Мамонов Антон 2ИСиП-19-1

1 Часть

Создаем таблицу users

```sql
CREATE TABLE users(
    id integer PRIMARY KEY GENERATED ALWAYS AS IDENTITY,
    name text
);
```

Создаем таблицу orders

```sql
CREATE TABLE orders(
    id integer PRIMARY KEY GENERATED ALWAYS AS IDENTITY,
    user_id integer REFERENCES users(id),
    amount numeric
);
```

Заполняем данными таблицу users

```sql
INSERT INTO users(name)
VALUES
    ('alice'),
    ('bob'),
    ('charlie');
```

Заполняем данными таблицу orders

```sql
INSERT INTO orders(amount, user_id)
    SELECT round( (random()*1000)::numeric, 2), u.id
    FROM users u, generate_series(1,3);
```

Создаем функцию для выбора данных из таблиц orders и users

```sql
CREATE FUNCTION get_users_w_orders(user_id integer) RETURNS jsonb
AS $$
SELECT jsonb_build_object(
    'user_id', u.id,
    'name',    u.name,
    'orders',  jsonb_agg(jsonb_build_object(
        'order_id', o.id,
        'amount',   o.amount
    ))
)
FROM users u
    JOIN orders o ON o.user_id = u.id
WHERE u.id = get_users_w_orders.user_id
GROUP BY u.id;
$$ LANGUAGE sql STABLE;
```

Проверяем

```
SELECT jsonb_pretty(get_users_w_orders(1));
```


```
{
  "name": "alice",
  "orders": [
    {
      "amount": 47.68,
      "order_id": 1
    },
    {
      "amount": 51.89,
      "order_id": 4
    },
    {
      "amount": 267.79,
      "order_id": 7
    }
  ],
  "user_id": 1
}
```


OK

Часть 2

Создаем таблицу cities_ml

```
CREATE TABLE cities_ml(
    id integer PRIMARY KEY GENERATED ALWAYS AS IDENTITY,
    data jsonb
);
```

Заполняем данными таблицу cities_ml

```
INSERT INTO cities_ml(data) VALUES
    ('{ "ru": "Москва",
        "en": "Moscow",
        "it": "Mosca" }'),
    ('{ "ru": "Санкт-Петербург",
        "en": "Saint Petersburg",
        "it": "San Pietroburgo" }');
```

Создаем представление для выборки данных таблицы cities_ml

```
CREATE VIEW cities AS
SELECT c.id,
       c.data ->> current_setting(
                      'translation.lang', /* missing_ok */true
                  ) AS name
FROM cities_ml c;
```

Устанавливаем значение параметра

```
1  SET translation.lang = 'ru';
```

Проверяем

```
1  SELECT * FROM cities;
```

Результат | План выполнения | Notificat

	id integer	name text
1	1	Москва
2	2	Санкт-Петербург

Ставим другое значение параметра

```
1  SET translation.lang = 'en';
```

Проверяем

```
1  SELECT * FROM cities;
```

Результат | План выполнения | Notificatio

	id integer	name text
1	1	Moscow
2	2	Saint Petersburg