SCHOOL OF COMPUTING

UNIVERSITY OF TEESSIDE

MIDDLESBROUGH

TS1 3BA

Advanced Databases Systems

BSc Computer Science

Kiril Anastasov

25th April 2014

Supervisors: Capper, Graham (Dr.),

Longstaff, Jim (Dr.)

Abstract

We are required to submit only a **single pdf** (portable document format) file that documents all our answers based on the following scenario SomeCity University holds interview days for candidates applying to its degree courses. The purpose is to appraise the students' ability and then make them an offer regarding acceptance onto a degree course; an offer is the total number of points they need to gain from their A-level or other qualifications. SomeCity would like to develop a database and associated applications to manage this activity. The following documents highlight the data in the system. Read through the documents carefully. If you have queries you can contact your local tutor who will act as a client from SomeCity University.

Acknowledgements

I am grateful for the help and support of my supervisors Capper, Graham (Dr) and Longstaff, Jim (Dr). Without the on-going support them gave me, it would not have been possible to complete this project. They were able to get in touch despite the time and place, in order to help for achieving most of the goals. They made the effort to personally get to know me and my project and give feedback where it was needed.

By: **Kiril Anastasov L1087591@live.tees.ac.uk**

CONTENTS

By: Kiril Anastasov L1087591@live.tees.ac.uk

Table of Figures

By: **Kiril Anastasov L1087591@live.tees.ac.uk**

By: **Kiril Anastasov L1087591@live.tees.ac.uk**

Task1:

The corrections that were made are:

1. There is inconsistency of notations (Chen, Crow's foot) on many places that have been used in Document 1. To solve this problem UML class diagram was used.

2. Optional, And were used in the generalisation where the correct way is Mandatory, Or .

3. Candidate needs to be directly associated with Appointment which was deducted from document 3, Appointment have details about Candidate.

4. Candidate should not be linked with InterviewDay, the details about Candidate are not required according to document 2.

5. Staff should be directly associated with InterviewDay as the details about Staff are required in the IntreviewDay according to document 2.

By: Kiril Anastasov L1087591@live.tees.ac.uk

Corrected diagram:

It was used the UML notation to redraw the diagram.

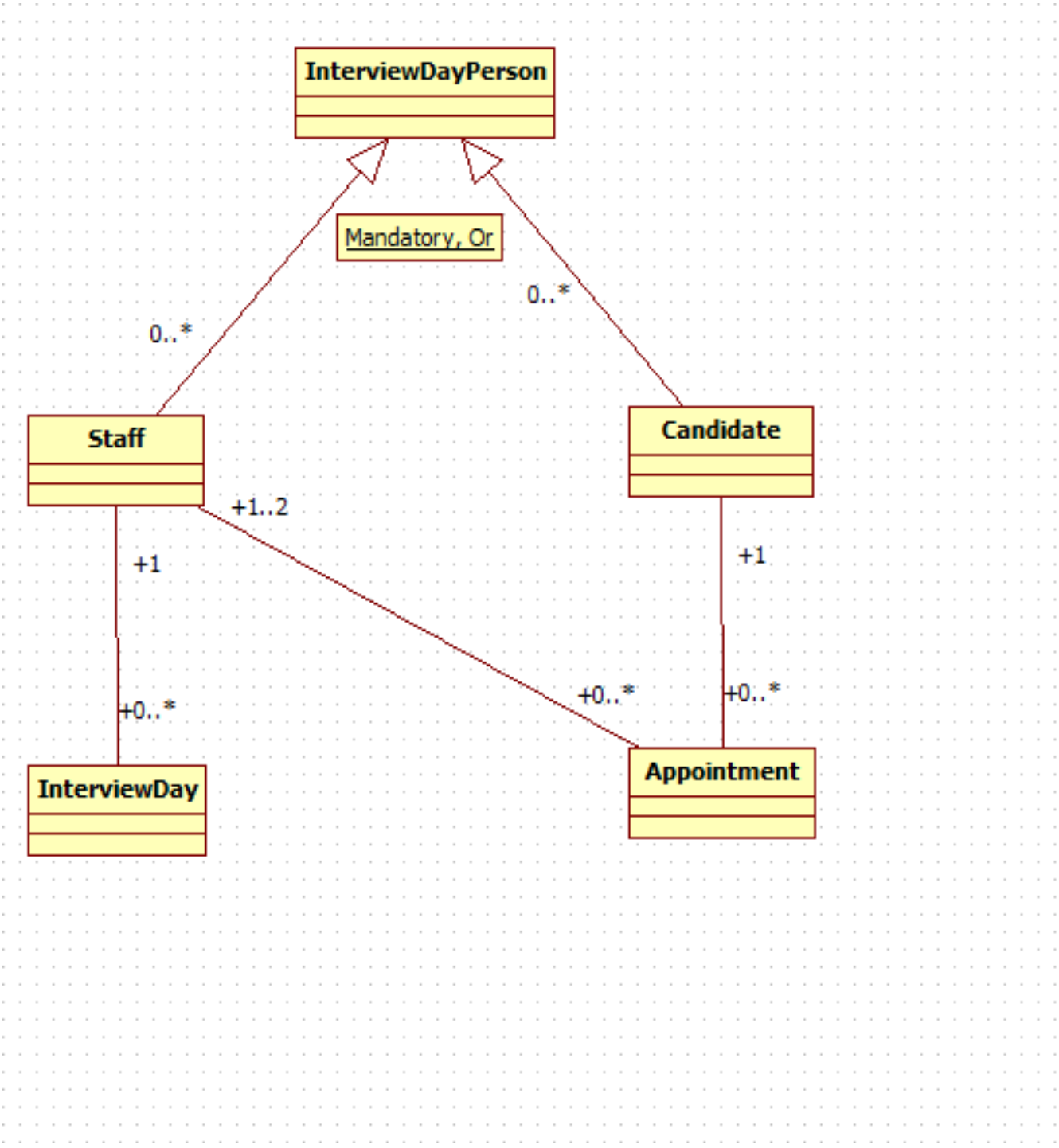


Figure 1 Corrected UML diagram

By: Kiril Anastasov L1087591@live.tees.ac.uk

Task 2: Full Data Model: Notation: **UML** diagram

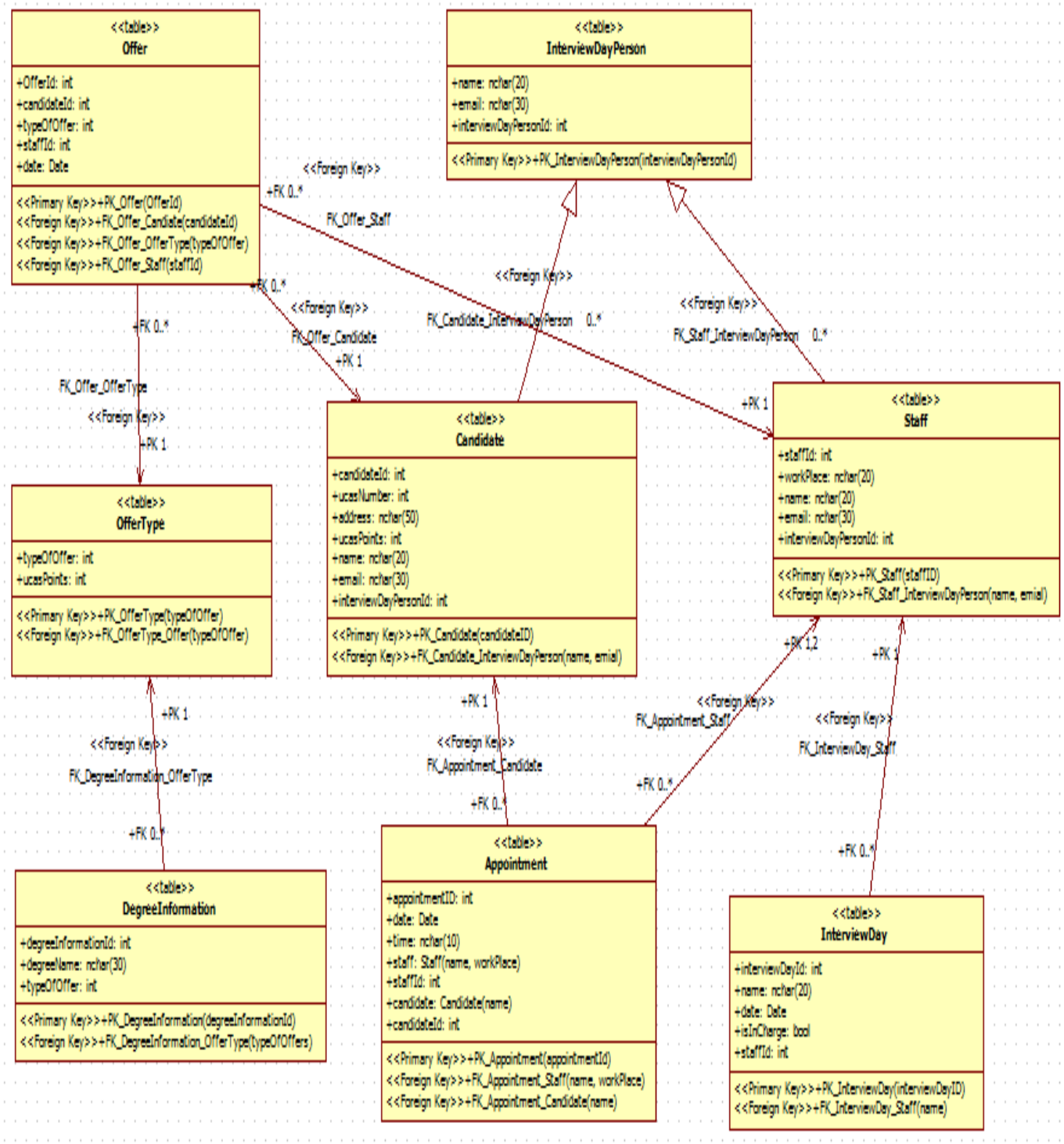


Figure 2: Full data model

The attributes *name* and *email* were defined in Candidate and Staff tables even though they inherit it from the superclass InterviewDayPerson.

By: **Kiril Anastasov L1087591@live.tees.ac.uk**

Task3: SQL Server Database Diagram:

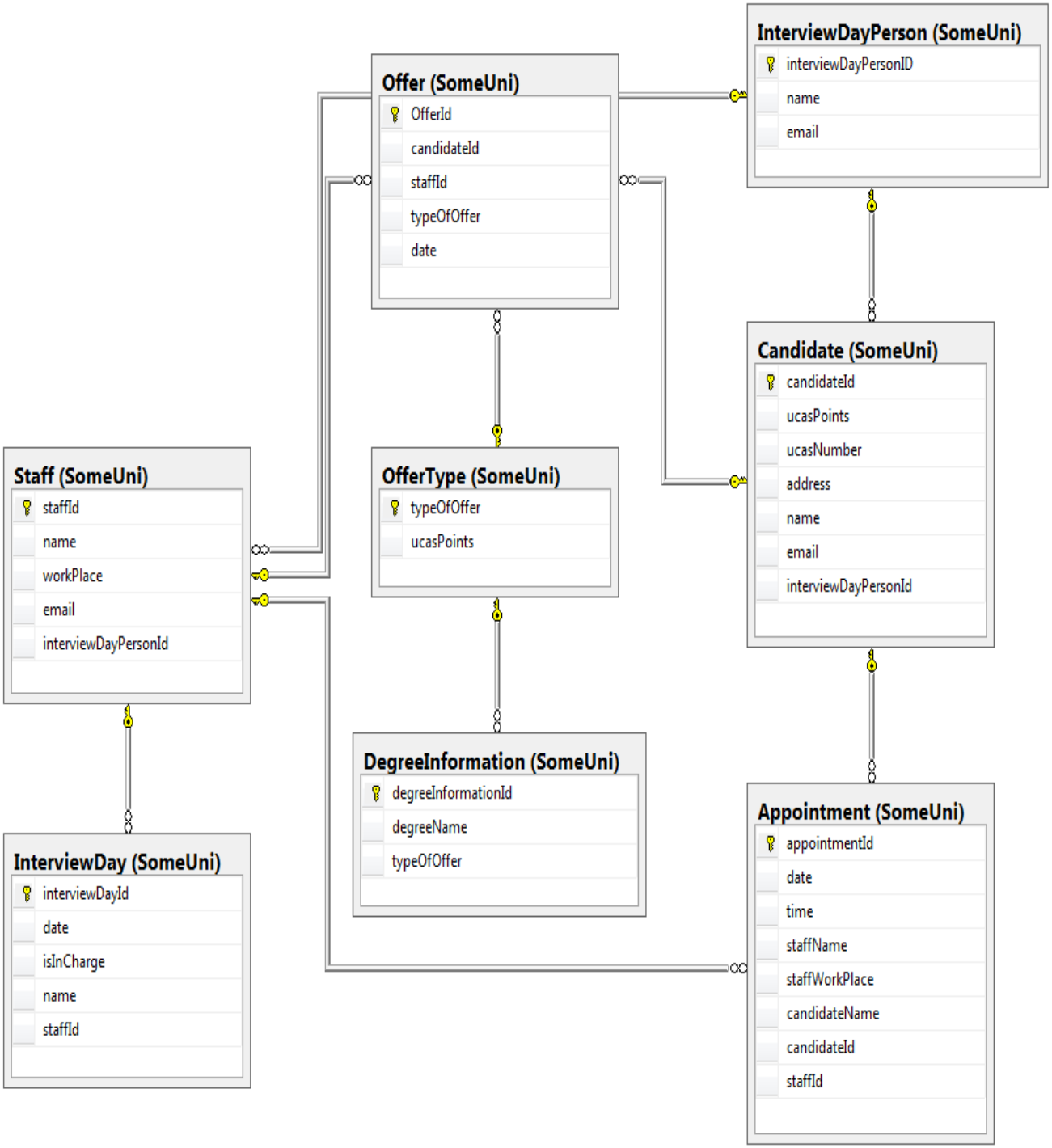


Figure 3: Database diagram

In Figure 3 we can see the implemented SQLServer database tables that correspond to the model from Task 2 (Figure 2 Full Data Model)

By: Kiril Anastasov L1087591@live.tees.ac.uk

These two associated tables were selected: Staff and Appointment in the UML diagram.

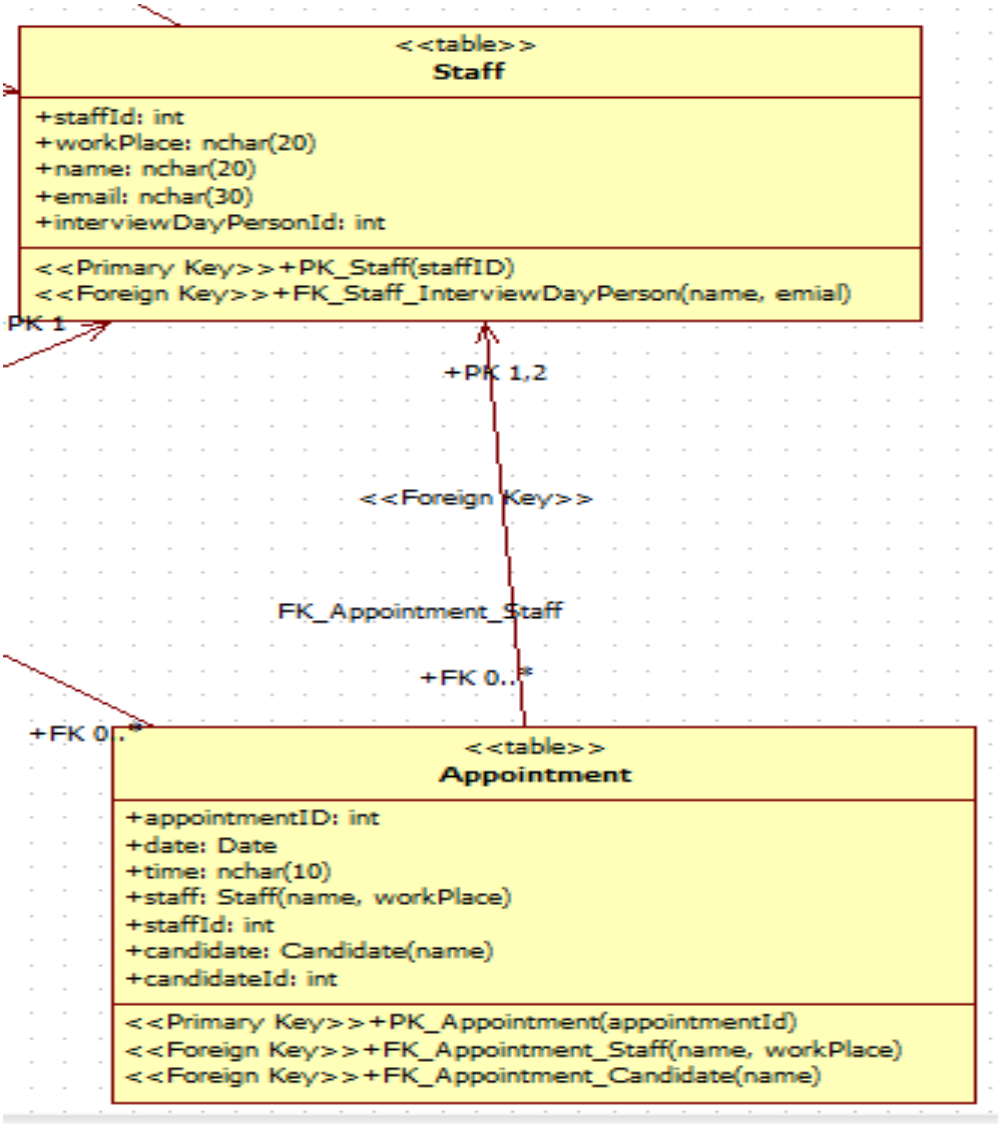


Figure 4: UML associated tables

In the diagram we have the names of the tables, all the attributes and the Primary Foreign Key relationships. By using the UML diagram for these two tables we were able to create the SQL Database Staff and Appointment tables as it can be seen on the next page in Figure 5.

By: Kiril Anastasov L1087591@live.tees.ac.uk

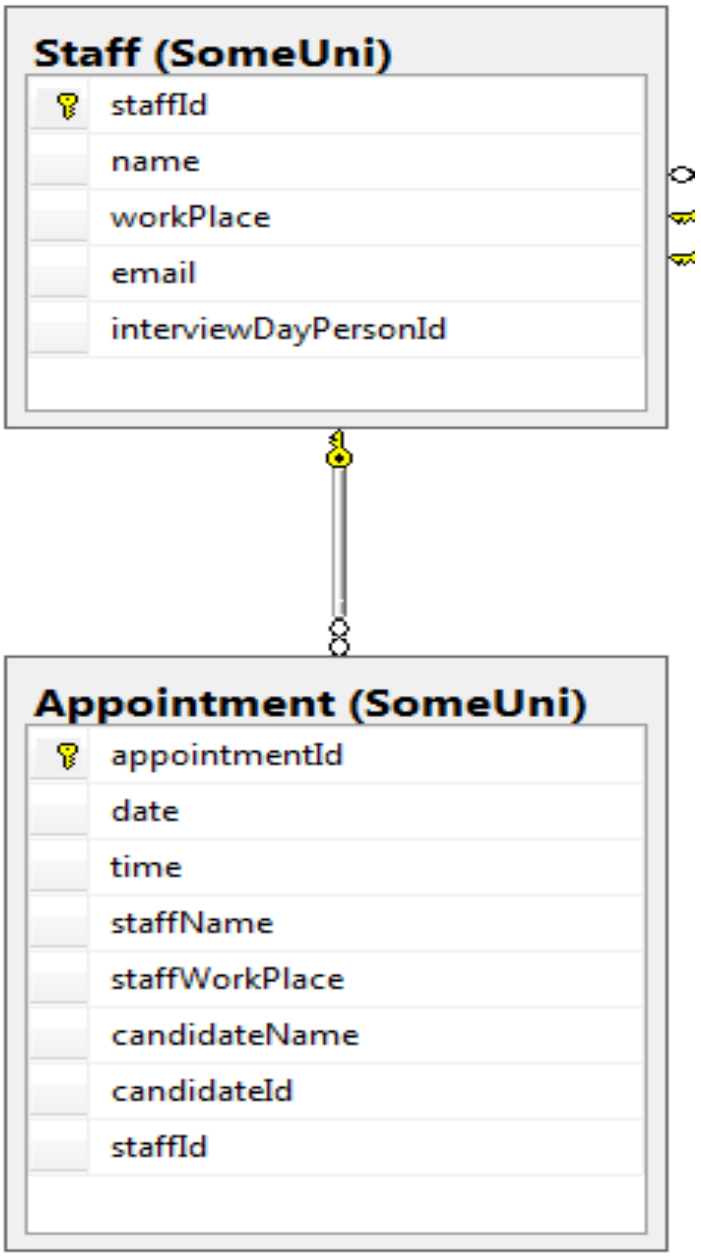


Figure 5: SQL associated tables

In Figure 5 we can see the two associated tables that correspond to the same tables from Figure 4 with the same variables, primary and foreign keys.

By: Kiril Anastasov L1087591@live.tees.ac.uk

Data Definition Language code (**T-SQL**) listing for **Appointment** and **Staff** tables.

```
USE [l1087591]
GO
/****** Object:  Table [SomeUni].[Appointment]    Script Date: 07/01/2014 12:46:31
******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [SomeUni].[Appointment](
	[appointmentId] [int] IDENTITY(1,1) NOT NULL,
	[date] [date] NOT NULL,
	[time] [nchar](10) NOT NULL,
	[staffName] [nchar](20) NOT NULL,
	[staffWorkPlace] [nchar](20) NOT NULL,
	[candidateName] [nchar](20) NOT NULL,
	[candidateId] [int] NOT NULL,
	[staffId] [int] NOT NULL,
 CONSTRAINT [PK_Appointment_1] PRIMARY KEY CLUSTERED
(
	[appointmentId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/****** Object:  Table [SomeUni].[Staff]    Script Date: 07/01/2014 12:46:31 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [SomeUni].[Staff](
	[staffId] [int] IDENTITY(1,1) NOT NULL,
	[name] [nchar](20) NOT NULL,
	[workPlace] [nchar](20) NOT NULL,
	[email] [nchar](30) NOT NULL,
	[interviewDayPersonId] [int] NOT NULL,
 CONSTRAINT [PK_Staff_1] PRIMARY KEY CLUSTERED
(
	[staffId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
```


By: Kiril Anastasov L1087591@live.tees.ac.uk

```
) ON [PRIMARY]

GO
ALTER TABLE [SomeUni].[Appointment]  WITH CHECK ADD  CONSTRAINT
[FK_Appointment_Candidate] FOREIGN KEY([candidateId])
REFERENCES [SomeUni].[Candidate] ([candidateId])
GO
ALTER TABLE [SomeUni].[Appointment] CHECK CONSTRAINT
[FK_Appointment_Candidate]
GO
ALTER TABLE [SomeUni].[Appointment]  WITH CHECK ADD  CONSTRAINT
[FK_Appointment_Staff] FOREIGN KEY([staffId])
REFERENCES [SomeUni].[Staff] ([staffId])
GO
ALTER TABLE [SomeUni].[Appointment] CHECK CONSTRAINT
[FK_Appointment_Staff]
GO
ALTER TABLE [SomeUni].[Staff]  WITH CHECK ADD  CONSTRAINT
[FK_Staff_InterviewDayPerson] FOREIGN KEY([interviewDayPersonId])
REFERENCES [SomeUni].[InterviewDayPerson] ([interviewDayPersonID])
GO
ALTER TABLE [SomeUni].[Staff] CHECK CONSTRAINT
[FK_Staff_InterviewDayPerson]
GO
```

The most important parts of the T-SQL code:

In the beginning with: `USE [l1087591]` we specify which database to use.

```
CREATE TABLE [SomeUni].[Appointment](
        [appointmentId] [int] IDENTITY(1,1) NOT NULL,
        [date] [date] NOT NULL,
        [time] [nchar](10) NOT NULL,
        [staffName] [nchar](20) NOT NULL,
        [staffWorkPlace] [nchar](20) NOT NULL,
        [candidateName] [nchar](20) NOT NULL,
        [candidateId] [int] NOT NULL,
        [staffId] [int] NOT NULL,
 CONSTRAINT [PK_Appointment_1] PRIMARY KEY CLUSTERED
(
        [appointmentId] ASC
)       )
```


By: Kiril Anastasov L1087591@live.tees.ac.uk

- *Create table* is responsible for creating a new table with name Appointment in the SomeUni Scheme.
- Then we create the first column *appointmentId* of type int, set it to be the primary key of the table and do not allow Null as arguments.
- The second column we create is date of type date and do not allow Null as arguments.
- The third column we create is time of type nchar with maximum of 10 symbols/length and do not allow null as arguments.
- The Forth column we create is *staffName* of type nchar with maximum of 20 symbols and do not allow Null as arguments.
- The Fifth column we create is *staffWorkPlace* of type nchar with maximum of 20 symbols and do not allow Null as arguments.
- The Sixth column we create is *candidateName* of type nchar with maximum of 20 symbols and do not allow Null as arguments.
- The Seventh column name we create is *candidateId* of type int and do not allow Null as arguments. T
- The last column name we create in the Appointment table is *staffId* of type int and do not allow Null as arguments.
- After that we Constrain on the primary key *appointmentId* Clustered. Some people say that frequently allowing your primary key to be clustered is bad, but for the purpose of our database it is acceptable.

This was the most important T-SQL code for the Appointment table in the SomeUni Schema. Let's examine the code for the Staff Table.

```
CREATE TABLE [SomeUni].[Staff](
        [staffId] [int] IDENTITY(1,1) NOT NULL,
        [name] [nchar](20) NOT NULL,
        [workPlace] [nchar](20) NOT NULL,
        [email] [nchar](30) NOT NULL,
        [interviewDayPersonId] [int] NOT NULL,
 CONSTRAINT [PK_Staff_1] PRIMARY KEY CLUSTERED
(
[staffId] ASC
)       )
```


By: Kiril Anastasov L1087591@live.tees.ac.uk

- *Create table* is responsible for creating a new table with name Staff in the SomeUni Scheme.
- Then we create the first column *staffId* of type int, set it to be primary key of the Staff table and do not allow Null as its arguments.
- The second column we create is name of type nchar with maximum of 20 symbols/length table and do not allow Null as its arguments.
- The thirds column we create is workplace of type nchar with maximum of 20 symbols table and do not allow Null as its arguments.
- The forth column we create is email of type nchar with maximum of 30 symbols table and do not allow Null as its arguments.
- The last column we create in the Staff table is *interviewDayPersonId* of type int table and do not allow Null as its arguments.
- After that we Constrain on the primary key *staffId* Clustered.

```
GO

ALTER TABLE [SomeUni].[Appointment]  WITH CHECK ADD  CONSTRAINT
[FK_Appointment_Candidate] FOREIGN KEY([candidateId])

REFERENCES [SomeUni].[Candidate] ([candidateId])
```

- With the following code we alter the Appointment table and add constraint for the Appointment Candidate table relationship where *candidateId* is a foreign key in the Appointment table.

```
GO
ALTER TABLE [SomeUni].[Appointment]  WITH CHECK ADD  CONSTRAINT
[FK_Appointment_Staff] FOREIGN KEY([staffId])
REFERENCES [SomeUni].[Staff] ([staffId])
```

- With the following code we alter the Appointment table and add constraint for the Appointment Staff table relationship where *staffId* is a foreign key in the Appointment table.

