указали допустимые значения. Проверьте гипотезу об избыточности ограничения NOT NULL в данном случае. Для этого модифицируйте таблицу, убрав ограничение NOT NULL, и попробуйте добавить в нее строку с отсутствующим значением атрибута term (или mark).

Результат выполнения задания представлен на рисунке 2.

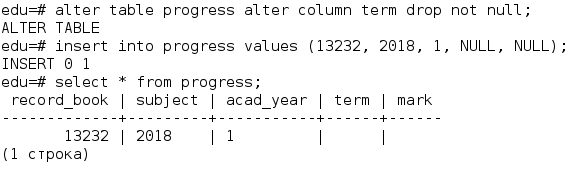


Рисунок 2 – Задание 3

1. В стандарте SQL сказано, что при наличии ограничения уникальности, включающего один или более столбцов, все же возможны повторяющиеся значения этих столбцов в разных строках, но лишь в том случае, если это значения — NULL. PostgreSQL придерживается такого же подхода. Модифицируйте определение табливщцы «Студенты» (students), добавив ограничение уникальности по двум столбцам: doc\_ser и doc\_num. А затем проверьте вышеприведенное утверждение, добавив в таблицу не только строки, содержащие кштыонкретные значения этих двух столбцов, но также и по две строки, имеющие следующие свойства:

– одинаковые значения столбца doc\_ser и NULL-значения столбца doc\_num;

– NULL-значения столбца doc\_num и столбца doc\_ser.

Результат выполнения задания представлен на рисунке 3.

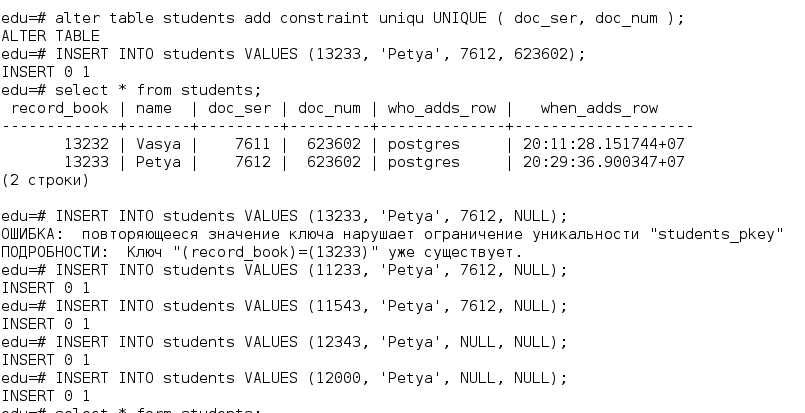


Рисунок 3 – Задание 5

1. Модифицируйте определение таблицы «Успеваемость» (progress), а если потребуется, то и определение таблицы «Студенты» (students), чтобы изучить все варианты реагирования СУБД на обновление строк в ссылочной таблице, в данном случае — students. Последовательно изменяйте определение внешнего ключа таблицы progress, испробовав варианты ON UPDATE CASCADE, ON UPDATE RESTRICT, ON UPDATE SET NULL и ON UPDATE SET DEFAULT. Для получения информативной картины введите несколько строк в обе таблицы, а затем выполняйте операцию UPDATE, подбирая значения ключевых атрибутов таким образом, чтобы вызвать ожидаемую реакцию СУБД. Учтите, что при использовании фразы ON UPDATE SET DEFAULT необходимо, чтобы, во-первых, с помощью ключевого слова DEFAULT было установлено значение по умолчанию для атрибута внешнего ключа в ссылающейся таблице, а во-вторых, это DEFAULT-значение все равно должно присутствовать в одной из строк ссылочной таблицы. Как вы считаете, с учетом сказанного, возможно ли использование ON UPDATE SET DEFAULT в нашем случае? Попробуйте обосновать или, наоборот, опровергнуть целесообразность использования каждой из этих политик — CASCADE, RESTRICT, SET NULL и SET DEFAULT — при выполнении операции UPDATE в реальной информационной системе, предназначенной для учета успеваемости студентов.

Результат выполнения задания представлен на рисунках 4-5.

