

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

Изменение 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 «Изменение 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 следующих частей:

- 4 Выполнить задания из главы 7 из книги на е-курсах.

2 ХОД РАБОТЫ

2.1 Задание 1

На рисунках 1, 2, 3 показан результат выполнения задания.

```
demo=# ALTER TABLE aircrafts_log  
demo=# ALTER COLUMN when_add  
demo=# SET DEFAULT current_timestamp;  
ALTER TABLE
```

Рисунок 1 — Модификация таблицы

```
demo=# WITH update_row AS (  
demo(#   UPDATE aircrafts_tmp  
demo(#   SET range = range * 1.2  
demo(#   WHERE model ~ '^Boe'  
demo(#   RETURNING aircraft_code, model, range  
demo(# )  
demo=# INSERT INTO aircrafts_log (aircraft_code, model, range, operation)  
demo=# SELECT  
demo=#   ur.aircraft_code,  
demo=#   ur.model,  
demo=#   ur.range,  
demo=#   'UPDATE'  
demo=# FROM update_row ur;  
INSERT 0 3
```

Рисунок 2 — Модификация запроса

```
demo=# WITH delete_row AS  
demo=# ( DELETE FROM aircrafts_tmp  
demo=# WHERE model ~ '^Bom'  
demo=# RETURNING *  
demo=# )  
demo=# INSERT INTO aircrafts_log (aircraft_code, model, range, operation)  
demo=# SELECT dr.aircraft_code, dr.model, dr.range, 'DELETE'  
demo=# FROM delete_row dr;  
INSERT 0 1
```

Рисунок 3 — Модификация запроса

2.2 Задание 2

На рисунке 4 показан результат выполнения задания.

```
demo=# WITH add_row AS  
demo=# ( INSERT INTO aircrafts_tmp  
demo=# SELECT * FROM aircrafts  
demo=# RETURNING aircraft_code, model, range,  
demo=# current_timestamp, 'INSERT'  
demo=# )  
demo=# INSERT INTO aircrafts_log  
demo=# SELECT * FROM add_row;
```

Рисунок 4 — Модифицированный запрос

2.3 Задание 3

На рисунке 5 показан результат выполнения задания. Возвращаемый результат будет представлять таблицу вставленных данных.

```
demo=# INSERT INTO aircrafts_tmp SELECT * FROM aircrafts RETURNING *;
```

Рисунок 5 — Запрос

2.4 Задание 4

На рисунках с 6 по 10 показан результат выполнения задания.

```
demo=# -- Создание копии таблицы
demo=# CREATE TABLE seats_copy (
demo(#   aircraft_code CHAR(3),
demo(#   seat_no VARCHAR(4),
demo(#   fare_conditions TEXT,
demo(#   PRIMARY KEY (aircraft_code, seat_no)
demo(# );
CREATE TABLE
demo=#
demo=# -- Копирование данных из seats в seats_copy
demo=# INSERT INTO seats_copy
demo=# SELECT * FROM seats;
INSERT 0 1339
```

Рисунок 6 — Создание таблицы

```
demo=# ALTER TABLE seats_copy
demo=# ADD CONSTRAINT seats_fare_conditions_check CHECK (fare_conditions::text = ANY (ARRAY['Economy'::character varying::text, 'Comfort'::character varying::text, 'Business'::character varying::text]));
ALTER TABLE
demo=#
demo=# ALTER TABLE seats_copy
demo=# ADD CONSTRAINT seats_aircraft_code_fkey FOREIGN KEY (aircraft_code) REFERENCES aircrafts(aircraft_code) ON DELETE CASCADE;
ALTER TABLE
```

Рисунок 7 — Добавление ограничений

```
demo=# INSERT INTO seats_copy (aircraft_code, seat_no, fare_conditions)
demo=# VALUES ('SU9', '1A', 'Economy')
demo=# ON CONFLICT (aircraft_code, seat_no)
demo=# DO UPDATE SET fare_conditions = EXCLUDED.fare_conditions;
INSERT 0 1
```

Рисунок 8 — Вставка вариантом 1

```
demo=# INSERT INTO seats_copy (aircraft_code, seat_no, fare_conditions)
demo=# VALUES ('SU9', '1A', 'Economy')
demo=# ON CONFLICT ON CONSTRAINT seats_copy_pkey
demo=# DO UPDATE SET fare_conditions = EXCLUDED.fare_conditions;
INSERT 0 1
```

Рисунок 9 — Вставка вариантом 2

2.5 Задание 5

На рисунке 10 показан результат выполнения задания.

```
demo=# INSERT INTO seats_copy (aircraft_code, seat_no, fare_conditions)
demo=# VALUES ('SU9', '2A', 'Business')
demo=# ON CONFLICT (aircraft_code, seat_no)
demo=# DO UPDATE SET fare_conditions = EXCLUDED.fare_conditions
demo=# WHERE seats_copy.fare_conditions = 'Business';
INSERT 0 1
```

Рисунок 10 — Применение WHERE

2.6 Задание 6

На рисунке 11, показан результат выполнения задания.

```
SELECT * FROM aircrafts_tmp;
```

aircraft_code	model	range
...		
CN1	Cessna 208 Caravan	1200
CR2	Bombardier CRJ-200	2700
IL9	Ilyushin IL96	9800
I93	Ilyushin IL96-300	9800

(11 строк)

Рисунок 11 — Пример вывода

Смещение вправо происходит из-за того, что «PostgreSQL» разделяет csv файл по разделителю, пробел он считает частью строки. Поэтому и не удаляет.

