

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

Основы языка определения данных

тема

Преподаватель

подпись, дата

А. Д. Вожжов

инициалы, фамилия

Студент КИ23-17/16, 032320521

номер группы, зачётной книжки

подпись, дата

А. С. Лысаковский

инициалы, фамилия

Красноярск 2025

1 ВВЕДЕНИЕ

1.1 Цель работы

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

1.2 Задачи

В рамках данной практической работы необходимо выполнить следующие задачи:

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

1.3 Задание

Задание данной практической работы состоит из 4 частей:

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

2 ХОД РАБОТЫ

2.1 Задание 1

На рисунке 1 показан результат выполнения задания.

```
dtb_pw3=# CREATE TABLE students (  
dtb_pw3=# record_book numeric( 5 ) NOT NULL,  
dtb_pw3=# name text NOT NULL,  
dtb_pw3=# doc_ser numeric( 4 ),  
dtb_pw3=# doc_num numeric( 6 ),  
dtb_pw3=# who_adds_row text DEFAULT current_user,  
dtb_pw3=# PRIMARY KEY ( record_book )  
dtb_pw3=# );  
CREATE TABLE  
dtb_pw3=# INSERT INTO students ( record_book, name, doc_ser, doc_num ) VALUES ( 12300, 'Иванов Иван Иванович', 0402, 543  
281 );  
INSERT 0 1  
dtb_pw3=# select * from students;  
 record_book |      name      | doc_ser | doc_num | who_adds_row  
-----+-----+-----+-----+-----  
    12300 | Иванов Иван Иванович |    402 | 543281 | postgres  
(1 строка)  
  
dtb_pw3=# alter table students  
dtb_pw3=# add column time_of_add timestamp default current_timestamp;  
ALTER TABLE  
dtb_pw3=# select * from students;  
 record_book |      name      | doc_ser | doc_num | who_adds_row |      time_of_add  
-----+-----+-----+-----+-----+-----  
    12300 | Иванов Иван Иванович |    402 | 543281 | postgres | 2025-02-27 21:39:12.726446  
(1 строка)  
  
dtb_pw3=# INSERT INTO students ( record_book, name, doc_ser, doc_num ) VALUES ( 12200, 'Иванов Иван Иванович', 0402, 543  
281 );  
INSERT 0 1  
dtb_pw3=# select * from students;  
 record_book |      name      | doc_ser | doc_num | who_adds_row |      time_of_add  
-----+-----+-----+-----+-----+-----  
    12300 | Иванов Иван Иванович |    402 | 543281 | postgres | 2025-02-27 21:39:12.726446  
    12200 | Иванов Иван Иванович |    402 | 543281 | postgres | 2025-02-27 21:39:46.468745  
(2 строки)
```

Рисунок 1 — Выполнение задания 1

2.2 Задание 2

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

```
dtb_pw3=# CREATE TABLE progress ( record_book numeric( 5 ) NOT NULL, subject text NOT NULL, acad_year text NOT NULL, ter  
m numeric( 1 ) NOT NULL CHECK ( term = 1 OR term = 2 ), mark numeric( 1 ) NOT NULL CHECK ( mark >= 3 AND mark <= 5 ) DEF  
AULT 5, FOREIGN KEY ( record_book ) REFERENCES students ( record_book ) ON DELETE CASCADE ON UPDATE CASCADE );  
CREATE TABLE  
dtb_pw3=# \d progress  
  
      Таблица "public.progress"  
+-----+-----+-----+-----+-----+  
| Столбец | Тип | Правило сортировки | Допустимость NULL | По умолчанию |  
+-----+-----+-----+-----+-----+  
| record_book | numeric(5,0) | | not null | |  
| subject | text | | not null | |  
| acad_year | text | | not null | |  
| term | numeric(1,0) | | not null | |  
| mark | numeric(1,0) | | not null | 5  
Ограничения-проверки:  
    "progress_mark_check" CHECK (mark >= 3::numeric AND mark <= 5::numeric)  
    "progress_term_check" CHECK (term = 1::numeric OR term = 2::numeric)  
Ограничения внешнего ключа:  
    "progress_record_book_fkey" FOREIGN KEY (record_book) REFERENCES students(record_book) ON UPDATE CASCADE ON DELETE C  
ASCADE
```

Рисунок 2 — Создание таблицы и вывод сведений

```
dtb_pw3=# ALTER TABLE progress
dtb_pw3=# ADD COLUMN test_form TEXT;
```

Рисунок 3 — Модификация столбца

```
dtb_pw3=# INSERT INTO progress
dtb_pw3=# VALUES (12300, 'Математический анализ', '1', 1, 5, 'экзамен');
INSERT 0 1
dtb_pw3=# INSERT INTO progress
dtb_pw3=# VALUES (12300, 'Введение в профессиональную деятельность', '1', 2, 1, 'зачёт');
ОШИБКА: новая строка в отношении "progress" нарушает ограничение-проверку "progress_mark_check"
ПОДРОБНОСТИ: Ошибочная строка содержит (12300, Введение в профессиональную деяте..., 1, 2, 1, зачёт).
```

Рисунок 4 — Вставка значений

Можно добавить ограничения на не пустоту subject.