By: Kiril Anastasov L1087591@live.tees.ac.uk

Task 4: DML code listing:

```
USE [l1087591]
GO

/****** Object:  StoredProcedure [dbo].[parseTree]    Script Date: 11/01/2014 13:17:27
******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE procedure [dbo].[parseTree](@candidateName nchar(20))
as
select c.name,c.ucasPoints, o.typeOfOffer, d.degreeName
from SomeUni.Candidate as c
join SomeUni.Offer as o
on c.candidateId = o.candidateId
join SomeUni.DegreeInformation as d
on o.typeOfOffer = d.typeOfOffer
where c.name = @candidateName
GO
```


By: Kiril Anastasov L1087591@live.tees.ac.uk

Executing the store procedure parseTree

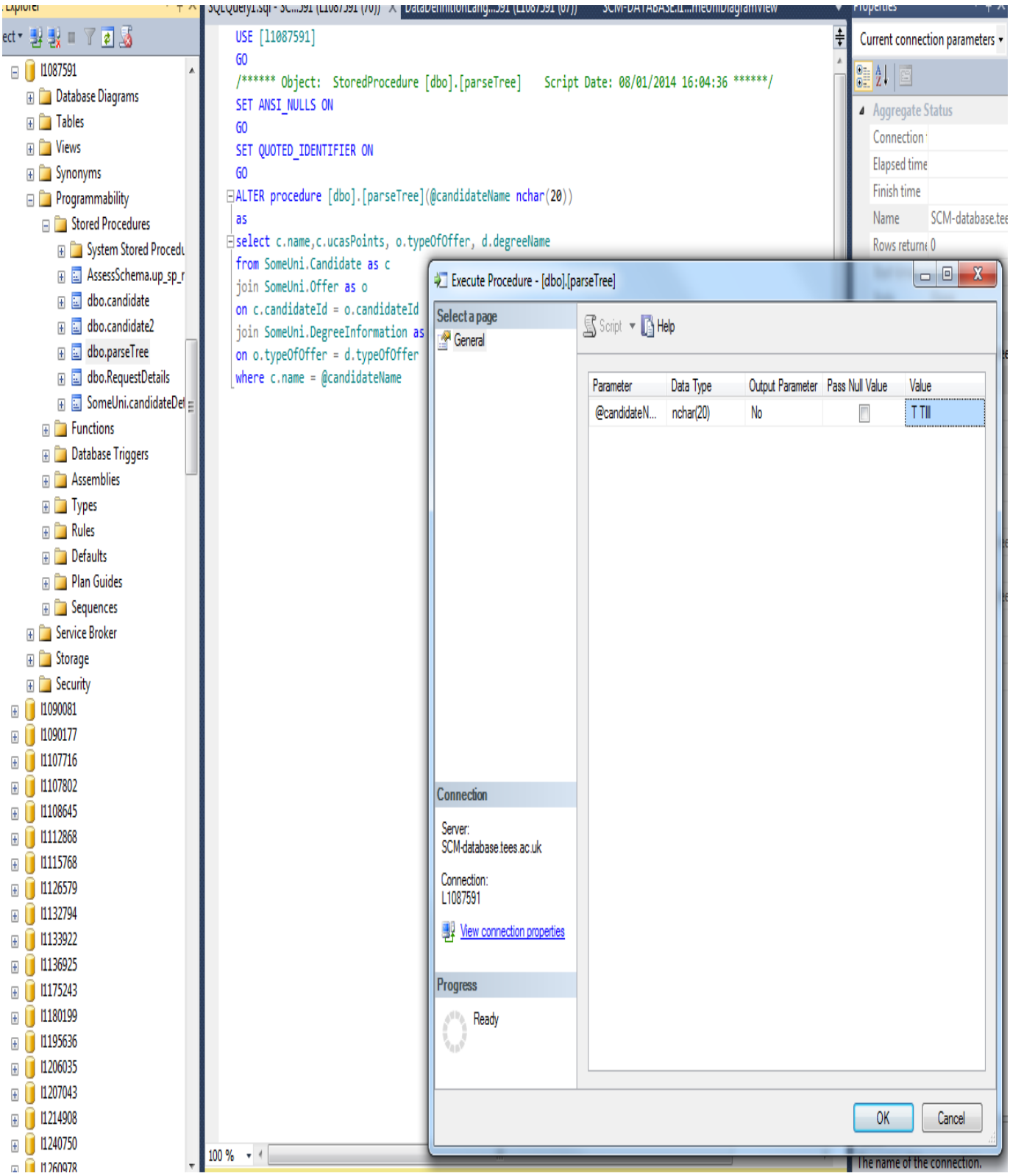


Figure 6: Executing store procedure.

By: Kiril Anastasov L1087591@live.tees.ac.uk

Result from the execution of the store procedure.

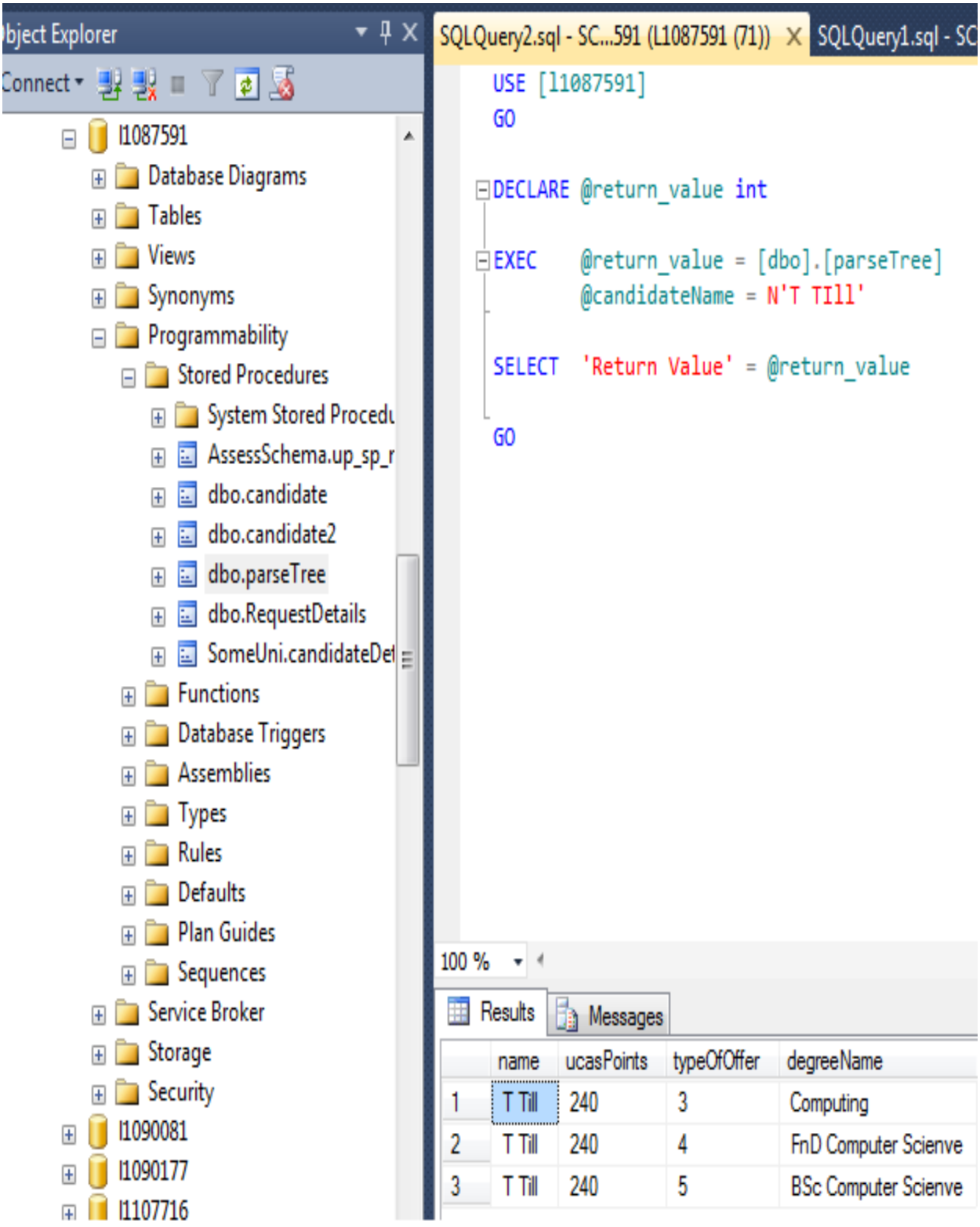


Figure 7: Result of the stored procedure.

By: Kiril Anastasov L1087591@live.tees.ac.uk

In Figure 8 we can see an alternative way to execute the store procedure: Using exec

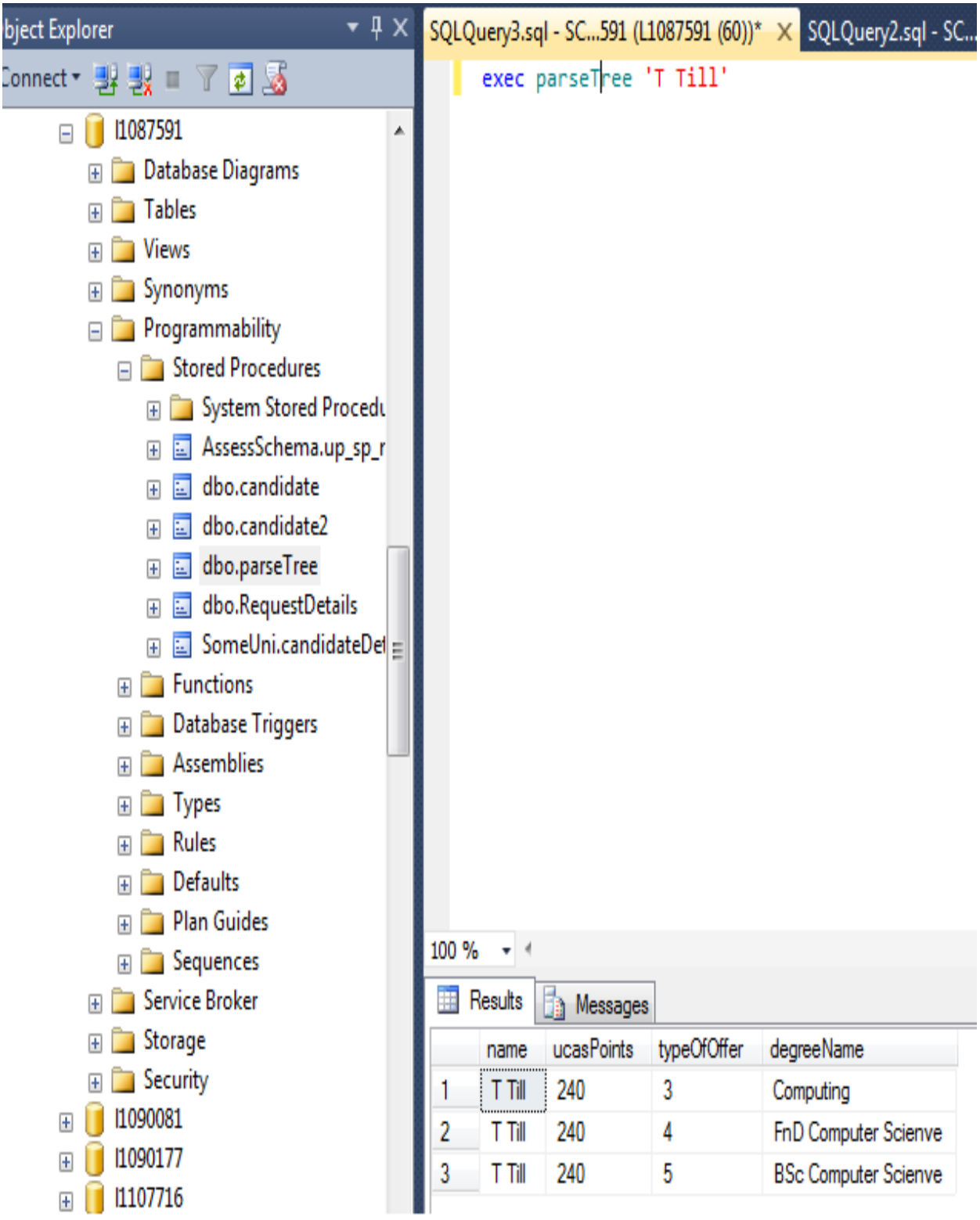


Figure 8: Alternative way of executing the store procedure

By: Kiril Anastasov L1087591@live.tees.ac.uk

Parse Tree: Optimised solutions for the Parse Tree.

It starts from the Candidate and goes up to SQL result**.**

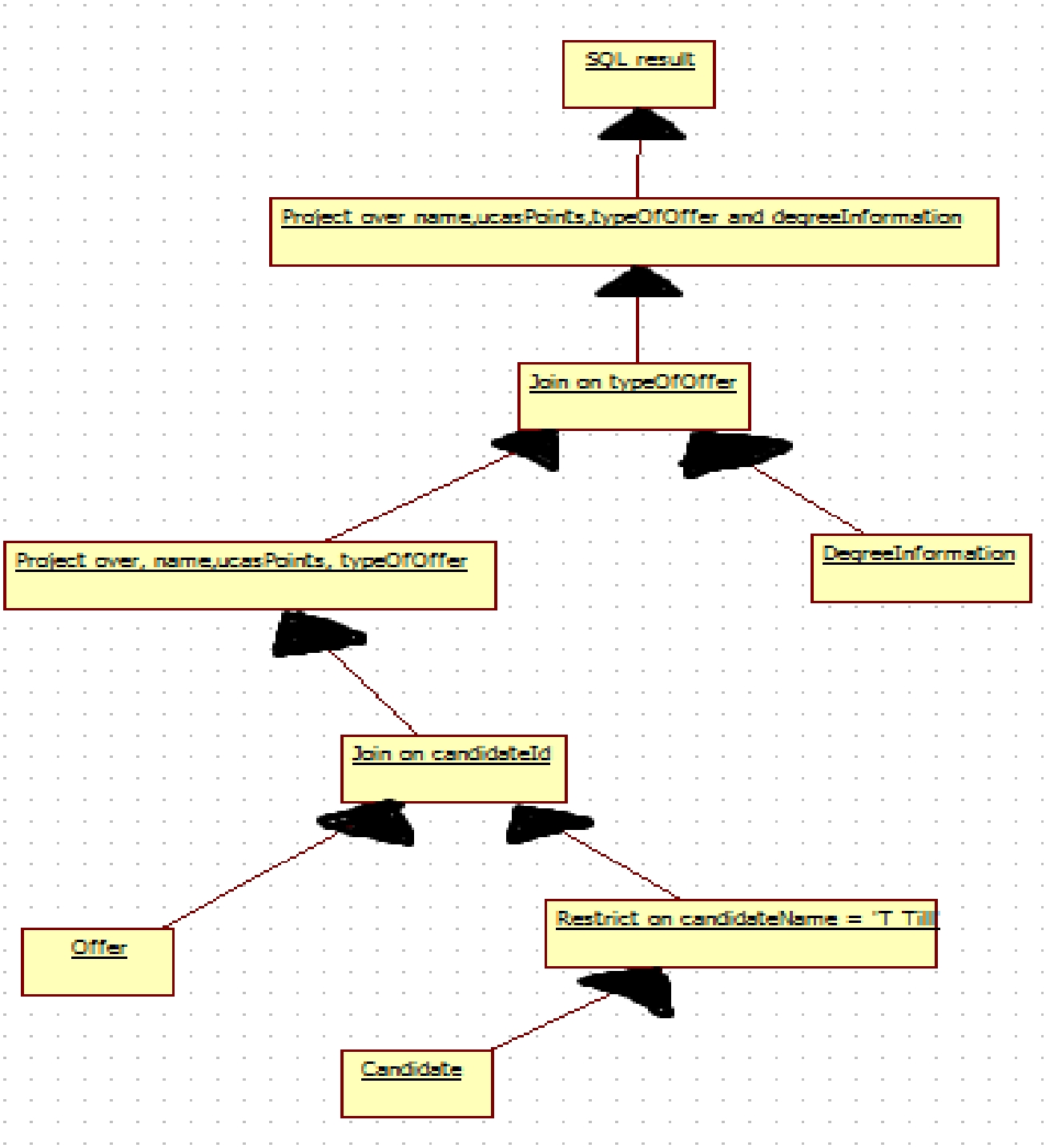


Figure 9: Optimised parse tree

By: Kiril Anastasov L1087591@live.tees.ac.uk

Task 5

Here is the code for the manipulating the staff table.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SomeUniApp
{


/**Program is responsible for creating a drop down menu
 *and linking the database with the entity framework so
 *that it can be modified through c# app
 **/
    class Program
    {
        static void Main(string[] args)
        {
            dropMenu();
        }

        /**dropMenu() is responsible for queries in the student db from the entity data
model
        *
        */
        private static void dropMenu()
        {

            bool more = true;
            do
            {
                Console.WriteLine("--------------------------------");
                Console.WriteLine("Select an Option");
                Console.WriteLine("--------------------------------");
                Console.WriteLine("1 add staff");
                Console.WriteLine("2 edit staff");
                Console.WriteLine("3 delete staff");
                Console.WriteLine("0 Exit");
                Console.WriteLine("--------------------------------\n");
                ConsoleKeyInfo key = Console.ReadKey();
```


By: Kiril Anastasov L1087591@live.tees.ac.uk

```
            switch (key.KeyChar.ToString())
            {
                case "1":
                    insertStaff(setName(), setWorkPlace(), setEmail(), setId());
                    break;
                case "2":
                    editStaff(setName());
                    break;
                case "3":
                    deleteStaff(setName());
                    break;
                case "0":
                    more = false;
                    break;
                default:
                    //more = false;
                    Console.WriteLine("Invalid choise please chose between 0 - 3 ");
                    break;
            }

        } while (more);

    }

    /**setName is responsible to set the name of the Staff
     * */
    private static String setName()
    {
        Console.WriteLine("Enter name: \n");
        String myName = Console.ReadLine();
        return myName ;
    }

    /**setWorkPlace is responsible to set the work place of the Staff
    * */
    private static String setWorkPlace()
    {
        Console.WriteLine("Enter Workplace: \n");
        String myWorkPlace = Console.ReadLine();
        return myWorkPlace;
    }

    /**setEmail is responsible to set the email of the Staff
    * */
    private static String setEmail()
    {
```


By: Kiril Anastasov L1087591@live.tees.ac.uk

```
            Console.WriteLine("Enter Email: \n");
            String myEmail = Console.ReadLine();
            return myEmail;
        }

        /**setId is responsible to set the id of the Staff
        * */
        private static String setId()
        {
            Console.WriteLine("Enter Id: \n");
            String id = Console.ReadLine();
            return id;
        }



        /**insertStaff is responsible for creating a new staff in the db, inserting it and
saving it.
         * @param myStaffName is the name of the staff.
         * @param myWorkPlace is the where the staff works from.
         * @param myEmail is the email of the staff.
         * @param id is the id of the staff.
         */
        private static void insertStaff(String myStaffName, String myWorkPlace, String
myEmail, String id)
        {
            using (var context = new l1087591Entities2())
            {
                //create a staff object
                Staff staff = new Staff();

                staff.name = myStaffName;
                staff.workPlace = myWorkPlace;
                staff.email = myEmail;
                staff.interviewDayPersonId = Convert.ToInt32(id);
                //add object to container's result entity set and save, catch exceptions
                try
                {
                    context.Staffs.AddObject(staff);
                    context.SaveChanges();
                    Console.WriteLine(" New staff " + myStaffName + " added ... ");
                }
                catch (Exception ex)
                {
                    Console.WriteLine("error: " + ex.Message);
                }
```


Kiril Anastasov L1087591@live.tees.ac.uk

```
    }
}


/**editStaff is responsible for editing the staff details  in the db.
 * @param myStaffName is the name of the staff.
 * @param myWorkPlace is the where the staff works from.
 * @param myEmail is the email of the staff.
 * @param id is the id of the staff.
 */
private static void editStaff(String myStaffName)
{
    using (var context = new l1087591Entities2())
    {
        //create a staff object
        Staff staff = new Staff();

        staff.name = myStaffName;
        //add object to container's result entity set and save, catch exceptions
        try
        {
            //get the row containing the given primary key
            var myStaff = (from m in context.Staffs
                           where m.name == myStaffName
                           select m).First();
            //get user input -see function above
            string newName = setName();
            string newWorkPlace = setWorkPlace();
            string newEmail = setEmail();
            string newId = setId();
            //change module entity and save
            myStaff.name = newName;
            myStaff.workPlace = newWorkPlace;
            myStaff.email = newEmail;
            myStaff.staffId = Convert.ToInt32(newId);
            // context.Modules.
            //context.Modules.DeleteObject(module);
            context.SaveChanges();
            Console.WriteLine("... editing staff " + myStaffName + " done");
        }
        catch (Exception ex)
        {
            Console.WriteLine("error: " + ex.Message);
        }
```

```
        }
    }

    /*deleteStaff is responsible for deleting a staff in the db by name
    * @param myStaffName is the name of the staff.
    */
    private static void deleteStaff(String myStaffName)
    {
        using (var context = new l1087591Entities2())
        {
            //create a staff object
            Staff staff = new Staff();

