

BI4ALL Models New Documentation

8th January 2020
Version V1.0

www.instantbi.com

Hello and Welcome

Hello and welcome to this blog post on how we are now documenting our BI4ALL Models . Thank you very much for taking the time to read this blog post!

First a little history. Back in the 00's, when I was implementing the Sybase version of these sorts of models, they were much harder to understand. We used to plaster the walls with data model diagrams printed from PowerDesigner!

In 2002 I invented a technique which meant that those of us who were very experienced no longer needed pictures to understand the models. But those who were not very experienced still wanted pictures.

The late Sean Kelly (may he rest in peace) constantly asked me to try new ways of printing his version of the models. I tried numerous “data modeling” tools but the time and effort to maintain the models was very high. So we continued to use Excel to develop models and tell our customers “you don't really need diagrams of these models to understand them.”

Well what do you know? All these years later I have stumbled across a great new way to document data models that is much faster and much easier than any of the data modeling tools I have ever seen.

So if you want to create better documentation of data models, especially Data Warehouse Data Models, you will want to read this blog post.

The way we have found is to use the Meta5 DBA tool to draw data model diagrams of the BI4ALL models. We then scale up the VM we are working on and take a screen shot of the data model.

Simply put, we develop our models in normal HD on a 1920x1080 screen on a VM running Windows. We then blow up the screen to UHD, resize the Meta5 desktop to fit the UHD screen size and take screen shots. This works even when the actual physical screen we are using is only 1920x1080.

Sure, we have to do a bit of scrolling around. But I believe you will see that the end result is worth the effort!

Not only can we draw these diagrams but we developed a way of updating the “User Business Names” and “User Help Descriptions” that are presented by the Meta5 DBA Tool directly from the BI4ALL Data Model Workbooks.

The images you see on the following pages are screen captured from the Meta5 DBA Tool. The BI users, when they use Meta5, will see the “User Business Names” that are presented on the following diagrams.

These “User Business Names” are text which is entered in to the BI4ALL Data Model Workbooks. The text can be anything that can go in to an Excel cell and can be up to 128 characters.

Special characters can also be used. For example “%” and “#” signs.

Further, the BI4ALL Models can be easily maintained in up to 10 languages for one data model. This feature has been included in the ability to build Meta5 dictionaries. So business users can be presented with screens that have their fields and their help text in their language.

In short, the data models that you see on the following pages could easily be maintained in up to 10 languages and presented to business users in up to 10 languages. This is particularly important in Europe where so many different languages are used in business.

We are very pleased to be able to develop data models of this detail and place them on a single “page” because this is helping us in our development work.

We would say, even if you do not wish to work with us to implement our BI4ALL models, you might want to use Meta5 and our BI4ALL Models workbooks so you too can develop data model diagrams like the ones you see on the following pages. The BI4ALL Model workbooks and functions come free with SeETL. The data models themselves are not free of course. We still sell them.

We are quite experts in many data modeling tools and we are well aware that it is very time consuming to maintain this level of documentation in one language let alone up to 10 languages.

As at the time of writing the Meta5 Starter Edition only costs USD180 per year. You can get your copy from the Meta5 web site www.meta5.com.

On the remainder of this front page we will show you some pictures of how we manage our data models so you can see the source for the diagrams on the following pages.

On the following pages we present just a very small snippet of the more than 850 tables in the BI4ALL Models. Obviously we are not going to place the entire suite of data models in to the public.

The models are relatively easy to understand by experienced Star Schema Data Modelers.

We are presenting this blog post for your interest and education in that such pictures are now easy to create for very low cost. Of course, if you might see something you like in the models and want the library of models as a resource for your company? We sell the models.

The newly developed ability to integrate our BI4ALL Models workbooks with the Meta5 DBA tool has enabled us to improve our ability to document our data models. I believe many of my BI contacts would like to have the same ability to create the same sort of documentation for your company.

In our BI4ALL Models Workbook (Excel) we record information about the views that will present data out to the business users.

For example, in the case below, you can see we are recording the view name, the submodel of the view, the view number, & the view description in language 01.

You can see that there is a spot for View Description Language 02 but it is not filled in.

If it was filled in this second language could be used to maintain another Meta5 Dictionary that could be used to document the data model in that second language.

Use	This Sheet contains the BI4ALL Models Definitions and Directives					
N	Number of Languages Definitions					
N	Row Typ Number of Languages					
Y	nl	1				
N	Input Output Definitions					
N	Row Typ	Target Dire	Target Directory	Lang 02	Target Directory La	Target Direct
Y	dir	####	Create na	na	na	na
N	View Definitions and Short Descriptions					
N	Row Typ	View Owner	View Name	Submodel	View Number	View Description Lang01
Y	vdsd	dbo	vm_party	BI4ALLBASE	0	d-Party
Y	vdsd	dbo	vm_prospect	BI4ALLBASE	0	d-Prospect
Y	vdsd	dbo	vm_person	BI4ALLBASE	0	d-Person
Y	vdsd	dbo	vm_person_preparer	BI4ALLBASE	0	d-Person Preparer
Y	vdsd	dbo	vm_person_vendor	BI4ALLBASE	0	d-Person Vendor
Y	vdsd	dbo	vm_sales_rep	BI4ALLBASE	0	d-Sales Rep
Y	vdsd	dbo	vm_primary_sales_rep	BI4ALLBASE	0	d-Primary Sales Rep
Y	vdsd	dbo	vm_account_close_rsn_code	BI4ALLBASE	101	d-Account Close Rsn Code
Y	vdsd	dbo	vm_account_status	BI4ALLBASE	102	d-Account Status
Y	vdsd	dbo	vm_ap_account_status	BI4ALLBASE	123	d-AP Account Status
Y	vdsd	dbo	vm_adjustment_type	BI4ALLBASE	103	d-Adjustment Type
Y	vdsd	dbo	vm_ap_invoice_status	BI4ALLBASE	104	d-AP Invoice Status