2.7 Задание 7

На рисунках 12–14 показан результат выполнения задания.

```
demo=# SELECT flight_no, flight_id, departure_city,
demo=# arrival_city, scheduled_departure
demo=# FROM flights_v
demo=# WHERE scheduled_departure
demo=# BETWEEN bookings.now() AND bookings.now() + INTERVAL '15 days'
demo=# AND ( departure_city, arrival_city ) IN
demo=# ( ( 'Красноярск', 'Москва' ),
demo=# ( 'Москва', 'Сочи' ),
demo=# ( 'Сочи', 'Москва' ),
demo=# ( 'Сочи', 'Красноярск' )
demo=# )
demo=# ORDER BY departure_city, arrival_city, scheduled_departure;
```

flight_no	flight_id	departure_city	arrival_city	scheduled_departure
PG0548	13841	Красноярск	Москва	2016-10-14 11:40:00+07
PG0548	13829	Красноярск	Москва	2016-10-15 11:40:00+07
PG0548	13860	Красноярск	Москва	2016-10-16 11:40:00+07

Рисунок 12 — Выборка данных

```

demo=# WITH sell_tickets AS
demo=# ( INSERT INTO ticket_flights_tmp
demo=# ( ticket_no, flight_id, fare_conditions, amount )
demo=# VALUES ( '1234567890123', 13829, 'Economy', 10500 ),
demo=# ( '1234567890123', 4728, 'Economy', 3400 ),
demo=# ( '1234567890123', 30523, 'Economy', 3400 ),
demo=# ( '1234567890123', 7757, 'Economy', 3400 ),
demo=# ( '1234567890123', 30829, 'Economy', 12800 )
demo=# RETURNING *
demo=# )
demo=# UPDATE tickets_directions td
demo=# SET last_ticket_time = current_timestamp,
demo=# tickets_num = tickets_num +
demo=# ( SELECT count( * )
demo=# FROM sell_tickets st, flights_v f
demo=# WHERE st.flight_id = f.flight_id
demo=# AND f.departure_city = td.departure_city
demo=# AND f.arrival_city = td.arrival_city
demo=# )
demo=# WHERE ( td.departure_city, td.arrival_city ) IN
demo=# ( SELECT departure_city, arrival_city
demo=# FROM flights_v
demo=# WHERE flight_id IN ( SELECT flight_id FROM sell_tickets )
demo=# );

```

Рисунок 13 — Обновление данных

```

demo=# SELECT * FROM aircrafts_tmp;

```

aircraft_code	model	range
320	Airbus A320-200	5700
321	Airbus A321-200	5600
319	Airbus A319-100	6700
CN1	Cessna 208 Caravan	1200
773	Boeing 777-300	13320
763	Boeing 767-300	9480
733	Boeing 737-300	5040

(7 строк)

```

demo=# COPY aircrafts_tmp
demo=# FROM 'C:/Program Files/PostgreSQL/17/data/aircrafts_tmp.csv' WITH ( FORMAT csv );
ОШИБКА: повторяющееся значение ключа нарушает ограничение уникальности "aircrafts_tmp_pkey"
ПОДРОБНОСТИ: Ключ "(aircraft_code)=(773)" уже существует.
КОНТЕКСТ: COPY aircrafts_tmp, строка 1
demo=# SELECT * FROM aircrafts_tmp;

```

aircraft_code	model	range
320	Airbus A320-200	5700
321	Airbus A321-200	5600
319	Airbus A319-100	6700
CN1	Cessna 208 Caravan	1200
773	Boeing 777-300	13320
763	Boeing 767-300	9480
733	Boeing 737-300	5040

(7 строк)

