Florida Hospital Centra Care Provider Performance Data Mart

Marbin Noel Díaz Díaz
Master in Computer Science
Nelliud Torres Batista, DBA
Electrical & Computer Engineering and Computer Science Department
Polytechnic University of Puerto Rico

Abstract *— Querying a transactional database for aggregates reports can result in poorly responding queries and reports. Developing a data mart is a highly effective solution to overcome these limitations. This paper demonstrates the steps needed to create, load and read a data mart. The provider performance data mart will provide key metrics for measuring how medical staff are abiding by the policies and guidelines of the corporation. The goal of this article is to provide the reader an introduction to the world of business intelligence and analytics.*

Key Terms *—Business Intelligence, Data Mart, Data Modeling, ETL.*

INTRODUCTION

Currently some organizations are wasting valuable time and resources creating ad-hoc reports directly from their production transactional systems. This practice not only makes it harder to produce those reports; it is also very time consuming to develop the business rules. In many cases, the performance of these reports is very poor. Another impact of this practice is the performance of the transactional system. It could be affected by the use of complex queries that make the database system use most of its available resources. For this kind of situations there are multiple techniques to get around over some of this problems.

One solution could be to create a copy of the production transactional database that would be used only for reporting purposes. Another solution is creating a data mart with most of the business logic already included in the data store. This project will focus on how the creation of a data mart will greatly reduce the time consuming processes of gathering and calculating different type of metrics. This particular data mart will be used to evaluate the performance of medical staff in one of the largest, non-profit urgent care practice in the state of Florida. These metrics can help to identify possible incorrect and or ineffective procedures that are being performed by the medical staff. It can also help to measure if their performance meets the organization expected goals.

THINGS TO CONSIDER WHEN DESIGNING A DATA MART

According to Turban, E. [1] "Whereas a data warehouse combines databases across an entire enterprise, a data mart is usually smaller and focuses on a particular subject or department. A data mart is a subset of a data warehouse, typically consisting of a single subject area". In another book by Ponniah, P. [2] "… provide analytical and reporting capabilities for specific business subjects based on the dimensional data model". Both authors agreed that a data mart targets a particular subject area. There are multiple types of data mart architectures:

- Independent data marts are a small warehouse where its source isn't an enterprise data warehouse.
- Dependent data marts are a subset that is created directly from the data warehouse (EDW).
- Data mart bus is an alternative way for the independent data marts to be linked together to maintain data consistency and sharing of dimensions.

Since the scope of this project is to facilitate the reporting capabilities of a business unit and provide the foundations for the addition and integration of other departments, the bus data mart architecture is the best alternative to fulfill the needs of the end user, the urgent care medical director. It also permits the organization to create a full featured data warehouse by combining multiple data marts that follow this bus architecture.

After deciding on the architecture, the next step is to begin with the data modeling process. The first step is to define the conceptual model. Camilovic [3] specified the following; “In relational environment the conceptual data model is usually presented by an ER diagram. However, ER models are not very suitable for data modeling in data warehousing”. Another modeling technique recommended is known as the ‘Dot Model’. Dot modeling has been accepted by people that lack technical skills.

After finishing the definition of the conceptual model, the logical data model should be designed. Logical modeling transforms the conceptual data model into a dimensional model, like a star schema, that consists of a fact table and one or multiple dimensional table(s). Dimension tables can be defined into three commonly used types known as slowly changing dimensions (SCD). These techniques are:

- SCD Type 1 is where the data is overwritten without preserving history.
- SCD Type 2 is used when a preservation of history is needed. When an attribute value changes, an entirely new record is created with all the unaffected attributes unchanged. The only exception is that affected value is changed to the new one.
- In SCD Type 3, the history is preserved in the same row by adding new columns that will reflect the old value along side the new values.

The last step is known as the physical data model in where the realization in the database schema of the logical data model is specified. According to Ponniah [3], there are several steps in the physical design process for a data mart or data warehouse. These are:

- Develop standards
- Create an aggregate plan
- Determine the data partitioning schema
- Establish clustering option
- Prepare an indexing strategy
- Create an storage plan
- Complete the physical model

After the tables are physically created in the database a process must be configured to load the data mart. This process is known as Extraction transformation and loading (ETL).

- Extraction of data from one or multiple sources.
- Transformation is the step that converts the extracted data from its previous form into the form that the data mart requires.
- Loading is the phase that the data is inserted into the database.

ETL is extremely important for data integration, the purpose is to load the data mart with clean and integrated data.

Another very important phase, testing, is where end users will make sure that the information on the data mart is precise. Among the methodologies used for testing, one of them is comparing the aggregated data in the data mart against the source system to find if there is any disparity in the information.