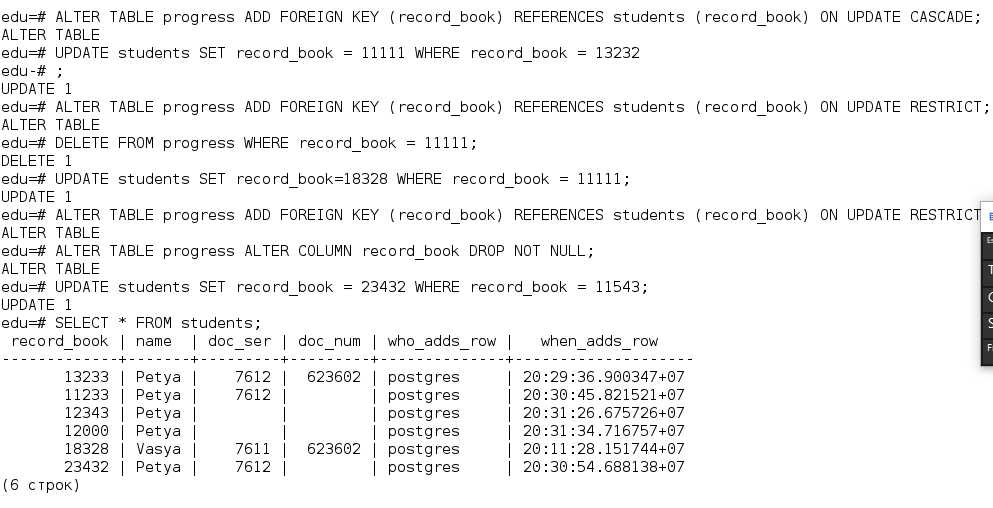


Рисунок 4 – Задание 7

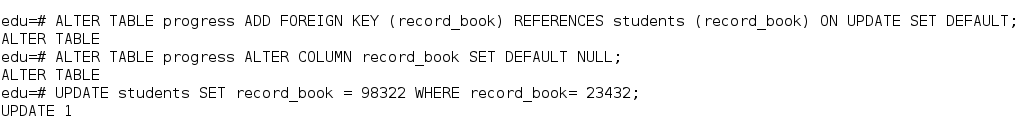


Рисунок 5 – Задание 7

1. В таблице «Студенты» (students) есть текстовый атрибут name, на который наложено ограничение NOT NULL. Как вы думаете, что будет, если при вводе новой строки в эту таблицу дать атрибуту name в качестве значения пустую строку? Наверное, проектируя эту таблицу, мы хотели бы все же, чтобы пустые строки в качестве значения атрибута name не проходили в базу данных? Какое решение вы можете предложить? Видимо, нужно добавить ограничение CHECK для столбца name. Добавив ограничение, попробуйте теперь вставить в таблицу students сроку (row), в которой значение атрибута name было бы пустой строкой (string). Давайте продолжим эксперименты и предложим в качестве значения атрибута name строку, содержащую сначала один пробел, а потом — два пробела. Оказывается, эти невидимые значения имеют ненулевую длину. Что делать, чтобы не допустить таких значений-невидимок? Один из способов: возложить проверку таких ситуаций на прикладную программу. А что можно сделать на уровне определения таблицы students? Какое ограничение нужно предложить? В разделе 9.4 «Строковые функции и операторы» есть функция trim(). Попробуйте воспользоваться ею. Посмотрите и таблицу «Успеваемость» (progress) на предмет подобных слабых мест.

Результат выполнения задания представлен на рисунке 6.