            staff.name = myStaffName;
            //add object to container's result entity set and save, catch exceptions
            try
            {
                //get the row containing the given name
                var myStaff = (from m in context.Staffs
                               where m.name == myStaffName
                               select m).First();
                // context.Modules.
                context.Staffs.DeleteObject(myStaff);
                context.SaveChanges();
                Console.WriteLine("The staff " + myStaffName + " has been deleted ");
            }
            catch (Exception ex)
            {
                Console.WriteLine("error: " + ex.Message);
            }
        }
    }
}
```


By: Kiril Anastasov L1087591@live.tees.ac.uk

In Figure 10 we can see the result of the application and getting the option menu.

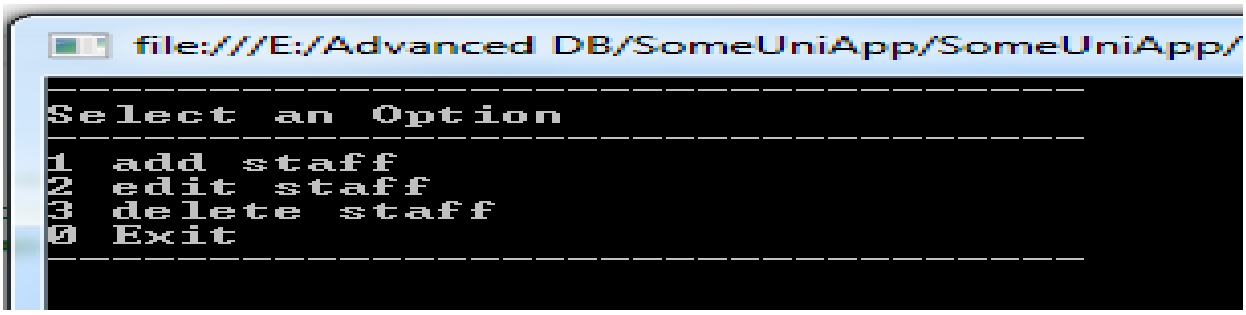


Figure 10: Select an option

If you enter a wrong Option number you will get this result due to the validation.

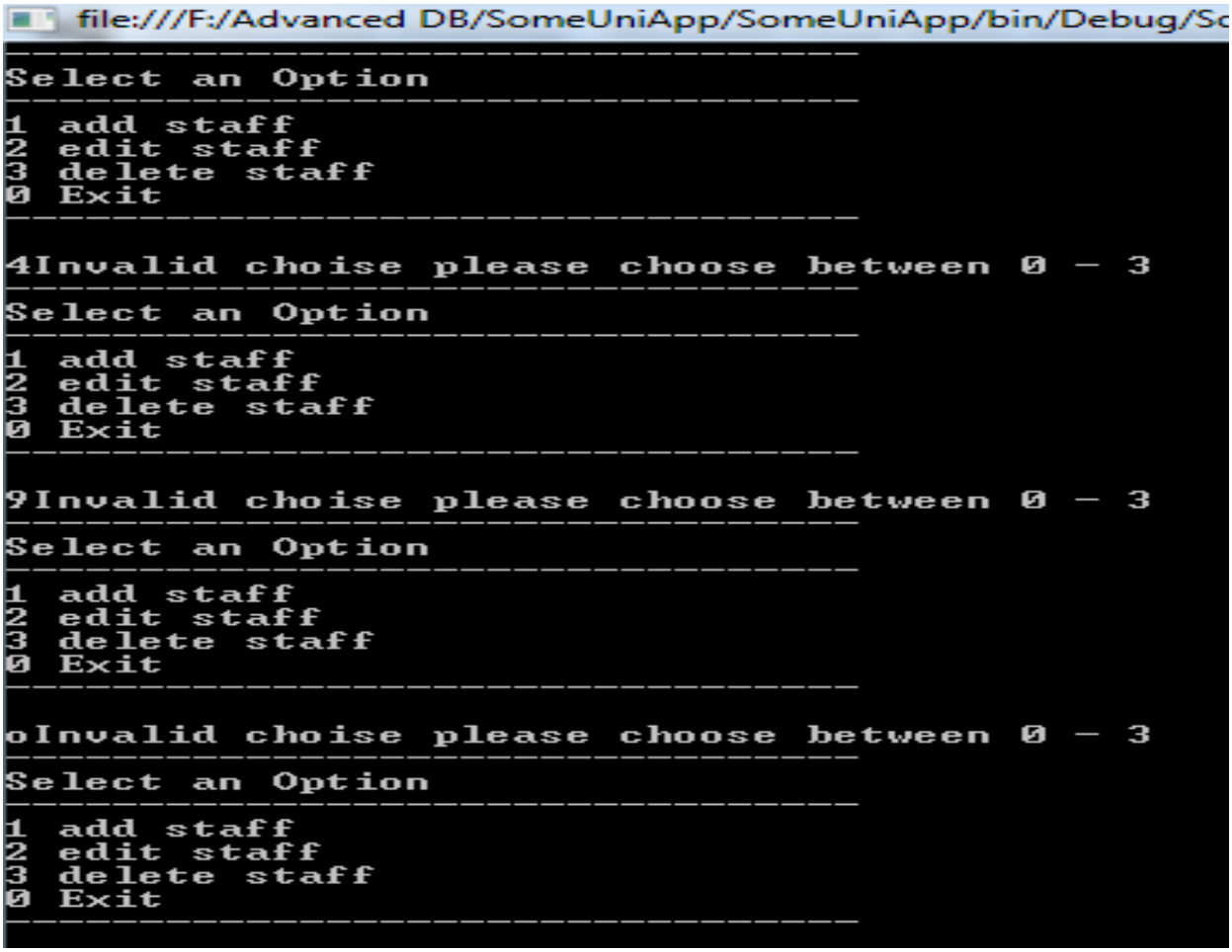


Figure 11: Validation

By: Kiril Anastasov L1087591@live.tees.ac.uk

In Figure 12 we can see what happens when option 1 add staff is selected: you will need to enter the staff name, workplace, email, id .

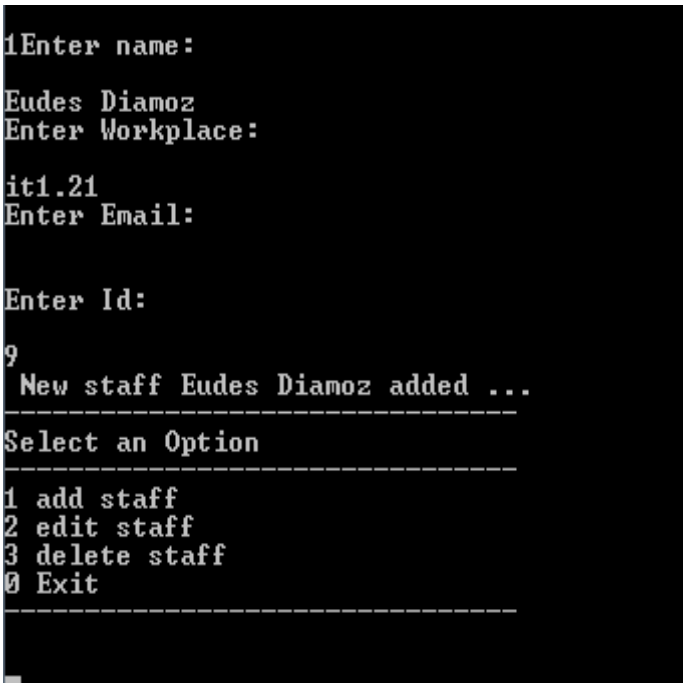


Figure 12: New Staff

In Figure 13 we can see that the new staff was added successfully in SQL.

Results | Messages

	staffId	name	workPlace	email	interviewDayPersonId
1	1	A Smiths	it05	asmith@gmail.com	1
2	2	B Jones	it06	bjones@gmail.com	2
3	3	C White	it07	cwhite@gmail.com	3
4	4	D Brown	it08	dbrown@gmail.com	4
5	5	E Black	g05	eblack@yahoo.co.uk	5
6	6	F Smith	g05	fsmith@yahoo.co.uk	6
7	7	G Green	g06	ggreen@yahoo.co.uk	7
8	16	K Anastasov	it1.15	kanastasov@gmail.com	15
9	17	Eudes Diamoz	it1.21		9

Figure 13: New staff in SQL

By: **Kiril Anastasov L1087591@live.tees.ac.uk**

In Figure 14 we can see the edit staff and what happens when we chose the second option to edit a staff from the menu.

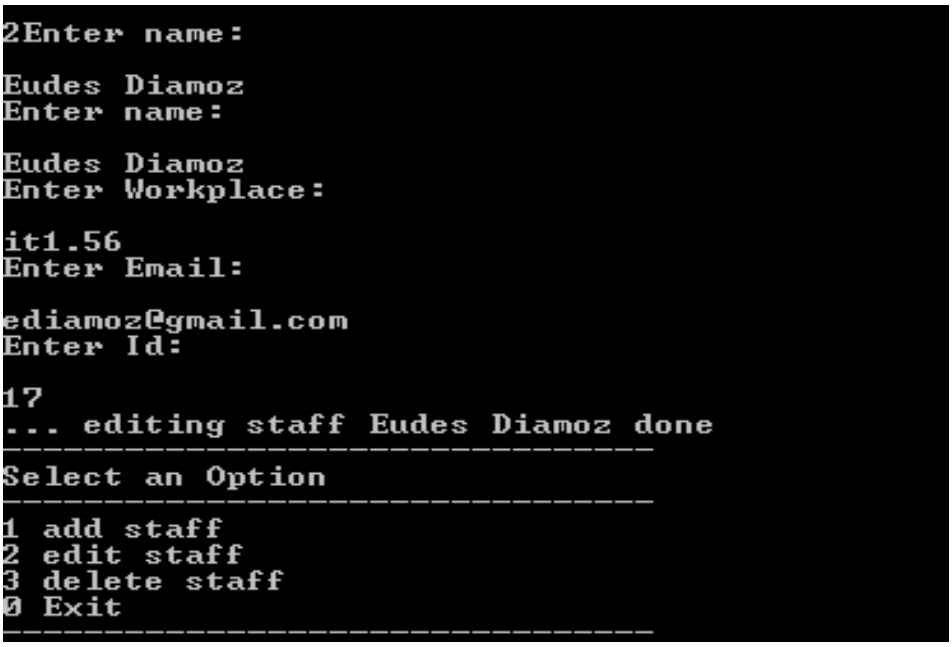
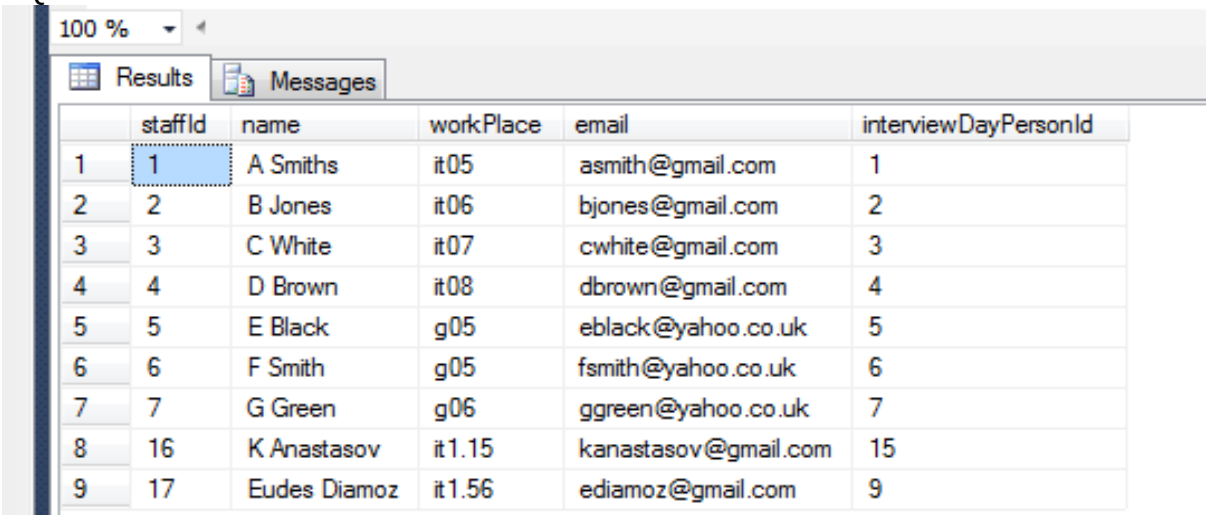


Figure14: Editing staff

You will need to enter the name of the staff you need to change than if you want to change the name enter different name in our case we enter the same name. After that we change the workplace and put an email for the staff.

In Figure 15 we can see that the changes that we made have applied to the SQL.



	staffId	name	workPlace	email	interviewDayPersonId
1	1	A Smiths	it05	asmith@gmail.com	1
2	2	B Jones	it06	bjones@gmail.com	2
3	3	C White	it07	cwhite@gmail.com	3
4	4	D Brown	it08	dbrown@gmail.com	4
5	5	E Black	g05	eblack@yahoo.co.uk	5
6	6	F Smith	g05	fsmith@yahoo.co.uk	6
7	7	G Green	g06	ggreen@yahoo.co.uk	7
8	16	K Anastasov	it1.15	kanastasov@gmail.com	15
9	17	Eudes Diamoz	it1.56	ediamoz@gmail.com	9

Figure 15: Edited staff in SQL

By: Kiril Anastasov L1087591@live.tees.ac.uk

In Figure 16 we can see what happens when we chose the third option from the menu to delete a staff we will need to enter the name of the staff we want to delete.

Figure 16: Delete staff

In Figure 17 we can see that the staff has been deleted from the SQL.

	staffId	name	workPlace	email	interviewDayPersonId
1	1	A Smiths	it05	asmith@gmail.com	1
2	2	B Jones	it06	bjones@gmail.com	2
3	3	C White	it07	cwhite@gmail.com	3
4	4	D Brown	it08	dbrown@gmail.com	4
5	5	E Black	g05	eblack@yahoo.co.uk	5
6	6	F Smith	g05	fsmith@yahoo.co.uk	6
7	7	G Green	g06	ggreen@yahoo.co.uk	7
8	16	K Anastasov	it1.15	kanastasov@gmail.com	15

Figure 17: Deleted staff in SQL

By: Kiril Anastasov L1087591@live.tees.ac.uk

In Figure 18 we have made another demonstration for deleting a staff.

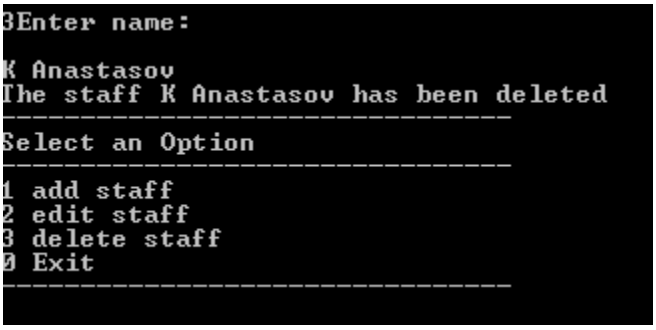


Figure 18: Delete staff

In Figure 19.1 we can see that the staff has been deleted from the SQL.

	staffId	name	workPlace	email	interviewDayPersonId
1	1	A Smiths	it05	asmith@gmail.com	1
2	2	B Jones	it06	bjones@gmail.com	2
3	3	C White	it07	cwhite@gmail.com	3
4	4	D Brown	it08	dbrown@gmail.com	4
5	5	E Black	g05	eblack@yahoo.co.uk	5
6	6	F Smith	g05	fsmith@yahoo.co.uk	6
7	7	G Green	g06	ggreen@yahoo.co.uk	7

Figure 19.1: Deleted staff in SQL

Based on the C# source code and all the examples it can be concluded that the add, edit and delete menu features are working as expected.

By: Kiril Anastasov L1087591@live.tees.ac.uk

Identify a potential problem existing in database design for task5 and possible solution.

- When we add new staff we can we can add a new staff with the same name, workplace and email. Even though it is possible for two people to work in the same University with the same name and workplace it will be better to make a query in the database and match the name if it is the same display a message with error like: “The name already exists in the database, we cannot create a new staff with the same name”. If by any chance this is the case in SomeUni then the administrator can add the staff with the same name.

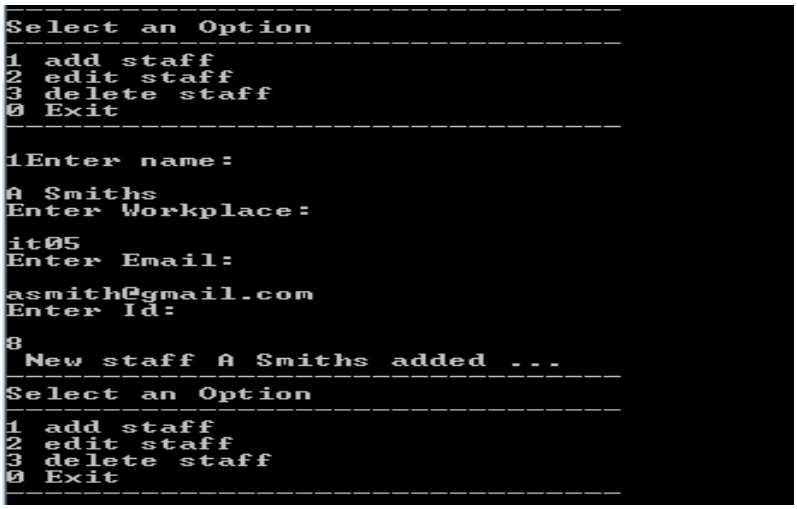


Figure 19.2: New staff

