|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Министерство науки и высшего образования РФ | | | | | |
| Федеральное государственное автономное | | | | | |
| образовательное учреждение высшего образования | | | | | |
| **«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»** | | | | | |
|  | | | | | |
| Институт космических и информационных технологий | | | | | |
| институт | | | | | |
| Программная инженерия | | | | | |
| кафедра | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
| **ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ №3** | | | | | |
| Основы языка определения данных | | | | | |
| тема | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
| Преподаватель | |  |  |  | А. Д. Вожжов |
|  | |  | подпись, дата |  | инициалы, фамилия |
| Студент | КИ23-17/1б, 032320521 |  |  |  | А. С. Гуртякин |
|  | номер группы, зачётной книжки |  | подпись, дата |  | инициалы, фамилия |
|  | | | | | |
|  | | | | | |
| Красноярск 2025 | | | | | |

# ВВЕДЕНИЕ

## Цель работы

Изучить теоретический материал по теме «Основы языка определения данных». Выполнить задания.

## Задачи

В рамках данной практической работы необходимо выполнить следующие задачи:

1. изучить теоретический материал по предложенной теме;
2. выполнить задание;
3. предоставить отчёт преподавателю.

## Задание

Задание данной практической работы состоит из 4 частей:

1. Выполнить задания из главы 5 из книги на е-курсах.

# ХОД РАБОТЫ

## Задание 1

При использовании значений по умолчанию с ключевым словом DEFAULT воз-

можны и ситуации, когда типичным будет не конкретное значение данных,

а способ его получения.

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 табли-

цы students для решения этой задачи, а затем выполните команду INSERT для

проверки полученного решения.

На рисунке 1 показан результат выполнения задания.

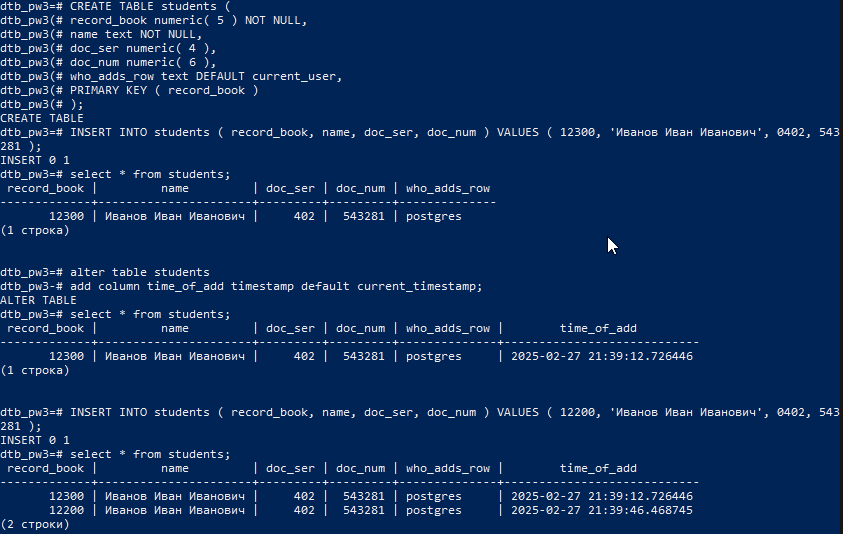


Рисунок 1 — Выполнение задания 1

## Задание 2

Посмотрите, какие ограничения уже наложены на атрибуты таблицы «Успеваемость» (progress). Воспользуйтесь командой \d утилиты psql. А теперь предложите для этой таблицы ограничение уровня таблицы.

В качестве примера рассмотрим такой вариант. Добавьте в таблицу progress еще один атрибут — «Форма проверки знаний» (test\_form), который может принимать только два значения: «экзамен» или «зачет». Тогда набор допустимых значений атрибута «Оценка» (mark) будет зависеть от того, экзамен или зачет предусмотрены по данной дисциплине. Если предусмотрен экзамен, тогда допускаются значения 3, 4, 5, если зачет — тогда 0 (не зачтено) или 1 (зачтено)

На рисунках 2, 3, 4 показан результат выполнения задания.

