Project Report

Health Care System Data Warehouse

(CalMedCare)

**Instructor:** Dr. Chong

IT532-01 Data Warehousing & BI

California Lutheran University

Fall Term 2018

by

Yu-Chun, Chou

Neelima Epuru

Neel Bhayani

Najla Alasqah

**Introduction:**

Data Warehousing is a technique for assembling and managing data from various sources for the purpose of answering business questions. Thus, making decisions that were not possible previously. Health Care industry or Hospital Management is the latest organization to include Data Warehousing and BI analytics into their frame. In today’s world, where data is calculated in Peta bytes, every organization is inundated with data and by using the appropriate BI tools, lot of information and knowledge can be obtained for future answers. The hospital management comprises of many divisions, some or independent and some are inter-related with each other. The focus of our project is to build Data Warehouse for Medical-Orders of patients in a hospital which is similar to any Product - Sales data warehouse of any store. (subject -oriented)

**Project Data Source:** To build a Data Warehouse for Cal Med Care hospital systems.

Cal Med Care is the data base which we have created in the IT-509 course during Spring 2018. The database consists of super types, sub types and various attributes, for the purpose of the project to create Data Warehouse, deploy the Cube and run OLAP tools we have considered a subset of the data base which can also be termed as a data mart for Medical Orders received by the hospital. To build any data warehouse from its RDBMS, first we need to identify our Fact table and its Measures. The Fact table measures are Dosage, Quantity and UnitCost and the Dimension tables are Physicians, Patients, Pharmacy, Medication, MedicalOrders and MedicalOrderMedication.

**Business Definition**

Business Challenge:

* Creating a Data Warehouse for Cal Med Care hospital.
* Creating dimension tables and Fact tables using SSAS.
* Loading the fact table using SSIS.
* Deploying the Cube.
* Creating hierarchy for Date dimension table.
* OLAP applications for

- ROLLUP: Retrieves information about total cost of medication sold by year.

- DRILL DOWN: Retrieves information about total cost of medication sold by quarter.

- KPI: Set goals for patient type – resident, outpatient

: Average medication cost for resident patient and outpatient.

- DRILL THROUGH: Retrieves information about the OrderDetails and limits the number of rows to 5

* Reports: From SSAS, exporting to excel and using pivot table, reports are generated which gives medication name, dosage, unit cost and quantity for all years

**Application Tools/ Technologies:**

The tools used to create dimension tables, and fact tables and to load data are SSAS and SSIS.

**SQL Server Analysis Services (SSAS)** is the Microsoft technology to create cubes using databases or data marts/ data warehouses and to develop OLAP (Online Analytical Processing) solutions. The following steps are involved in creating the cube, deploying it and to use analytical tools for querying.

* Design and develop a star-schema
* Create dimensions, hierarchies, and cubes
* Process and deploy a cube
* Develop calculated measures and named sets using MDX
* Browse the cube data using Excel as the client tool

**SQL Server Integration Services (SSIS)** is a Microsoft tool for building enterprise-level data integration and data transformations solutions.

SSIS is also used to solve complex business problems by downloading files, loading data warehouses and managing SQL Server objects and data.

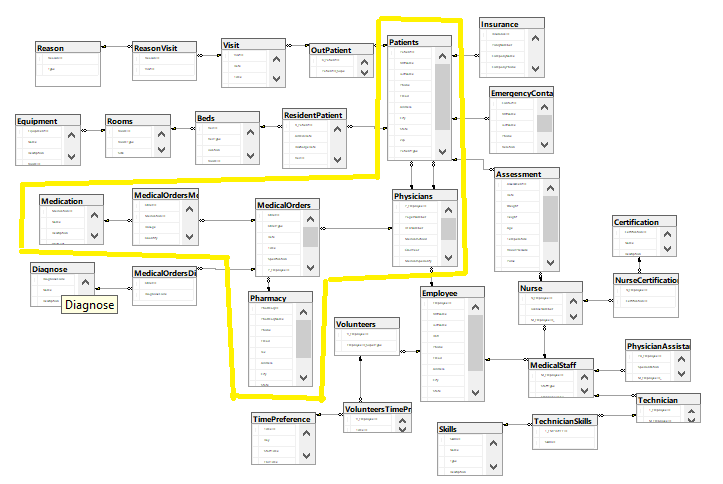
SSIS can extract and transform data from variety of sources like XML data files, flat files and relational data sources and then load the data into the destination file.

**Dimension Model:**

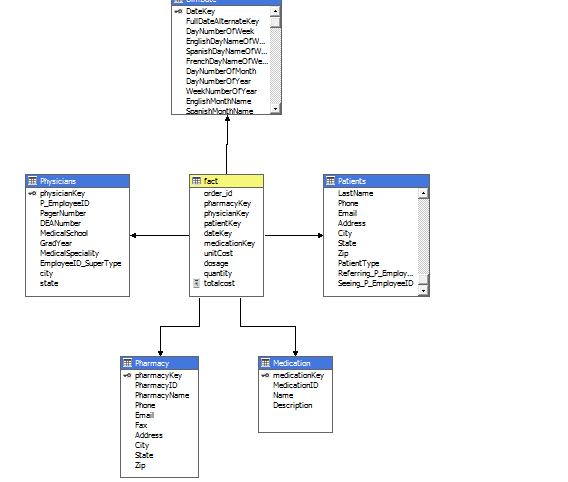
Project cube consists of Fact table with measures as UnitCost, Dosage, Quantity.

Dimension tables consists Physicians table, Patients table, Pharmacy table, Medication table and Date table.

**ERD Schema:**



**Star Schema:**



**Physical Database Design**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table: | Physicians | |  | |
| ColumnName | Type/Length | | Constraints | |
| P\_EmployeeID | char(6) | | Primary Key | |
| PagerNumber | varchar(20) | |  | |
| DEANumber | varchar(8) | |  | |
| MedicalSchool | varchar(30) | |  | |
| GradYear | char(4) | |  | |
| MedicalSpeciality | varchar(30) | |  | |
| Table: | | Patients | |  | |
| Column Name | | Type/Length | | Constraints | |
| PatientID | | char(6) | | Primary Key | |
| FirstName | | varchar(50) | |  | |
| LastName | | varchar(50) | |  | |
| Phone | | varchar(20) | |  | |
| Email | | varchar(60) | |  | |
| Address | | varchar(95) | |  | |
| City | | varchar(30) | |  | |
| State | | char(2) | |  | |
| Zip | | char(5) | |  | |
| PatientType | | char(2) | |  | |
| Referring\_P\_EmployeeID | | char(6) | | Foreign Key of Physicians Table | |
| Seeing\_P\_EmployeeID | | char(6) | | Foreign Key of Physicians Table | |
| Table: | Pharmacy | |  | |
| Column Name | Type/Length | | Constraints | |
| PharmacyID | char(6) | | Primary Key | |
| PharmacyName | varchar(30) | |  | |
| Phone | varchar(20) | |  | |
| Email | varchar(60) | |  | |
| Fax | varchar(10) | |  | |
| Address | varchar(95) | |  | |
| City | varchar(35) | |  | |
| State | char(2) | |  | |
| Zip | char(5) | |  | |

