Data Warehouse in Access. Ridiculous? Not so! Part 2

Recap:

What is a data warehouse? Most people think of a data warehouse in terms of storage space and processing power.

But a data warehouse is really defined by function and topology.

Function

Schools of Thought:

Bill Inmon: “A data warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process.”

Ralph Kimball: “A data warehouse is a copy of transaction data specifically structured for query and analysis.”

Nothing in either definition mentions size or speed. There’s nothing that excludes Access.

Data warehouse architecture

These store the data (discussed last time):

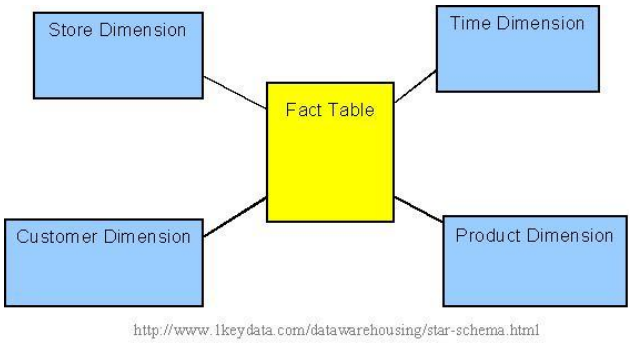
- **Data Source Layer**
- **Data Extraction Layer**
- **Staging Area**
- **ETL Layer**
- **Data Storage Layer**
- **Data Logic Layer**

These report the data:

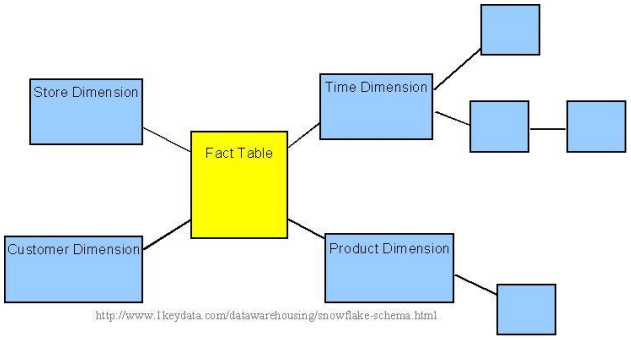
- **Data Presentation Layer**
- **Metadata Layer**
- **System Operations Layer**

There’s no particular reason that each of these layers can’t be built in Access.

Topology



Sample star schema



Sample snowflake schema

Data Warehousing Paradigms

Bill Inmon's paradigm: Data warehouse is one part of the overall business intelligence system. An enterprise has one data warehouse, and data marts source their information from the data warehouse. In the data warehouse, information is stored in 3rd normal form.

Ralph Kimball's paradigm: Data warehouse is the conglomerate of all data marts within the enterprise. Information is always stored in the dimensional model.

There is nothing inherent in any of the above definitions that precludes Access as an implementation platform. Granted most data warehouses will hold more data than Access is capable of, but that is a limit of scale not of kind. Inmon would call it a data mart rather than a data warehouse, so if you’re more comfortable thinking of what I’m doing here as a data mart, that’s fine with me.

The Problem

I work at the second largest hospital system in Michigan. Despite what you've heard, health care quality improvement is a major initiative. Our system has put a lot of time, effort, and money behind it. We've targeted 40 major conditions that represent 70% of all activity in the hospitals.

Conditions

A condition is a disease, procedures, or surgery.

Condition Examples: AMI, hip replacement, vaginal delivery, heart failure.

Each Condition has a team of doctors, nurses, and other specialists who meet periodically to drive improvement in that condition.

Indicators

But before you can improve something, you have to measure it – know where you are, where you want to go, if or when you ever get there. So each team decides on measurement indicators – things that can be measured – to show whether patient outcomes are improving.

Indicator Examples:

- Length of Stay: patient outcomes have been tied to appropriate length of stay. We want patients out of the hospital as soon as possible, but not too soon.
- Mortality: obviously we want to know if patients die in our care.
- Aspirin on Arrival: research has shown that heart attack patients that receive an aspirin within 30 minutes of arrival at the hospital have better outcomes.
- Age: age is an important factor when evaluating the other indicators.

Indicator Types

- Boolean values – measures things that either happened or they didn't. A patient died or he didn't. A patient received and aspirin or she didn't.
- Scalar values – measure things that have a range of values, like length of stay, age, costs

Reporting Examples:

Dashboard: Aspirin, Mortality, Length Of Stay, Beta Blocker for 2009

AMI

January 1, 2009 - December 31, 2009

	Aspirin On Arrival	Mortality	LOS	Beta Blocker
Hospital A	53%	16%	3.7	3%
Hospital B	92%	4%	2.7	3%

Chart: Aspirin by Physician for 2009

Physician	Aspirin
051000	0.744
140012	0.318
610000	0.583
102910	0.75
008061	0.491
040711	0.267
170610	0.648
713020	0.412
400038	0.424
120185	0.568
020110	0.727
010320	0.409

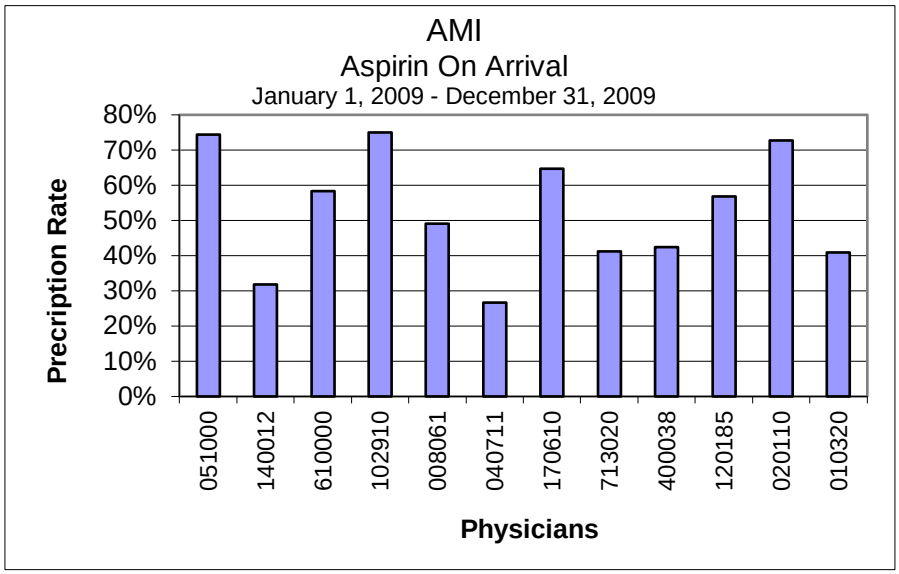
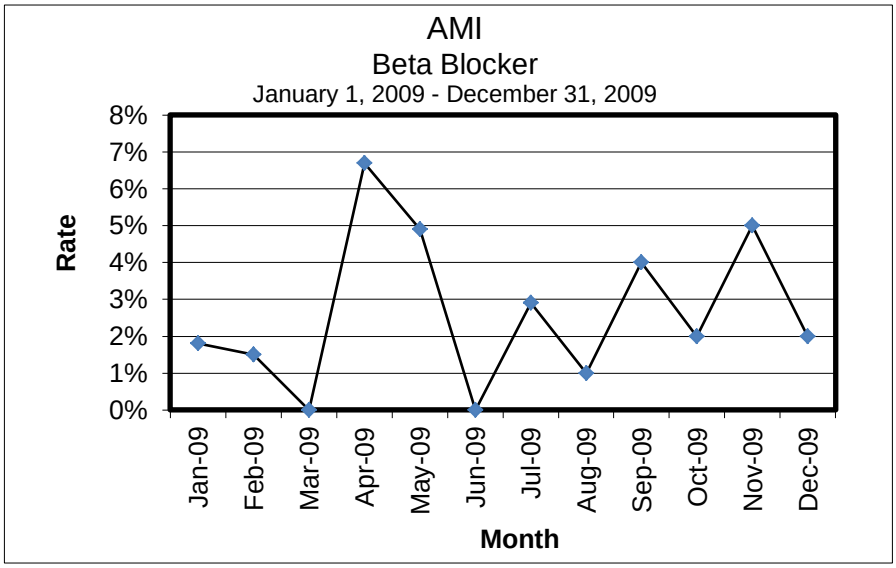


Chart: Beta Blocker by Month for 2009

Month	Rate
Jan-09	0.02
Feb-09	0.01
Mar-09	0.00
Apr-09	0.07
May-09	0.05
Jun-09	0.00
Jul-09	0.03
Aug-09	0.01
Sep-09	0.04
Oct-09	0.02
Nov-09	0.05
Dec-09	0.02



Implementing a Warehouse in Access

Last time I talked about creating the dimensional model for the warehouse, which comprises these levels of the data warehouse model:

- **Data Source Layer**
  This represents the different data sources that feed data into the data warehouse.
- **Data Extraction Layer**
  Data gets pulled from the data source into the data warehouse system.
- **Staging Area / ETL Layer**
  This is where data sits prior to being scrubbed and transformed. This is where data gains its "intelligence", as logic is applied to transform the data from a transactional nature to an analytical nature.
- **Data Storage Layer**
  This is where the transformed and cleansed data sit.
- **Data Logic Layer**
  This is where business rules are stored.