2.3 Задание 3

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

```
dtb_pw3=# INSERT INTO progress
dtb_pw3=# VALUES ( 12300, 'Алгебра и Геометрия', 'экзамен' );
INSERT 0 1
```

Рисунок 5 — Вставка значений

```
dtb_pw3=# ALTER TABLE progress
dtb_pw3=# ALTER COLUMN term DROP NOT NULL,
dtb_pw3=# ALTER COLUMN mark DROP NOT NULL;
ALTER TABLE
```

Рисунок 6 — Модификация столбцов

```
dtb_pw3=# select * from progress
dtb_pw3=# ;
record_book | subject | acad_year | term | mark | test_form
-----+-----+-----+-----+-----+-----
12300 | Математический анализ | 1 | 1 | 5 | экзамен
12300 | Алгебра и Геометрия | экзамен | | 5 |
(2 строки)
```

Рисунок 7 — Вывод результата

Гипотеза подтвердилась. Проверка нужна.

2.4 Задание 4

На рисунках 8, 9 показан результат выполнения задания.

```
dtb_pw3=# ALTER TABLE progress
dtb_pw3=# ALTER COLUMN mark SET DEFAULT 6;
ALTER TABLE
```

Рисунок 8 — Модификация столбца

```
dtb_pw3=# INSERT INTO progress ( record_book, subject, acad_year, term ) VALUES ( 12300, 'Физика', '2016/2017', 1 );
ОШИБКА: новая строка в отношении "progress" нарушает ограничение-проверку "progress_check"
ПОДРОБНОСТИ: Ошибочная строка содержит (12300, Физика, 2016/2017, 1, 6, null).
```

Рисунок 9 — Вставка значений

Проверка значений происходит на этапе вставки.

2.5 Задание 5

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

```
dtb_pw3=# INSERT INTO students
dtb_pw3=# VALUES (12200, 'Иванов Иван Иванович', 401, 543281);
ОШИБКА: повторяющееся значение ключа нарушает ограничение уникальности "doc_num_unique"
ПОДРОБНОСТИ: Ключ "(doc_num)=(543281)" уже существует.
dtb_pw3=# INSERT INTO students
dtb_pw3=# VALUES (12200, 'Иванов Иван Иванович', 401);
INSERT 0 1
```

Рисунок 10 — Вставка значений, проверка, часть 1

```
dtb_pw3=# INSERT INTO students
dtb_pw3=# VALUES (12200, 'Иванов Иван Иванович', 401, 543281);
ОШИБКА: повторяющееся значение ключа нарушает ограничение уникальности "doc_num_unique"
ПОДРОБНОСТИ: Ключ "(doc_num)=(543281)" уже существует.
```

Рисунок 11 — Вставка значений, проверка, часть 2

```
dtb_pw3=# INSERT INTO students
dtb_pw3=# VALUES (12200, 'Иванов Иван Иванович', 402, 543281);
ОШИБКА: повторяющееся значение ключа нарушает ограничение уникальности "doc_ser_unique"
ПОДРОБНОСТИ: Ключ "(doc_ser)=(402)" уже существует.
```

Рисунок 12 — Вставка значений, проверка, часть 3

```
dtb_pw3=# ALTER TABLE students
dtb_pw3=# ADD CONSTRAINT doc_ser_unique UNIQUE (doc_ser),
dtb_pw3=# ADD CONSTRAINT doc_num_unique UNIQUE (doc_num);
```

Рисунок 13 — Модификация столбцов

```
dtb_pw3=# INSERT INTO students
dtb_pw3=# VALUES (12400, 'Иванов Иван Иванович');
INSERT 0 1
```

Рисунок 14 — Вставка значений, проверка

2.6 Задание 6

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

```
dtb_pw3=# ALTER TABLE public.progress
dtb_pw3=# ADD CONSTRAINT fk_progress_students
dtb_pw3=# FOREIGN KEY (doc_ser, doc_num)
dtb_pw3=# REFERENCES students (doc_ser, doc_num)
dtb_pw3=# ON DELETE CASCADE
dtb_pw3=# ON UPDATE CASCADE;
ALTER TABLE
```

Рисунок 15 — Модификация таблицы, добавление внешних ключей

```
dtb_pw3=# ALTER TABLE students
dtb_pw3=# DROP CONSTRAINT students_pkey;
ALTER TABLE
```

Рисунок 16 — Модификация таблицы, смена ключа, часть 1

```
dtb_pw3=# ALTER TABLE progress
dtb_pw3=# DROP CONSTRAINT progress_record_book_fkey;
ALTER TABLE
```

Рисунок 17 — Модификация таблицы, смена ключа, часть 2

```
dtb_pw3=# INSERT INTO students
dtb_pw3=# VALUES (11000, 'Райн Гослинг ВФильмеДрайв', 385, 500);
INSERT 0 1
```

Рисунок 18 — Вставка значений, проверка

2.7 Задание 7

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

```

dtb_pw3=# ALTER TABLE progress
dtb_pw3=# DROP CONSTRAINT fk_progress_students;
ALTER TABLE
dtb_pw3=#
dtb_pw3=# ALTER TABLE progress
dtb_pw3=# ALTER COLUMN doc_ser SET DEFAULT 0000,
dtb_pw3=# ALTER COLUMN doc_num SET DEFAULT 000000;
ALTER TABLE
dtb_pw3=#
dtb_pw3=# ALTER TABLE progress
dtb_pw3=# ADD CONSTRAINT fk_progress_students
dtb_pw3=# FOREIGN KEY (doc_ser, doc_num)
dtb_pw3=# REFERENCES students (doc_ser, doc_num)
dtb_pw3=# ON DELETE CASCADE ON UPDATE SET DEFAULT;
ALTER TABLE
dtb_pw3=#
dtb_pw3=# INSERT INTO students (record_book, name, doc_ser, doc_num)
dtb_pw3=# VALUES (10003, 'Резервный студент', 0000, 000000);
INSERT 0 1