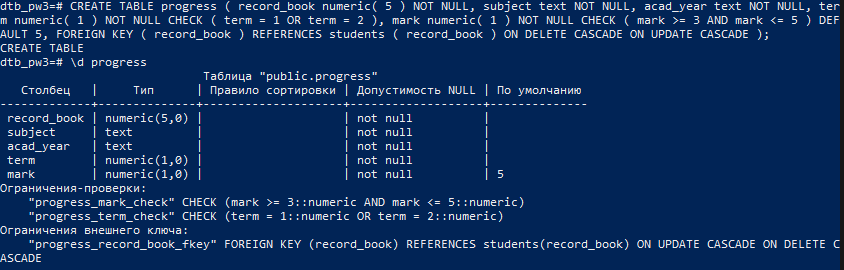


Рисунок 2 — Создание таблицы и вывод сведений



Рисунок 3 — Модификация столбца

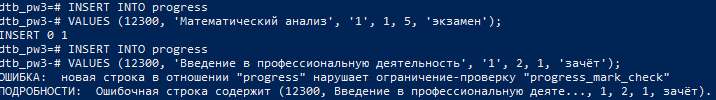


Рисунок 4 — Вставка значений

Можно добавить ограничения на не пустоту subject.

## Задание 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)

На рисунках 5, 6, 7 показан результат выполнения задания.



Рисунок 5 — Вставка значений



Рисунок 6 — Модификация столбцов

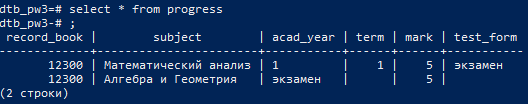


Рисунок 7 — Вывод результата

Гипотеза подтвердилась. Проверка нужна.

## Задание 4

В определении таблицы «Успеваемость» (progress) для атрибута mark не только задано ограничение CHECK, но и установлено значение по умолчанию с помощью ключевого слова DEFAULT:

Как вы думаете, что будет, если в ограничении DEFAULT мы «случайно» допустим ошибку, написав DEFAULT 6? Если в команде INSERT не указать значение для атрибута mark, то на каком этапе эта ошибка будет выявлена: уже на этапе создания таблицы или только при вставке строки в нее?

На рисунках 8, 9 показан результат выполнения задания.



Рисунок 8 — Модификация столбца



Рисунок 9 — Вставка значений

Проверка значений происходит на этапе вставки.

## Задание 5

В стандарте SQL сказано, что при наличии ограничения уникальности, вклю-чающего 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.

Подобные вещи возможны, так как NULL-значения не считаются совпадающи-

ми.

На рисунках 10, 11, 12, 13, 14 показан результат выполнения задания.

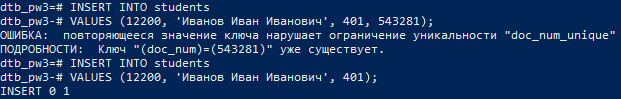


Рисунок 10 — Вставка значений, проверка, часть 1



Рисунок 11 — Вставка значений, проверка, часть 2



Рисунок 12 — Вставка значений, проверка, часть 3



Рисунок 13 — Модификация столбцов



Рисунок 14 — Вставка значений, проверка

## Задание 6

Модифицируйте определения таблиц «Студенты» (students) и «Успеваемость» (progress). В таблице students в качестве первичного ключа назначьте комбинацию атрибутов doc\_ser и doc\_num, а в таблице progress соответствующим образом измените определение внешнего ключа.

На рисунках 15, 16, 17, 18, показан результат выполнения задания.

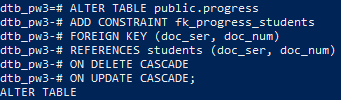


Рисунок 15 — Модификация таблицы, добавление внешних ключей



Рисунок 16 — Модификация таблицы, смена ключа, часть 1



Рисунок 17 — Модификация таблицы, смена ключа, часть 2



Рисунок 18 — Вставка значений, проверка

## Задание 7

Модифицируйте определение таблицы «Успеваемость» (progress), а если потребуется, то и определение таблицы «Студенты» (students), чтобы изучитьвсе 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.

На рисунках 19, 20, 21, 22, 23, 24, 25 показан результат выполнения задания.