|  |  |  |
| --- | --- | --- |
| Table: | Medication |  |
| Column Name | Type/Length | Constraints |
| MedicationID | char(6) | Primary Key |
| Name | varchar(30) |  |
| Description | varchar(250) |  |

|  |  |  |
| --- | --- | --- |
| Table: | MedicalOrders |  |
| Column Name | Type/Length | Constraints |
| OrderID | char(6) | Primary Key |
| OrderType | varchar(30) |  |
| Date | Date |  |
| Time | Time |  |
| Specification | varchar(30) |  |
| P\_EmployeeID | char(6) | Foreign key of Physician table |
| PharmacyID | char(6) | Foreign key of Pharmacy table |

|  |  |  |
| --- | --- | --- |
| Table: | MedicalOrdersMedication |  |
| Column Name | Type/Length | Constraints |
| OrderID | char(6) | Primary Key ; Foreign Key of MedicalOrders |
| MedicationID | char(6) | Primary Key ; Foreign Key of Medication |
| Dosage | int |  |
| Quantity | int |  |
| Instructions | varchar(30) |  |

|  |  |  |
| --- | --- | --- |
| Table: | Date |  |
| Column Name | Type/Length | Constraints |
| FullDateAlternateKey | date | Primary Key |
| DayNumberOfWeek | tinyint |  |
| EnglishDayNameOfWeek | nvarchar(10) |  |
| SpanishDayNameOfWeek | nvarchar(10) |  |
| FrenchDayNameOfWeek | nvarchar(10) |  |
| DayNumberOfMonth | tinyint |  |
| DayNumberOfYear | smallint |  |
| WeekNumberOfYear | tinyint |  |
| EnglishMonthName | nvarchar(10) |  |
| SpanishMonthName | nvarchar(10) |  |
| FrenchMonthName | nvarchar(10) |  |
| MonthNumberOfYear | tinyint |  |
| CalendarQuarter | tinyint |  |
| CalendarYear | smallint |  |
| CalendarSemester | tinyint |  |
| FiscalQuarter | tinyint |  |
| FiscalYear | smallint |  |
| FiscalSemester | tinyint |  |

**Implementation and Business Report**

**Creating Fact and Dimension Tables:**

SSAS is used to create the dimension tables

# **Create table schema in Data warehouse**

* Create data warehouse in database
* Create database CMC\_DW;
* Create fact table schema

USE [CMC\_DW]

GO

/\*\*\*\*\*\* Object: Table [dbo].[fact] Script Date: 10/26/2018 11:04:16 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[fact](

[order\_id] [char](6) NULL,

[pharmacyKey] [char](6) NULL,

[physicianKey] [char](6) NULL,

[patientKey] [char](6) NULL,

[dateKey] [int] NULL,

[medicationKey] [char](6) NULL,

[unitCost] [money] NULL,

[dosage] [int] NULL,

[quantity] [int] NULL

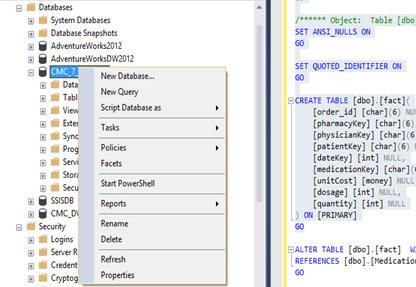
) ON [PRIMARY]

GO

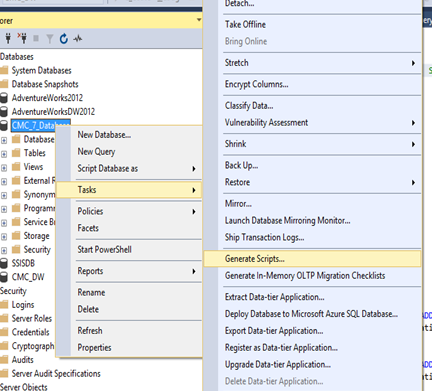
* Use SSMS script generate function to generate schema from CMC\_7\_Database

|  |  |
| --- | --- |
| Medication | Schema and data |
| Pharmacy | Schema and data |
| Physicians | Schema and data |
| Patients | Schema and data |
|  |  |

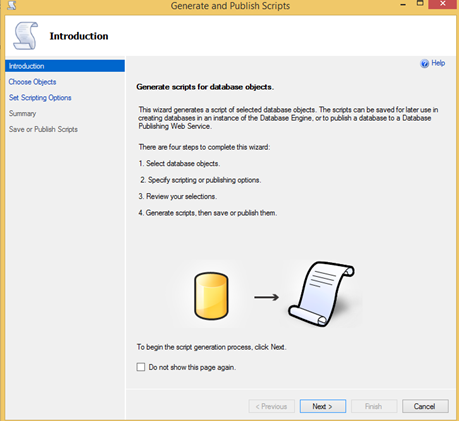
1. In SSMS, right click CMC\_7\_Database