```

Рисунок 19 — Тест ON UPDATE SET DEFAULT

```

dtb_pw3=# UPDATE students
dtb_pw3=# SET doc_ser = 1234, doc_num = 123456
dtb_pw3=# WHERE doc_ser = 9999 AND doc_num = 999999;
UPDATE 1

```

Рисунок 20 — Вставка значений, проверка

```

dtb_pw3=# ALTER TABLE progress
dtb_pw3=# DROP CONSTRAINT fk_progress_students;
ALTER TABLE
dtb_pw3=#
dtb_pw3=# ALTER TABLE progress
dtb_pw3=# ADD CONSTRAINT fk_progress_students
dtb_pw3=# FOREIGN KEY (doc_ser, doc_num)
dtb_pw3=# REFERENCES students (doc_ser, doc_num)
dtb_pw3=# ON DELETE CASCADE ON UPDATE SET NULL;
ALTER TABLE

```

Рисунок 21 — Тест ON UPDATE SET NULL

```

dtb_pw3=# UPDATE students
dtb_pw3=# SET doc_ser = 1234, doc_num = 123456
dtb_pw3=# WHERE doc_ser = 9999 AND doc_num = 999999;
ОШИБКА: UPDATE или DELETE в таблице "students" нарушает ограничение внешнего ключа "fk_progress_students" таблицы "progress"
ПОДРОБНОСТИ: На ключ (doc_ser, doc_num)=(9999, 999999) всё ещё есть ссылки в таблице "progress".

```

Рисунок 22 — Вставка значений, проверка

```

dtb_pw3=# ALTER TABLE progress
dtb_pw3=# DROP CONSTRAINT fk_progress_students;
ALTER TABLE
dtb_pw3=#
dtb_pw3=# ALTER TABLE progress
dtb_pw3=# ADD CONSTRAINT fk_progress_students
dtb_pw3=# FOREIGN KEY (doc_ser, doc_num)
dtb_pw3=# REFERENCES students (doc_ser, doc_num)
dtb_pw3=# ON DELETE CASCADE ON UPDATE RESTRICT;
ALTER TABLE

```

Рисунок 23 — Тест ON UPDATE RESTRICT

```

dtb_pw3=# UPDATE students
dtb_pw3=# SET doc_ser = 9999, doc_num = 999999
dtb_pw3=# WHERE doc_ser = 1234 AND doc_num = 123456;
UPDATE 1

```

Рисунок 24 — Вставка значений

```

dtb_pw3=# UPDATE students
dtb_pw3=# SET doc_ser = 9999, doc_num = 999999
dtb_pw3=# WHERE doc_ser = 1234 AND doc_num = 123456;
UPDATE 1

```

Рисунок 25 — Обновление значений

2.8 Задание 8

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

```

dtb_pw3=# ALTER TABLE progress
dtb_pw3=# ADD CONSTRAINT fk_progress_subjects
dtb_pw3=# FOREIGN KEY (subject)
dtb_pw3=# REFERENCES subjects (subject_id);
ALTER TABLE

```

Рисунок 26 — Модификация таблицы, добавление внешнего ключа

```

dtb_pw3=# INSERT INTO subjects VALUES
dtb_pw3=# (3, 'Химия');
INSERT 0 1

```

Рисунок 27 — Вставка данных

```

dtb_pw3=# ALTER TABLE subjects
dtb_pw3=# ALTER COLUMN subject_id SET NOT NULL
dtb_pw3=# ,
dtb_pw3=# ADD PRIMARY KEY(subject_id);
ALTER TABLE

```

Рисунок 28 — Модификация таблицы


```
dtb_pw3=# CREATE TABLE subjects (  
dtb_pw3(# subject_id INTEGER,  
dtb_pw3(# subject TEXT);
```

Рисунок 29 — Создание таблицы

```
dtb_pw3=# INSERT INTO progress VALUES  
dtb_pw3-# (5678, 654321, 3, 2024-2025, 1, 5);  
INSERT 0 1
```

Рисунок 30 — Вставка значений

2.9 Задание 9

На рисунках 31, 32 показан результат выполнения задания.

```
dtb_pw3=# ADD CONSTRAINT check_name_not_empty  
dtb_pw3-# CHECK (TRIM(name) <> '');  
ALTER TABLE
```

Рисунок 31 — Добавление ограничения

```
dtb_pw3=# INSERT INTO students VALUES  
dtb_pw3-# (10005, ' ', 1235, 123457);  
ОШИБКА: новая строка в отношении "students" нарушает ограничение-проверку "check_name_not_empty"  
ПОДРОБНОСТИ: Ошибочная строка содержит (10005, , 1235, 123457).
```

Рисунок 32 — Вставка значений

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

2.10 Задание 10

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

```
dtb_pw3=# ALTER TABLE progress ADD CONSTRAINT fk_progress_students FOREIGN KEY (doc_ser, doc_num) REFERENCES students (d  
oc_ser, doc_num) ON DELETE CASCADE ON UPDATE CASCADE;  
ALTER TABLE
```

