**ISCG8049 - Data Warehouse Architecture**

Lecturer: Dr. Sira Yongchareon



Lu, Wang - 1122491

Yang, Langhua - 1452377

Wu, Hao - 1446122

Unitec Institute of Technology

1/10/2015

**(wiseGEEK, 2015)**

Assignment 2 – Technical Report

Contents

[Abstract 2](#_Toc432765876)

[1. Introduction 3](#_Toc432765877)

[1.1. Data Warehouse concept 3](#_Toc432765878)

[1.2. Complete DW solution design 3](#_Toc432765879)

[2. Solution Design and Development 5](#_Toc432765880)

[2.1. Requirements analysis 5](#_Toc432765881)

[2.2. Dimensional Modelling 5](#_Toc432765882)

[2.3. Physical Database Design 7](#_Toc432765883)

[2.4. ETL Design & Development 9](#_Toc432765884)

[2.5. Data Integration 16](#_Toc432765885)

[2.6. Data Quality Assurance 17](#_Toc432765886)

[2.6.1. Data cleaning 17](#_Toc432765895)

[2.6.2. Testing 20](#_Toc432765896)

[2.7. Reporting 21](#_Toc432765897)

[3. Evaluation and Lesson Learned 26](#_Toc432765898)

[4. Conclusion and Future Development 30](#_Toc432765899)

[5. References 31](#_Toc432765900)

# Abstract

This report give the overview of data warehouse methodology including requirement analysis, dimensional modelling, physical data base design, ETL design & development, data cleaning and solution deployment.

In this study, SQL server 2014 was used as the data base, SQL server business intelligence data tool was used for ETL process, and SQL server reporting service was used to answer the business questions queries in report format.

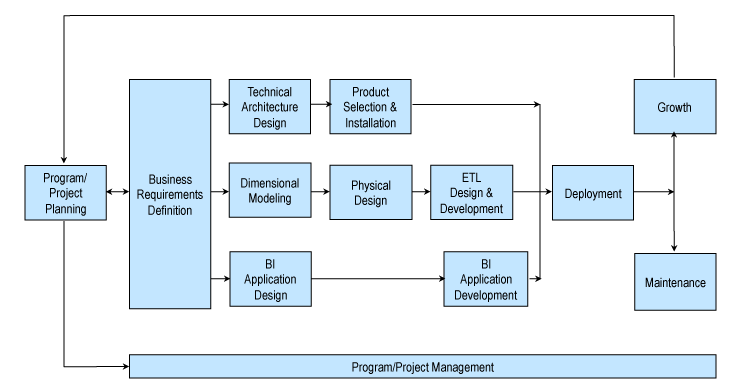
During the project design and development process, data cleaning is the most time consuming part as it requires a lot of testing and data quality rules implementation. By completing the OLAP data base design and implementation, all the procedure in the whole process were fully considerate and practiced.

# Introduction

## Data Warehouse concept

There are several definitions for a data warehouse, while the most popular one came from Bill Inmon, who provides a definition like following: Data warehouse is a relational database with the features of subject oriented, integrated, non-volatile, time variant, and it’s often used for supporting business decision making. Data warehouse often includes historical data which is derived from diversified sources (Chaudhuri & Dayal, 1997). Data warehouse and online analytical processing (OLAP) usually used to support decision making systems or other BI systems. Data in data warehouse is structured to be analysed.

## Complete DW solution design



1. The Kimball lifecycle (Kimball, 2015)

The Kimball lifecycle is a DW development framework which has been widely used after it has been proposed. there are several advantages of Kimball lifecycle, such as easy to query, higher performance, fixable to change, easy to supporting real-time update and so on(Ralph Kimball & Caserta, 2004). In this paper, we will follow the Kimball lifecycle to build our DW solution. According to the Kimball lifecycle, we could see that the following steps should be emphasized:

1. Business requirement definition
2. Dimensional modelling
3. Physical database design
4. ETL (Extract, Transaction, Load)design and development
5. Generate business reporting

The aim of this paper is to design a data warehouse solution by using the Adventure Works2012 sample databases as a case study. And we focus on the sales, so the sales relative business requirements are collected and analysed in this paper and will be given in the followed section. And after the analysis of business requirements, dimensional modelling is used in the design of data warehouse databases to organize the data for efficiency of queries that are intended to analyse and summarize large volumes of data. As we learned from the lectures that a data warehouse includes a fact table and several dimension tables which central to a star schema or a snowflake schema. The ETL design is an essential part of the data warehouse design and is also one of the most annoying challenges confronted by a DW project. ETL system provides extraction, cleaning, conformation, deliverable and management capabilities. After the above steps, we are ready to prepare the information to end users.

SQL Server 2014 provides a wide range of powerful and easy to use tools you can use to create a data warehouse and analyse the data it contains, so in this paper, we choose the SQL Server 2014 to support the data warehouse building. Main while, the SSIS (SQL Server Integration Services) and SSRS (SQL Server Reporting Services) of visual studio are also been used to help the ETL design and the business report.

# Solution Design and Development

## Requirements analysis

Business requirements analysis is one of the most important step when design and develop a data warehouse solution (Abai, Yahaya, & Deraman, 2013). In this paper, because we focus on the Sales-sales focus, so based on the OLTP schema, we select a set of reports that out designed data warehouse will support.

1. **Total sales with customer details in each territory**

This can be used to identify the top customer’s sales, and with customer shopping statics by territory dimension, the sales amount could be evaluated and forecasted.

1. **Sales details for each sales person for each territory**

This can be an important indicators to evaluate the performance of each sales person within one territory.

1. **List of credit sales for each sales person**

This can be used to identify the credit sales of each sales person so that we know the percentage of the credit sales and percentage of cash sales.

1. **List of all discounted sales for each territory**

This could be used to identify that which territory is more sensitive for the price change and then help the business to make price strategies towards different territories.

1. **Sales detail for all products with discount sales quarterly**

This can be used to help the business to evaluate whether the product positioning of each products is proper or not.

## Dimensional Modelling

Dimensional modelling is an important part of data warehouse design data modelling methods which focuses on the particular way of store data to retrieve information in an easy and quick manner. To make the query performs fast and easy to understand, it includes two kinds of tables namely “Fact” and “Dimension”. (R. Kimball & Ross, 2013)

The fact table contains usually the business related numeric data, and the dimension often only stores descriptive elements for those business measures to group, label and filter the information in fact table. (R. Kimball & Ross, 2013) Compare to the traditional data model, dimensional models are utilizing the readability, summarize ability and analysis ability of information.

For the fact table, there are four major types (Ballard et al., 2012):

1. **Transactional**: This is the most essential type for fact table, it is commonly known as each transaction has a single record in table. It is the most basic point by point level with a great number of measures.
2. **Periodic snapshots**: This kind of table contains a snapshot of a table for a specific time. For instance, last year for a particular store.
3. **Accumulating snapshots**: This is for snapshots a well streamlined procedure with a start time and an end time. When the process ends, the order will be closed.
4. **Temporal snapshots**: This is an alternative of daily snapshots, saving a large number of space with the concept of time interval.

It is believed that dimensional modelling is important due to four main aspects (R. Kimball & Ross, 2013):

1. It enables users optimizing the data warehouse performance of complex queries.
2. It minimize the costs for developing and maintenance the data warehouse due to the business processes and rules are simplified by the structure.
3. It contributes documentation and business intelligence tool usage as it organize data in a more user friendly way for self-explanation and understanding.
4. Dimensional data models are the bases for well-structured data marts and well-designed data warehouse.

To organize the data in a more understandable, extend able and high query performance structure, the data model basically structured in different types of schemas of links between dimension and fact table namely star and the improved version – snowflake schema, in which the dimensional table is no moralized for better performance for filter and group queries (Agrawal & Kumar, 2015).

By analysis the business requirements, the schema used in this design is snowflake schema as most of the needed dimensions are all linked directly with sales order fact table, and quarter dimension for year is a optimize of quarterly query question speed. There’s no need to normalize other dimension as the schema provides the best readability and understand ability of the data, the concept design diagram as below:

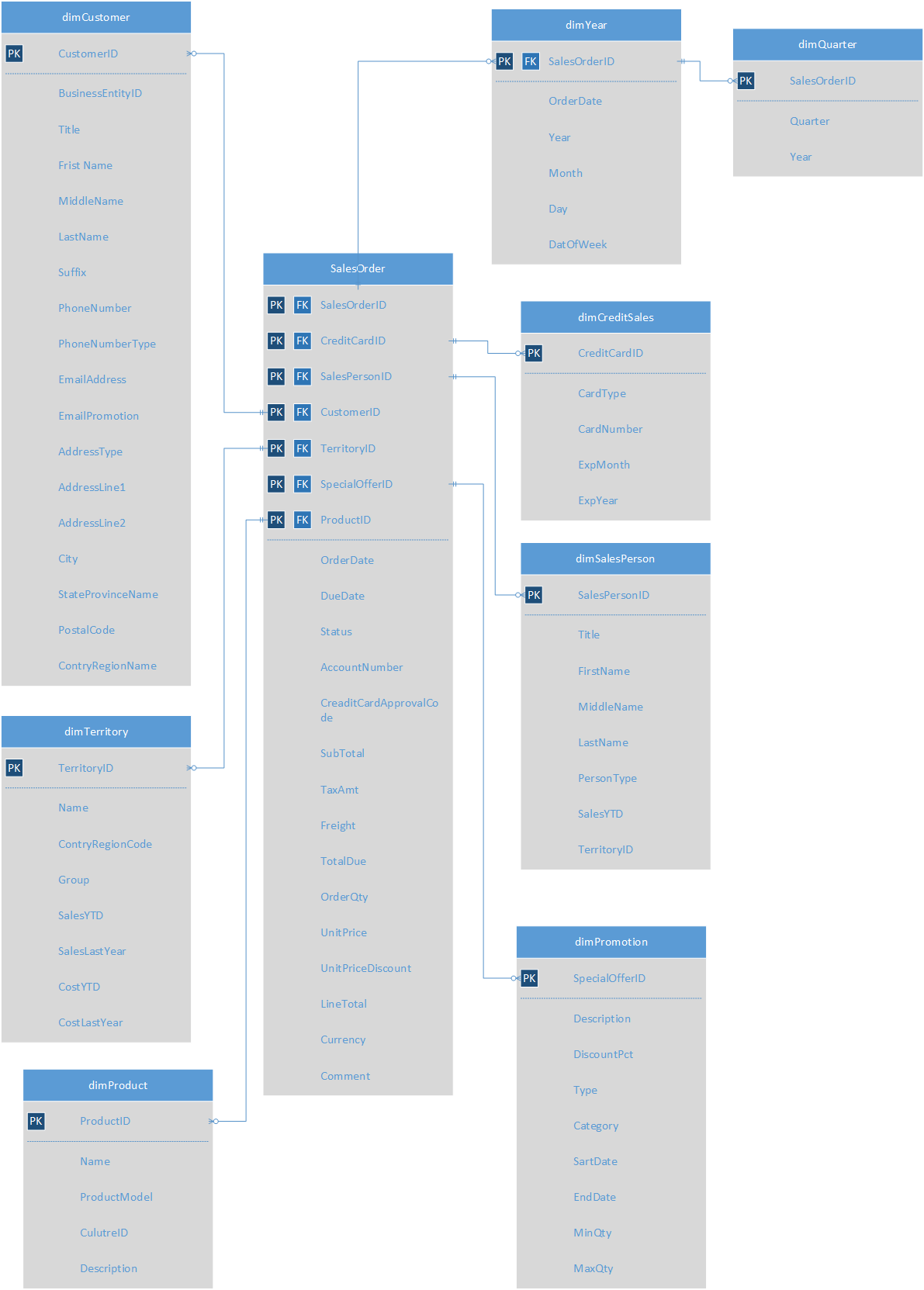


1. Sales focus dimensional modelling design

## Physical Database Design

After the schema, dimensions and facts have been decided, in order to create OLAP database, it is needed to transfer these dimensions and facts into actual table design with details such as, column elements that needed, and OLTP tables that need to be merged.

