**Домашняя работа 4**

Углубленный уровень SQL

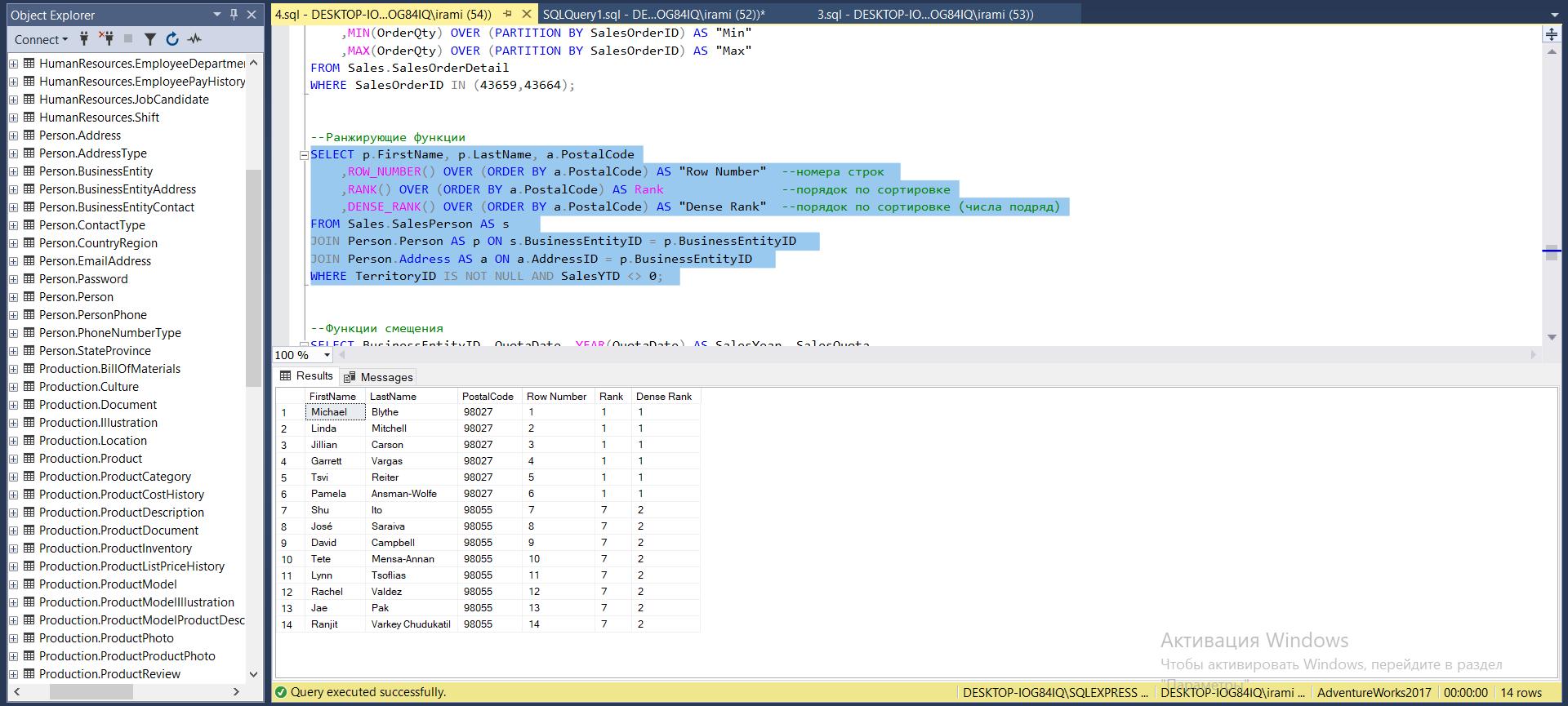
1. Посмотрите, какие ещё оконные функции cуществуют:

* [Ранжирующие функции](https://docs.microsoft.com/ru-ru/sql/t-sql/functions/ranking-functions-transact-sql?view=sql-server-ver15)
* [Агрегатные функции](https://docs.microsoft.com/ru-ru/sql/t-sql/functions/aggregate-functions-transact-sql?view=sql-server-ver15)
* [Аналитические функции](https://docs.microsoft.com/ru-ru/sql/t-sql/functions/analytic-functions-transact-sql?view=sql-server-ver15)

При каких значениях оконные функции Row Number, Rank и Dense Rank вернут одинаковый результат?

Когда значения данных, которые сравниваем или по которым сортируем, не равны друг другу.

