

Data Warehouse Newsletter #7

A Data Warehouse Generator

Details

Version 1.0

16th October 2002

Author: Peter Nolan

pnolan@ozemail.com.au

Table of Contents

1. CHANGE CONTROL LOG	3
2. AUDIENCE	4
3. WHAT DATABASE DESIGN METHODS ARE SUPPORTED?	5
4. HIGH LEVEL DESIGN FEATURES	6
5. STAR SCHEMA MANAGEMENT	6
6. THIRD NORMAL FORM SUPPORT	7
7. TIME VARIANCE WITH STABILITY ANALYSIS SUPPORT	8
8. GENERAL FEATURES	9
9. WHAT TABLES EXIST?	10
10. WHAT PROGRAMS EXIST?	11
11. CURRENT LIMITS?	13

1. CHANGE CONTROL LOG

#	Date	Name	Description
1.0	16/10/02	P. Nolan	Initial publication to the web.

2. AUDIENCE

The intended audience for this Newsletter is:

- IT Managers responsible for Data Warehouse Initiatives.
- Technical developers on the Data Warehouse project.

This document is intended to be read by IT Professionals who are:

- Considering building their first iteration of a Data Warehouse.
- In the learning phases of how to write the ETL for the Data Warehouse.
- In later iterations of the Data Warehouse but wondering if there is a better way to go about populating star schemas or time variant models.

Reading this document is still relevant if you are buying an ETL tool since this document will allow you to be clearer on what the ETL tools do not do.

3. WHAT DATABASE DESIGN METHODS ARE SUPPORTED?

The three most used database design methods for the Data Warehouse are supported:

1. Third Normal Form

The simplest of database schemas, the Third Normal Form, is supported by transferring data directly from operational tables, including the ability to reformat and or transform the data in the record, and then loading it into the Data Warehouse using insert/update. If required the record can be made time variant so that many versions of the record can be inserted into the Data Warehouse. The Third Normal Form tables for a Data Warehouse accommodate the low volume of change tables.

2. Stability Analysis, Access Analysis and Time Variance

When the volume of change of data elements on a record varies greatly there is significant advantage in splitting the record according to data element volatility. For example a record that stores bank account details may contain the opening branch number as well as the balance. If one stores a new copy of the entire record on a daily or weekly basis then one would be storing much unchanged data for accounts as the opening branch would be replicated in each record. If the record were split into three then the opening branch may be stored in one row and the account balance could be stored in another row. This results in dramatic efficiencies of disk usage. This splitting of the record into three levels of volatility, high, medium and low is known as stability analysis. Another criteria for splitting the operational record is the frequency with which the field is likely to be accessed. If there are fields that are very rarely accessed in the Data Warehouse they can be split out to separate records and stored physically away from the data that is accessed frequently.

The Data Warehouse Generator allows for splitting an incoming record from the operational system into three distinct records for high, medium and low volatility areas of the records.

3. Star Schema

The Star Schema is easily the most efficient method of storing, accessing and analysing large volumes of transaction records. Today, it is the industry standard way of storing, managing and manipulating very large volumes of data. The Data Warehouse Generator was first conceived as a fast way of implementing a Star Schema so that colleagues could learn quickly how it worked. The Data Warehouse Generator manages incoming transactions with character keys and performs all processing required to store the transactions in the Star Schema. The Data Warehouse Generator is well worth investigating simply to get working code that manages a Star Schema.