Рисунок 33 — Модификация таблицы

```

dtb_pw3=# ALTER TABLE progress
dtb_pw3=# DROP CONSTRAINT fk_progress_students;
ALTER TABLE
dtb_pw3=# ALTER TABLE students
dtb_pw3=# DROP CONSTRAINT students_pkey;
ALTER TABLE
dtb_pw3=# ALTER TABLE students
dtb_pw3=# ALTER COLUMN doc_ser TYPE varchar(4)
dtb_pw3=# USING (LPAD(doc_ser::TEXT, 4, '0'));
ALTER TABLE
dtb_pw3=# ALTER TABLE progress
dtb_pw3=# ALTER COLUMN doc_ser TYPE varchar(4)
dtb_pw3=# USING (LPAD(doc_ser::TEXT, 4, '0'));
ALTER TABLE
dtb_pw3=# ALTER TABLE students
dtb_pw3=# ADD PRIMARY KEY (doc_ser, doc_num);
ALTER TABLE

```

Рисунок 34 — Прогресс выполнения задания

```

dtb_pw3=# select * from students;
record_book |      name      | doc_ser | doc_num
-----+-----+-----+-----
      10002 | Пётр Петров   |    5678 |    654321
      10003 | Резервный студент |    0000 |         0
      10001 | Иван Иванов   |    9999 |    999999
      10004 | Пётр Иванов   |    1234 |    123456
(4 строки)

```

Рисунок 35 — Результат

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

2.11 Задание 11

На рисунках 36, 37 показан результат выполнения задания.

```

dtb_pw3=# ALTER TABLE DROP CONSTRAINT flights_check1
dtb_pw3=# ;
ОШИБКА:  ошибка синтаксиса (примерное положение: "CONSTRAINT")
СТРОКА 1: ALTER TABLE DROP CONSTRAINT flights_check1
      ^
dtb_pw3=# ALTER TABLE flights DROP CONSTRAINT flights_check1
dtb_pw3=# ;
ALTER TABLE
dtb_pw3=# ALTER TABLE flights ADD CHECK ( actual_arrival IS NULL OR ( actual_departure IS NOT NULL AND actual_arrival >
actual_departure ) );
ALTER TABLE

```

Рисунок 36 — Модификация таблицы

```

dtb_pw3=# INSERT INTO airports (airport_code, airport_name, city, longitude, latitude, timezone)
dtb_pw3=# VALUES ('DME', 'Domodedovo', 'Moscow', 37.9063, 55.4086, 'Europe/Moscow'),
dtb_pw3=# ('SVO', 'Sheremetyevo', 'Moscow', 37.4146, 55.9726, 'Europe/Moscow');
INSERT 0 2
dtb_pw3=# INSERT INTO flights (flight_no, scheduled_departure, scheduled_arrival, departure_airport, arrival_airport, status, aircraft_code)
dtb_pw3=# VALUES ('PG001', '2025-03-01 10:00:00+03', '2025-03-01 12:00:00+03', 'DME', 'SVO', 'Scheduled', 'SU9');
INSERT 0 1
dtb_pw3=# UPDATE flights
dtb_pw3=# SET actual_departure = NULL, actual_arrival = NULL
dtb_pw3=# WHERE flight_no = 'PG001';
UPDATE 1
dtb_pw3=# UPDATE flights
dtb_pw3=# SET actual_departure = '2025-03-01 10:05:00+03', actual_arrival = NULL
dtb_pw3=# WHERE flight_no = 'PG001';
UPDATE 1
dtb_pw3=# UPDATE flights
dtb_pw3=# SET actual_departure = '2025-03-01 10:05:00+03', actual_arrival = '2025-03-01 12:05:00+03'
dtb_pw3=# WHERE flight_no = 'PG001';
UPDATE 1
dtb_pw3=# UPDATE flights
dtb_pw3=# SET actual_departure = '2025-03-01 12:05:00+03', actual_arrival = '2025-03-01 10:05:00+03'
dtb_pw3=# WHERE flight_no = 'PG001';
ОШИБКА: новая строка в отношении "flights" нарушает ограничение-проверку "flights_check1"
ПОДРОБНОСТИ: Ошибочная строка содержит (1, PG001, 2025-03-01 14:00:00+07, 2025-03-01 16:00:00+07, DME, SVO, Scheduled, SU9, 2025-03-01 16:05:00+07, 2025-03-01 14:05:00+07).
dtb_pw3=# UPDATE flights
dtb_pw3=# SET actual_departure = NULL, actual_arrival = '2025-03-01 12:05:00+03'
dtb_pw3=# WHERE flight_no = 'PG001';
ОШИБКА: новая строка в отношении "flights" нарушает ограничение-проверку "flights_check1"
ПОДРОБНОСТИ: Ошибочная строка содержит (1, PG001, 2025-03-01 14:00:00+07, 2025-03-01 16:00:00+07, DME, SVO, Scheduled, SU9, null, 2025-03-01 16:05:00+07).

```

Рисунок 37 — Прогресс выполнения задания

2.12 Задание 12

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

```

dtb_pw3=# ALTER TABLE flights RENAME TO flights_schedule;
ALTER TABLE
dtb_pw3=# \d flights_schedule