Например: в данной таблице сортировка происходит по значению PostalCode. Если б значение PostalCode ранжировалось от 98027 до 98040 (допустим разница в 1), то значение Row Number, Rank и Dense Rank были равны друг другу.



1. Решите на базе данных AdventureWorks2017 следующие задачи.
2. Изучите данные в таблице Production.UnitMeasure. Проверьте, есть ли здесь UnitMeasureCode, начинающиеся на букву ‘Т’. Сколько всего различных кодов здесь есть?

Отсутствует. Количество строк – 38

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Вставьте следующий набор данных в таблицу:

* TT1, Test 1, 9 сентября 2020
* TT2, Test 2, getdate()

Проверьте теперь, есть ли здесь UnitMeasureCode, начинающиеся на букву ‘Т’.

Добавлены 2 строки, общее количество строк – 40

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

1. Теперь загрузите вставленный набор в новую, не существующую таблицу Production.UnitMeasureTest. Догрузите сюда информацию из Production.UnitMeasure по UnitMeasureCode = ‘CAN’. Посмотрите результат в отсортированном виде по коду.

Изображение выглядит как текст

Автоматически созданное описание

1. Измените UnitMeasureCode для всего набора из Production.UnitMeasureTest на ‘TTT’.

Изображение выглядит как текст

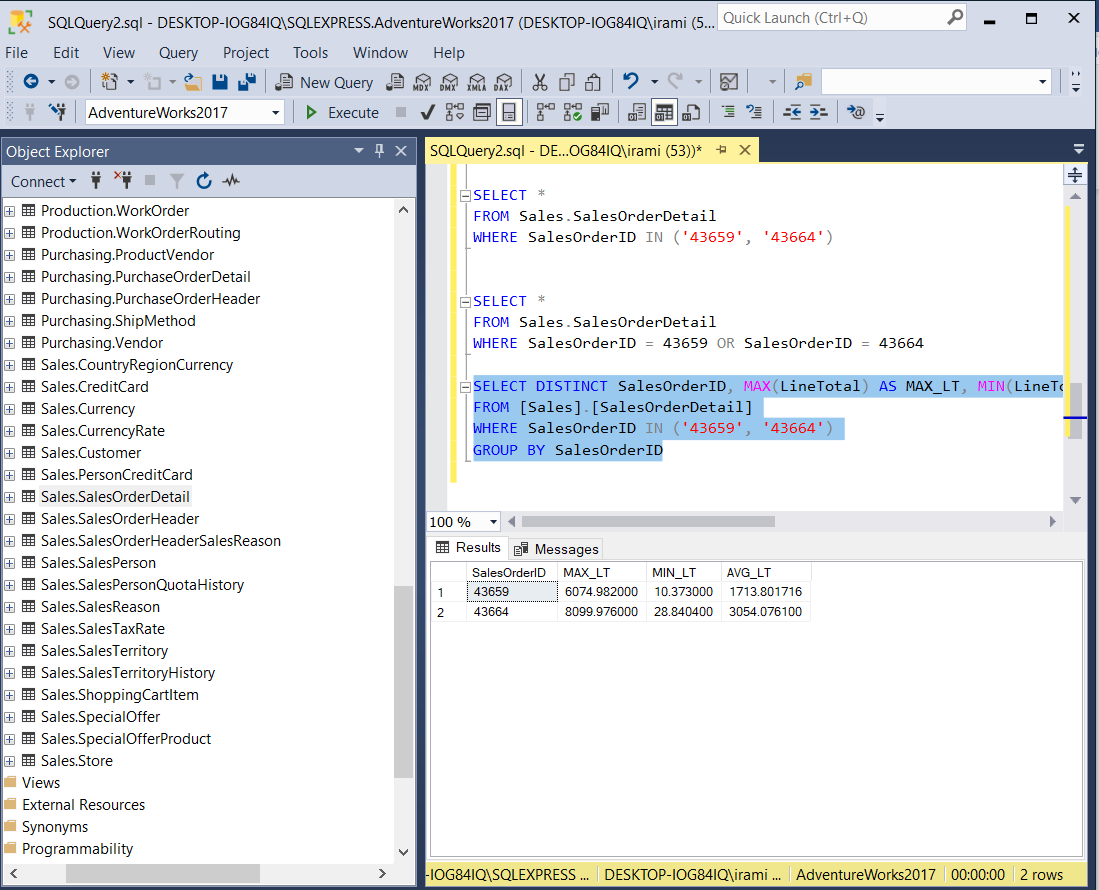
Автоматически созданное описание

1. Удалите все строки из Production.UnitMeasureTest.