- When we deleted a staff the main bad consequence is that there are dependent records in other tables (Offer, InterviewDay, and Appointment) whose referential integrity is lost when the parent record goes away. There a few ways you can deal with the referential integrity. The first way is to use cascading deletions with care. With this method you can eliminate most referential integrity problems by carefully controlling the update process. The second way to control update anomalies if you do not want to cascade a deletion. Instead change the child’s table’s foreign key to a Null value.

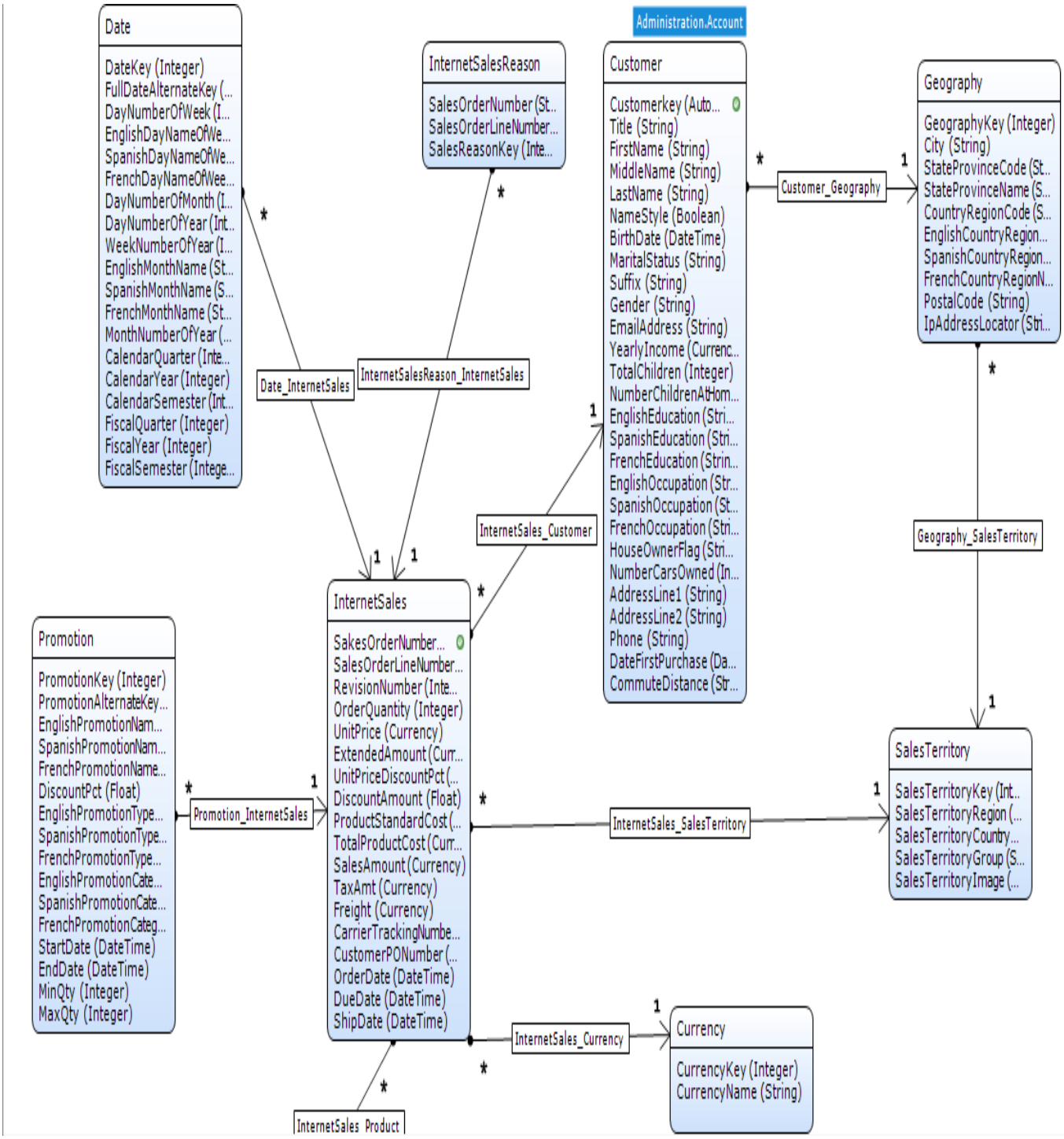
- When we delete a staff, then there is no record of the staff. This could be a problem, if for example the police are looking for him or anyone from SomeUni wants to contact him. We can create a table called StaffWorkedInSomeUni that store the same information about Staff but when staffs is deleted from Staff table the information about the staff remains in the StaffWorkedInSomeUni.

By: Kiril Anastasov L1087591@live.tees.ac.uk

Databases Part 2

Task 6 Mendix

6.1 Mendix domain model



By: Kiril Anastasov L1087591@live.tees.ac.uk

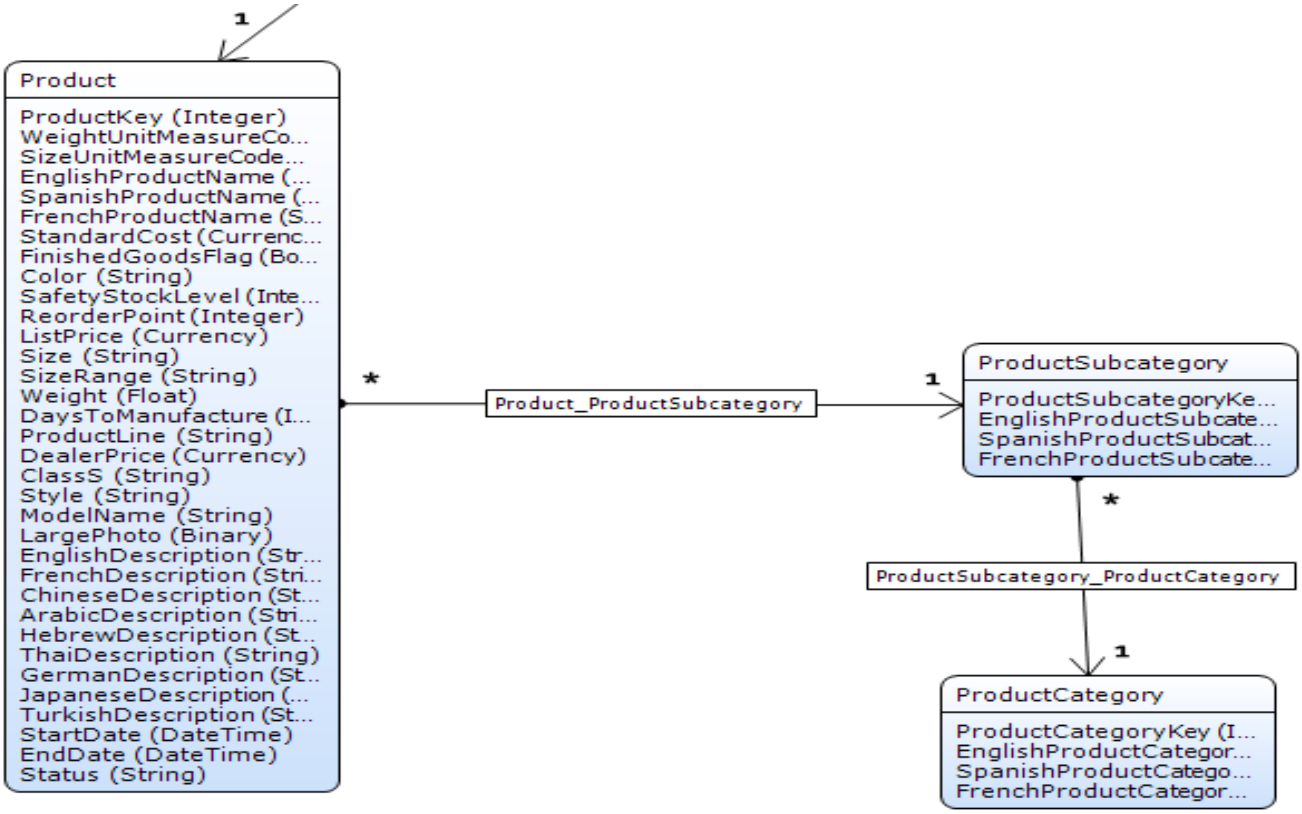


Figure 20: Mendix model

In Figure 20 we can see the Mendix model corresponds to the tables and relationships in the Internet Sales database diagram of AdwentureWorksDW2012. There are no Primary and Foreign Keys in the Mendix model because Mendix does not need a Primary and Foreign Key relationships. The attributes in the tables have similar data type. For example in Mendix attributes that are defined as String correspond to nvarchar() in SQL and Integer in Mendix correspond to all the number types in SQL like float, double and integer.

By: Kiril Anastasov L1087591@live.tees.ac.uk

6.2 Mendix BCNF

InternetSales is strictly in BCNF because there are no extra dependencies which do not involve the primary key (SalesOrderNumber and SalesOrderLine).

Currency is strictly in BCNF because there are no extra dependencies which do not involve the primary key (CurrencyKey).

Customer is strictly NOT in BCNF because there are extra dependencies which involve the primary key (CustomerKey).

Date is strictly NOT in BCNF because there are extra dependencies which involve the primary key (DateKey).

InternetSalesReason is strictly NOT in BCNF because there are extra dependencies which involve the primary key (SalesOrderNumber, SalesOrderLineNumber and SalesReasonKey).

Geography is strictly NOT in BCNF because there are extra dependencies which involve the primary key (GeographyKey).

SalesTerritory is strictly NOT in BCNF because there are extra dependencies which involve the primary key (SalesTerritoryKey).

Promotion is strictly NOT in BCNF because there are extra dependencies which involve the primary key (PromotionKey).

Product is strictly NOT in BCNF because there are extra dependencies which involve the primary key (ProductKey).

ProductSubCategory is strictly NOT in BCNF because there are extra dependencies which involve the primary key (ProductSubCategoryKey).

ProductCategory is strictly NOT in BCNF because there are extra dependencies which involve the primary key (ProductCategoryKey).

By: Kiril Anastasov L1087591@live.tees.ac.uk

6.3 Mendix Form

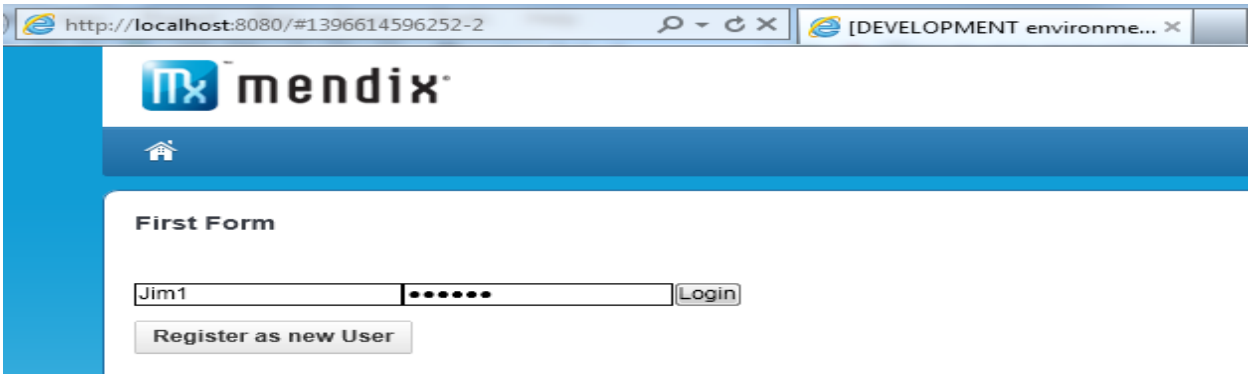


Figure 21: Mendix LogIn

In figure 21 we can see our mendix application deployed. We can Login with current user Jim1 and password 123456 or Register as a new User.

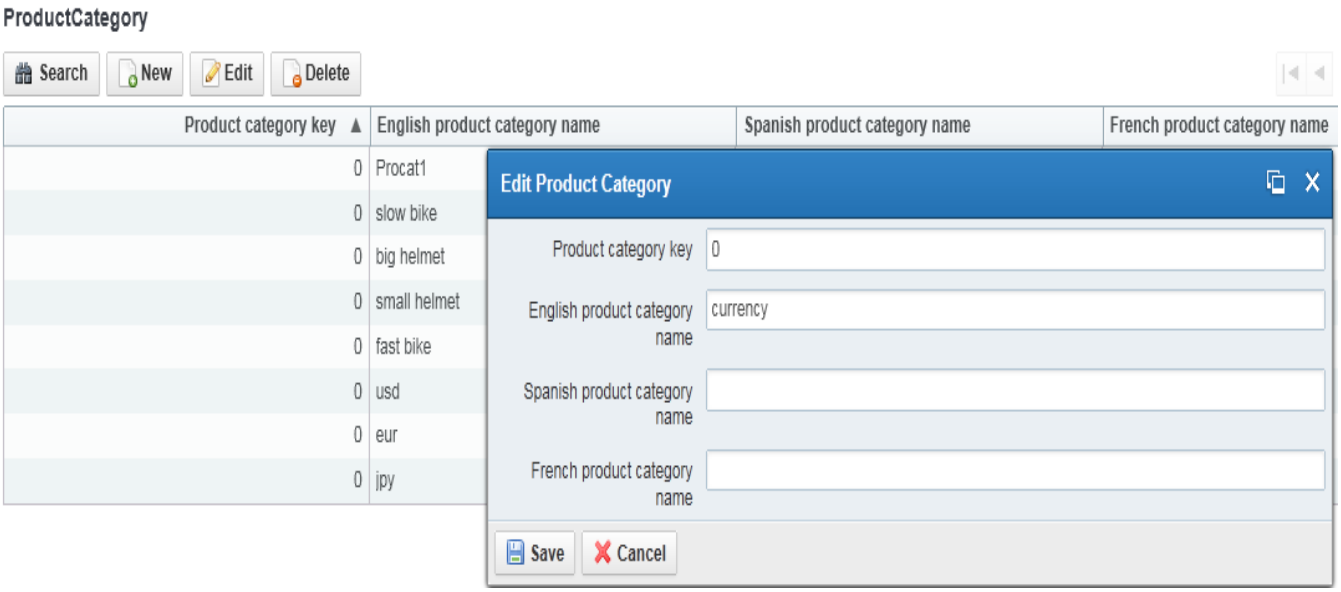


Figure 22: Product category

In Figure 22 we can create or edit a Product category; in this case we are creating a new product called currency.

By: Kiril Anastasov L1087591@live.tees.ac.uk

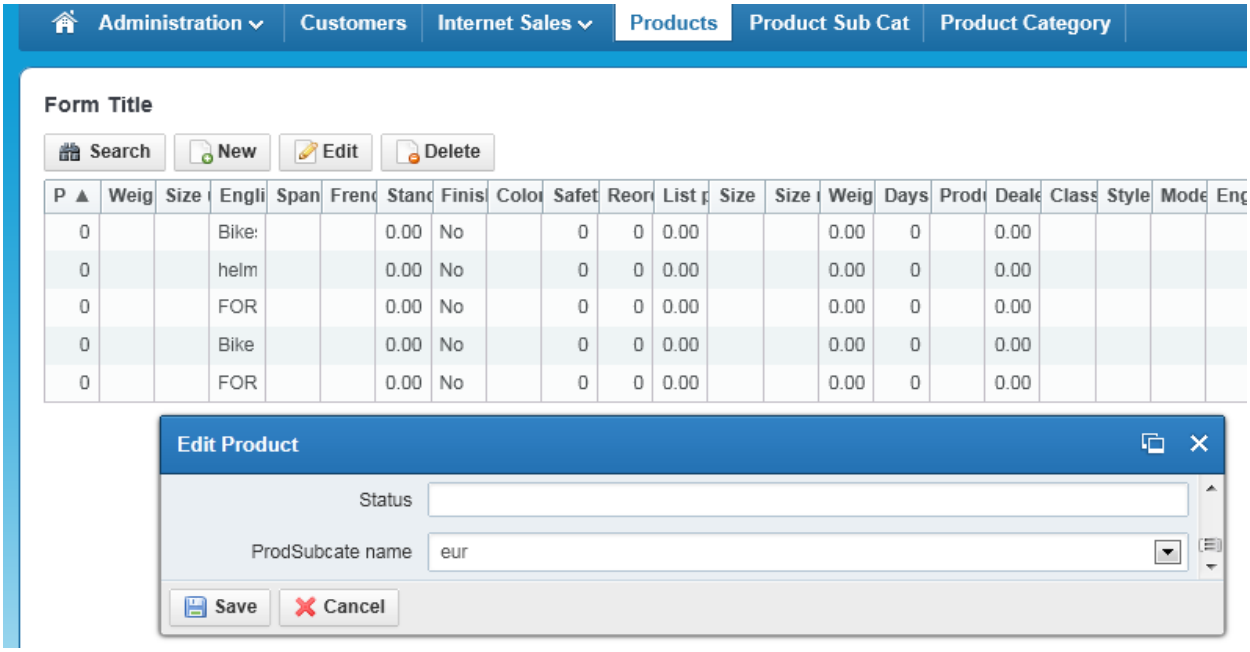


Figure 23: Products

In Figure 23 we have created new product FOREX and we have select the product sub category name euro.

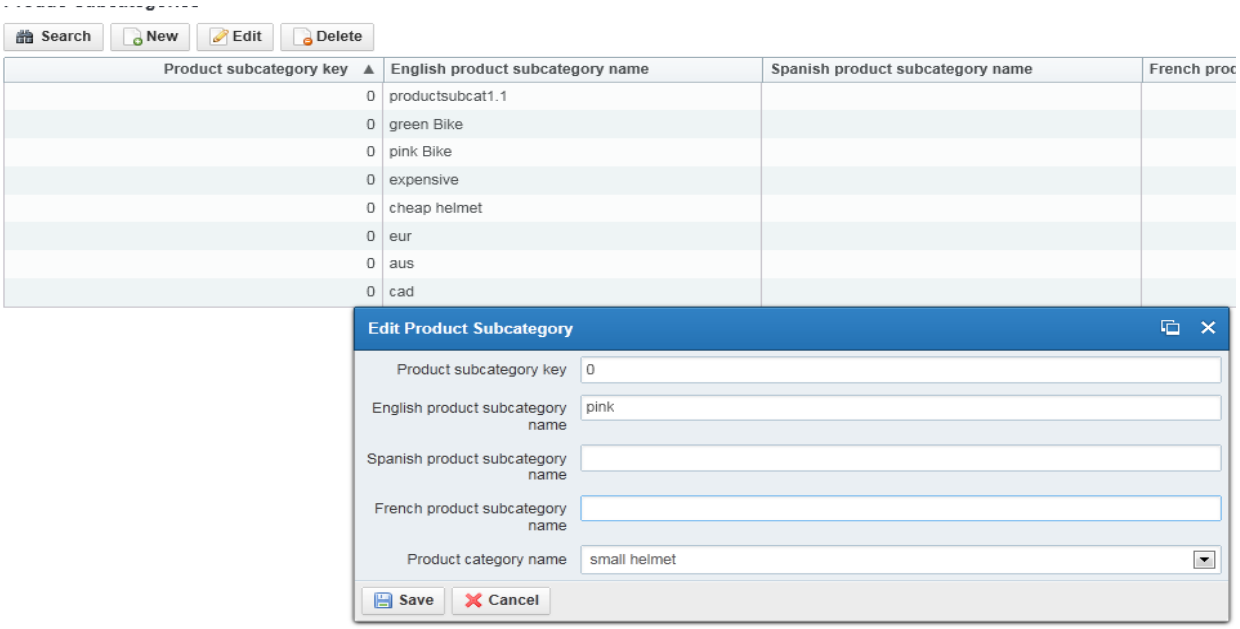


Figure 24: Product sub category

In Figure 24 we have created a product subcategory cheap helmet and have chosen a product category small helmet.

By: Kiril Anastasov L1087591@live.tees.ac.uk

Internet Sales for logged-in Customer

Search

1 to 6 of 6

Sakes order number	Sales order line number ▲	English product name	English product sub category name	English product category name
2454879	0			
987654321	0	Bikes	productsubcat1.1	Procat1
111	0	FOREX		
112	0	Bike	green Bike	fast bike
113	0	FOREX	aus	eur
A1	1	helmet		

Figure 25: Internet sales for Logged-in Customer

In Figure 25 we have a few Product names with product category and product sub categories. For example we have Bike as product category, green Bike as product sub category and fast bike as product name. Also we have FOREX as product name and aus as product sub category and eur for product category name to demonstrate that the relationship between the category, sub category and product is working as expected.

By: Kiril Anastasov L1087591@live.tees.ac.uk

Task 7 Adventure data cube

7.1 Data Cube

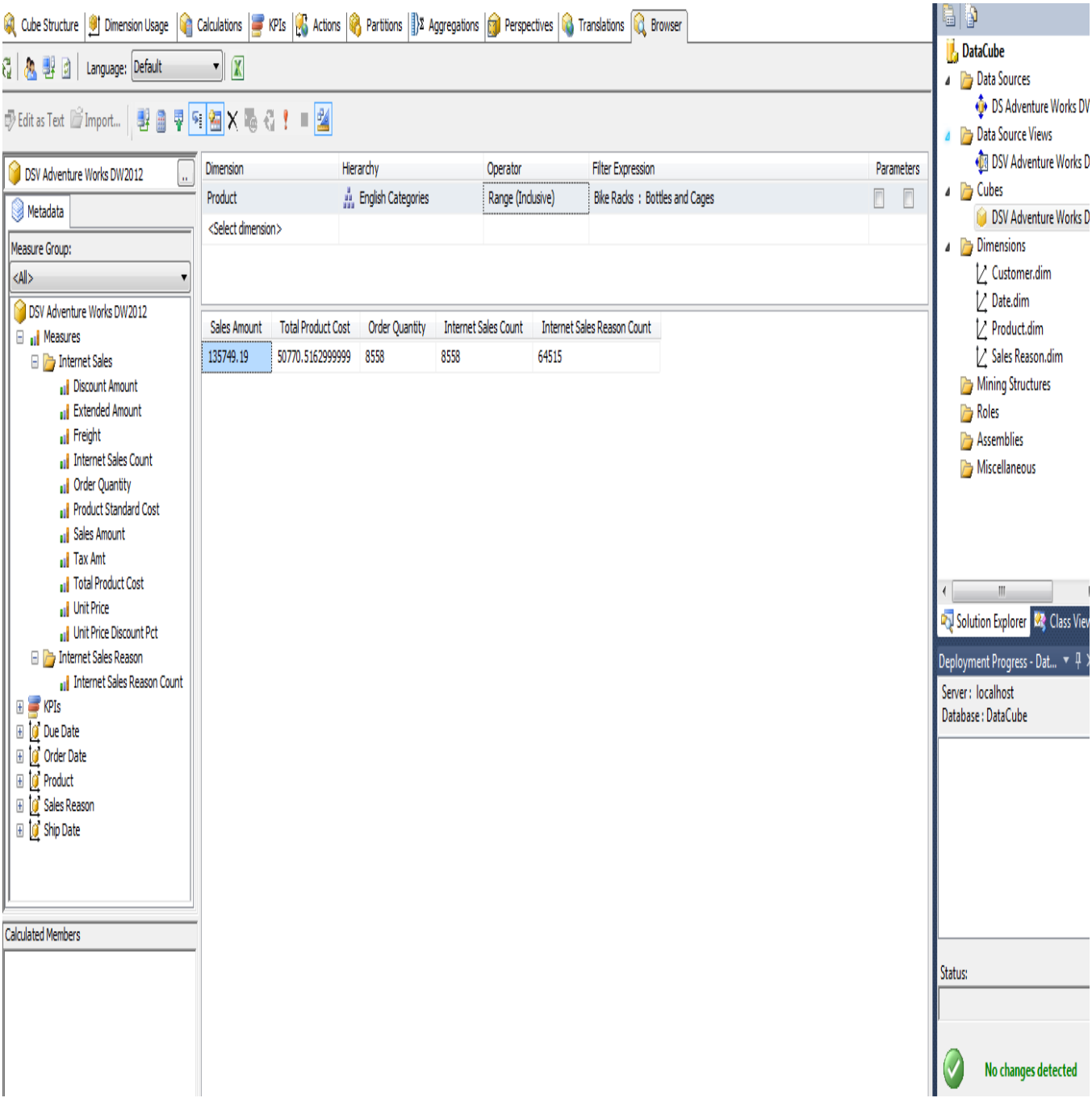


Figure 26: Browser

In figure 26 we can see the data cube that was created with the dimensions and measures.