Next we have to maintain the views over the actual tables. This is done in the same worksheet but with columns of a different format. You can see the format below.

You can see that the worksheet contains the table name, and the column name of the column on the underlying table.

You can then see that there is a view name lang 01 placed over the top of the table. We can actually also store the view names in up to 10 languages over the base tables which are always in English.

We then have View Column Lang 01 which is the column name of the view in English.

We can also have view column names in up to 10 languages if that is desired.

Next you can see the Short Help Text Lang 01. This is the field we are using to be the name of the field in the Meta5 dictionary. Meta5 will use the View Column Lang 01 field as the column name of the view it reads. This is the name that will be generated in the SQL statements generated by Meta5.

However, the name in Short Help Text Lang01 is the name that will be presented in the Meta5 DBA Tool and in all the Meta5 query tools.

N	Row	Type	Table	Table name	Col name	View	View Name	View Column Lang 01	View Column Short He	View Column Long Help Text Lang01
Y	vcd	b	dbo	TD0100	pk_TD0100	dbo	vm_person	pk_vm_person	Pk Vm Person	Primary key for vm_person
Y	vcd	m	dbo	TD0100	varchar255_01	dbo	vm_person	prefix	Prefix	Partys prefix for addressing. Eg. Mr. Mrs. Dr. Sir
Y	vcd	m	dbo	TD0100	varchar255_02	dbo	vm_person	first_name	First Name	The first name of the party
Y	vcd	m	dbo	TD0100	varchar255_03	dbo	vm_person	second_name	Second Name	This is the second name of the party
Y	vcd	m	dbo	TD0100	varchar255_04	dbo	vm_person	third_name	Third Name	This is the third name of the party
Y	vcd	m	dbo	TD0100	varchar255_05	dbo	vm_person	surname	Surname	This is the surname name of the party
Y	vcd	m	dbo	TD0100	varchar255_06	dbo	vm_person	suffix	Suffix	This is the suffix to the name of the party
Y	vcd	m	dbo	TD0100	varchar255_07	dbo	vm_person	concat_name	Concat Name	Subscribers concatenated name to be placed on mailings and messages.
Y	vcd	m	dbo	TD0100	varchar255_08	dbo	vm_person	address_to_name	Address To Name	The name that is used on the outside of a letter to address the letter to.
Y	vcd	m	dbo	TD0100	varchar255_09	dbo	vm_person	formal_salutation	Formal Salutation	If the person is to be addressed formally
Y	vcd	m	dbo	TD0100	varchar255_10	dbo	vm_person	informal_salutation	Informal Salutation	Preferred salutation the person may have asked to be used
Y	vcd	m	dbo	TD0100	varchar255_11	dbo	vm_person	written_salutation	Written Salutation	What you should put at the top of a letter
Y	vcd	m	dbo	TD0100	varchar255_12	dbo	vm_person	email_address	Email Address	This is the email address of the party
Y	vcd	m	dbo	TD0100	varchar255_13	dbo	vm_person	web_address	Web Address	Lots of people have their own web addresses now.
Y	vcd	m	dbo	TD0100	varchar255_14	dbo	vm_person	tax_number	Tax Number	This is the tax number for the party.
Y	vcd	m	dbo	TD0100	varchar255_15	dbo	vm_person	govt_identifier_type	Govt Identifier Type	The type of government identifier used to identify the subscriber to provide the s
Y	vcd	m	dbo	TD0100	varchar255_16	dbo	vm_person	govt_identifier	Govt Identifier	The actual number of the government provided identifier.
Y	vcd	m	dbo	TD0100	varchar255_17	dbo	vm_person	ss_person_key	Ss Person Key	Source system person key
Y	vcd	m	dbo	TD0100	varchar255_18	dbo	vm_person	marital_status	Marital Status	Marital status. Eg. Single; Married; Separated; Divorced; Widowed etc
Y	vcd	m	dbo	TD0100	varchar255_19	dbo	vm_person	phone_home_num	Phone Home Num	The home phone number of the party
Y	vcd	m	dbo	TD0100	varchar255_20	dbo	vm_person	phone_work_num	Phone Work Num	This is the work phone number of party
Y	vcd	m	dbo	TD0100	varchar255_21	dbo	vm_person	phone_mobile_num	Phone Mobile Num	This is the mobile number of the party
Y	vcd	m	dbo	TD0100	varchar255_22	dbo	vm_person	phone_fax_num	Phone Fax Num	The main fax number of the party
Y	vcd	m	dbo	TD0100	date_01	dbo	vm_person	birthdate	Birthdate	The date of birth of the person this row represents.

