**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра «Математическая Кибернетика и Информационные технологии»

Лабораторная работа №6

Основы выборки SQL

Выполнил: Студент группы

БВТ2303

Морозова Ольга Андреевна

**Цель работы:**

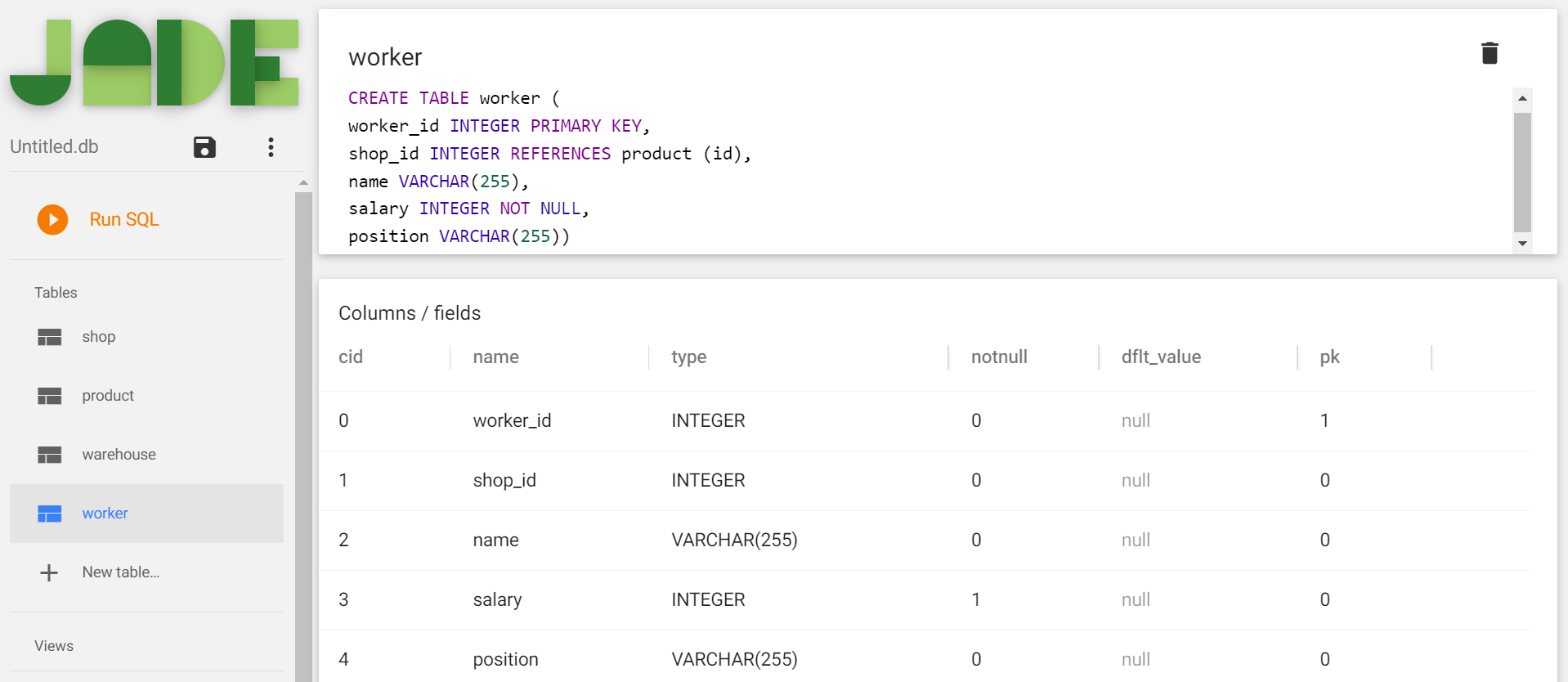
Изучение операторов сортировки, группировки и агрегатных функций в SQL.

**Ход работы:**

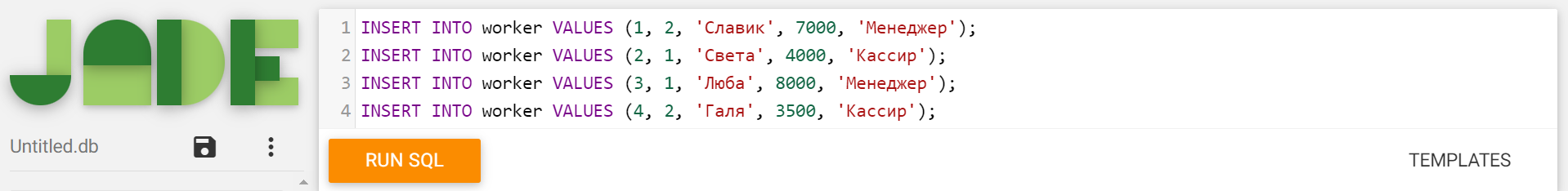
Продолжаем работать на сайте <https://sunnygoyal.com/jade/> .

Используем схему БД из 5 лабораторной работы (про магазины).

1. Создать дополнительную таблицу “Сотрудник” с помощью кода из методички.