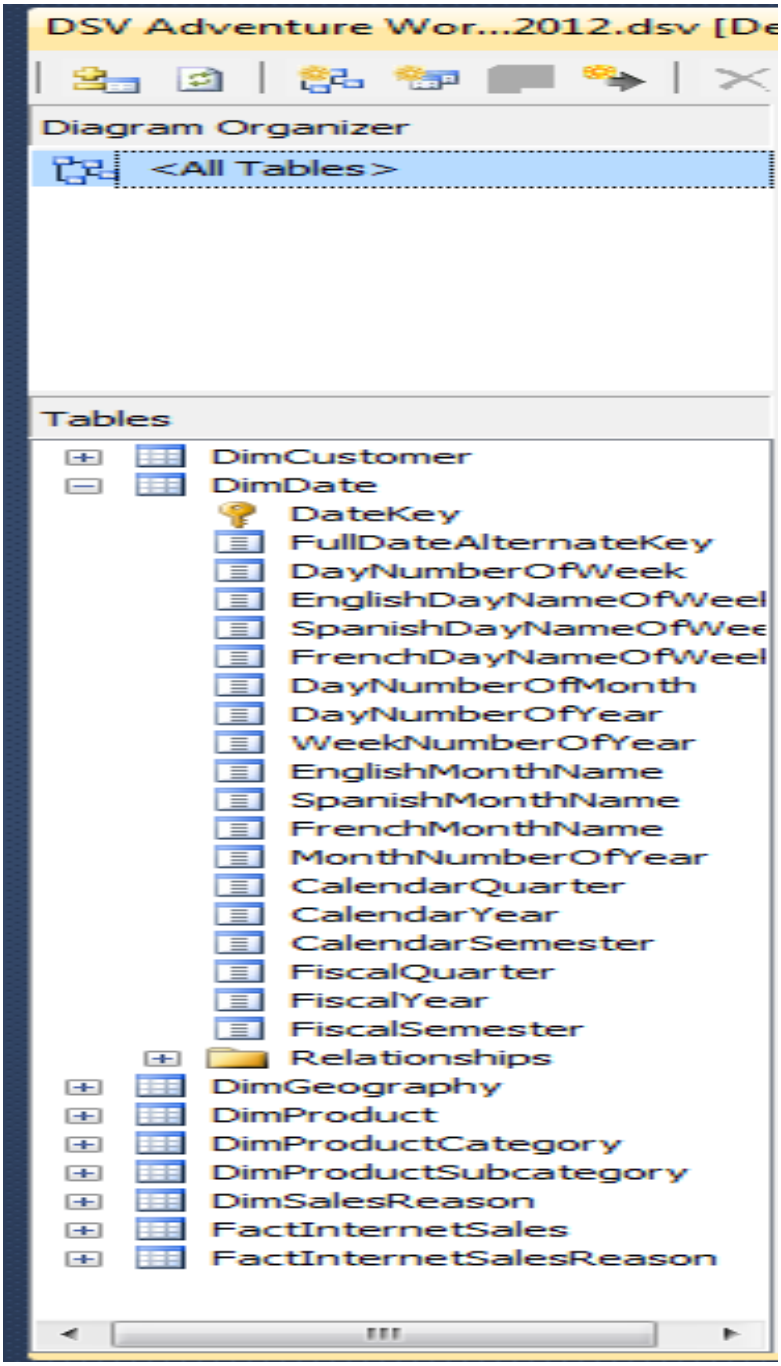


Figure 27: Tables Cube structure

The Diagram Organizer pane creates sub diagrams that allow you to view subsets of the data source view. The table pane is where the tables and their schema elements are displayed in a tree data structure view.

By: Kiril Anastasov L1087591@live.tees.ac.uk

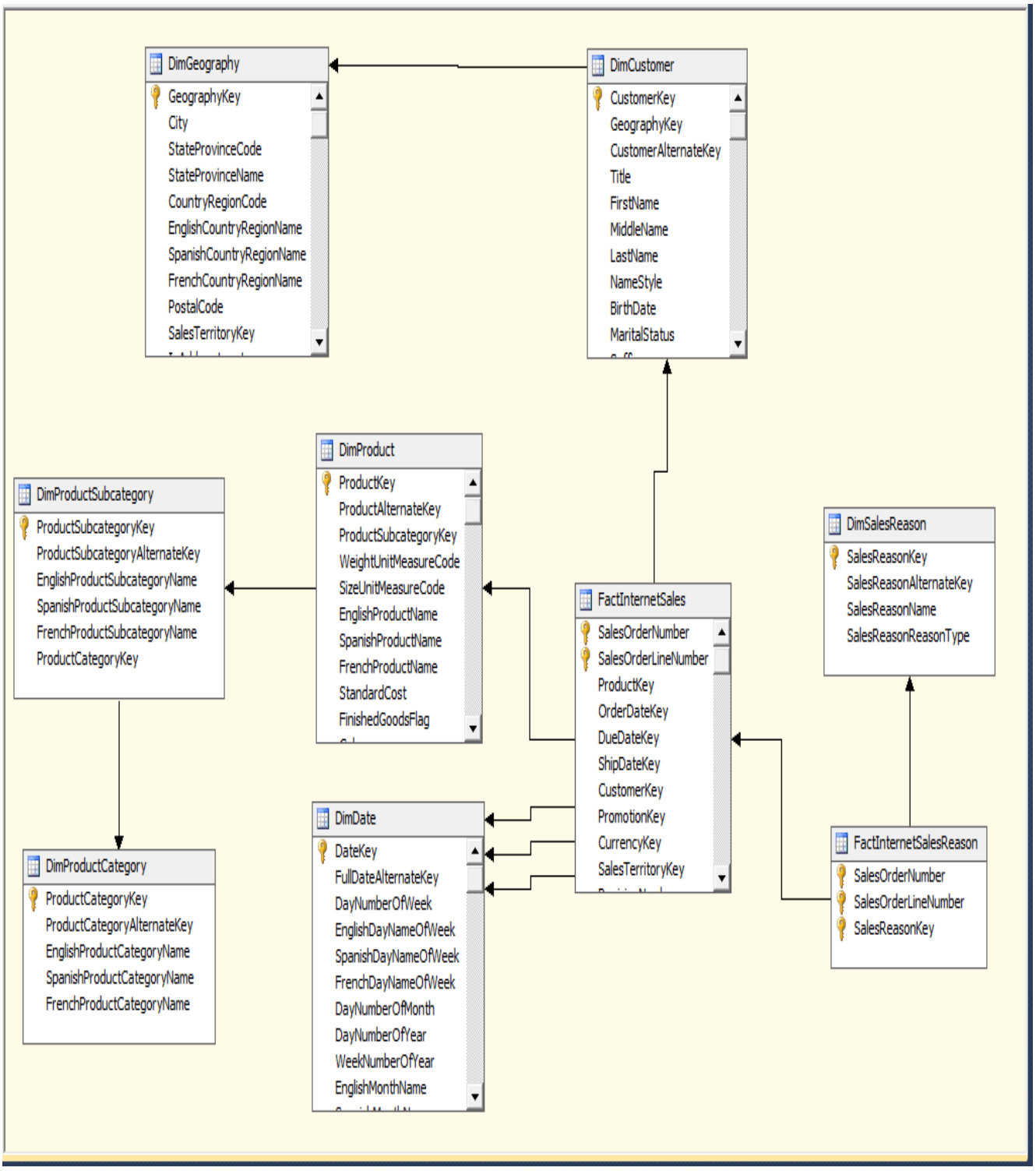


Figure 28: Cube structure

In figure 28 we see again the cube structure with tables and their attributes. The Diagram Pane is where the tables and the relationships are represented graphically.

By: Kiril Anastasov L1087591@live.tees.ac.uk

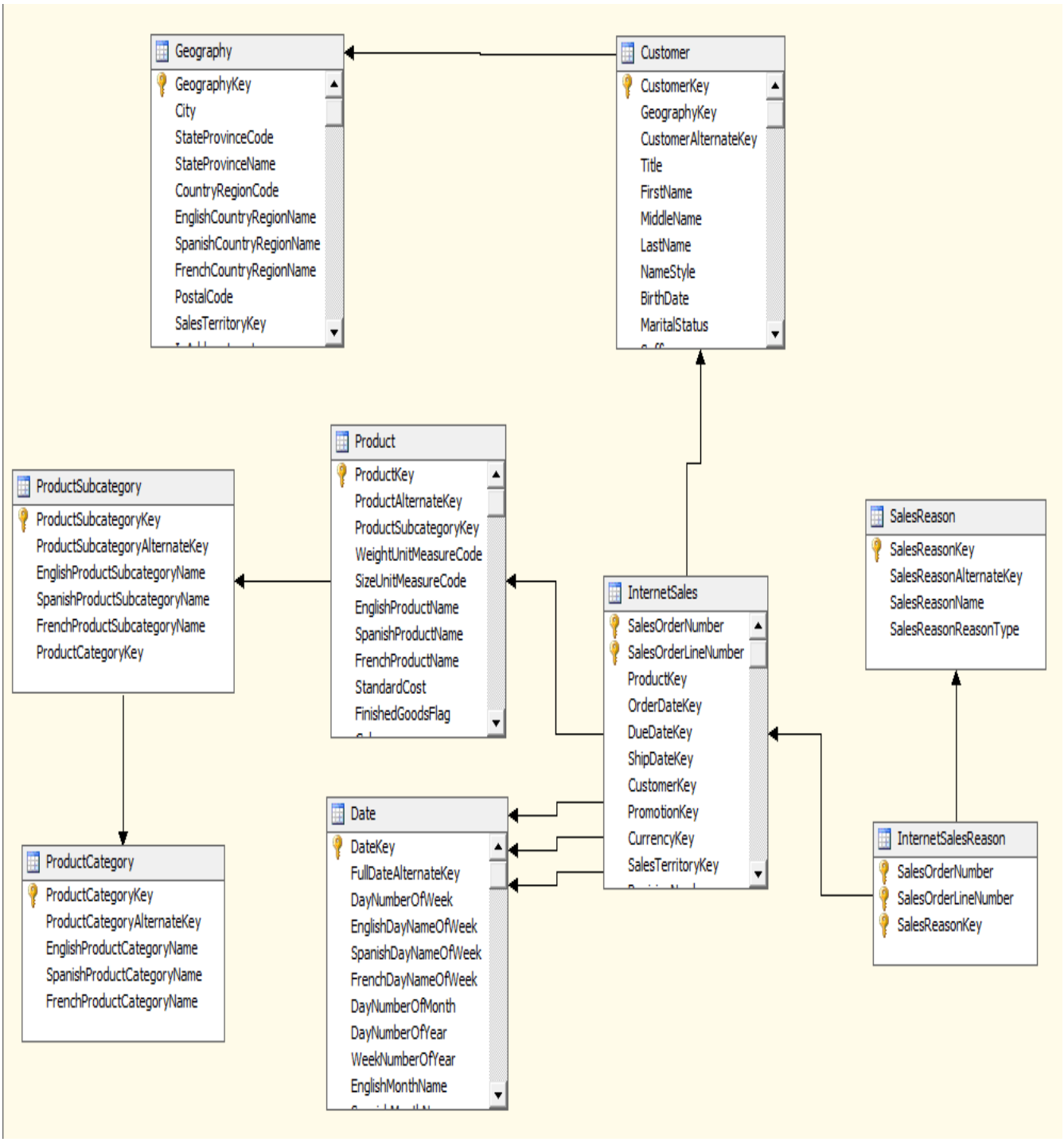


Figure 29: Friendly Cube Structure

In Figure 29 it can be seen that the names of the tables have been changed to more friendly or readable ones. All the Dim and Fact names have been removed so that the tables are more readable in this way.

By: Kiril Anastasov L1087591@live.tees.ac.uk

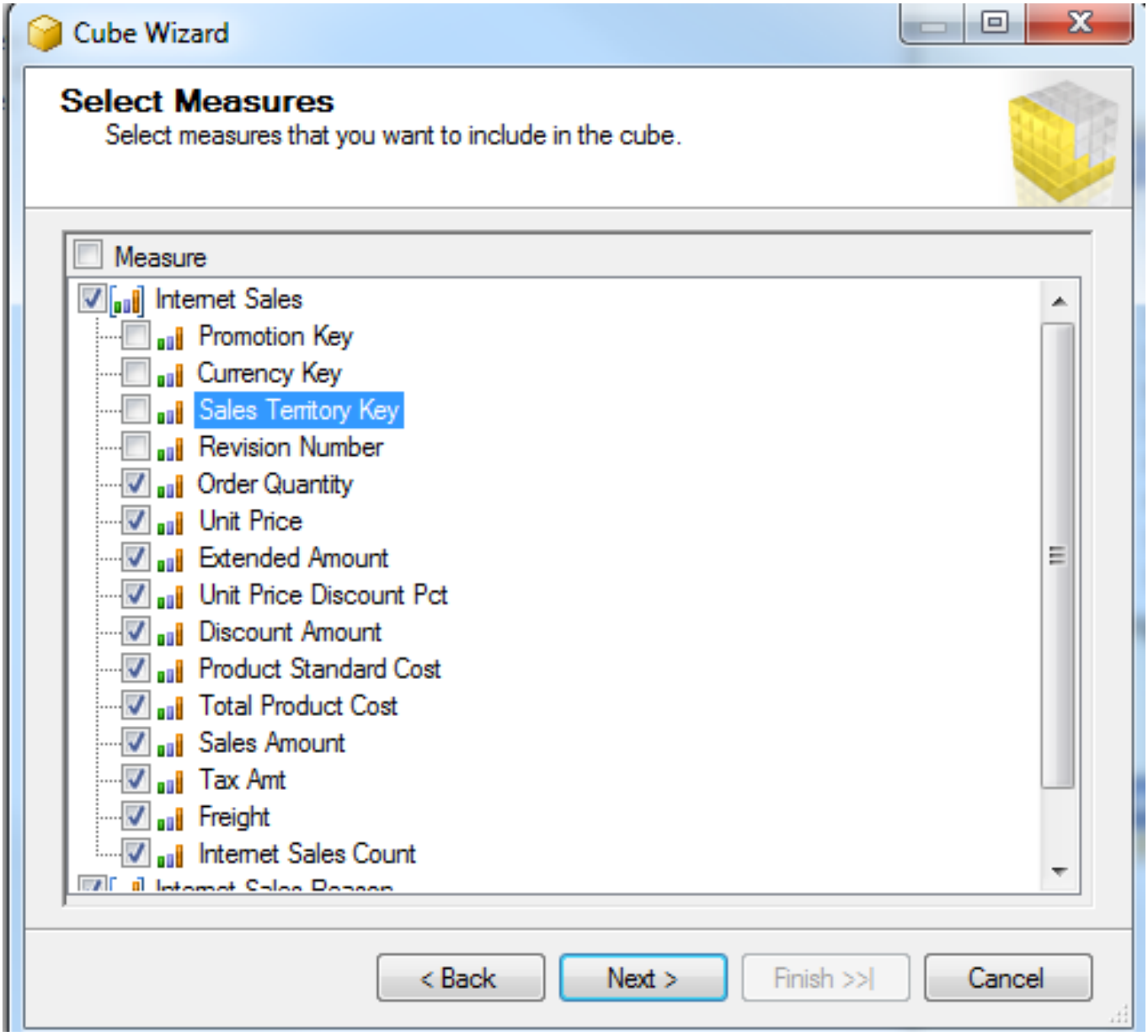


Figure 30: Cube Measures

In Figure 30 we can see that by default, the wizard selects as measures all numeric columns in the fact table that are not linked to dimensions. However, these four columns are not actual measures. The first three are key values that link the fact table with dimension tables that are not used in the initial version of this cube and the measures were not selected.

By: Kiril Anastasov L1087591@live.tees.ac.uk

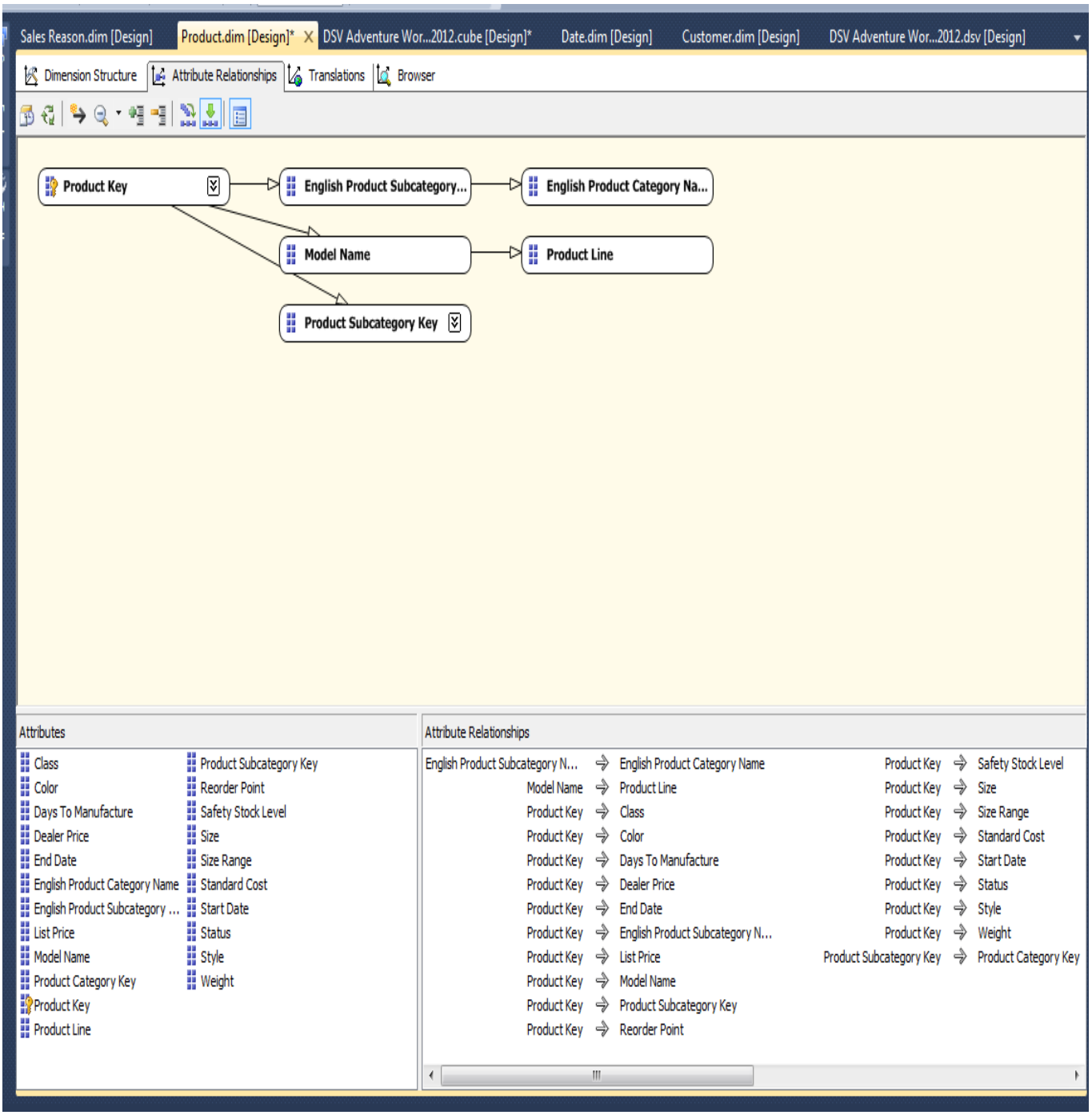


Figure 31: Attribute Relationships

In Figure 31 we can see the attribute relationships. Product Key is connected with English Product Subcategory which is connected with English Product Category and Product Key is connected with Model Name and Model Name is connected with Product Line.

By: Kiril Anastasov L1087591@live.tees.ac.uk

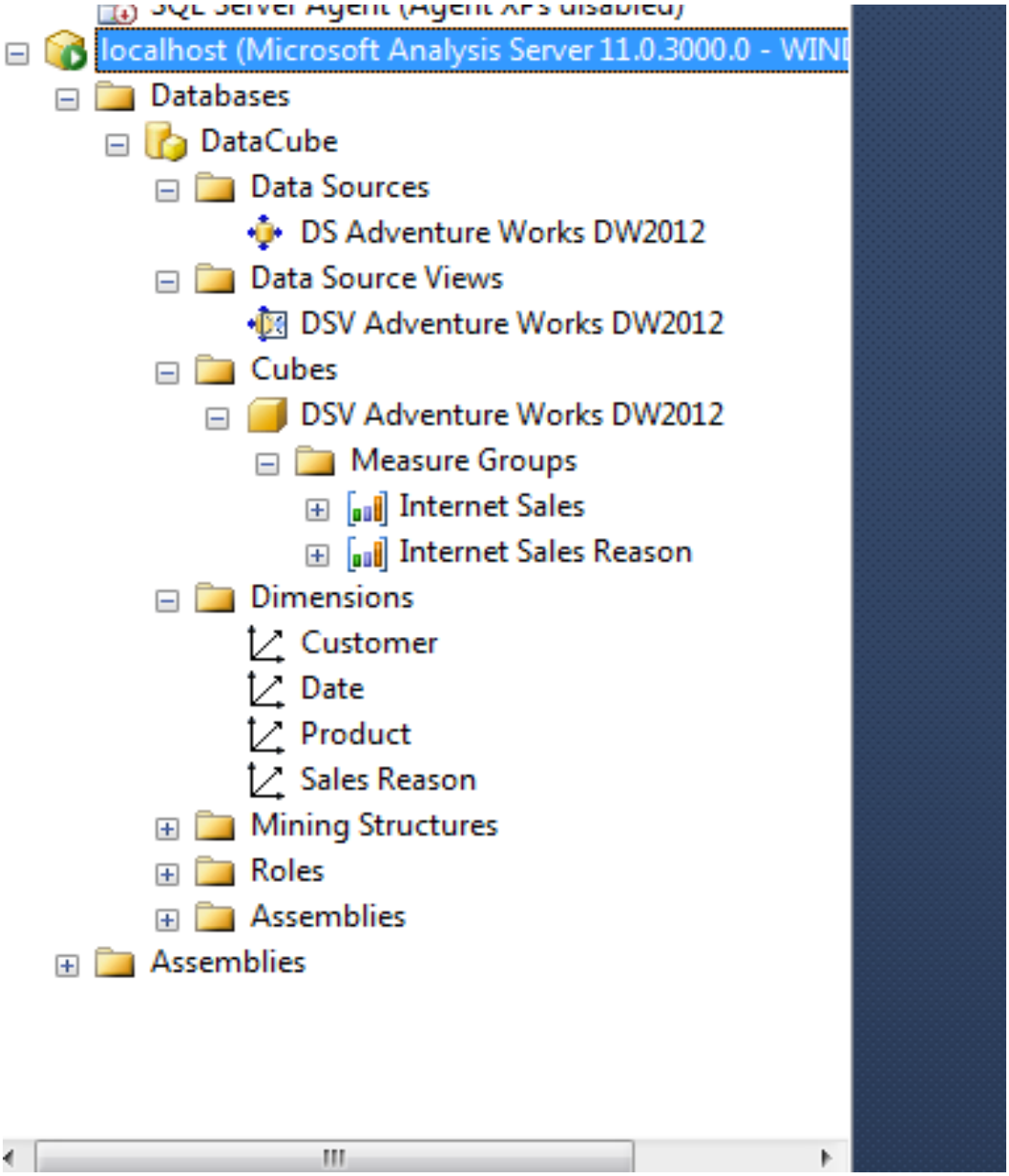


Figure 32: SQL management studio

In Figure 32 we can see that the data cube that was created in the business intelligence can be accessed from the localhost Microsoft Analysis Server. We can see the measured groups, the data sources, the data source views and the dimensions of the data cube. The cube is called DSV Adventure Works DW2012.

By: Kiril Anastasov L1087591@live.tees.ac.uk

7.2 Attribute hierarchies

After careful research there were a few main advantages of user-defined attribute hierarchies for dimensions. The first one is performance of the aggregations. Attributes are one of many elements inside the dimension and does not necessarily have to conform of what the field is, so we can change the format of the text, numbers and make it more user friendly, for example instead of having a name English Month Name we can rename it to Month and the Source Attribute will still have the original name. Also we have attribute type allowing choosing what type of attribute is loaded from a template. The next advantage of creating a hierarchy is that you give the users of the Cube a pre-defined hierarchy which will be a common with all the users. This will save users time because they will not have to drag and drop several members every time. Also if you are working with Payroll Personnel System (PPS), the filters can work directly on the Hierarchy to offer users only one filter with drill down options instead of having the users to choose from multiple filters. Lastly, PPS hierarchies allow the users to do auto drill down in PPS charts. It is also a Microsoft best practice to add big dimensions in some sort of hierarchy and set the AttributeHierachyVisible property to false so the attribute is enable to be browsed through the hierarchy but not alone, which prevents the users of dragging a attribute with many rows to the cube to break it (Diego 2012).

By: Kiril Anastasov L1087591@live.tees.ac.uk

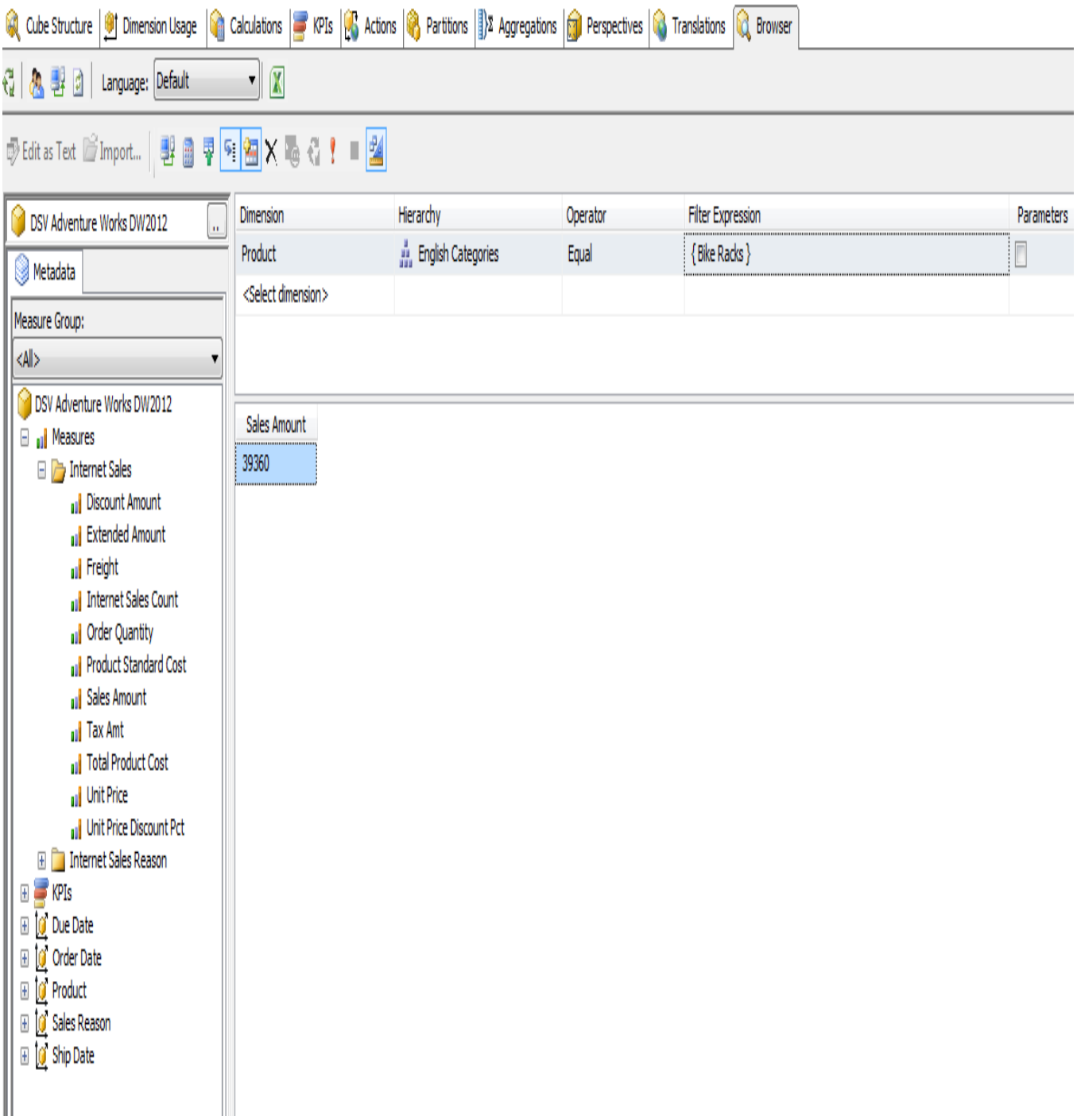


Figure 33: Cube attributes hierarchies for dimensions 1

In Figure 33 it was selected Product for Dimension, English Categories for Hierarchy, Equal for the Operator and it was selected Bike and Racks for the Filter Expression and no Parameters. For the measures it was selected Sales Amount and after execution the result is: 39360.

By: Kiril Anastasov L1087591@live.tees.ac.uk

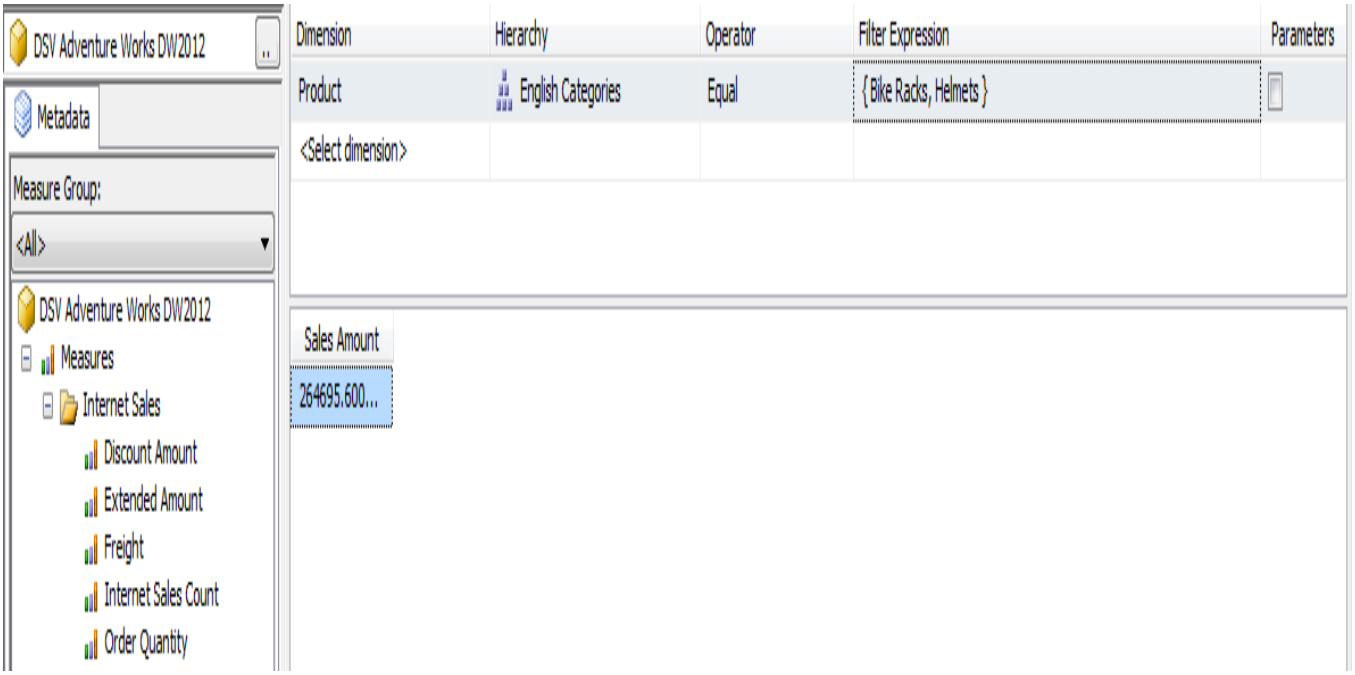


Figure 34: Cube attributes hierarchies for dimensions 2

In Figure 34 it was selected Product for Dimension, English Categories for Hierarchy, Equal for the Operator and it was selected Bike, Racks and Helmets for the Filter Expression and no Parameters. For the measures it was selected Sales Amount and after execution the result is: 264695.600...

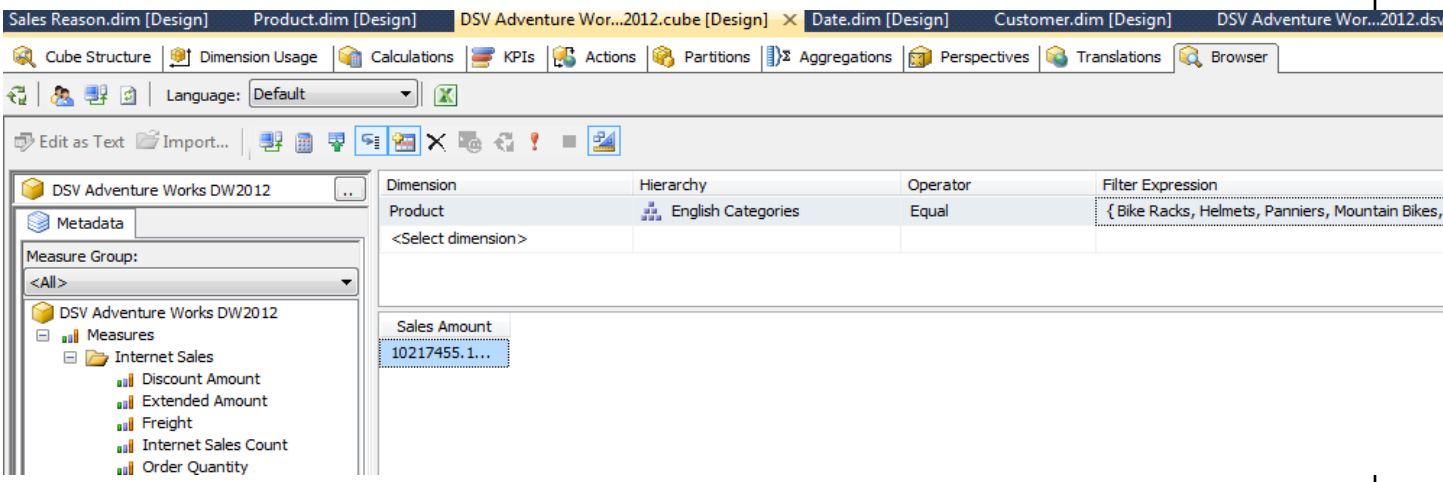


Figure 35: Cube attributes hierarchies for dimensions 3

In Figure 35 it was selected Product for Dimension, English Categories for Hierarchy, Equal for the Operator and it was selected Bike, Racks, Helmets, Panniers, Mountain Bikes and Bib-shorts for the Filter Expression and no Parameters. For the measures it was selected Sales Amount and after execution the result is: 10217455.1...

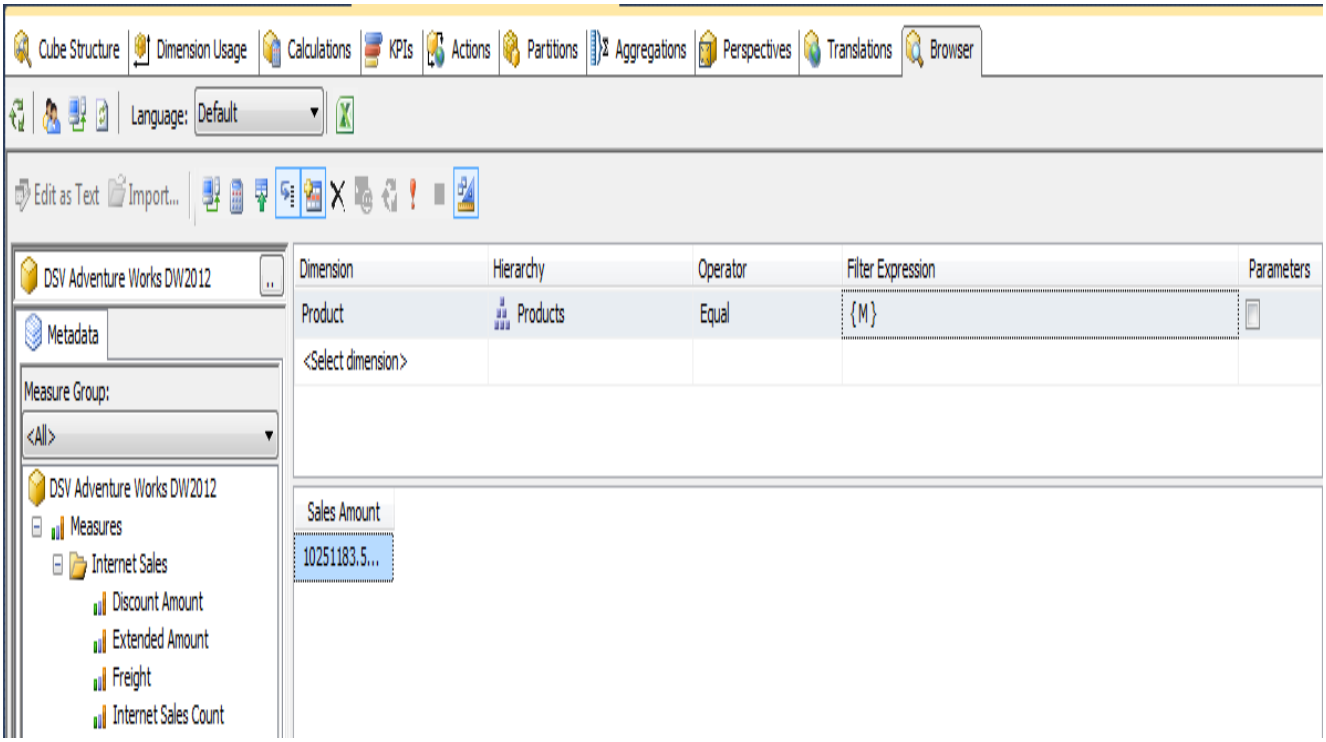


Figure 36: Cube attributes hierarchies for dimensions 4