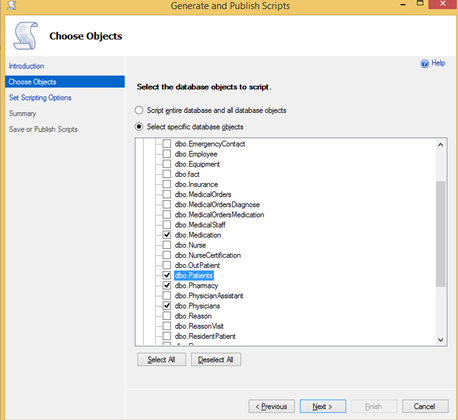
1. Choose Tasks🡪Generate Scripts



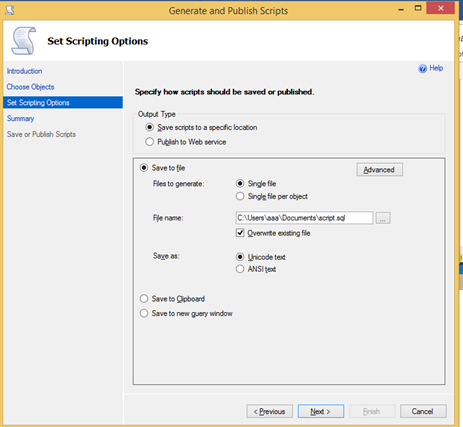
1. Click next button



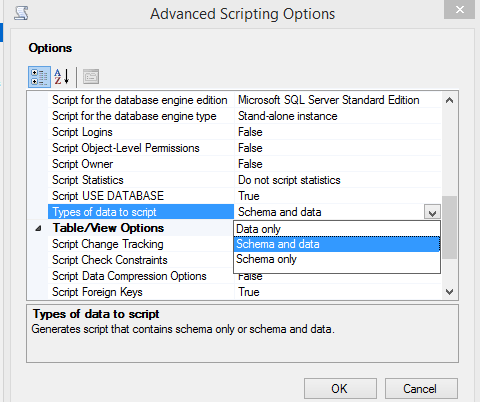
1. Click select specific database objects , and choose medication,patients,pharmacy,physicians ,click next button



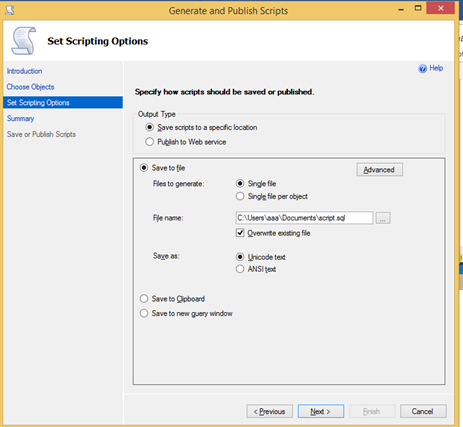
1. Click Advanced



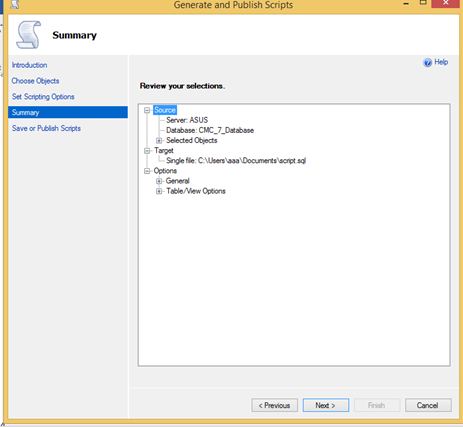
1. In ‘types of data to script’ option, select ‘schema and data’, click OK



1. Choose the script location, click next button



1. Click next generate the script



1. In CMC\_DW, run the generate script to create dimension schema and data

* Adjust physician schema

USE [CMC\_DW]

alter table physicians

add city varchar(35)

alter table physicians

add state char(2)

* Add dimension key column

USE [CMC\_DW]

ALTER TABLE pharmacy

add pharmacyKey char(6);

ALTER TABLE physicians

add physicianKey char(6);

ALTER TABLE medication

add medicationKey char(6);

ALTER TABLE patients

add patientKey char(6);

* Use the original ID to generate dimension table key

USE [CMC\_DW]

update pharmacy set pharmacyKey = substring(pharmacyID,3,100)

update physicians set physicianKey = substring(p\_employeeid,2,100)

update patients set patientKey = substring(patientID,2,100)

update medication set medicationKey = substring(medicationID,2,100)

* Use SSMS script generate function to generate schema from AdventureWorksDW2012

|  |  |
| --- | --- |
| DimDate | Schema and data |
| DimGeography | Schema and data |

* Modify dimension schema to DROP original primary key

//must run statement after statement

ALTER TABLE patients

DROP constraint Patients\_FK1;

ALTER TABLE patients

DROP constraint Patients\_FK2;

ALTER TABLE pharmacy

DROP constraint pharmacy\_pk;

ALTER TABLE physicians

DROP constraint physicians\_pk;

ALTER TABLE medication

DROP constraint medication\_pk;

ALTER TABLE patients

DROP constraint patients\_pk;

* Modify dimension schema to add new primary key

//must run statement after statement

ALTER TABLE Pharmacy

alter column pharmacykey char(6) NOT NULL;

ALTER TABLE pharmacy

ADD PRIMARY KEY (pharmacykey);

ALTER TABLE physicians

alter column physiciankey char(6) NOT NULL;

ALTER TABLE physicians

ADD PRIMARY KEY (physiciankey);

ALTER TABLE medication

alter column medicationkey char(6) NOT NULL;

ALTER TABLE medication

ADD PRIMARY KEY (medicationkey);

ALTER TABLE patients

alter column patientkey char(6) NOT NULL;

ALTER TABLE patients

ADD PRIMARY KEY (patientkey);

**ETL (Extract, Transform, Load) Data**

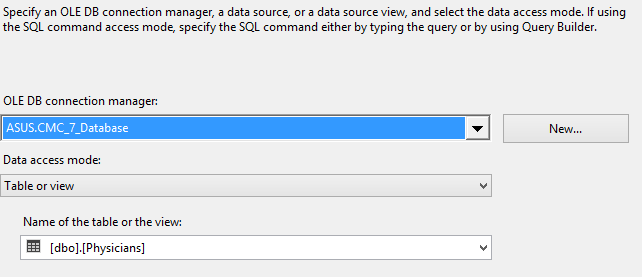
SSIS is used to create Fact table and load the data into Fact table.

