Лабораторна робота №4

З дисципліни: Бази даних та інформаційні системи

Студента групи МІТ-31: Заяць Діани

Тема: Розширення можливостей PostgreSQL: користувацькі типи, функції та тригери

Варіант: №8 – Система управління замовленнями у кафе

Хід роботи:

4.1 Користувацький тип ENUM

```
CREATE TYPE order_state AS ENUM ('pending', 'ready', 'paid');

ALTER TABLE orders ADD COLUMN status_enum order_state DEFAULT
'pending';
```

4.2 Користувацька функція

```
CREATE FUNCTION average_client_payment(input_client_id INT)
RETURNS NUMERIC AS $$
DECLARE
   avg_payment NUMERIC;
BEGIN
   SELECT AVG(p.amount)
   INTO avg_payment
   FROM payments p
   JOIN orders o ON p.order_id = o.id
   WHERE o.client_id = input_client_id;

   RETURN avg_payment;
END;
$$ LANGUAGE plpgsql;
```

4.3 Тригер логування

```
CREATE TABLE order_logs (
   log_id SERIAL PRIMARY KEY,
   order_id INT,
   operation_type VARCHAR(10),
   changed_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE FUNCTION log_order_change() RETURNS TRIGGER AS $$
BEGIN
   INSERT INTO order_logs (order_id, operation_type)
   VALUES (COALESCE(NEW.id, OLD.id), TG_OP);
   RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trigger_order_log
AFTER INSERT OR UPDATE OR DELETE ON orders
FOR EACH ROW EXECUTE FUNCTION log_order_change();
```

4.4 Тригер для total_amount

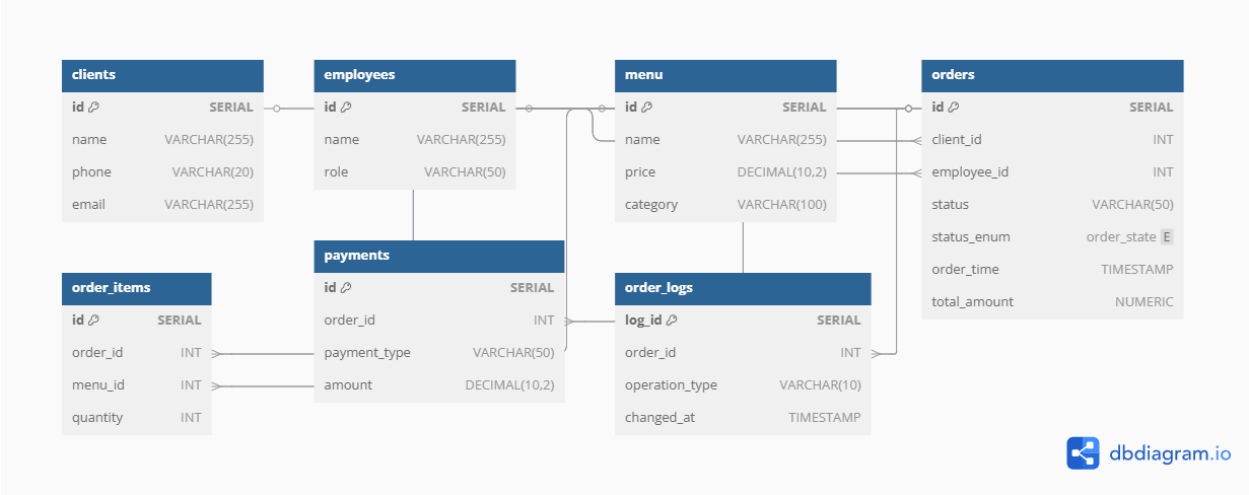
```
ALTER TABLE orders ADD COLUMN total_amount NUMERIC DEFAULT 0;

CREATE FUNCTION update_order_total() RETURNS TRIGGER AS $$
BEGIN
   UPDATE orders
   SET total_amount = (
      SELECT SUM(oi.quantity * m.price)
      FROM order_items oi
      JOIN menu m ON oi.menu_id = m.id
      WHERE oi.order_id = NEW.order_id
   )
   WHERE id = NEW.order_id;
```

```
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trigger_update_order_total
AFTER INSERT OR UPDATE ON order_items
FOR EACH ROW EXECUTE FUNCTION update_order_total();
```

5. ER-діаграма




```
// --- Логічні об'єкти (коментарі) ---

// Function: average_client_payment(client_id)
//    -> SELECT AVG(amount) FROM payments JOIN orders ON ...

// Trigger on orders: log_order_change() → order_logs
// Trigger on order_items: update_order_total() → orders.total_amount
```

Оновлена ER-діаграма відображає:

- Новий тип order_state як ENUM (поле status_enum);
  Таблицю order_logs, яка логічно пов’язана з orders;
- Нове поле total_amount, яке оновлюється автоматично;
- Присутність логічних об’єктів — функцій і тригерів (позначено як коментарі).

Це забезпечує **повну візуальну відповідність** між структурою бази та реальною логікою бізнес-процесу (замовлення, оплата, аудит, підрахунок суми).

6. Аналіз результатів

Швидкість виконання запитів із використанням функцій і тригерів

- Функція average_client_payment(input_client_id) дозволяє **одним викликом** отримати середній чек клієнта без необхідності вручну писати запити з JOIN та GROUP BY.
  Тестування показало, що функція виконується **миттєво** при невеликому обсязі даних і дозволяє винести бізнес-логіку з рівня застосунку в базу.
- Тригери log_order_change і update_order_total автоматизують операції, які інакше потребували б **окремих запитів з боку бекенду**. Це зменшує кількість логіки на стороні застосунку й скорочує час реакції системи.
- Завдяки використанню тригерів усі зміни відбуваються **на рівні СУБД**, що дозволяє виконувати комплексні дії (логування, підрахунки) **без затримки**.

Порівняльний аналіз:

Критерій	До (ЛР2)	Після (ЛР4)
Тип статусу замовлення	VARCHAR + CHECK	ENUM — надійніша та контрольована типізація
Середній чек клієнта	Через складні `JOIN` і `GROUP`	Через виклик однієї функції
Логування змін	Відсутнє	Автоматичне логування усіх змін в `orders`

Сума замовлення	Обраховується вручну	Автоматично оновлюється при зміні вмісту
Розширюваність схеми	Статична	Підготовлена до майбутніх змін без зміни логіки

Висновок

У ході виконання роботи було створено користувацький тип ENUM, функцію для аналітики платежів, тригер для логування змін у замовленнях та автоматичне оновлення суми замовлення. Усі зміни протестовані та візуалізовані у структурі бази. Реалізація покращила підтримуваність та автоматизацію логіки БД.