|  |  |  |
| --- | --- | --- |
|  | | |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования  «МИРЭА — Российский технологический университет»  РТУ МИРЭА | | |
|  | | |
| Институт информационных технологий (ИТ) | | |
| Кафедра инструментального и прикладного программного обеспечения (ИППО) | | |
| **ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4** | | |
| **по дисциплине**  «Хранилища данных и OLAP-технологии» | | |
|  | | |
| Выполнил студент группы ИКБО-02-15 | | Апальков П. Ю. |
| Принял | | Абаев Р. Г. |
| Работа выполнена | « » 2018 г. | *(подпись студента)* |
| Работа зачтена | « » 2018 г. | *(подпись руководителя)* |
| Москва 2018 | | |

Содержание

[Задание 3](#_Toc531708928)

[Выполнение работы 3](#_Toc531708929)

[Формирование MDX запросов 7](#_Toc531708931)

[Заключение 19](#_Toc531708932)

[Заключение 20](#_Toc531708933)

# Задание

В рамках данной лабораторной работы необходимо рассмотреть основы языка MDX (Multidimensional Expressions) и применить его к созданному хранилищу данных Northwind.

# Выполнение работы

## Часть 1

Для настройки источника данных для запросов устанавливаем соединение между созданным хранилищем данных и Pentaho Community Edition. В главном меню выбираем «Manage Data Source» - «Data Source Wizard». Указываем «Reporting and Analysis (Requires Star Schema)» при создании источника данных, выбираем базу данных northwind\_connection и нажимаем «Next» (Рис.1).

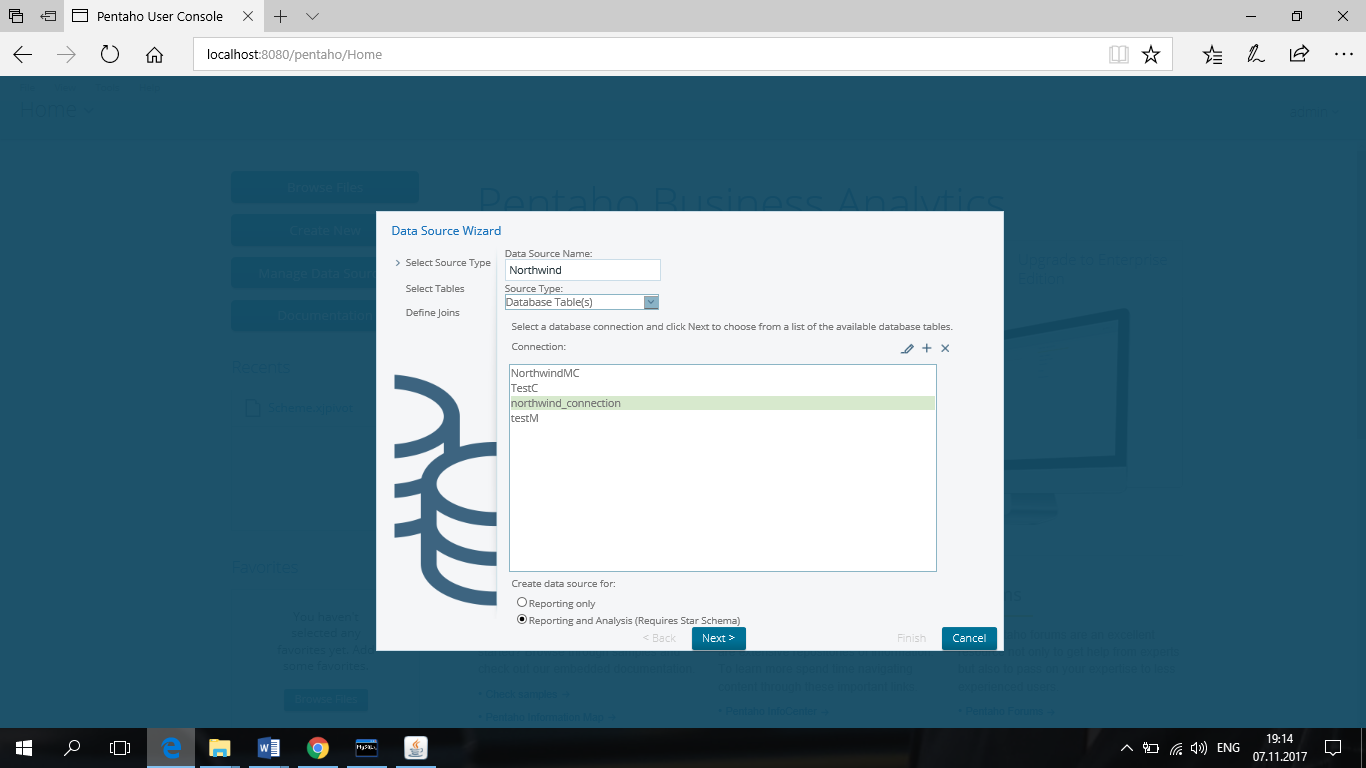


Рис.1. Выбор источника данных

Далее выбираются таблицы, которые будут участвовать при создании MDX запросов, доступ к данным которых будет разрешён. Выбираем все и переносим с помощью стрелок в правый список (Рис.2). В поле «Fact Table» указываем таблицу фактов (sales\_fact) (Рис. 3). Она будет определять в дальнейшем меры для формирования запроса.

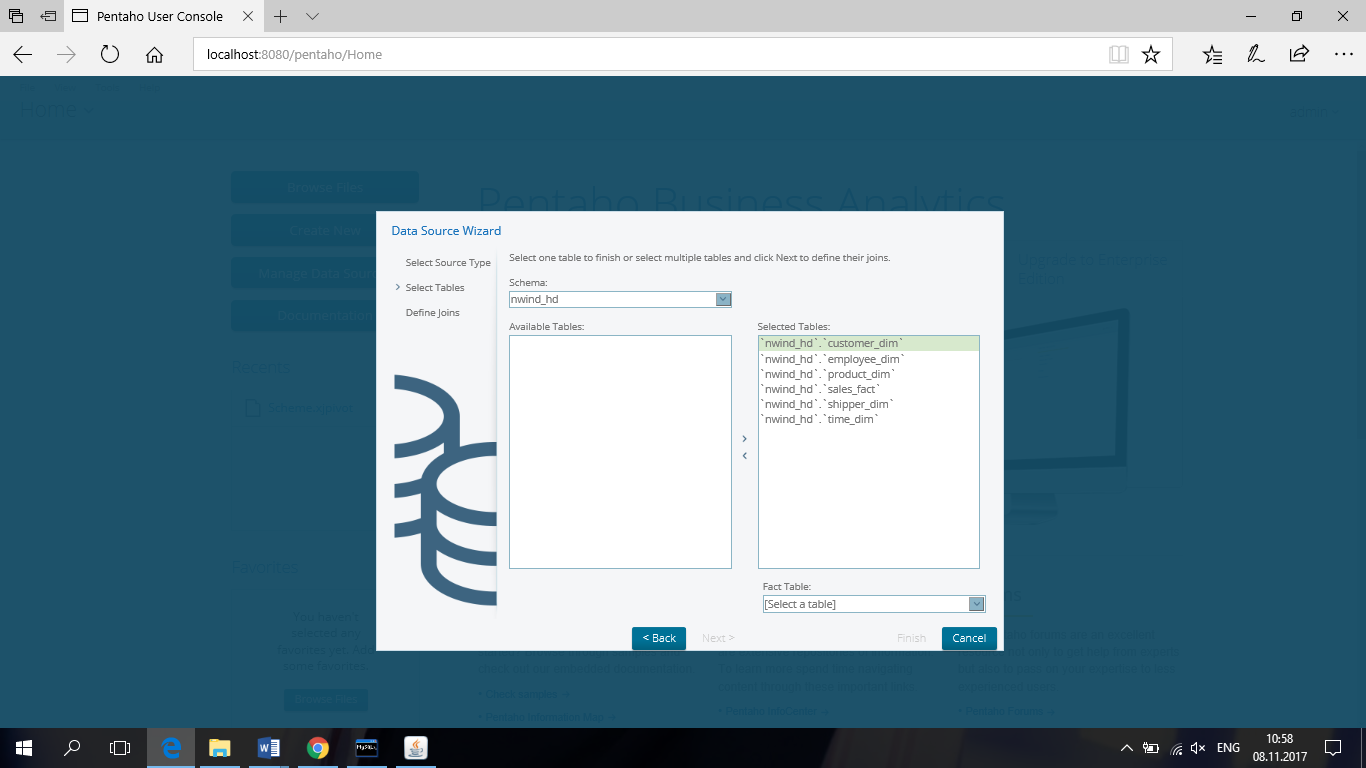


Рис.2. Выбор таблиц

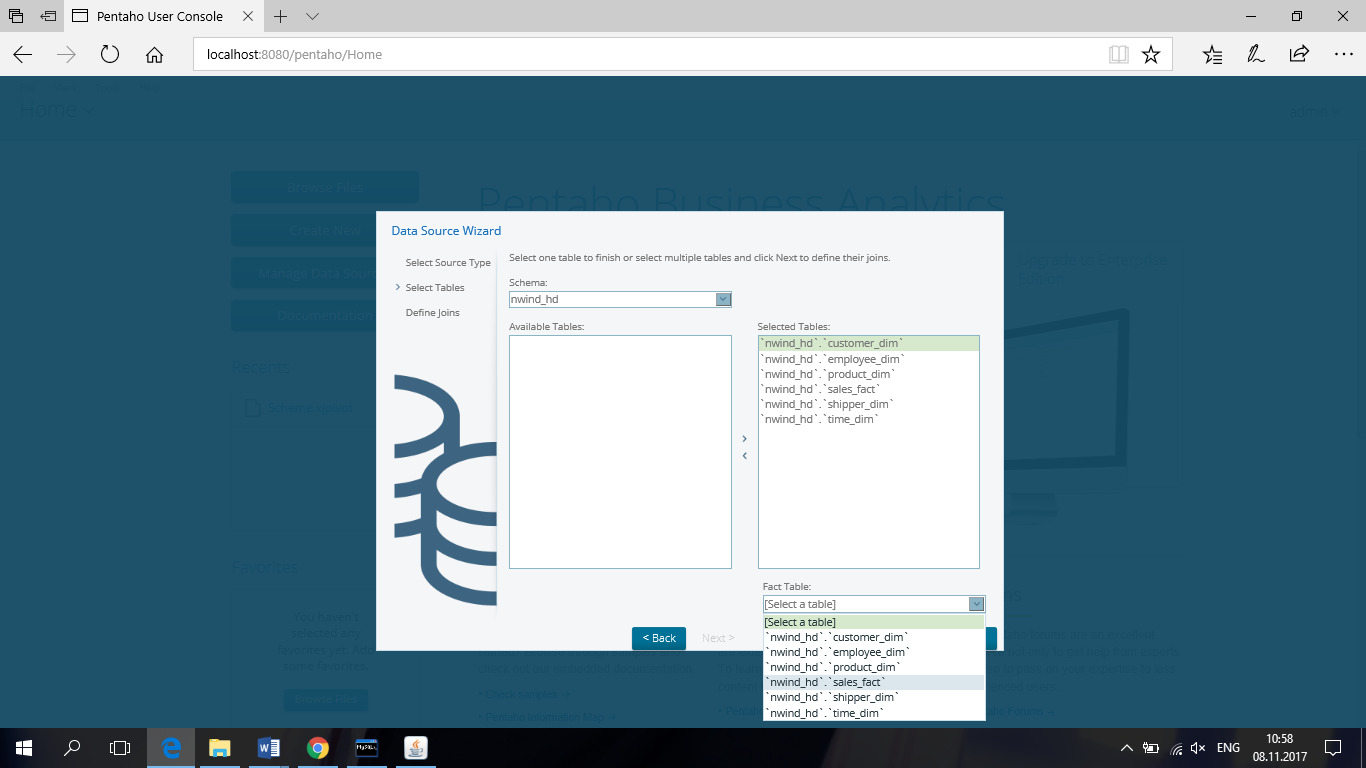


Рис.3. Указание таблицы фактов

Далее устанавливаем связи между таблицей фактов и всеми остальными. В левой таблице выбирается поле с внешним ключом, а справа соответствующая таблица, в которой хранится тоже ключ в качестве уникального. Нажимаем “create join”, и связь добавляется к списку имеющихся. (Рис. 4).

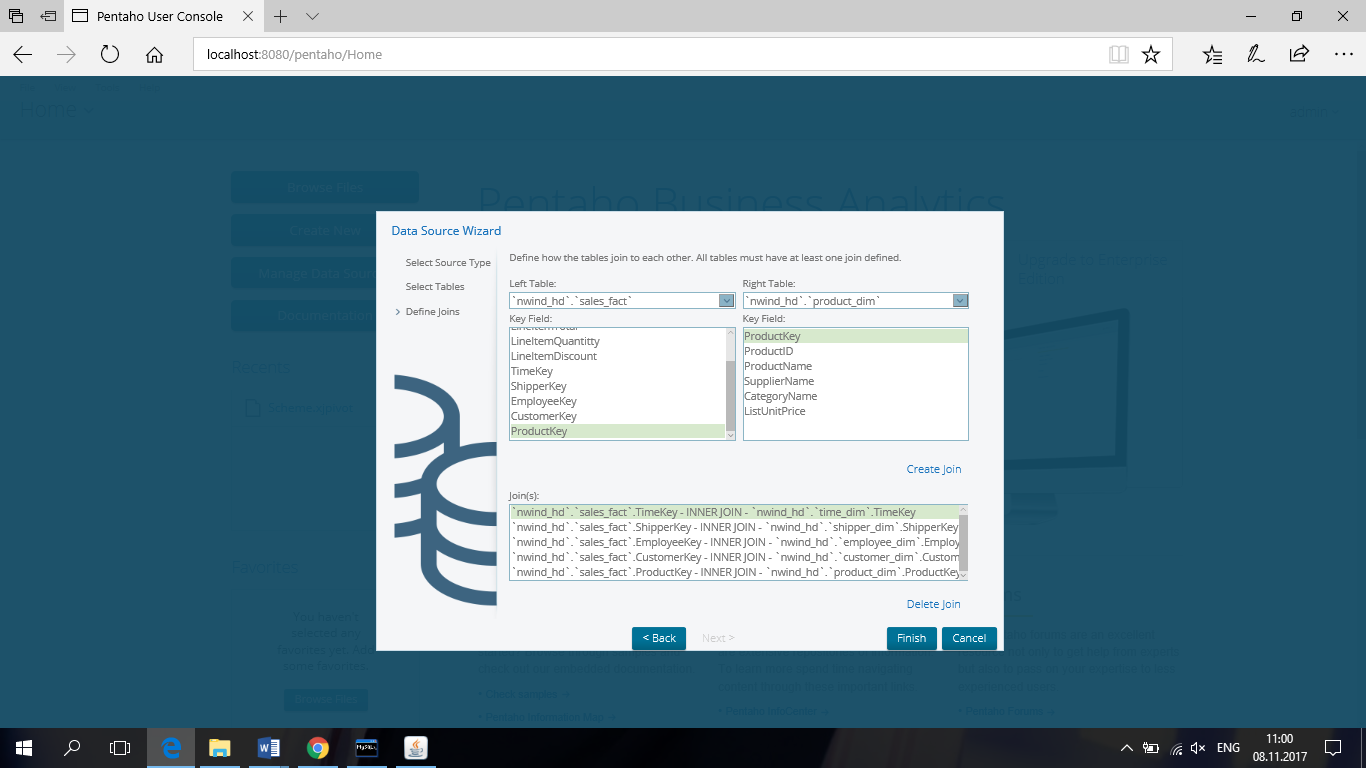


Рис.4. Установление связей между таблицами

Во всплывшем окне пользователю предлагается создать модель по умолчанию (“Keep default model”) или настроить её вручную (“Customize model new”). Выбираем второй пункт для пользовательской настройки модели.