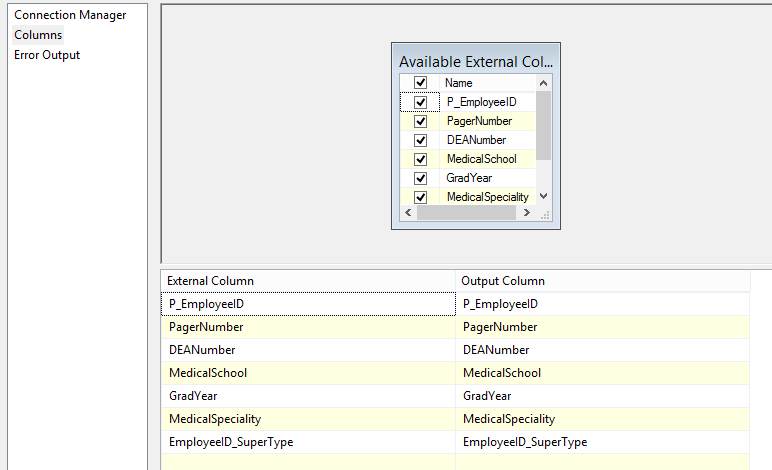
* ETL dimension table Physicians data

1. Create SSIS project
2. Create connection manager

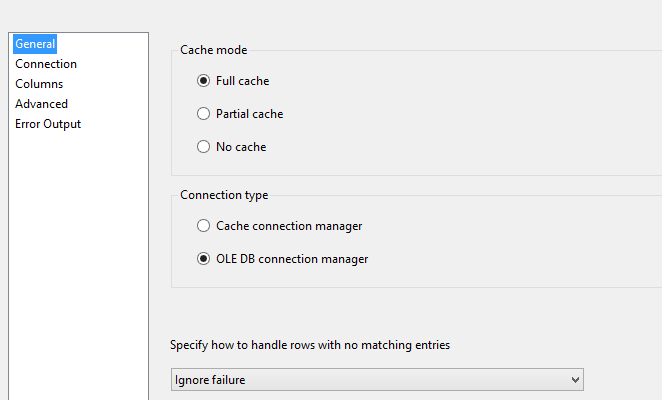
|  |  |
| --- | --- |
| CMC\_7\_Database |  |
| CMC\_DW |  |

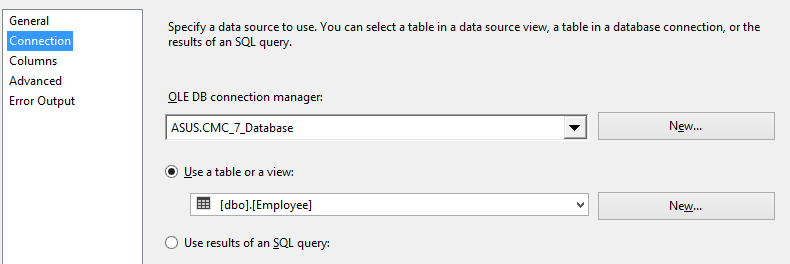
1. Create control flow , data flow
2. In data flow , choose CMC\_7\_Database connection manager as source as below

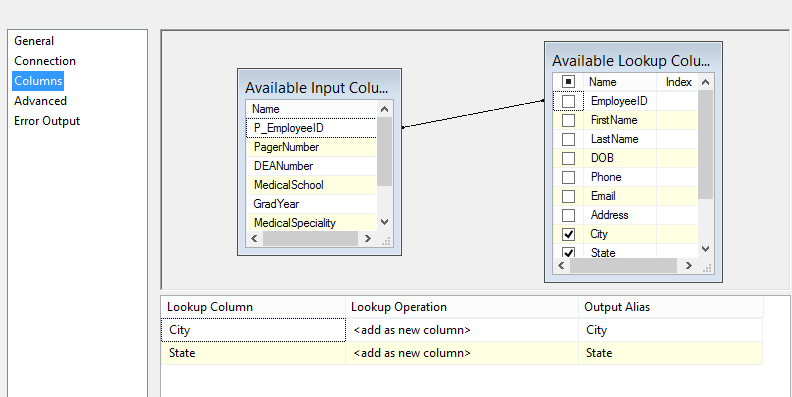




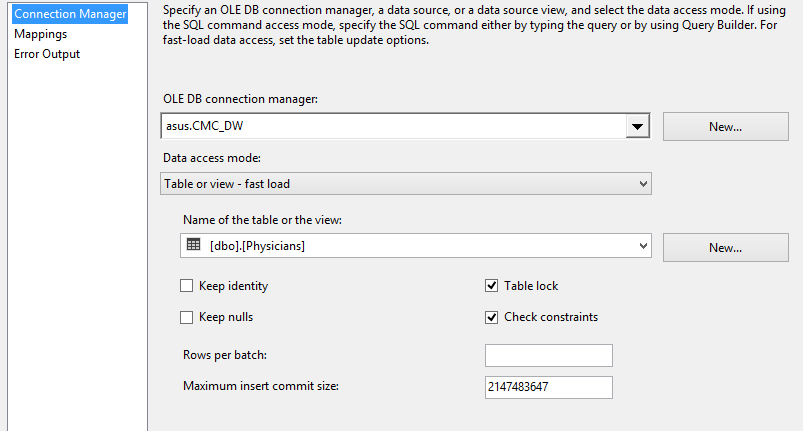
1. In data flow, create a lookup by the settings as below







1. In Data Flow, choose CMC\_DW connection manager as destination



* ETL fact table data

1. Use the setting as below

|  |  |
| --- | --- |
| Source | * CMC\_7\_Database * Table : medicalOrder |
| lookup | MedicialOrdersmedication  Join : order\_id  Output : medicationID , Dosage,Quantity |
| lookup | medication  Join : medication\_id  Output : unit\_cost |
| lookup | pharmacy  Join : pharmacy\_id |
| lookup | physicians  Join : P\_employee\_id |
| lookup | Patients  Join : P\_employee\_id🡪seeing\_ P\_employee\_id  Output :Patient\_id |
| lookup | Date  Join :datekey |
| Destination | * CMC\_DW * Table : fact |

c. Modify fact table FK data

update fact set pharmacyKey = substring(pharmacyKey,3,100)

update fact set physicianKey = substring(physicianKey,2,100)