This time I want to talk about reporting the information I've stored in my warehouse

- **Data Presentation Layer**

This refers to the information that reaches the users.

I'm going to spend the majority of my talk on this layer, so I'll skip it for now

  - Excel 2010
  - DH_QBF
  - Generator.accdb

- **Metadata Layer**
  This is where information about the data stored in the data warehouse system is stored.

  In this case, the metadata is stored throughout the system. Each database stores information about its layer. The relationships window in DW_Stage is where the data model for the ETL Stage. The individual queries in DW_Stage hold the logic for calculating the fields in the Fact Table (Patient_Indicator).

- **System Operations Layer**
  This layer includes information on how the data warehouse system operates,

Again, this information is scattered throughout the various programs. While it is certainly possible to completely automate this process, in practice, I find it useful to run each stage manually, although each stage is as automated as possible. Mostly, this is because my actual system is fed from multiple, external data sources over which I have no control. Sometimes human judgment is the best solution to a problem.

Data Presentation Layer

Okay, let's return to the presentation layer. Obviously, this is the whole purpose of the data warehouse. It's the part that is often referred to as Business Intelligence or the Decision Support system.

A decision support system will usually two parts: Standard Reports and Ad-hoc Reports. Standard reports can be run by decision makers and they usually give a 30,000 foot view. These are often created by analysts and placed in a library for the decision maker to run. Ad-hoc reports are often run by analysts who understand the system better to answer questions that may not be on the standard reports. Useful ad-hoc reports often become standard reports.

Another key feature of decision support systems is the ability to drill down into progressively finer granularity. For instance, my report may show the Aspirin on Arrival rate per quarter. I may want drill down to the month level. I may even want to drill down to the actual patient identifiers to investigate why the failure occurred.

All of the major data warehouse programs have sophisticated decision support engines. In fact, MicroStrategy gives away its Reporting Suite for free.
http://www.microstrategy.com/freereportingsoftware/

Three Solutions:

My intent here is not to give you comprehensive solutions or a working utility. My intent is to show several approaches that you can use to create flexible and easy to use reporting services for your data warehouse users.

Duane Hookom's Query By Form

http://www.rogersaccesslibrary.com/forum/hookom-duane_forum4.html

Query By Form was written by Duane Hookom as a simple query creation form. I've modified it slightly in the way that it interacts with Excel. In the original, it just exports the query to a new spreadsheet named "MyQuery.xls". I modified it to name the file after the report and used the TransferSpreadsheet method instead of OutputTo.

For years, I've been using a trick of the TransferSpreadsheet method in Access to do some simple Office Automation. When you export a query with TransferSpreadsheet , it will create a sheet with the name of the query on the tab and a named range on the sheet with the same name. If you export the query again to the same Excel file, it will over-write the existing tab with data from the new export.

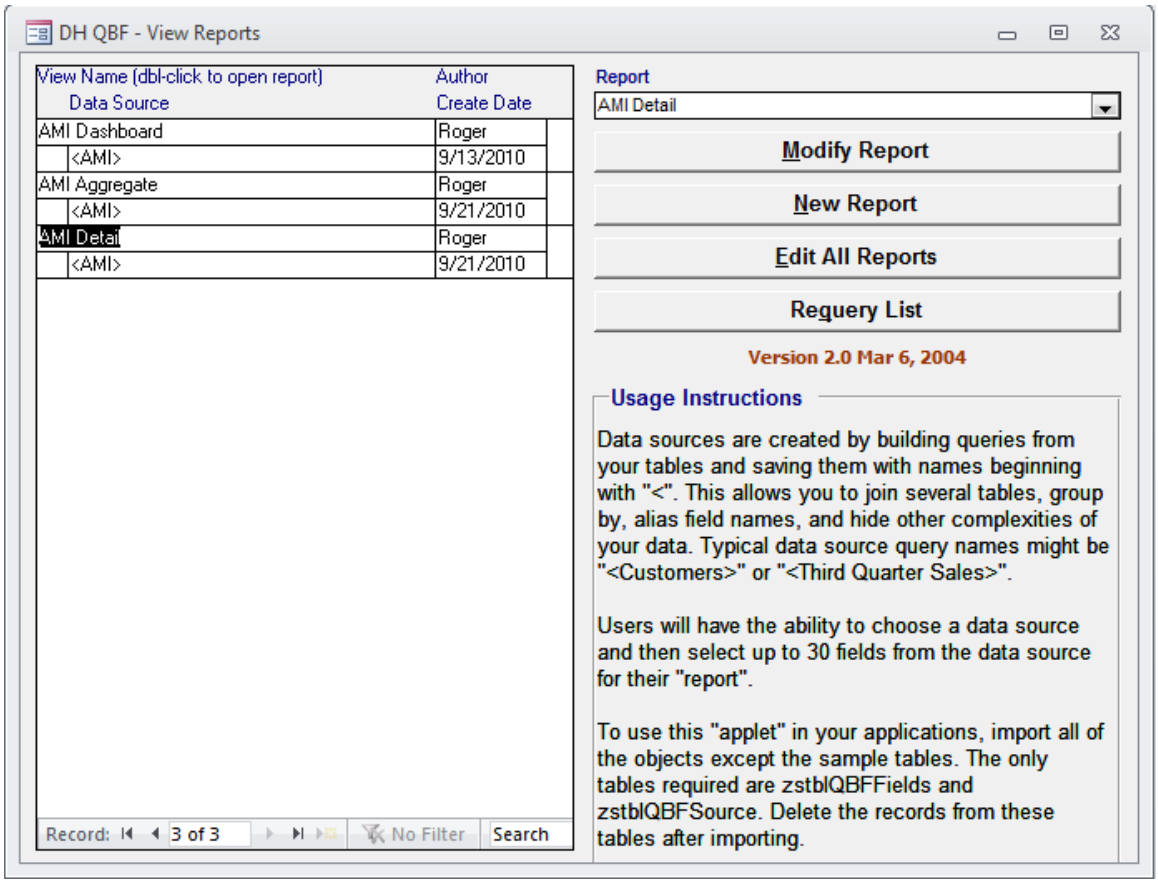
So what I do, is export the query once, then create a new sheet (Sheet1) and link the cells from the new sheet back to the query sheet (MyQuery). I can format the cells in Sheet1 any way I like, and the next time I export, the data on MyQuery sheet is replaced, and the linked cells on Sheet1 are updated.

Dashboards

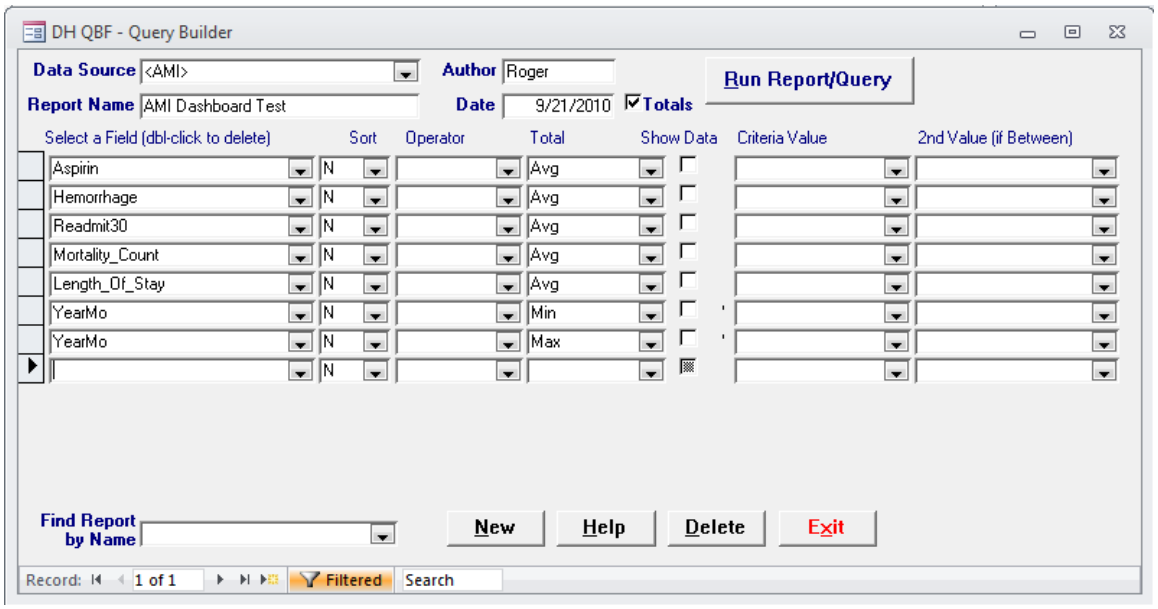
1. Create Query with brackets: <HF> (Select * from HF) In this case, am using the Cubes, but I could at this point, Join HF_Dim, Patient_Indicator and Time_Dim:

```
SELECT patient_id, Date_DIM.Year, Date_DIM.Month, Date_DIM.YearMo,
Date_DIM.MoYear, Beta_Blocker, DVT, DVT_Prophylaxis, Length_Of_Stay,
Mortality_Count, Readmit30
FROM (Date_DIM INNER JOIN Patient_Indicator ON Date_DIM.Date_ID =
Patient_Indicator.Date_ID) INNER JOIN HF_DIM ON patient_id = HF_DIM.patient_id;
```

