Федеральное государственное автономное

образовательное учреждение

Высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

|  |
| --- |
| Институт космических и информационных технологий |
| институт |
| Информатика |
| кафедра |

**ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ**

|  |
| --- |
| 3. Основы языка определения данных |
| тема |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Преподаватель | |  |  |  | Е. П. Моргунов |
|  | |  | подпись, дата |  | инициалы, фамилия |
| Студент | КИ21-17/2Б, 032156940 |  |  |  | Н. А. Самарин |
|  | номер группы, зачётной книжки |  | подпись, дата |  | инициалы, фамилия |

Красноярск 2023

**СОДЕРЖАНИЕ**

1 Задание............................................................................................................... 3

2 Ход работы........................................................................................................ 3

**1 Задание**

- Изучить материал главы 5 учебного пособия: Моргунов, Е. П.  
PostgreSQL. Основы языка SQL [Текст] : учеб. пособие / Е. П. Моргунов ; под  
ред. Е. В. Рогова, П. В. Лузанова. – СПб. : БХВ-Петербург, 2018. – 336 с. : ил. –  
ISBN 978-5-9775-4022-3. Можно дополнительно использовать презентации и  
видеоуроки: Моргунов, Е. П. Язык SQL [Электронный ресурс] : практический  
курс. – М. : Постгрес Профессиональный, 2019;

- Выполнить задания, приведенные в разделе "Контрольные вопросы и  
задания";

- Подготовить отчет. Включить в него тексты заданий из пособия. Для  
каждого задания сделать снимки экрана с введенными командами языка SQL и  
результатами их выполнения в среде утилиты psql.

**2 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» 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. CREATE TABLE  
students ( record\_book numeric( 5 ) NOT NULL, name text NOT NULL, doc\_ser  
numeric( 4 ), doc\_num numeric( 6 ), who\_adds\_row text DEFAULT current\_user, --  
добавленный столбец PRIMARY KEY ( record\_book ) ); 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: INSERT INTO students ( record\_book, name, doc\_ser,  
doc\_num ) VALUES ( 12300, 'Иванов Иван Иванович', 0402, 543281 ); 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.

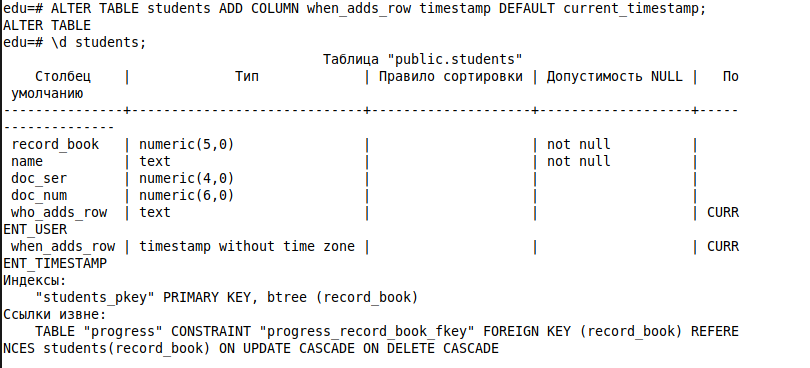


Рисунок 1 – Выполнение задания 1

3. \* 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 в  
ограничении CHECK явно указаны 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).

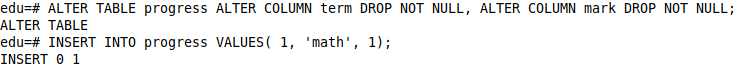


Рисунок 2 – Выполнение задания 3

5. 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 таблицы «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 конкретные 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.

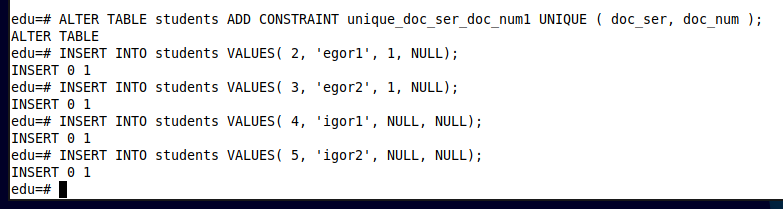


Рисунок 3 – Выполнение задания 5

7. 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.