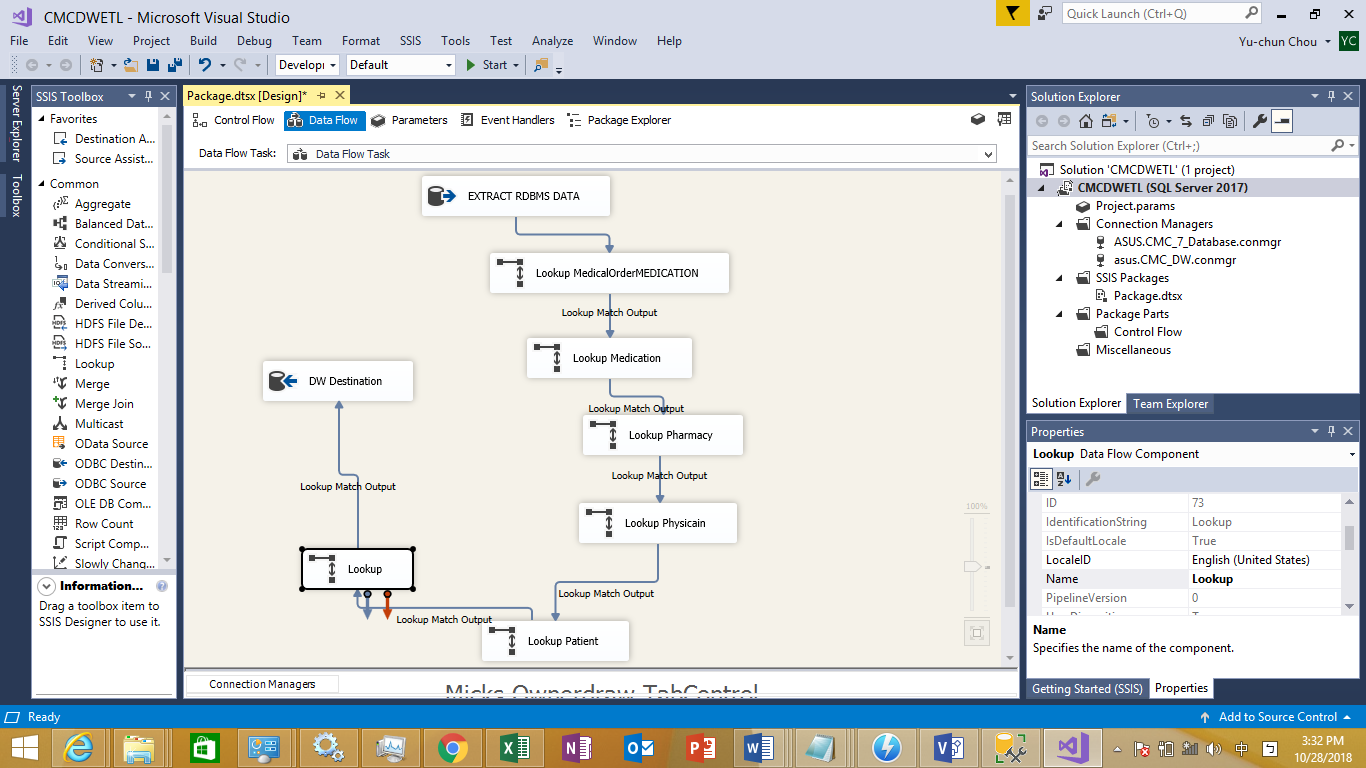
update fact set patientKey = substring(patientKey,2,100)

update fact set medicationKey = substring(medicationKey,2,100)

* Generate mock date data

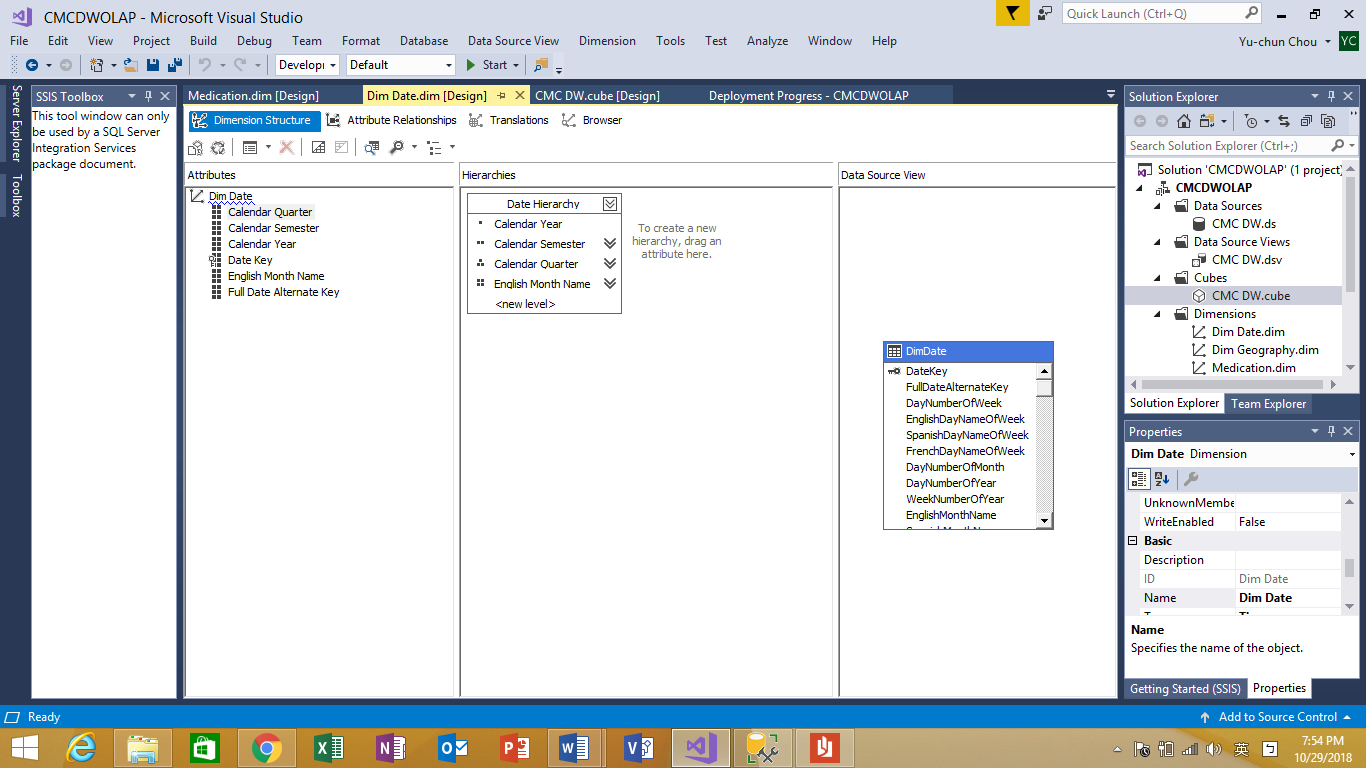
UPDAte fact set datekey= cast(FLOOR(RAND(CHECKSUM(NEWID()))\*(2007-2005+1)+2005) as char(4)) + RIGHT('00'+cast(FLOOR(RAND(CHECKSUM(NEWID()))\*(12-1+1)+1) as varchar(2)),2) + RIGHT('00'+cast(FLOOR(RAND(CHECKSUM(NEWID()))\*(28-1+1)+1) as varchar(2)),2)

The Lookup Transform Editor enables the simple joins between the input and the reference set.