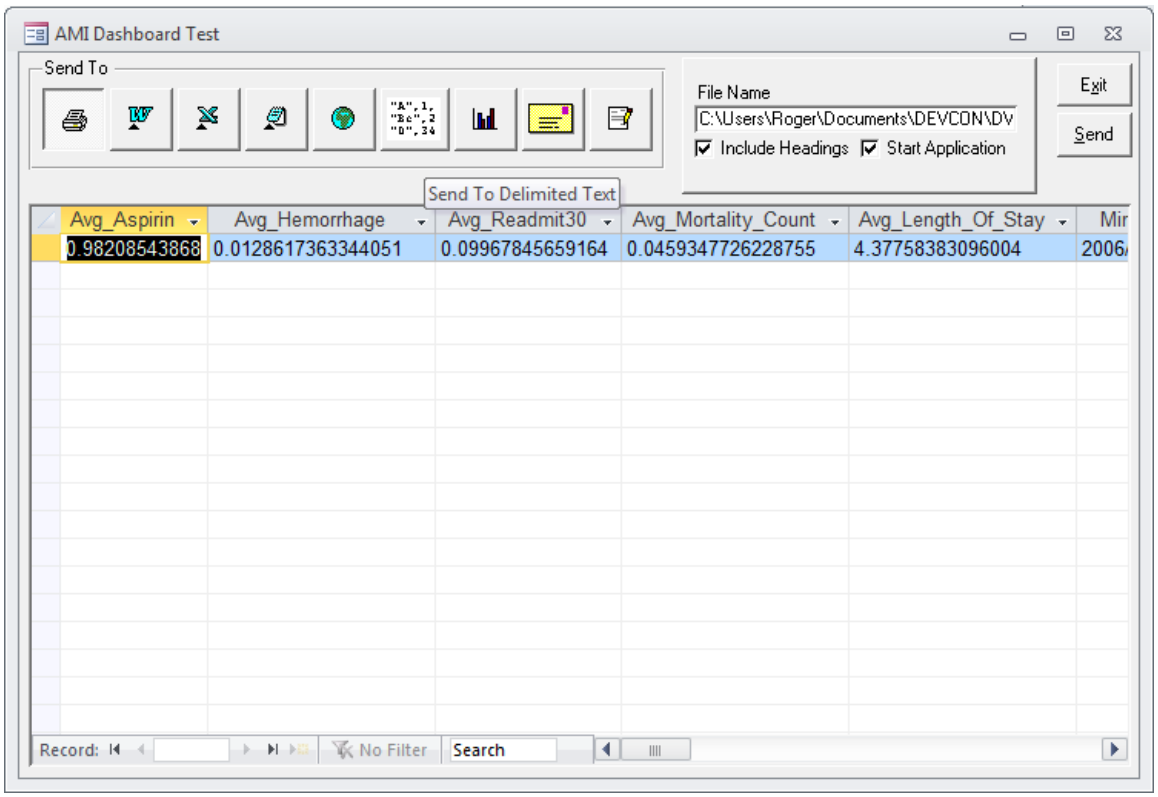
2. From the View Reports form, choose New Report:



3. Choose a data source, give it a name, choose fields, set the Totals, and choose AVG for all fields



4. Run the query:



5. Demonstrate the Send to buttons and show the files created. End with Excel.

6. In Excel, copy the data in the Myquery tab and PasteLink it into a new sheet. Format the table, format the cells (percent and number), add a title and timeframe.

	A	B	C	D	E
1					
2	**AMI Dashboard**				
3	**2006/07 - 2008/06**				
4	Avg_Aspirin	Avg_Hemorrhage	Avg_Readmit30	Avg_Mortality_Count	Avg_Length_Of_Stay
5	98.21%	1.29%	9.97%	4.59%	4.38
6					

7. Delete the data and re-run the report. The numbers appear again.

8. Change the criteria of the query (<#1/1/2008#). Re-run and compare the numbers.

This is all well and good but what about charts?

Charts

9. Create a new report (AMI Aggregate) to show rates grouped at monthly intervals. From this we can create a run chart showing the trends for each indicator.
10. Run the report. And export it to Excel. There is a single sheet called “myquery”.

	A	B	C	D	E	F	G
1	YearMo	Avg_Aspir	Avg_Hem	Avg_Read	Avg_Mort	Avg_Length_Of_Stay	
2	2006/07	1	0.010417	0.166667	0.020833	5.166667	
3	2006/08	1	0	0.163043	0.076087	3.663043	
4	2006/09	0.974026	0.025974	0.077922	0.025974	4.064935	
5	2006/10	1	0.009709	0.07767	0.009709	3.669903	
6	2006/11	0.978261	0	0.184783	0.032609	4.956522	
7	2006/12	0.990196	0.029412	0.068627	0.068627	3.705882	
8	2007/01	0.962025	0.050633	0.126582	0.063291	4.468354	
9	2007/02	1	0	0.125	0.027778	4.541667	
10	2007/03	0.958333	0.020833	0.114583	0.0625	3.927083	
11	2007/04	0.989247	0.010753	0.11828	0.086022	5.075269	
12	2007/05	0.988636	0.011364	0.090909	0.045455	4.727273	
13	2007/06	0.945455	0	0.109091	0.045455	4.436364	
14	2007/07	0.976744	0.011628	0.104651	0.069767	4.395349	
15	2007/08	0.965517	0.011494	0.034483	0.057471	3.793103	
16	2007/09	1	0.016393	0.065574	0.032787	4.344262	
17	2007/10	0.978723	0.010638	0.095745	0.053191	4.340426	
18	2007/11	0.986667	0.013333	0.04	0	4.906667	
19	2007/12	0.991453	0.025641	0.08547	0.051282	4.273504	
20	2008/01	0.989011	0.021978	0.131868	0.043956	5.78022	
21	2008/02	0.987805	0	0.04878	0.060976	4.182927	
22	2008/03	0.98913	0.01087	0.163043	0.01087	5	
23	2008/04	0.980198	0	0.09901	0.039604	3.70297	

myquery

11. Add a sheet and name it **Aspirin**
12. Select YearMo and Aspirin columns, copy and paste link into Aspirin sheet
13. Select data and create chart. Format as required.
14. Save as Template and re-run.
15. Show AMI Aggregate Complete template. Run and demonstrate completed charts.

Now this is monthly data, but suppose it consisted of a 30-day rolling data. If an executive had a link to this spreadsheet on his or her desk, he or she could have the lastest daily data every morning. All you'd need is an automated process to create the data and output it to a network folder.

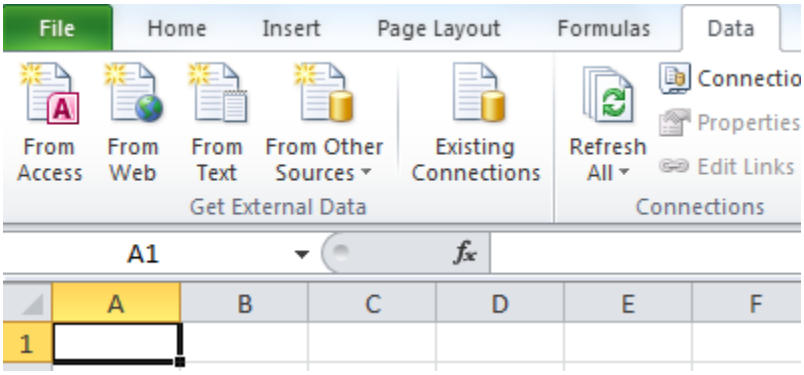
Excel 2010: Pivot Tables / Pivot Charts

Excel 2010 has some exciting Business Intelligence features that can be used to report from your data warehouse.