We are using the view “short description” in the BI4ALL Models Workbook to be the view name that is presented in the data model. The view “long description” is then used as the “help text” for the view name.

To alter a view name in the data model the DBA merely updates this workbook and then runs some commands to get the changes to flow through to the Meta5 dictionary.

It is possible to maintain very complex data models in our BI4ALL Models Workbook.

It has never been possible to draw diagrams of them easily.

Sales Transactions Category

Notes

Below is an image of the BI4ALL Sales Category loaded in to the Meta5 DBA Tool. You can see from the top left hand corner there is "01-BI4ALL-DBA" mentioned in the DBA tool header window. Inside that window we have opened the "Sales Transaction" Category window by clicking on the category and clicking the "show joins" button.

In the Meta5 DBA tool a collection of tables that are routinely queried together is called a "Category". You can have as many categories as you like. Tables can appear in as many categories as you like. In this way you build a library of query icons linked to specific categories with specific parameters. When users want to build a query they go to the filing cabinet for that area of query and select the query icon that most closely fits their new need and copy it to the capsule they are developing.

Although the diagram looks a little complicated it is actually very simple once you understand it. All of BI4ALL is actually very simple once you understand it very well. It is part of what makes building BI4ALL models so fast.

You can see that dimension tables have been prefixed with "d-" and fact tables with "f-". So you can see obvious dimension tables like "Sale Date", "Customers", "Customer Demographics", "Product", "Sales Rep" etc.

In the first column of each table you can see "→" pointing to the first column with the table name in the first column. This represents the primary key of the table. All BI4ALL tables have a unique integer primary key. Fact tables have a bigint primary key. So you can see "→ f-Sale Txn" as the first column of the f-Sale Txn table.

In the f-Sale Txn table you can see many links to many dimension tables. The "←" represents a foreign key to another table and it names the table it is pointing to. Now, with 850+ tables and thousands of joins, please excuse us if we have missed one or two, ok?

So, for example, on the f-Sale-Txn table you can see the links to d-Sale Date, d-Product, d-Customers etc. You will also notice links to fact tables such as f-Solicitation-Txn. So if this sale was the result of a solicitation then we can link that solicitation record to the sale.

By looking at the dimension table columns you can get a very good idea of what is in the dimension table. The obvious example being d-Customers which contains fields such as names, email address, web address, government identifiers and many other things that are standard attributes of customers. The way we teach new people how to read the data models is to simply get them to read all the dimension tables and start memorizing what is on all the dimension tables.

Then they read the fact tables and start memorizing the fact tables. Then, as a matter of faith, they just presume that if two things should be linked they will be. Or linking them is very easy.

In fact tables we limit the fields in the standard models to only those fields that are very common. So in the f-Sale Txn fact table you will see fields like Sale Unit Amount, Sale Extended Amount, Cost Unit Amount, Cost Extended Amount, Sale Units, Taxes and short and long descriptions of the product sold.

For retailers we include a return flag, a return probability, and a flag to say the return period has expired. By providing just the very common facts that are found on specific fact tables, like sales facts, BI4ALL allows you to create your own fact fields with their own names. You can use those names you are already used to rather than our field names.

Lastly, towards the right hand side of the diagram you will see three tables that will look "strange" to you. These are f-Account Dims Asoc, f-Customer Dims Asoc and f-Campaign Dims Asoc. They will look strange to you because they don't actually have fields that look like facts on them. They just have links to dimension tables and a set of date from, date to and current flag fields.