It is very important to get this physical database design done nicely. DW developer should not only consider what data that able to retrieve from these OLTP tables, but also think over how to achieve high performance, easy query and reporting at the same time (Vaisman & Zimányi, 2014). In this DW solution, only necessary columns has been imported in OLAP database, and the index has been applied to all OALP tables for performance improvement.



1. Physical design for SalesFocus OLAP

According to the requirements, five different business questions need to answers. The snow flake schema has been chosen to link eight dimension tables. The snow flake schema is the schema that support dimension of dimension with fast query, high performance and easy in analysis and reporting (Agrawal & Kumar, 2015).

In the very first step, eight dimension tables has been indemnified, they are: Customer, SalesPerson, Territory, Product, Promotions, CreditSales, Year and Quarter. These eight tables will able to supply all the necessary data that needed for business requirements.

In the second step, a fact table called SalesOrder table has been created to link dimension tables. This SalesOrder table includes measurements elements and all foreign keys of dimension tables. The quarter dimension and year dimension could share the same primary key, so there are seven foreign keys existed at fact table, they are: CustomerID, SalePersonID, ProductID, TerritoryID, SpecialOfferID, CreditCardID and SalesOrderID. All of these foreign keys can be found from OLTP SalesOrderHeader tables, SalesOrderdetails tables and SpecialOfferProducts tables. So this new SalesOrder fact table could be created by merging these three OLTP table.

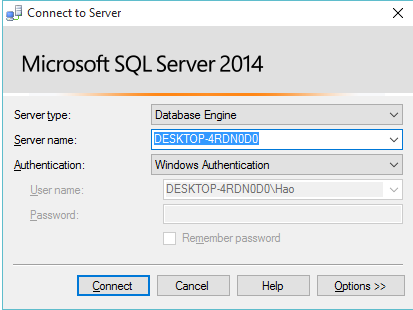
In the third step, the content for all eight dimensional tables have been decided. For example, existing OLTP tables such as: Salesperson, Territory, credit card and Special Offer tables has been chosen to be reused, most elements from these tables were remained at new OLAP tables. For the product OLAP dimensional table, because of the requirements are not focus on much products details, only take a few select elements has been chosen to use. The customer dimension is more complex than others, it involves OLTP tables such as customer, person, address, EmailAddress, and PersonPhone and so on. At last, date elements at SalesOrderHeader table have been extracted to create Year dimension table. And the new quarter dimension was created based on the Year dimension.

## ETL Design & Development

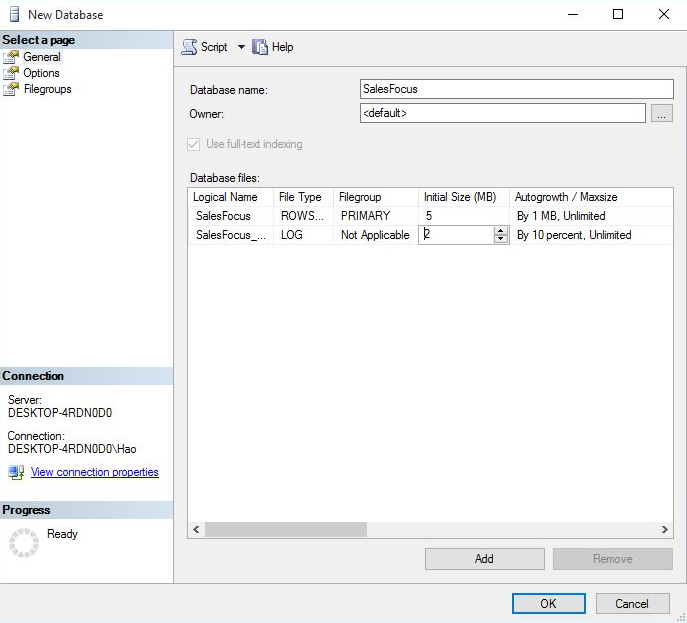
As the core step in our DW solution, ESL process allows us to only (E) extract the data that needed from business requirements,(T) transform and clean these data to make them more readable, meaningful and consistence, and (L) load them into a smaller OLAP database that just for usages of reporting and analysis.

In practise, we were using SQL Server 2014 to store and manage both existing OLTP source database and new OLAP destination databases. And SQL Server Integration Services (SSIS) has been used to development and execute our ETL processes, extracting useful data from source, clean them and load them to redefined destinations. The whole ELT processes and will be described in following Screenshot step by step.

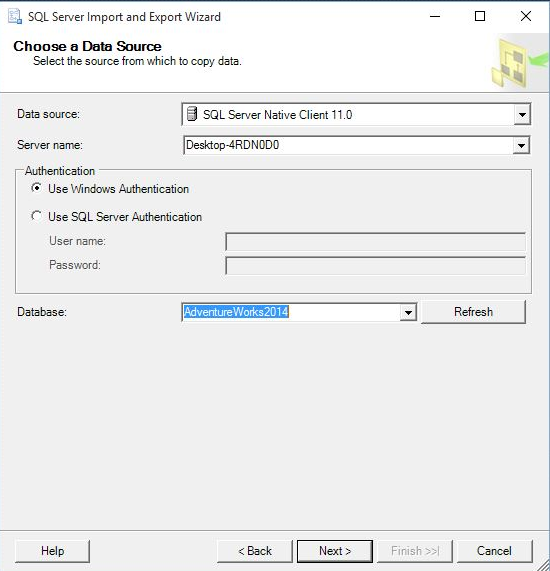
Step 1. Connect and login to SQL Server 2014.



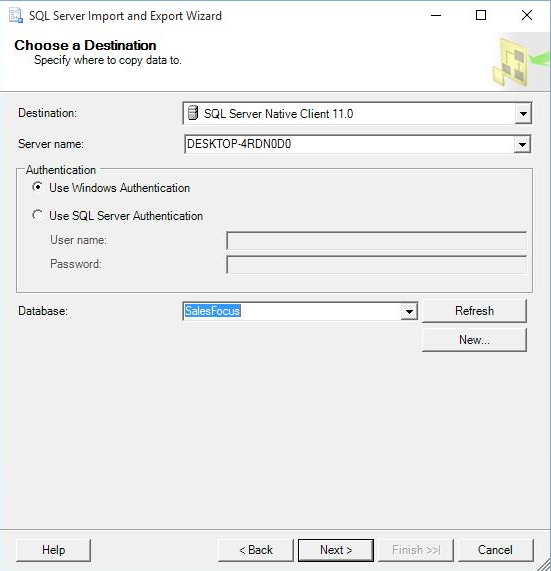
Step 2. Create new OLAP database by right click on the Database folder, then entry OLAP database name.



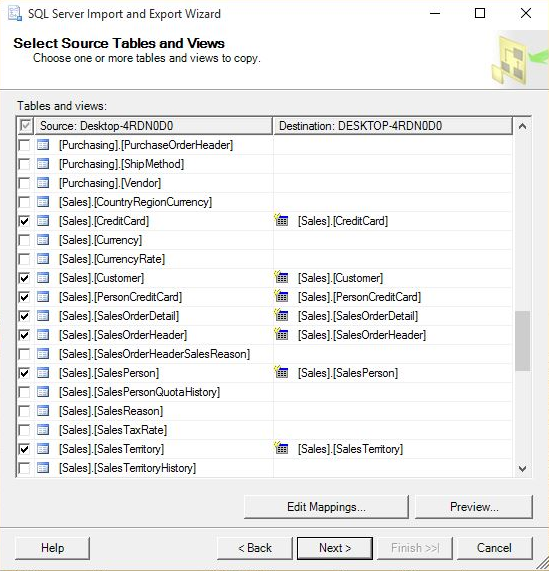
Step 3. Go to the new database, right click and choose tasks and go to import data. Select Server name, and the OLTP database AdventureWorks2014.



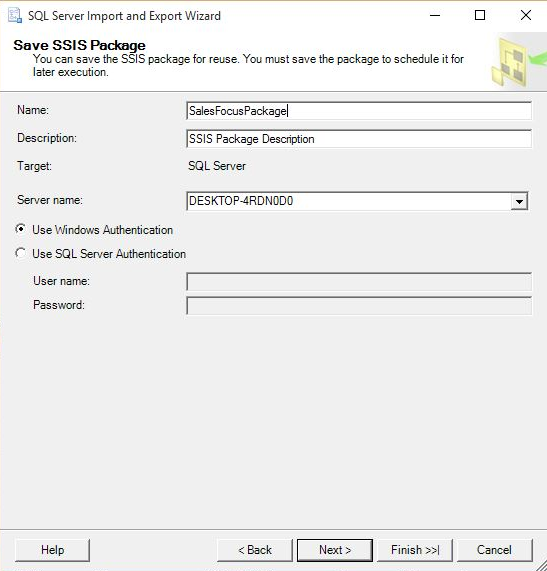
Step 4: Select a destination OLAP database that new created

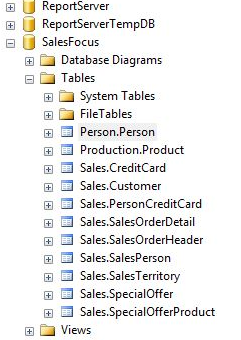


Step 5: Select source tables and views that that ready to be extract into OLAP database through ETL