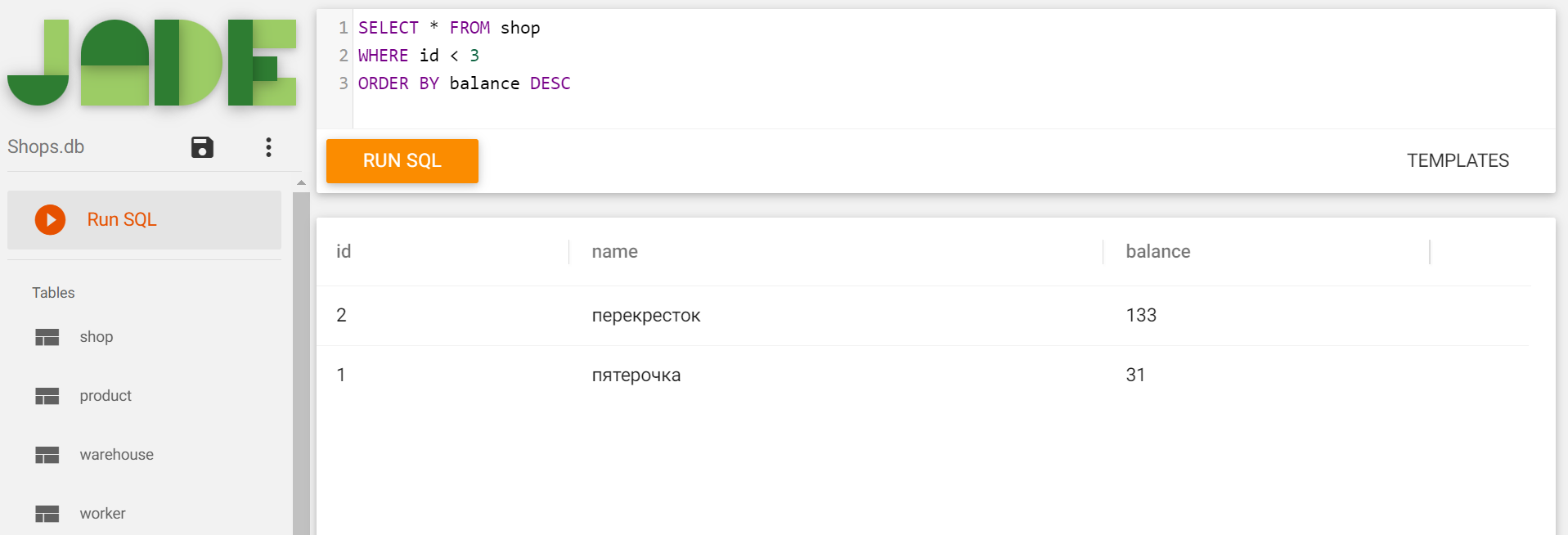
1. Внести в эту таблицу данные.



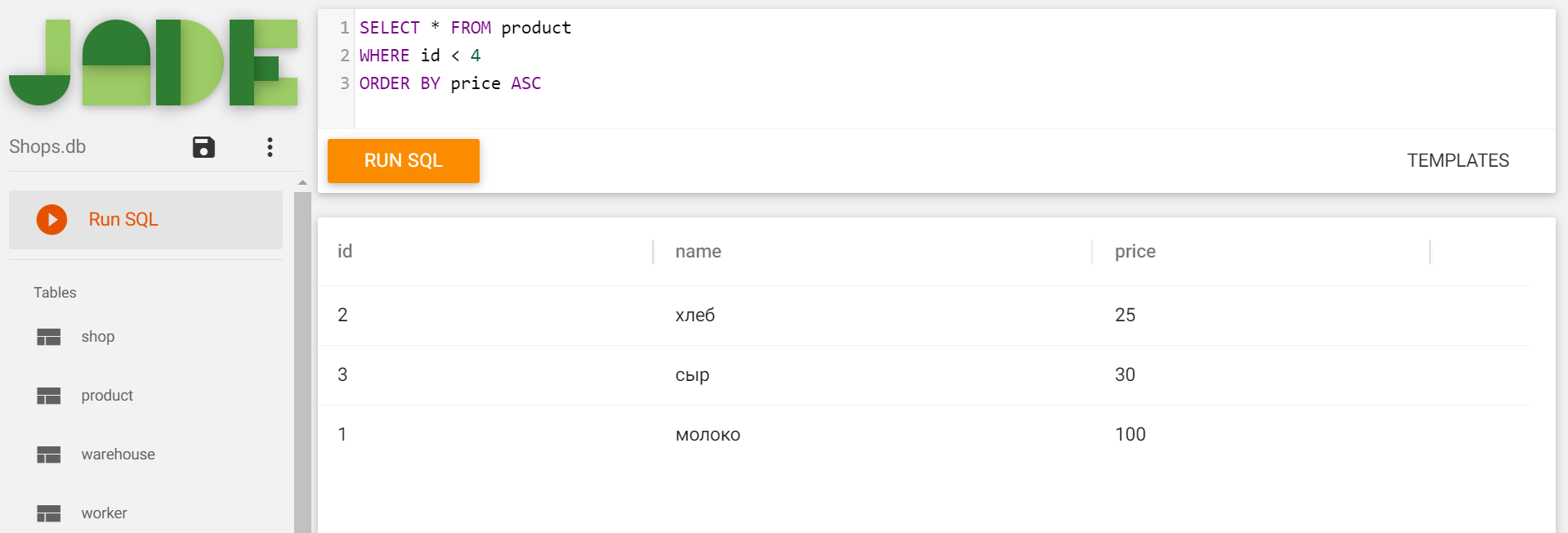
1. Запросы для таблиц.
   1. Запросы с сортировкой.

Используя ORDER BY, мы сортируем строки по указанному столбцу (столбцам). ASC означает сортировку по возрастанию (по умолчанию), а DESC – сортировку по убыванию.