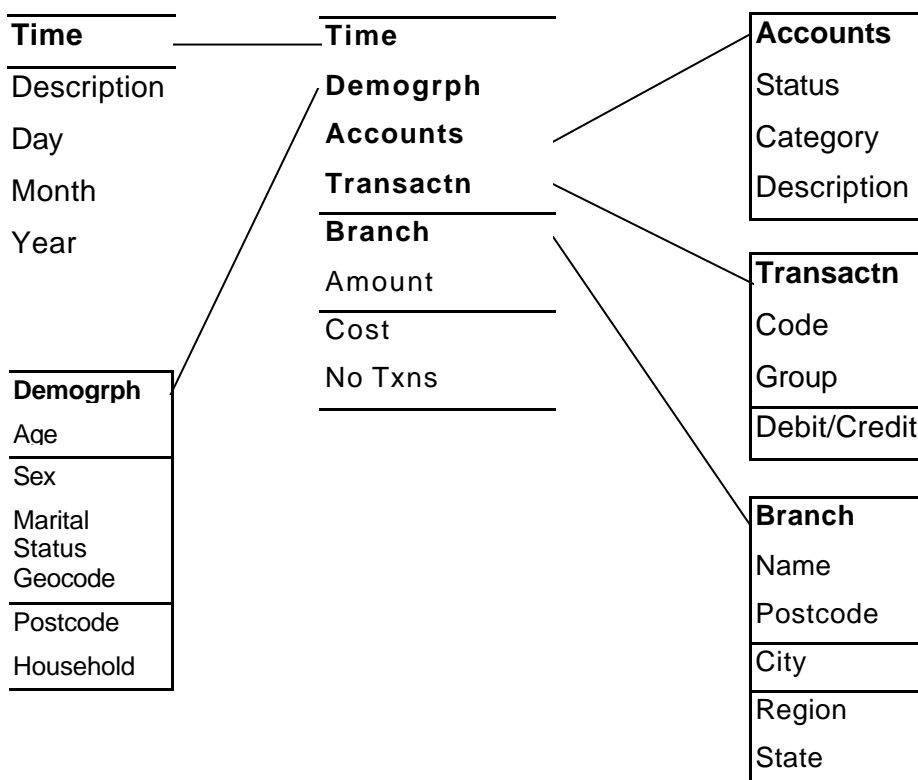
4. HIGH LEVEL DESIGN FEATURES

This section describes in one or two short paragraphs the various features of the Data Warehouse Generator.

5. STAR SCHEMA MANAGEMENT

The Data Warehouse Generator has the following features in relationship to Star Schema databases.

Firstly, what does a Star Schema look like? The following diagram is a simple Star Schema that might be useful for a banking business.



Input Transaction Record

The transaction record to be warehoused can contain as many fields of as many types as are supported by the selected database manager.

Dimension Tables

There can be as many dimension tables as can be supported by the selected database manager.

Each dimension table can have up to nine summary levels defined for the dimension table. Usually the top level is known as 'Total' though this is not a fixed requirement. The number of aggregate levels for dimensions can be changed however the Data Warehouse Generator assumes that the number of summary levels possible (not necessarily used) for each dimension is the same.

Dimension tables are designed so that you add any number of columns you would like. Thus the dimension tables can contain anything that you want to select by. There are only a small number of required fields, all others are for selection and query processing.

Fact Tables

There can be any number of fact tables. You can simply keep altering the parameter files to increase the number of fact tables.

Number of Summary Levels

One of the great features of the Star Schema is that one can maintain many levels of summary information in a single fact table. It can also distribute summaries over a number of fact tables. For example, because the code is generated it is possible to create a single table for every summary level and not pay a code maintenance penalty. This would require an aggregate level navigator. The Data Warehouse Generator allows any number of summary levels to be placed into any number of summary fact tables. Since each dimension table can contain up to 9 levels of aggregate the number of combinations of summaries is $10 \times 10 \times 10 \dots$ for as many dimension tables can have summaries.

The way that aggregates are defined is by the aggregate control table. Because the aggregation control table is keyed on fact table name you can place any number of different summaries in any number of different fact tables should you so desire. Experience has shown that for **very** large fact tables it sometimes helps to split some of the summaries into different tables to make the indexes less deep.

The aggregation control table contains a record for each aggregation level that a specific fact table can have setting the level as integers from 0-9.

Incremental Updates of Summaries

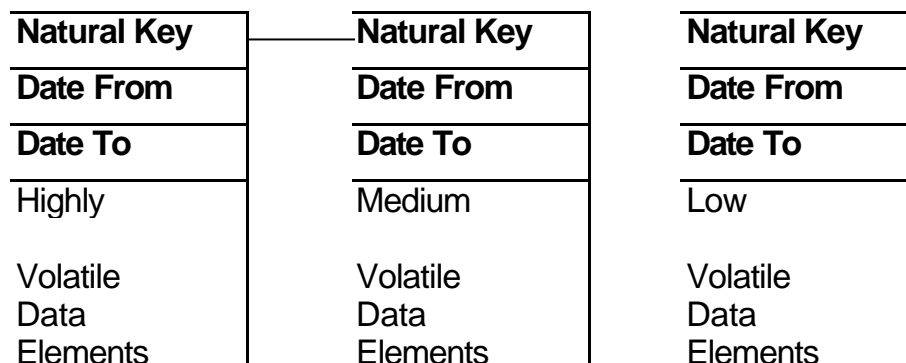
Many Star Schema designs I have seen require that data is unloaded from the fact tables (or maintained on tapes) and sorted/summarised/reloaded every time the Data Warehouse is run. This is a heavy CPU user. The Data Warehouse Generator has been designed to always accept incremental updates using transaction files. Thus to maintain summary information the incoming transactions are consolidated with the data that exists in the fact tables. This dramatically reduces the CPU workload of maintaining summaries.

6. *THIRD NORMAL FORM SUPPORT*

Programs exist that will extract data from an Oracle/DB2 view to simulate the extract from an existing operational system. This data is then passed through a reformat and data transformation program which can perform any operations on the data that is required. Finally, the reformatted data is passed to an update program that performs inserts and updates on the target table. For Third Normal Form tables this is very straight forward code. Note that most of the data transformation is the responsibility of the customer since you will have your own rules for 'cleaning' and 'transforming data'.