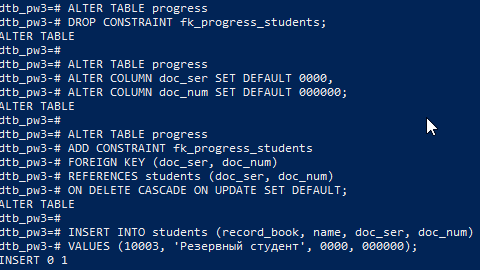


Рисунок 19 — Тест ON UPDATE SET DEFAULT



Рисунок 20 — Вставка значений, проверка

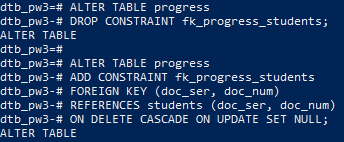


Рисунок 21 — Тест ON UPDATE SET NULL

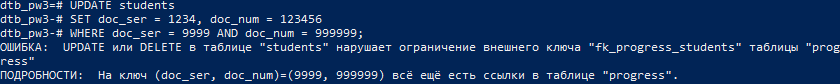


Рисунок 22 — Вставка значений, проверка

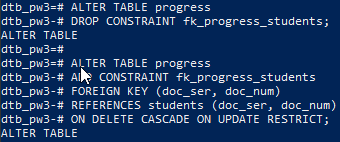


Рисунок 23 — Тест ON UPDATE RESCTRICT



Рисунок 24 — Вставка значений



Рисунок 25 — Обновление значений

## Задание 8

В таблице «Успеваемость» (progress) есть атрибут «Учебная дисциплина»(subject). Это текстовый атрибут. Одинаковые наименования учебных дисциплин записываются в таблицу progress многократно. Создайте еще одну таблицу — «Учебные дисциплины» (subjects), в которой будет два атрибута: «Идентификатор учебной дисциплины» (subject\_id) и «Учебная дисциплина» (subject). Тип данных первого из них будет integer, а второго — text. В качестве первичного ключа будет служить subject\_id, а второй атрибут будет уникальным. Введите в новую таблицу две-три строки для различных учебных дисциплин.

Модифицируйте таблицу progress, заменив атрибут subject на subject\_id. Тип данных нового атрибута будет integer. Поскольку тип данных изменится, то для замены первоначальных значений, хранящихся в этом столбце, на новые придется использовать конструкцию USING (о ней говорится в тексте главы). Добавьте в определение таблицы progress еще один внешний ключ, который будет ссылаться на таблицу subjects. В составе этого внешнего ключа будет только один атрибут — subject\_id. Мы видим, что таблица может иметь больше одного внешнего ключа. Таким образом, структура связей в реальной базе данных может оказаться весьма сложной.

Теперь введите несколько строк и в таблицу progress, учитывая ее связь с новой таблицей subjects.

На рисунках 26, 27, 28, 29, 30 показан результат выполнения задания.

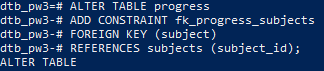


Рисунок 26 — Модификация таблицы, добавление внешнего ключа



Рисунок 27 — Вставка данных

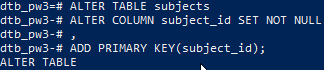


Рисунок 28 — Модификация таблицы



Рисунок 29 — Создание таблицы



Рисунок 30 — Вставка значений

## Задание 9

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

На рисунках 31, 32 показан результат выполнения задания.



Рисунок 31 — Добавление ограничения



Рисунок 32 — Вставка значений

Есть такие же недочёты, столбец subject не имеет проверку на пустоту.

## Задание 10

В таблице «Студенты» (students) атрибут «Серия документа, удостоверяющего личность» (doc\_ser) имеет числовой тип, однако в сериях таких документов могут встречаться лидирующие нули, которые в числовых столбцах не сохраняются. Например, при записи значения серии «0402» первый ноль не сохранится. Модифицируйте таблицу students, заменив числовой тип данных на символьный, например, character. Как вы думаете, эта операция пройдет без затруднений или они все же возможны? Проверьте ваши предположения, выполнив модификацию.

На рисунках 33, 34, 35 показан результат выполнения задания.