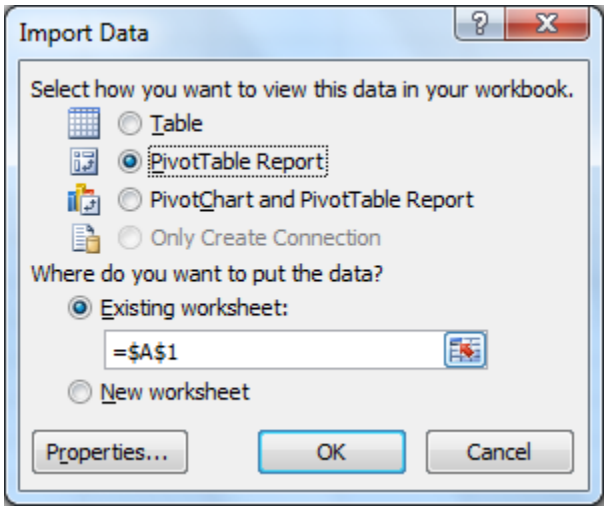
Pulling the Data

Creating Pivot Tables

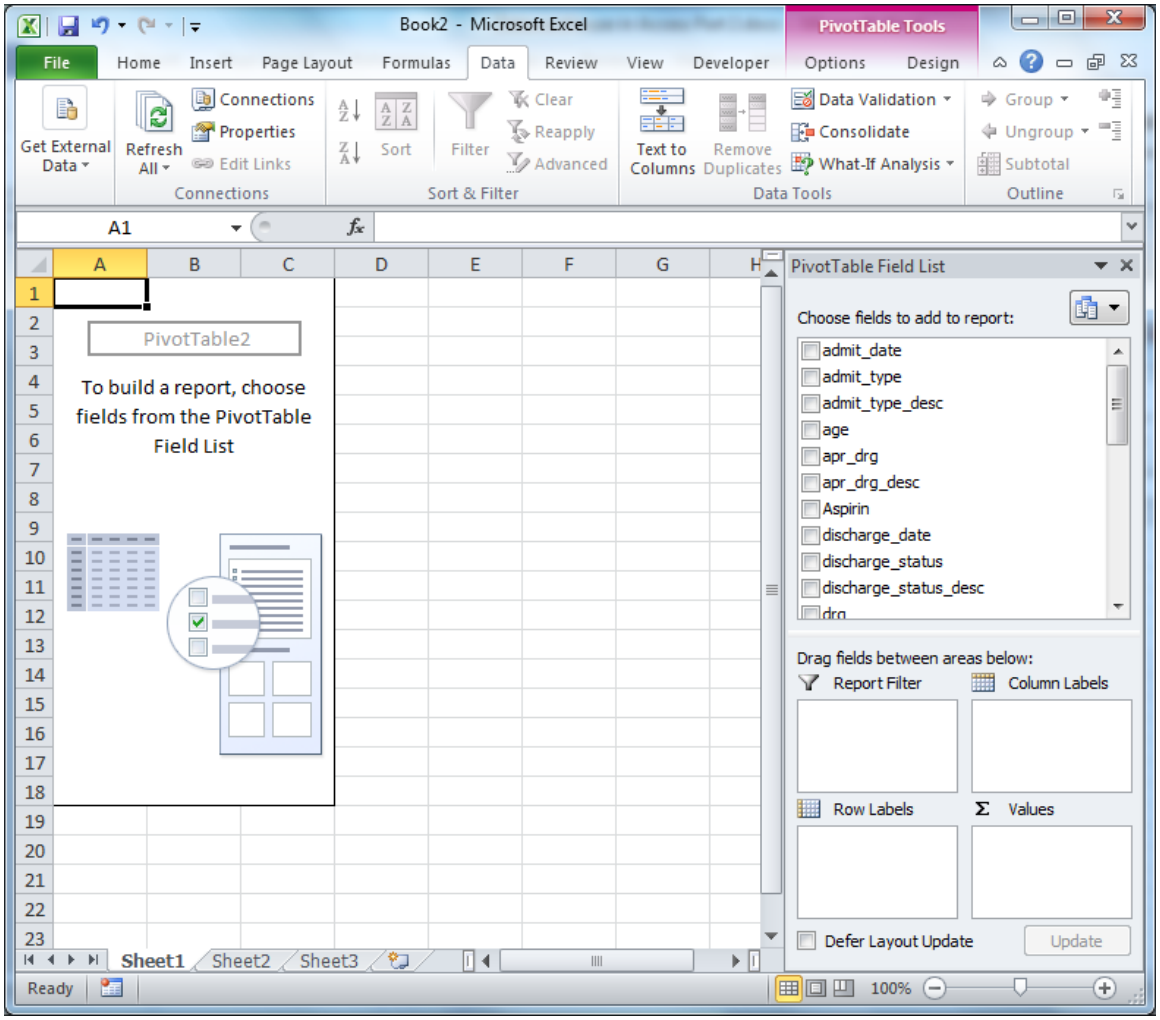
1. Create new workbook
2. Get External data from Access (data tab)



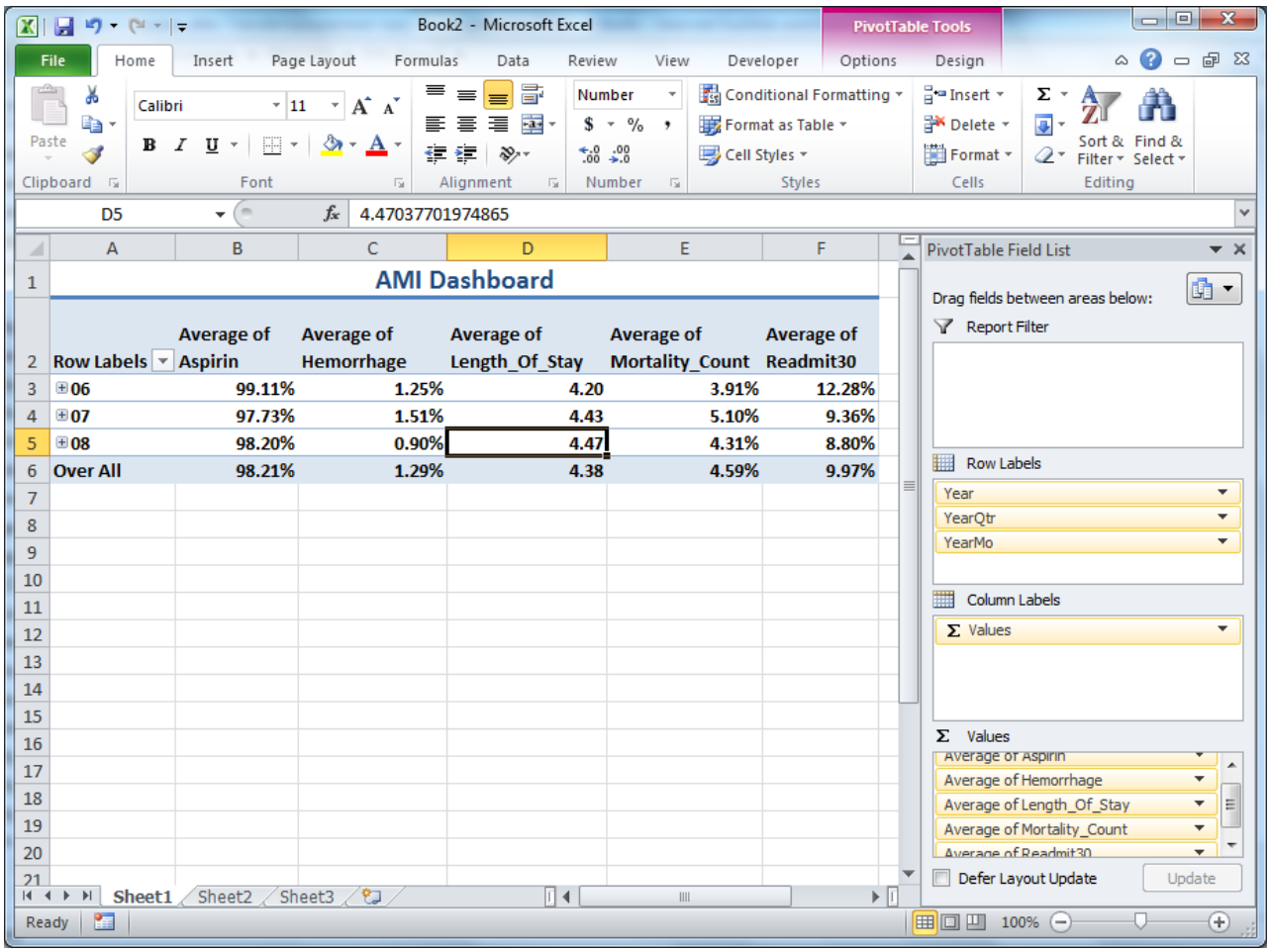
3. Browse to **C:\Users\Roger\Documents\DEVCON\DW_Report\pivotdata\DW_BE.accdb**
4. Select **AMI** and choose **PivotTable Report**



5. The screen looks like this:



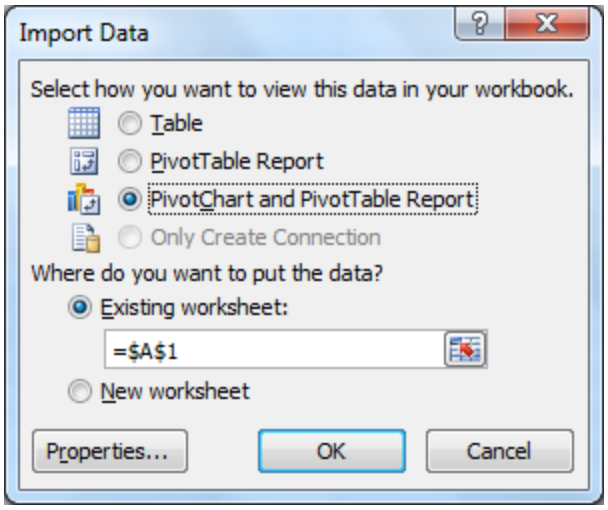
6. Add fields and format like this:

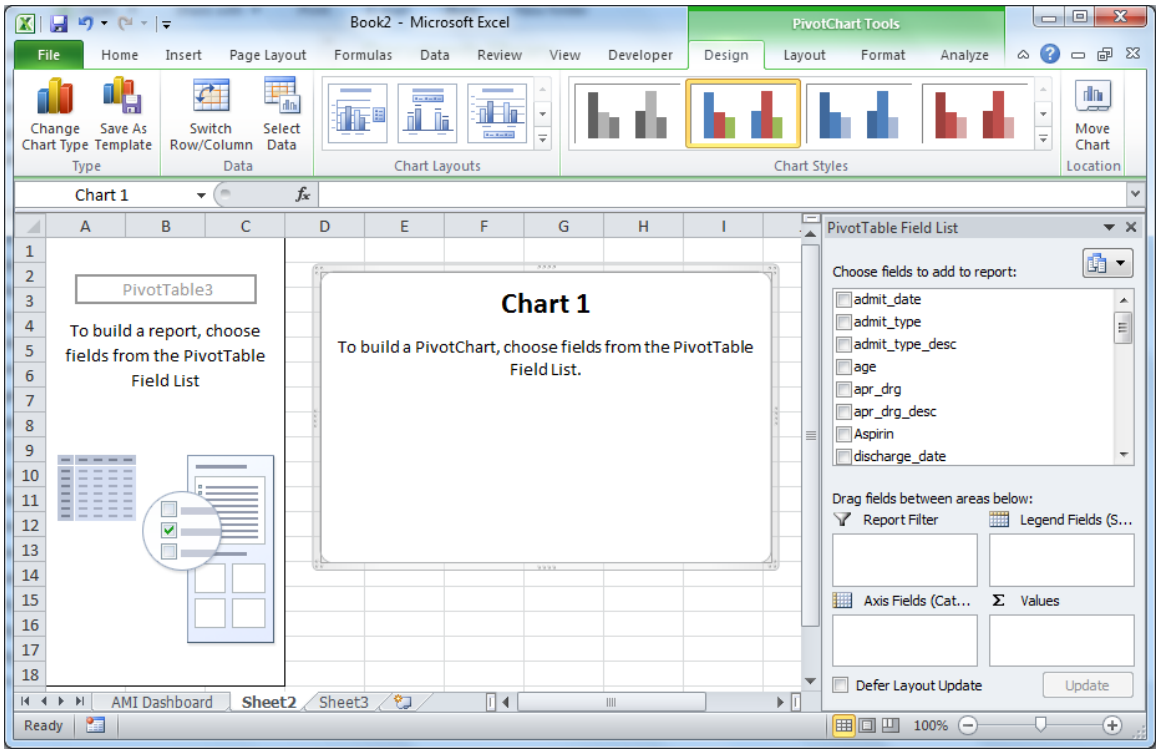


7. Expand dates to show quarters and months. Click on a row to show detail for that time frame.

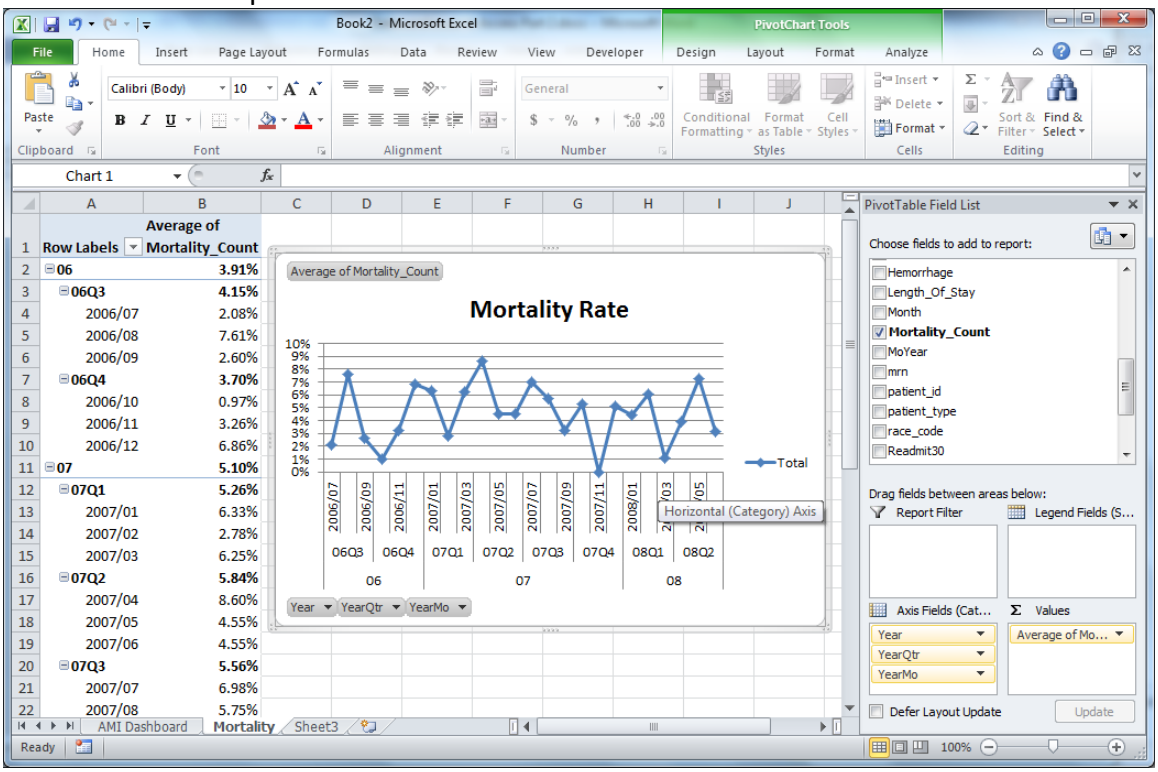
8. Name sheet Dashboard

9. To new sheet, get external Data from Access. Choose **Pivot Chart and Pivot Table Report**


Import Data
Select how you want to view this data in your workbook.
Table
PivotTable Report
PivotChart and PivotTable Report
Only Create Connection
Where do you want to put the data?
Existing worksheet:
=A1
New worksheet
Properties...
OK
Cancel

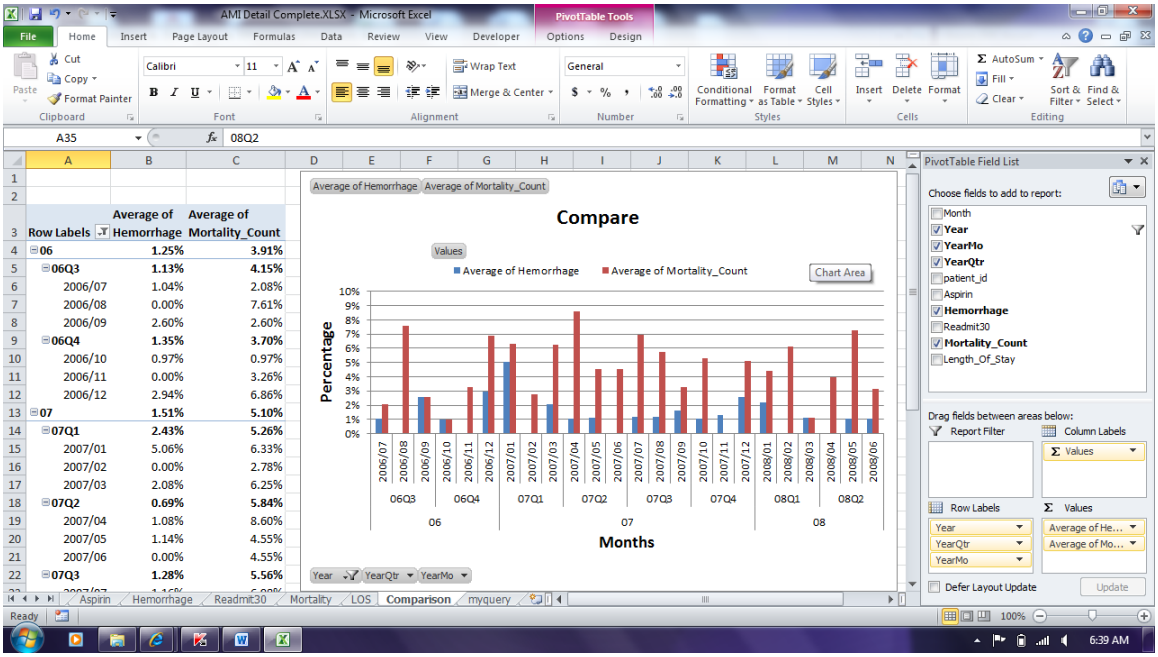

Book2 - Microsoft Excel
PivotChart Tools
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Developer
Design
Layout
Format
Analyze
Change Chart Type
Save As Template
Type
Switch Row/Column
Select Data
Data
Chart Layouts
Chart Styles
Move Chart
Location
Chart 1
PivotTable3
To build a report, choose fields from the PivotTable Field List
Chart 1
To build a PivotChart, choose fields from the PivotTable Field List.
PivotTable Field List
Choose fields to add to report:
admit_date
admit_type
admit_type_desc
age
apr_drg
apr_drg_desc
Aspirin
discharge_date
Drag fields between areas below:
Report Filter
Legend Fields (S...
Axis Fields (Cat...
Values
Defer Layout Update
Update
AMI Dashboard
Sheet2
Sheet3
Ready
100%