In Figure 36 it was selected Products for Dimension, Products for Hierarchy, Equal for the Operator and it was selected M for the Filter Expression and no Parameters. For the measures it was selected Sales Amount and after execution the result is: 10251183

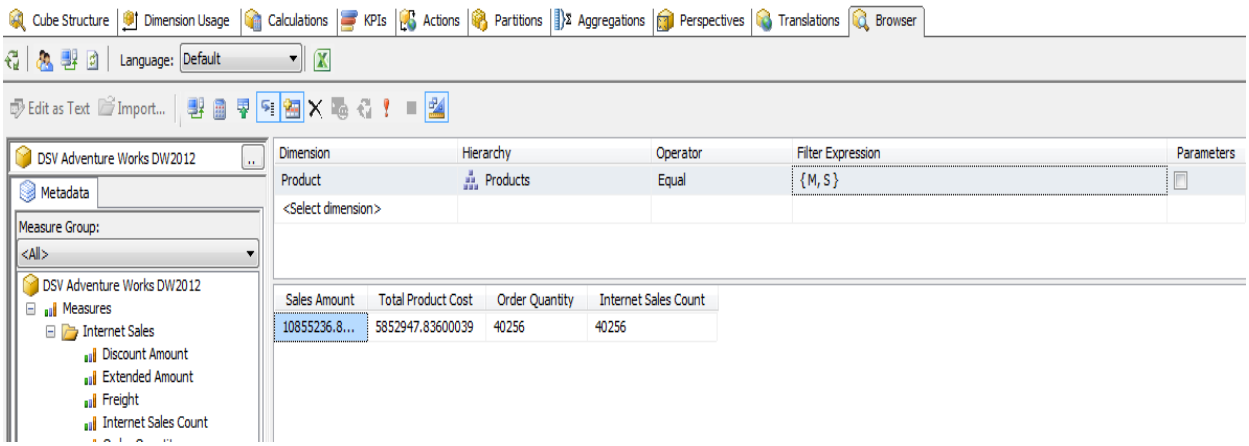


Figure 37: Cube attributes hierarchies for dimensions 5

In Figure 37 it was selected Products for Dimension, Products for Hierarchy, Equal for the Operator and it was selected M and S for the Filter Expression and no Parameters. For the measures it was selected Sales Amount, Total Product Cost, Order Quantity and Internet Sales Count and after execution the result is:

By: Kiril Anastasov L1087591@live.tees.ac.uk

Sales Amount – 10855236.8…

Total Product Cost – 5852957.8360039

Order Quantity - 40256

Internet Sales Count - 40256

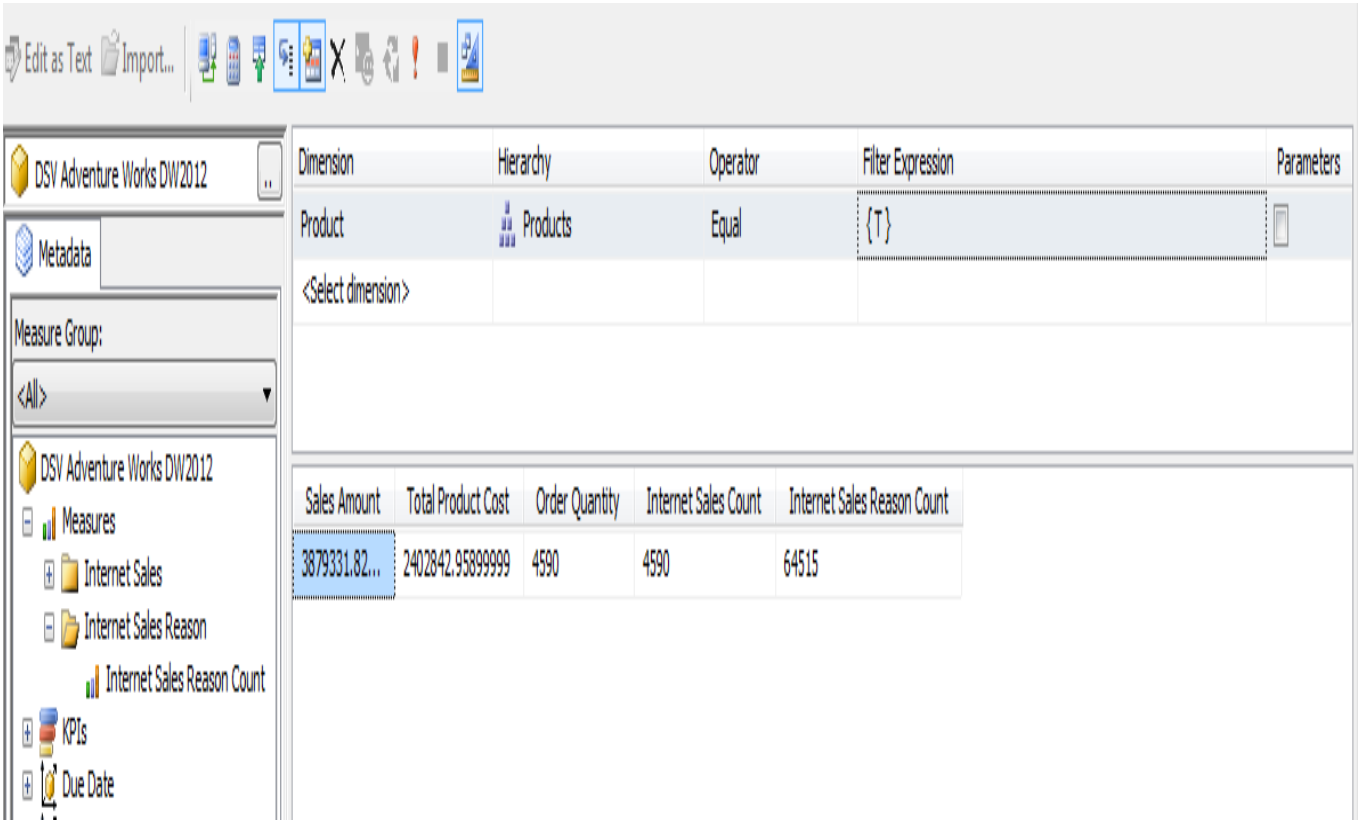


Figure 38: Cube attributes hierarchies for dimensions 6

In Figure 38 it was selected Products for Dimension, Products for Hierarchy, Equal for the Operator and it was selected T for the Filter Expression and no Parameters. For the measures it was selected Sales Amount, Total Product Cost, Order Quantity and Internet Sales Count and after execution the result is:

Sales Amount – 3879331.82…

Total Product Cost – 2402842.95899999

Order Quantity - 4590

Internet Sales Count - 64515

By: Kiril Anastasov L1087591@live.tees.ac.uk

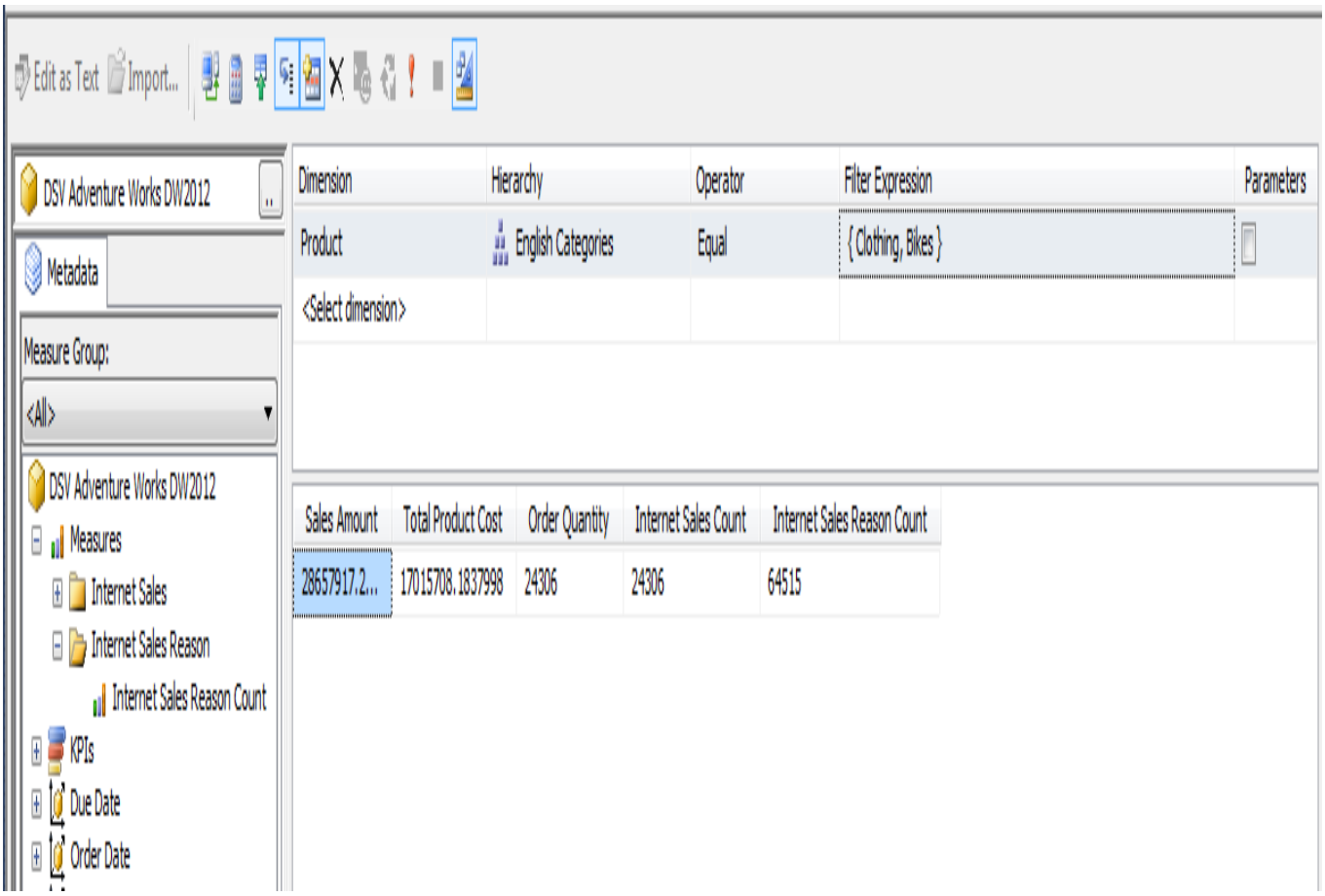


Figure 39: Cube attributes hierarchies for dimensions 7

In Figure 39 it was selected Products for Dimension, English Categories for Hierarchy, Equal for the Operator and it was selected Clothing and Bikes for the Filter Expression and no Parameters. For the measures it was selected Sales Amount, Total Product Cost, Order Quantity, Internet Sales Count and Internet Sales Count and after execution the result is:

Sales Amount – 28657917. 2…

Total Product Cost – 17015708.1837998

Order Quantity – 24306

Internet Sales Count - 24306

Internet Sales Count - 64515

By: Kiril Anastasov L1087591@live.tees.ac.uk

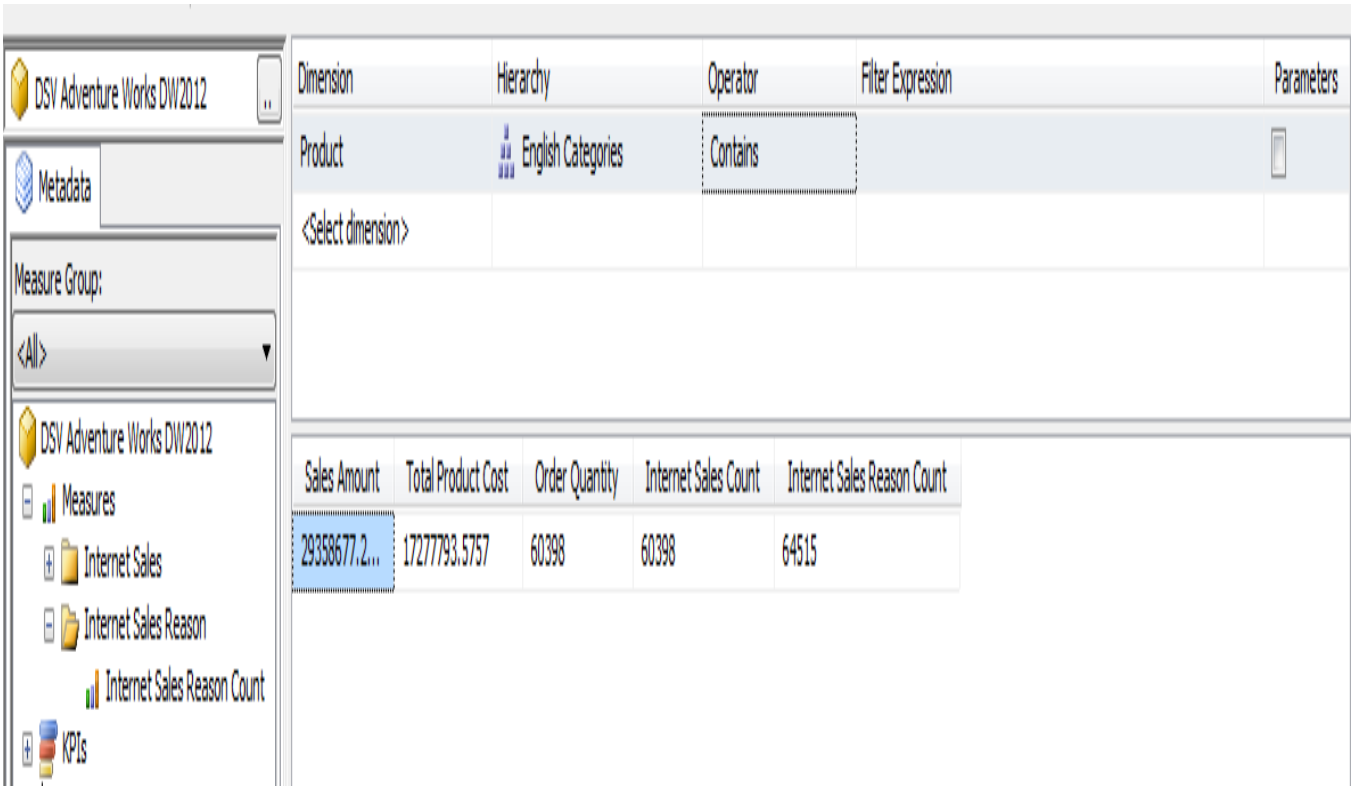


Figure 40: Cube attributes hierarchies for dimensions 8

In Figure 40 it was selected Products for Dimension, English Categories for Hierarchy, Contains for the Operator and no Parameters. For the measures it was selected Sales Amount, Total Product Cost, Order Quantity, Internet Sales Count and Internet Sales Count and after execution the result is:

Sales Amount – 29358677. 2…

Total Product Cost – 17277793.5757

Order Quantity – 60398

Internet Sales Count - 60398

Internet Sales Count - 64515

By: Kiril Anastasov L1087591@live.tees.ac.uk

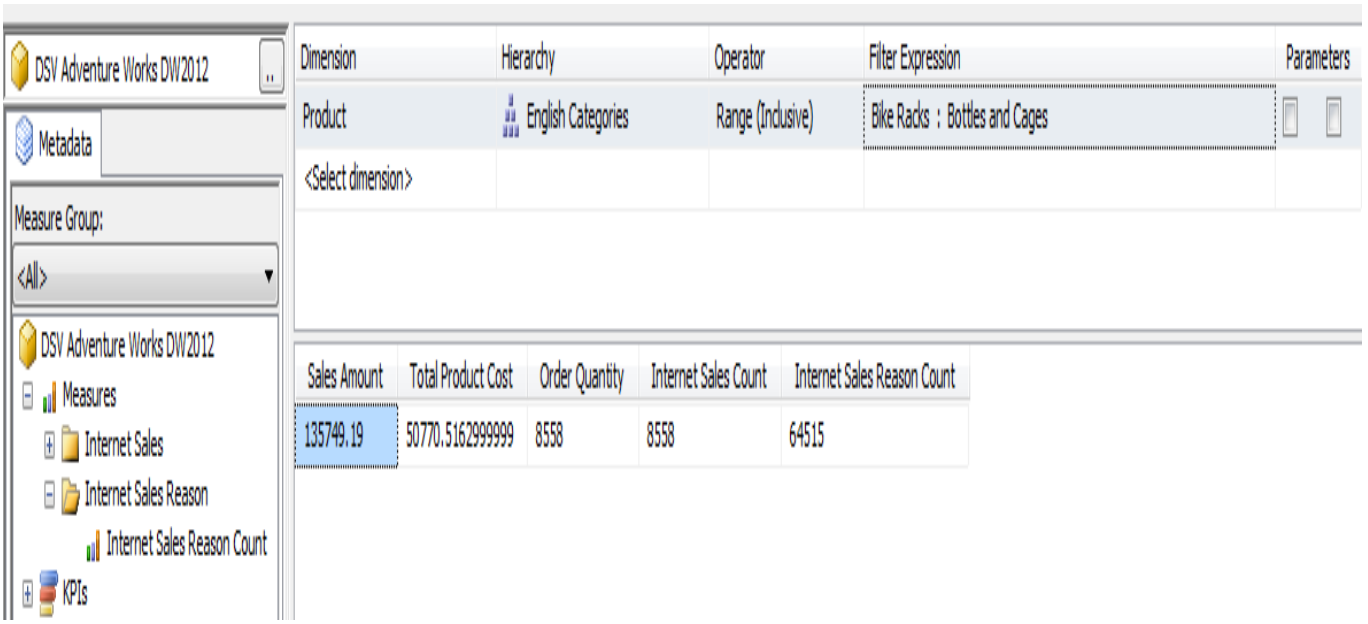


Figure 41: Cube attributes hierarchies for dimensions 9

In Figure 41 it was selected Products for Dimension, English Categories for Hierarchy, Range (Inclusive) for the Operator and it was selected Bike Racks: Bottle of Cages the Filter Expression and no Parameters. For the measures it was selected Sales Amount, Total Product Cost, Order Quantity, Internet Sales Count and Internet Sales Count and after execution the result is:

Sales Amount – 135749.19

Total Product Cost – 50770.5162999999

Order Quantity – 8558

Internet Sales Count - 8558

Internet Sales Count - 64515

By: Kiril Anastasov L1087591@live.tees.ac.uk

7.3 AdventureWorksDW2012 tables

We can use index to access the specific information quickly in database table. In database, index is a data structure, which orders data of one or more columns in database table. For example, SalesOrderNumber column in a FactInternetSalesReason table, if a developer wants to search specific sale with SalesOrderNumber, and compared it with table, index can help the developer access the information more quickly.

Index is a database structure. It can be created by using one or more columns of a database table, providing the basis for both rapid random look ups and efficient access of ordered records.

Index could provide the data pointer that specified to the data value set which is stored in specified column, and then order these pointers according to the specified sort order. Index in database is very similar to the index in using books: it searches index and finds out the special value, then finds out the column in this value following with the pointers.

In a relational database an index is a copy of part of a table. In "INDEX/KEY" property page of the selected table, developer can CREATE, EDIT, or DELETE the type of each index. Indexes may be defined as unique or non-unique. A unique index acts as a constraint on the table by preventing duplicate entries in the index and thus, the backing table. When stored the index to the attach table or stored in relational graph where the table is, index will be stored in the database. (What is index, 2014)