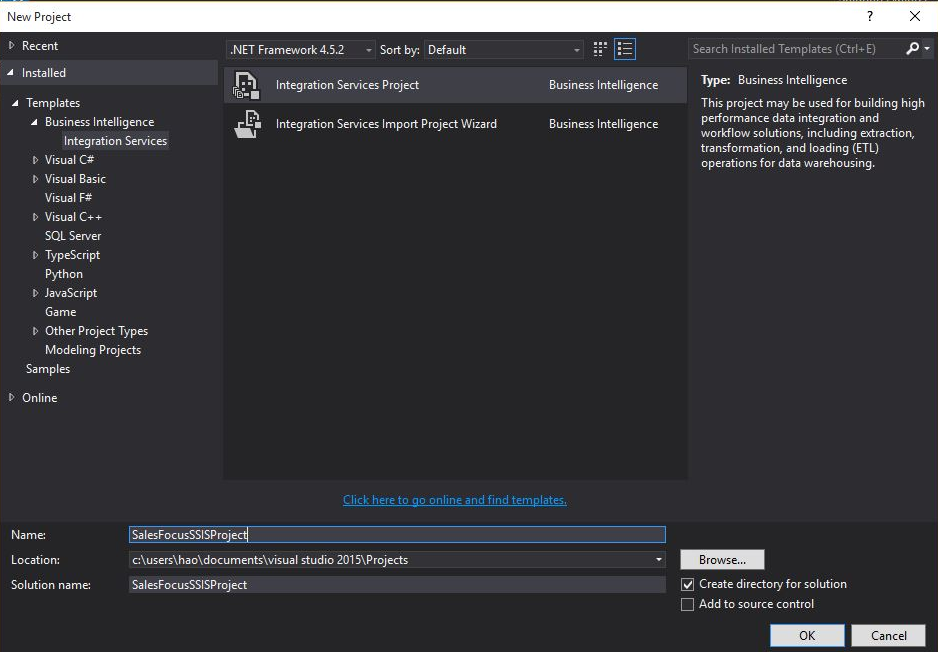
Step 6: Save database as a SSIS Package, so that we could see selected source table are ready to be processed in the newly create OLAP database: SalesFocus





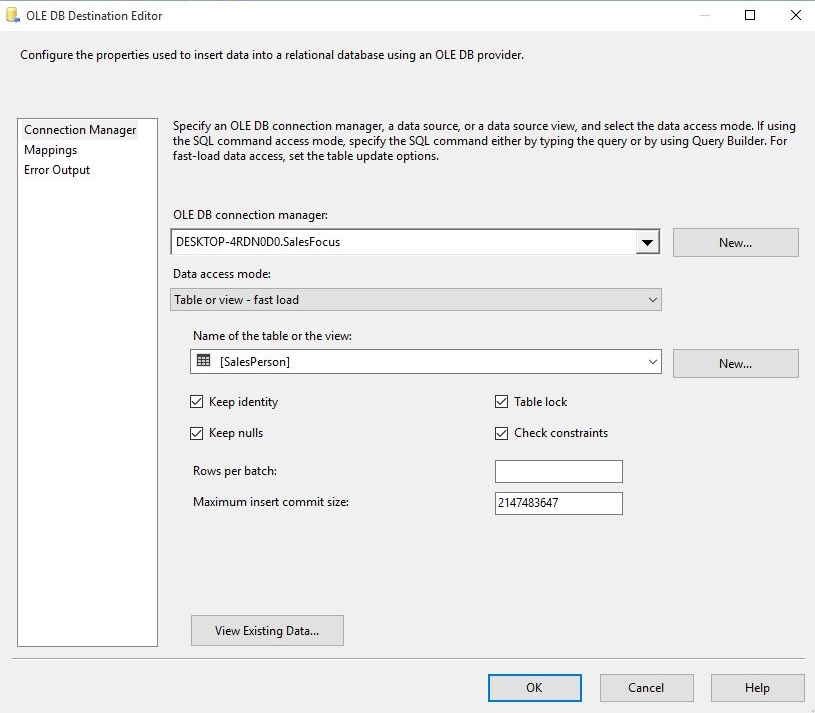
Once the data has been uploaded into SalesFocus OLAP database, we designed and implemented the ETL development process by SQL Server Integrate servers (SSIS), here are our steps with screenshots:

Step 1.Creating the SSIS project

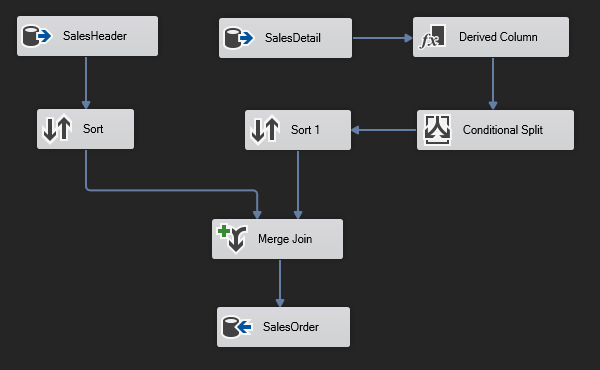


1. Start SQL Server Business Intelligence Development Studio (SSIS), Select New under File menu, and click Project.
2. Under project type, find and click Business Intelligence Project and click Integration Service. Issue a new project name: SalesFocusSSISPrject.

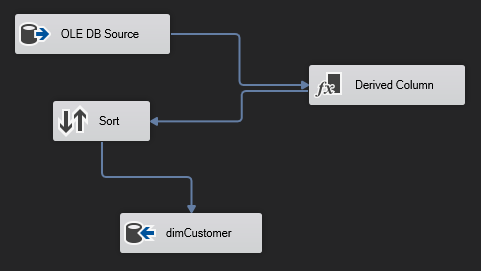
Step 2. Start with ETL design & development by configuring data flow source, transformation nodes and data flow destination.



1. Inside the Tool box of Control Flow page, drag Data Flow Task component into grid space.
2. Under the Tool box menu of BIDS, drag OLE DB Source into the Data Flow design space.
3. Open the OLE DB source by double click, type the server name to connect imported source database, select table that ready to be copy and click OK.
4. Same as above, drag Sort node, merge join node, derived Column node or any other nodes if needed. Open these node to configure transformation logics, such sorting, merging or other value replacing and so on.
5. Drag OLE DB Destination from Data Flow Destination group, configure the destination database source and element mappings and click OK.



The whole ETL design and development processes can be done within this Data Flow control platform by drag and drop. We just need to configure these nodes according the requirements and link them together, and it simple change configure if we have any modification within ETL process.



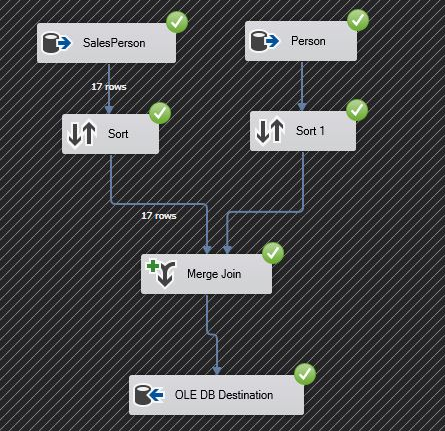
Sort Transformation: this node has been used to make sure input data in the right order, the sort logics can be selected and modified by simply right click.

Merge Join Transformation: this node has been used to join multiple data source table to create new OLAP table. It has been sued to map needed columns from different source into one destination.

Derived Column Transformation: this node has been applied to do value replacing data cleaning task, to check all TerritoryIDs at Salesperson table, and replace all NULL TerritoryIDs to zero. To match this change, one more row has been added at territory table, NULL has been defined as other territory with TerritoryID: zero.

Conditional Split Transformation: this node has been used to filtering imported data by different conditions. As the result, the OLAP imported data won`t contains price or quantity with negative value.

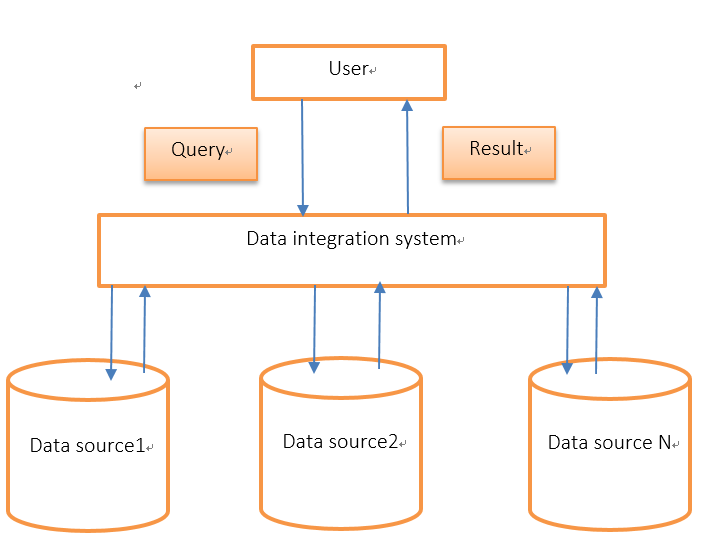
Step3. Execute and deploy the change.



Click the green start button, these ETL processes will be executed, and data will be loaded into the SalesFocus OLAP database.

## Data Integration

Data integration is an essential problems in designing the data warehouse and supporting decision make process. Data integration is the process of logically or physically integrating several disparate data sources into a unified database. The core task of data integration is integrating distributed heterogeneous data sources into an interconnected database, and the data integration allows users to transparently access these data sources. Integration means to maintain data consistency, improve the information sharing and utilization efficiency; transparent manner means that users do not need to know how to implement data access to heterogeneous data sources (Calvanese, De Giacomo, Lenzerini, Nardi, & Rosati, 2001). Data integration system (the schematic diagram of which is showed in figure2), which provides users with a unified data access interface, response user’s requests of accessing to data.



1. The schematic diagram of the data integration system

ETL can be an ideal solution for integrate large volumes of data. Packaged ETL products also offer advanced transformation capabilities.

In general, a data integration solution combines the data which came from diversified data sources, and offers a unified, reconciled organization of these data, which could be effectively queried by the end user. The ultimate goal of a data integration system is to effectively response the queries which are made by the users.