URGENT CARE PRACTICE DATA MART NEED

The corporate world is a very competitive one. If you don’t have the correct information at the right time, the corporation can lose money. This is where business intelligence comes into play. It unites a set of professionals with a specific skill set that will help build systems with the ability to look in the past, present and future by using different techniques like data warehousing, data mining, reporting, dashboards, text mining, among others. The problem that is going to be solved in this project is one that is very common in mid-size businesses with limited budget and a lot of information that isn’t analyzed and/or mined. This urgent care practice called Florida Hospital Centra Care has many offices across Central Florida. They are the first place that people think about when they get sick and their primary doctor isn’t available right away. They don’t have the budget for a business intelligence department. Their reporting capabilities are very limited and the reporting repository is an exact copy of the main application database (OLTP) that isn't optimized for reporting related queries. Among all departments within this corporation one of them has a particular need. This department is in charge of all the medical staff that works in the Urgent Care offices. They have always had the need to create a monthly scorecard in where shows different metrics on how doctors are performing. Those metrics show them how many patients per hour a doctor is seeing, whether prescriptions are on paper or digital, if the doctor is following the correct procedures for certain conditions, survey scores, average time from the walk in to the clinic to the walk in to the patient room, etc. It has to respond faster than the actual reports that point to the raw data source. Everything is recorded on a daily basis. To accomplish this request a data mart is designed that includes a full-featured ETL process and a final data-driven report with a monthly schedule that will send the scorecard into each mailbox of each doctor. This data mart not only serves the initial request needs but also is designed with the idea of becoming part of a big unified data warehouse when more data marts are later added and integrated.

CREATING THE DATA MART

Following the guidelines for data mart creation and the requirements of the medical director the next drawing shows the dot diagram was generated showing a general view of the data mart design.

Figure 1 shows that the data mart is composed of three main objects.

- FactProviderDailyPerformace (Fact table with the metrics).
- DimLocation (Office description).
- DimProvider (Doctor description).
- DimDate (Date description with a hierarchy relationship).

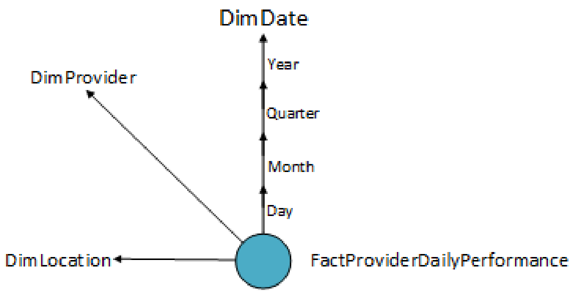
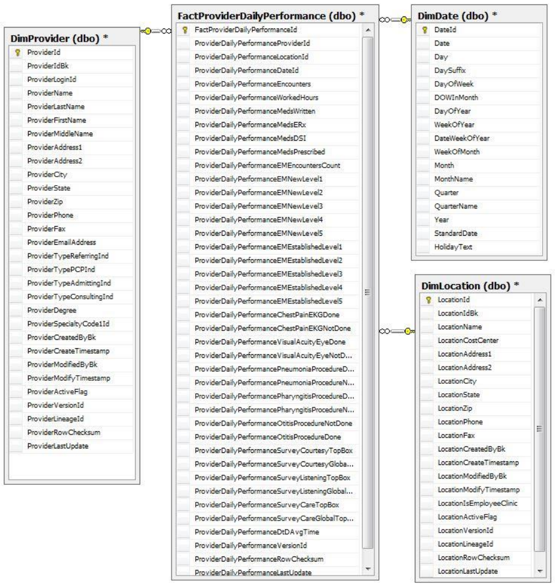


Figure 1
Conceptual Model Dot Diagram

This diagram facilitates the definition of the attributes of those tables by displaying a macro view of the data mart. For the Logical Data Model each of the objects is defined in more detail with the name of each of the attributes among other things.



DimProvider (dbo) *
ProviderId
ProviderIdBk
ProviderLoginId
ProviderName
ProviderLastName
ProviderFirstName
ProviderMiddleName
ProviderAddress1
ProviderAddress2
ProviderCity
ProviderState
ProviderZip
ProviderPhone
ProviderFax
ProviderEmailAddress
ProviderTypeReferringInd
ProviderTypePCPInd
ProviderTypeAdmittingInd
ProviderTypeConsultingInd
ProviderDegree
ProviderSpecialtyCode1Id
ProviderCreatedByBk
ProviderCreateTimestamp
ProviderModifiedByBk
ProviderModifyTimestamp
ProviderActiveFlag
ProviderVersionId
ProviderLineageId
ProviderRowChecksum
ProviderLastUpdate