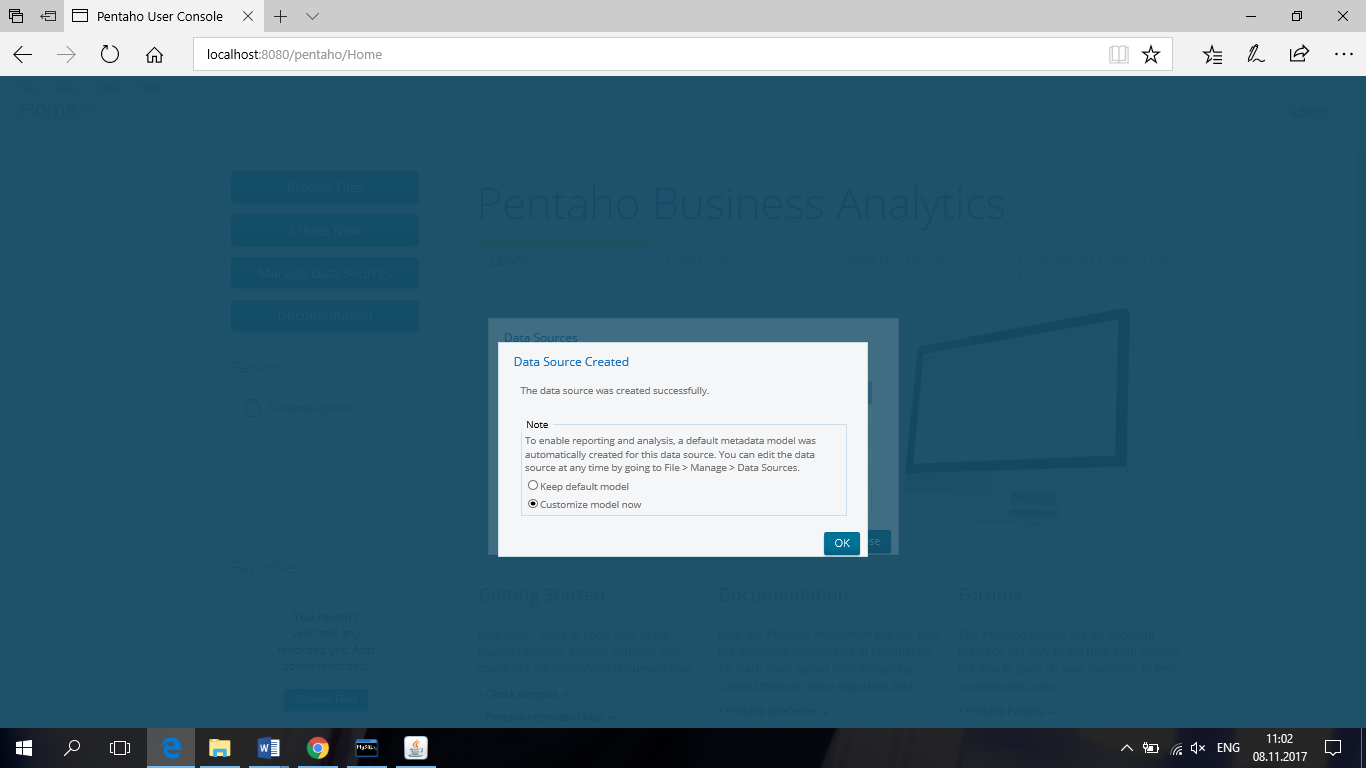


Рис. 5. Выбор модели хранилища данных

При создании мер необходимо создать те меры, которые в дальнейшем будут использоваться. Автоматически создаются меры для всех столбцов в таблице фактов (Рис.6), но нам понадобятся только две:

- LineItemQuantity – количество проданных товаров;

- LineItemTotal – стоимость проданных товаров.

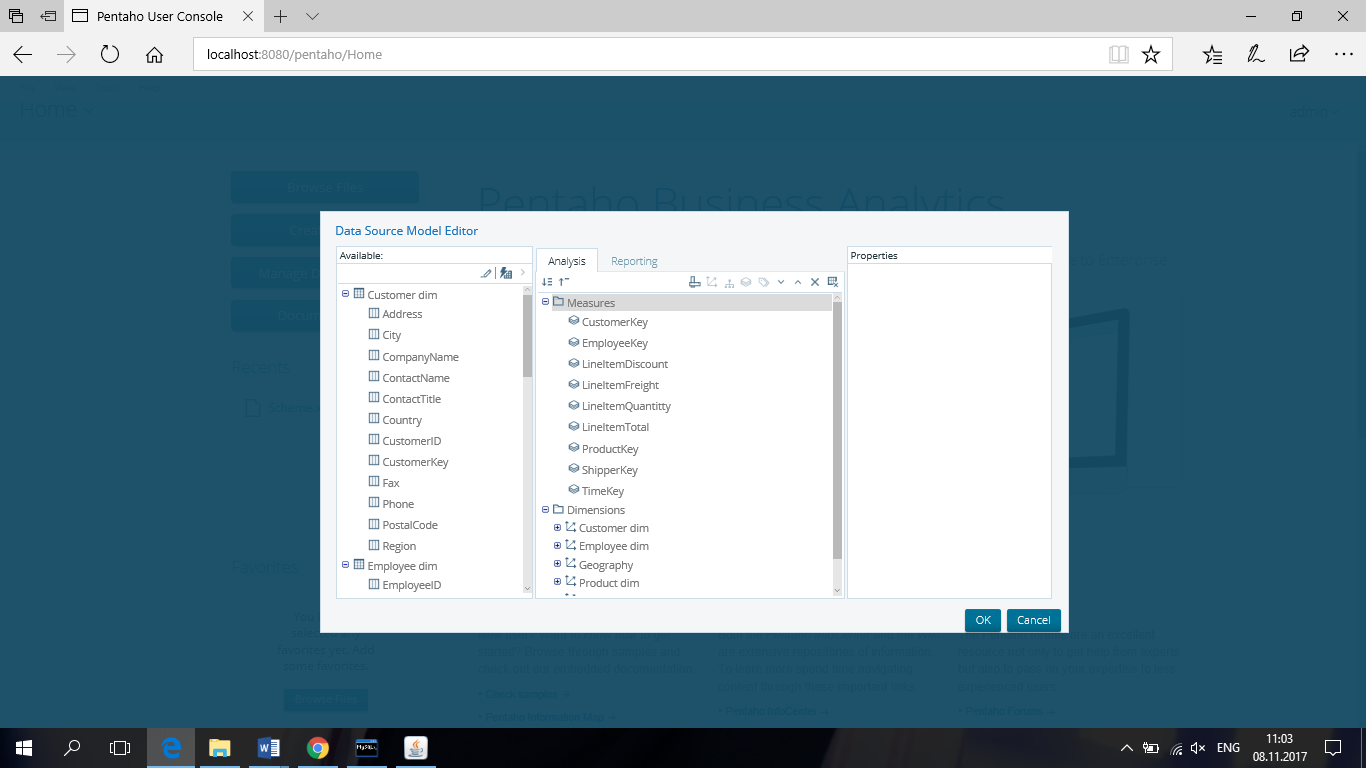


Рис.6. Автоматически созданные меры

Отредактируем список мер, оставив лишь необходимые и задав им поясняющие имена:

- UnitSales – единица товаров;

- CostSales – стоимость товаров.

Параметру агрегации установим значение суммирование (SUM) (Рис.7).

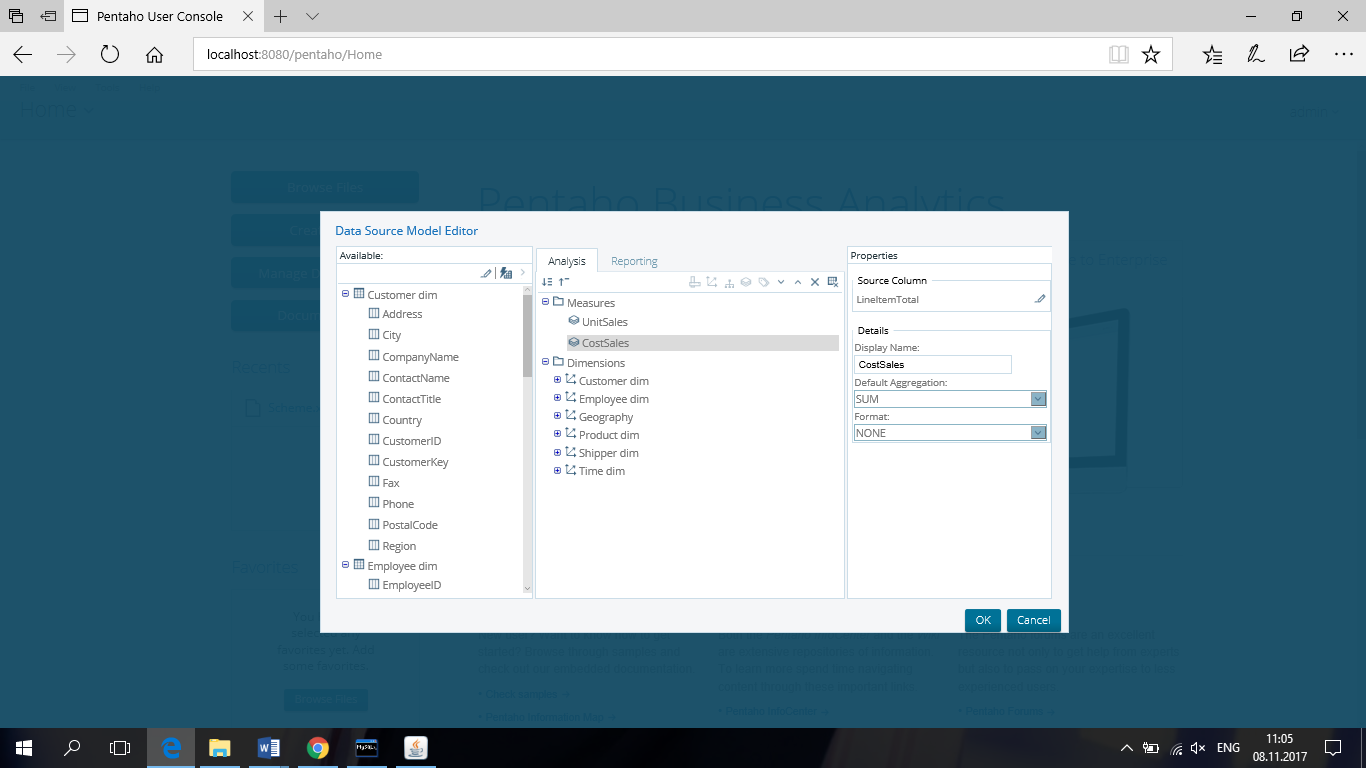


Рис.7. Редактирование мер

Далее отредактируем измерения. Из всего списка интервалов времени нас интересуют лишь год, квартал и месяц. Обязательно надо указать, что данное измерение является временем. Каждому полю (год, месяц, квартал) необходимо указать соответствующий уровень (Рис.8).

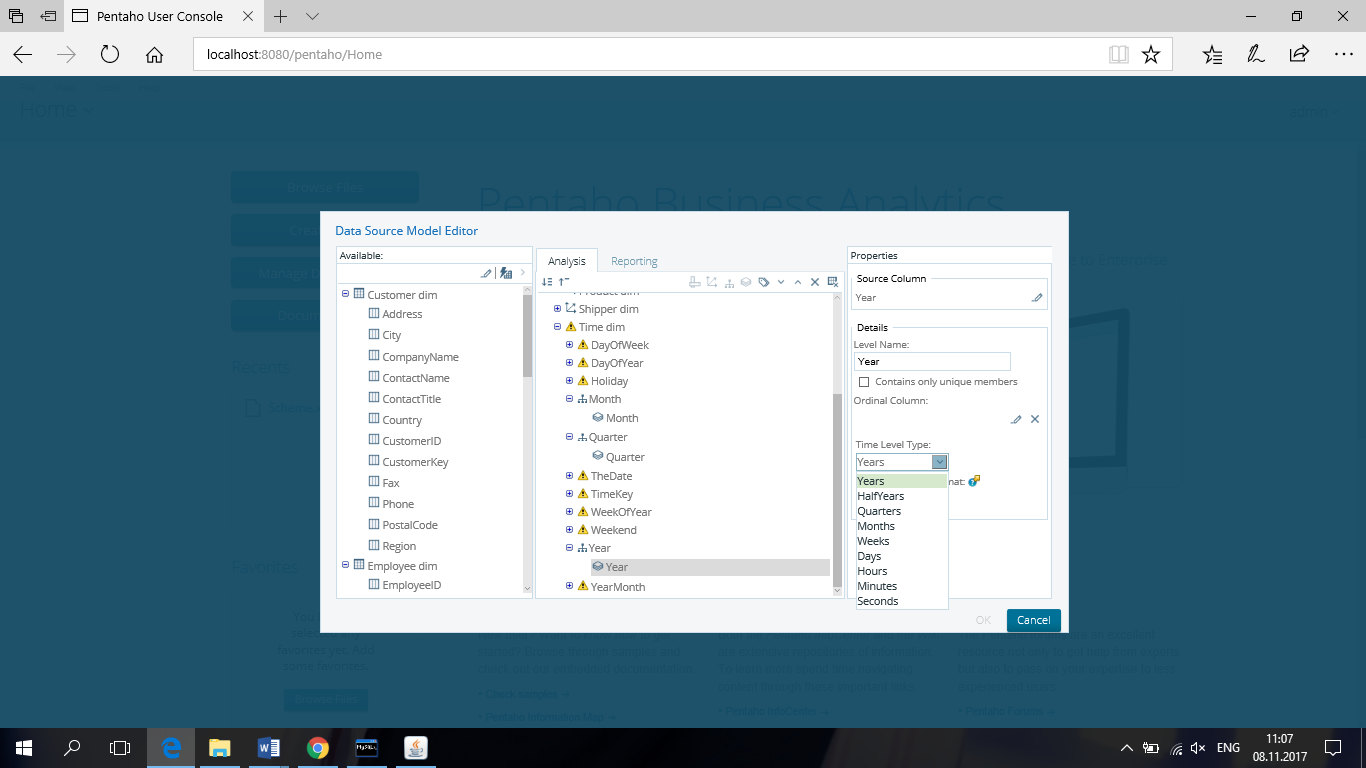


Рис.8. Редактирование измерения время и указание интервала времени

В измерении «geography» тип полей задаётся автоматически, но его стоит проверить (Рис.9). В остальных измерениях тип полей не определён и по умолчанию стоит NONE.