1. **Creating Date dimension Hierarchy**

CalendarYear -> CalendarSemester -> CalenderQuarter -> EnglishMonthName



**OLAP Tools**

* Rollup and Drill down

1. Using SQL to accomplish Rollup (rollup to calendar year)

-- rollup

SELECT d.CalendarYear, SUM(f.unitCost) AS cost

FROM fact f,DimDate d

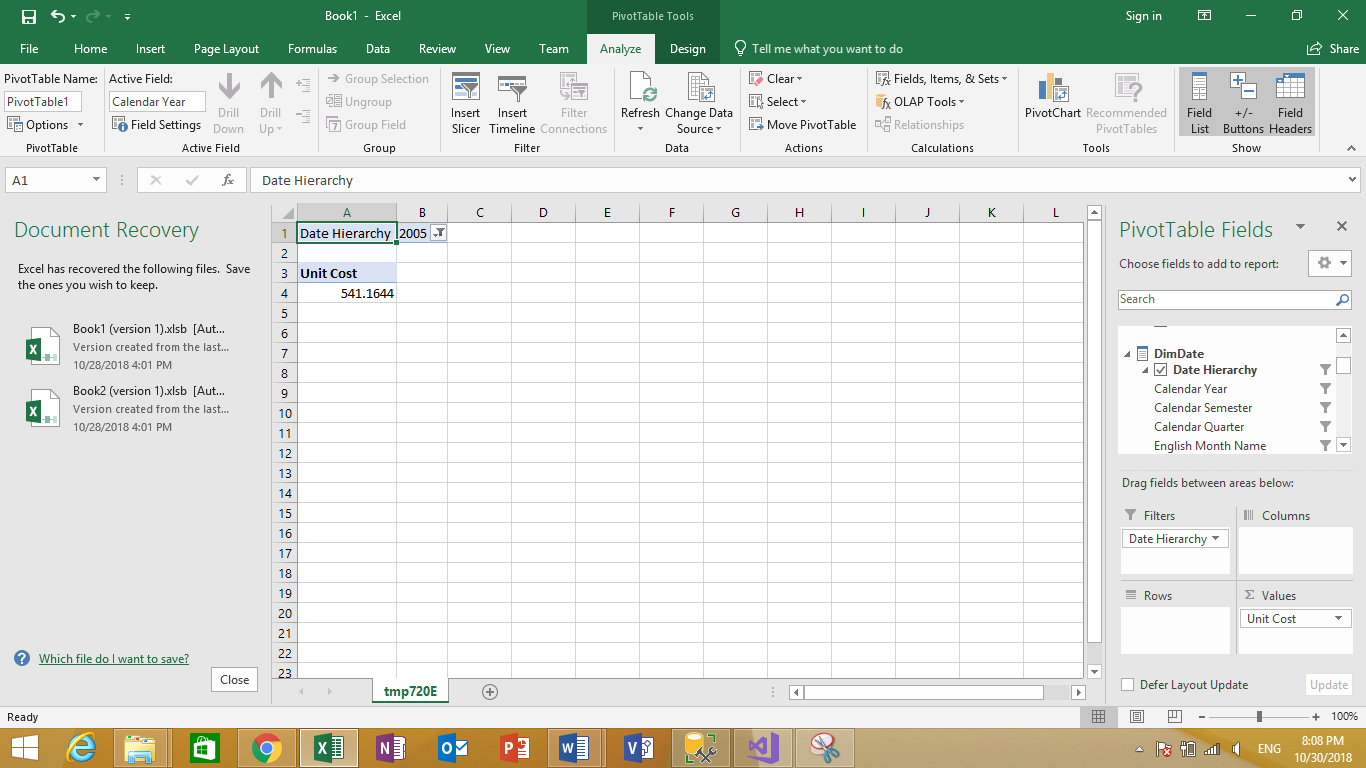
where f.dateKey = d.DateKey

GROUP BY d.CalendarYear WITH ROLLUP

Result:

|  |  |
| --- | --- |
| CalendarYear | Cost |
| 2005 | 280.50 |
| 2006 | 268.50 |
| 2007 | 371.00 |
| NULL | 920.00 |

Roll up to year by excel pivot table (for example : roll up to year 2005)



1. Using SQL to accomplish drilldown(drill down to calendar quarter)

|  |  |
| --- | --- |
| CalendarYear | Cost |
| 2005 | 280.50 |
| 2006 | 268.50 |
| 2007 | 371.00 |
| NULL | 920.00 |

drill down to month by excel pivot table (for example : January 2015)

-- query before drill down

SELECT d.CalendarYear, SUM(f.unitCost) AS cost

FROM fact f,DimDate d

where f.dateKey = d.DateKey

GROUP BY GROUPING SETS(([CalendarYear]), ())