## Data Quality Assurance



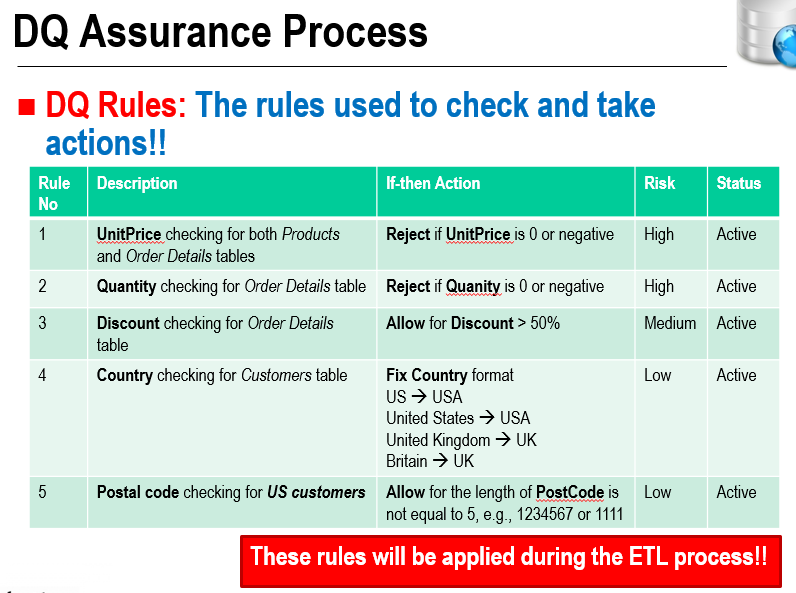
### Data cleaning

Data cleaning is most common used methodology to resolve the data quality problems. Data cleaning is especially useful in dealing with the inconsistence and the duplicate records when integrate diverse data sources(Rahm & Do, 2000).

As showed in the figure, the process of data cleaning could be summarized into the below steps:

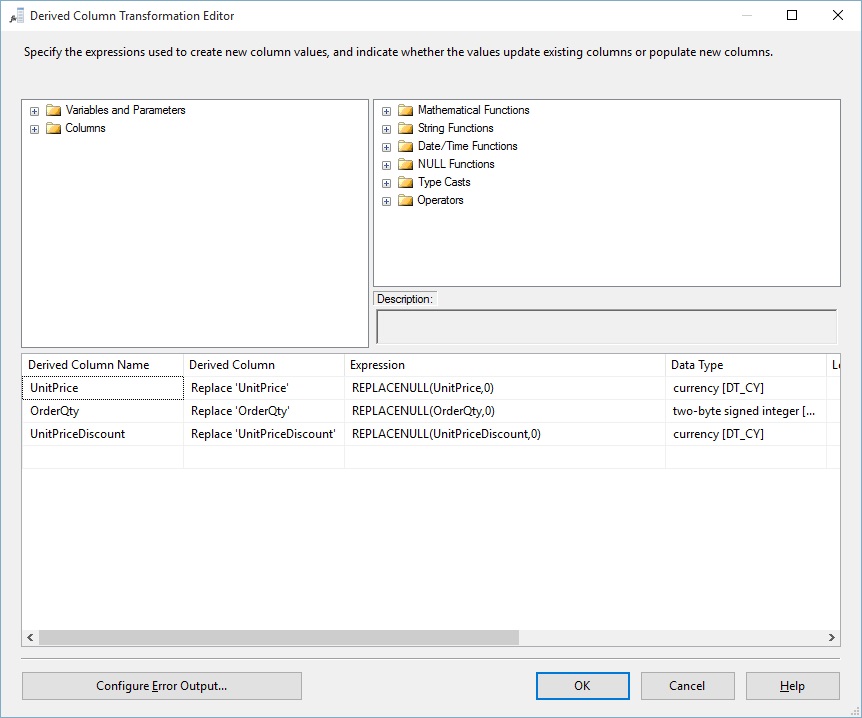
1. The Data cleaning cycles
   * **Data analysis**: in order to decide which sort of errors and inconsistencies should be detected and deal with, data analysis is the first step of data cleaning.
   * **Define work flow and mapping policies**: according to the level of heterogeneity and the category of the dirty data, the corresponding cleaning work should be done.
   * **Verification**: to evaluate the transformation work flow and mapping strategies, verification is necessary.
   * **Transformation**: implement the cleaning work by executing the ETL transformation workflow.
   * **Backflow of the cleaned data**: once the dirty data has been processed, the cleaned data should be back to the sources in order to avoid repeating the data cleaning work.

The rules we used to check and take actions:



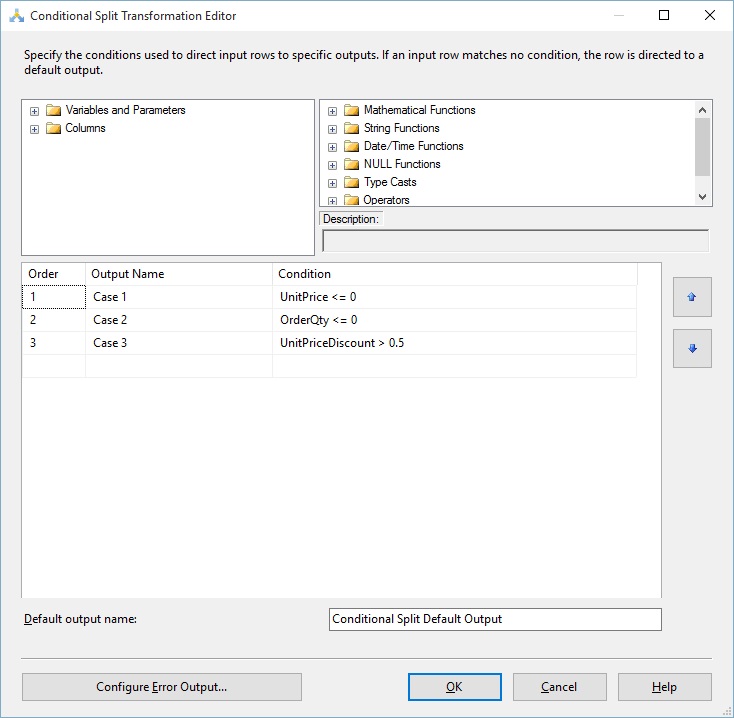
1. Data cleaning rules (Sira, 2015)

For rule1, 2, and 3, firstly, we use the Derived Column tools of the SSIS, to deal with the NULL value by replacing the NULL value using 0.



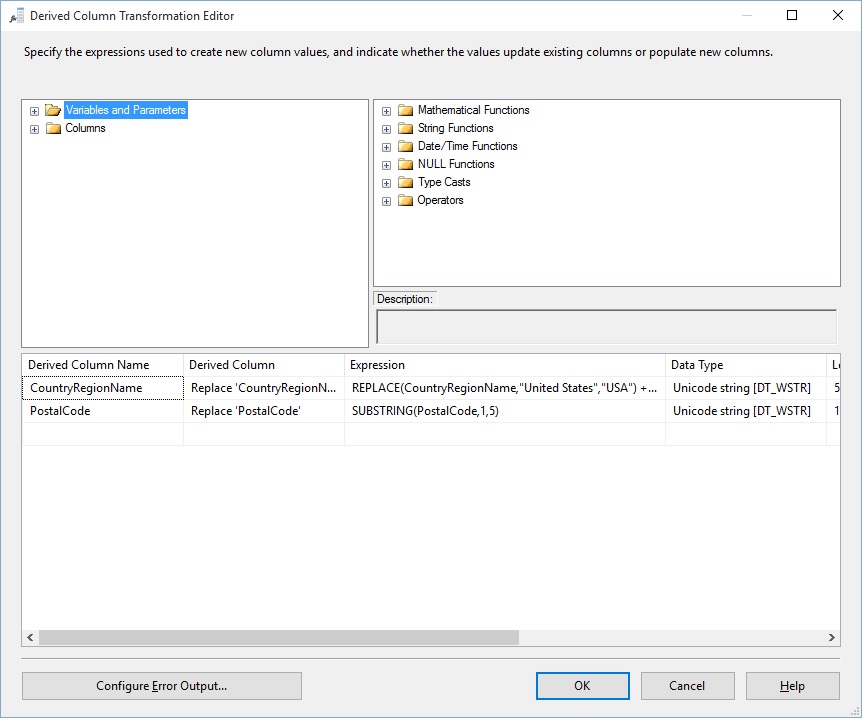
1. Data cleaning of NULL value

Then, using the conditional split of the SSIS, add conditions to filter the values which are not legal.



1. Data cleaning for rule1, 2, and 3

For the data cleaning rule 4 and 5, using the derived column tool, write the corresponding expression to deal with the values which cannot meet the rules.



1. Data cleaning for rule 4 and 5

### Testing

Testing is an important part of data warehouse to ensure that the work you done is correct and efficient. Generally speaking, there are mainly three kind of testing can be addressed on the data warehouse project.

* + **Unit Testing:**

Unit testing mains that every component in the unit testing is separated from each other and every module, such as procedure, SQL Script and so on are all tested. And mostly, the unit testing is implemented by the developer.

* + **Integration Testing**

As the name shows, integration testing integrate different modules of the application and test according to the number of the inputs. And it is often used after the data integration.

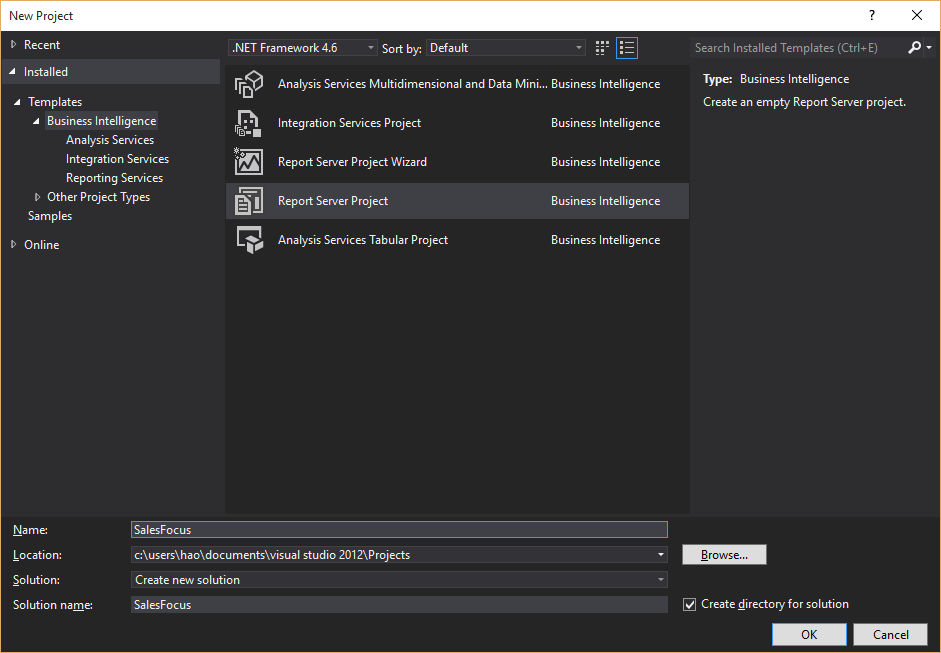
* + **System Testing**

In the system testing, the data warehouse project is tested as a whole and the ultimate goal of the system testing is to evaluate the whole system is working well or not. The common used system testing often focus on the following areas: such as the query performance, the management and scheduling tools, the overnight processing and so on(Berndt, Fisher, Hevner, & Studnicki, 2001).