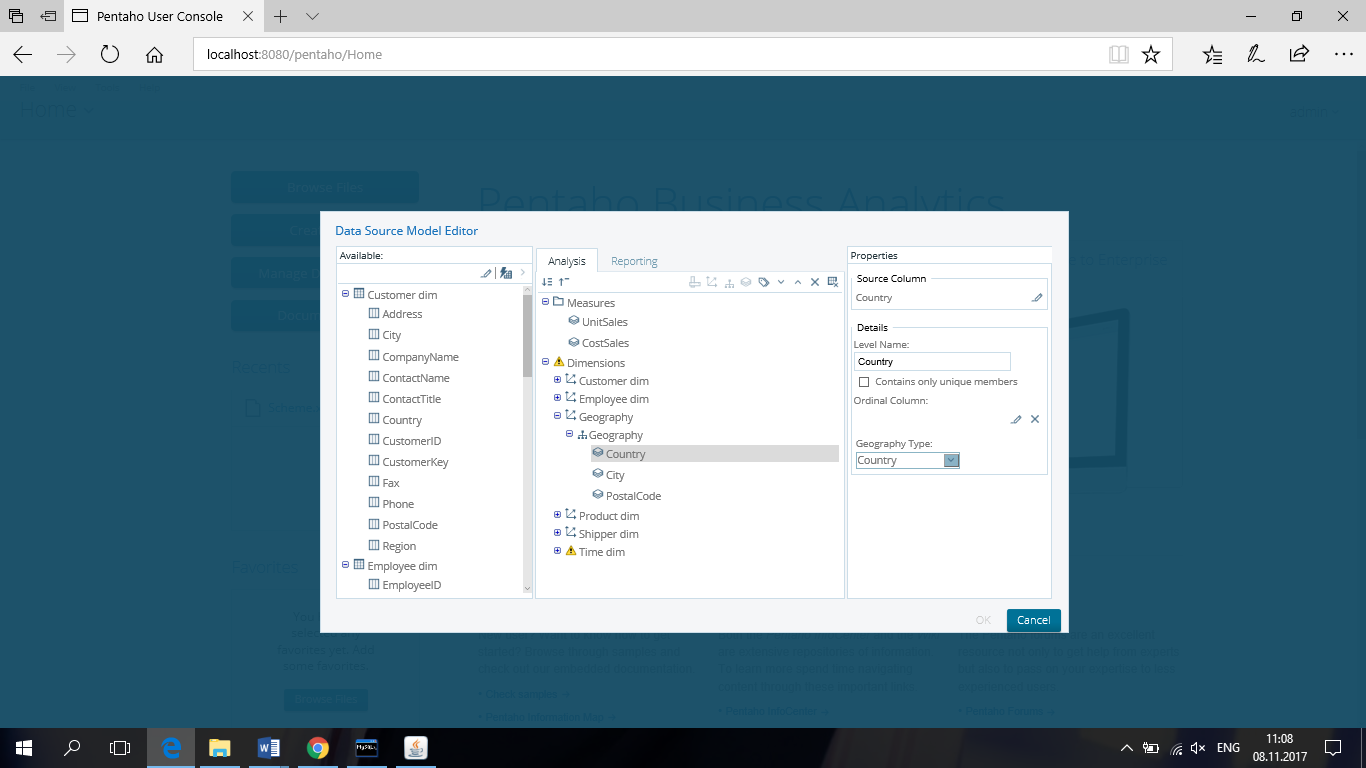


Рис.9. Тип полей измерения geography

Хранилище данных настроено, и можно переходить к формированию MDX запросов.

# Формирование MDX запросов

На главной странице Pentaho User Concole выбераем «File» - «New» - «Saiku Analytics». На открывшейся странице, на вкладке Enterprise в разделе Quick Links нажимаем «создать новый запрос» («Create a new query»).

В первую очередь необходимо указать куб, с которым будем работать в процессе формирования запросов. Saiku Analytics позволяет визуально создавать нужный запрос к хранилищу. Рассмотрим на примере Northwind (Рис.10).

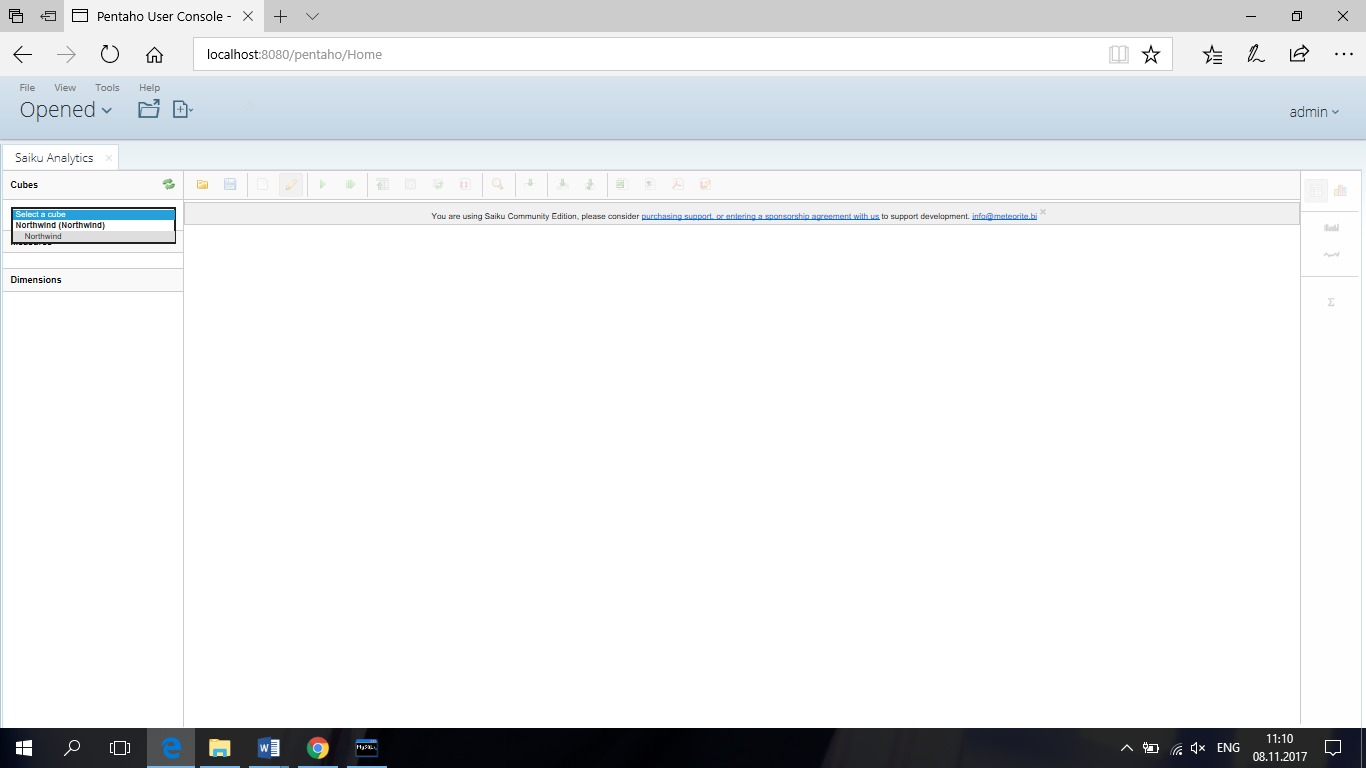


Рис.10. Выбор куба, к которому формируется запрос

Запрос №1: *Каково число продаж было совершено в каждом месяце в каждом конкретном городе с офисом компании за три года?*

1) Укажите меру. В нашем случае UnitSales – число продаж.

2) Укажите измерения времени – год и месяц. Сначала указываются измерения в строке, а затем заполняются столбцы. Измерение можно вручную переместить из столбца в строку (Рис.11).

3) Укажите измерение месторасположения – страна, город.

По мере того, как детализируется запрос, справа меняется таблица вывода. Наглядно можно убедиться в разнице запросов, если в строках сначала указывается месяц, а потом год и наоборот.

Результат запроса представлен на Рис.12.

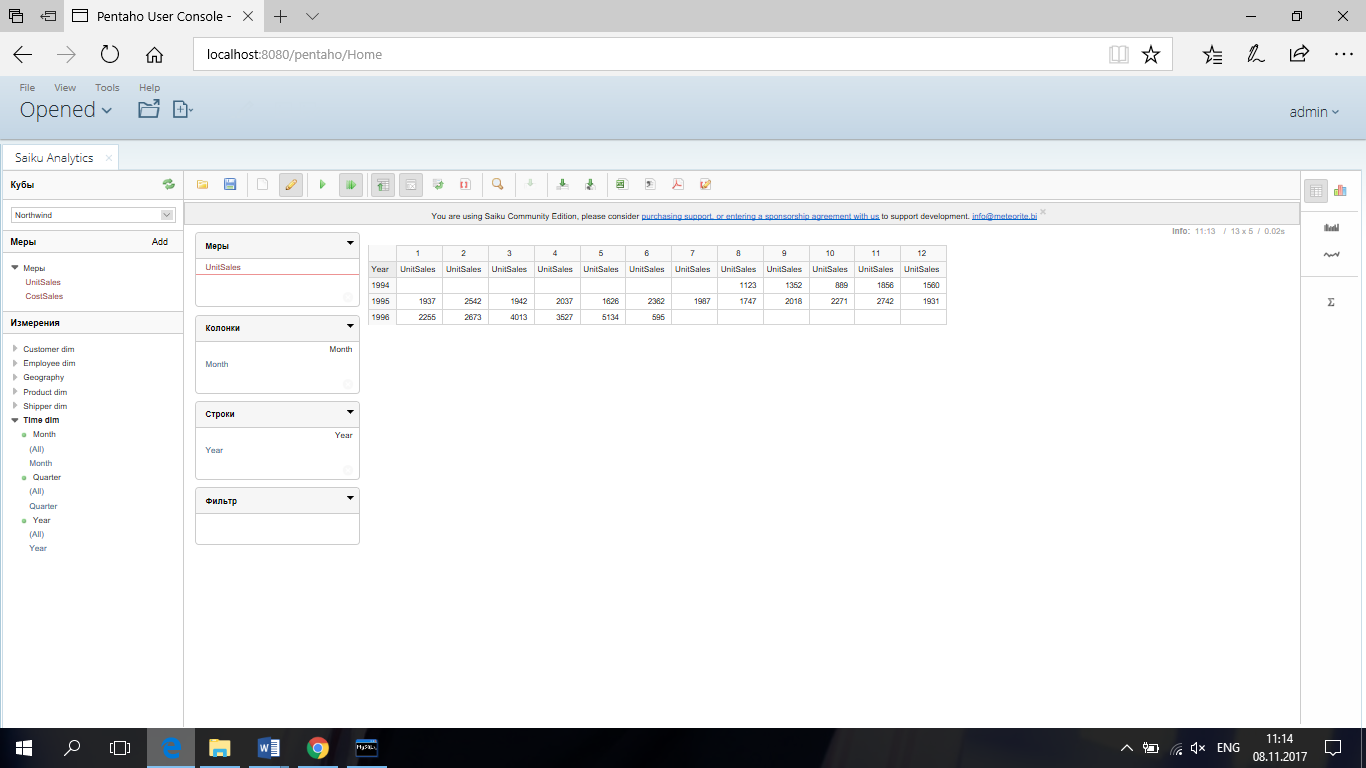


Рис.11. Задание строк и столбцов

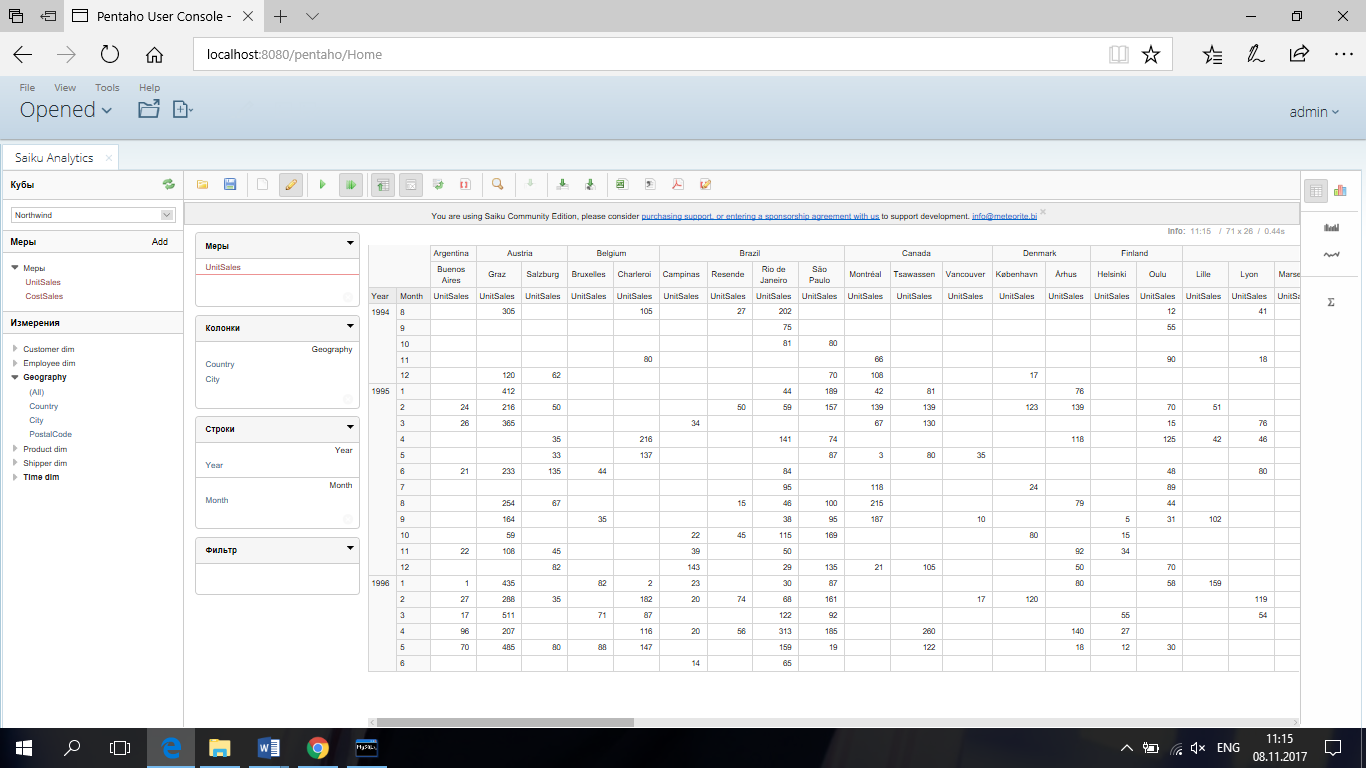


Рис.12. Результат проведения запроса №1

Как выглядит написание приведённого выше запроса №1 вручную, можно посмотреть, кликнув на панели управления “switch to MDX Mode”:

Листинг.1 – Запрос №1

WITH

SET [~COLUMNS] AS

Hierarchize({{[Geography].[Country].Members}, {[Geography].[City].Members}})

SET [~ROWS\_Time .Month] AS

{[Time .Month].[Month].Members}

SET [~ROWS\_Time .Year] AS

{[Time .Year].[Year].Members}

SELECT

NON EMPTY CrossJoin([~COLUMNS], {[Measures].[UnitSales]}) ON COLUMNS,

NON EMPTY NonEmptyCrossJoin([~ROWS\_Time .Month], [~ROWS\_Time .Year]) ON ROWS

FROM [Northwind]

Рассмотрим написание простого запроса вручную.