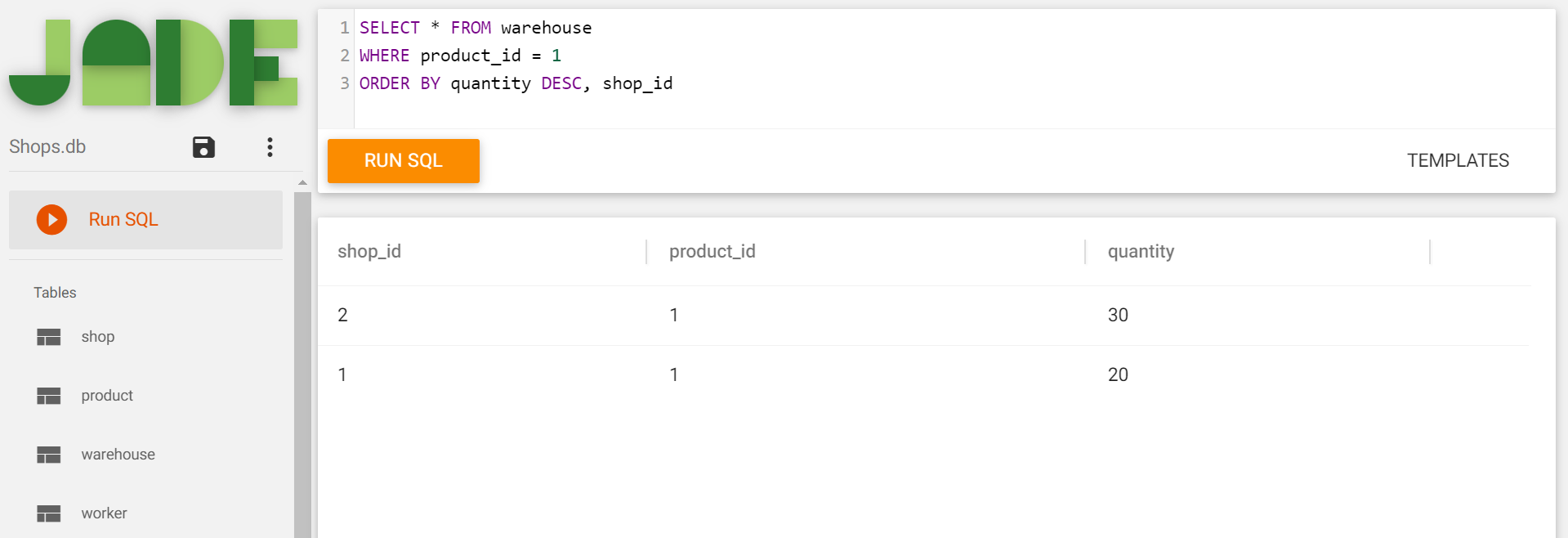
Для таблицы Shop:



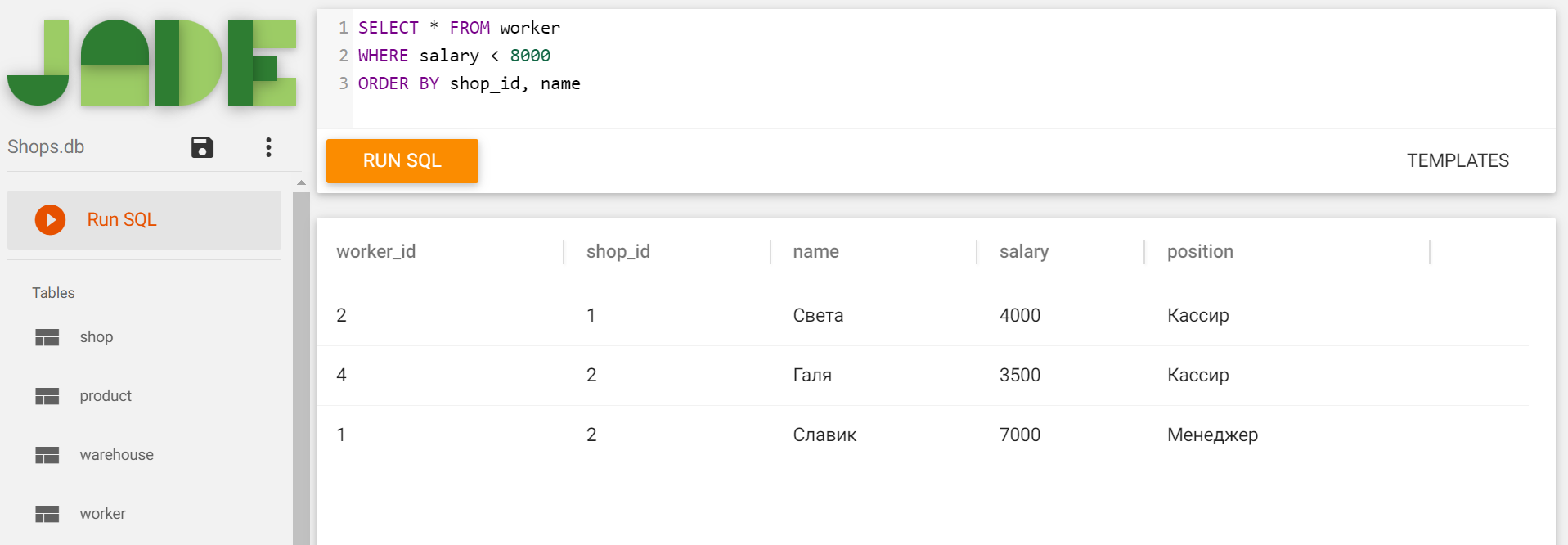
Для таблицы Product:



Для таблицы Warehouse (следует отметить, что здесь сортировка производится по столбцу quantity по убыванию, в то время как сортировка по столбцу shop\_id по возрастанию (т.к. не указано обратное, это происходит по умолчанию) не происходит, т.к. они противоречат друг другу и в приоритете является сортировка по первому из указанных столбцов):



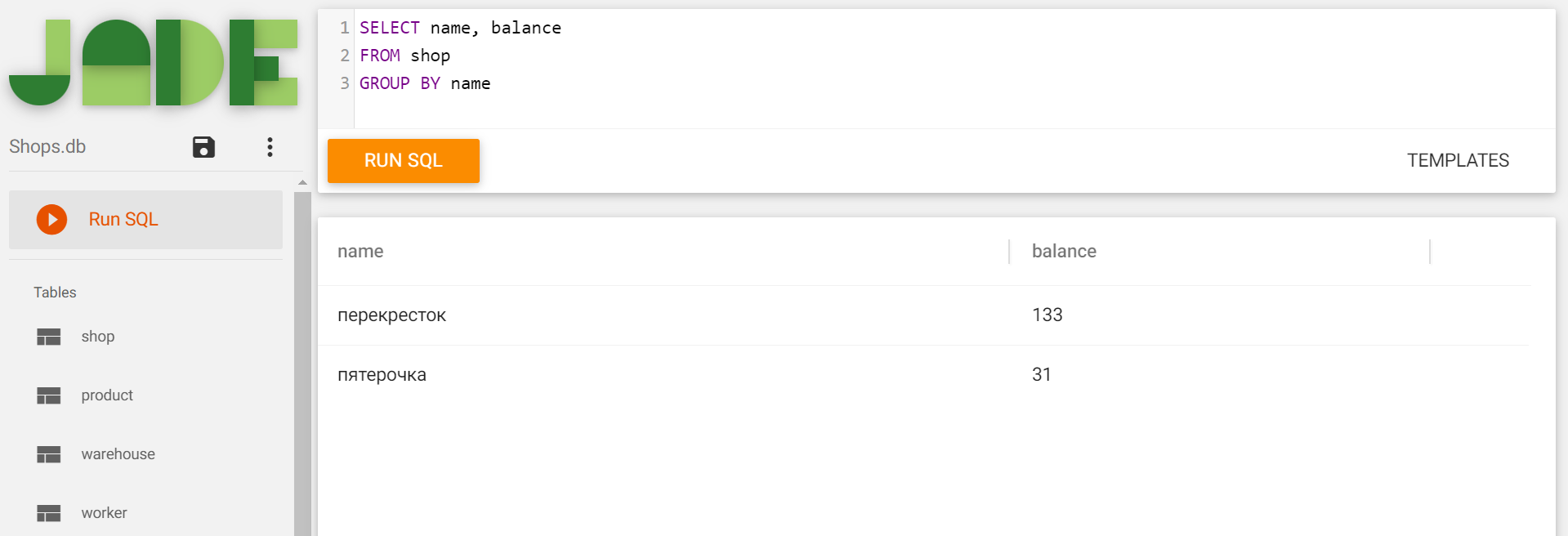
Для таблицы Worker (в данном случае сортировка производится по обоим указанным стобцам (по возрастанию, т.к. это стоит по умолчанию): сперва по shop\_id Света оказалась на первом месте, а после по name Галя оказывается выше Славика, хотя изначально он стоял выше неё, что видно по id):



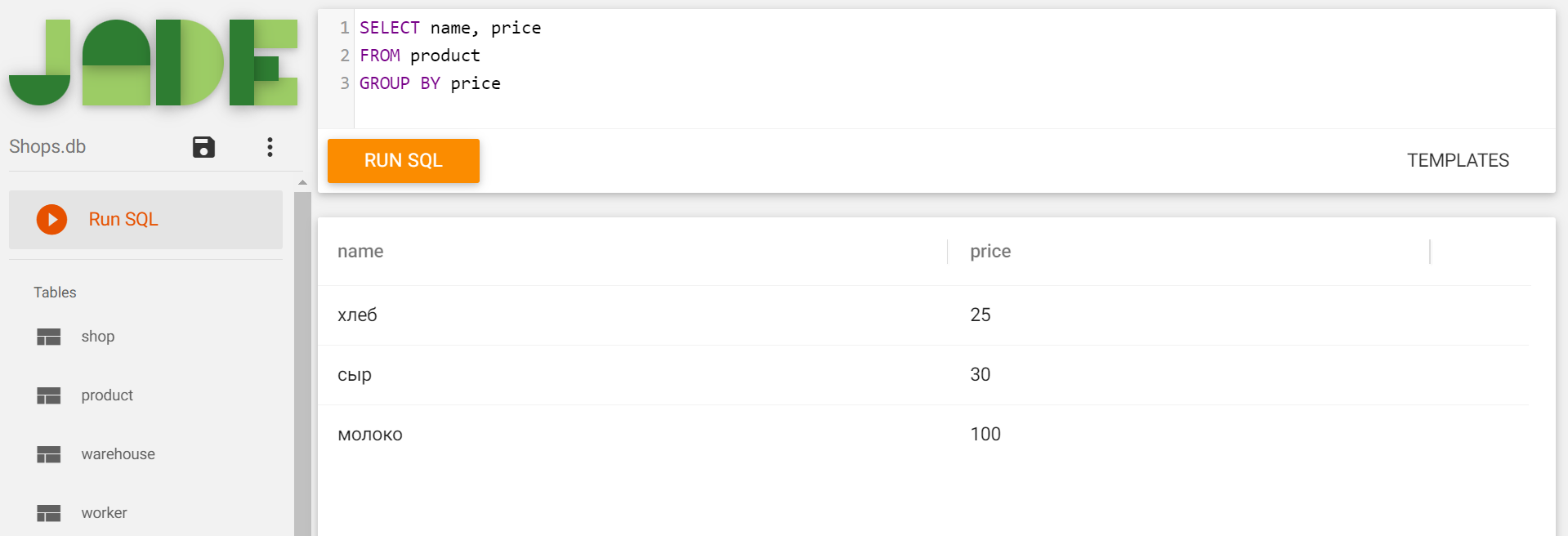
* 1. Запросы с группировкой.