Рисунок 33 — Модификация таблицы

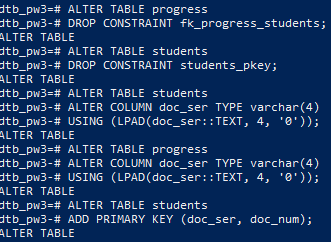


Рисунок 34 — Прогресс выполнения задания

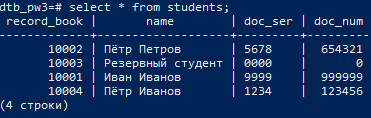


Рисунок 35 — Результат

Операция прошла не без затруднений. Нельзя так просто заменить данные.

## Задание 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.

Проверьте ваши предположения на практике.

На рисунках 36, 37 показан результат выполнения задания.

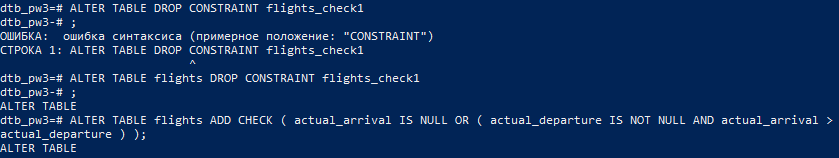


Рисунок 36 — Модификация таблицы

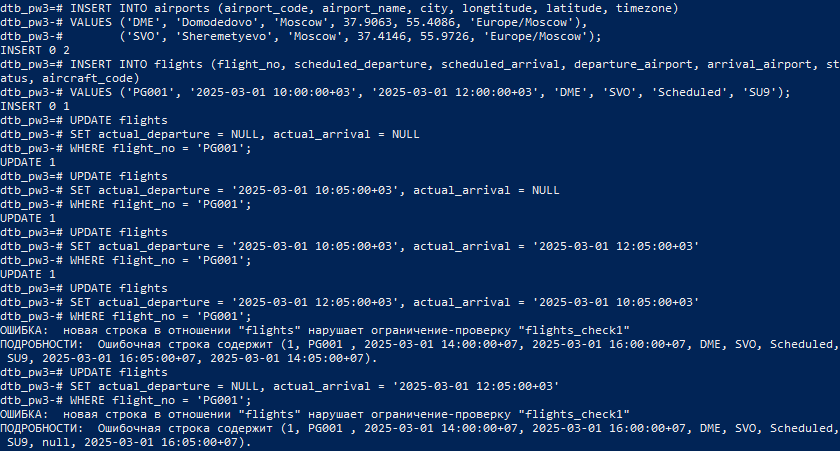


Рисунок 37 — Прогресс выполнения задания

## Задание 12

Поскольку в командах создания таблиц базы данных «Авиаперевозки» мы не указывали имена ограничений для первичных и внешних ключей, то их имена были сформированы автоматически самой СУБД. Как вы думаете, получили ли эти ограничения новые имена после переименования таблицы? Проверьте ваши предположения, выполнив такую операцию с одной из таблиц базы данных «Авиаперевозки», имеющих внешние ключи

На рисунках 38, 39, 40 показан результат выполнения задания.