10. Add fields to report and format like so:



11. Expand and collapse time periods, Year, YearQtr, YearMo

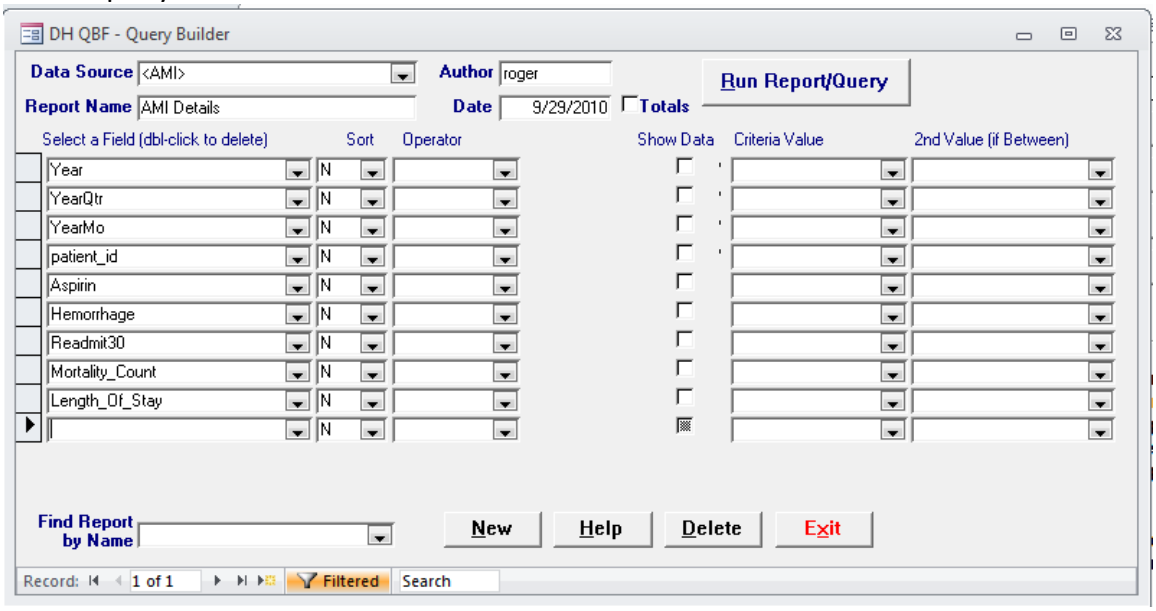
a. Show **AMI Pull Complete.xlsx**. Go to **Comparison** tab and compare Hemorrhage to Mortality Add Mortality. Change to Average.



Pushing the Data with Pivot Tables

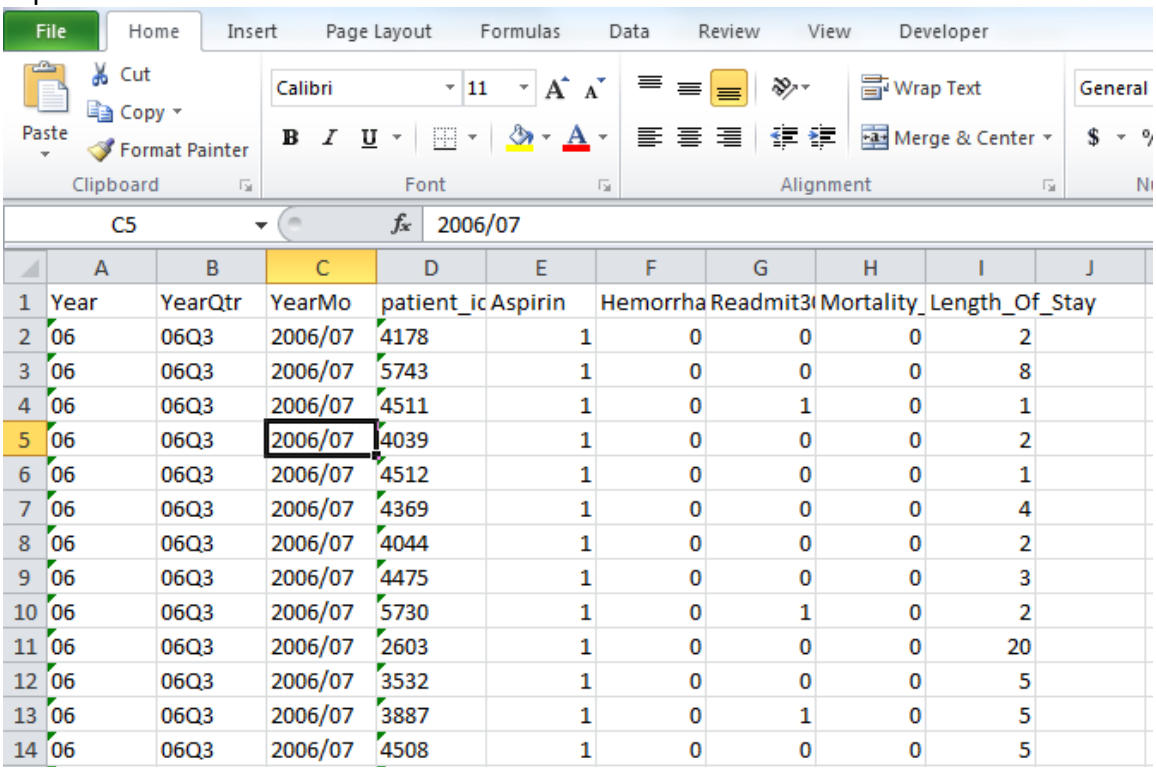
It's also possible to push the data to pivot tables. For that, I'll go back to the Query By Form and create a query that simply exports the data at its lowest granularity.

1. Create query AMI Details.



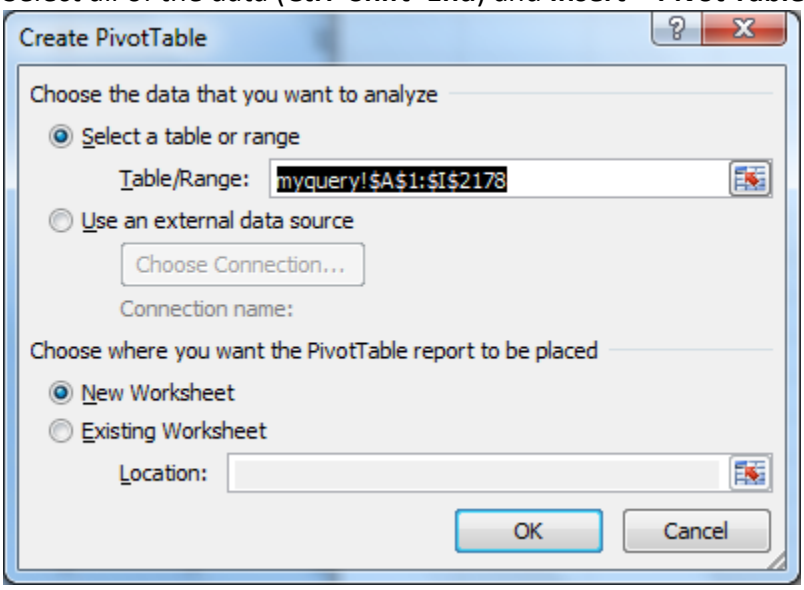
Notice there is no Totals because the Pivot Table will do the aggregation for me.

2. Export the data to Excel:



	A	B	C	D	E	F	G	H	I	J
1	Year	YearQtr	YearMo	patient_ic	Aspirin	Hemorrha	Readmit3	Mortality_	Length_Of_Stay	
2	06	06Q3	2006/07	4178	1	0	0	0	2	
3	06	06Q3	2006/07	5743	1	0	0	0	8	
4	06	06Q3	2006/07	4511	1	0	1	0	1	
5	06	06Q3	2006/07	4039	1	0	0	0	2	
6	06	06Q3	2006/07	4512	1	0	0	0	1	
7	06	06Q3	2006/07	4369	1	0	0	0	4	
8	06	06Q3	2006/07	4044	1	0	0	0	2	
9	06	06Q3	2006/07	4475	1	0	0	0	3	
10	06	06Q3	2006/07	5730	1	0	1	0	2	
11	06	06Q3	2006/07	2603	1	0	0	0	20	
12	06	06Q3	2006/07	3532	1	0	0	0	5	
13	06	06Q3	2006/07	3887	1	0	1	0	5	
14	06	06Q3	2006/07	4508	1	0	0	0	5	