A primary key is a LOGICAL concept – it is the unique identifier for a row in a table. As such, it has a bunch of attributes - it may not be null, and it must be unique. Naturally, you are likely to be searching for records by their unique identifier a lot, it would be good to have an index on the primary key. A clustered index is a PHYSICAL concept – it is an index that affects the order in which records are stored on disk. This makes it a very fast index when accessing data, though it may slow down writes if your primary key is not a sequential number. You can have a primary key without a clustered index - and sometimes, it is the better option (for example when your primary key is a combination of foreign keys on a joining table, and you do not want to incur the disk shuffle overhead when writing). You can create a clustered index on columns that are not a primary key. (Neville, 2013)

The analyses service works with the FactInternetSales, FactInternetSalesReason and DimSalesReason tables and the database indexes service are structured correctly for queries for these tables.

By: Kiril Anastasov L1087591@live.tees.ac.uk

Task 8 Data Mining

8.1.1 Decision Tree

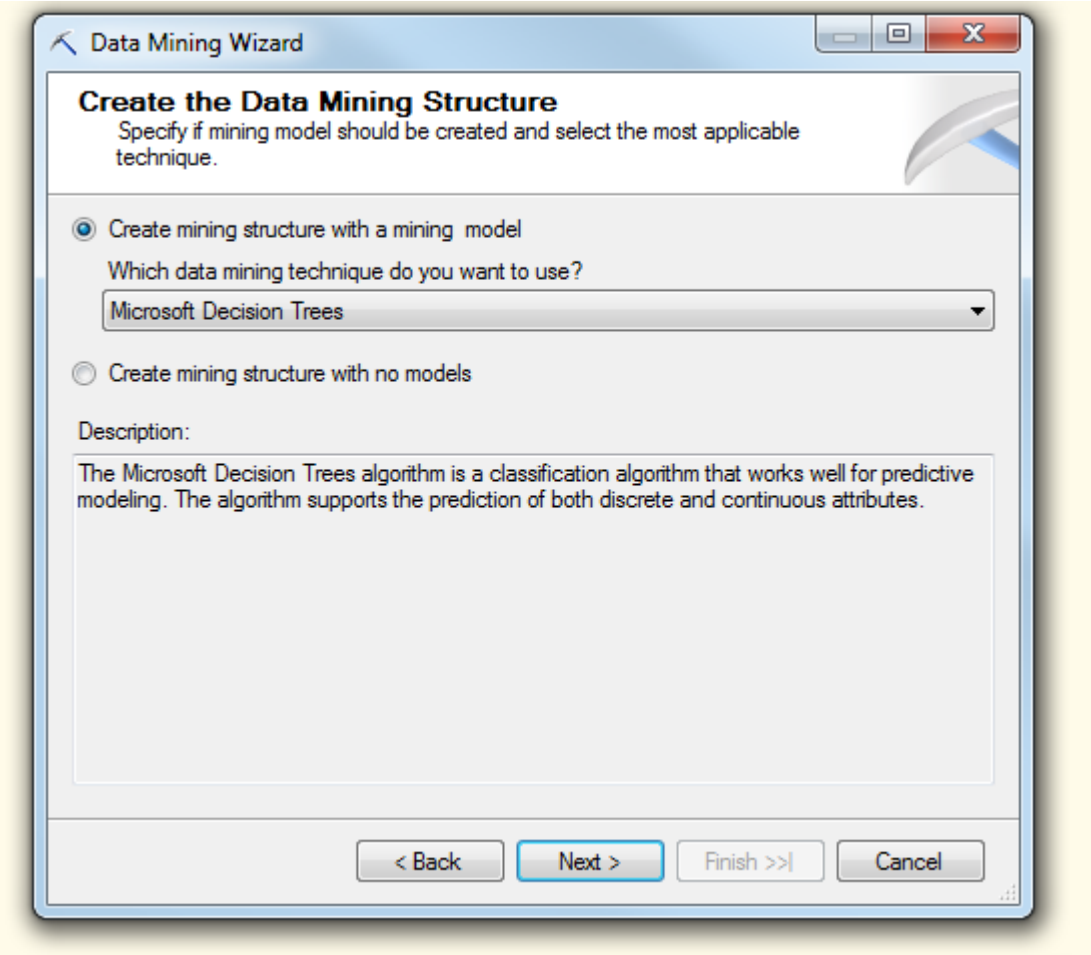


Figure 42: Decision Tree

In Figure 42 we start the creating of the Data Mining with the decision tree Algorithm through the Data Mining Wizard.

By: Kiril Anastasov L1087591@live.tees.ac.uk

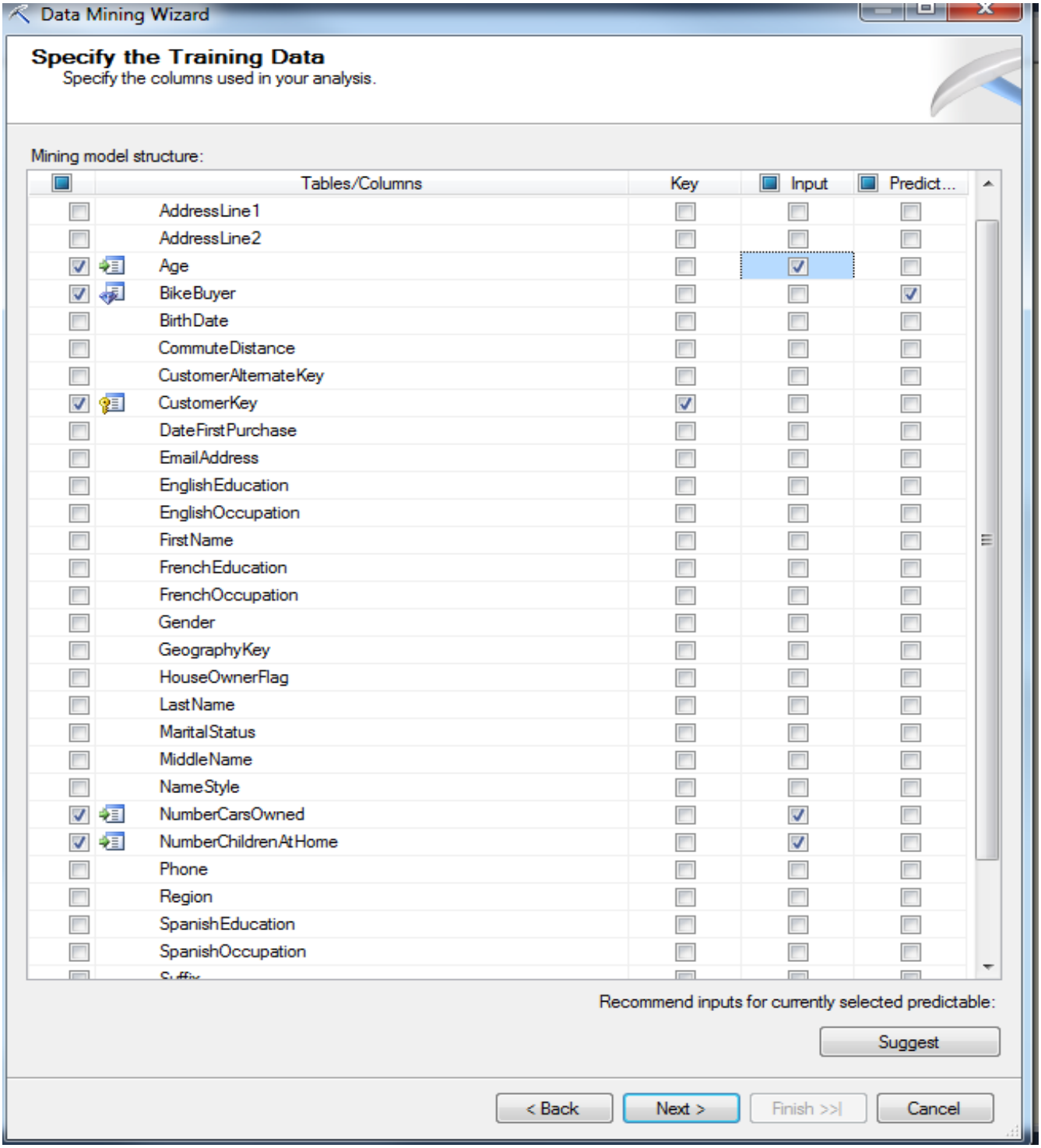


Figure 43: Training data

In Figure 43 we specify the training data like Input data parameters and what parameter we want to predict in our case BikeBuyers,

By: Kiril Anastasov L1087591@live.tees.ac.uk

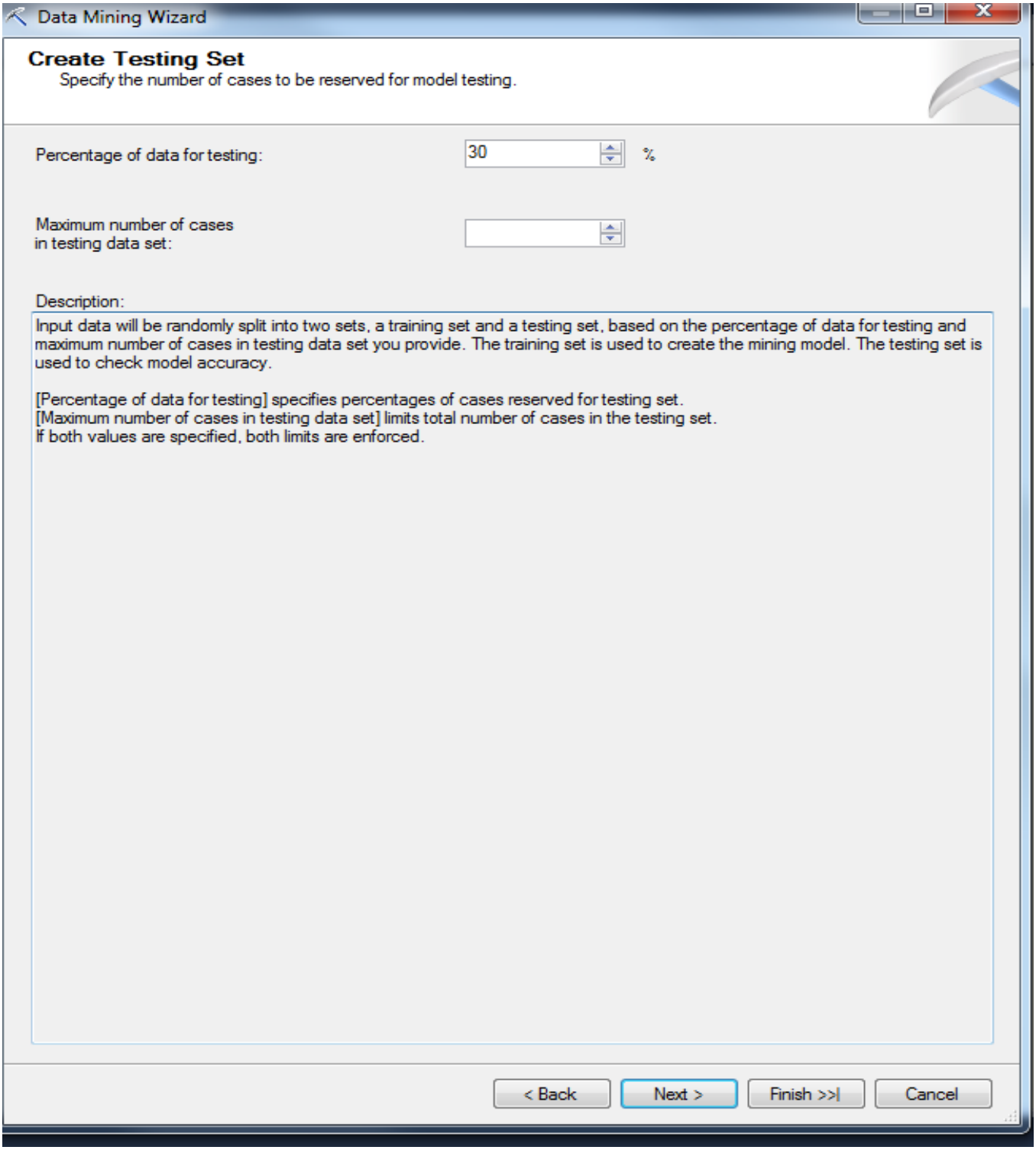


Figure 44: Testing set

In Figure 44 we specify the percentage of data for testing. We leave the default value 30%. The testing set is used to check the model accuracy.

By: Kiril Anastasov L1087591@live.tees.ac.uk

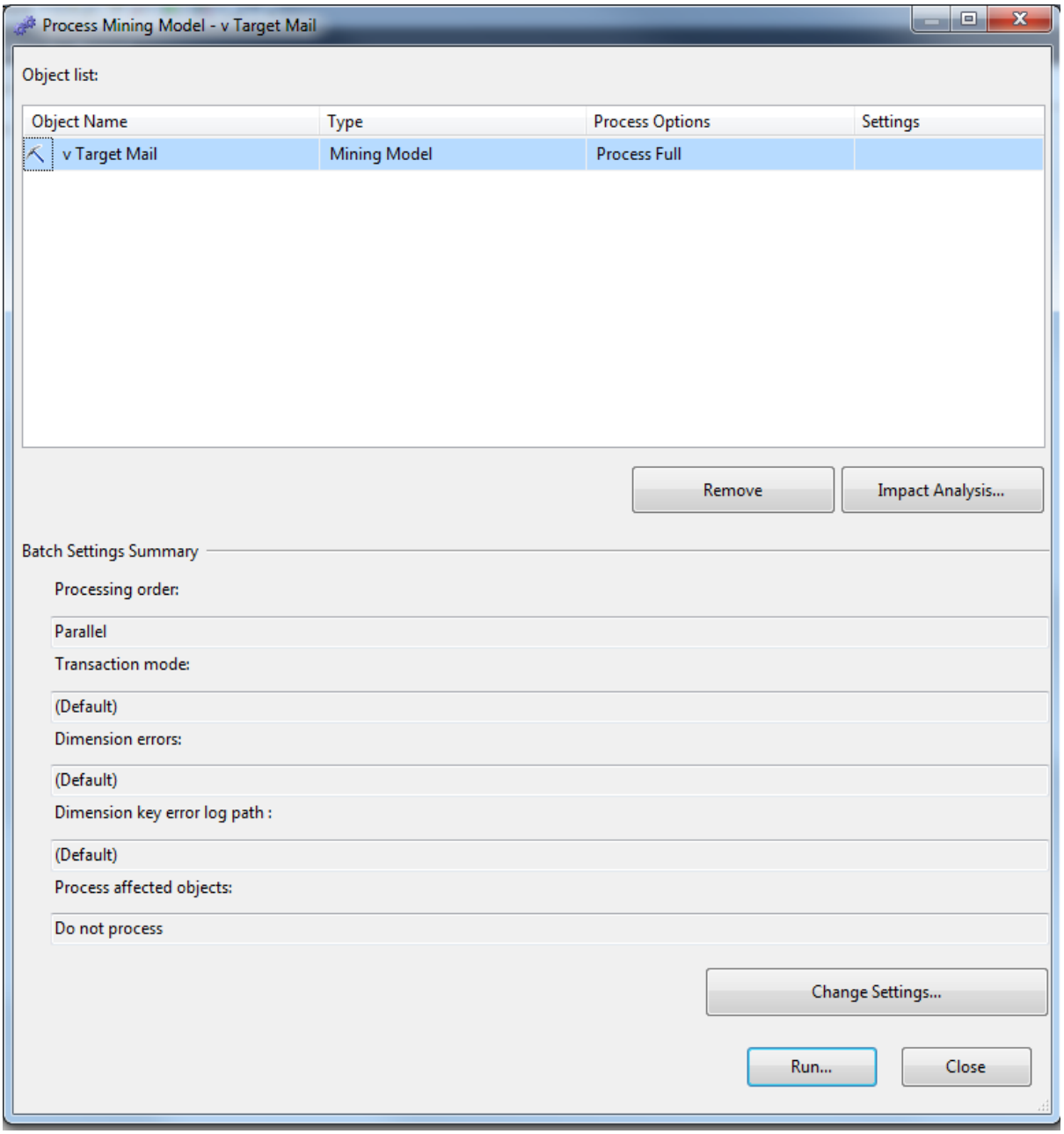


Figure 45: Processing model

In Figure 45 we can see the Process Mining Model.

By: Kiril Anastasov L1087591@live.tees.ac.uk

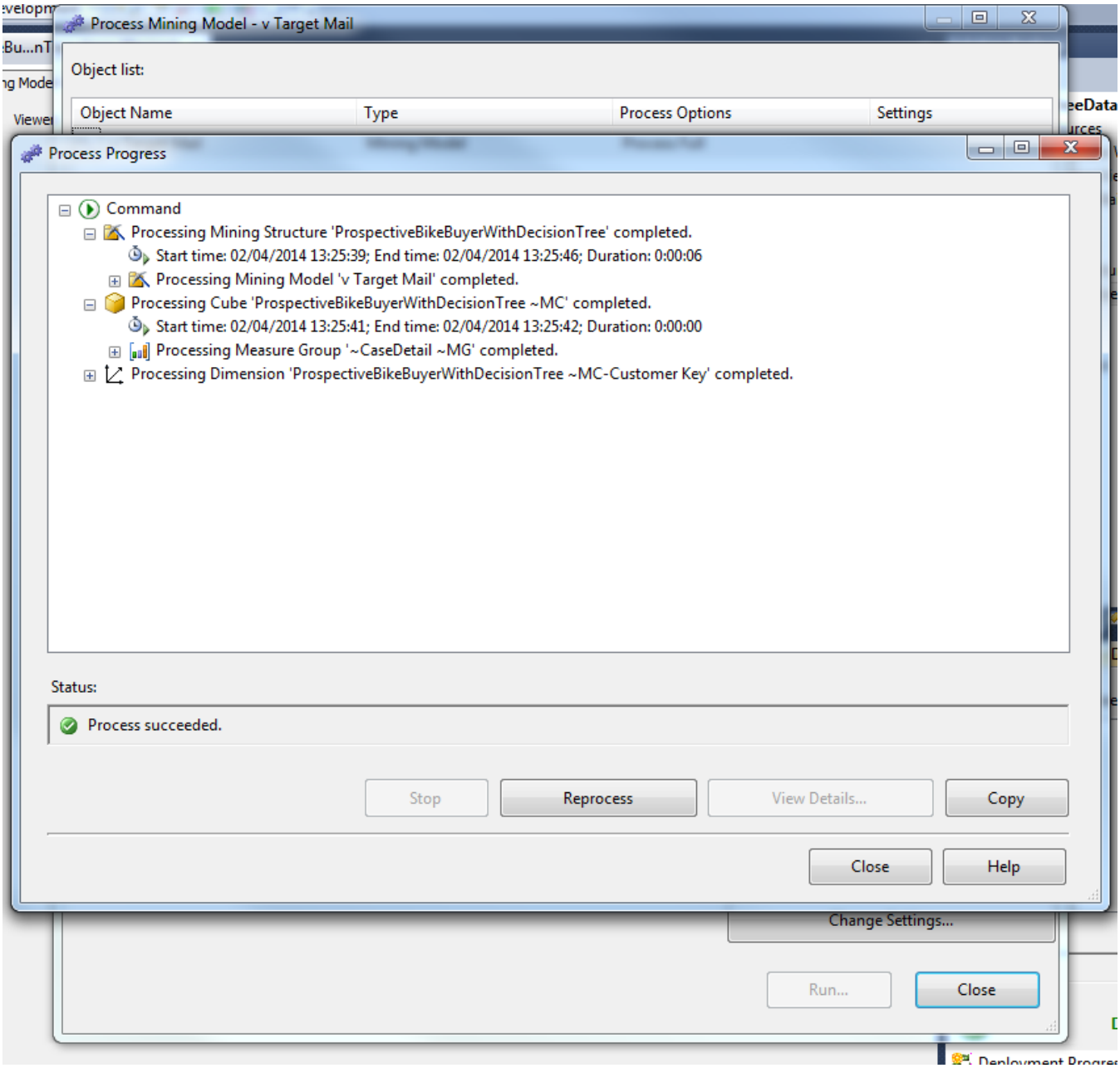
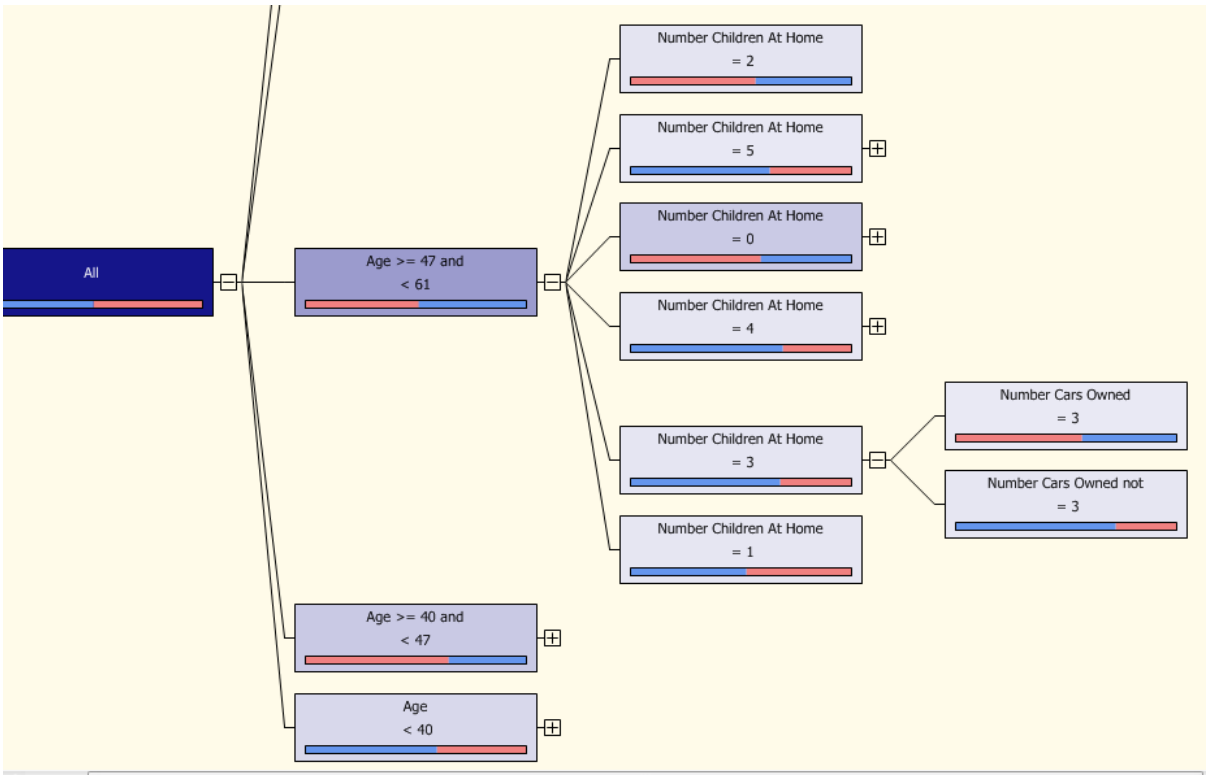


Figure 46: Process progress

In Figure 46 we run the Process Mining Model and it can be seen that the status of the Process is successful.

By: Kiril Anastasov L1087591@live.tees.ac.uk



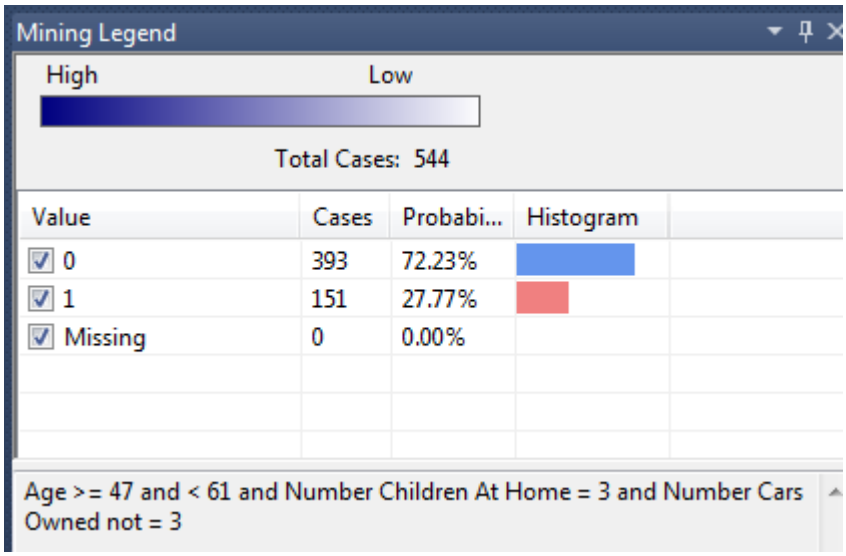


Figure 47: Data mining with Decision Tree

In Figure 47 we can see the Decision Tree and that there is 72.23% probability for someone with our characteristics to buy a bike when they do not have one and 27.77% to buy a bike when they already have one.

By: Kiril Anastasov L1087591@live.tees.ac.uk

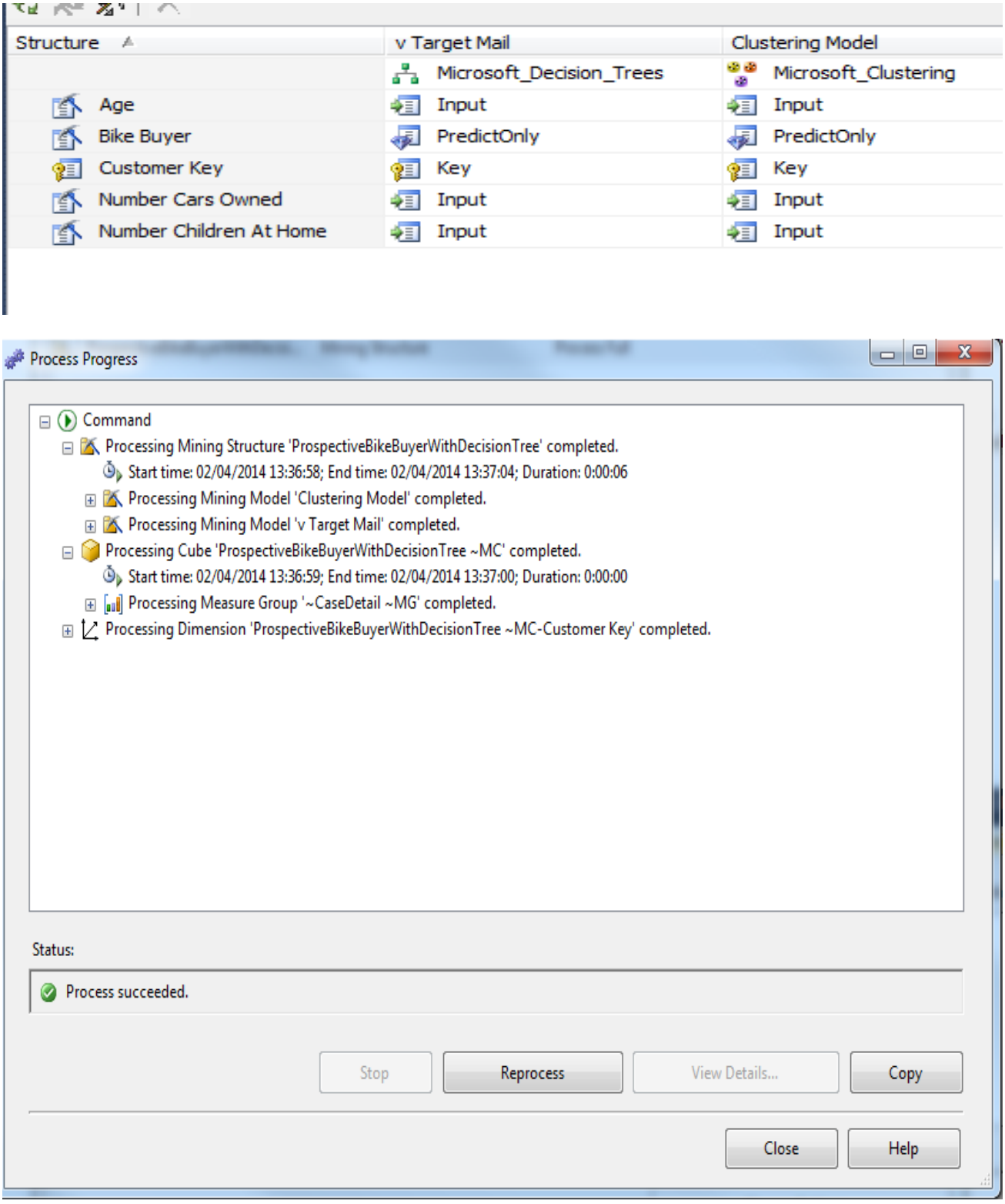


Figure 48: Data mining with clustering

In Figure 48 we can see the Decision Tree and the clustering Models.

By: Kiril Anastasov L1087591@live.tees.ac.uk

8.1.2 Clustering Model

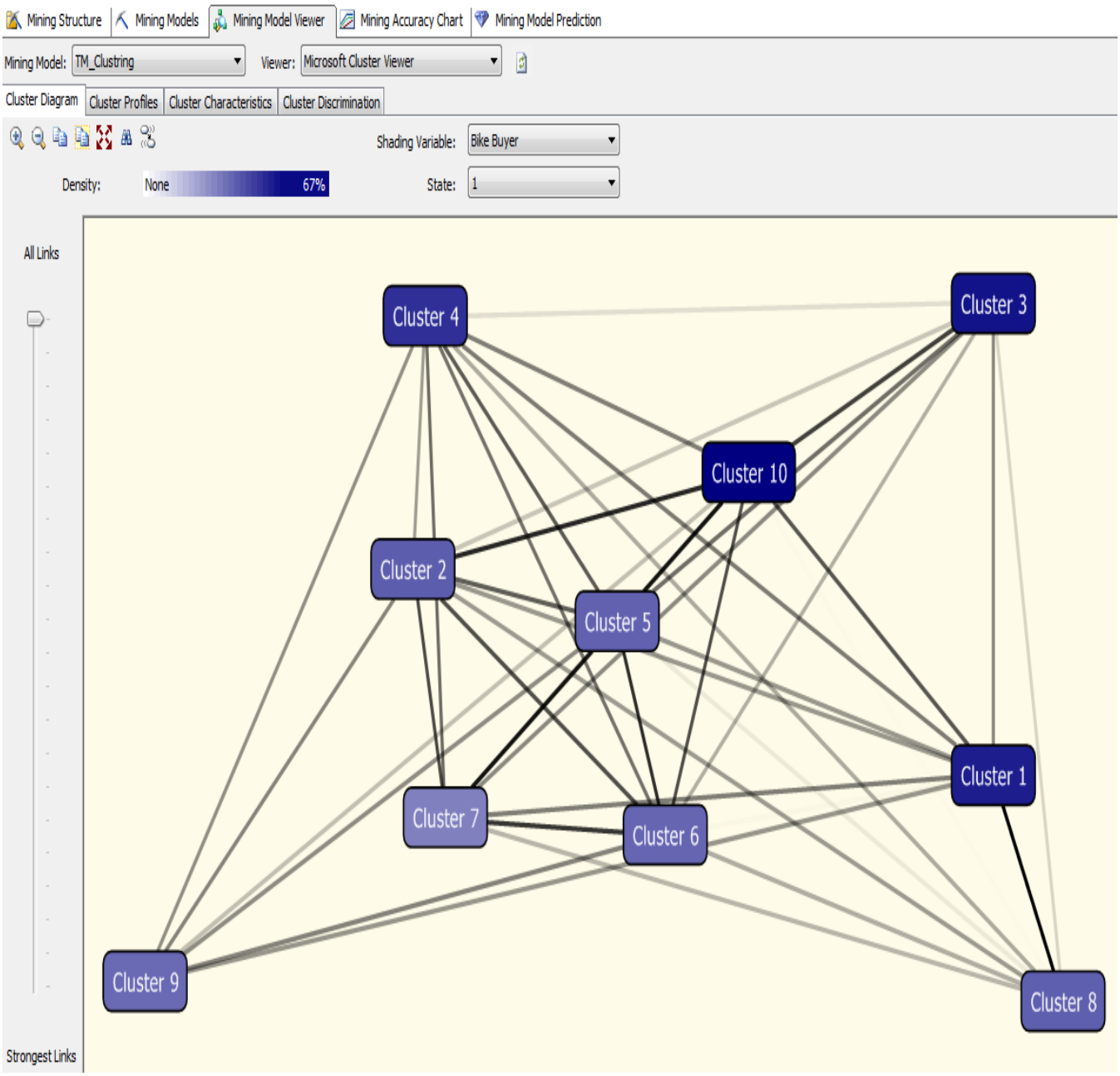


Figure 49: Cluster model

In Figure 49 we can see the cluster model. There are 10 Clusters and cluster number 10 is the strongest (with most intense color). The probability to buy a bike is 67% which is similar to the percentage we have from the decision tree.

By: Kiril Anastasov L1087591@live.tees.ac.uk

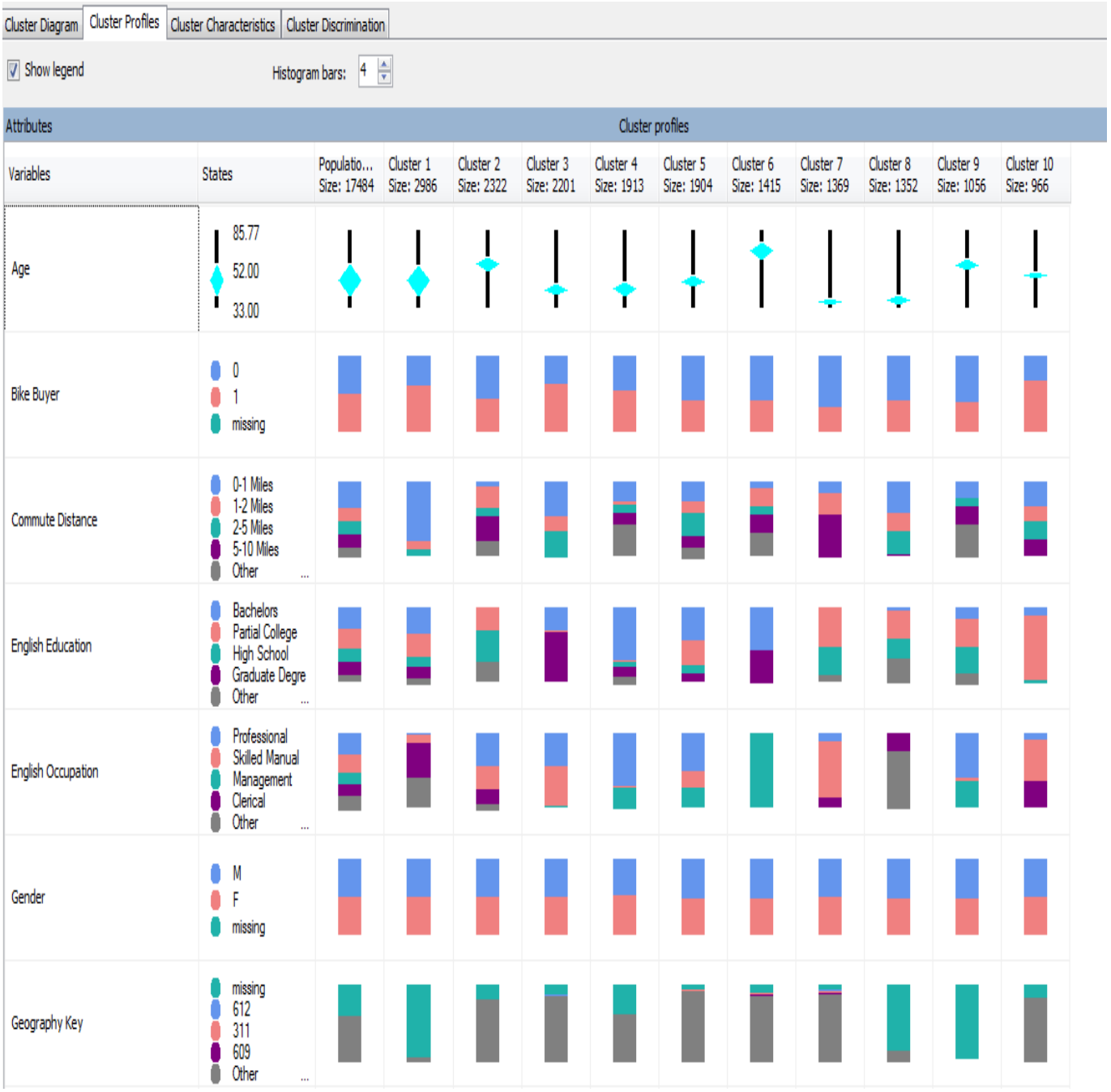


Figure 50: Cluster profiles

In Figure 50 we can see the Cluster profiles. We can see the variables like Age, Bike Buyers, Commute Distance, English Education, English Occupation, Gender and Geography Key.

By: Kiril Anastasov L1087591@live.tees.ac.uk

Characteristics for Population (All)

Variables	Values	Probability
House Owner Flag	1	
Number Children At Home	0	
Marital Status	M	
Region	North America	
Bike Buyer	0	
Gender	M	
Gender	F	
Bike Buyer	1	
Marital Status	S	
Geography Key	missing	
Number Cars Owned	2	
Commute Distance	0-1 Miles	
House Owner Flag	0	
Region	Europe	
English Occupation	Professional	
English Education	Bachelors	
Total Children	0	
English Education	Partial College	
Number Cars Owned	1	
Yearly Income	57,244.9 - 79,029.9	
Yearly Income	35,459.9 - 57,244.9	
Age	52 - 59	
Age	45 - 51	
Yearly Income	79,029.9 - 154,140.5	
Age	60 - 85	
English Occupation	Skilled Manual	
Number Cars Owned	0	
Total Children	2	

Figure 51: Cluster Characteristics

In Figure 51 we can see the Characteristics for all the population, the values and the Probability

By: Kiril Anastasov L1087591@live.tees.ac.uk

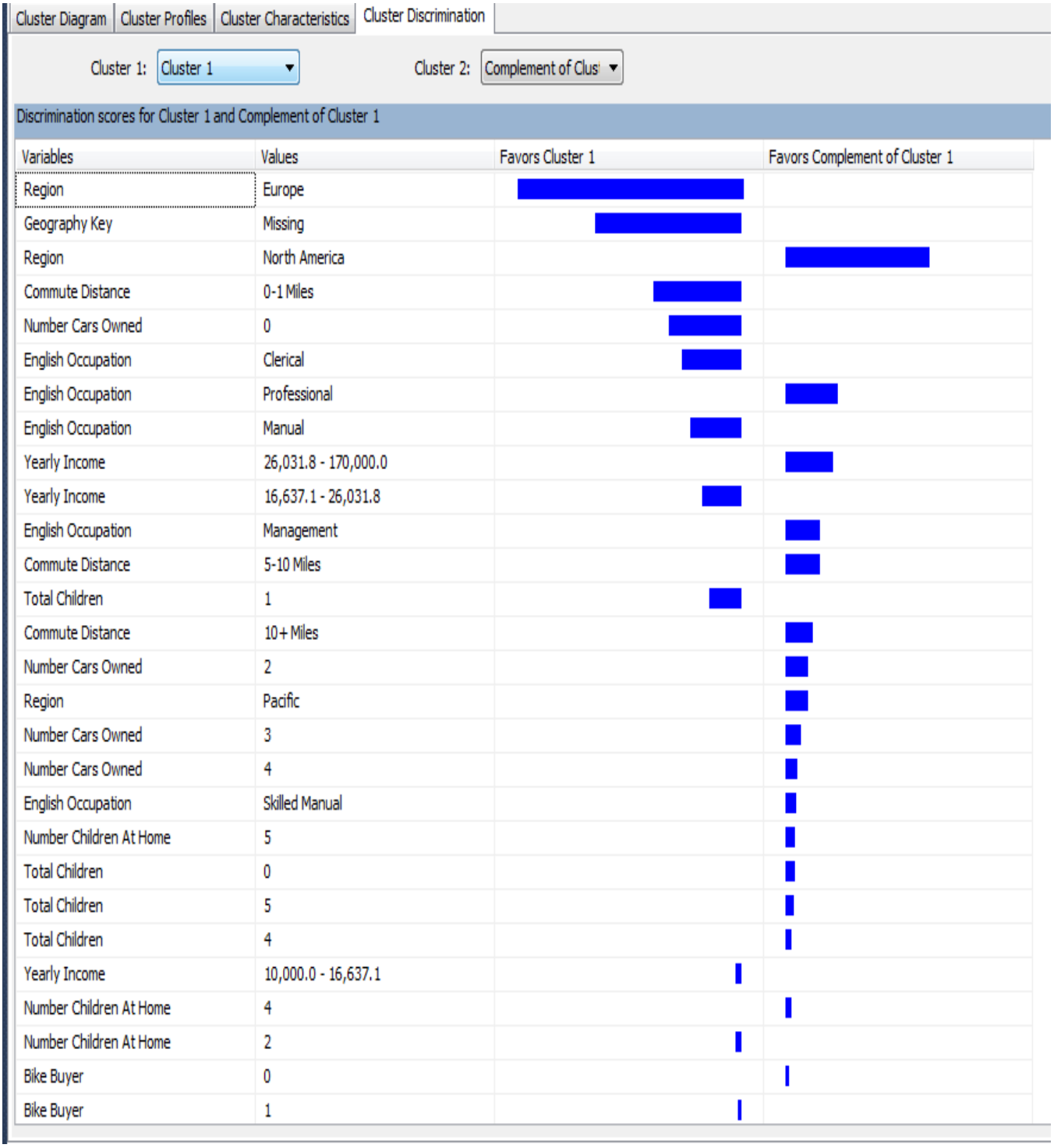


Figure 52: Cluster Discrimination

In Figure 52 we can see the Discrimination scores for Cluster 1 and Complement of cluster 1. All the variables, Values, Favours Cluster 1 and Favors Complements of Cluster 1 can be seen.

By: Kiril Anastasov L1087591@live.tees.ac.uk

8.2 Singleton Queries

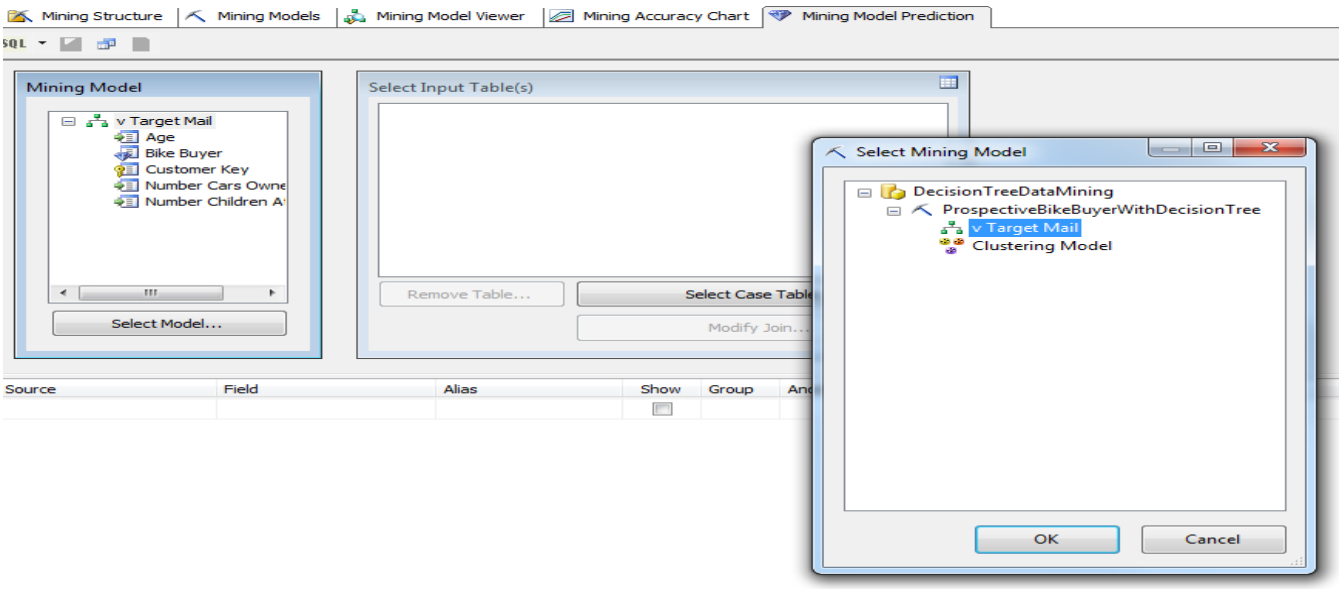


Figure 53 Singleton queries for decision tree

Decision Tree Singleton

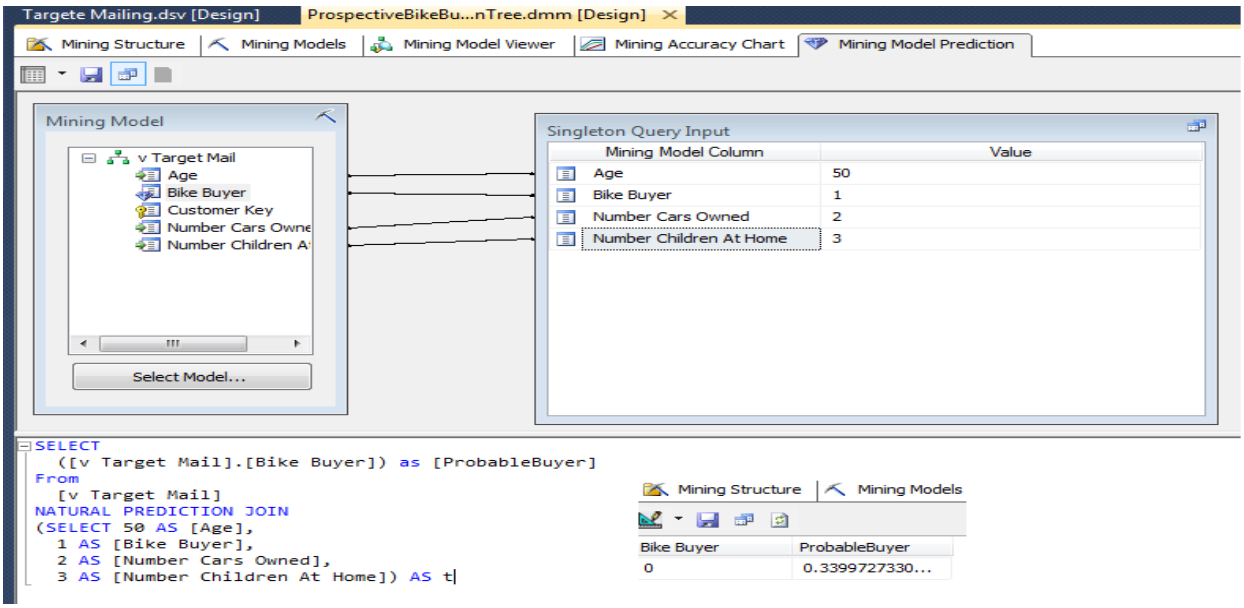


Figure 54: Singleton queries mining model

In Figure 54 we can see that the result for the Singleton query for the decision tree model give approximately 0.34 % of buying a bike for a customer who is 50 years old, has 3 children and 2 cars.

By: Kiril Anastasov L1087591@live.tees.ac.uk

Clustering singleton

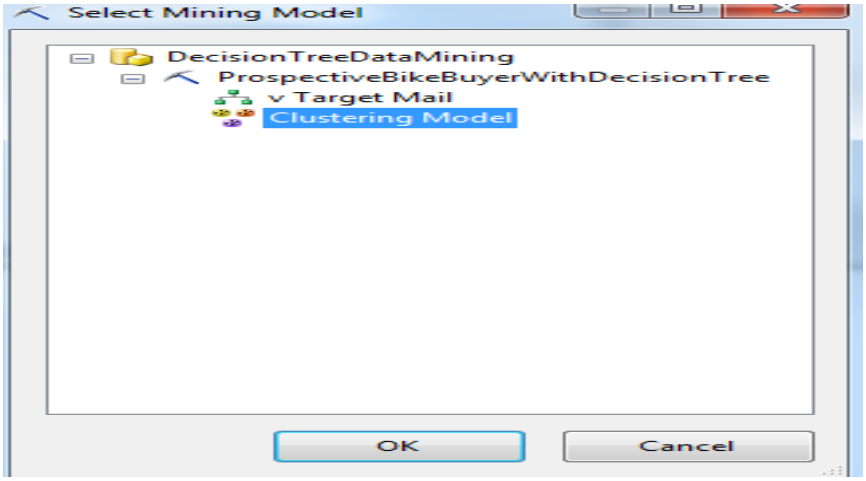
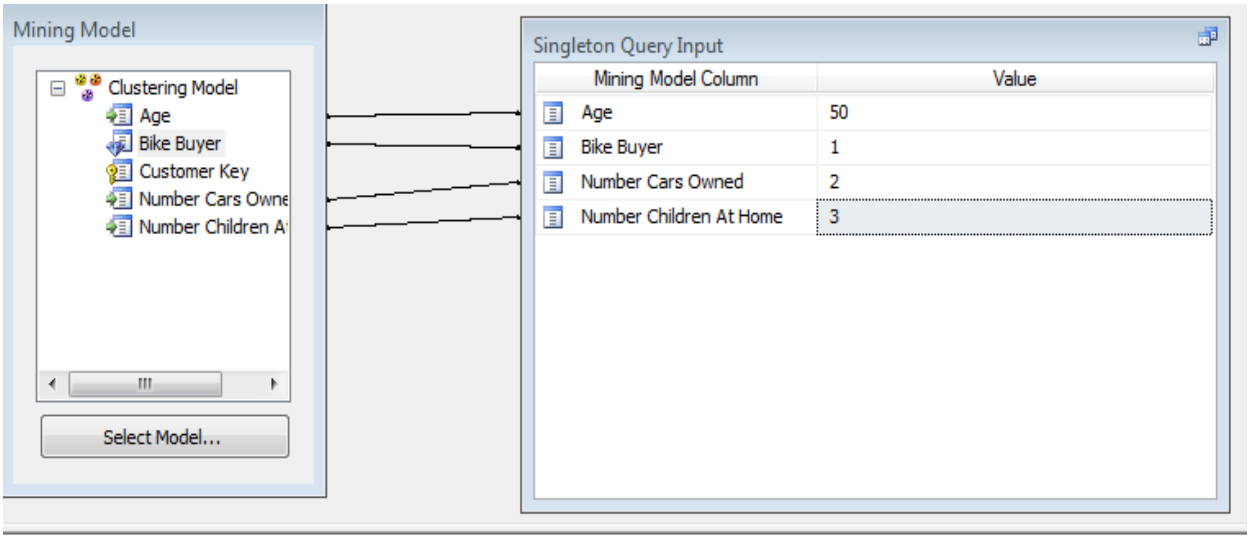


Figure 55: Singleton queries for clustering