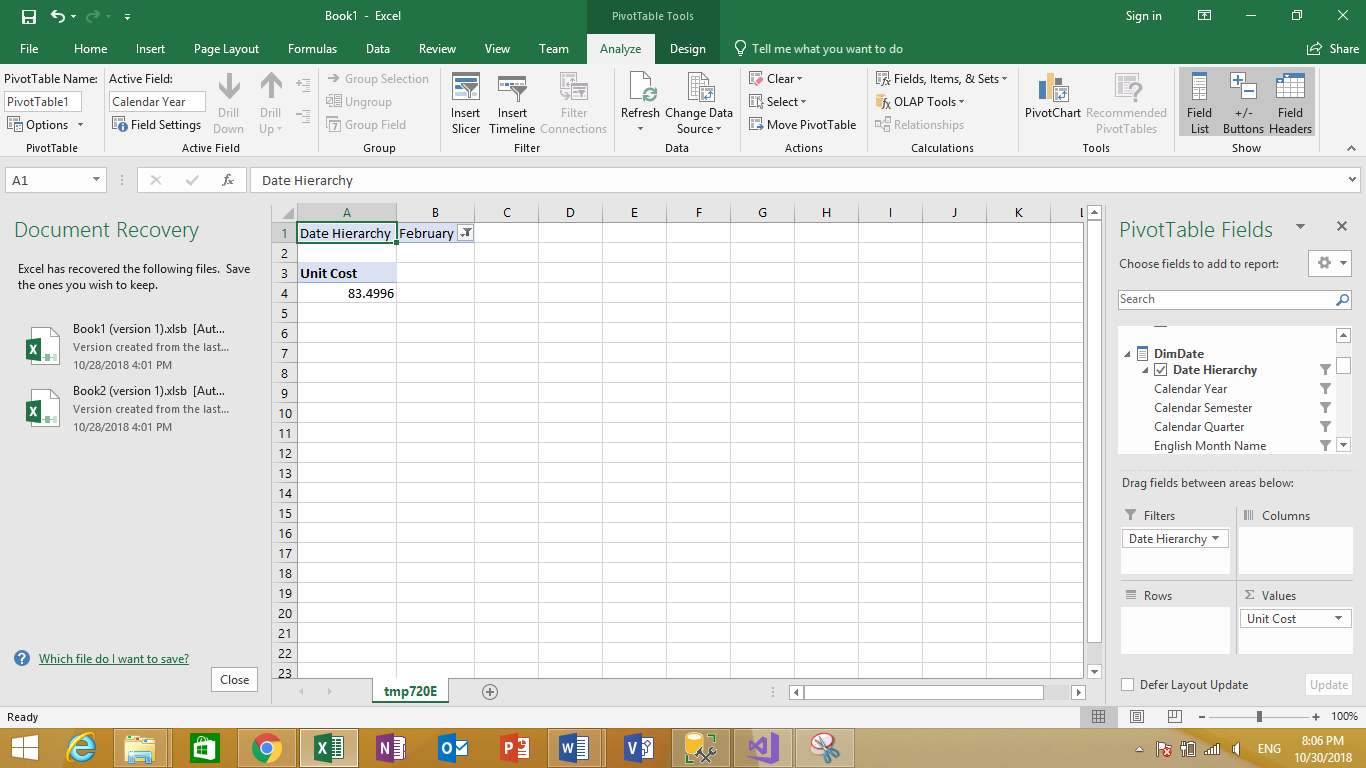
-- query after drill down

SELECT d.CalendarYear,d.CalendarQuarter, SUM(f.unitCost) AS cost

FROM fact f,DimDate d

where f.dateKey = d.DateKey

GROUP BY GROUPING SETS(([CalendarYear],[CalendarQuarter]), ([CalendarYear]))



Result:

|  |  |  |
| --- | --- | --- |
| CalendarYear | CalendarQuarter | cost |
| 2005 | 1 | 94.50 |
| 2005 | 2 | 47.50 |
| 2005 | 3 | 60.00 |
| 2005 | 4 | 78.50 |
| 2005 | NULL | 280.50 |
| 2006 | 1 | 48.00 |
| 2006 | 2 | 103.50 |
| 2006 | 3 | 78.50 |
| 2006 | 4 | 38.50 |
| 2006 | NULL | 268.50 |
| 2007 | 1 | 126.50 |
| 2007 | 2 | 83.50 |
| 2007 | 3 | 72.00 |
| 2007 | 4 | 89.00 |
| 2007 | NULL | 371.00 |

## KPI

1. Add ‘count of column’ to cube
2. Follow IT-532-01 to create KPI
3. In the **Name** box, change the name of this new calculated measure to **[average patient costs]**
4. In the **Expression** box, create the following MDX expression:

([Measures].[average patient costs]

1. Set KPI goal

Case

When [Patients].[PatientType].CurrentMember Is

[Patients].[PatientType].[O]

Then 100

When [Patients].[PatientType].CurrentMember Is

[Patients].[PatientType].[R]