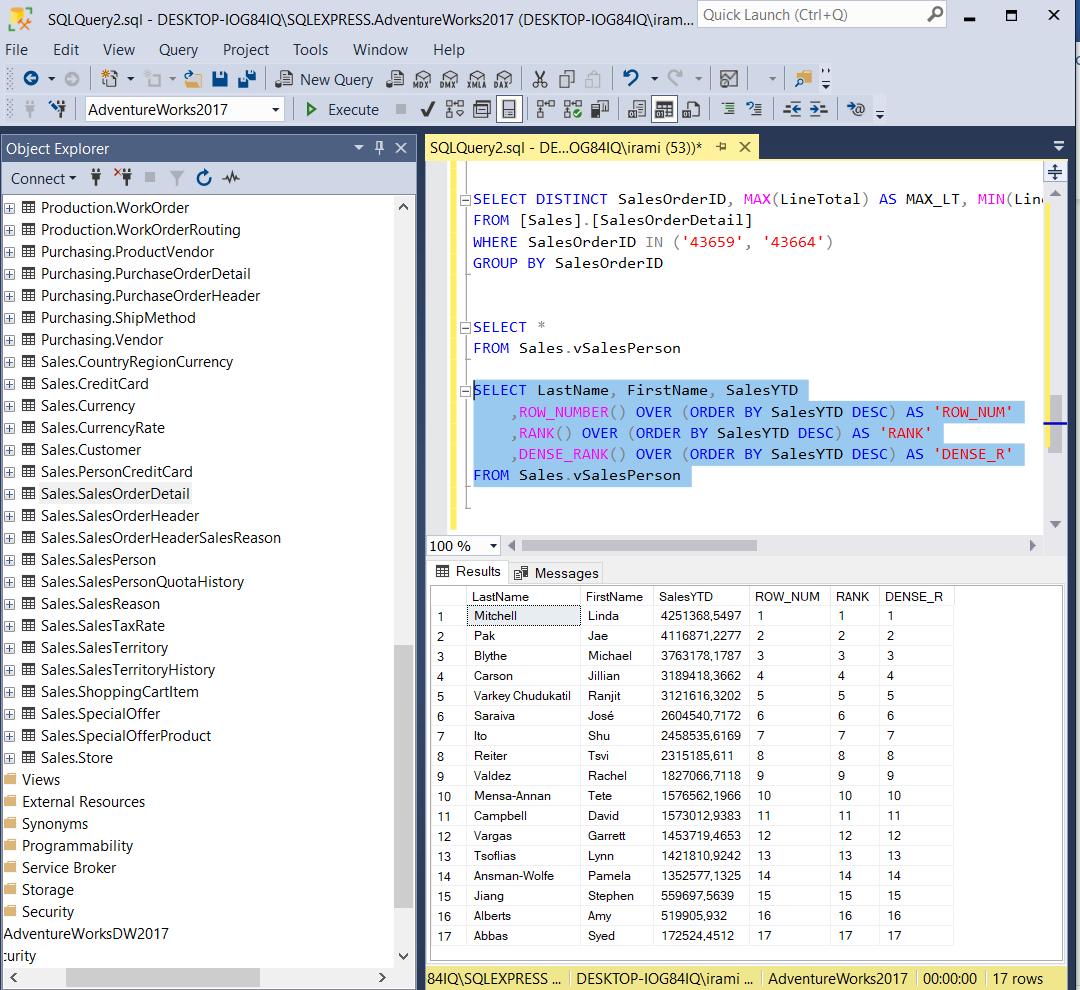
Изображение выглядит как текст

Автоматически созданное описание

1. Найдите информацию из Sales.SalesOrderDetail по заказам 43659,43664. С помощью оконных функций MAX, MIN, AVG найдем агрегаты по LineTotal для каждого SalesOrderID.