```
SELECT
  ([Clustering Model].[Bike Buyer]) as [ProspectiveBuyer]
From
  [Clustering Model]
NATURAL PREDICTION JOIN
(SELECT 50 AS [Age],
  1 AS [Bike Buyer],
  2 AS [Number Cars Owned],
  3 AS [Number Children At Home]) AS t
```

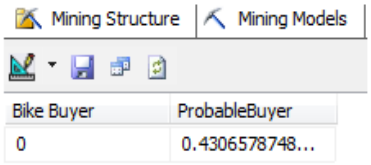
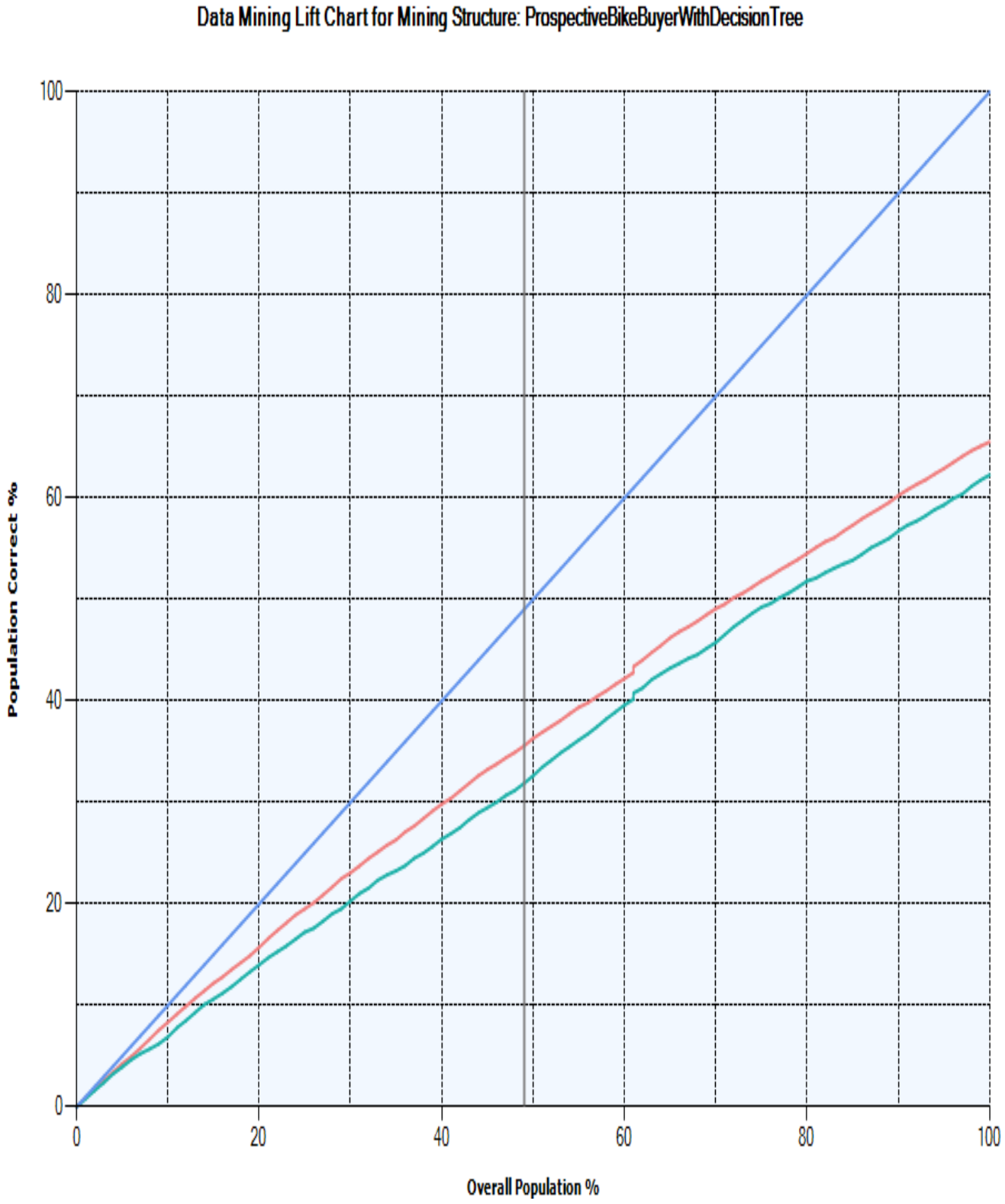


Figure 56: Singleton queries clustering model

In Figure 56 we can see that the result for the Singleton query for the clustering model give approximately 0.43% of buying a bike for a customer who is 50 years old, has 3 children and 2 cars.

By: Kiril Anastasov L1087591@live.tees.ac.uk

8.3 Better model.



By: Kiril Anastasov L1087591@live.tees.ac.uk

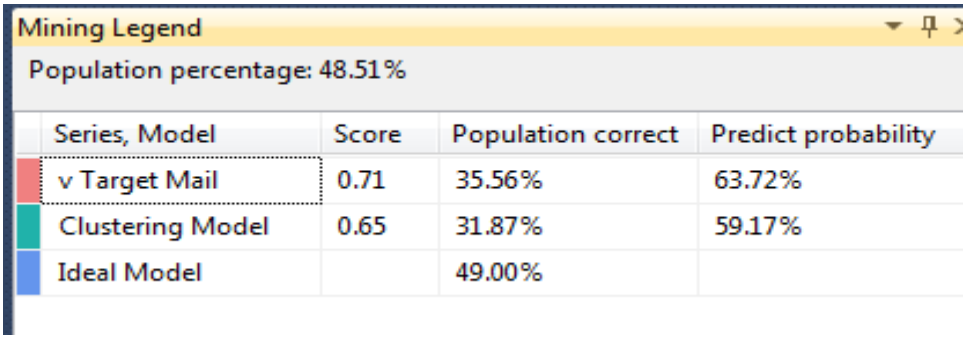


Figure 57: Lift Chart

In Figure 57 we can see 3 lines: The blue line represents the Ideal Model, the red line represents the decision tree model and the green line represents the Clustering model. It can be deducted that the red line (decision tree model) was more suited to the explorations and predictions that were carried out in parts one and two for two reasons. First of all, the red line (decision tree model) is closer to the Ideal Model, which can be seen from Figure 57. Second of all, the red line (decision tree model) gives better results. For example the Score of the decision tree model (0.71)is higher to the score of the clustering model (green line, 0.65), the population correct variable (35.56%) is higher for the decision tree model compared to the clustering model (31.87%) and the predict probability variable(63.72%) is higher for the decision tree model compared to the clustering one(59.17%). For those reasons, the decision tree model was more appropriate for the given data, desired prediction and input variables. It can be taken into consideration that both the decision tree model and the Clustering model use already implemented algorithms from Microsoft. If the algorithms are implemented differently and different data set is provided then the Clustering model may give better results.

By: Kiril Anastasov L1087591@live.tees.ac.uk

Task 9 Big Data

Big data analytics is the use of advanced analytic techniques against very large, diverse data sets that include different types such as structured/unstructured and streaming/batch and different sizes from terabytes to zettabytes. Big data is a term applied to data sets whose size or type is beyond the ability of traditional relational databases to capture, manage, and process the data with low-latency. And it has one or more of the following characteristics – high volume, high velocity, or high variety. Big data comes from sensors, devices, video/audio, networks, log files, transactional applications, web, and social media - much of it generated in real time and in a very large scale. Analysing big data allows analysts, researchers, and business users to make better and faster decisions using data that was previously inaccessible or unusable. Using advanced analytics techniques such as text analytics, machine learning, predictive analytics, data mining, statistics, and natural language processing, businesses can analyse previously untapped data sources independent or together with their existing enterprise data to gain new insights resulting in significantly better and faster decisions. (Big Data Analytics, 2014).

BI tools such as MicroStrategy, Tableau, IBM Cognos, and others provide business users with direct access to data warehouse insights. First, the business user can create reports and complex analysis quickly and easily using these tools. As a result, there is a trend in many data warehouse sites towards end-user self-service. Business users can easily demand more reports than IT has staffing to provide. More important than self-service however, is that the users become intimately familiar with the data. They can run a report, discover they missed a metric or filter, make an adjustment, and run their report again all within minutes. This process results in significant changes in business users' understanding the business and their decision-making process. First, users stop asking trivial questions and start asking more complex strategic questions. Generally, the more complex and strategic the report, the more revenue and cost savings the user captures. This leads to some users becoming "power users" in a company. These individuals become wizards at teasing business value from the data and supplying valuable strategic information to the executive staff. Every data warehouse has anywhere from two to 20 power users.

Query performance with BI tools lowers the analytic pain threshold. If it takes 24 hours to ask and get an answer, users only ask once. If it takes minutes, they will ask dozens of questions. For example, a major retailer was comparing stock-on-hand to planned newspaper coupon advertising. Initially they ran an eight-hour report that analyzed hundreds of stores. One power user saw they could make more money if the advertising was customized for stores by geographic region. By adding filters and constraints and selecting small groups of regional stores, the by-region query ran in two

By: Kiril Anastasov L1087591@live.tees.ac.uk

minutes. They added more constraints and filters and ran it again. They discovered that inventory and regional preferences would sell more and increase profits. Wherean eight-hourquery was discouraging, two-minute queries were an enabler. The power user was then willing to spend a few hours analyzing each region for the best sales, inventory, and profit mix. The lower pain threshold to analytics was enabled by data warehouse performance andthe interactivity of the BI tools.

Hadoop and the data warehouse will often work together in a single information supply chain. When it comes to Big Data, Hadoop excels in handling raw,unstructured and complex datawith vast programming flexibility. Data warehouses also manage big structured data, integrating subject areas and providing interactive performance through BI tools. It is rapidly becoming a symbiotic relationship. Some differences are clear, and identifying workloads or data that runs best on one or the other will be dependent on your organization and use cases. As with all platform selections, careful analysis of the business and technical requirements should be done before platform selection to ensure the best outcome. Having both Hadoop and a data warehouse onsite greatly helps everyone learn when to use which. (Awadallah, A. Graham D. 2012)

Technologies such as OLAP, spatial analysis, statistical analysis, and predictive analytics are hardly new to data warehousing and business intelligence. However, OLAP products typically have their own calculation engine, statistics products have their own data engine, and predictive analytics products have their own mining engines. In short, an enterprise-wide business intelligence environment could maintain a half dozen different types of ‘data engines’, each requiring their own servers, their own copies of the data, their own management infrastructure, their own security administration, and their own high-availability infrastructure. Each engine has its own API’s and its own set of developer tools and end-user tools. The complexity and cost of replicating entire stacks of BI technologies is significant. Oracle Database provides a completely different approach by, first, continuing to extend the SQL language to perform more calculations within standard SQL and second by integrating analytics inside the database engine. Instead of moving data from a data warehouse to other analytic engines forfurther analysis, Oracle has instead brought the advanced analytic algorithms into its database, where the data resides. Beyond the considerable advantages of consolidating the back-end data architecture of an enterprise business intelligence environment, the integration of analytics within the Oracle Database provides a host of advantages unavailable to stand-alone environments. For example, does your standalone OLAP server scale across large clusters of servers? How easily does your statistics engine integrate into your user authentication server? And can it transparently implement all of your data security policies? How easily can you integrate the results of your spatial analysis with your data warehouse data? Within Oracle Database, all of these issues are solved simply due to the deep integration of analytic capabilities in the database.

By: Kiril Anastasov L1087591@live.tees.ac.uk

Oracle Advanced Analytics offers a combination of powerful in-database predictive-analytics algorithms and open source R algorithms, accessible via SQL and R languages. These analytic capabilities include a dozen data-mining algorithms implemented in the Oracle Database (including algorithms for classification, clustering, regression, anomaly detection, and associations);SQL functions for basic statistical techniques; and tight server-side integration with open-source R to enable R programmers to realize the full performance and scalability of the Oracle database platform and also provide access to the entire functionality of the R ecosystem on data stored in the Oracle database. By providing a range of GUI and IDE options, Oracle Advanced Analytics allows business users, statisticians, and data scientists to tap directly into the large data volumes and extensive processing capacity of the Oracle Database, using an environment appropriate to each end-user. SQL-savvy data scientists can directly use SQL, statisticians with experience in R can continues to write R programs and utilize R-based guis and business users can leverage Oracle's Data Miner extension in SQL Developer. Oracle OLAP is a full-feature online analytical processing (OLAP) engine embedded in the Oracle Database. Oracle OLAP enhances data warehouses by improving query performance (as discussed in the performance section) and by adding enriched analytical content.The core feature of Oracle OLAP is cubes. Managed within the Oracle database, cubes store data within a highly optimized multidimensional format. Cubes provide scalable and compressed storage of dimensional data, fast incremental update, fast query performance, and the ability to compute or store advanced analytical calculations. Oracle's strategy with Oracle OLAP is to bring these core OLAP advantages into the data warehouse. This is achieved be exposing the key capabilities of Oracle OLAP via standard SQL, so that any business intelligence tools or other SQL-based application can leverage OLAP.The Oracle Database is the market leader for data warehousing, built upon a solid foundation of scalability and performance, and augmented by innovative features such as Oracle's unique readconsistency model for near-real-time data warehouses and a flexible and powerful set of in-database analytic capabilities.. The combination of the Oracle Database and an Oracle Exadata storage grid delivers the highest levels of performance for IO intensive workloads, and, with the Oracle Exadata Database Machine, Oracle delivers a complete hardware and software solution for data warehousing. (Oracle Database 12c for Data Warehousing and Big Data, 2013)

NoSQL is often used for storing Big Data. This is a new type of database which is becoming more and more popular among web companies today. Proponents of NoSQL solutions state that they provide simpler scalability and improved performance relative to traditional relational databases. These products excel at storing "unstructured data," and the category includes open source products such as Cassandra, MongoDB, and Redis.(Feinleib, 2012)

By: Kiril Anastasov L1087591@live.tees.ac.uk

10 References:

Awadallah, A. Graham D. (2012), Hadoop and the Date Warehouse: When to Use Which, Available at: https://www.google.co.uk/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&ved=0CF8QFjAG&url=http%3A%2F%2Fwww.teradata.co.uk%2Fwhite-papers%2FHadoop-and-the-Data-Warehouse-When-to-Use-Which%2F%3Ftype%3DWP&ei=8dtYU-rtFMfrPODVgegO&usg=AFQjCNGsjKLOdPaLaxKMx2lU4v7fdVkrdA&bvm=bv.65397613,d.ZWU&cad=rja (Accessed: 04 April 2014).

Big Data Analytics (2014) Available at: http://www-01.ibm.com/software/data/infosphere/hadoop/what-is-big-data-analytics.html(Accessed: 04 April 2014).

Diego (2012), What is the advantage of creating hierarchy in SSAS? , Available at: http://stackoverflow.com/questions/11143694/what-is-the-advantages-of-creating-hierarchy-in-ssas (Accessed: 04 April 2014).

Feinleib, D. (2012) Big Data and NoSQL Available at: http://www.forbes.com/sites/davefeinleib/2012/10/08/big-data-and-nosql-five-key-insights/ (Accessed: 04 April 2014).

Neville, K. (2013), Relationship of Primary Key and Clustered Index) Available at: http://stackoverflow.com/questions/15051869/relationship-of-primary-key-and-clustered-index (Accessed: 04 April 2014).

Oracle Database 12c for Data Warehousing and Big Data (2013) Available at: http://www.oracle.com/technetwork/database/bi-datawarehousing/data-warehousing-wp-12c-1896097.pdf (Accessed: 04 April 2014).

What is index, (2014) Available at http://www.databasecompare.com/What-is-index-(database).html (Accessed: 04 April 2014).

11 Appendix

Go to the link below ⬇

https://github.com/kanastasov/Advanced-Databses.git

For any of the following:

- ✓ **To download the report.**
- ✓ **To download the Specifications.**
- ✓ **To download the Database.**
- ✓ **To download the Mendix model**
- ✓ **To download the C# code for the Entity Framework**

By: **Kiril Anastasov L1087591@live.tees.ac.uk**