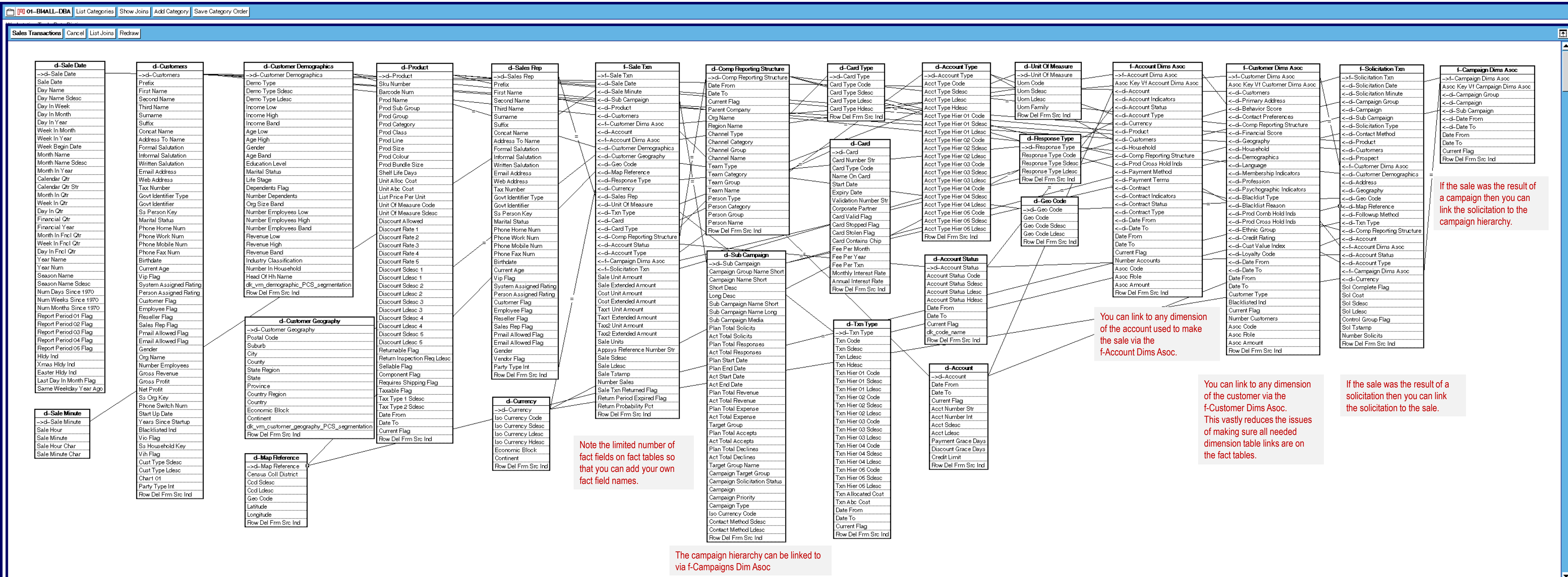
These will be explained in a little more detail on later slides. These are called "association tables" and they "associate" something for a period of date from to date to with the current association having current flag set to 1.

These association tables record history in the manner Bill Inmon proposed in this Time Variance and Stability Analysis (TV+SA) models in the late 80s. Associations achieve the same outcome Bill Inmon recommended, only cheaper.

Associations were the idea that allowed us to implement Bill Inmons ("TV+SA") proposed models perfectly inside a the Star Schema Models proposed by Ralph Kimball.

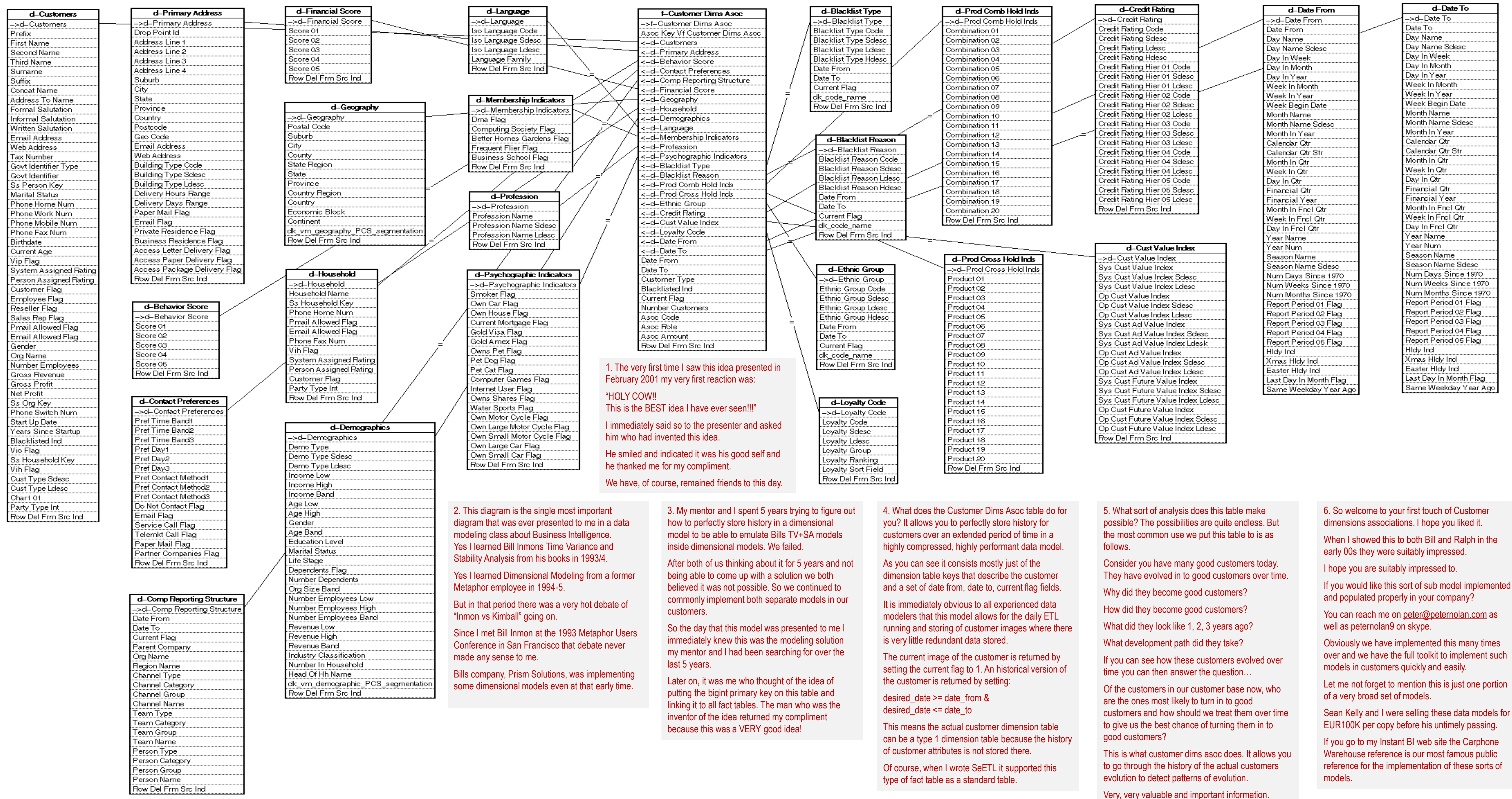
Both Bill and Ralph are friends of mine and I showed them how this works back in the early 00s when I first came across this design technique at Sybase. The idea I added to this design technique was to put a bigint key on the association table and link the association table to all transactions. That was my idea. And you can see here that the f-Customer Dims Asoc table is linked to both the f-Solicitation Txn table and the f-Sales Txn table.