Then 200

Else 100

End

1. Set KPI status

Case

When KpiValue( "Hospital average medication cost" ) /

KpiGoal ( "Hospital average medication cost" ) >= .90

Then 1

When KpiValue( "Hospital average medication cost" ) /

KpiGoal ( "Hospital average medication cost" ) < .90

And

KpiValue( "Hospital average medication cost" ) /

KpiGoal ( "Hospital average medication cost" ) >= .80

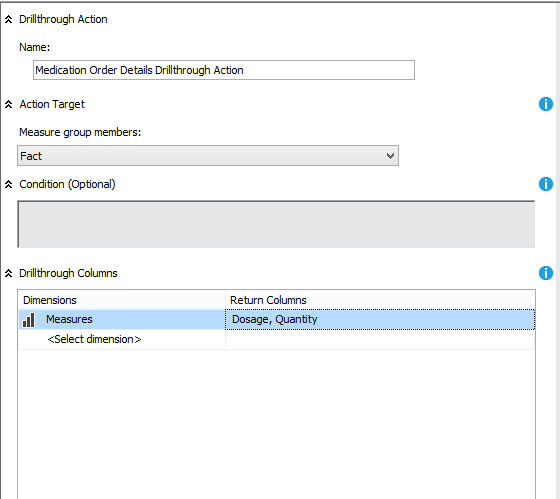
Then 0

Else -1

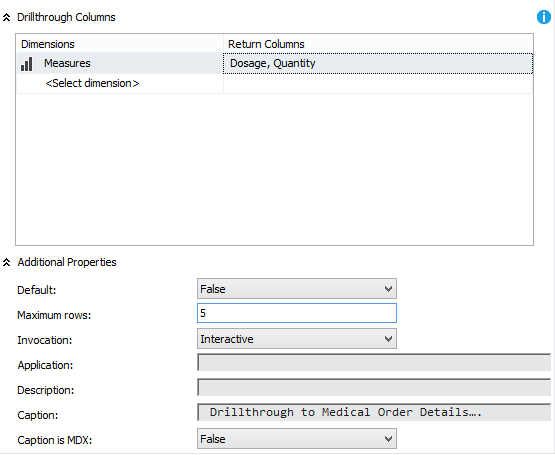
End

* + Drill through

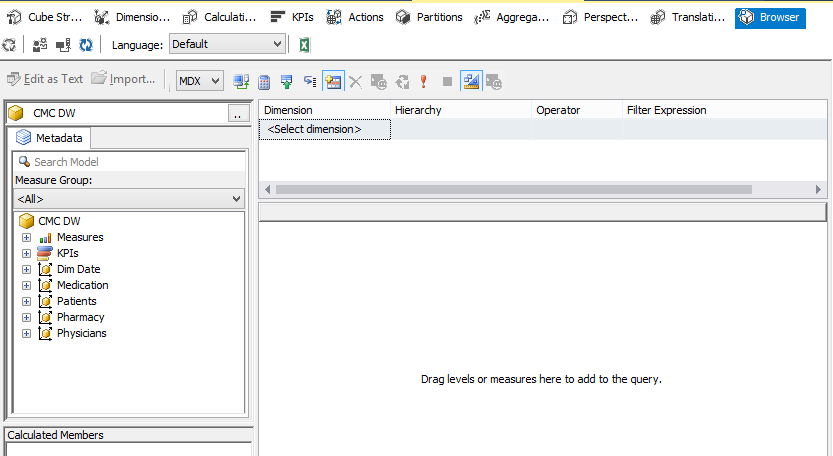
1. The setting is as below



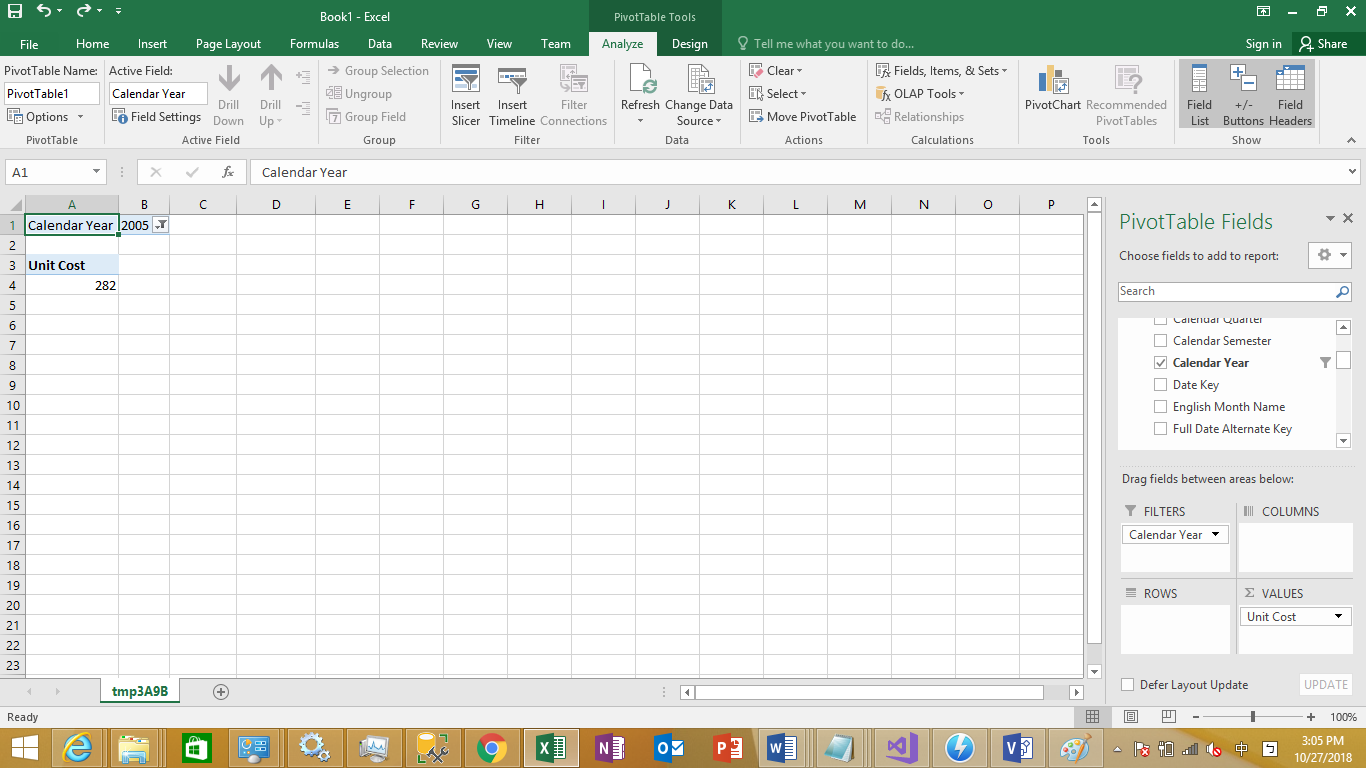
1. Click the ‘additional properties’



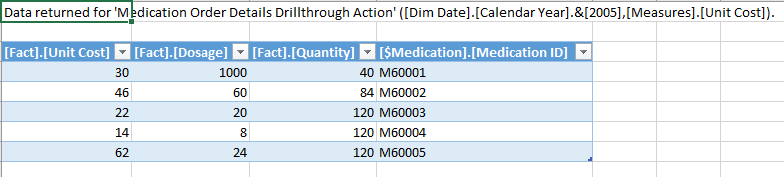
1. In browser, open excel



1. In excel, select calendar as filter and choose 2005, and select unit cost in values



1. In unit cost result, right click, point to **Additional Actions**, and then click **Drillthrough to Order Details**.
2. New drill through result will show, now we limit to 5



**Report:**

* Pivot table

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Column Labels** |  |  |  |  |  |  |  |  |  |  |  |
|  | **Dosage** |  |  | **Quantity** |  |  | **Unit Cost** |  |  | **Total Dosage** | **Total Quantity** | **Total Unit Cost** |
| **Row Labels** | **2005** | **2006** | **2007** | **2005** | **2006** | **2007** | **2005** | **2006** | **2007** |  |  |  |
| Acidity | 10 | 80 | 110 | 16 | 128 | 176 | 2.5 | 20 | 27.5 | 200 | 320 | 50 |
| Allergy medication | 10 | 18 | 12 | 150 | 270 | 180 | 17.5 | 31.5 | 21 | 40 | 600 | 70 |
| Diabetic medicines | 30 | 36 | 54 | 150 | 180 | 270 | 77.5 | 93 | 139.5 | 120 | 600 | 310 |
| Sudafed | 1000 | 1500 | 2500 | 40 | 60 | 100 | 30 | 45 | 75 | 5000 | 200 | 150 |
| Theraflu | 135 | 60 | 105 | 189 | 84 | 147 | 103.5 | 46 | 80.5 | 300 | 420 | 230 |
| Thyroid medication | 45 | 30 | 25 | 270 | 180 | 150 | 49.5 | 33 | 27.5 | 100 | 600 | 110 |
| Unknown | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Grand Total** | **1230** | **1724** | **2806** | **815** | **902** | **1023** | **280.5** | **268.5** | **371** | **5760** | **2740** | **920** |

Total Cost of medication sold by Year:

|  |  |
| --- | --- |
| 2005 | 280.50 |
| 2006 | 268.50 |
| 2007 | 371.00 |

Total Cost of medication sold by Quarter:

|  |  |  |
| --- | --- | --- |
| CalendarYear | CalendarQuarter | Cost |
| 2005 | 1 | 94.50 |
| 2005 | 2 | 47.50 |
| 2005 | 3 | 60.00 |
| 2005 | 4 | 78.50 |
| 2005 | NULL | 280.50 |
| 2006 | 1 | 48.00 |
| 2006 | 2 | 103.50 |
| 2006 | 3 | 78.50 |
| 2006 | 4 | 38.50 |
| 2006 | NULL | 268.50 |
| 2007 | 1 | 126.50 |
| 2007 | 2 | 83.50 |
| 2007 | 3 | 72.00 |
| 2007 | 4 | 89.00 |
| 2007 | NULL | 371.00 |

Total Quantity of Allergy and Flu Medication sold by year:

Allergy TheraFlu

Year Quantity(mg) Quantity(mg)

2005 150 189

2006 270 84

2007 180 147