```

Столбец	Тип	Правило сортировки	Допустимость NULL	По умолчанию
flight_id	integer		not null	nextval('flights_flight_id_seq'::regclass)
flight_no	character(6)		not null	
scheduled_departure	timestamp with time zone		not null	
scheduled_arrival	timestamp with time zone		not null	
departure_airport	character(3)		not null	
arrival_airport	character(3)		not null	
status	character varying(20)		not null	
aircraft_code	character(3)		not null	
actual_departure	timestamp with time zone			
actual_arrival	timestamp with time zone			

```

Индексы:
    "flights_pkey" PRIMARY KEY, btree (flight_id)
    "flights_flight_no_scheduled_departure_key" UNIQUE CONSTRAINT, btree (flight_no, scheduled_departure)
Ограничения-проверки:
    "flights_check" CHECK (scheduled_arrival > scheduled_departure)
    "flights_check1" CHECK (actual_arrival IS NULL OR actual_departure IS NOT NULL AND actual_arrival > actual_departure)
    "flights_status_check" CHECK (status::text = ANY (ARRAY['On Time'::character varying, 'Delayed'::character varying, 'Departed'::character varying, 'Arrived'::character varying, 'Scheduled'::character varying, 'Canceled'::character varying]))
Ограничения внешнего ключа:
    "flights_aircraft_code_fkey" FOREIGN KEY (aircraft_code) REFERENCES aircrafts(aircraft_code)
    "flights_arrival_airport_fkey" FOREIGN KEY (arrival_airport) REFERENCES airports(airport_code)
    "flights_departure_airport_fkey" FOREIGN KEY (departure_airport) REFERENCES airports(airport_code)

```

Рисунок 38 — Переименование таблицы

dtb_pw3=# \d flights

Таблица "public.flights"				
Столбец	Тип	Правило сортировки	Допустимость NULL	По умолчанию
flight_id	integer		not null	nextval('flights_flight_id_se
flight_no	character(6)		not null	
scheduled_departure	timestamp with time zone		not null	
scheduled_arrival	timestamp with time zone		not null	
departure_airport	character(3)		not null	
arrival_airport	character(3)		not null	
status	character varying(20)		not null	
aircraft_code	character(3)		not null	
actual_departure	timestamp with time zone			
actual_arrival	timestamp with time zone			

Индексы:
 "flights_pkey" PRIMARY KEY, btree (flight_id)
 "flights_flight_no_scheduled_departure_key" UNIQUE CONSTRAINT, btree (flight_no, scheduled_departure)

Ограничения-проверки:
 "flights_check" CHECK (scheduled_arrival > scheduled_departure)
 "flights_check1" CHECK (actual_arrival IS NULL OR actual_departure IS NOT NULL AND actual_arrival > actual_departure)

Ограничения внешнего ключа:
 "flights_aircraft_code_fkey" FOREIGN KEY (aircraft_code) REFERENCES aircrafts(aircraft_code)
 "flights_arrival_airport_fkey" FOREIGN KEY (arrival_airport) REFERENCES airports(airport_code)
 "flights_departure_airport_fkey" FOREIGN KEY (departure_airport) REFERENCES airports(airport_code)

Рисунок 39 — Вид таблицы

```
dtb_pw3=# INSERT INTO aircrafts (aircraft_code, model, range) VALUES ('SU9', 'Sukhoi Superjet', 3000);
INSERT 0 1
dtb_pw3=# INSERT INTO airports (airport_code, airport_name, city, longitude, latitude, timezone)VALUES ('DME', 'Domodedovo', 'Moscow', 37.9063, 55.4086, 'Europe/Moscow');
INSERT 0 1
dtb_pw3=# INSERT INTO flights_schedule (flight_no, scheduled_departure, scheduled_arrival, departure_airport, arrival_airport, status, aircraft_code)VALUES ('P0001', '2025-03-01 10:00:00+03', '2025-03-01 12:00:00+03', 'DME', 'DME', 'Scheduled', 'SU9');
INSERT 0 1
dtb_pw3=# INSERT INTO flights_schedule (flight_no, scheduled_departure, scheduled_arrival, departure_airport, arrival_airport, status, aircraft_code)VALUES ('P0002', '2025-03-01 14:00:00+03', '2025-03-01 16:00:00+03', 'XXX', 'DME', 'Scheduled', 'SU9');
ОШИБКА: INSERT или UPDATE в таблице "flights_schedule" нарушает ограничение внешнего ключа "flights_departure_airport_fkey"
СУПЕРПОМОЩЬ: Ключ (departure_airport)=(XXX) отсутствует в таблице "airports".
```

Рисунок 40 — Вставка значений

Ответ на вопрос в задании: да, ограничения перенеслись.

2.13 Задание 13

На рисунке 41 показан результат выполнения задания.

```

dtb_pw3=# CREATE MATERIALIZED VIEW routes AS
dtb_pw3=# SELECT
dtb_pw3=#     f.flight_no,
dtb_pw3=#     f.departure_airport,
dtb_pw3=#     da.airport_name AS departure_airport_name,
dtb_pw3=#     da.city AS departure_city,
dtb_pw3=#     f.arrival_airport,
dtb_pw3=#     aa.airport_name AS arrival_airport_name,
dtb_pw3=#     aa.city AS arrival_city,
dtb_pw3=#     f.aircraft_code, -- Код самолёта IATA
dtb_pw3=#     (f.scheduled_arrival - f.scheduled_departure) AS duration,
dtb_pw3=#     ARRAYSPLIT(EXTRACT(DOW FROM f.scheduled_departure)::INTEGER + 1) AS days_of_week
dtb_pw3=# FROM flights f
dtb_pw3=# LEFT JOIN airports da ON f.departure_airport = da.airport_code
dtb_pw3=# LEFT JOIN airports aa ON f.arrival_airport = aa.airport_code;
SELECT 1
dtb_pw3=# select routes;
ОШИБКА: столбец "routes" не существует
ЦПРОВА 1: select routes;
dtb_pw3=# select * from routes;
 flight_no | departure_airport | departure_city | arrival_airport | arrival_airport_name | arrival_city | aircraft_code | duration | days_of_week
-----
 PG001    | DME               | Domodedovo    | Moscow         | Moscow              | Moscow      | SU9           | 02:00:00 | {7}
(1 строка)

dtb_pw3=# DROP TABLE airports;
ОШИБКА: удалить объект таблицы airports нельзя, так как от него зависят другие объекты
ПОДРОБНОСТИ: ограничение flights_arrival_airport_fkey в отношении таблицы flights зависит от объекта таблицы airports
ограничение flights_departure_airport_fkey в отношении таблицы flights зависит от объекта таблицы airports
материализованное представление routes зависит от объекта таблицы airports
ПОДСКАЗКА: Для удаления зависимых объектов используйте DROP ... CASCADE.
dtb_pw3=# DROP TABLE airports CASCADE;
ЗАМЕЧАНИЕ: удаление распространяется на ещё 3 объекта
ПОДРОБНОСТИ: удаление распространяется на объект ограничение flights_arrival_airport_fkey в отношении таблицы flights
удаление распространяется на объект ограничение flights_departure_airport_fkey в отношении таблицы flights
удаление распространяется на объект материализованное представление routes
DROP TABLE