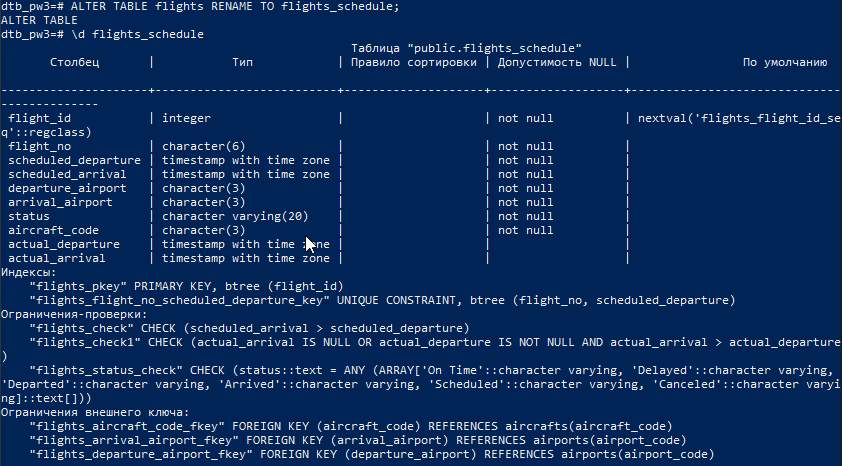


Рисунок 38 — Переименование таблицы

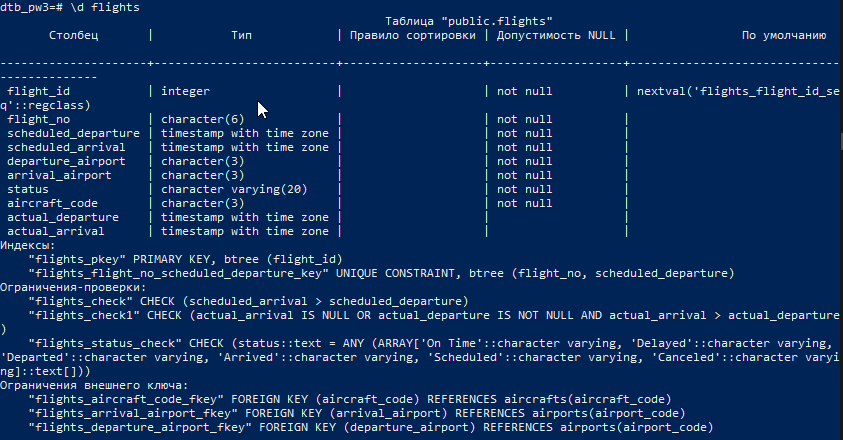


Рисунок 39 — Вид таблицы

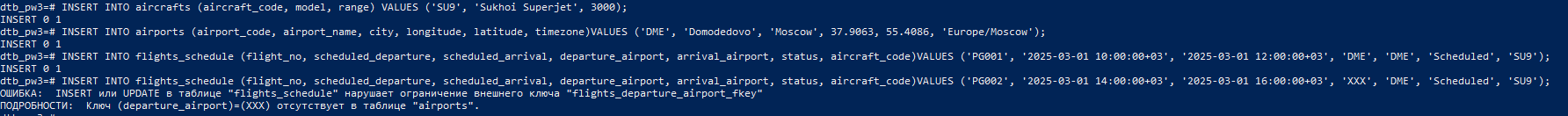


Рисунок 40 — Вставка значений

Ответ на вопрос в задании: да, ограничения перенеслись.

## Задание 13

И представление «Рейсы» (flights\_v), и материализованное представление

«Маршруты» (routes) построены на основе таблиц «Рейсы» (flights) и «Аэро-

порты» (airports). Логично предположить, что при каскадном удалении, на-

пример, таблицы «Аэропорты», представление «Рейсы» будет также удалено,

поскольку при удалении базовой таблицы этому представлению просто неот-

куда будет брать данные.

А что вы можете предположить насчет материализованного представления

«Маршруты»: будет ли оно также удалено или нет? Ведь оно уже содержит дан-

ные, в отличие от обычного представления. Так ли, условно говоря, сильна его

связь с таблицами, на основе которых оно сконструировано?

На рисунке 41 показан результат выполнения задания.