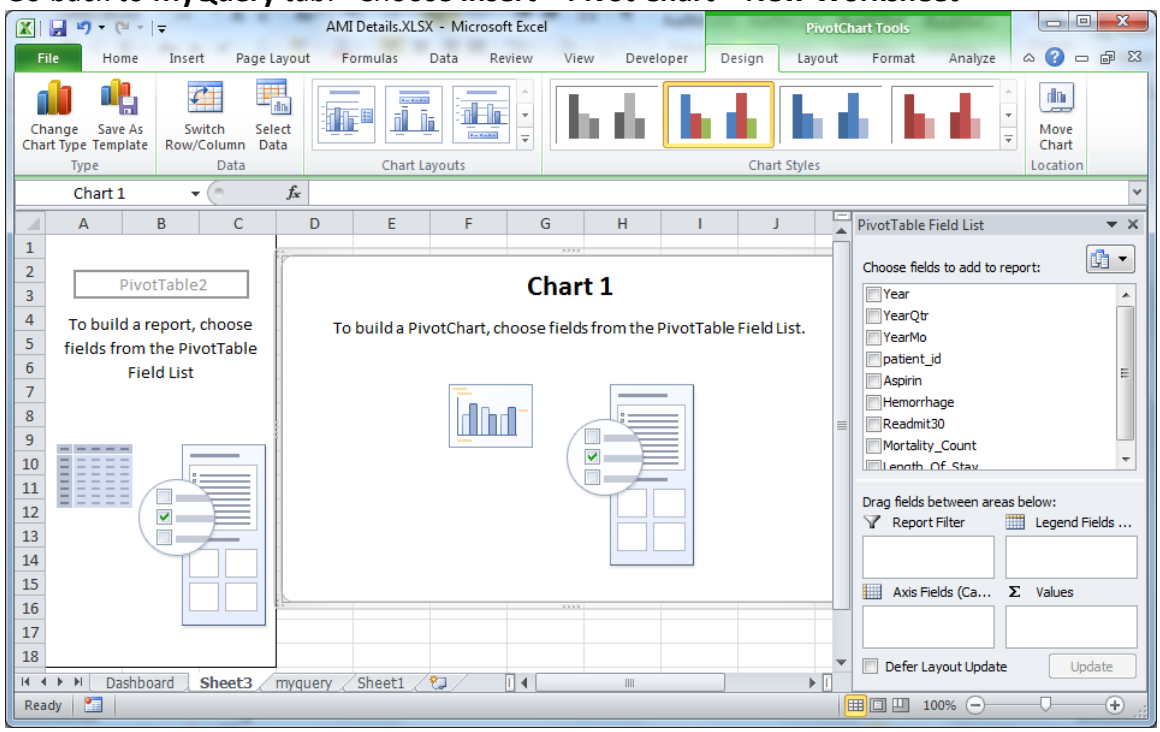
3. Select all of the data (**Ctrl+Shift+End**) and **Insert > Pivot Table**



4. Repeat the steps to create the Dashboard before:
    a. Add year, yearqtr, yearmo to Rows
    b. Add Aspirin, Hemorrhage, etc to Values
    c. Change values to Average
    d. Format values as percentage (LOS as number)

5. Now we have essentially the same report (Dashboard) as before, but this report does not require a connection to the data source because we sent the data on to the workbook. Compare to **AMI Pull Complete**.

6. Go back to **MyQuery** tab. Choose **Insert > Pivot Chart > New Worksheet**



7. Basically, now it's the same steps as before, and I don't really need to go through them again.
8. Open **AMI Details Complete.xlsx**

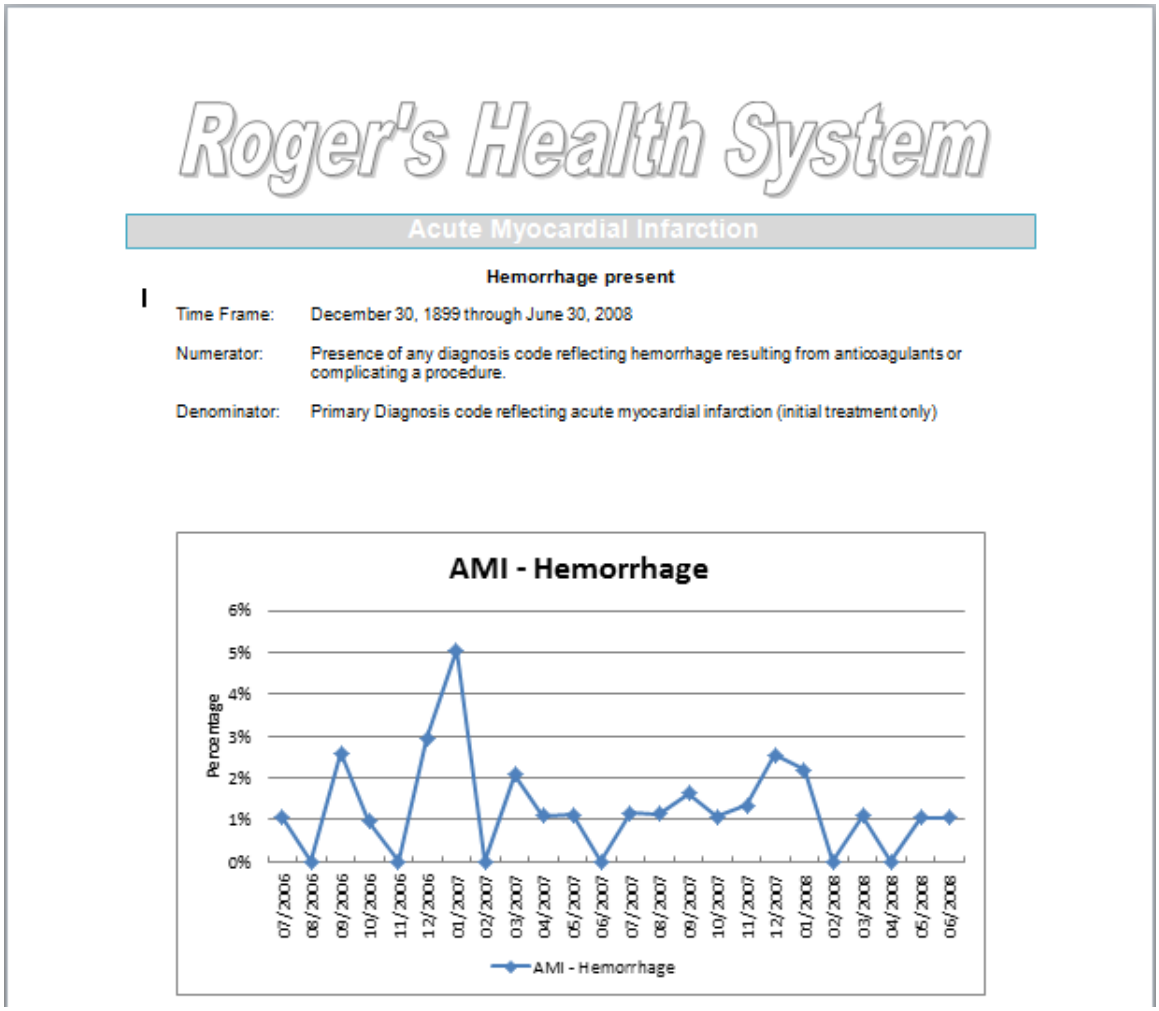
So now we've actually got a mixed system where some reports are static, some reports are pivot tables/charts. Which should you use? It all depends on your business and how the reports need to be used. You may have a decision-maker with limited excel skills. Static Excel dashboards and charts may be perfect for them. For those more experienced, who want to slice and dice, pulling the data into pivot tables may be ideal. For a decision maker who needs to take this information on the road, pushing the data out to stand alone workbooks with pivot tables may be most useful.

The point here is you've got the flexibility to report from your warehouse in any number of ways. By creating a separate reporting system, you give the decision-maker an off-line snapshot of the data that he or she can play with safely on the road, or whatever. Since we've pre-calculated the data, it's small enough to fit in a portable Access file to take on the road.

Custom Report Generator

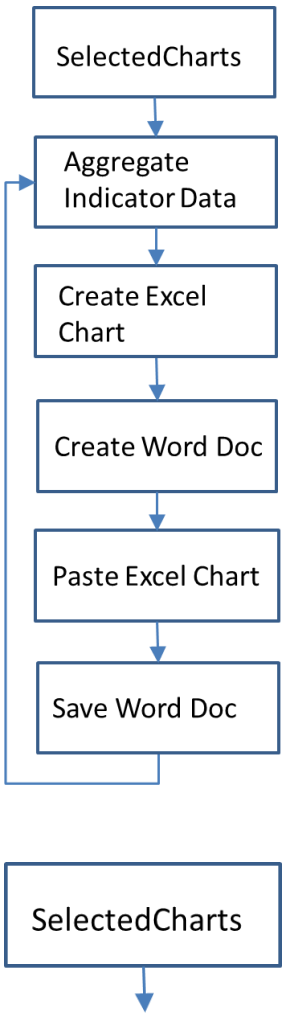
What we've seen so far is great for ad-hoc or periodic reporting by individual decision makers. But in my situation, I'm required to create a 1300 page document with text and charts every month, reporting every condition and indicator. This is too much to do individually with any method I've shown so far.

My solution was to create a custom report generator.



Dashboard

Overall process



Naturally, if I'm going to automate the process, I have to store the parameters necessary to create the report. For instance, the text for the Numerator and Denominator definitions or which indicators belong to which conditions. The Conditions And Indicators form allows me to easily create new conditions and corresponding indicators or modify or delete old ones:

Conditions And Indicators

ConditionID 28 ConditionPrefix: AMI

Condition_Code AMI

Condition_Desc Acute Myocardial Infarction

BaseTable* AMI

BaseTable_Desc* Acute Myocardial Infarction

* differ ONLY for conditions that are Groups ie MMPC

Conditon Milestones

Date Selected	Milestone

ReAdmitDashboard

Close

Indicators

ScalarIndicator	NumDen	Indicator_Code	Indicator_Desc
☑	☐	Age	Patient Age on Admission
☑	☐	apr_drg_weight	apr_drg_weight
☐	☐	Hemorrhage	Hemorrhage present
☑	☐	Length_of_stay	Length of Stay

Record: 1 of 8 Unfiltered Search

Indicator Specifc Milestones

Similarly, if I'm going to create many indicators at once by condition, I need to store information about which indicators belong to which conditions. I call this a Profile and you can see that AMI has for conditions: Hemorrhage, Readmit30, Mortality, and LOS. With the **Edit Profiles** form, it is easy to add or remove indicators from a profile.