We have been doing this since 2002 to very good effect. Now I am making that little gem of an idea public. These association tables provide what we now call "the mesh" in that they link everything to everything that should be linked. Over the last 18 years "the mesh" has made data model development much faster and much easier for us as we implemented the various versions of these data models all around the world.



Customer Dimensions Association Category

Customer Dims Assoc Cancel List Joins Redraw



1. The very first time I saw this idea presented in February 2001 my very first reaction was: "HOLY COW!! This is the BEST idea I have ever seen!!!" I immediately said so to the presenter and asked him who had invented this idea. He smiled and indicated it was his good self and he thanked me for my compliment. We have, of course, remained friends to this day.

2. This diagram is the single most important diagram that was ever presented to me in a data modeling class about Business Intelligence. Yes I learned Bill Inmons Time Variance and Stability Analysis from his books in 1993/4. Yes I learned Dimensional Modeling from a former Metaphor employee in 1994-5. But in that period there was a very hot debate of "Inmon vs Kimball" going on. Since I met Bill Inmon at the 1993 Metaphor Users Conference in San Francisco that debate never made any sense to me. Bills company, Prism Solutions, was implementing some dimensional models even at that early time.

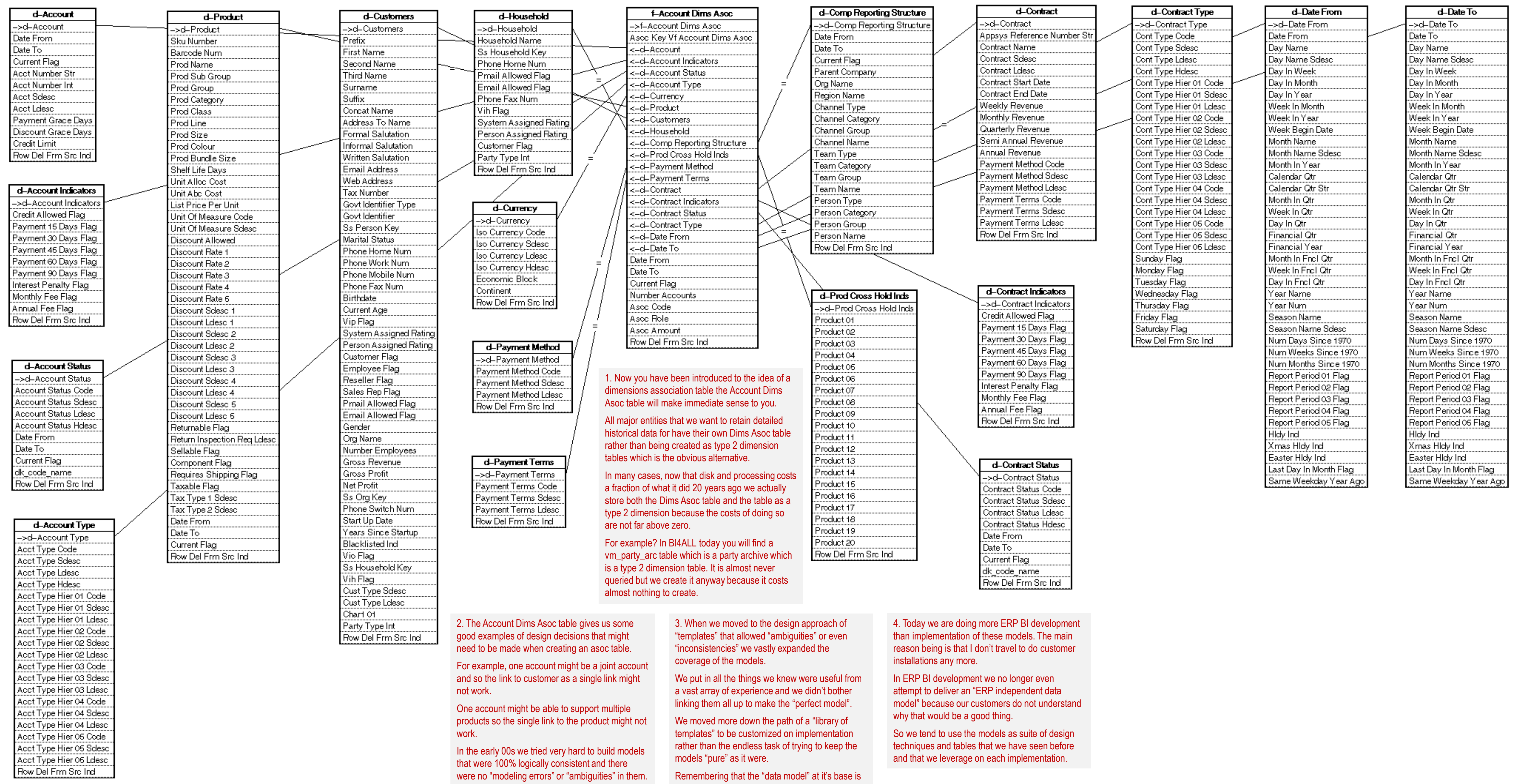
3. My mentor and I spent 5 years trying to figure out how to perfectly store history in a dimensional model to be able to emulate Bills TV+SA models inside dimensional models. We failed. After both of us thinking about it for 5 years and not being able to come up with a solution we both believed it was not possible. So we continued to commonly implement both separate models in our customers. So the day that this model was presented to me I immediately knew this was the modeling solution my mentor and I had been searching for over the last 5 years. Later on, it was me who thought of the idea of putting the bigint primary key on this table and linking it to all fact tables. The man who was the inventor of the idea returned my compliment because this was a VERY good idea!