Используя GROUP BY, мы группируем строки из таблицы в более крупные наборы результатов на основании одного или нескольких столбцов.

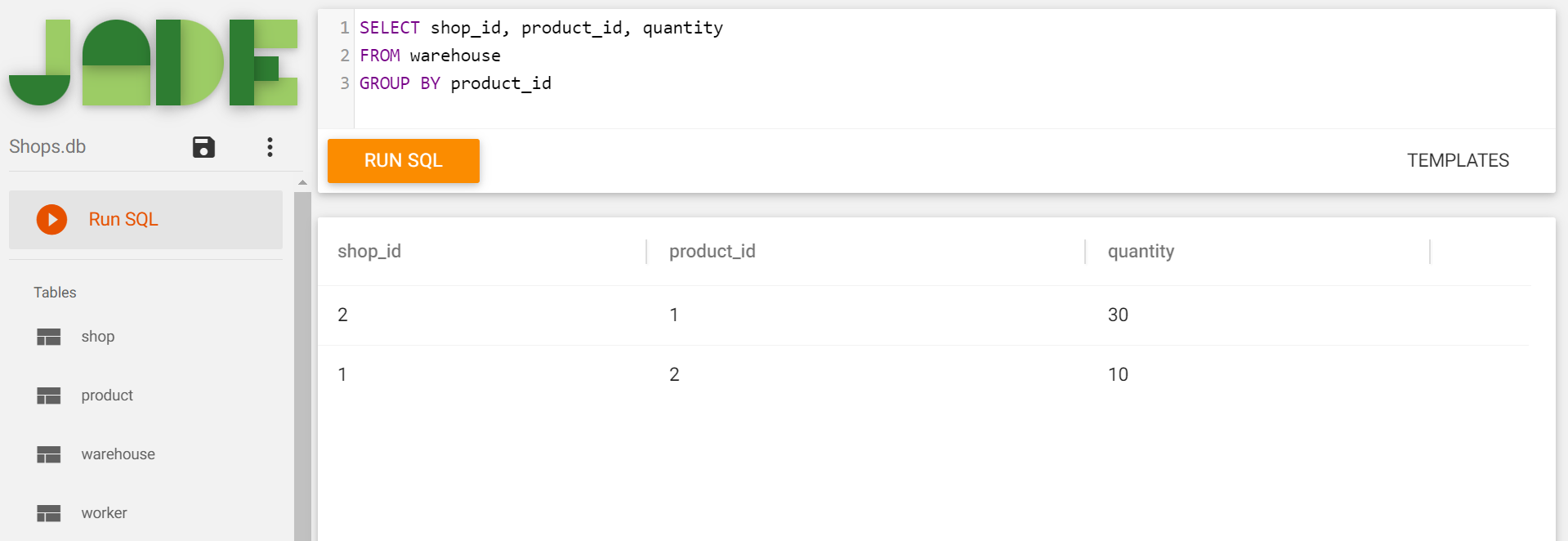
Для таблицы Shop:



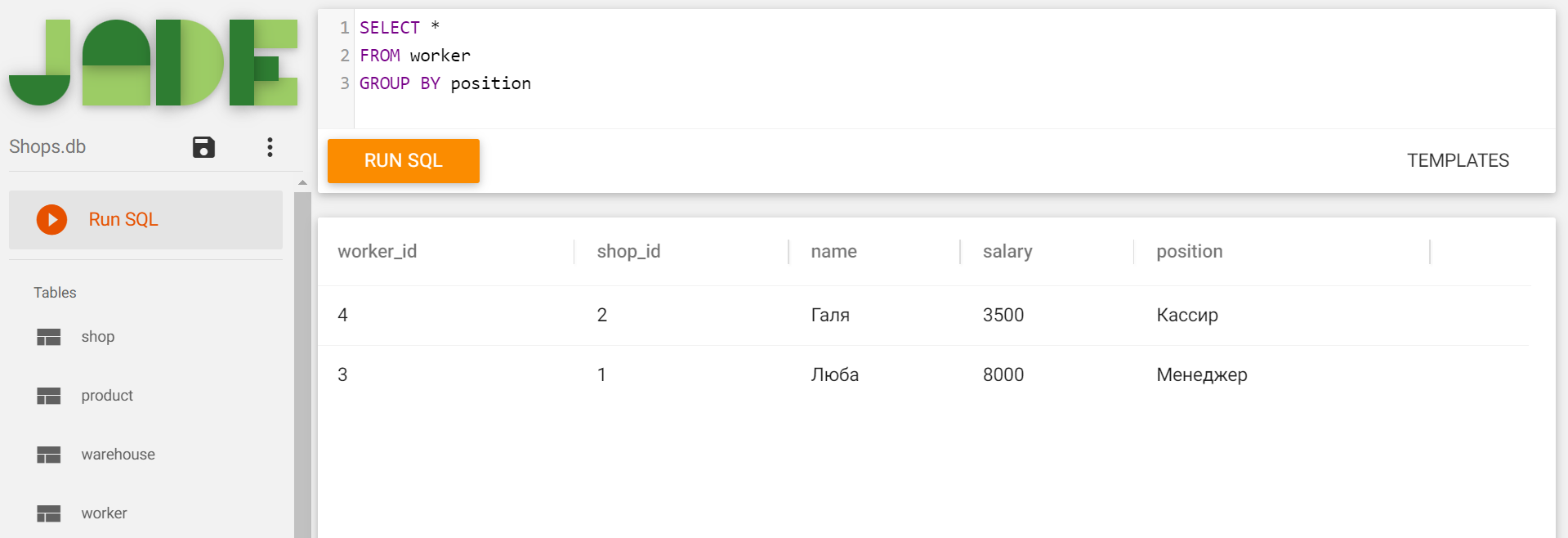
Для таблицы Product:



Для таблицы Warehouse (как мы видим, в следствие использования оператора группировки одна строка исчезла, т.к. тоже имела в product\_id, по которому мы группируем, значение 1. При этом примечательно то, что исчезла 1 строка, т.к. в ней значения shop\_id и quantity были меньше (1 и 20)):



Для таблицы Worker (Здесь примечательно то, что при группировке по столбцу position на должности Кассира осталась Галя, т.к. у неё больше значения worker\_id и shop\_id, хотя значения name и salary у Светы больше (Света, 4000). Почти такая же ситуация у Любы, у неё worker\_id больше, чем у Славика):

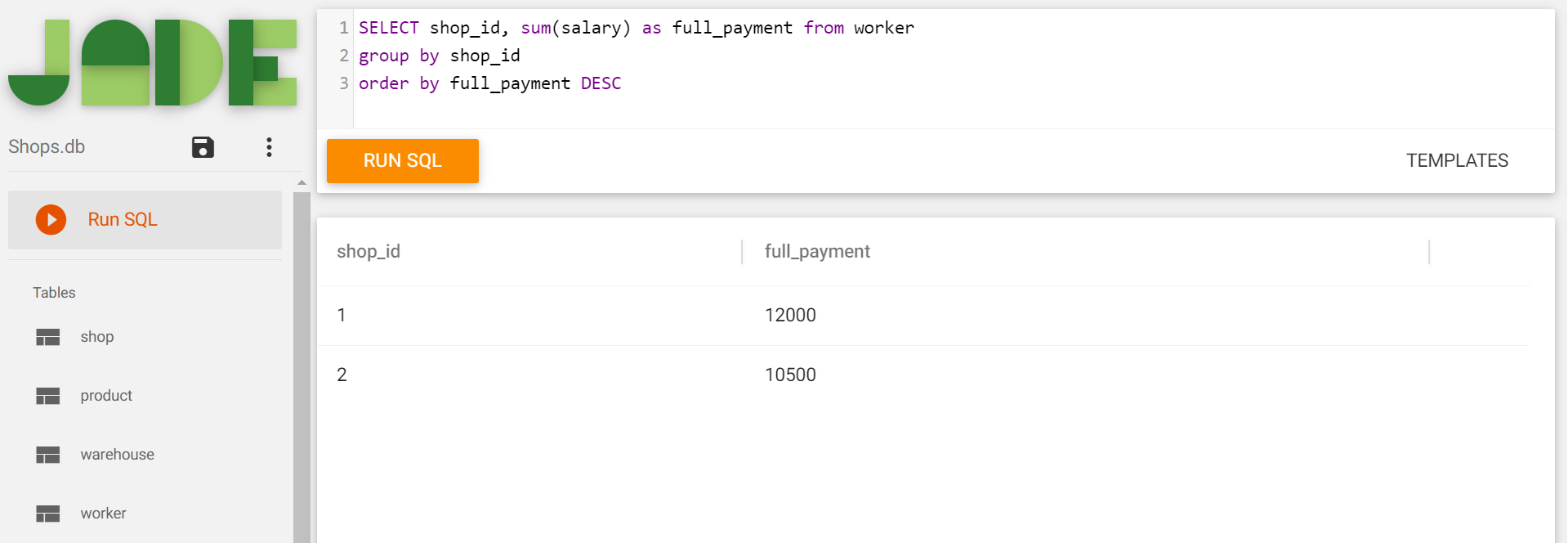


* 1. Запросы с использованием агрегатных функций.

