Лабораторная работа №6

Тема: Глава 7 из книги. Упр 1, 2, 4

Группа: M8O-109CB-24

Выполнил: **Гимазетдинов Дмитрий Русланович**

вернуться на главную

Упражнение 1

Дано:

Добавьте в определение таблицы `aircrafts_log` значение по умолчанию current_timestamp и соответствующим образом измените команды INSERT, приведенные в тексте главы.

Решение:

Пусть значением по умолчанию будет первое января 2024 года.

```
ALTER TABLE aircrafts_log
ADD COLUMN when_add timestamp default '01-01-2024'::timestamp;
```

Я допустил ошибку, надое ее поправить:

```
ALTER TABLE aircrafts_log
ALTER COLUMN when_add SET DEFAULT current_timestamp;
```

```
                                                  Table
"pg_temp_4.aircrafts_log"
    Column     |            Type             | Collation | Nullable |
Default      | Storage  | Compression | Stats target | Description
---------------+-----------------------------+-----------+----------+------
-------------+----------+-------------+--------------+-------------
 aircraft_code | character(3)                |           |          |
| extended |             |              |
 model         | text                        |           |          |
| extended |             |              |
 range         | integer                     |           |          |
| plain    |             |              |
 when_add      | timestamp without time zone |           |          |
CURRENT_TIMESTAMP | plain    |             |              |
 operation     | text                        |           |          |
```

```
| extended |             |              |
Access method: heap
```

А теперь можно идти дальше. Изменим команды `INSERT` из главы 7 учебника!

```
WITH add_row AS (
    INSERT INTO aircrafts_tmp
    SELECT * FROM aircrafts
    RETURNING *
)
INSERT INTO aircrafts_log (aircraft_code, model, range, operation)
SELECT add_row.aircraft_code, add_row.model, add_row.range, 'INSERT'
FROM add_row;
```

В конфликтной ситуации:

```
WITH add_row AS
( INSERT INTO aircrafts_tmp
    VALUES ( 'SU9', 'Sukhoi SuperJet-100', 3000 )
    ON CONFLICT DO NOTHING
    RETURNING *
)
INSERT INTO aircrafts_log (aircraft_code, model, range, operation)
SELECT add_row.aircraft_code, add_row.model, add_row.range,
current_timestamp, 'INSERT'
FROM add_row;
```

```
WITH add_row AS
( INSERT INTO aircrafts_tmp
  VALUES ( 'SU9', 'Sukhoi SuperJet-100', 3000 )
  ON CONFLICT DO NOTHING
  RETURNING *
)
INSERT INTO aircrafts_log (aircraft_code, model, range, operation)
SELECT add_row.aircraft_code, add_row.model, add_row.range,
current_timestamp, 'INSERT'
FROM add_row;
```

```
WITH add_row AS
(
    INSERT INTO aircrafts_tmp
    VALUES ( 'S99', 'Sukhoi SuperJet-100', 3000 )
    ON CONFLICT ( aircraft_code ) DO NOTHING
    RETURNING *;
)
```

```
INSERT INTO aircrafts_log (aircraft_code, model, range, operation)
SELECT add_row.aircraft_code, add_row.model, add_row.range,
current_timestamp, 'INSERT'
FROM add_row;
```

Упражнение 2

Дано:

В предложении RETURNING можно указывать не только символ «∗», означающий выбор всех столбцов таблицы, но и более сложные выражения, сформированные на основе этих столбцов. В тексте главы мы копировали содержимое таблицы «Самолеты» в таблицу aircrafts_tmp, используя в предложении RETURNING именно «∗». Однако возможен и другой вариант запроса:

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

Что нужно написать в этом запросе вместо вопросительного знака?

Решение:

Вместо вопросительного знака в данном запросе необходимо перечислить те столбцы, которые должны быть вставлены в таблицу `aircrafts_log`, а именно:

```
aircraft_code, model, range, current_timestamp, INSERT
```

т.е.

```
WITH add_row AS (
    INSERT INTO aircrafts_tmp
    SELECT * FROM aircrafts
    RETURNING aircraft_code, model, range, current_timestamp, 'INSERT'
)
INSERT INTO aircrafts_log (aircraft_code, model, range, when_add,
action_type)
```

```
SELECT aircraft_code, model, range, current_timestamp, 'INSERT'
FROM add_row;
```

Упражнение 4

Дано:

В тексте главы в предложениях ON CONFLICT команды INSERT мы использовали только выражения, состоящие из имени одного столбца. Однако в таблице «Места» (seats) первичный ключ является составным и включает два столбца. Напишите команду INSERT для вставки новой строки в эту таблицу и предусмотрите возможный конфликт добавляемой строки со строкой, уже имеющейся в таблице. Сделайте два варианта предложения ON CONFLICT: первый — с использованием перечисления имен столбцов для проверки наличия дублирования, второй — с использованием предложения ON CONSTRAINT.

Решение:

Попробуем ввести значение, которое уже имеется.

```
INSERT INTO seats (aircraft_code, seat_no, fare_conditions, meals)
VALUES ('319', '2A', 'Economy', '{"breakfast": "true"}')
ON CONFLICT (aircraft_code, seat_no)
DO UPDATE SET fare_conditions = EXCLUDED.fare_conditions;
```

Как видим конфлик обошелся успешно!

```
demo=# INSERT INTO seats (aircraft_code, seat_no, fare_conditions, meals)
VALUES ('319', '2A', 'Economy', '{"breakfast": "true"}')
ON CONFLICT (aircraft_code, seat_no)
DO UPDATE SET fare_conditions = EXCLUDED.fare_conditions;
INSERT 0 1
```

Произведем поиск по суррогатному ключу.

```
demo=# select * from seats where aircraft_code = '319' and seat_no = '2A'
demo-# ;
 aircraft_code | seat_no | fare_conditions | meals
---------------+---------+-----------------+-------
 319           | 2A      | Economy         |
(1 row)
```

В этом варианте мы ссылаемся на конкретное ограничение (индекс), определяющее первичный ключ, а не перечисляем столбцы:

```
INSERT INTO seats_copy (aircraft_code, seat_no, fare_conditions, meals)
VALUES ('ABC', '12A', 'Economy', '{"breakfast": "true"}')
ON CONFLICT ON CONSTRAINT seats_pkey
DO UPDATE SET fare_conditions = EXCLUDED.fare_conditions;
```

Как видно если обращаться исключительно к суррогатному ключу, то тоже все работает:

```
demo=# INSERT INTO seats (aircraft_code, seat_no, fare_conditions, meals)
VALUES ('319', '2A', 'Comfort', '{"breakfast": "true"}')
ON CONFLICT ON CONSTRAINT seats_pkey
DO UPDATE SET fare_conditions = EXCLUDED.fare_conditions;
INSERT 0 1
demo=# select * from seats where aircraft_code = '319' and seat_no = '2A';
 aircraft_code | seat_no | fare_conditions | meals
---------------+---------+-----------------+-------
 319           | 2A      | Comfort         |
(1 row)
```