Edit Profiles Find Profile AMI

Profile Name: AMI Sequence: 25 Current ☑ UsePrefix ☐ DashboardMonths: 12 ChartMonths: 24

Show Profile

Close Form

Conditions

AMI	Acute Myocardial Infarction
Carotid_Endarterectomy	Carotid_Endarterectomy
EP_Ablation	EP_Ablation
HF	Congestive Heart Failure (HF)
HipReplace	Hip Replacement
KneeReplace	Knee Replacement
Pacemaker_ICDefib	Pacemaker_ICDefib
PCI	PCI
PedsAppendectomy	Pediatric Appendectomy
PedsAsthma	Pediatric Asthma
PedsChemo	Pediatric Chemotherapy
UGIBleed	Gastrointestinal Bleeding and Ulcers

Profile Indicators:

IndicatorID	Sequence
Hemorrhage	0
Readmit30	5
Mortality_Count	10
Length_of_stay	15

All of this information goes to build the **SelectedIndicators** query, the information of which is used to create the Source Data, Excel Chart, and Word Document.

Condition_Code	Condition_Desc	Indicator_Code	Indicator_Desc	Numerator	Denominator
AMI	Acute Myocardial Infarction	Hemorrhage	Hemorrhage present	Presence of any diagnosis code reflecting hemorrhage resulting from anticoagulants or complicating a	Primary Diagnosis code reflecting acute myocardial infarction (initial treatment
AMI	Acute Myocardial Infarction	Length_of_stay	Length of Stay	A numerical figure representing the Length of Stay as calculated by Spectrum Health. A patient admitted	Primary Diagnosis code reflecting acute myocardial infarction (initial treatment
AMI	Acute Myocardial Infarction	Mortality_Count	Mortality Rate Based on Discharge Code	Presence of a match in Discharge Status which relates to Death Discharge Codes (20 through 29 and 40	Primary Diagnosis code reflecting acute myocardial infarction (initial treatment
AMI	Acute Myocardial Infarction	Readmit30	Readmission within 30 Days for Any Reason	Patients readmitted within 30 days regardless of the APR-DRG.	Primary Diagnosis code reflecting acute myocardial infarction (initial treatment

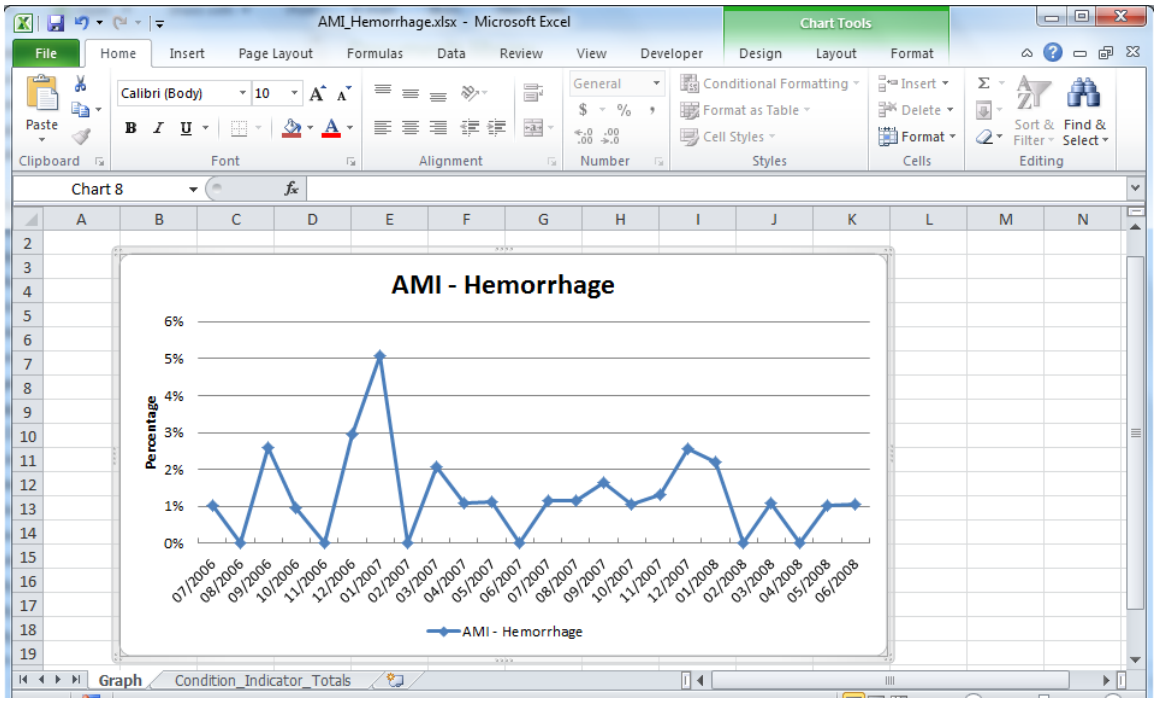
Aggregate Indicator Data

Start with the first indicator information and programmatically create the totals query that aggregates the data (Average by Year/Month) using the information in qrySelectedIndicators.

```
SELECT [month] & "/20" & [year] AS MoYear, avg(Hemorrhage) AS [AMI - Hemorrhage],
NULL AS Percentage
FROM AMI
GROUP BY Year, Month, [month] & "/20" & [year]
ORDER BY Year, Month;
```

Create Excel Chart

Export the data to an Excel template (as we did before) which will feed a chart(AMI_Hemorrhage.xlsx).



Since I'm creating the chart, I might as well save it

Create Word Doc

Open a Word template (much as we opened the Excel template), fill in appropriate information from qrySelectedIndicators

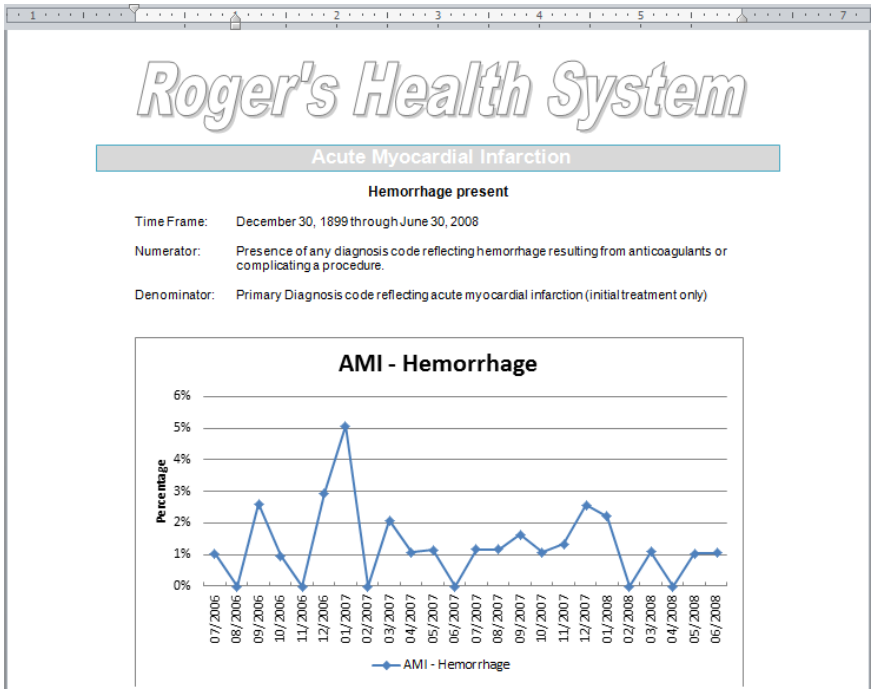
Roger's Health System

Acute Myocardial Infarction

Hemorrhage present

Time Frame:	December 30, 1899 through June 30, 2008
Numerator:	Presence of any diagnosis code reflecting hemorrhage resulting from anticoagulants or complicating a procedure.
Denominator:	Primary Diagnosis code reflecting acute myocardial infarction (initial treatment only)

Paste Excel Chart



Roger's Health System

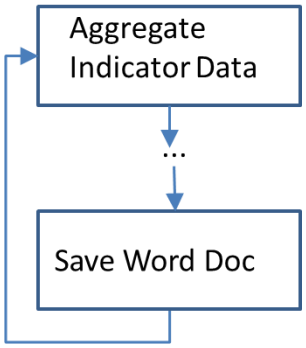
Acute Myocardial Infarction

Hemorrhage present

Time Frame: December 30, 1899 through June 30, 2008

Numerator: Presence of any diagnosis code reflecting hemorrhage resulting from anticoagulants or complicating a procedure.

Denominator: Primary Diagnosis code reflecting acute myocardial infarction (initial treatment only)



Save the Word document. Rinse and Repeat.

Conclusion

Now the purpose of this is obviously not to demonstrate how to create a reporting system for your local healthcare provider. This isn't a heathcare conference. However, I hope I've demonstrated how it is possible and even practical to develop a reporting system for your transactional system based on data warehouse principles that will give you improved performance for you and your users.