В моём приложении используются следующие SQL-запросы:

**1. Запрос для получения упрощенного отчета по остаткам:**

*SELECT*

*m.medicine\_id as id,*

*m.name as medicine,*

*m.quantity\_in\_stock as stock,*

*s.name as supplier*

*FROM medicines m*

*LEFT JOIN suppliers s ON m.supplier\_id = s.supplier\_id*

*ORDER BY m.quantity\_in\_stock ASC*

Получает информацию о лекарствах, их остатках и поставщиках, сортируя по возрастанию количества оставшихся единиц.

**2. Запрос будущих поставок:**

*SELECT p.purchase\_id, p.purchase\_date, s.name as supplier,*

*m.name as medicine, pi.quantity, pi.price*

*FROM purchases p*

*JOIN purchase\_items pi ON p.purchase\_id = pi.purchase\_id*

*JOIN medicines m ON pi.medicine\_id = m.medicine\_id*

*JOIN suppliers s ON p.supplier\_id = s.supplier\_id*

*WHERE p.purchase\_date > CURRENT\_DATE*

*ORDER BY p.purchase\_date*

Выбирает информацию о предстоящих поставках (после текущей даты), включая данные о поставщике, лекарствах, количестве и цене.

**3. Запрос продаж за последние 30 дней:**

*SELECT s.sale\_id, s.sale\_date,*

*m.name as medicine, si.quantity, si.price*

*FROM sales s*

*JOIN sale\_items si ON s.sale\_id = si.sale\_id*

*JOIN medicines m ON si.medicine\_id = m.medicine\_id*

*WHERE s.sale\_date >= CURRENT\_DATE - INTERVAL '30 days'*

*ORDER BY s.sale\_date DESC*

Показывает продажи за последний месяц, включая название лекарства, количество и цену, сортируя по дате продажи в обратном порядке.

**4. Запрос статистики расхода:**

*SELECT m.medicine\_id, m.name,*

*COALESCE(SUM(si.quantity), 0) as total\_sold,*

*COALESCE(SUM(si.quantity), 0) / 30.0 as avg\_daily\_consumption,*

*m.quantity\_in\_stock as current\_stock*

*FROM medicines m*

*LEFT JOIN sale\_items si ON m.medicine\_id = si.medicine\_id*

*LEFT JOIN sales s ON si.sale\_id = s.sale\_id*

*WHERE s.sale\_date >= CURRENT\_DATE - INTERVAL '30 days'*

*GROUP BY m.medicine\_id, m.name, m.quantity\_in\_stock*

*ORDER BY avg\_daily\_consumption DESC*

Вычисляет статистику продаж за последние 30 дней, включая общее количество проданных единиц, среднесуточное потребление и текущие остатки.

**5. Запросы для таблиц:**

- Клиенты:

*SELECT customer\_id, first\_name, last\_name, phone, email, address FROM customers*

- Сотрудники:

*SELECT employee\_id, first\_name, last\_name, position, phone, email, hire\_date, salary FROM employees*

- Поставщики:

*SELECT supplier\_id, name, contact\_person, phone, email, address FROM suppliers*

- Лекарства:

*SELECT m.medicine\_id, m.name, m.description, m.manufacturer,*

*m.price, m.quantity\_in\_stock, m.expiration\_date,*

*s.name as supplier*

*FROM medicines m*

*LEFT JOIN suppliers s ON m.supplier\_id = s.supplier\_id*

*ORDER BY m.quantity\_in\_stock ASC*

- Рецепты:

*SELECT p.prescription\_id,*

*c.first\_name || ' ' || c.last\_name as customer,*

*m.name as medicine,*

*p.doctor\_name, p.issue\_date, p.expiration\_date*

*FROM prescriptions p*

*LEFT JOIN customers c ON p.customer\_id = c.customer\_id*

*LEFT JOIN medicines m ON p.medicine\_id = m.medicine\_id*

- Закупки:

*SELECT pu.purchase\_id, pu.purchase\_date,*

*s.name as supplier,*

*pu.total\_amount*

*FROM purchases pu*

*LEFT JOIN suppliers s ON pu.supplier\_id = s.supplier\_id*

- Продажи:

*SELECT s.sale\_id, s.sale\_date,*

*c.first\_name || ' ' || c.last\_name as customer,*

*e.first\_name || ' ' || e.last\_name as employee,*

*s.total\_amount*

*FROM sales s*

*LEFT JOIN customers c ON s.customer\_id = c.customer\_id*

*LEFT JOIN employees e ON s.employee\_id = e.employee\_id*

*ORDER BY s.sale\_date DESC*

- Пункты закупок:

*SELECT pi.purchase\_item\_id,*

*pi.purchase\_id,*

*m.name as medicine,*

*pi.quantity, pi.price*

*FROM purchase\_items pi*

*LEFT JOIN medicines m ON pi.medicine\_id = m.medicine\_id*

- Пункты продаж:

*SELECT si.sale\_item\_id,*

*si.sale\_id,*

*m.name as medicine,*

*si.quantity, si.price*

*FROM sale\_items si*

*LEFT JOIN medicines m ON si.medicine\_id = m.medicine\_id*

**6. Запрос деталей продажи:**

*SELECT s.sale\_id, s.sale\_date, s.total\_amount,*

*c.first\_name || ' ' || c.last\_name as customer,*

*e.first\_name || ' ' || e.last\_name as employee*

*FROM sales s*

*LEFT JOIN customers c ON s.customer\_id = c.customer\_id*

*LEFT JOIN employees e ON s.employee\_id = e.employee\_id*

*WHERE s.sale\_id = %s*

Получает основную информацию о конкретной продаже.

**7. Запрос товаров в продаже:**

*SELECT m.name, m.medicine\_id, si.quantity, si.price,*

*(si.quantity \* si.price) as total*

*FROM sale\_items si*

*JOIN medicines m ON si.medicine\_id = m.medicine\_id*

*WHERE si.sale\_id = %s*

*ORDER BY m.name*

Получает список всех товаров в конкретной продаже.

**8. Запрос проверки наличия лекарств:**

*SELECT quantity\_in\_stock FROM medicines WHERE medicine\_id = %s*

Проверяет текущее количество определенного лекарства на складе.

**9. Запрос обновления запасов:**

*UPDATE medicines SET quantity\_in\_stock = %s WHERE medicine\_id = %s*

Обновляет количество определенного лекарства на складе.

**10. Запрос создания пользователя:**

*CREATE TABLE IF NOT EXISTS app\_users (*

*user\_id SERIAL PRIMARY KEY,*

*username VARCHAR(255) UNIQUE NOT NULL,*

*password\_hash VARCHAR(255) NOT NULL,*

*role VARCHAR(50) NOT NULL*

*)*

Создает таблицу пользователей, если она не существует.

**11. Запрос регистрации нового пользователя:**

*INSERT INTO app\_users (username, password\_hash, role)*

*VALUES (%s, %s, %s)*

Добавляет нового пользователя в систему.

**12. Запрос входа пользователя:**

*SELECT user\_id, password\_hash, role FROM app\_users WHERE username = %s*

Проверяет учетные данные пользователя при входе в систему.

**13. Запрос удаления записи:**

*DELETE FROM {table\_name} WHERE {primary\_key\_column} = %s*

Удаляет запись из указанной таблицы по первичному ключу.

**14. Запрос добавления новой продажи:**

*INSERT INTO sales (sale\_date, customer\_id, employee\_id, total\_amount)*

*VALUES (CURRENT\_TIMESTAMP, %s, %s, %s)*

*RETURNING sale\_id*

Создает новую запись о продаже и возвращает её ID.

**15. Запрос добавления товара в продажу:**

*INSERT INTO sale\_items (sale\_id, medicine\_id, quantity, price)*

*VALUES (%s, %s, %s, %s)*

Добавляет позицию товара в существующую продажу.