Запрос №2: *Сравнительный анализ числа продаж и стоимостей в различных городах.*

Листинг.2 – Запрос №2

SELECT {[Measures].[UnitSales], [Measures].[CostSales]} ON COLUMNS,

{[Geography].[City].Members} ON ROWS

FROM Northwind

Результатом данного запроса является длинная таблица со списком всех городов базы данных (Рис.13).

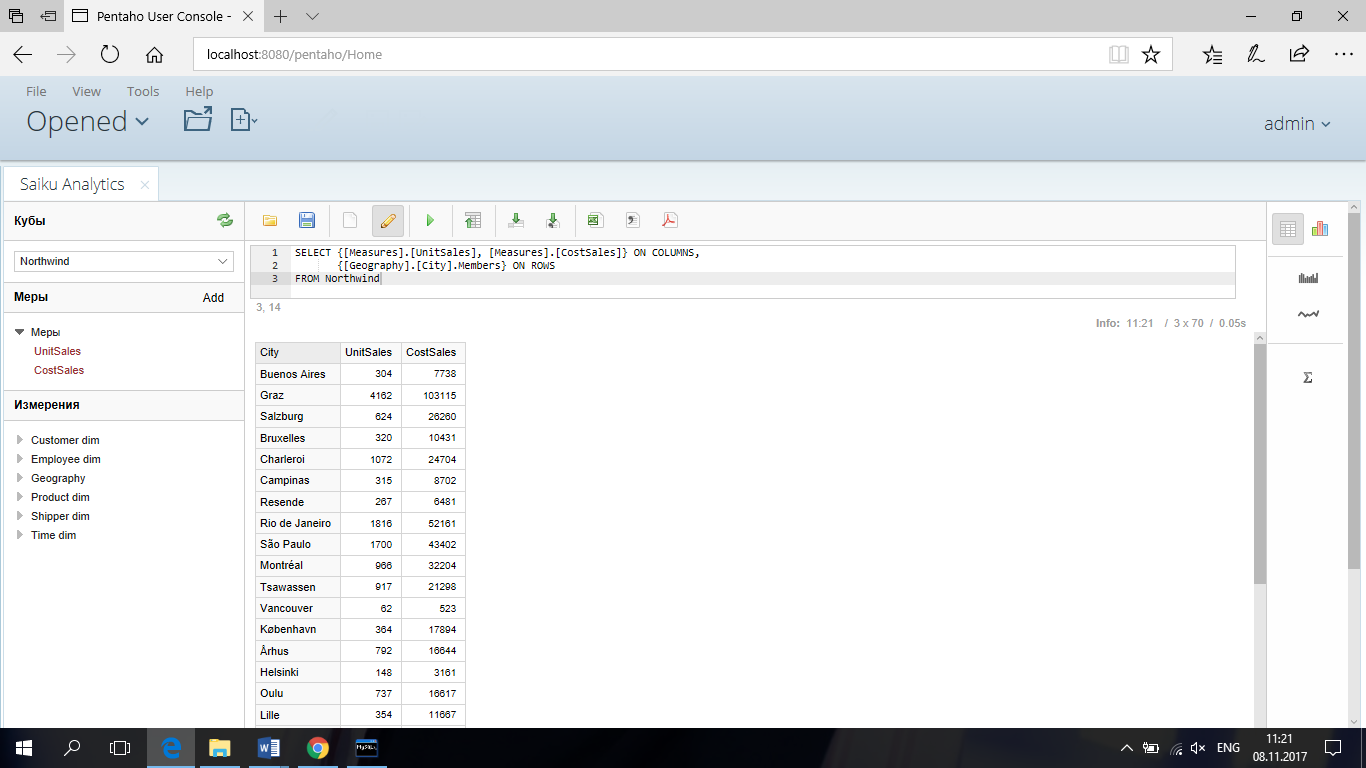


Рис.13. Результат запроса №2

Помимо таблиц результат запроса можно выводить в графическом виде. Это могут быть кольцевые или столбчатые диаграммы, широкий выбор которых представлен справа при смене режима Table Mode на Chart Mode. Результат запроса №2 в графическом виде представлен на Рис.14.

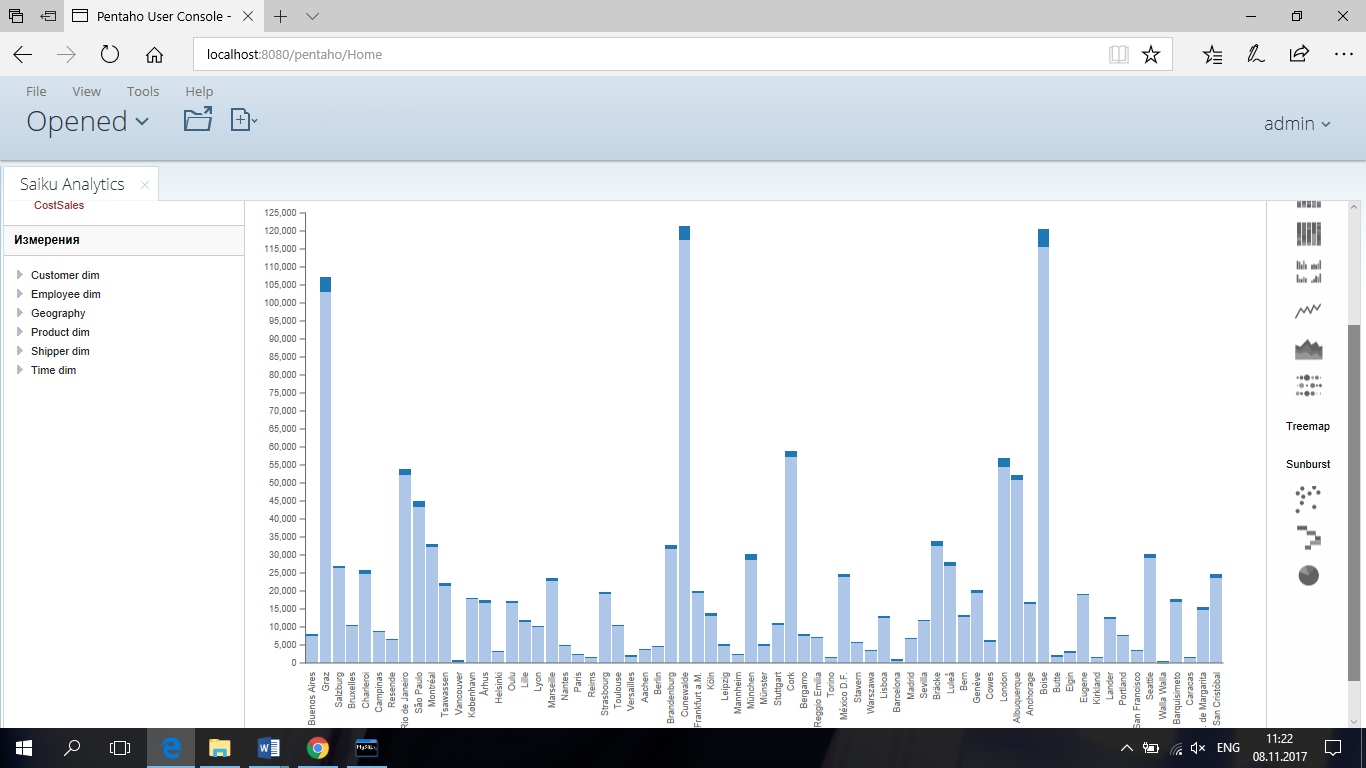


Рис. 14. Результат запроса №2 в графическом виде

Запрос №3: *Сравнительный анализ числа продаж и стоимостей в различных странах.*

Листинг.3 – Запрос №3

SELECT {[Measures].[UnitSales], [Measures].[CostSales]} ON COLUMNS,

NON EMPTY {[Geography].[Country].Members} ON ROWS

FROM Northwind

Примером отображения запроса в виде круговых диаграмм является результат запроса №3 на Рис.15.

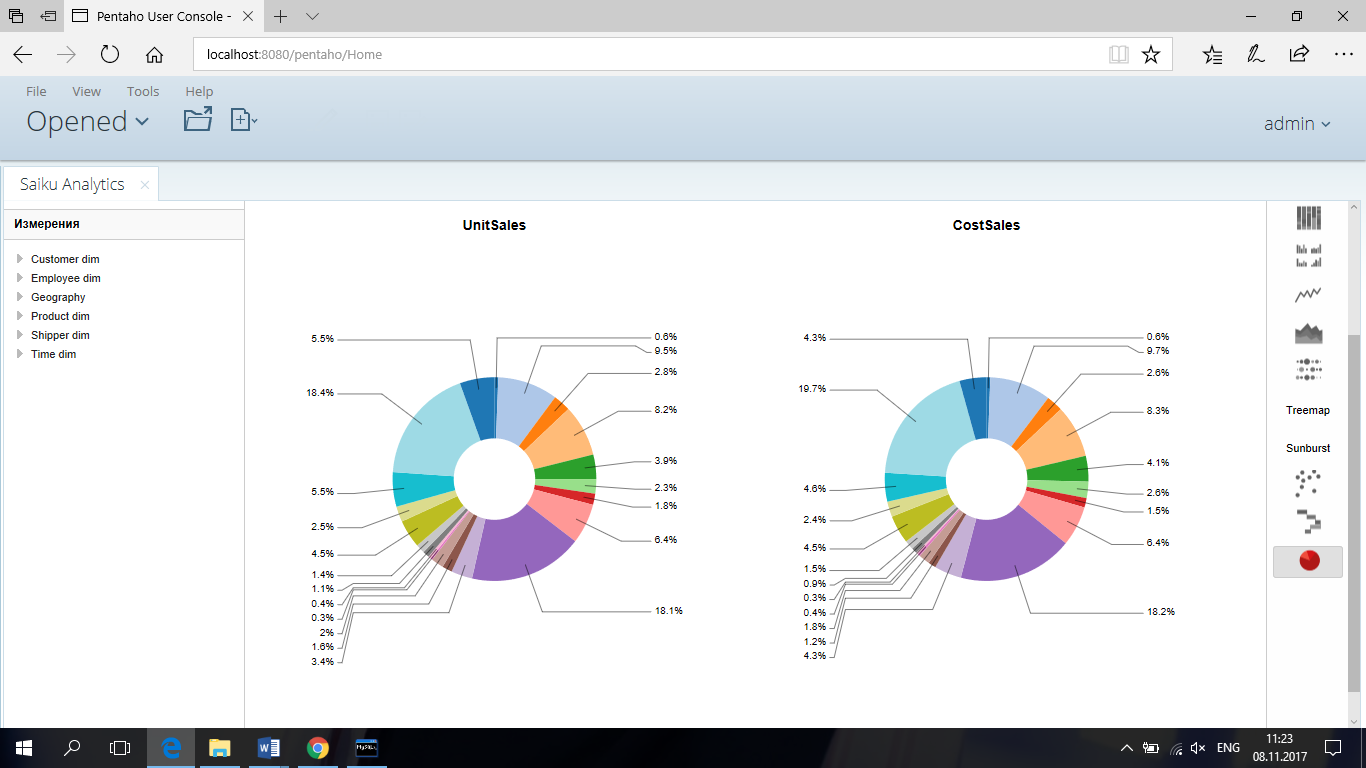


Рис.15. Результат запроса №3 в виде круговых диаграмм

**Часть 2**

1) Функция .Members

Возвращает набор элементов в измерении, уровне или иерархии. Реализация функции представлена в листинге 4. Результат работы функции отображен на рис.16. В графическом виде данные представлены в виде круговой диаграммы (рис.17).