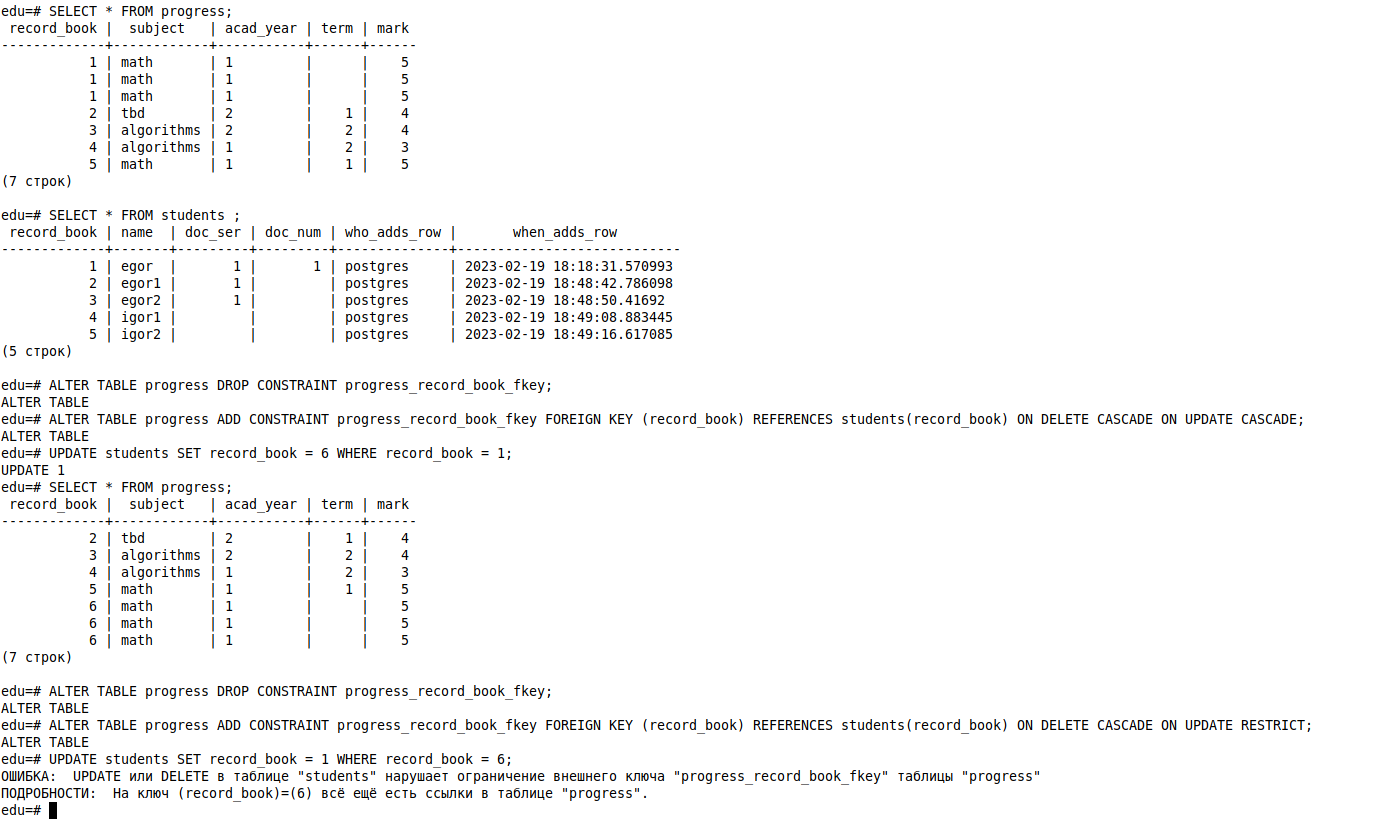


Рисунок 4 – Выполнение задания 7 часть 1

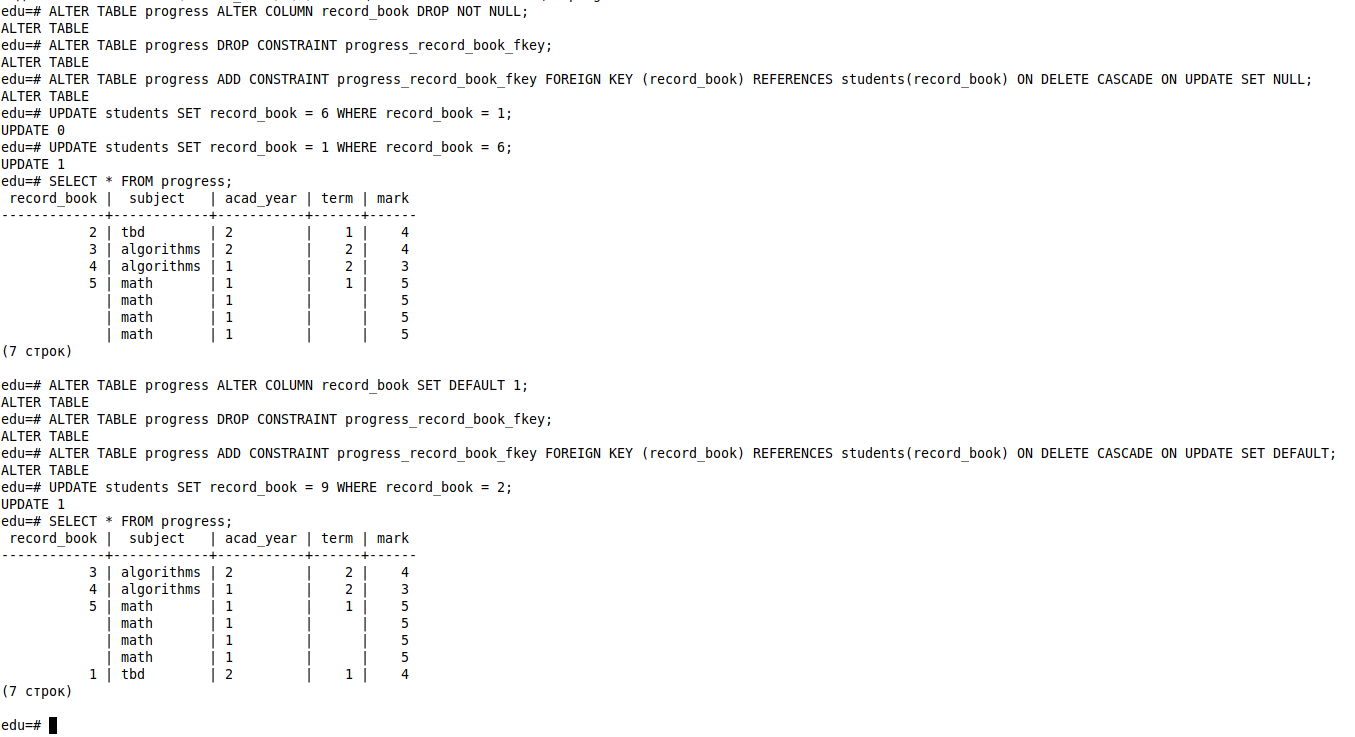


Рисунок 5 – Выполнение задания 7 часть 2

9. Давайте продолжим эксперименты и предложим в качестве значения  
атрибута name строку, содержащую сначала один пробел, а потом — два  
пробела. INSERT INTO students VALUES ( 12346, ' ', 0406, 112233 ); INSERT  
INTO students VALUES ( 12347, ' ', 0407, 112234 ); Для того чтобы «увидеть»  
эти пробелы в выборке, сделаем так: SELECT \*, length( name ) FROM students; 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 документации «Строковые функции и операторы» есть функция  
trim. Попробуйте воспользоваться ею. Если вы еще не изучили команду ALTER  
TABLE, то удалите таблицу students и создайте ее заново с учетом нового  
ограничения, а если уже познакомились с ней, то сделайте так: ALTER TABLE  
students ADD CHECK (...);