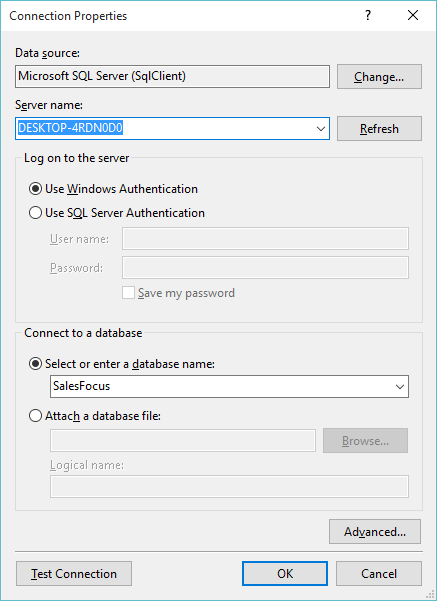
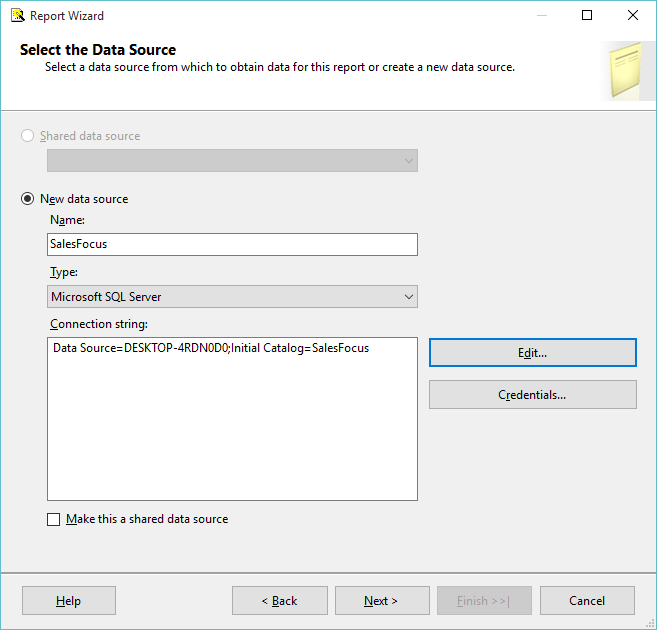
## Reporting

For clearly generate and easily maintain the report queries from the data warehouse, SQL Server Reporting Service (SSRS) was used. In Visual Studio 2012, download and install the Microsoft SQL Server Data Tools - Business Intelligence for Visual Studio 2012.

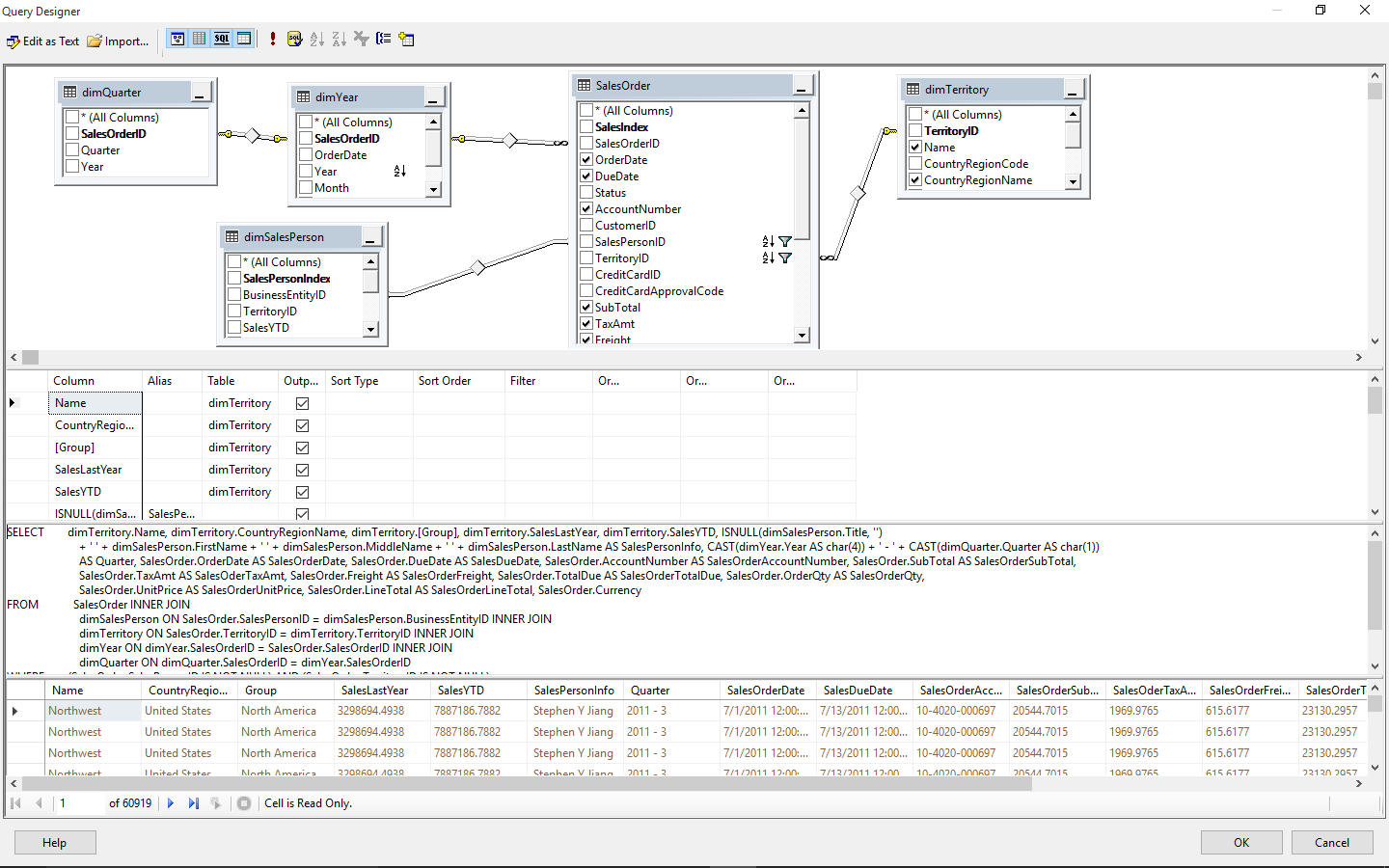
1. Open Visual Studio 2012, open Files -> New Projects ->Business Intelligence -> Reporting Service, select “Report Service Project”, rename the name and click “OK” button.



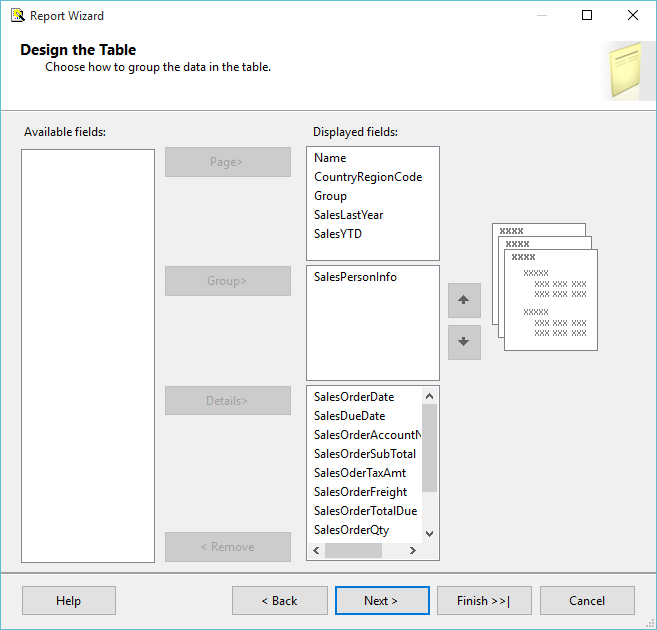
1. In “Solution Explorer”, right click “Reports” and select “Add New Report”.
2. Create a new data source by enter the data source name and connection string of the data source SQL server instance and data base in connection string “Edit” dialog box:



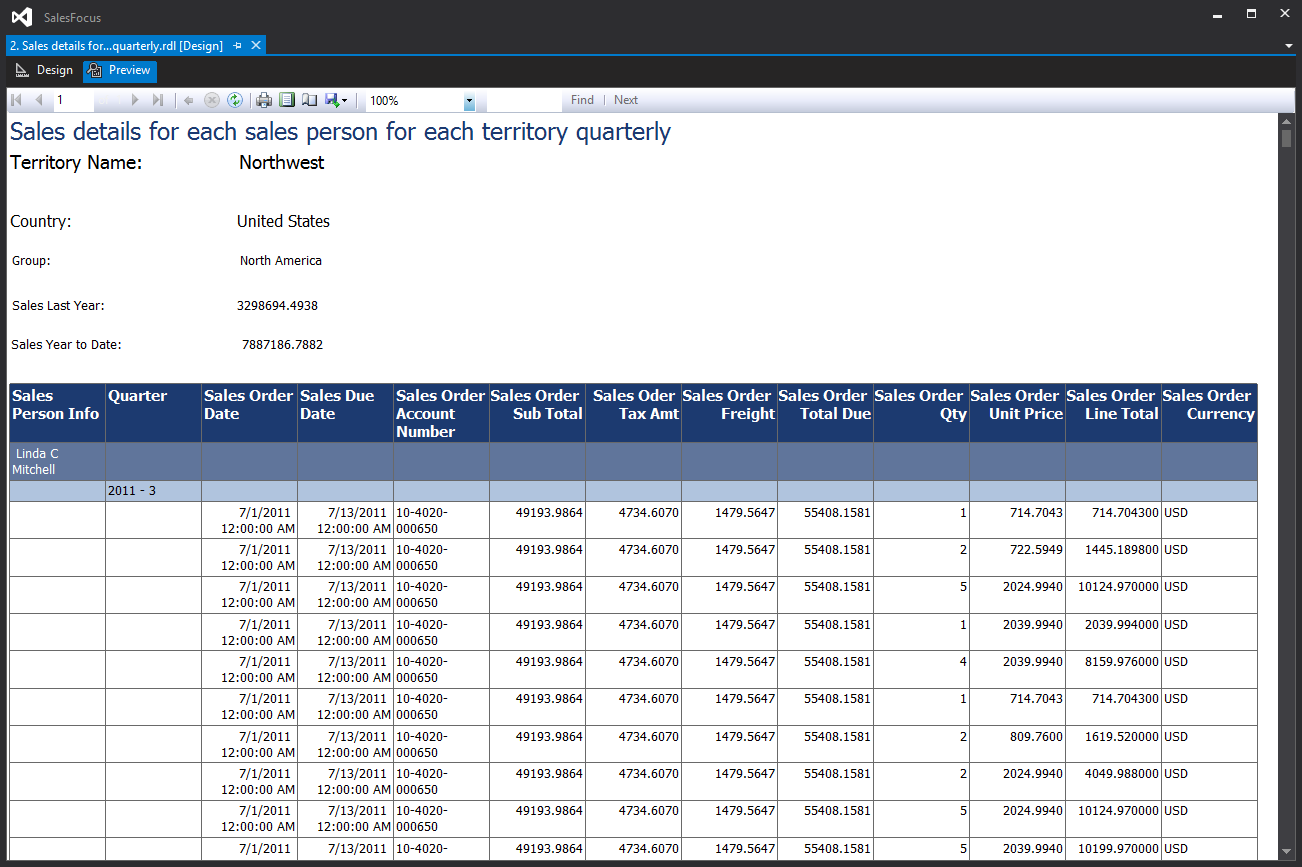
1. Click “Next” and click “Query Builder”, in the Query Designer dialog box, click “Add Tables” to select the tables for the business requirement questions. For instance, select “SalesOrders”, “SalesPerson” and “Territory” table for answering business requirement question “Total sales for each sales person for each territory”.
2. Then select needed columns in each table, modify query SQL scripts to have the demand output.
3. After SQL compile, click “Run” to preview the retrieved data:



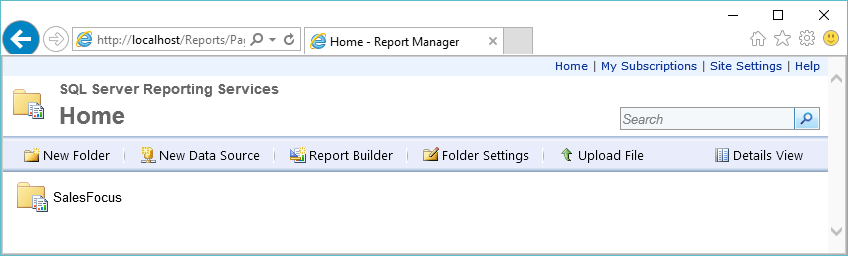
1. In “Select the Report Type” dialog box, select “Tabular”, click “Next”.
2. In “Design the Table” dialog box, select Territory information as the “Page” fields, select sales person information for “Group” field, and other sales order information for “Details”:



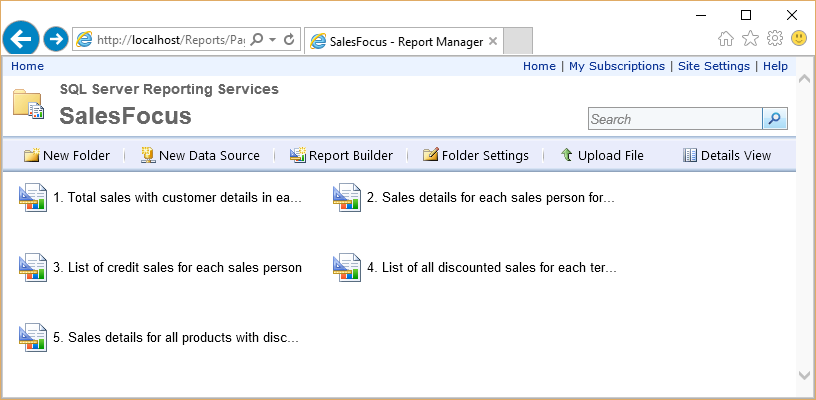
1. In “Choose the Table Layout” dialog box, select “Stepped” and click “Next”.
2. In “Choose the Table Style” dialog box, select “Corporate” and click “Next”.
3. Then, name the report as “Total sales for each sales person for each territory” and click “Finish”.
4. In the Design view, add labels to page fields, and click “Preview: to check the report outputs:



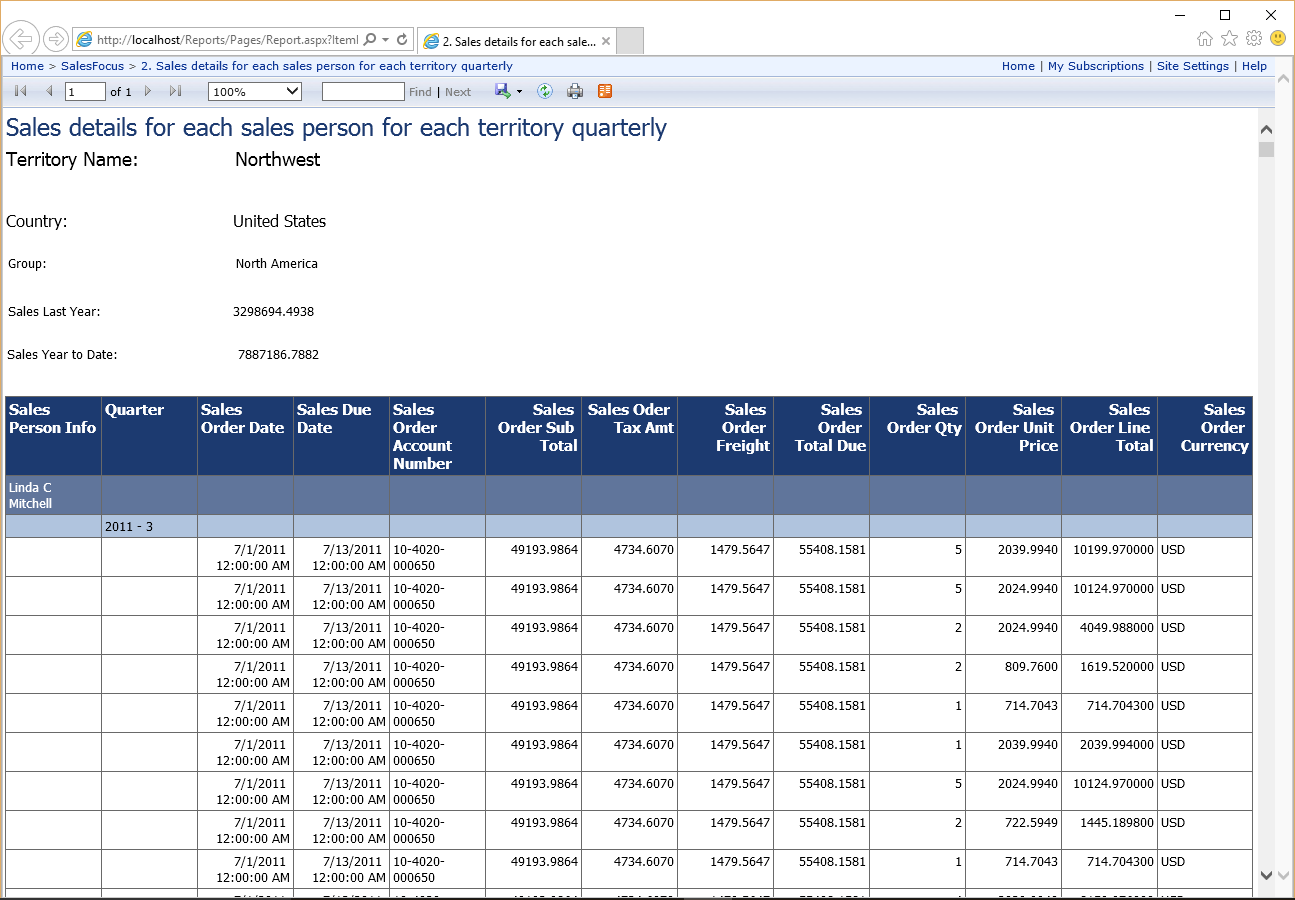
1. Then right click the report in “Solution Explorer” and click “Deploy”.
2. Open Internet Explorer as administrator and go to <http://localhost/Reports>
3. Click the folder “SalesFocus”



1. Click the report just deployed.



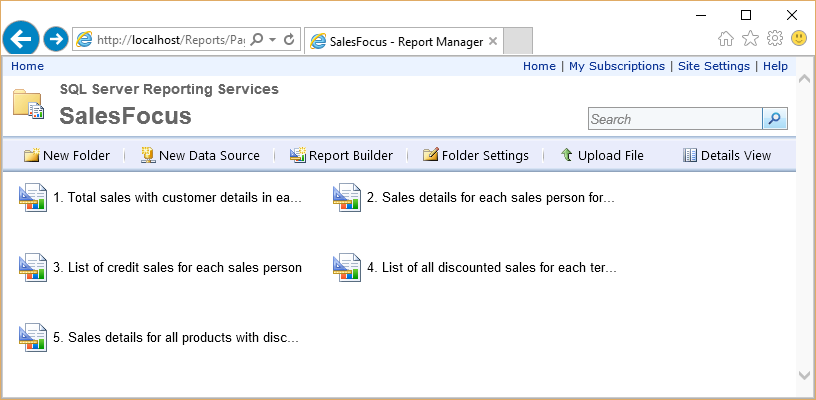
1. Then the report shows via SSRS in browser:

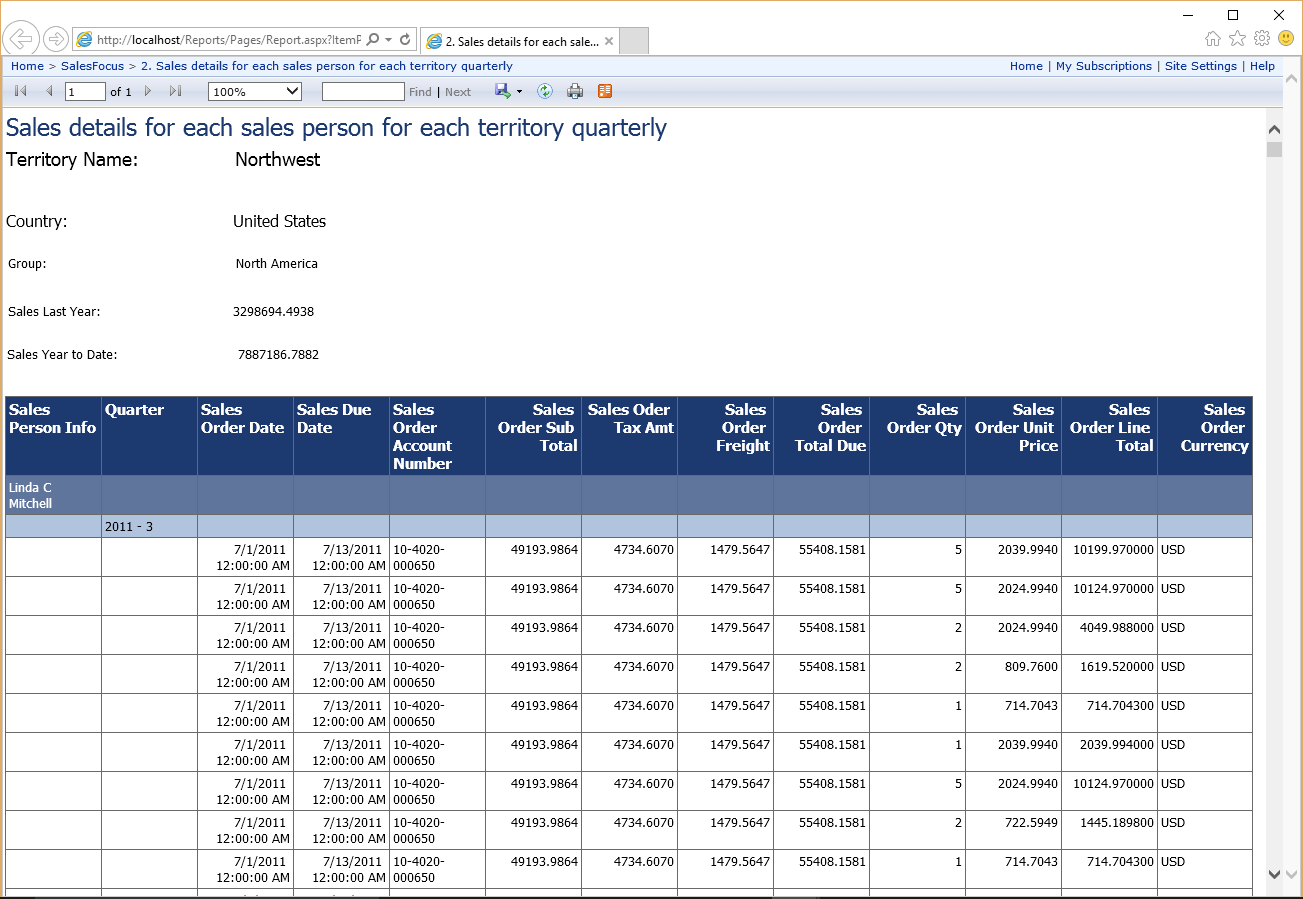
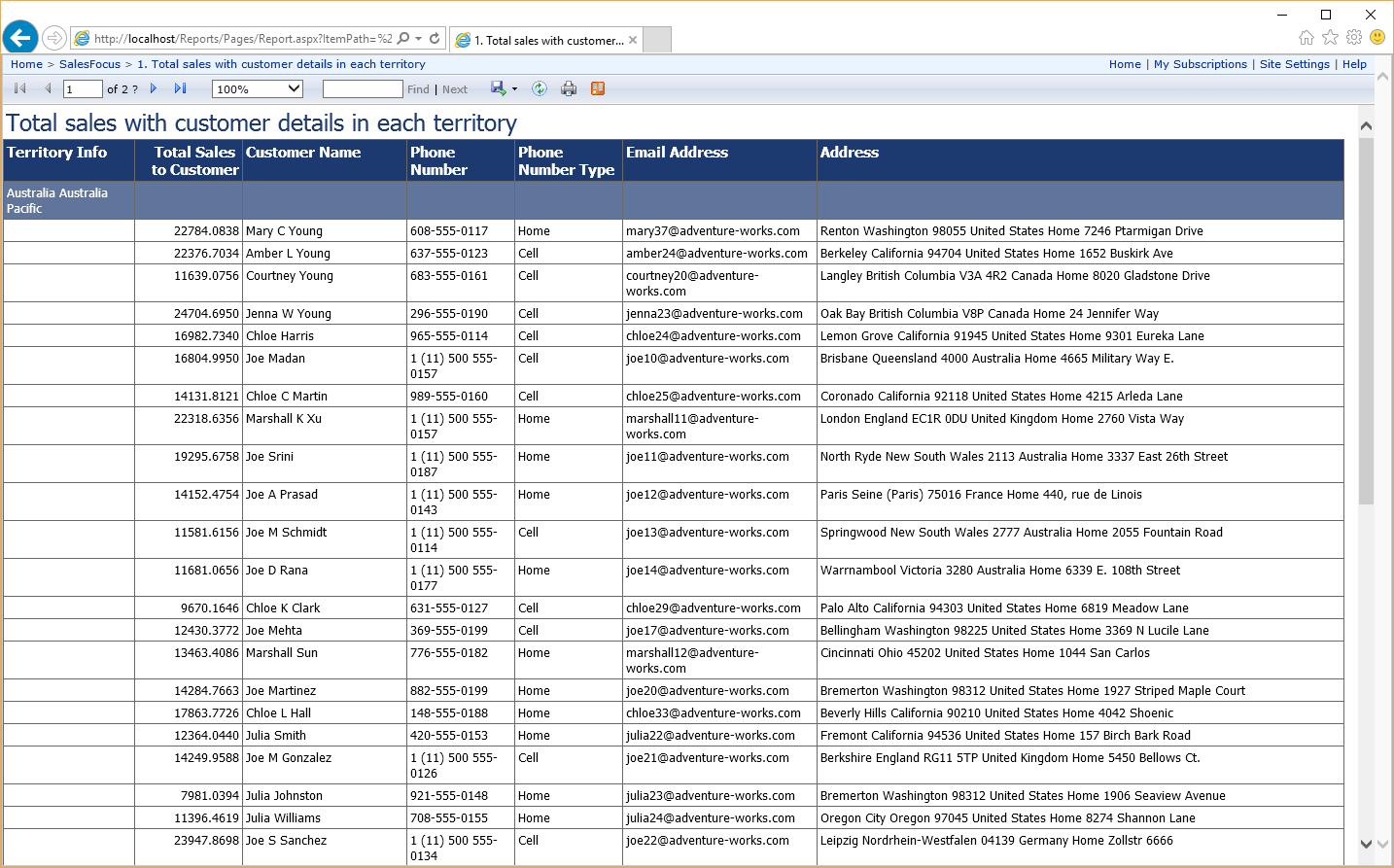