Листинг 4

Select

[Measures].[UnitSales] ON COLUMNS,

[Time dim].[Month].members ON ROWS

From

[Northwind]

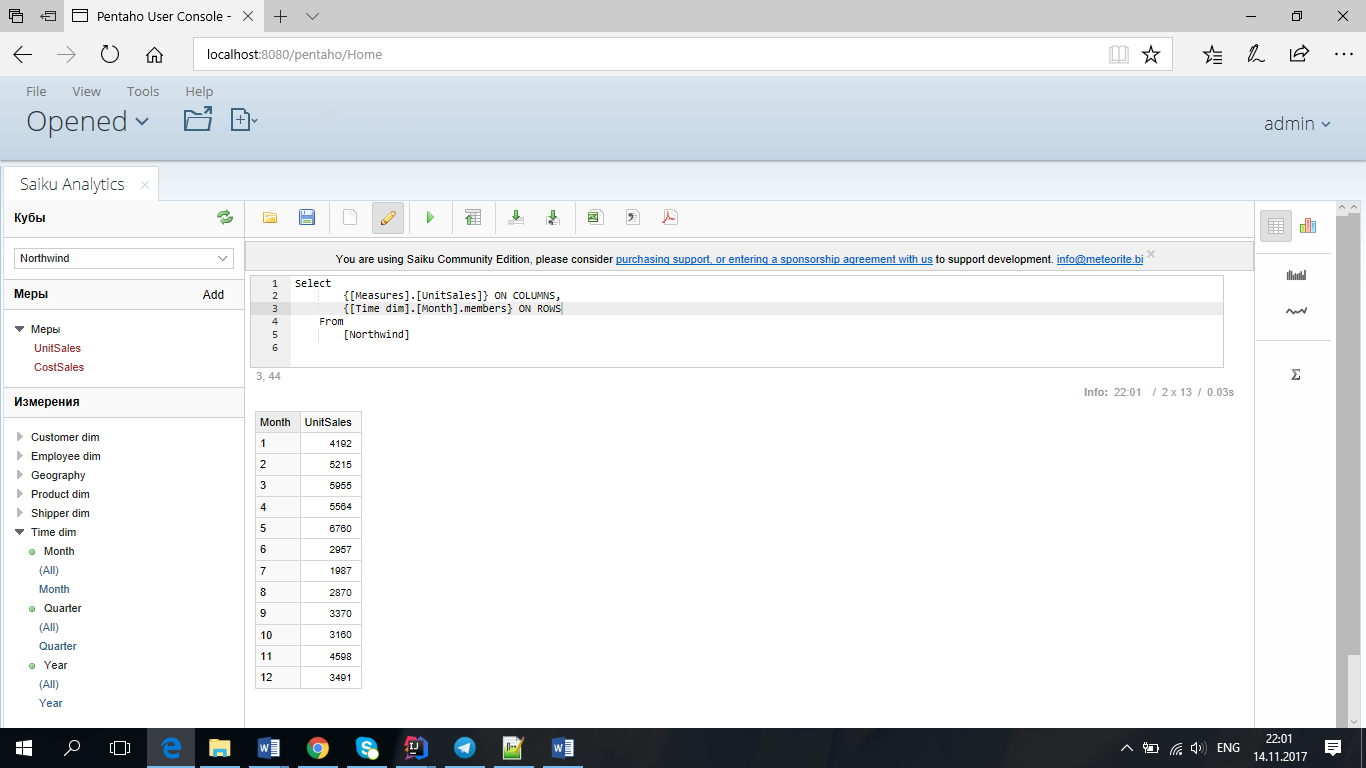


Рис.16. Результат работы функции .Members

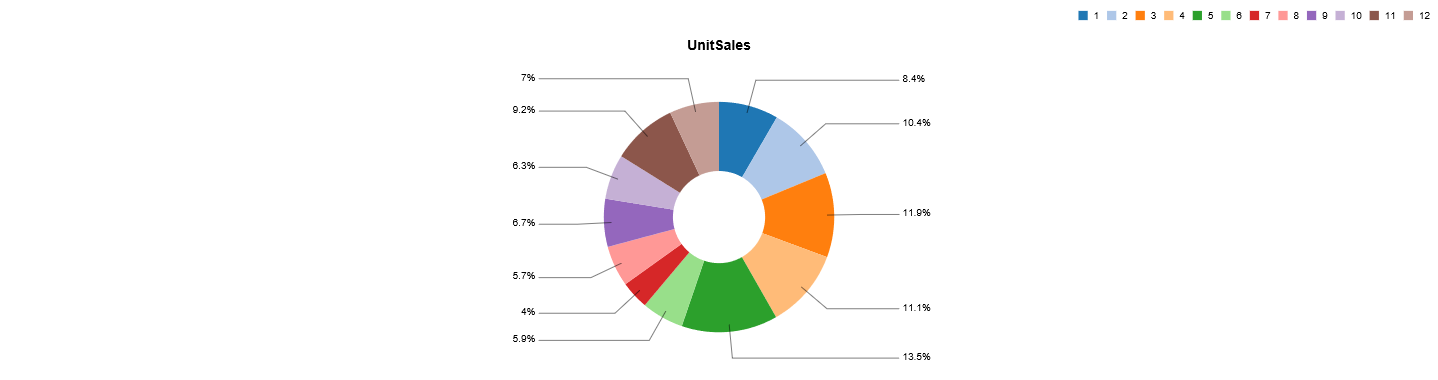


Рис. 17. Результат работы функции .Members в графическом виде

2) Функция .NextMember

NextMember функция возвращает следующий элемент на том же уровне, который содержит указанный элемент. Реализация функции представлена в листинге 5. Результат работы функции отображен на рис.18. В графическом виде данные представлены в виде Heat Grid (рис.19).

Листинг 5

Select

[Measures].[CostSales] ON COLUMNS,

[Time dim].[Month].[4].NextMember on rows

From

[Northwind]

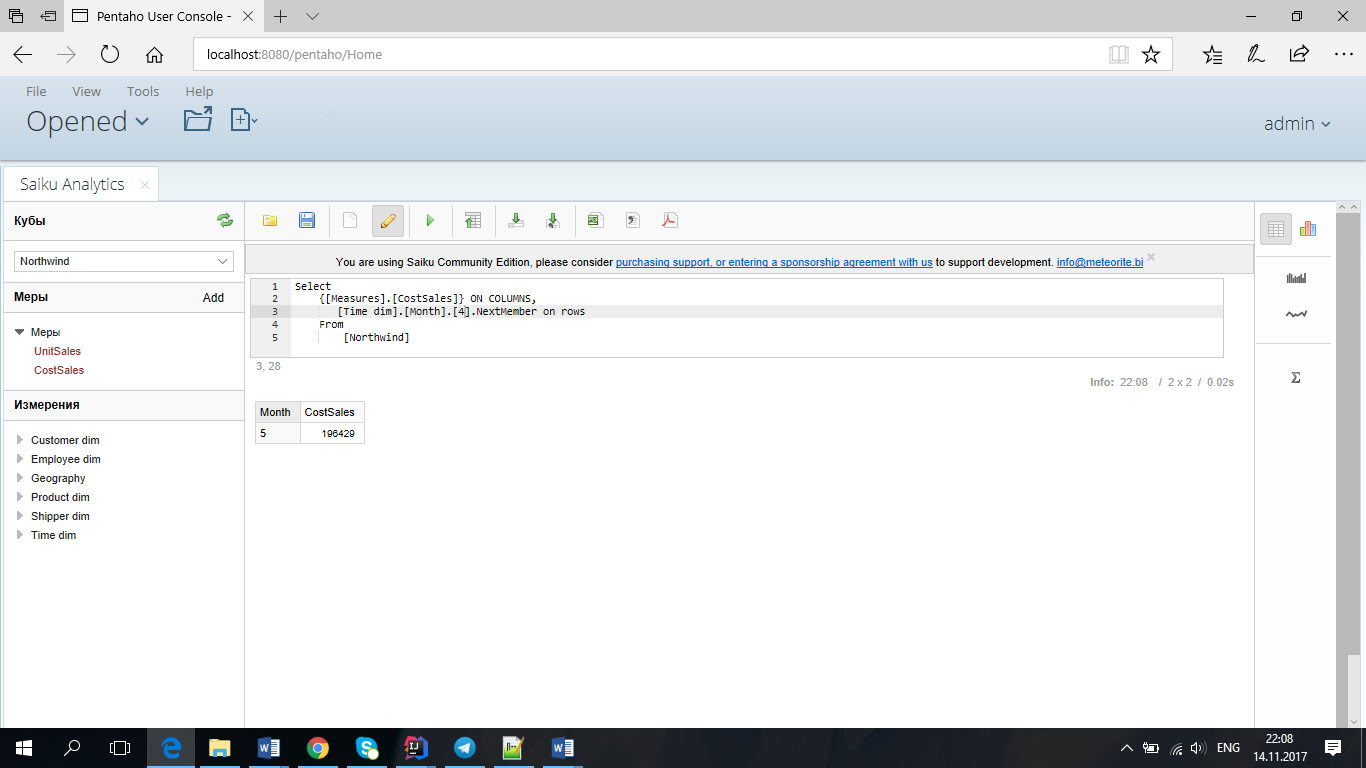


Рис.18. Результата работы функции .NextMember

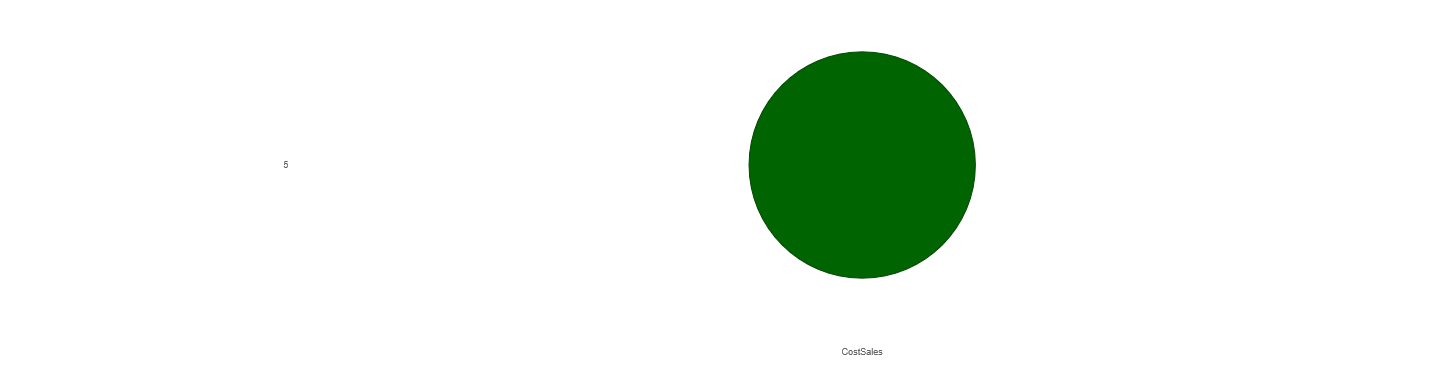


Рис. 19. Результат работы функции .NextMember в графическом виде

3) Функция .PrevMember

Функция .PrevMember возвращает предыдущий элемент уровня, содержащего указанный элемент. Реализация функции представлена в листинге 6. Результат работы функции отображен на рис.20. В графическом виде данные представлены в виде Heat Grid (рис.21).

Листинг 6.

Select

[Measures].[CostSales] ON COLUMNS,

[Time dim].[Month].[6].PrevMember on rows

From

[Northwind]

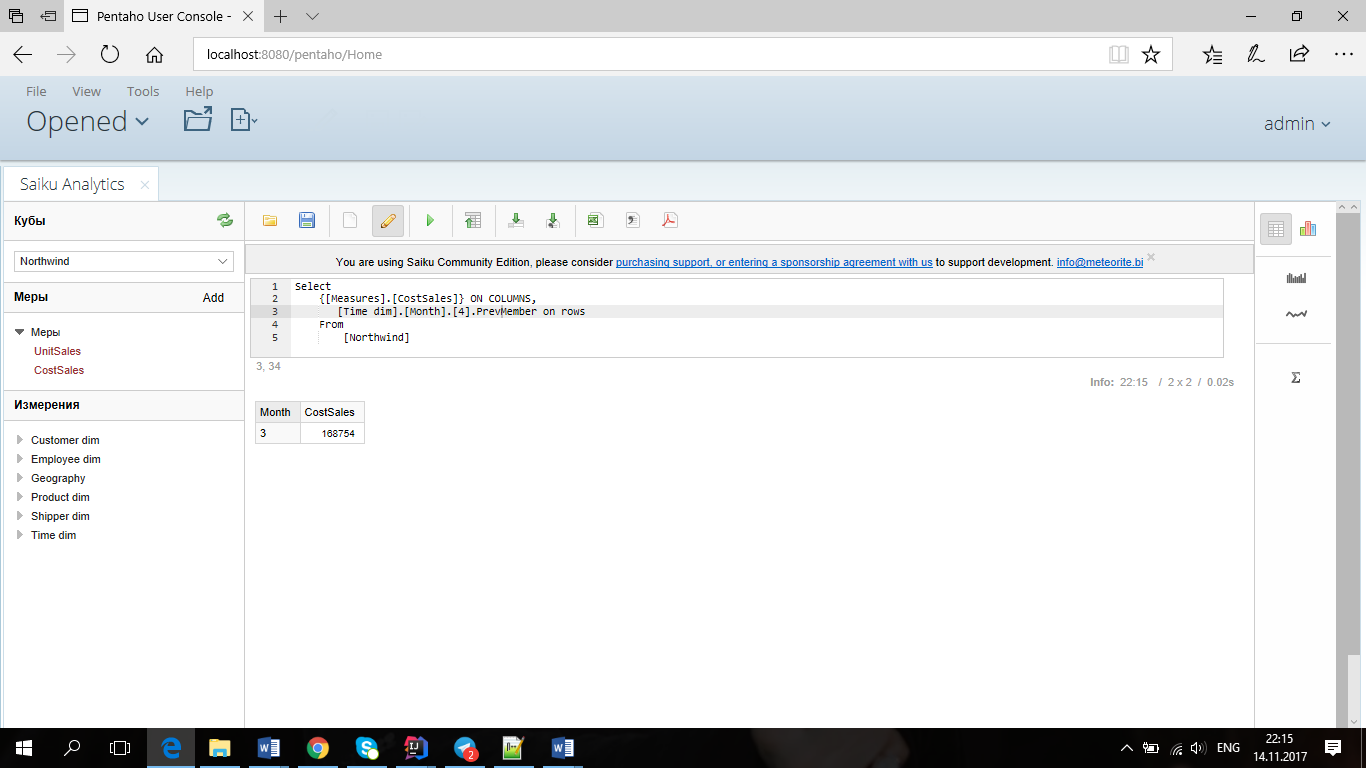


Рис.20 Результата работы функции .PrevMember

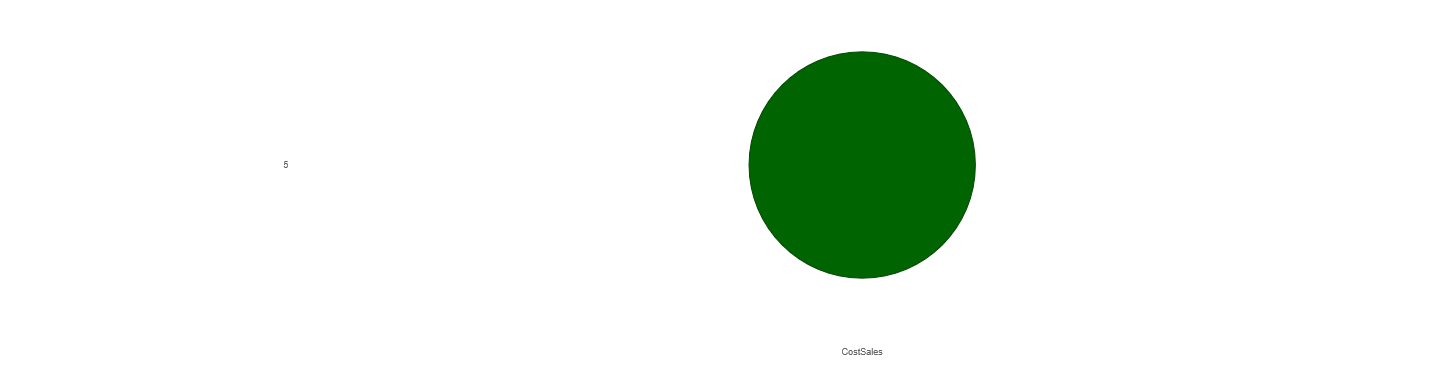


Рис. 21. Результат работы функции .PrevMember в графическом виде

4) Функция .Lag(N)

Функция .Lag(N) возвращает элемент, который находится на указанное количество позиций ранее заданного элемента на его уровне. Реализация функции представлена в листинге 7. Результат работы функции отображен на рис.22. В графическом виде данные представлены в виде Heat Grid (рис.23).