4. What does the Customer Dims Assoc table do for you? It allows you to perfectly store history for customers over an extended period of time in a highly compressed, highly performant data model. As you can see it consists mostly just of the dimension table keys that describe the customer and a set of date from, date to, current flag fields. It is immediately obvious to all experienced data modelers that this model allows for the daily ETL running and storing of customer images where there is very little redundant data stored. The current image of the customer is returned by setting the current flag to 1. An historical version of the customer is returned by setting:
desired_date >= date_from &
desired_date <= date_to
This means the actual customer dimension table can be a type 1 dimension table because the history of customer attributes is not stored there. Of course, when I wrote SeETL it supported this type of fact table as a standard table.

5. What sort of analysis does this table make possible? The possibilities are quite endless. But the most common use we put this table to is as follows. Consider you have many good customers today. They have evolved in to good customers over time. Why did they become good customers? How did they become good customers? What did they look like 1, 2, 3 years ago? What development path did they take? If you can see how these customers evolved over time you can then answer the question... Of the customers in our customer base now, who are the ones most likely to turn in to good customers and how should we treat them over time to give us the best chance of turning them in to good customers? This is what customer dims assoc does. It allows you to go through the history of the actual customers evolution to detect patterns of evolution. Very, very valuable and important information.

6. So welcome to your first touch of Customer dimensions associations. I hope you liked it. When I showed this to both Bill and Ralph in the early 00s they were suitably impressed. I hope you are suitably impressed to. If you would like this sort of sub model implemented and populated properly in your company? You can reach me on peter@peterolan.com as well as peterolan9 on skype. Obviously we have implemented this many times over and we have the full toolkit to implement such models in customers quickly and easily. Let me not forget to mention this is just one portion of a very broad set of models. Sean Kelly and I were selling these data models for EUR100K per copy before his untimely passing. If you go to my Instant BI web site the Carphone Warehouse reference is our most famous public reference for the implementation of these sorts of models.

Account Dimensions Association



1. Now you have been introduced to the idea of a dimensions association table the Account Dims Assoc table will make immediate sense to you.

All major entities that we want to retain detailed historical data for have their own Dims Assoc table rather than being created as type 2 dimension tables which is the obvious alternative.

In many cases, now that disk and processing costs a fraction of what it did 20 years ago we actually store both the Dims Assoc table and the table as a type 2 dimension because the costs of doing so are not far above zero.

For example? In BI4ALL today you will find a vm_party_arc table which is a party archive which is a type 2 dimension table. It is almost never queried but we create it anyway because it costs almost nothing to create.

2. The Account Dims Assoc table gives us some good examples of design decisions that might need to be made when creating an assoc table.

For example, one account might be a joint account and so the link to customer as a single link might not work.

One account might be able to support multiple products so the single link to the product might not work.

In the early 00s we tried very hard to build models that were 100% logically consistent and there were no "modeling errors" or "ambiguities" in them. By 2010 it had become obvious this was not a useful strategy and we moved to the idea of building "templates" to be updated on implementation more than complete models.

3. When we moved to the design approach of "templates" that allowed "ambiguities" or even "inconsistencies" we vastly expanded the coverage of the models.

We put in all the things we knew were useful from a vast array of experience and we didn't bother linking them all up to make the "perfect model".