FactProviderDailyPerformance (dbo) *
FactProviderDailyPerformanceId
ProviderDailyPerformanceProviderId
ProviderDailyPerformanceLocationId
ProviderDailyPerformanceDateId
ProviderDailyPerformanceEncounters
ProviderDailyPerformanceWorkedHours
ProviderDailyPerformanceMedsWritten
ProviderDailyPerformanceMedsERx
ProviderDailyPerformanceMedsDSI
ProviderDailyPerformanceMedsPrescribed
ProviderDailyPerformanceEMEncountersCount
ProviderDailyPerformanceEMNewLevel1
ProviderDailyPerformanceEMNewLevel2
ProviderDailyPerformanceEMNewLevel3
ProviderDailyPerformanceEMNewLevel4
ProviderDailyPerformanceEMNewLevel5
ProviderDailyPerformanceEMEstablishedLevel1
ProviderDailyPerformanceEMEstablishedLevel2
ProviderDailyPerformanceEMEstablishedLevel3
ProviderDailyPerformanceEMEstablishedLevel4
ProviderDailyPerformanceEMEstablishedLevel5
ProviderDailyPerformanceChestPainEKGDone
ProviderDailyPerformanceChestPainEKGNotDone
ProviderDailyPerformanceVisualAcuityEyeDone
ProviderDailyPerformanceVisualAcuityEyeNotD...
ProviderDailyPerformancePneumoniaProcedureD...
ProviderDailyPerformancePneumoniaProcedureN...
ProviderDailyPerformancePharyngitisProcedureD...
ProviderDailyPerformancePharyngitisProcedureN...
ProviderDailyPerformanceOtitisProcedureNotDone
ProviderDailyPerformanceOtitisProcedureDone
ProviderDailyPerformanceSurveyCourtesyTopBox
ProviderDailyPerformanceSurveyCourtesyGloba...
ProviderDailyPerformanceSurveyListeningTopBox
ProviderDailyPerformanceSurveyListeningGlobal...
ProviderDailyPerformanceSurveyCareTopBox
ProviderDailyPerformanceSurveyCareGlobalTop...
ProviderDailyPerformanceDtDAvgTime
ProviderDailyPerformanceVersionId
ProviderDailyPerformanceRowChecksum
ProviderDailyPerformanceLastUpdate

DimDate (dbo) *
DateId
Date
Day
DaySuffix
DayOfWeek
DOWInMonth
DayOfYear
WeekOfYear
DateWeekOfYear
WeekOfMonth
Month
MonthName
Quarter
QuarterName
Year
StandardDate
HolidayText

DimLocation (dbo) *
LocationId
LocationIdBk
LocationName
LocationCostCenter
LocationAddress1
LocationAddress2
LocationCity
LocationState
LocationZip
LocationPhone
LocationFax
LocationCreatedByBk
LocationCreateTimestamp
LocationModifiedByBk
LocationModifyTimestamp
LocationIsEmployeeClinic
LocationActiveFlag
LocationVersionId
LocationLineageId
LocationRowChecksum
LocationLastUpdate

Figure 2
Logical Data Model Macro View

The next step is to define the Physical Data Model that will contain the data type for each of the attributes.

This macro view shown in Figure 3 reveals how the tables in the data mart are arranged.

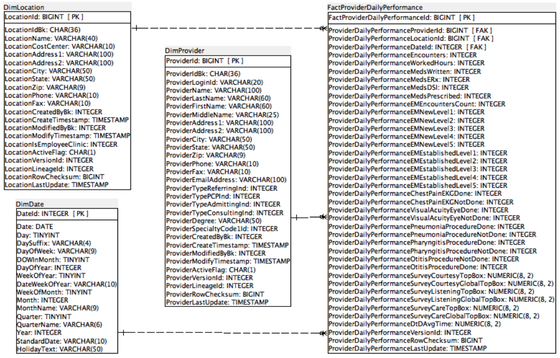
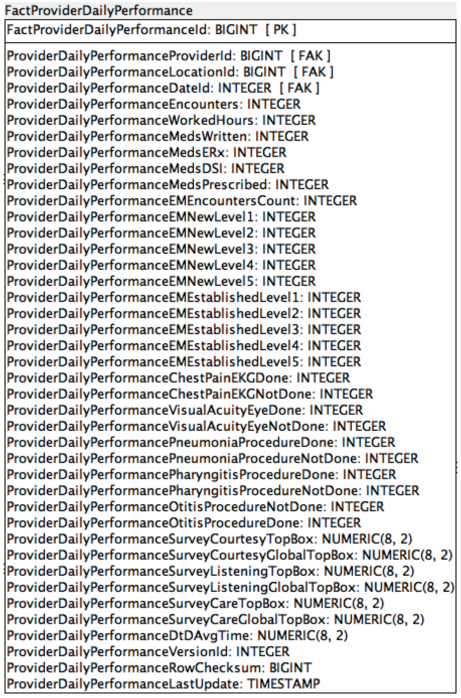


Figure 3

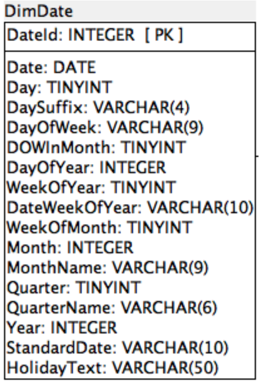
Physical Data Model Macro View – DimLocation (top left), DimDate (bottom left), DimProvider (center), FactProviderDailyPerformance (right side).