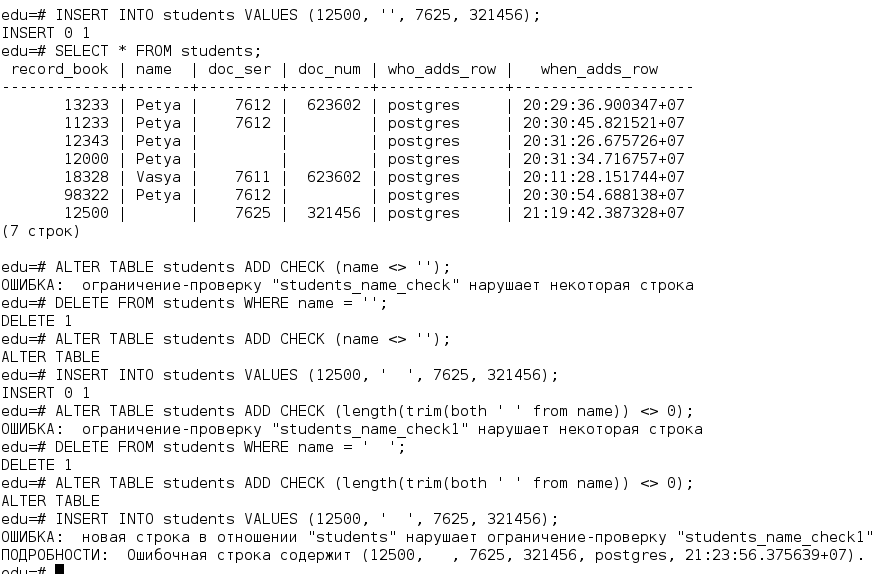


Рисунок 6 – Задание 9

1. В таблице «Рейсы» (flights) есть ограничение, которое регулирует соотношения значений фактического времени вылета и фактического времени прилета. Как вы думаете, не является ли выражение actual\_arrival IS NOT NULL во второй части условного оператора OR избыточным? Проверьте ваши предположения на практике. Попробуйте добавить в таблицу flights две-три строки, подбирая такие значения атрибутов actual\_departure и actual\_arrival, чтобы проверить все возможные исходы этих проверок. Конечно, вместо добавления новых строк можно модифицировать одну и ту же строку с помощью команды UPDATE.

Результат выполнения задания представлен на рисунке 7.

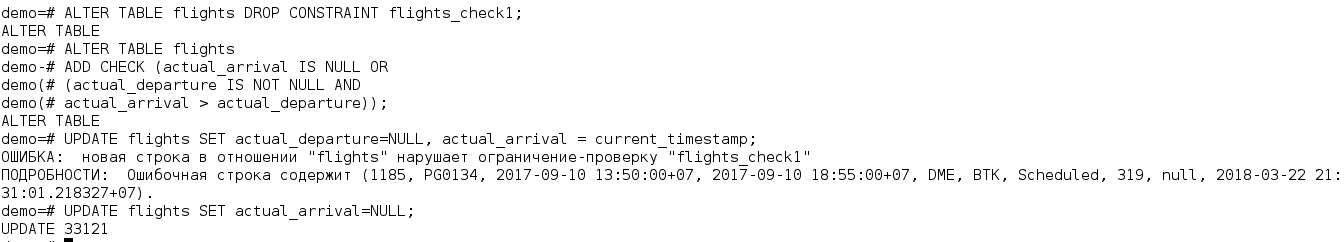


Рисунок 7 – Задание 11

1. Представление «Рейсы» (flights\_v) и материализованное представление «Маршруты» (routes) построены на основе таблиц «Рейсы» (flights) и «Аэропорты» (airports). Логично предположить, что при каскадном удалении, например, таблицы «Аэропорты», представление «Рейсы» будет также удалено, поскольку при удалении базовой таблицы этому представлению просто неоткуда будет брать данные. А что вы можете предположить насчет материализованного представления «Маршруты»: будет ли оно также удалено или нет? Ведь оно уже содержит данные, в отличие от обычного представления. Так ли, условно говоря, сильна его связь с таблицами, на основе которых оно сконструировано? Проведите необходимые эксперименты.

Результат выполнения задания представлен на рисунке 8.

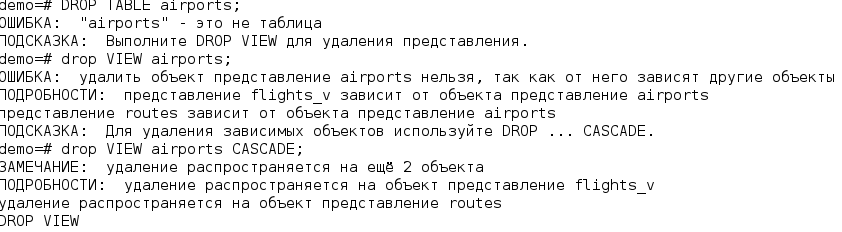


Рисунок 8 – Задание 13

1. Определение таблицы можно изменить с помощью команды ALTER TABLE. Аналогичные команды существуют и для изменения представлений и материализованных представлений: ALTER VIEW и ALTER MATERIALIZED VIEW. Самостоятельно ознакомьтесь с их возможностями с помощью документации.

Результат выполнения задания представлен на рисунке 9.

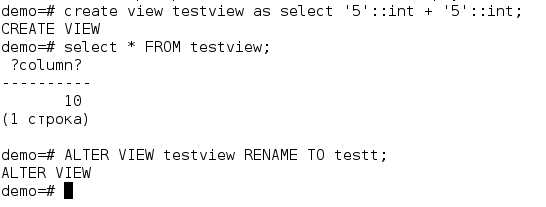


Рисунок 8 – Задание 15

1. Представления могут быть, условно говоря, вертикальными и горизонтальными. При создании вертикального представления в список его столбцов включается лишь часть столбцов базовой таблицы (таблиц). В горизонтальное представление включаются не все строки базовой таблицы (таблиц), а производится их отбор с помощью фраз WHERE или HAVING. Конечно, вполне возможен и смешанный вариант, когда ограничивается как список столбцов, так и множество строк при создании представления. Подумайте, какие представления было бы целесообразно создать для нашей базы данных «Авиаперевозки». Необходимо учесть наличие различных групп пользователей, например: пилоты, диспетчеры, пассажиры, кассиры. Создайте представления и проверьте их в работе.