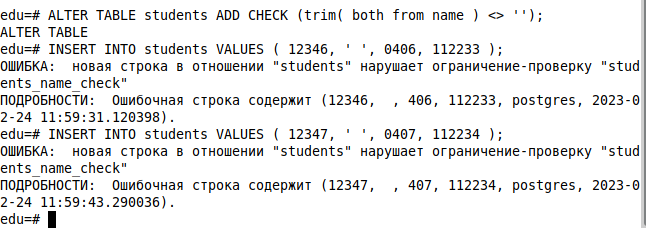


Рисунок 6 – Выполнение задания 9

11. 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? CREATE TABLE flights  
( ... CHECK ( actual\_arrival IS NULL OR ( actual\_departure IS NOT NULL AND  
actual\_arrival IS NOT NULL AND actual\_arrival > actual\_departure ) ), ...  
Проверьте ваши предположения на практике.

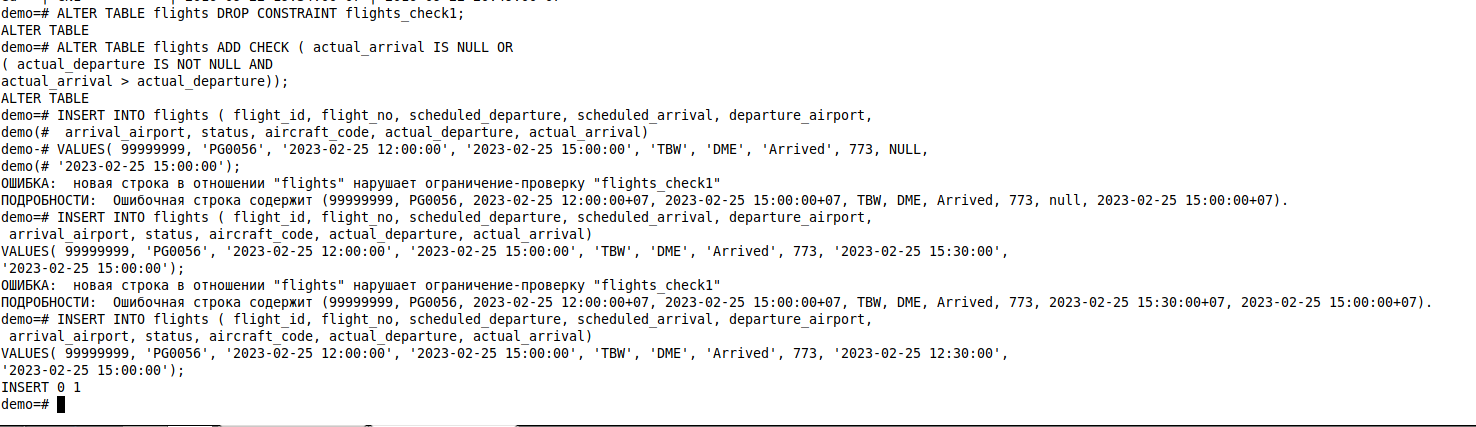


Рисунок 7 – Выполнение задания 11

13. И 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, начав с команды  
DROP TABLE airports;