FactProviderDailyPerformance
FactProviderDailyPerformanceId: BIGINT [PK]
ProviderDailyPerformanceProviderId: BIGINT [FAK]
ProviderDailyPerformanceLocationId: BIGINT [FAK]
ProviderDailyPerformanceDateId: INTEGER [FAK]
ProviderDailyPerformanceEncounters: INTEGER
ProviderDailyPerformanceWorkedHours: INTEGER
ProviderDailyPerformanceMedsWritten: INTEGER
ProviderDailyPerformanceMedsERx: INTEGER
ProviderDailyPerformanceMedsDSI: INTEGER
ProviderDailyPerformanceMedsPrescribed: INTEGER
ProviderDailyPerformanceEMEncountersCount: INTEGER
ProviderDailyPerformanceEMNewLevel1: INTEGER
ProviderDailyPerformanceEMNewLevel2: INTEGER
ProviderDailyPerformanceEMNewLevel3: INTEGER
ProviderDailyPerformanceEMNewLevel4: INTEGER
ProviderDailyPerformanceEMNewLevel5: INTEGER
ProviderDailyPerformanceEMEstablishedLevel1: INTEGER
ProviderDailyPerformanceEMEstablishedLevel2: INTEGER
ProviderDailyPerformanceEMEstablishedLevel3: INTEGER
ProviderDailyPerformanceEMEstablishedLevel4: INTEGER
ProviderDailyPerformanceEMEstablishedLevel5: INTEGER
ProviderDailyPerformanceChestPainEKGDone: INTEGER
ProviderDailyPerformanceChestPainEKGNotDone: INTEGER
ProviderDailyPerformanceVisualAcuityEyeDone: INTEGER
ProviderDailyPerformanceVisualAcuityEyeNotDone: INTEGER
ProviderDailyPerformancePneumoniaProcedureDone: INTEGER
ProviderDailyPerformancePneumoniaProcedureNotDone: INTEGER
ProviderDailyPerformancePharyngitisProcedureDone: INTEGER
ProviderDailyPerformancePharyngitisProcedureNotDone: INTEGER
ProviderDailyPerformanceOtitisProcedureNotDone: INTEGER
ProviderDailyPerformanceOtitisProcedureDone: INTEGER
ProviderDailyPerformanceSurveyCourtesyTopBox: NUMERIC(8, 2)
ProviderDailyPerformanceSurveyCourtesyGlobalTopBox: NUMERIC(8, 2)
ProviderDailyPerformanceSurveyListeningTopBox: NUMERIC(8, 2)
ProviderDailyPerformanceSurveyListeningGlobalTopBox: NUMERIC(8, 2)
ProviderDailyPerformanceSurveyCareTopBox: NUMERIC(8, 2)
ProviderDailyPerformanceSurveyCareGlobalTopBox: NUMERIC(8, 2)
ProviderDailyPerformanceDtDAvgTime: NUMERIC(8, 2)
ProviderDailyPerformanceVersionId: INTEGER
ProviderDailyPerformanceRowChecksum: BIGINT
ProviderDailyPerformanceLastUpdate: TIMESTAMP

Figure 4

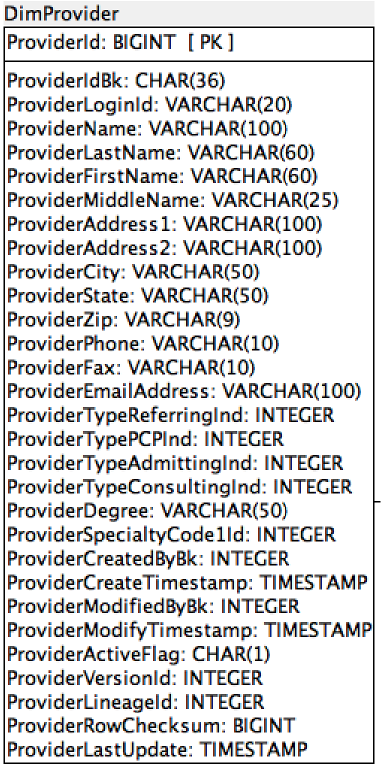
Physical Data Model FactProviderDailyPerformance