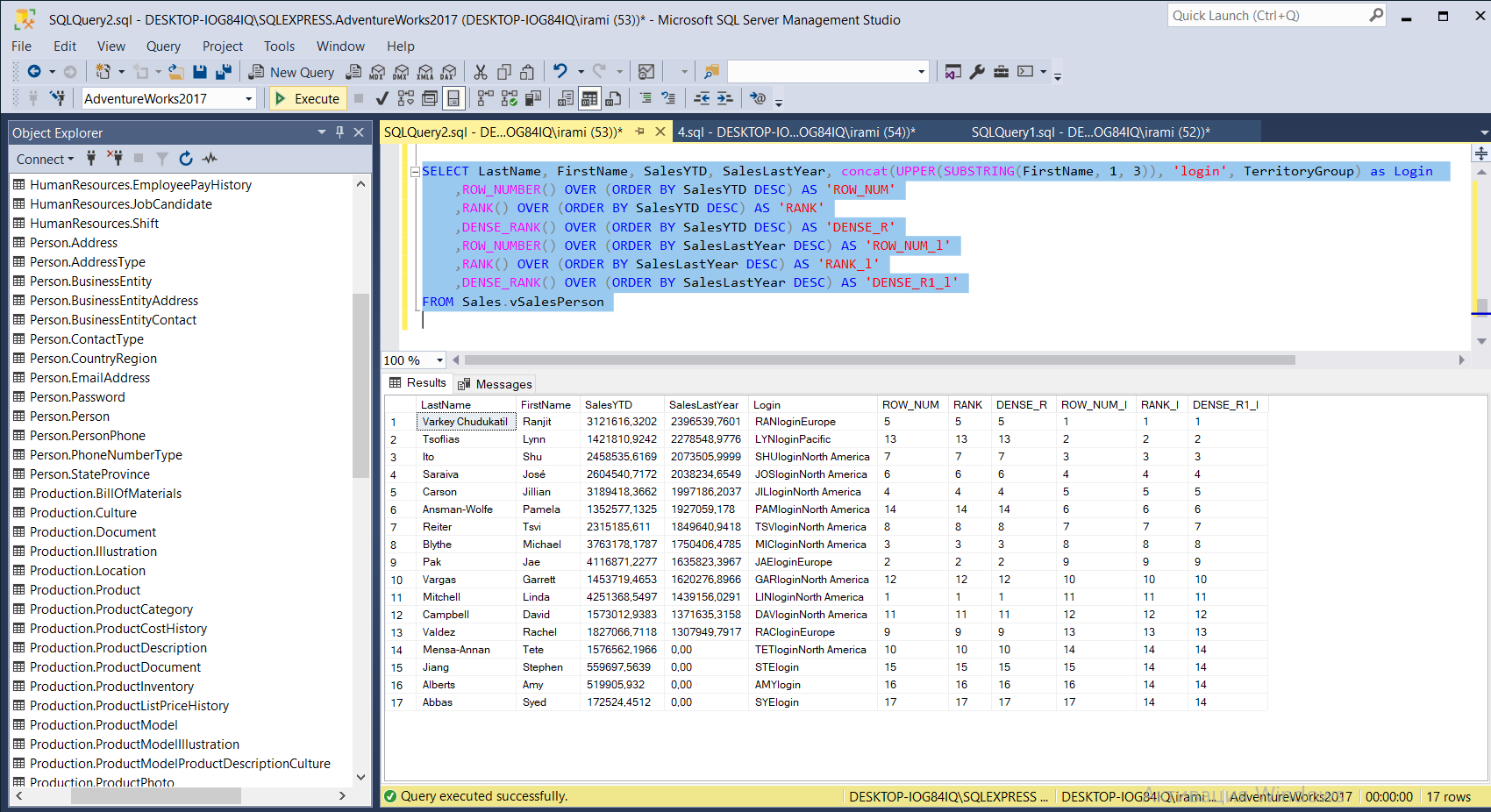
1. Изучите данные в объекте Sales.vSalesPerson. Создайте рейтинг cреди продавцов на основе годовых продаж SalesYTD, используя ранжирующую оконную функцию.