Листинг 7.

Select

[Measures].[CostSales] ON COLUMNS,

[Time dim].[Month].[6].Lag(3) on rows

From

[Northwind]

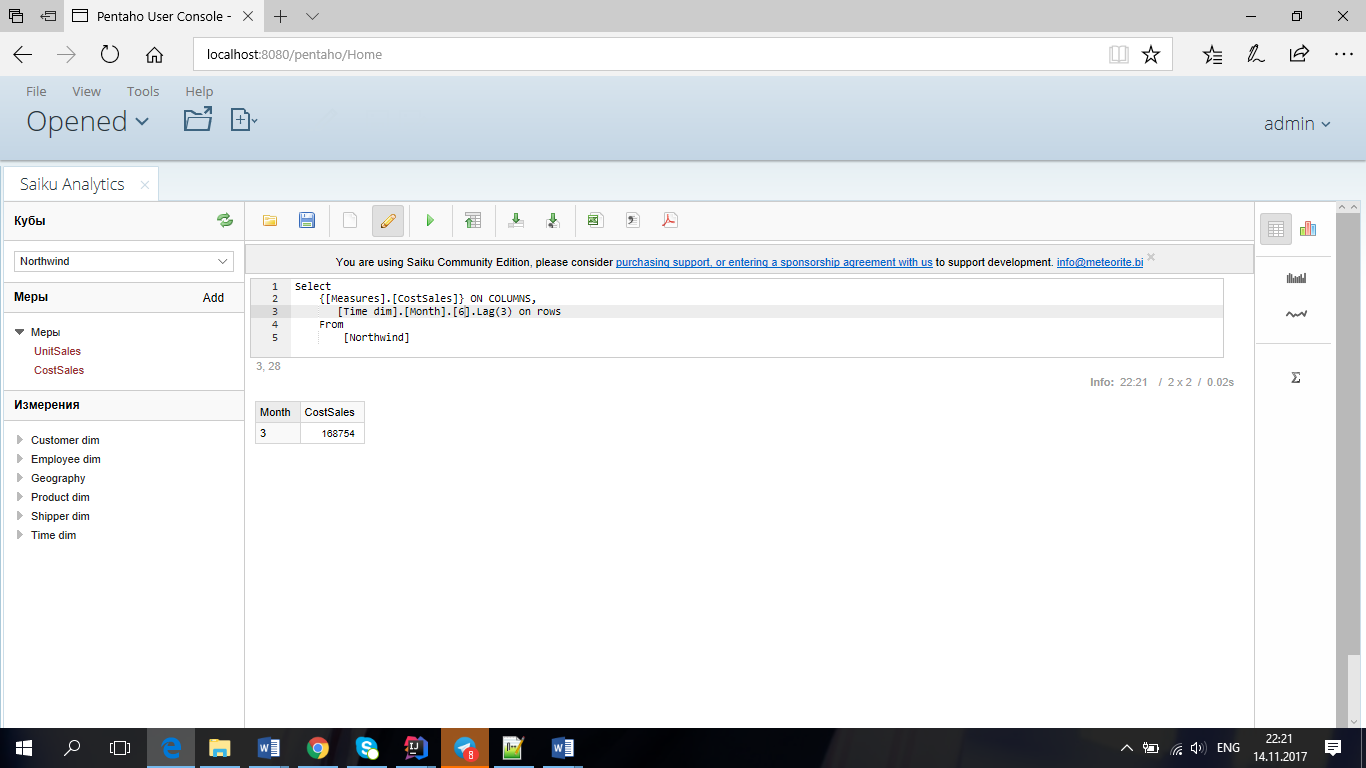


Рис.22. Результат работы функции .Lag(N)

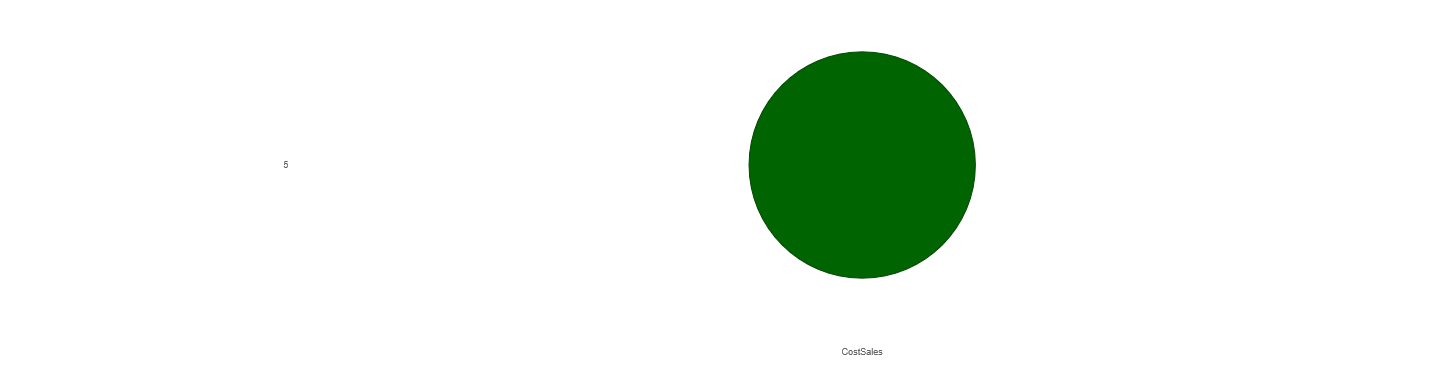


Рис. 23. Результат работы функции .Lag(N) в графическом виде

5) Функция .Lead(N)

Функция .Lead(N) возвращает элемент, который следует за заданным элементом через указанное число позиций на уровне элемента. Реализация функции представлена в листинге 8. Результат работы функции отображен на рис.24. В графическом виде данные представлены в виде Heat Grid (рис.25).

Листинг 8.

Select

[Measures].[CostSales] ON COLUMNS,

[Time dim].[Month].[6].Lead(3) on rows

From

[Northwind]

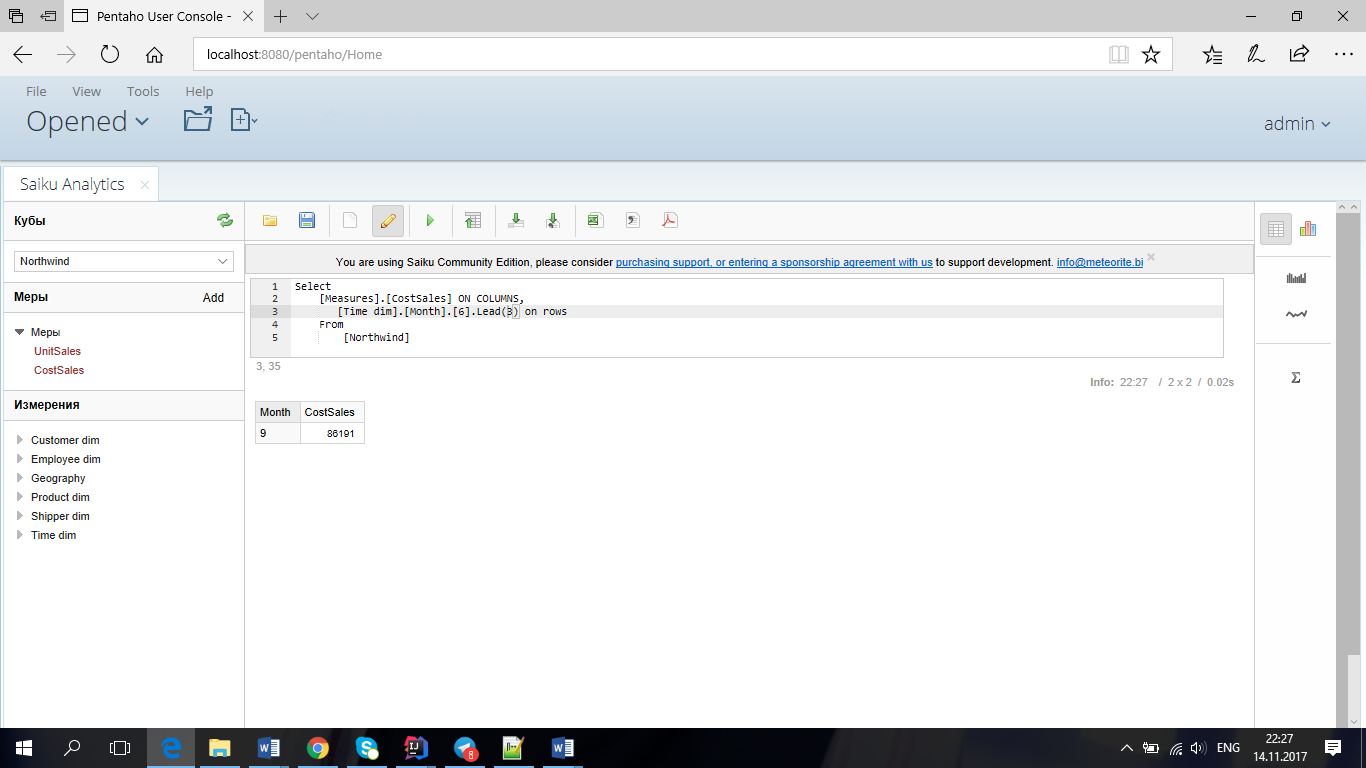


Рис.24. Результат работы функции .Lead(N)

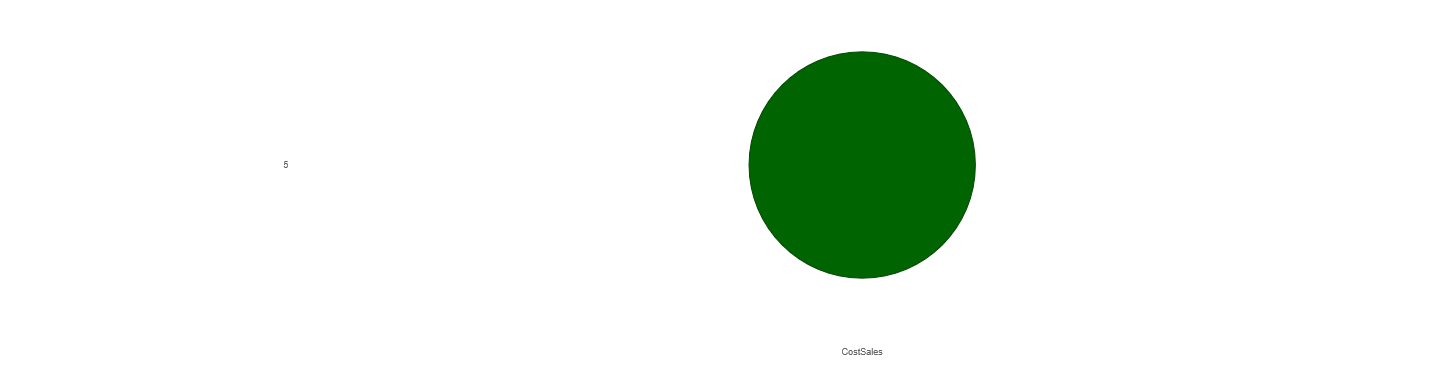


Рис. 25. Результат работы функции .Lead(N) в графическом виде

6) Функция .Children

Функция .Children возвращает набор потомков указанного элемента. Реализация функции представлена в листинге 9. Результат работы функции отображен на рис.26. В графическом виде данные представлены в виде графика (рис.27).

Листинг 9.