```

Рисунок 41 — Создание таблицы

2.14 Задание 14

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

```

dtb_pw3=# INSERT INTO aircrafts_view (aircraft_code, model, range)
dtb_pw3=# VALUES ('B73', 'Boeing 737', 5500);
INSERT 0 1
dtb_pw3=# UPDATE aircrafts_view
dtb_pw3=# SET range = 3500
dtb_pw3=# WHERE aircraft_code = 'SU9';
UPDATE 1
dtb_pw3=# DELETE FROM aircrafts_view
dtb_pw3=# WHERE aircraft_code = 'A32';
DELETE 1
dtb_pw3=# INSERT INTO aircrafts_view (aircraft_code, model, range)
dtb_pw3=# VALUES ('A31', 'Airbus A321', -100);
ОШИБКА: новая строка в отношении "aircrafts" нарушает ограничение-проверку "aircrafts_range_check"
ПОДРОБНОСТИ: Ошибочная строка содержит (A31, Airbus A321, -100).
dtb_pw3=# INSERT INTO aircrafts_view (aircraft_code, model, range)
dtb_pw3=# VALUES ('SU9', 'Sukhoi Superjet 2', 4000);
ОШИБКА: повторяющееся значение ключа нарушает ограничение уникальности "aircrafts_pkey"
ПОДРОБНОСТИ: Ключ "(aircraft_code)=(SU9)" уже существует.

```

Рисунок 42 — Проверка обычного представления

```

dtb_pw3=# CREATE VIEW aircrafts_view AS
dtb_pw3=# SELECT
dtb_pw3=#     aircraft_code,
dtb_pw3=#     model,
dtb_pw3=#     range
dtb_pw3=# FROM aircrafts;
CREATE VIEW

```

Рисунок 43 — Создание представления

```
dtb_pw3=# INSERT INTO aircrafts (aircraft_code, model, range)
dtb_pw3=# VALUES
dtb_pw3-#      ('SU9', 'Sukhoi Superjet', 3000),
dtb_pw3-#      ('A32', 'Airbus A320', 6100);
INSERT 0 2
dtb_pw3=# SELECT * FROM aircrafts_view;
 aircraft_code |      model      | range
-----+-----+-----
SU9             | Sukhoi Superjet | 3000
A32             | Airbus A320     | 6100
(2 строки)
```

Рисунок 44 — Вставка значений в таблицу

2.15 Задание 15

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

```
dtb_pw3=# \dm

Схема |      Имя      | Список отношений | Владелец
-----+-----+-----+-----
public | flight_routes | материализованное представление | postgres
(1 строка)
```

Рисунок 45 — Созданное материализованное представление

```
dtb_pw3=# ALTER MATERIALIZED VIEW routes RENAME TO flight_routes;
ALTER MATERIALIZED VIEW
```

Рисунок 46 — Модификация представления

```
dtb_pw3=# INSERT INTO airports (airport_code, airport_name, city, longitude, latitude, timezone) VALUES ('DME', 'Domodedovo', 'Moscow', 37.9063, 55.4088, 'Europe/Moscow');
INSERT 0 2
dtb_pw3=# INSERT INTO flights (flight_no, scheduled_departure, scheduled_arrival, departure_airport, arrival_airport, status, aircraft_code) VALUES ('PG001', '2025-03-03 10:00:00+03', '2025-03-03 12:00:00+03', 'DME', 'SVO', 'Scheduled', 'SU9');
INSERT 0 1
dtb_pw3=# ALTER VIEW flights_v RENAME TO flights_info;
ALTER VIEW
dtb_pw3=# \dv

Схема |      Имя      | Список отношений | Тип | Владелец
-----+-----+-----+-----+-----
public | aircrafts_view | представление | postgres
public | aircrafts_view1 | представление | postgres
public | flights_info | представление | postgres
(3 строки)

dtb_pw3=# SELECT * FROM flights_info;
 flight_no | departure_airport | departure_airport_name | arrival_airport | arrival_airport_name | status
-----+-----+-----+-----+-----+-----
PG001     | DME               | Domodedovo             | SVO              | Sheremetyevo         | Scheduled
(1 строка)

dtb_pw3=# SELECT * FROM flights;
 flight_id | flight_no | scheduled_departure | scheduled_arrival | departure_airport | arrival_airport | status | aircraft_code | actual_departure | actual_arrival
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
2         | PG001    | 2025-03-03 14:00:00+07 | 2025-03-03 16:00:00+07 | DME               | SVO              | Scheduled | SU9            |                  |
(1 строка)
```