We moved more down the path of a "library of templates" to be customized on implementation rather than the endless task of trying to keep the models "pure" as it were.

Remembering that the "data model" at it's base is just a spreadsheet that can be very easily added to, we just add to the spreadsheet when we have something new to add.

4. Today we are doing more ERP BI development than implementation of these models. The main reason being is that I don't travel to do customer installations any more.

In ERP BI development we no longer even attempt to deliver an "ERP independent data model" because our customers do not understand why that would be a good thing.

So we tend to use the models as suite of design techniques and tables that we have seen before and that we leverage on each implementation.

d-Campaign Group
->d-Campaign Group
Campaign Group Name Short
Campaign Group Name Long
Short Desc
Long Desc
Purpose Sdesc
Purpose Ldesc
Row Del Frm Src Ind

f-Campaign Dims Assoc
->f-Campaign Dims Assoc
Assoc Key Wf Campaign Dims Assoc
<-d-Campaign Group
<-d-Campaign
<-d-Sub Campaign
<-d-Date From
<-d-Date To
Date From
Date To
Current Flag
Row Del Frm Src Ind

d-Sub Campaign
->d-Sub Campaign
Campaign Group Name Short
Campaign Name Short
Short Desc
Long Desc
Sub Campaign Name Short
Sub Campaign Name Long
Sub Campaign Media
Plan Total Solicits
Act Total Solicits
Plan Total Responses
Act Total Responses
Plan Start Date
Plan End Date
Act Start Date
Act End Date
Plan Total Revenue
Act Total Revenue
Plan Total Expense
Act Total Expense
Target Group
Plan Total Accepts
Act Total Accepts
Plan Total Declines
Act Total Declines
Target Group Name
Campaign Target Group
Campaign Solicitation Status
Campaign
Campaign Priority
Campaign Type
Iso Currency Code
Contact Method Sdesc
Contact Method Ldesc
Row Del Frm Src Ind

d-Date From
->d-Date From
Date From
Day Name
Day Name Sdesc
Day In Week
Day In Month
Day In Year
Week In Month
Week In Year
Week Begin Date
Month Name
Month Name Sdesc
Month In Year
Calendar Qtr
Calendar Qtr Str
Month In Qtr
Week In Qtr
Day In Qtr
Financial Qtr
Financial Year
Month In FncI Qtr
Week In FncI Qtr
Day In FncI Qtr
Year Name
Year Num
Season Name
Season Name Sdesc
Num Days Since 1970
Num Weeks Since 1970
Num Months Since 1970
Report Period 01 Flag
Report Period 02 Flag
Report Period 03 Flag
Report Period 04 Flag
Report Period 05 Flag
Hldy Ind
Xmas Hldy Ind
Easter Hldy Ind
Last Day In Month Flag
Same Weekday Year Ago

d-Date To
->d-Date To
Date To
Day Name
Day Name Sdesc
Day In Week
Day In Month
Day In Year
Week In Month
Week In Year
Week Begin Date
Month Name
Month Name Sdesc
Month In Year
Calendar Qtr
Calendar Qtr Str
Month In Qtr
Week In Qtr
Day In Qtr
Financial Qtr
Financial Year
Month In FncI Qtr
Week In FncI Qtr
Day In FncI Qtr
Year Name
Year Num
Season Name
Season Name Sdesc
Num Days Since 1970
Num Weeks Since 1970
Num Months Since 1970
Report Period 01 Flag
Report Period 02 Flag
Report Period 03 Flag
Report Period 04 Flag
Report Period 05 Flag
Hldy Ind
Xmas Hldy Ind
Easter Hldy Ind
Last Day In Month Flag
Same Weekday Year Ago

d-Campaign
->d-Campaign
Campaign Group Name Short
Short Desc
Long Desc
Campaign Name Short
Campaign Name Long
Campaign Priority
Campaign Type
Plan Total Solicits
Act Total Solicits
Plan Total Responses
Act Total Responses
Plan Start Date
Plan End Date
Act Start Date
Act End Date
Plan Total Revenue
Act Total Revenue
Plan Total Expense
Act Total Expense
Target Group
Plan Total Accepts
Act Total Accepts
Plan Total Declines
Act Total Declines
Partition Column Part 01
clk_campaign_start_month
clk_campaign_start_day
clk_campaign_end_month
clk_campaign_end_day
Iso Currency Code
Row Del Frm Src Ind

1. Our long experience shows us that 75% of the value of doing BI is in increasing sales to make more profit. Only 25% of the value of doing BI is in decreasing costs to produce more profit. Even though this is true the overwhelming majority of BI projects focus on decreasing costs rather than increasing sales.

Because we know that there is much more profit to be made from increasing sales BI4ALL has a very robust Campaign Management component. Within BI4ALL a "campaign" is defined to be:

"Any business initiative intended to influence the profitability of the business".

So "campaigns" can also be training programs for staff, especially sales staff.

2. You can see on the model that there are three levels of hierarchy in campaigns.

Campaign Group.
Campaign.
Sub Campaign.

You can see the different attributes of each level. These three groupings have proven themselves to be very useful over an extended period of time.