Select

[Measures].[UnitSales] ON COLUMNS,

[Geography].Children on rows

From

[Northwind]

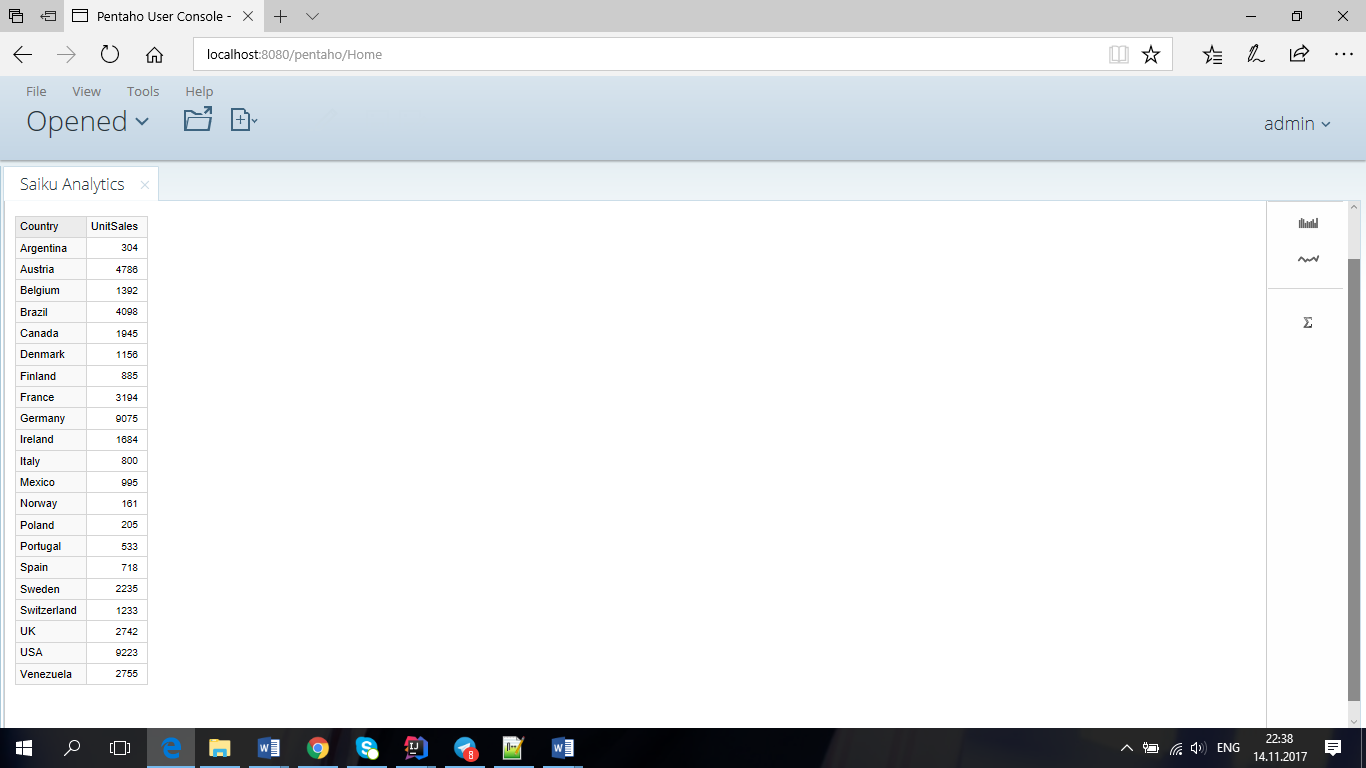


Рис.26. Результат работы функции .Children

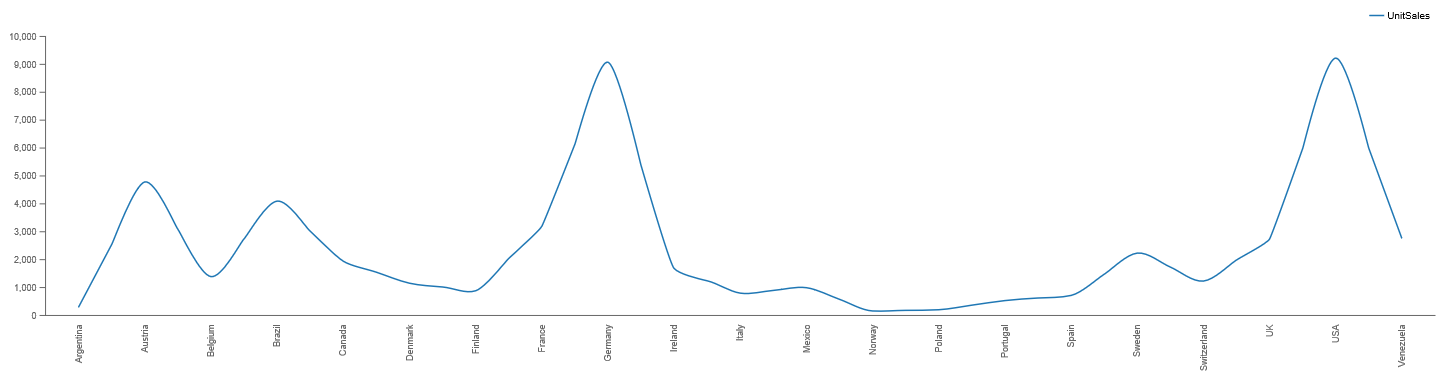


Рис. 27. Результат работы функции .Children в графическом виде

7) Функция .CrossJoin

Функция .CrossJoin возвращает перекрестное произведение двух или нескольких наборов. Реализация функции представлена в листинге 10. Результат работы функции отображен на рис. 28. В графическом виде данные представлены в виде столбчатой диаграммы (рис. 29).

Листинг 10.

Select [Measures].[CostSales] on columns,

crossjoin ([Time dim].[Month].members, [Geography].[Country].[USA])on rows

from [Northwind]

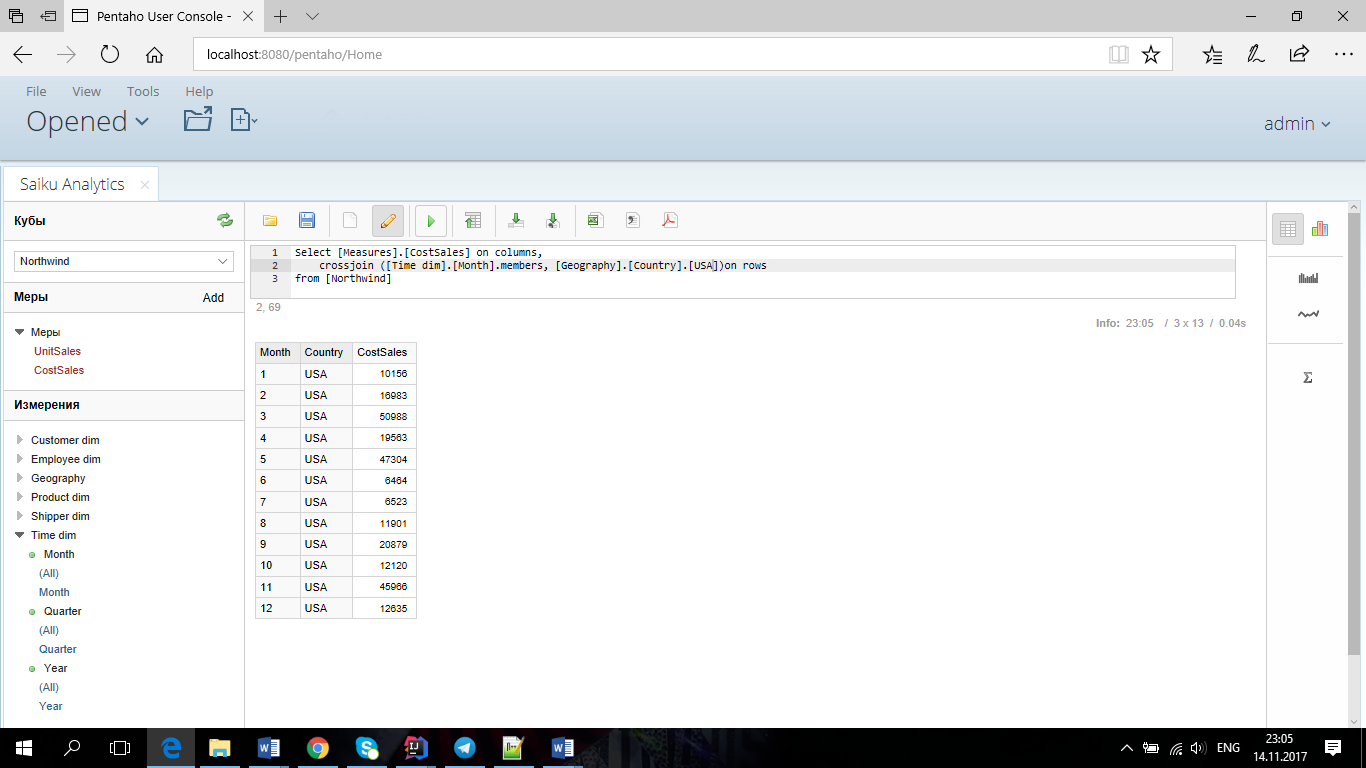


Рис.28. Результат работы функции crossJoin

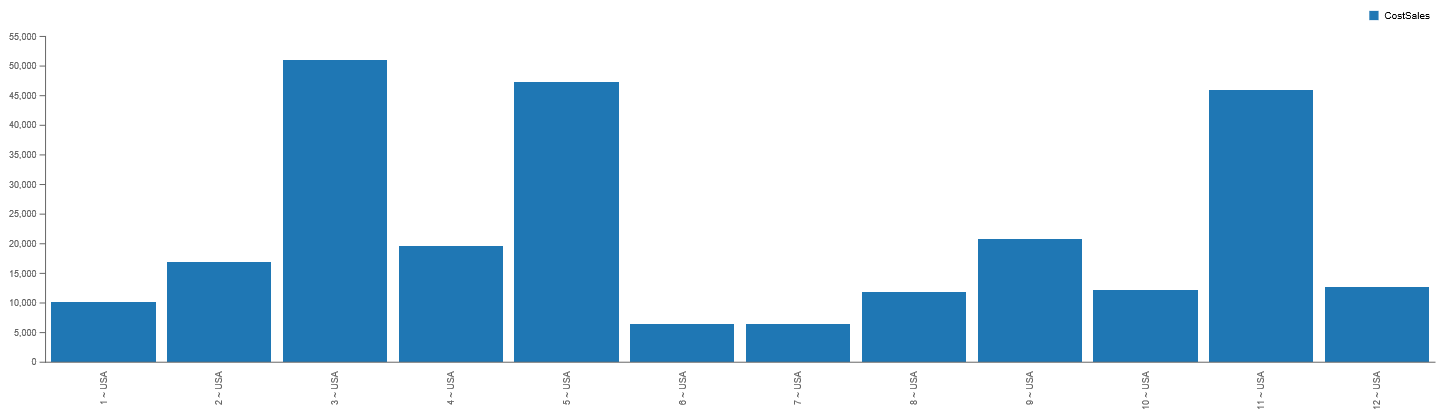


Рис. 29. Результат работы функции crossJoin в графическом виде

8) Функция Filter()

Функция Filter() возвращает набор, получающийся в результате фильтрации заданного набора на основе условия поиска. Реализация функции представлена в листинге 11. Результат работы функции отображен на рис.30. В графическом виде данные представлены в виде столбчатой диаграммы (рис.31).

Листинг 11.

Select [Measures].[CostSales] on columns,

Filter([Geography].[City].members,[Geography].CurrentMember.Parent IS [Geography].[USA] ) on rows from [Northwind]

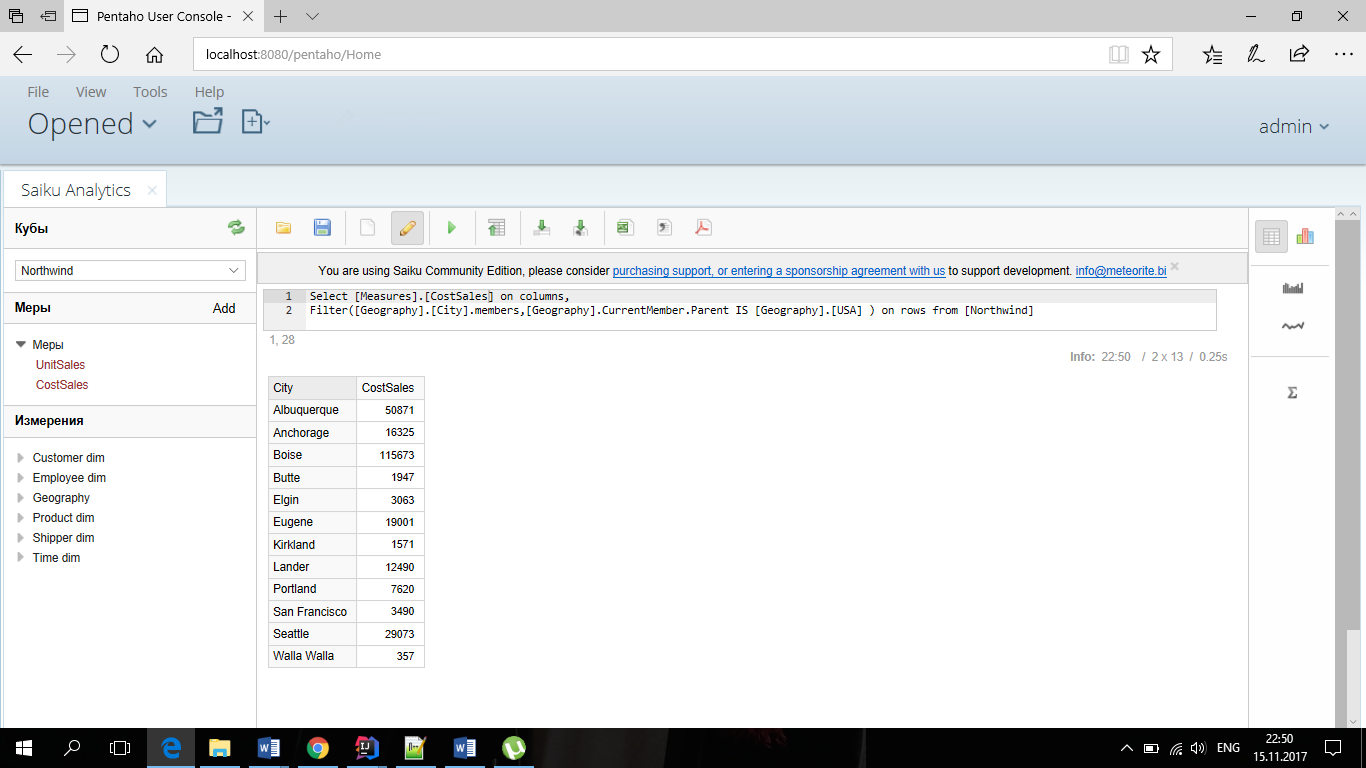


Рис.30. Результат работы функции Filter()

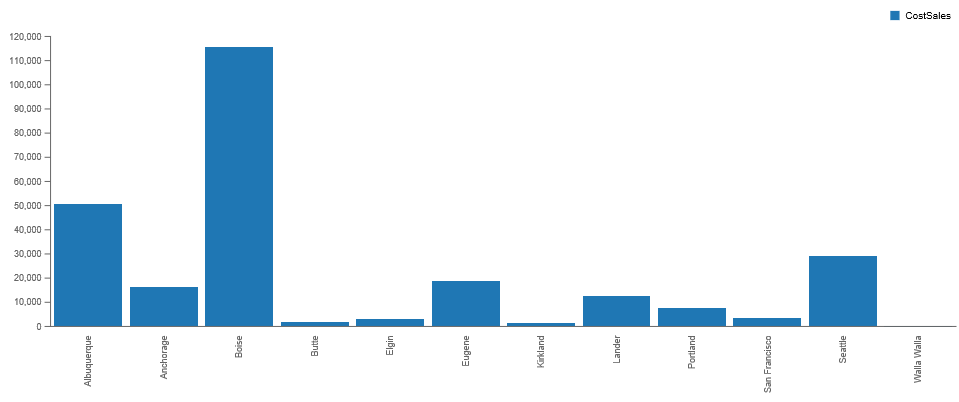


Рис. 31. Результат работы функции Filter() в графическом виде

9) Функция Order()

Функция Order() упорядочивает элементы указанного набора, по выбору сохраняя или нарушая иерархию. Реализация функции представлена в листинге 12. Результат работы функции отображен на рис.32. В графическом виде данные представлены в виде столбчатой диаграммы (рис.33).