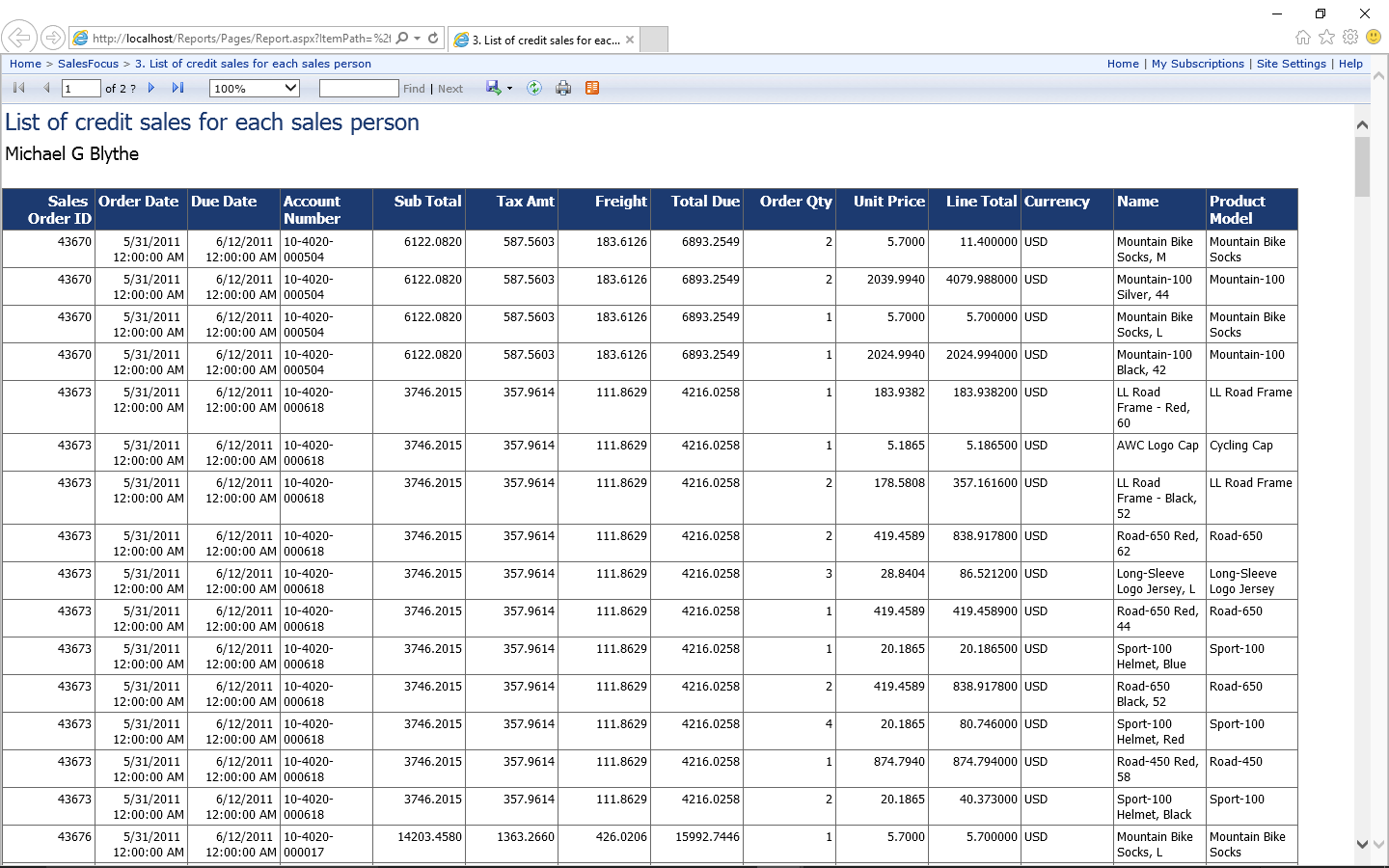
# Evaluation and Lesson Learned

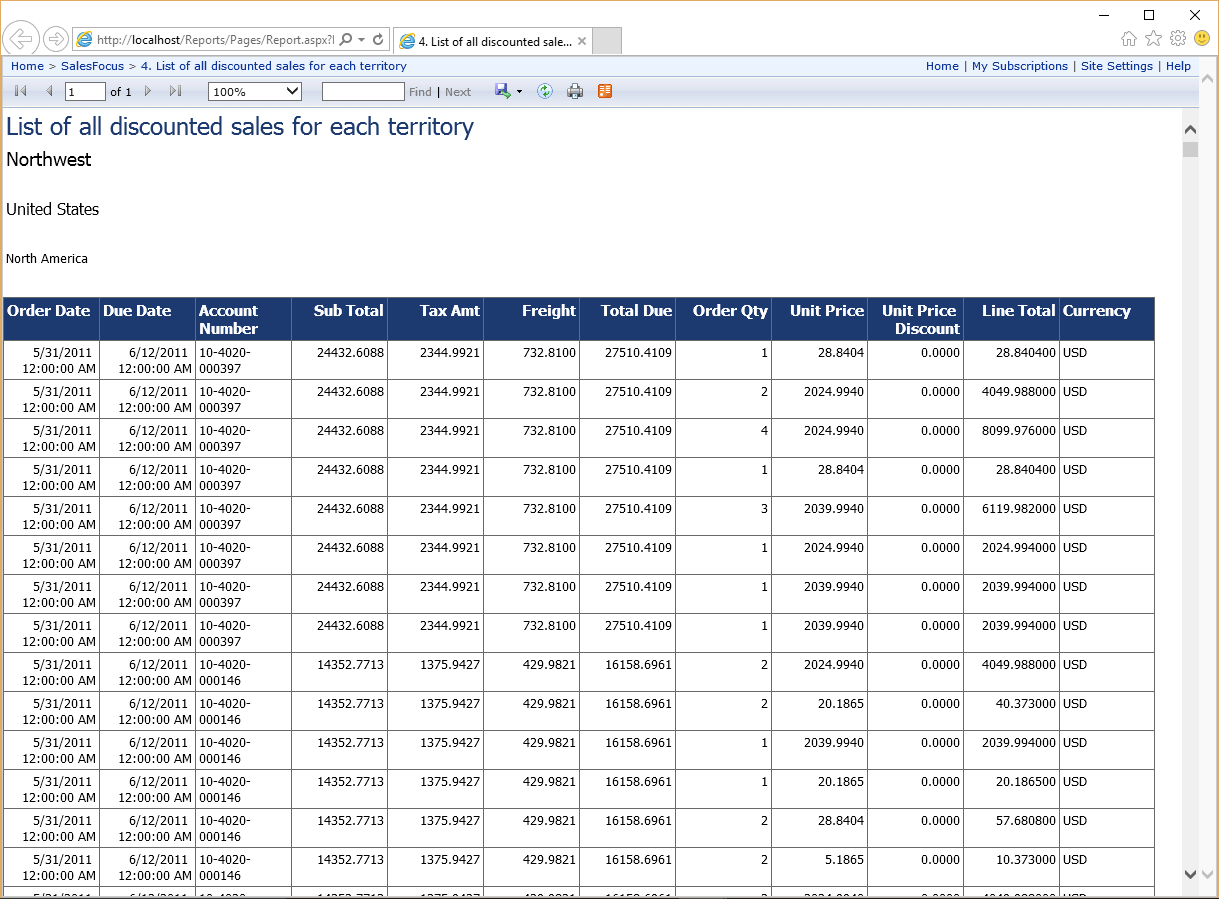
The data warehouse development results were fully satisfied the business requirements identified in a fast response, self-explainable and easy to maintain way, the query responding speed is pretty good, e.g. 60,000 sales records returned within 2 seconds for a three table and two dimension join query.

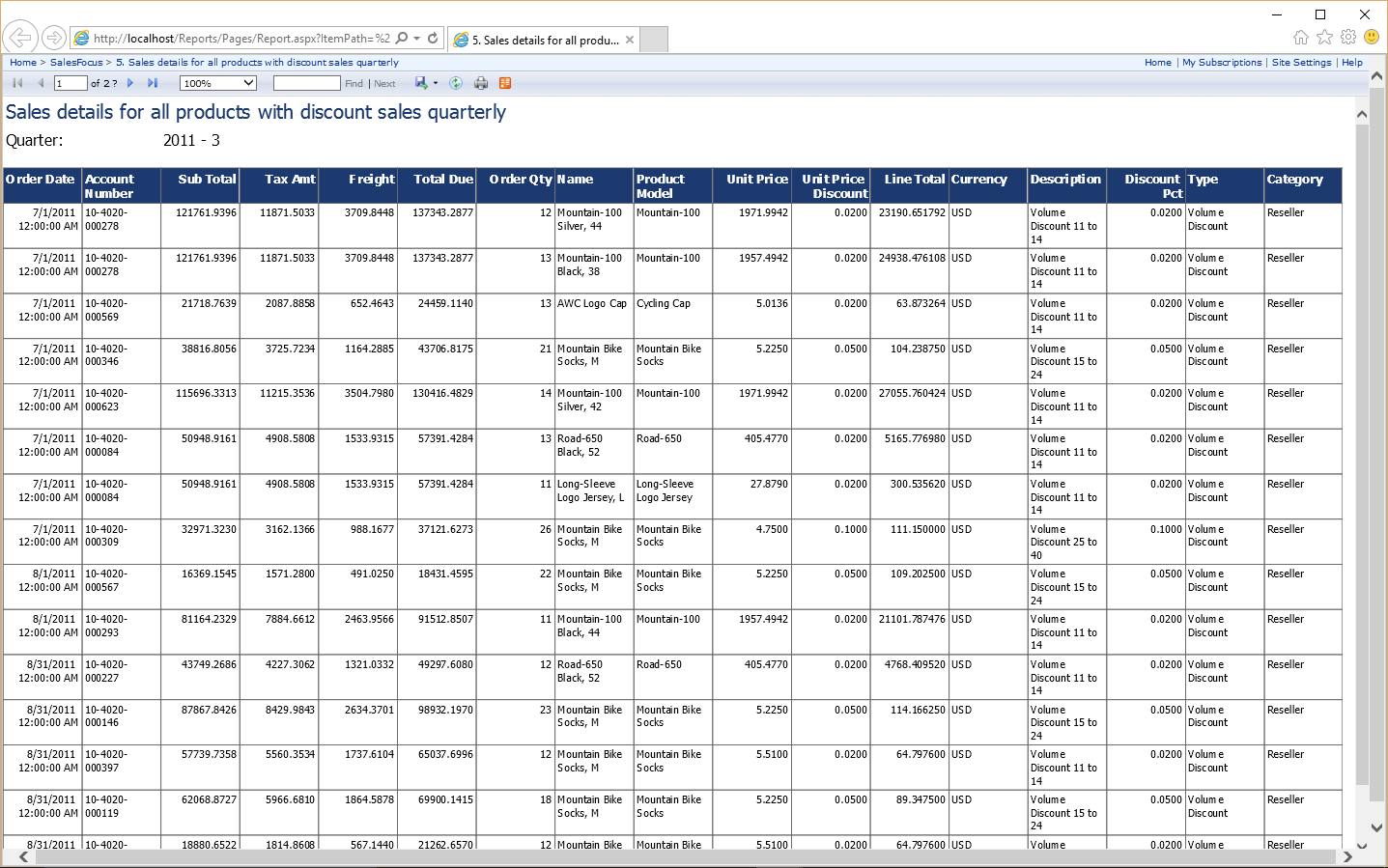
All the five business requirement questions were answered by the deployed SSRS reports, measurements are all contained in fact table, by grouping and filtering using dimension table:











The design and deployments of the DB details such as indexes, constrains, transformation and etc. are the assurance of fast queries, understand ability and easy maintenance.

And the SSRS reporting service via browser makes the access to the reports easy and handy, moreover, the Visual studio SSRS tools made the query management, deployment, update and modify in a more unified and standardized way which is cost effective and agile.

The most important lesson learned from the solution design is the comparison of different design of data structure and optimization, a lot of time and effort were spend on apply different kind of designs and compare their performance and maintainability, many detailed optimization and streamline skills were learned. For example, in dimension table, which columns should be split from fact table and set constrains and indexes to provide best performance. And when design SSRS report, how to optimize SQL scripts, display and group the sales information to get the best performance and readability for users. Moreover, the business intelligence data tool kit provides a large number of useful tools to streamline the design process which make the whole design and deploy process more efficient and effective.

# Conclusion and Future Development

The goal of this project is to creating a data warehouse based on SQL server to answer the five business questions fast and easily, with clear and self-explain data structure and data base table design. The aim also includes easy to maintain, update, and even change the question queries in existing query dimensions as well.

By comparing different design and implementation during the process of design and development, the required business query questions were fully answered and implemented with the best efficient and maintainability, those report could help business manager to analysis their business better, thus grow their business in their strength area.

As a class project, it doesn't involve in any commercial deployment or application. However, it is a great time for us of learning the whole process of data warehouse as well as utility of important terms in it. Moreover, we also discovered the patterns that could help to know a data warehouse in sales industry. Many details that have impact on the sales analysis were learned during the project.

For future development, the partitioning could be implemented in the data warehouse to get better performance and extend ability, the horizontal partitioning and vertical partitioning should be implemented and then stored in different database even servers for different scenarios to compare the performance and situation suitability.

And in the process, there are a lot of knowledge need to learn of data base, SQL scripts, stored procedure and Business Intelligence Data Tools, as more and more work had been done, the more we explore, the more we learn, the more we find out that need to be learn and tried to get a better outcome. Thus, there are a long way towards us to become a good Data Warehouse designer, we will develop our knowledge and skills through more practices in Data Warehouse design and implementation.

# References

Abai, N. H. Z., Yahaya, J. H., & Deraman, A. (2013). User Requirement Analysis in Data Warehouse Design: A Review. *Procedia Technology, 11*, 801-806.

Agrawal, A., & Kumar, S. (2015). Analysis of Multidimensional Modeling Related To Conceptual Level. *Analysis*, 119-123.

Ballard, C., Farrell, D. M., Gupta, A., Mazuela, C., Vohnik, S., & Redbooks, I. (2012). *Dimensional Modeling: In a Business Intelligence Environment*: IBM Redbooks.

Berndt, D. J., Fisher, J. W., Hevner, A. R., & Studnicki, J. (2001). Healthcare data warehousing and quality assurance. *Computer, 34*(12), 56-65.

Calvanese, D., De Giacomo, G., Lenzerini, M., Nardi, D., & Rosati, R. (2001). Data integration in data warehousing. *International Journal of Cooperative Information Systems, 10*(03), 237-271.

Chaudhuri, S., & Dayal, U. (1997). An overview of data warehousing and OLAP technology. *ACM Sigmod record, 26*(1), 65-74.

Kimball. (2015). Kimball DW/BI Lifecycle Methodology. Retrieved from <http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dw-bi-lifecycle-method/>

Kimball, R., & Caserta, J. (2004). *The data warehouse ETL toolkit*: John Wiley & Sons.

Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*: Wiley.

Rahm, E., & Do, H. H. (2000). Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull., 23*(4), 3-13.

Sira. (2015). Data Warehouse Architecture.

Vaisman, A., & Zimányi, E. (2014). *Data Warehouse Systems: Design and Implementation*: Springer.

wiseGEEK. (2015). What does a Data Warehouse Architect do? Retrieved from [http://www.wisegeek.com/what-does-a-data-warehouse-architect-do.htm#](http://www.wisegeek.com/what-does-a-data-warehouse-architect-do.htm)