Рисунок 47 — Вставка значений, проверка

```
dtb_pw3=# ALTER MATERIALIZED VIEW flight_routes
dtb_pw3=# RENAME COLUMN departure_city TO dep_city;
ALTER MATERIALIZED VIEW
dtb_pw3=# \dm

Схема |      Имя      | Список отношений | Владелец
-----+-----+-----+-----
public | flight_routes | материализованное представление | postgres
(1 строка)
```

Рисунок 48 — Модификация представления

2.16 Задание 16

На рисунке 49 показан результат выполнения задания.

```
dtb_pw3=# INSERT INTO flights (flight_no, scheduled_departure, scheduled_arrival, departure_airport, arrival_airport, status, aircraft_code)
dtb_pw3=# VALUES
dtb_pw3=# ('PG002', '2025-03-04 14:00:00+03', '2025-03-04 16:00:00+03', 'SVO', 'DME', 'Scheduled', 'SU9');
INSERT 0 1
dtb_pw3=# SELECT * FROM flights;
 flight_id | flight_no | scheduled_departure | scheduled_arrival | departure_airport | arrival_airport | status | aircraft_code | actual_departure | actual_arrival
-----
1 | PG001 | 2025-03-03 14:00:00+07 | 2025-03-03 16:00:00+07 | DME | SVO | Scheduled | SU9 |
2 | PG002 | 2025-03-04 18:00:00+07 | 2025-03-04 20:00:00+07 | SVO | DME | Scheduled | SU9 |
(2 строки)

dtb_pw3=# SELECT * FROM routes;
 flight_no | departure_airport | departure_airport_name | departure_city | arrival_airport | arrival_airport_name | arrival_city | aircraft_code | duration | days_of_week
-----
PG001 | DME | Domodedovo | Moscow | SVO | Sheremetyevo | Moscow | SU9 | 02:00:00 | {2}
(1 строка)

dtb_pw3=# REFRESH MATERIALIZED VIEW routes;
REFRESH MATERIALIZED VIEW
dtb_pw3=# SELECT * FROM routes;
 flight_no | departure_airport | departure_airport_name | departure_city | arrival_airport | arrival_airport_name | arrival_city | aircraft_code | duration | days_of_week
-----
PG001 | DME | Domodedovo | Moscow | SVO | Sheremetyevo | Moscow | SU9 | 02:00:00 | {2}
PG002 | SVO | Sheremetyevo | Moscow | DME | Domodedovo | Moscow | SU9 | 02:00:00 | {3}
(2 строки)
```

Рисунок 49 — Вставка значений

2.17 Задание 17

На рисунке 50 показан результат выполнения задания.

```

dtb_pw3=# CREATE VIEW pilot_flights AS
dtb_pw3=# SELECT
dtb_pw3=#     f.flight_no,
dtb_pw3=#     f.scheduled_departure,
dtb_pw3=#     f.scheduled_arrival,
dtb_pw3=#     f.departure_airport,
dtb_pw3=#     f.arrival_airport,
dtb_pw3=#     f.aircraft_code,
dtb_pw3=#     a.model AS aircraft_model,
dtb_pw3=#     f.status
dtb_pw3=# FROM flights f
dtb_pw3=# JOIN aircrafts a ON f.aircraft_code = a.aircraft_code
dtb_pw3=# WHERE f.status IN ('Scheduled', 'Departed');
CREATE VIEW
dtb_pw3=# SELECT * FROM pilot_flights;
flight_no | scheduled_departure | scheduled_arrival | departure_airport | arrival_airport | aircraft_code | aircraft_model | status
-----+-----+-----+-----+-----+-----+-----+-----
PG002     | 2025-03-04 18:00:00+07 | 2025-03-04 20:00:00+07 | SVO               | DME             | SU9           | Sukhoi Superjet | Scheduled
PG001     | 2025-03-03 14:00:00+07 | 2025-03-03 16:00:00+07 | DME               | SVO             | SU9           | Sukhoi Superjet | Scheduled
(2 строки)

dtb_pw3=#
dtb_pw3=# CREATE VIEW dispatcher_status AS
dtb_pw3=# SELECT
dtb_pw3=#     f.flight_no,
dtb_pw3=#     f.scheduled_departure,
dtb_pw3=#     f.scheduled_arrival,
dtb_pw3=#     f.departure_airport,
dtb_pw3=#     f.arrival_airport,
dtb_pw3=#     f.status,
dtb_pw3=#     f.actual_departure,
dtb_pw3=#     f.actual_arrival
dtb_pw3=# FROM flights f;
CREATE VIEW
dtb_pw3=# SELECT * FROM dispatcher_status;
flight_no | scheduled_departure | scheduled_arrival | departure_airport | arrival_airport | status | actual_departure | actual_arrival
-----+-----+-----+-----+-----+-----+-----+-----
PG001     | 2025-03-03 14:00:00+07 | 2025-03-03 16:00:00+07 | DME               | SVO             | Scheduled |                  |
PG002     | 2025-03-04 18:00:00+07 | 2025-03-04 20:00:00+07 | SVO               | DME             | Scheduled |                  |
(2 строки)

dtb_pw3=# CREATE VIEW passenger_schedule AS
dtb_pw3=# SELECT
dtb_pw3=#     f.flight_no,
dtb_pw3=#     f.scheduled_departure,
dtb_pw3=#     da.airport_name AS departure_airport_name,
dtb_pw3=#     da.city AS departure_city,
dtb_pw3=#     f.scheduled_arrival,
dtb_pw3=#     aa.airport_name AS arrival_airport_name,
dtb_pw3=#     aa.city AS arrival_city,
dtb_pw3=#     f.status
dtb_pw3=# FROM flights f
dtb_pw3=# JOIN airports da ON f.departure_airport = da.airport_code
dtb_pw3=# JOIN airports aa ON f.arrival_airport = aa.airport_code
dtb_pw3=# WHERE f.scheduled_departure > CURRENT_TIMESTAMP;
CREATE VIEW
dtb_pw3=# SELECT * FROM passenger_schedule;
flight_no | scheduled_departure | departure_airport_name | departure_city | scheduled_arrival | arrival_airport_name | arrival_city | status
-----+-----+-----+-----+-----+-----+-----+-----
PG001     | 2025-03-03 14:00:00+07 | Domodedovo            | Moscow        | 2025-03-03 16:00:00+07 | Sheremetyevo        | Moscow      | Scheduled
PG002     | 2025-03-04 18:00:00+07 | Sheremetyevo          | Moscow        | 2025-03-04 20:00:00+07 | Domodedovo          | Moscow      | Scheduled
(2 строки)