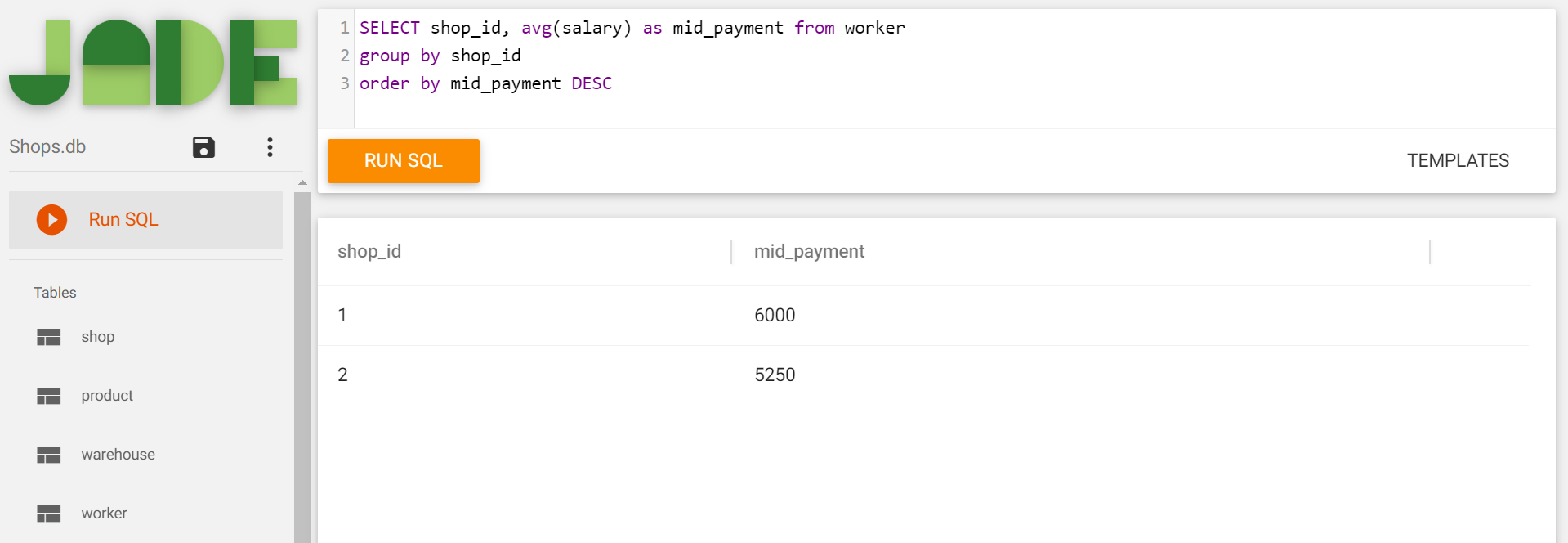
Агрегатная функция – это функция, которая выполняет вычисление на

наборе значений и возвращает одиночное значение.

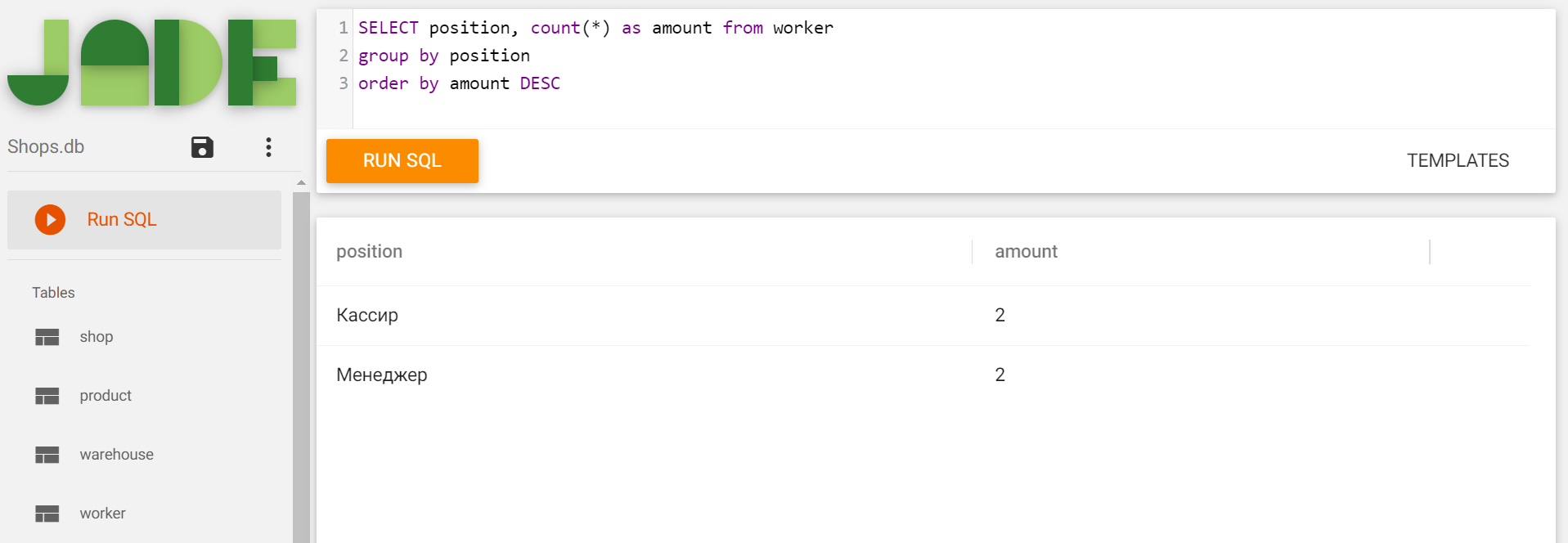
Здесь с помощью агрегатной функции суммы SUM я посчитала общие выплаты сотрудникам каждого магазина, сгруппировала по shop\_id и отсортировала в порядке убывания по full\_payment.