Результаты выполнения задания представлен на рисунках 10-11.

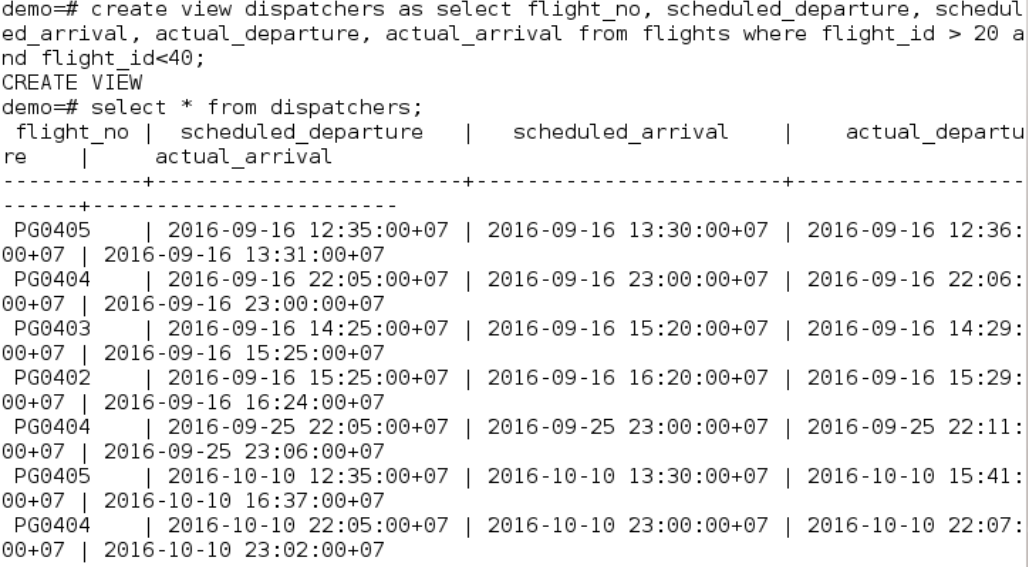


Рисунок 10 – Задание 17

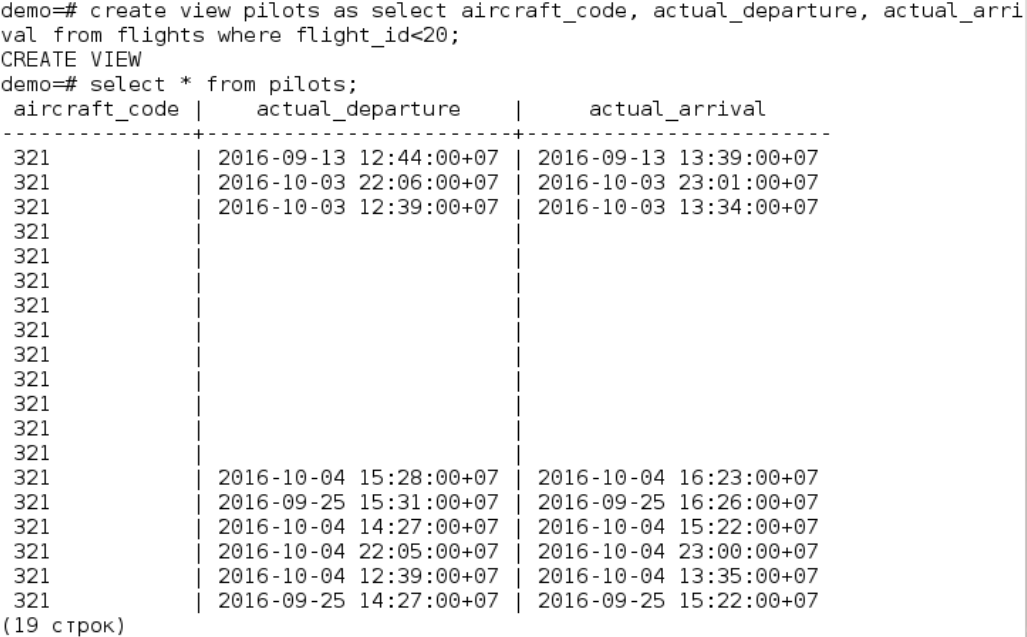


Рисунок 11 – Задание 17

# Вывод

В ходе выполнения практической работы мы ознакомились с основами языка определения данных СУБД PostgreSQL и полученная теория закреплена на практике.