DimDate
DateId: INTEGER [PK]
Date: DATE
Day: TINYINT
DaySuffix: VARCHAR(4)
DayOfWeek: VARCHAR(9)
DOWInMonth: TINYINT
DayOfYear: INTEGER
WeekOfYear: TINYINT
DateWeekOfYear: VARCHAR(10)
WeekOfMonth: TINYINT
Month: INTEGER
MonthName: VARCHAR(9)
Quarter: TINYINT
QuarterName: VARCHAR(6)
Year: INTEGER
StandardDate: VARCHAR(10)
HolidayText: VARCHAR(50)

Figure 5

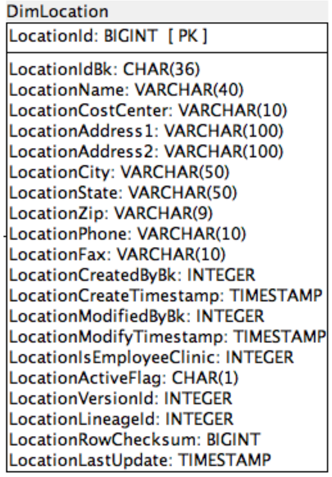
Physical Data Model DimDate



DimProvider
ProviderId: BIGINT [PK]
ProviderIdBk: CHAR(36)
ProviderLoginId: VARCHAR(20)
ProviderName: VARCHAR(100)
ProviderLastName: VARCHAR(60)
ProviderFirstName: VARCHAR(60)
ProviderMiddleName: VARCHAR(25)
ProviderAddress1: VARCHAR(100)
ProviderAddress2: VARCHAR(100)
ProviderCity: VARCHAR(50)
ProviderState: VARCHAR(50)
ProviderZip: VARCHAR(9)
ProviderPhone: VARCHAR(10)
ProviderFax: VARCHAR(10)
ProviderEmailAddress: VARCHAR(100)
ProviderTypeReferringInd: INTEGER
ProviderTypePCPInd: INTEGER
ProviderTypeAdmittingInd: INTEGER
ProviderTypeConsultingInd: INTEGER
ProviderDegree: VARCHAR(50)
ProviderSpecialtyCode1Id: INTEGER
ProviderCreatedByBk: INTEGER
ProviderCreateTimestamp: TIMESTAMP
ProviderModifiedByBk: INTEGER
ProviderModifyTimestamp: TIMESTAMP
ProviderActiveFlag: CHAR(1)
ProviderVersionId: INTEGER
ProviderLineageId: INTEGER
ProviderRowChecksum: BIGINT
ProviderLastUpdate: TIMESTAMP

Figure 6

Physical Data Model DimProvider



DimLocation
LocationId: BIGINT [PK]
LocationIdBk: CHAR(36)
LocationName: VARCHAR(40)
LocationCostCenter: VARCHAR(10)
LocationAddress1: VARCHAR(100)
LocationAddress2: VARCHAR(100)
LocationCity: VARCHAR(50)
LocationState: VARCHAR(50)
LocationZip: VARCHAR(9)
LocationPhone: VARCHAR(10)
LocationFax: VARCHAR(10)
LocationCreatedByBk: INTEGER
LocationCreateTimestamp: TIMESTAMP
LocationModifiedByBk: INTEGER
LocationModifyTimestamp: TIMESTAMP
LocationIsEmployeeClinic: INTEGER
LocationActiveFlag: CHAR(1)
LocationVersionId: INTEGER
LocationLineageId: INTEGER
LocationRowChecksum: BIGINT
LocationLastUpdate: TIMESTAMP

Figure 7

Physical Data Model DimLocation

EXTRACT, TRANSFORM AND LOAD

Microsoft Integration Services (ETL) was used to load the tables on a daily schedule that executes early in the morning. Data must be loaded in a specific order. Dimensions will be loaded first and then Facts. The reason for this is that facts need to have the surrogate key available in the dimension to use it as a reference number for the record that is going to be loaded.