You can see the plan vs actuals for such things as solicitations, responses, revenues, expenses, accepts and declines.

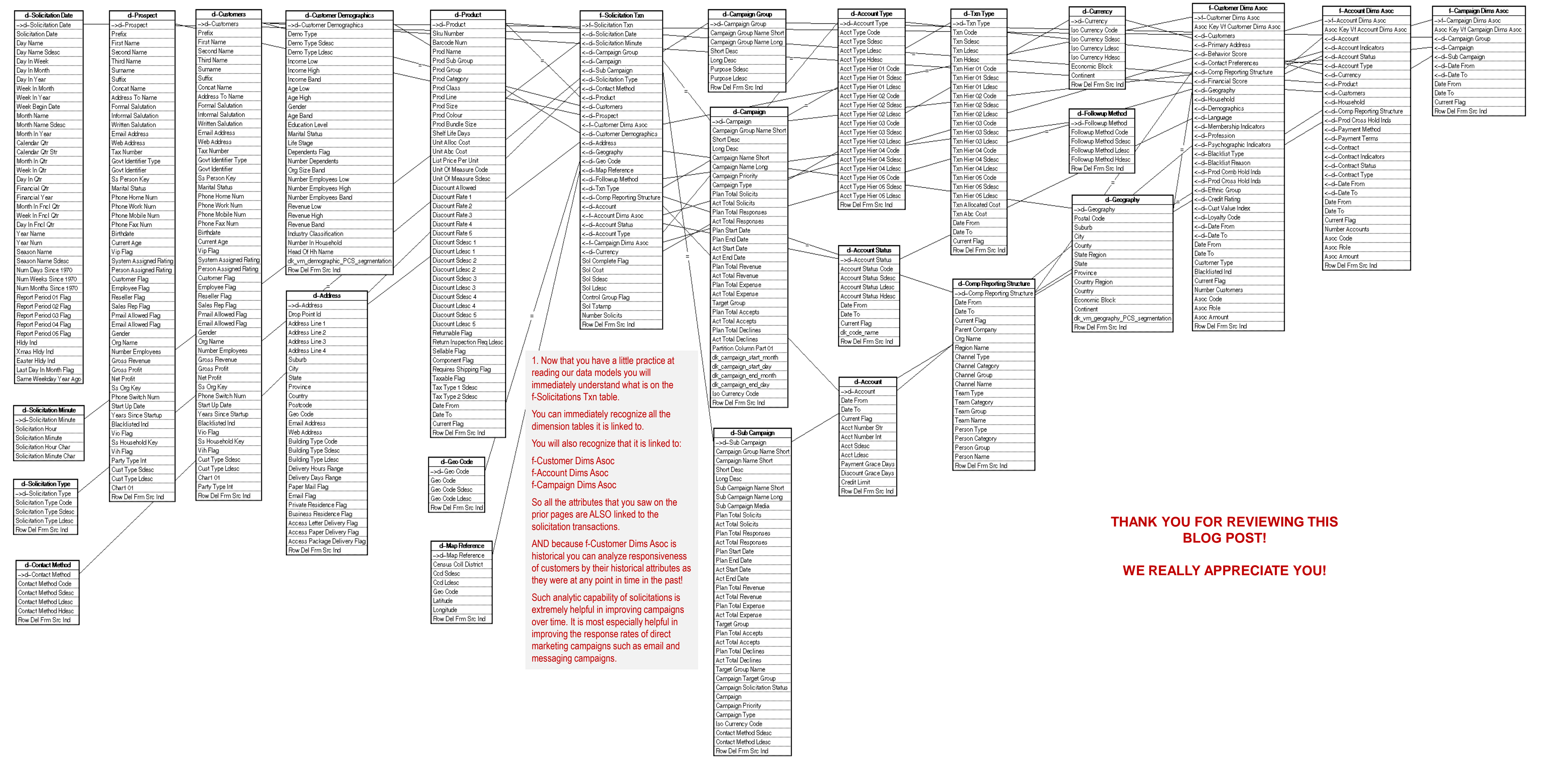
And the campaign dims asoc key can be placed on to any sales record where the sale can be sensibly tied to the campaign whether it is a direct response campaign or not.

3. In terms of campaigns the bottom line for most marketing directors is this:

Half your marketing budget is wasted.

We can show you which half and you can redeploy that marketing money to be useful.

Solicitations Transactions Category



1. Now that you have a little practice at reading our data models you will immediately understand what is on the f-Solicitations Txn table.

You can immediately recognize all the dimension tables it is linked to.

You will also recognize that it is linked to:

- f-Customer Dims Assoc
- f-Account Dims Assoc
- f-Campaign Dims Assoc

So all the attributes that you saw on the prior pages are ALSO linked to the solicitation transactions.

AND because f-Customer Dims Assoc is historical you can analyze responsiveness of customers by their historical attributes as they were at any point in time in the past!

Such analytic capability of solicitations is extremely helpful in improving campaigns over time. It is most especially helpful in improving the response rates of direct marketing campaigns such as email and messaging campaigns.

THANK YOU FOR REVIEWING THIS BLOG POST!

WE REALLY APPRECIATE YOU!