Рисунок 14 — Проверка гипотезы

2.8 Задание 8

На рисунке 15 показан результат выполнения задания.

```

demo=# CREATE TABLE tickets_directions (
demo(#   departure_city TEXT,
demo(#   arrival_city TEXT,
demo(#   fare_conditions TEXT,
demo(#   last_ticket_time TIMESTAMP,
demo(#   tickets_num INTEGER,
demo(#   PRIMARY KEY (departure_city, arrival_city, fare_conditions)
demo(# );
CREATE TABLE
demo=#
demo=# WITH sell_tickets AS (
demo(#   INSERT INTO ticket_flights_tmp (ticket_no, flight_id, fare_conditions, amount)
demo(#   VALUES
demo(#       ('1234567890123', 13829, 'Economy', 10500), -- Красноярск - Москва
demo(#       ('1234567890123', 4728, 'Economy', 3400),    -- Москва - Сочи
demo(#       ('1234567890123', 30523, 'Economy', 3400),  -- Сочи - Москва
demo(#       ('1234567890123', 7757, 'Business', 6800),  -- Москва - Сочи (добавлен Business)
demo(#       ('1234567890123', 30829, 'Comfort', 15000)  -- Сочи - Красноярск (добавлен Comfort)
demo(#   RETURNING *
demo(# )
demo=# UPDATE tickets_directions td
demo=# SET
demo=#   last_ticket_time = current_timestamp,
demo=#   tickets_num = COALESCE(tickets_num, 0) + (
demo(#       SELECT COUNT(*)
demo(#       FROM sell_tickets st
demo(#       JOIN flights_v f ON st.flight_id = f.flight_id
demo(#       WHERE f.departure_city = td.departure_city
demo(#       AND f.arrival_city = td.arrival_city
demo(#       AND st.fare_conditions = td.fare_conditions
demo(#   )
demo=# WHERE (td.departure_city, td.arrival_city, td.fare_conditions) IN (
demo(#   SELECT f.departure_city, f.arrival_city, st.fare_conditions
demo(#   FROM sell_tickets st
demo(#   JOIN flights_v f ON st.flight_id = f.flight_id
demo(# );

```

Рисунок 15 — Модифицированный запрос

2.9 Задание 9

На рисунке 21 показан результат выполнения задания.

```

demo=# CREATE VIEW aircrafts_seats AS
demo=# ( SELECT a.aircraft_code,
demo(#   a.model,
demo(#   left( a.model,
demo(#     strpos( a.model, ' ' ) - 1 ) AS company,
demo(#   count( * ) AS seats_num
demo(# FROM aircrafts a, seats s
demo(# WHERE a.aircraft_code = s.aircraft_code
demo(# GROUP BY 1, 2, 3
demo(# );
CREATE VIEW

```

Рисунок 16 — Создание представления для упрощения запроса

```

demo=# WITH aircrafts_seats AS (
demo(#   SELECT aircraft_code, company, seats_num, rank() OVER (PARTITION BY company ORDER BY seats_num)
demo(#   FROM aircrafts_seats
demo(# )
demo=# DELETE FROM aircrafts_tmp a
demo=# USING aircrafts_seats a_s
demo=# WHERE a.aircraft_code = a_s.aircraft_code
demo=# AND a_s.company IN ('Boeing', 'Airbus')
demo=# AND a_s.rank = 1
demo=# RETURNING *;

```

aircraft_code	model	range	aircraft_code	company	seats_num	rank
319	Airbus A319-100	6700	319	Airbus	116	1
733	Boeing 737-300	5040	733	Boeing	130	1

(2 строки)

Рисунок 17 — Упрощённый запрос

2.10 Задание 10

На рисунках 18 показан результат выполнения задания.

```

demo=# INSERT INTO seats ( aircraft_code, seat_no, fare_conditions )
demo=# SELECT aircraft_code, seat_row || letter, fare_condition
demo=# FROM
demo=# -- компоновки салонов
demo=# ( VALUES ( 'SU9', 3, 20, 'F' ),
demo(#   ( '773', 5, 30, 'I' ),
demo(#   ( '763', 4, 25, 'H' ),
demo(#   ( '733', 3, 20, 'F' ),
demo(#   ( '320', 5, 25, 'F' ),
demo(#   ( '321', 4, 20, 'F' ),
demo(#   ( '319', 3, 20, 'F' ),
demo(#   ( 'CN1', 0, 10, 'B' ),
demo(#   ( 'CR2', 2, 15, 'D' )
demo(# ) AS aircraft_info ( aircraft_code, max_seat_row_business, max_seat_row_economy, max_letter )
demo=# CROSS JOIN
demo=# -- классы обслуживания
demo=# ( VALUES ( 'Business' ), ( 'Economy' ) ) AS fare_conditions ( fare_condition )
demo=# CROSS JOIN
demo=# -- список номеров рядов кресел
demo=# generate_series(1, 30) AS seat_rows ( seat_row )
demo=# CROSS JOIN
demo=# -- список номеров (позиций) кресел в ряду
demo=# ( VALUES ( 'A' ), ( 'B' ), ( 'C' ), ( 'D' ), ( 'E' ), ( 'F' ), ( 'G' ), ( 'H' ), ( 'I' ) ) AS letters ( letter )
demo=# WHERE
demo=# CASE WHEN fare_condition = 'Business'
demo=#   THEN seat_row::integer <= max_seat_row_business AND ascii(letter) <= ascii(max_letter) - 2 -- 2 места меньше в ряду для бизнес-класса
demo=#   WHEN fare_condition = 'Economy'
demo=#   THEN seat_row::integer > max_seat_row_business AND seat_row::integer <= max_seat_row_economy AND letter <= max_letter
demo=# END
demo=# RETURNING *;

```

Рисунок 18 — Улучшенный запрос

3 ЗАКЛЮЧЕНИЕ

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