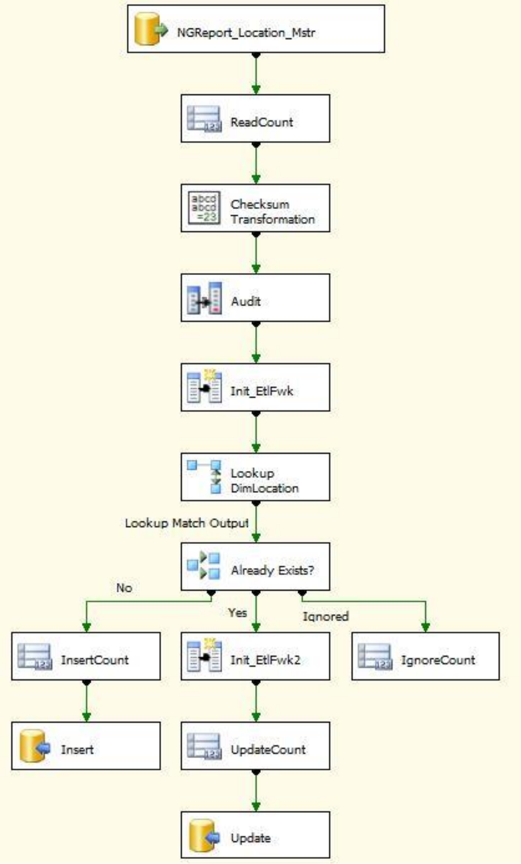


Figure 8
DimLocation ETL Process

The first step of this process as shown in Figure 8 is to read the location (offices) data from the source in this case is Nextgen EHR & PM system. It only reads data that has been modified or added with a timestamp newer than the previous execution. Minimizing the need for wasting resources in reading data that is already loaded in the target dimension table. For the process to know if the record was already loaded before and if has been modified a checksum value is calculated using all the values of the attributes. That checksum is later compared to the one saved in the dimension table to the same record that matches the incoming one. If the value is different, it means that the records changed and needs to be updated in the dimension table. On the other hand, if the value is the same, then no changes are applied to that record in the dimension table. In the case that no matching records is found in the target dimension table, a new one is inserted. But before loading it, the data is standardized to the data mart required format. That is the basic workflow inside an ETL job that loads a data warehouse or a data mart.

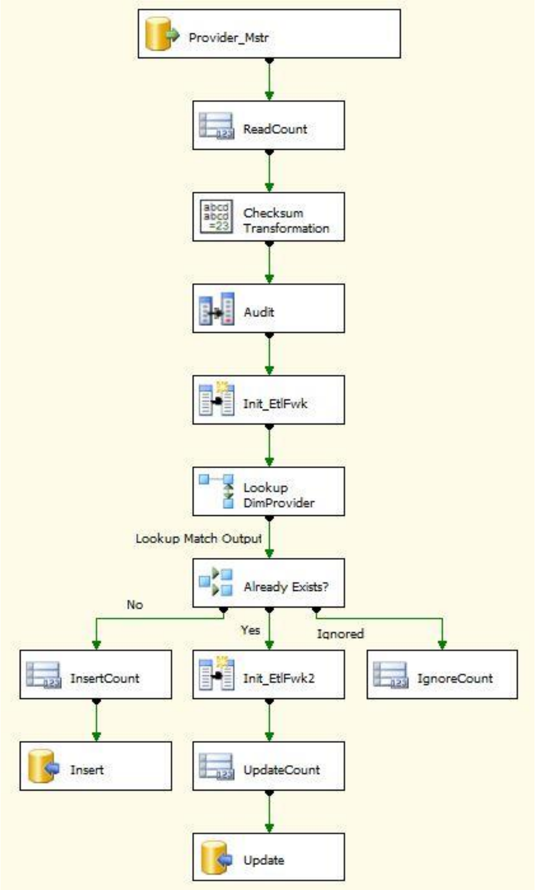


Figure 9
DimProvider ETL Process

Figure 9 follows the same basic steps of Figure 8. It reads the data, standardizes it and finally decides if is going to insert a new record or update an existing one.

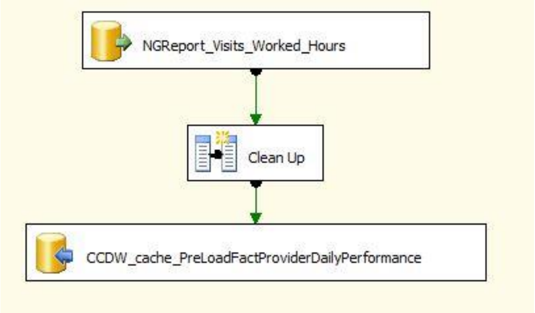

NGReport_Visits_Worked_Hours
Clean Up
CCDW_cache_PreLoadFactProviderDailyPerformance

Figure 10
FactProviderDailyPerformance – Pre Load

On some occasions, fact tables can become enormous data sets with millions of records. For this reason special care has to be taken to handle them and keep the process executing in an acceptable timeframe. For this reason, it is a good idea to set up a pre load process in where it will initially load only the primary and foreign keys from the target and used dimensions to reduce the footprint of memory needed when running the primary ETL process that will load the fact table. By doing this once the primary process is executed, the preload table will be used to lookup only the needed records and not the whole universe of data, reducing the waste of resources and improving the speed of the whole ETL job.

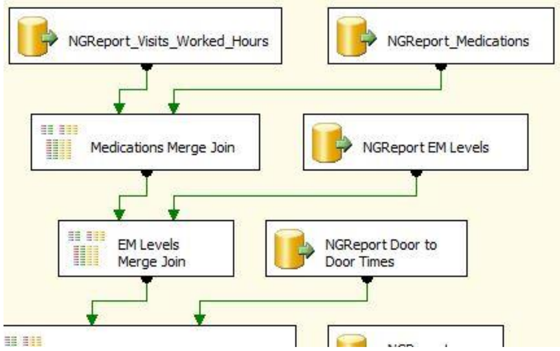


Figure 11
Reading steps for the FactProviderDailyPerformance table.