```

Рисунок 50 — Тест представлений

2.18 Задание 18

На рисунке 51 показан результат выполнения задания.


```

dtb_pw3=# ALTER TABLE airports
dtb_pw3=# ADD COLUMN additional_info JSONB;
ALTER TABLE
dtb_pw3=#
dtb_pw3=# -- Вставка данных
dtb_pw3=# INSERT INTO airports (airport_code, airport_name, city, longitude, latitude, timezone, additional_info)
dtb_pw3=# VALUES
dtb_pw3=# ('DME', 'Domodedovo', 'Moscow', 37.9063, 55.4086, 'Europe/Moscow', '{"services": ["shops", "lounges"], "operating_hours": "24/7"}'),
dtb_pw3=# ('SVO', 'Sheremetyevo', 'Moscow', 37.4146, 55.9726, 'Europe/Moscow', '{"services": ["cafes"], "operating_hours": "24/7"}');
ОШИБКА: повторяющееся значение ключа нарушает ограничение уникальности "airports_pkey"
ПОДРОБНОСТИ: Ключ "(airport_code)=(DME)" уже существует.
dtb_pw3=#
dtb_pw3=# -- Изменение
dtb_pw3=# UPDATE airports
dtb_pw3=# SET additional_info = additional_info || '{"weather": "clear"}'::jsonb
dtb_pw3=# WHERE airport_code = 'DME';
UPDATE 1
dtb_pw3=#
dtb_pw3=# UPDATE airports
dtb_pw3=# SET additional_info = '{"services": ["cafes", "wifi"], "operating_hours": "24/7", "weather": "snow"}'::jsonb
dtb_pw3=# WHERE airport_code = 'SVO';
UPDATE 1
dtb_pw3=# select * from airports
dtb_pw3=# ;

```

airport_code	airport_name	city	longitude	latitude	timezone	additional_info
DME	Domodedovo	Moscow	37.9063	55.4086	Europe/Moscow	{"services": ["shops", "lounges"], "operating_hours": "24/7", "weather": "clear"}
SVO	Sheremetyevo	Moscow	37.4146	55.9726	Europe/Moscow	{"services": ["cafes", "wifi"], "operating_hours": "24/7", "weather": "snow"}

(2 строки)

Рисунок 51 — Модификация таблиц, обновление значений

2.19 Задание 19

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

```

dtb_pw3=# ALTER TABLE aircrafts ADD COLUMN specifications jsonb;
ALTER TABLE

```

Рисунок 52 — Модификация столбца

```

dtb_pw3=# UPDATE flights
dtb_pw3=# SET flight_details = '{"delay_reason": "weather", "passenger_count": 180}'
dtb_pw3=# WHERE flight_no = 'PG002';
UPDATE 1
dtb_pw3=# select * from flights
dtb_pw3=# ;

```

flight_id	flight_no	scheduled_departure	scheduled_arrival	departure_airport	arrival_airport	status	aircraft_code	actual_departure	actual_arrival	flight_details
1	PG001	2025-03-03 14:00:00+07	2025-03-03 16:00:00+07	DME	SVO	Scheduled	SU9			{"delay_reason": null, "passenger_count": 150}
2	PG002	2025-03-04 16:00:00+07	2025-03-04 20:00:00+07	SVO	DME	Scheduled	SU9			{"delay_reason": "weather", "passenger_count": 180}

(2 строки)

Рисунок 53 — Обновление таблицы

```

dtb_pw3=# ALTER TABLE flights
dtb_pw3=# ADD COLUMN flight_details JSONB;
ALTER TABLE

```

Рисунок 54 — Добавление столбца

```

dtb_pw3=# UPDATE aircrafts SET specifications = '{"crew": 2, "engines": { "type": "IAE V2500", "num": 2 } }'::jsonb WHERE aircraft_code = 'SU9';
UPDATE 1
dtb_pw3=# SELECT model, specifications FROM aircrafts;

```

model	specifications
Sukhoi Superjet	{"crew": 2, "engines": {"num": 2, "type": "IAE V2500"}}

(1 строка)

Рисунок 55 — Обновление значения

3 ЗАКЛЮЧЕНИЕ

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