Листинг 12.

Select [Measures].[CostSales] on columns,

Order([Time dim].[Month].members,[Measures].[CostSales], DESC) on rows

from [Northwind]

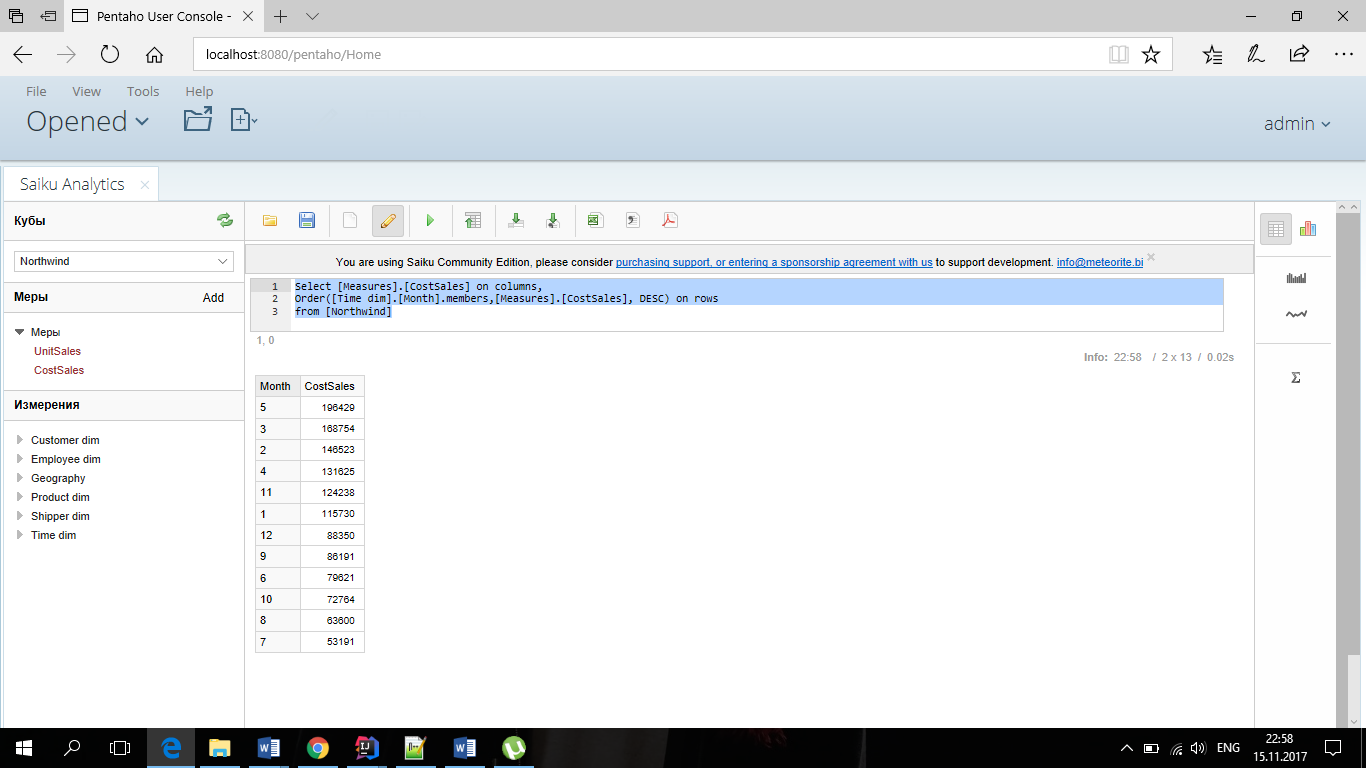


Рис.32. Результат работы функции Order()

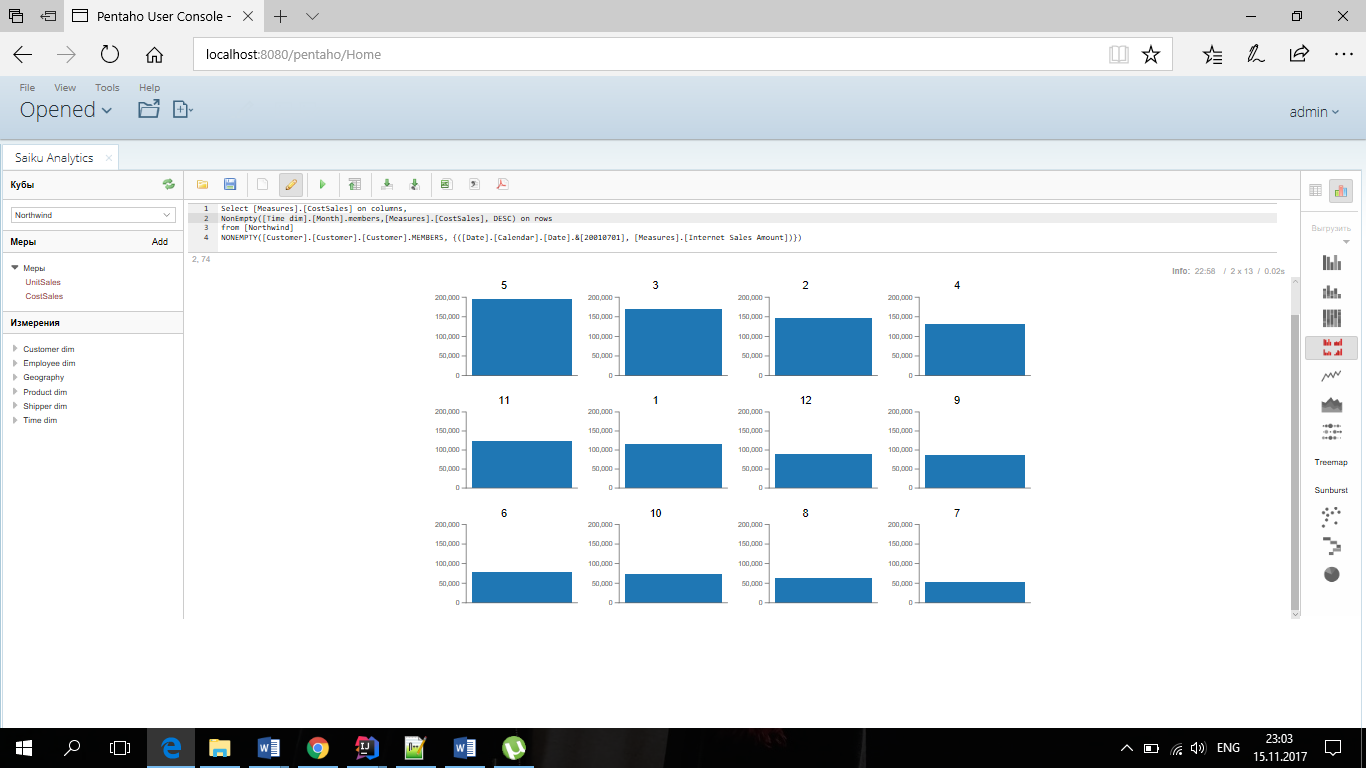


Рис. 33. Результат работы функции Order() в графическом виде

10) Функция NonEmpty

Функция NonEmpty возвращает набор непустых кортежей из заданного набора, основываясь на прямом произведении заданного набора со вторым набором. Реализация функции представлена в листинге 13. Результат работы функции отображен на рис.34. В графическом виде данные представлены в виде столбчатой диаграммы (рис.35).

Листинг 13.

SELECT

{[Employee dim.Name].[Name].Members} ON COLUMNS,

NON EMPTY {[Employee dim.HireDate].[HireDate].Members} ON ROWS

FROM [Northwind]

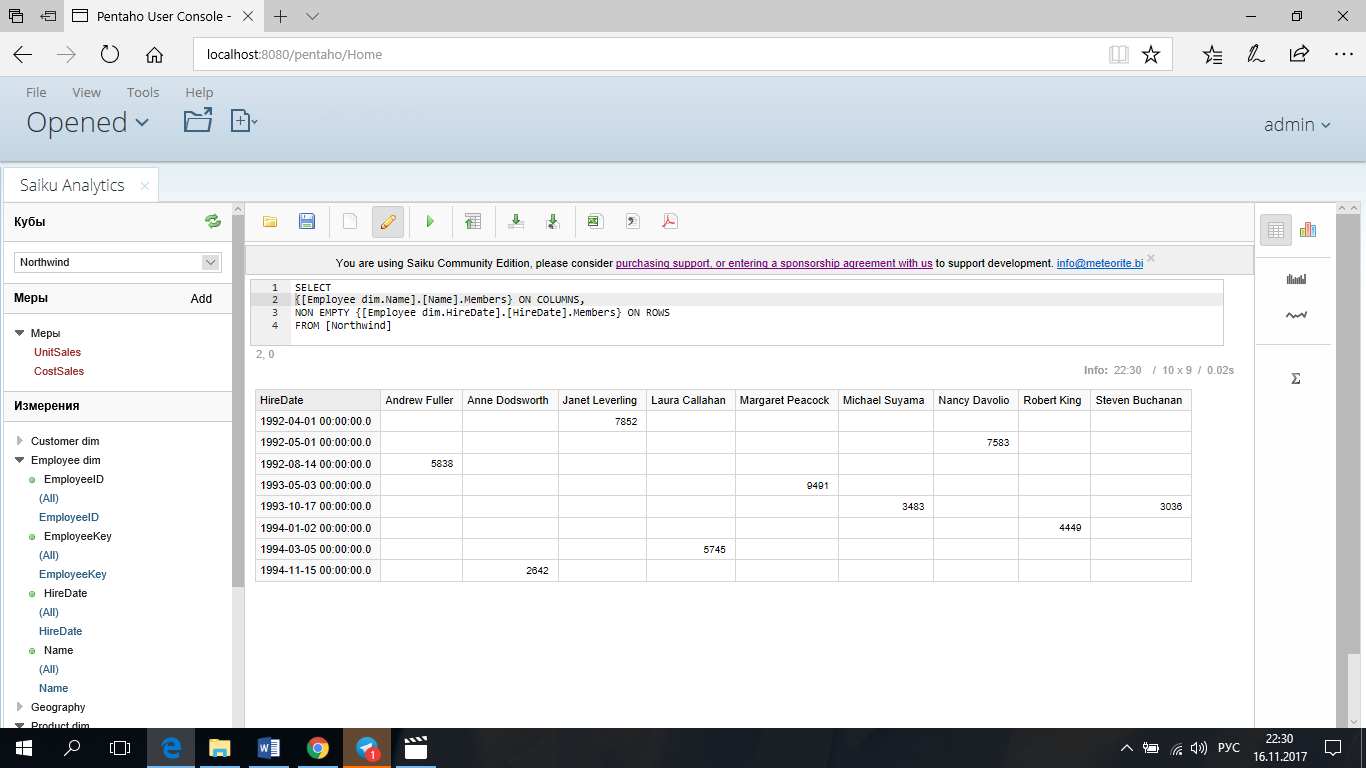


Рис.34. Результат работы функции NonEmpty

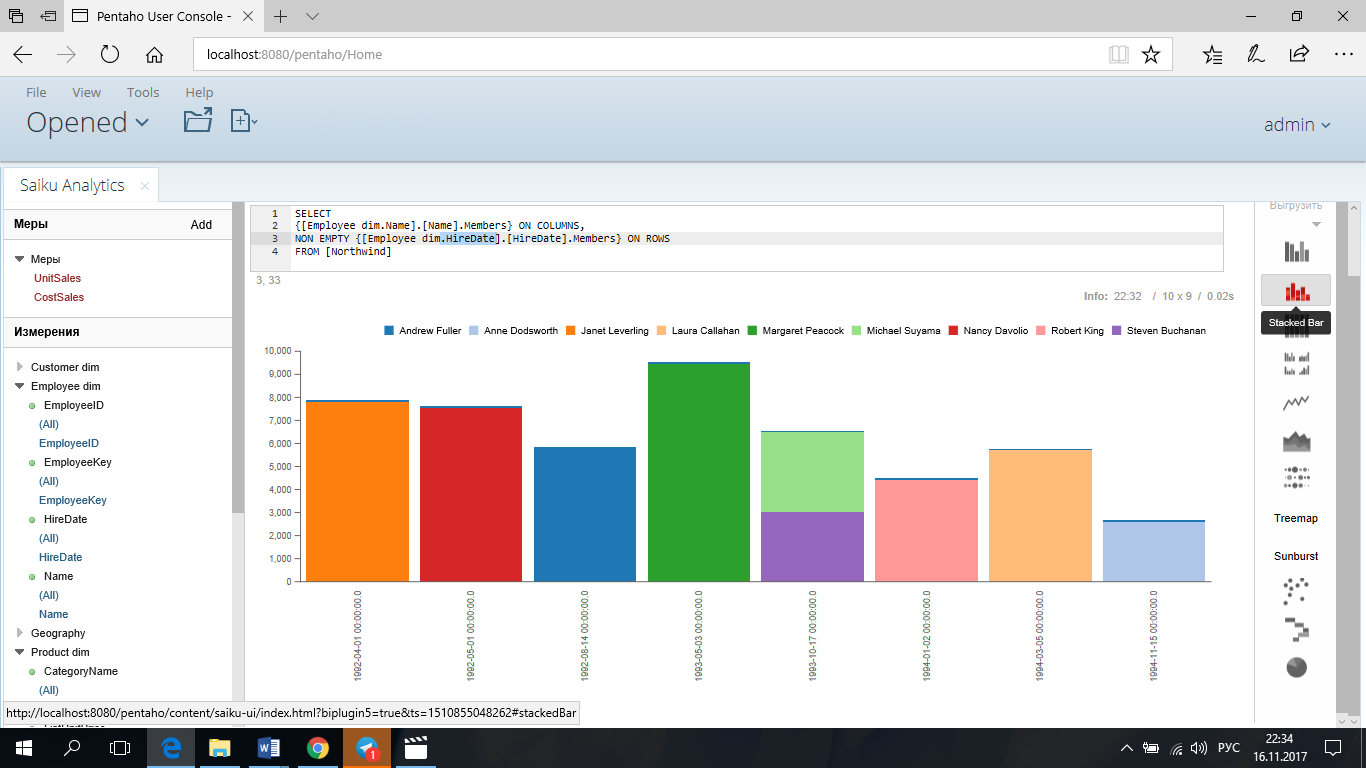


Рис. 35. Результат работы функции NonEmpty в графическом виде

# Заключение

В рамках данной лабораторной работы я рассмотрел основы языка MDX (Multidimensional Expressions) и применил его к созданному хранилищу данных Northwind с помощью Pentaho Community Edition.