In this step of the ETL job that loads the FactProviderDailyPerformance table. Aggregated data is joined together. The end user defined those metrics.

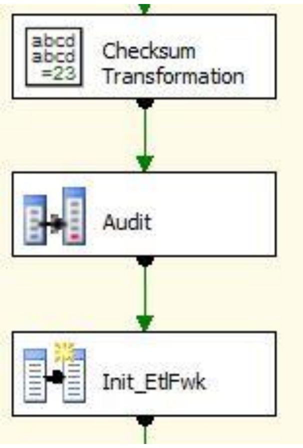


Figure 12
Checksum Calculation – FactProviderDailyPerformance ETL Process

Figure 14 shows that the same checksum techniques used on the dimension loading processes are used too in the process of loading the fact table. This technique is critical to help determine if the record will be flagged as an update or not.

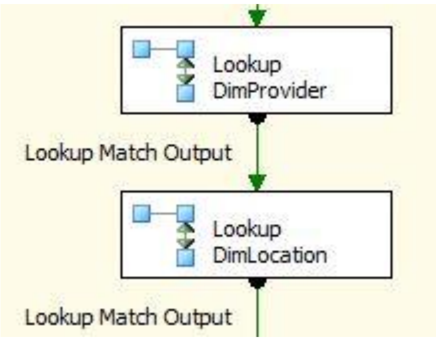


Figure 13
Dimension Lookup Steps – FactProviderDailyPerformance ETL Process

This is the reason for processing first the dimension tables and last the fact tables. The facts will need to look up the dimensions to get the surrogate key that will be used to reference those values when the tables are joined together. Fact tables only contain surrogate keys and metrics. It should not contain description about the data. Dimensions are the ones responsible for handling the description related to that record.

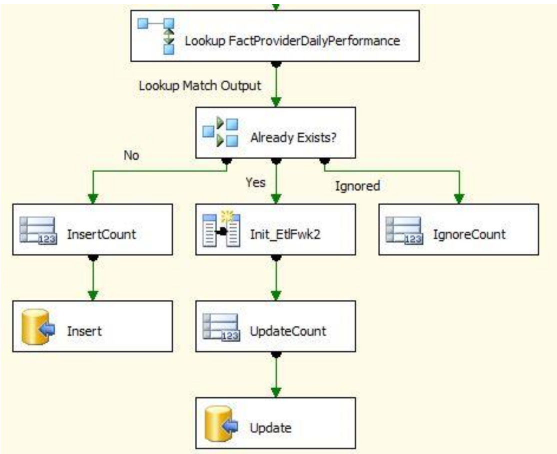
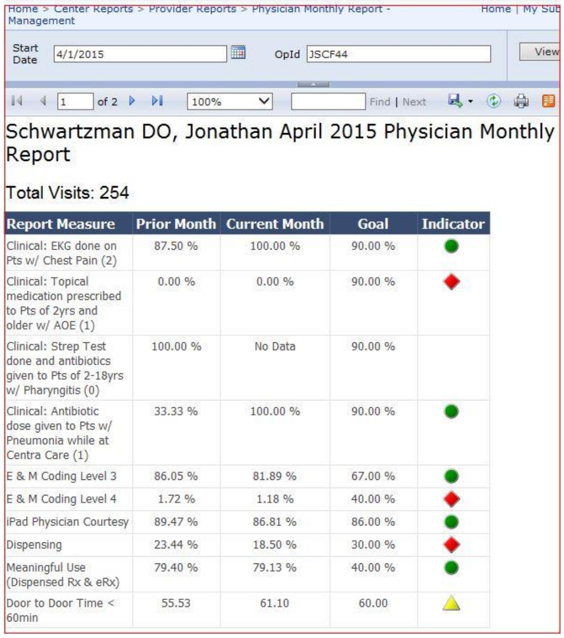


Figure 14
Loading Steps – FactProviderDaily Performance

The lookup step resource as shown in Figure 14 footprint and execution time can be greatly reduced by joining the pre-load table and the target fact table. This will only get those existing records that will needed to check if they exist in the target table. There is no need to load the whole fact table into that lookup object, especially in those cases that the table is millions of rows millions of rows.

END USER REPORT

The following images show an example of one of the many ways a data mart can be used. In this case a report was built to display the information in a meaningful way to end users.