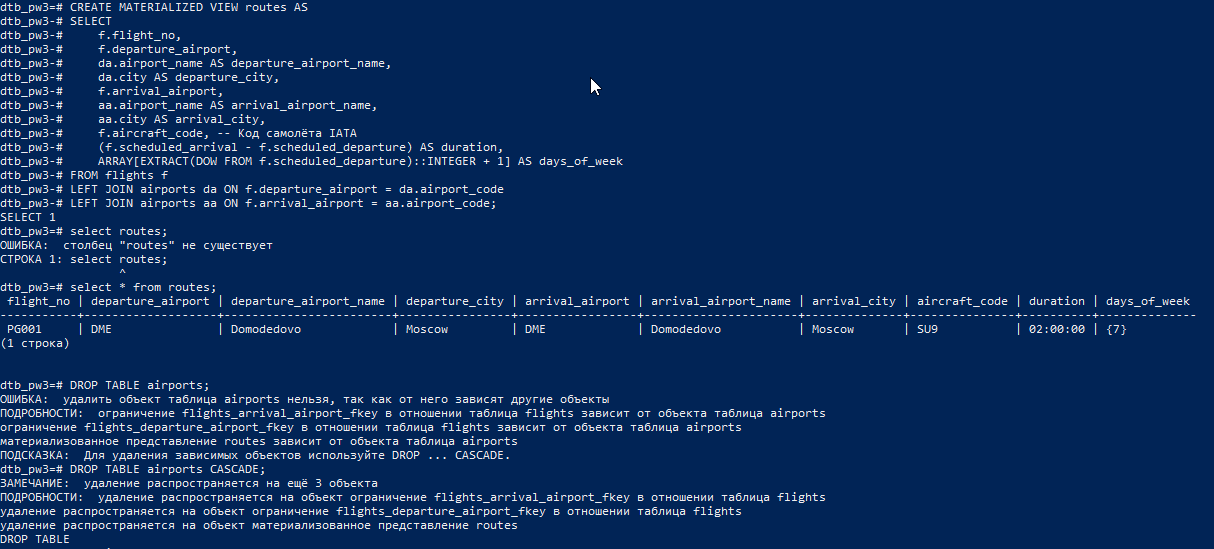


Рисунок 41 — Создание таблицы

## Задание 14

Представления (views) могут быть обновляемыми. Это значит, что можно с по-

мощью команд INSERT, UPDATE и DELETE, применяемых к представлению, вне-

сти изменения в таблицу, лежащую в основе этого представления.

Самостоятельно ознакомьтесь с этим вопросом с помощью документации (см.

описание команды CREATE VIEW) и, создав простое представление над одной

из таблиц базы данных «Авиаперевозки», выполните несколько команд с целью

внесения изменений в эту таблицу.

На рисунках 42, 43, 44 показан результат выполнения задания.

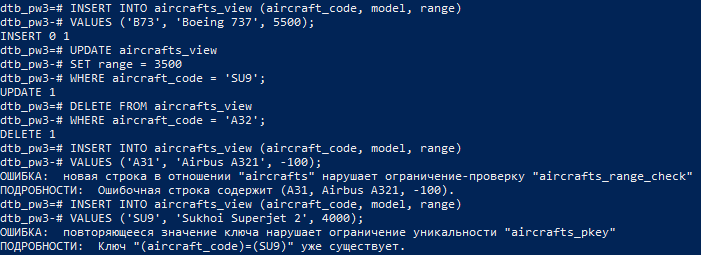


Рисунок 42 — Проверка обычного представления

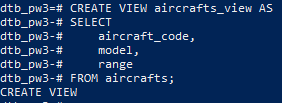


Рисунок 43 — Создание представления

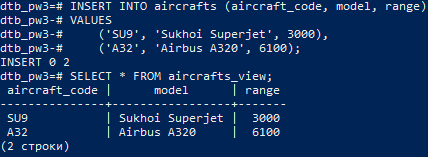


Рисунок 44 — Вставка значений в таблицу

## Задание 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.

На рисунках 45, 46, 47, 48 показан результат выполнения задания.