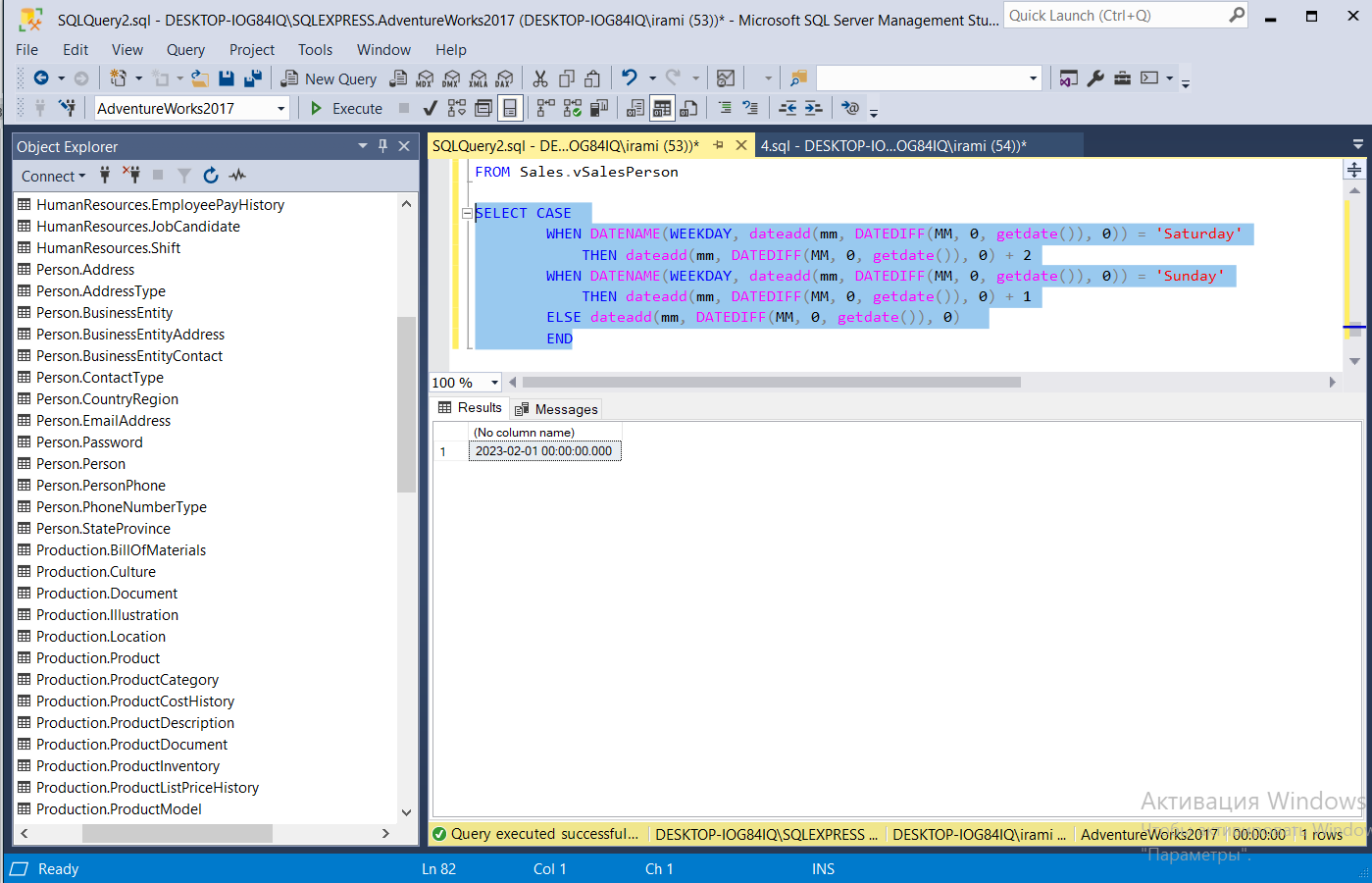
Добавьте поле Login, состоящий из 3 первых букв фамилии в верхнем регистре + ‘login’ + TerritoryGroup (Null заменить на пустое значение).

Кто возглавляет рейтинг? А кто возглавлял рейтинг в прошлом году (SalesLastYear).

В этом году рейтинг возглавляет Michael Linda, в прошлом – Varkey Chudukatil Ranjit.



1. Найдите первый будний день месяца (FROM не используем). Нужен стандартный код на все времена.



1. Давайте еще раз остановимся и отточим понимание функции count. Найдите значения count(1), count(name), count(id), count(\*) для следующей таблицы:

**Id**(PK) **Name DepName**

1 null A

2 null null

3 A C

4 B C

**Id**(PK) **Name DepName** count(1) count(name) count(id) count(\*)

1 null A 1 null 1 1

2 null null 1 null 1 1

3 A C 1 1 1 1

4 B C 1 1 1 1

Функция COUNT(\*) и COUNT(1) подсчитывает каждую строку отдельно. При этом учитываются и строки, содержащие значения NULL. Общее число сток для COUNT(\*) и COUNT(1) – по 4 в каждой.

Функция COUNT(NAME) рассчитывает количество значений (значения NULL не будут учитываться) указанного столбца. Общее число сток – 2.

Функция COUNT(Id) рассчитывает количество значений (значения NULL не будут учитываться) указанного столбца. Общее число сток – 4.