Home > Center Reports > Provider Reports > Physician Monthly Report - Management
Home | My Sub
View
1 of 2
100%
Find | Next

Start Date 4/1/2015

OpId JSCF44

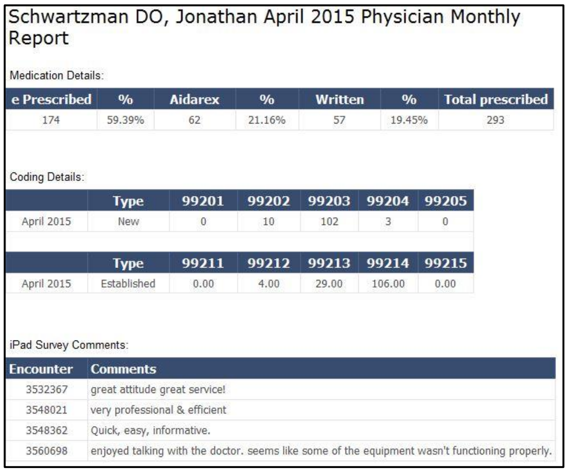
Schwartzman DO, Jonathan April 2015 Physician Monthly Report

Total Visits: 254

Report Measure	Prior Month	Current Month	Goal	Indicator
Clinical: EKG done on Pts w/ Chest Pain (2)	87.50 %	100.00 %	90.00 %	●
Clinical: Topical medication prescribed to Pts of 2yrs and older w/ AOE (1)	0.00 %	0.00 %	90.00 %	◆
Clinical: Strep Test done and antibiotics given to Pts of 2-18yrs w/ Pharyngitis (0)	100.00 %	No Data	90.00 %	
Clinical: Antibiotic dose given to Pts w/ Pneumonia while at Centra Care (1)	33.33 %	100.00 %	90.00 %	●
E & M Coding Level 3	86.05 %	81.89 %	67.00 %	●
E & M Coding Level 4	1.72 %	1.18 %	40.00 %	◆
iPad Physician Courtesy	89.47 %	86.81 %	86.00 %	●
Dispensing	23.44 %	18.50 %	30.00 %	◆
Meaningful Use (Dispensed Rx & eRx)	79.40 %	79.13 %	40.00 %	●
Door to Door Time < 60min	55.53	61.10	60.00	▲

Figure 15

First Page of Report that Uses the Data Mart to Display the Monthly Metrics of Doctors



Schwartzman DO, Jonathan April 2015 Physician Monthly Report

Medication Details:

e Prescribed	%	Aidarex	%	Written	%	Total prescribed
174	59.39%	62	21.16%	57	19.45%	293

Coding Details:

	Type	99201	99202	99203	99204	99205
April 2015	New	0	10	102	3	0

	Type	99211	99212	99213	99214	99215
April 2015	Established	0.00	4.00	29.00	106.00	0.00

iPad Survey Comments:

Encounter	Comments
3532367	great attitude great service!
3548021	very professional & efficient
3548362	Quick, easy, informative.
3560698	enjoyed talking with the doctor. seems like some of the equipment wasn't functioning properly.

Figure 16

Second Page of Report that Uses the Data Mart to Display the Monthly Metrics of Doctors

There are many ways to view the data from a data mart. People can query the database directly, Microsoft Excel, MS Access or a report in reporting service. Figure 15 and 16 show a report made using Microsoft Reporting Services that allows the medical director to access online and view the stats of each of his doctors. Figure 15 has a table with a comparison between previous month statistics and current month. A Key Performance Indicator (KPI) with a threshold defined by the end user indicates if the metric value is on the expected range for the current month. There are other reports that the medical director can use to compare two or more medics. Using the Power Pivot plugin of Excel he can also play with data and analyze it from different combination of values without the need of a programmer to create a static report for each of the views he would like to see. Using this functionality from Excel, the end user can organize the data in many possible ways.

CONCLUSION

This provider performance data mart has many business rules already defined on it. No longer do end user have to remember very complex rules. End users can easily extract the information from it and create different reports that can provide the necessary insight of how the providers are performing over time. Before the data mart existed, getting some of the metrics was a very complex process and very imprecise. The poor response of those reports lead to many complaints by the end users. It could take around 9 minutes to execute for a single doctor. Now with this new optimized content store known as the data mart, the time has been reduced to an almost instantaneous response. Since the metrics are currently pre-aggregated and the tables are properly indexed without affecting the transactional system performance, end users can access the same information quicker. They can go back in time to compare old metrics with the actual ones without having to wait a long time. Also the data mart allows them to analyze the data in multiple dimensions easier than before. A properly executed data mart can become the starting point for a data warehouse. This type of data store can potentially contain information not only about providers but the general operational data for reporting purposes, analysis, data mining, etc.

References

[1] T. Turban, et al., *Business Intelligence A Managerial Approach*, Vol. No. 2, 2011.

[2] P. Ponniah, *Data Warehousing Fundamentals for IT Professionals*, Vol. No. 2, 2010.

[3] D. Camilovic, *A Call Detail Records Data Mart: Data Modelling and OLAP Analysis. Computer Science & Information Systems*, Vol. No. 2, December 2009, pp. 87-110.