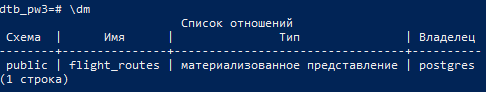


Рисунок 45 — Созданное материализованное представление



Рисунок 46 — Модификация представления

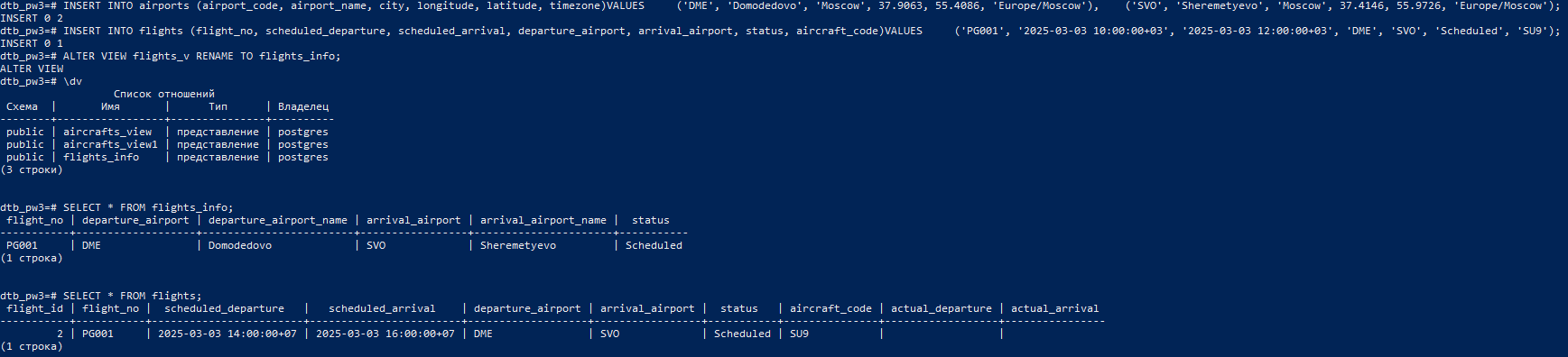


Рисунок 47 — Вставка значений, проверка

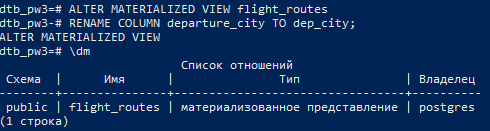


Рисунок 48 — Модификация представления

## Задание 16

Как вы думаете, при изменении данных в таблицах, на основе которых сконструировано материализованное представление, содержимое этого представления тоже синхронно изменяется или нет?

На рисунке 49 показан результат выполнения задания.



Рисунок 49 — Вставка значений

## Задание 17

Представления могут быть, условно говоря, вертикальными и горизонтальными. При создании вертикального представления в список его столбцов включается лишь часть столбцов базовой таблицы (таблиц)

На рисунке 50 показан результат выполнения задания.