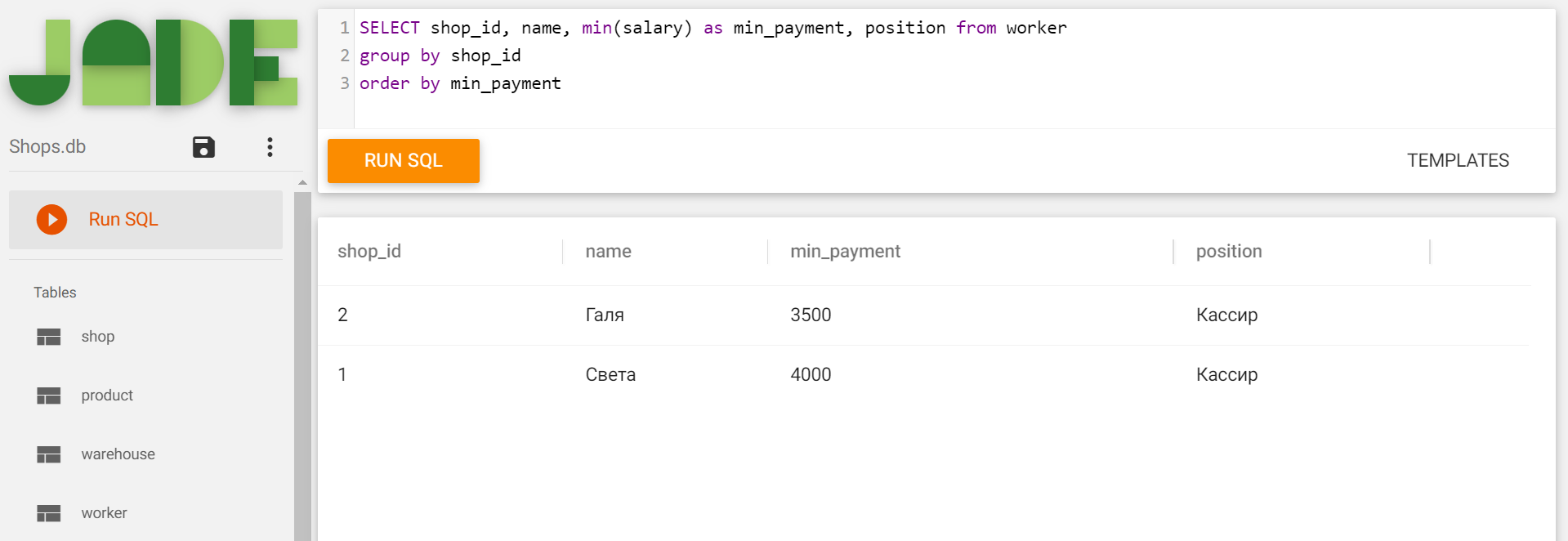
Здесь с помощью агрегатной функции среднего арифметического AVG я посчитала средние выплаты сотрудникам каждого магазина, сгруппировала по shop\_id и отсортировала в порядке убывания по mid\_payment.



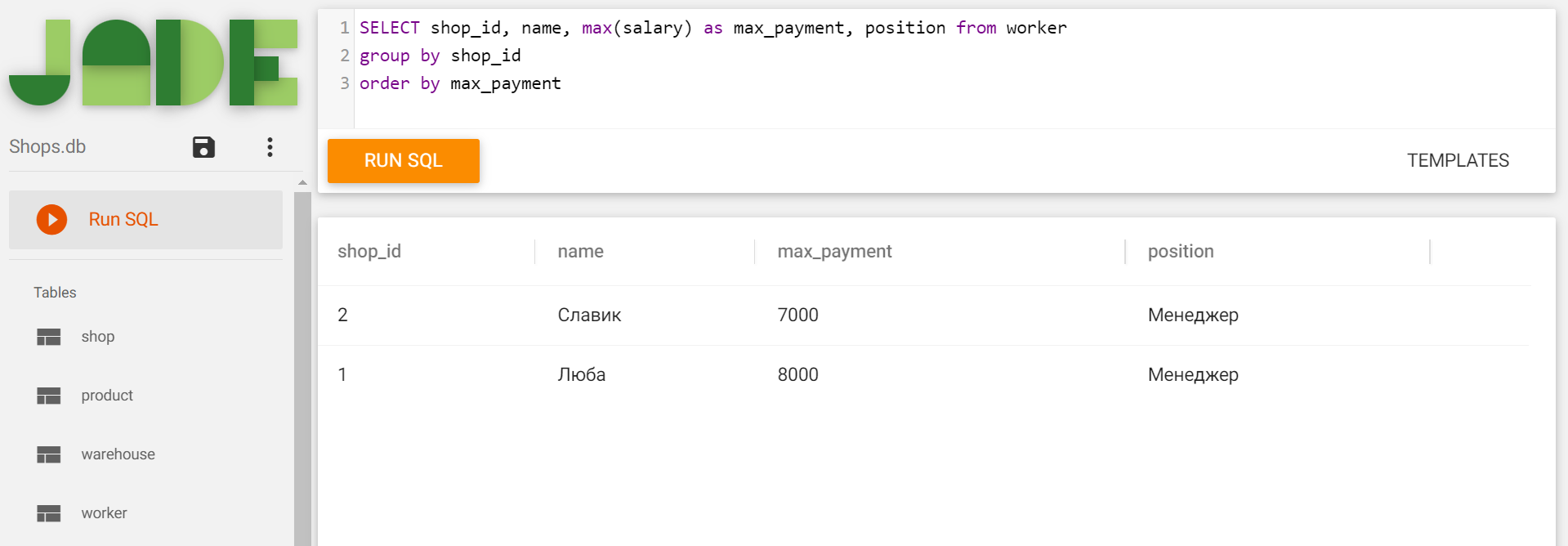
Здесь с помощью агрегатной функции подсчёта количества записей COUNT я посчитала количество сотрудников на каждой должности совместно в обоих магазинах, сгруппировала по position и отсортировала в порядке убывания по amount.



Здесь с помощью агрегатной функции получения минимума MIN я вывела сотрудника с минимальной зарплатой в каждом магазине, сгруппировала по shop\_id и отсортировала в порядке возрастания по min\_payment.



Здесь с помощью агрегатной функции получения максимума MAX я вывела сотрудника с максимальной зарплатой в каждом магазине, сгруппировала по shop\_id и отсортировала в порядке возрастания по max\_payment.



**Вывод:**

Мы научились использовать агрегатные функции, а также операторы сортировки и группировки при создании запросов в SQL.