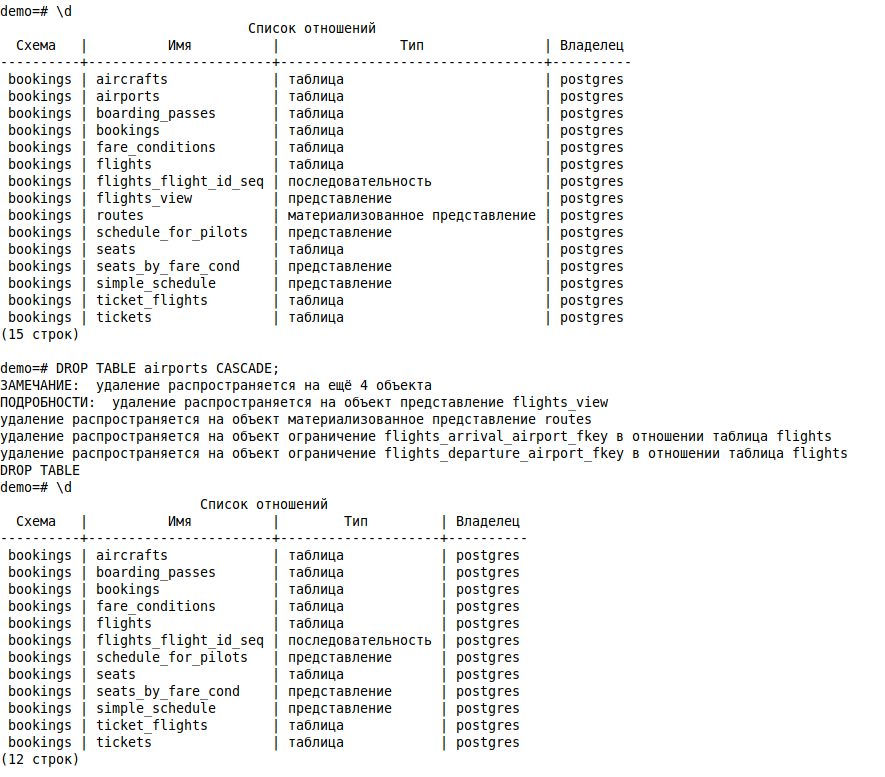


Рисунок 8 – Выполнение задания 13

15. 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.

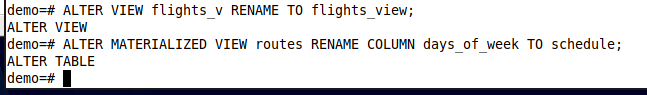


Рисунок 9 – Выполнение задания 15

17. 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.

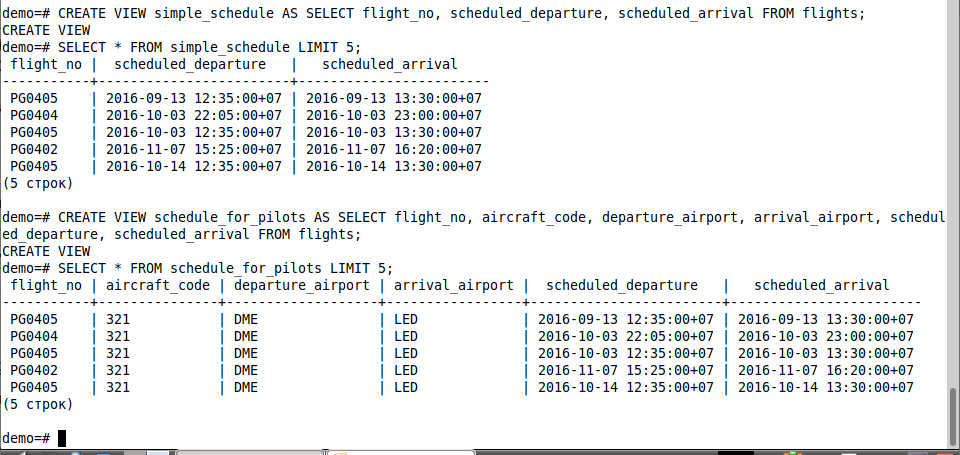


Рисунок 10 – Выполнение задания 17