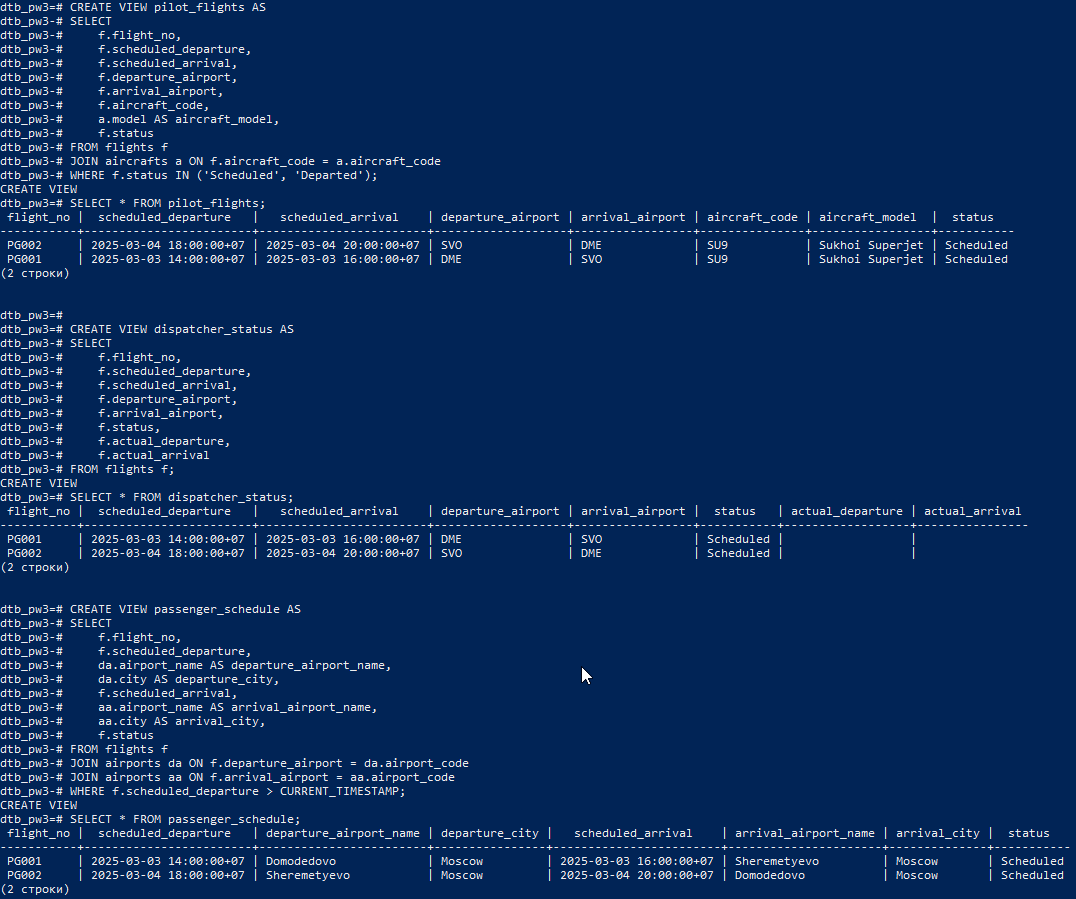


Рисунок 50 — Тест представлений

## Задание 18

Предположим, что нам понадобилось иметь в базе данных сведения о технических характеристиках самолетов, эксплуатируемых в авиакомпании. Пусть это будут такие сведения, как число членов экипажа (пилоты), тип двигателей и их количество.

На рисунке 51 показан результат выполнения задания.

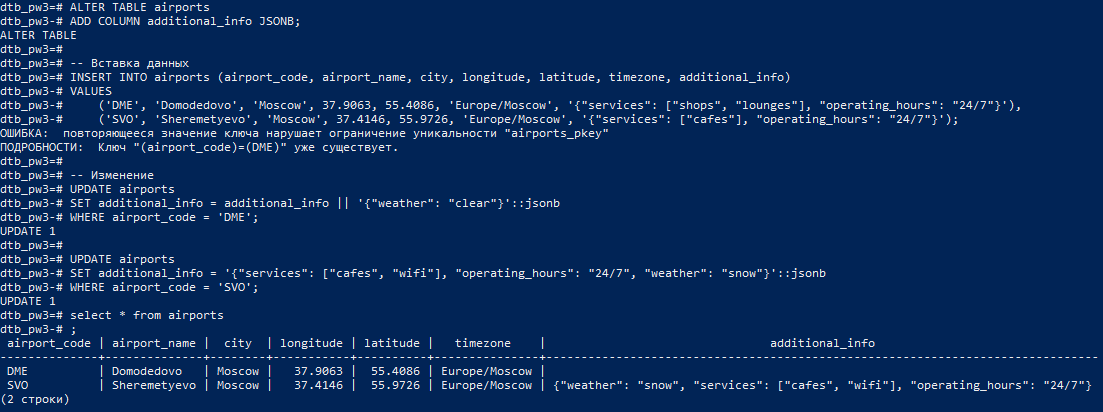


Рисунок 51 — Модификация таблиц, обновление значений

На рисунках 52, 53, 54, 55 показан результат выполнения задания.



Рисунок 52 — Модификация столбца

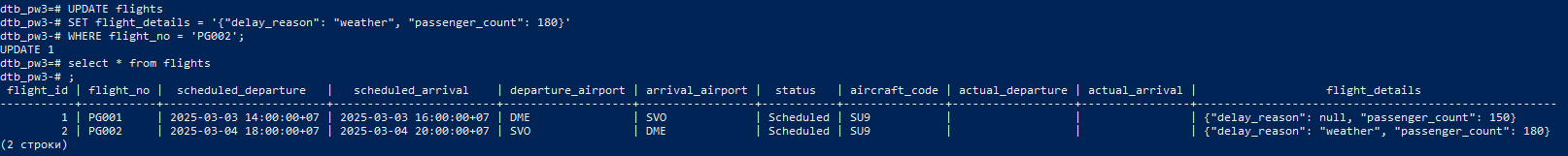


Рисунок 53 — Обновление таблицы



Рисунок 54 — Добавление столбца

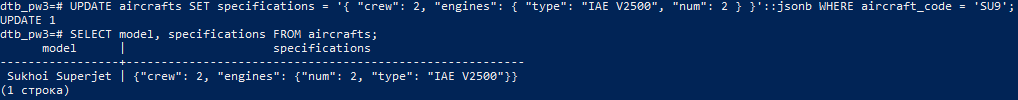


Рисунок 55 — Обновление значения

# ЗАКЛЮЧЕНИЕ

По результатам работы был изучен теоретический материал по теме «Основы языка определения данных». Все поставленные цели и задачи были выполнены.