7. TIME VARIANCE WITH STABILITY ANALYSIS SUPPORT

Firstly, what does a Time Variant with Stability Analysis model look like? The following diagram is a simple picture of a Time Variant with Stability Analysis model.



Stability Analysis

There exists code that will extract a record from a view, then pass it to a program to perform stability analysis. Stability analysis consists of:

- taking the single input record
- splitting it into three sub-records, one for each of high, medium and low volatile data
- determining if any of the fields in each part of the record have changed since the last time the record was fed into the data warehouse
- for those parts of the record that have changed a record is written to the appropriate output file to be loaded into the Data Warehouse.

Time Variance

When a record that has undergone stability analysis it is passed to the update routine and it is made time variant. That is, it has a Date_From and Date_To placed on the record along with the natural key. This requires going back into the Data Warehouse and updating the previous version of the record with a Date_To to end its effectiveness. In this way you can slice through the database on any given date and you will be given an accurate picture of what the business looked like on that date.

Note that when placing a Date_To on the previous version of the record you have the option of placing the same date as the effective date of the new version of the record, or you can place the effective_date - 1 day as the effective date of the new version. Choosing effective_date - 1 allows you the use of the SQL "between" statement.

8. **GENERAL FEATURES**

This section contains documentation on some of the general features of the Data Warehouse Generator.

Messaging and Audit Trails

All programs write messages to a message table in the Data Warehouse so that they are not lost. All programs issue messages when they start, stop and when they encounter some problem in processing. Messages can be:

- **'I'**nformation : to inform you of some routine event like a program starting
- **'W'**arning : to inform you of some unusual occurrence like a dimension table did not contain an appropriate record when the attribution process was running.
- **'S'**evere : to inform you a major error has occurred. At this point the Data Warehouse Generator program may or may not stop depending on the exact nature of the error. No severe error messages should be ignored.

All messages are time stamped so that operations and support staff can see what has been happening.

Each program produces rows in the Audit Table. Each program records the number of records processed by type of process as well as hash totals of numeric fields to verify accuracy (not actually completed).

9. WHAT TABLES EXIST?

The following section describes the tables that actually exist in the Data Warehouse Generator. Please note that each name actually refers to the view used to access the table in the code, not to the table itself.

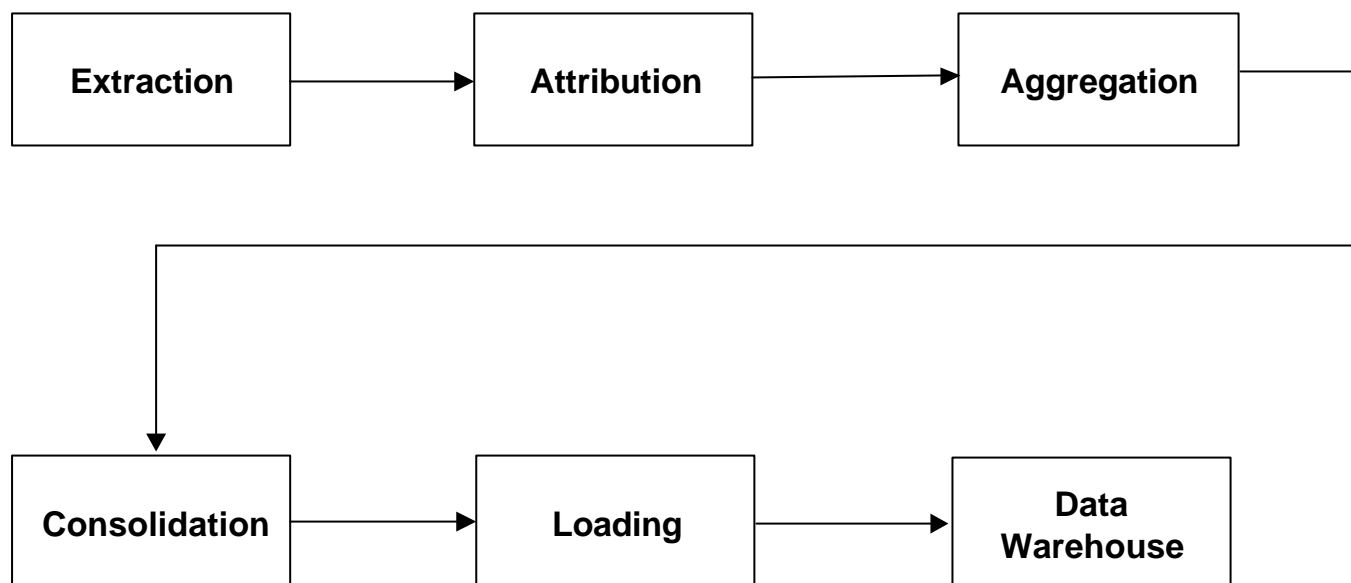
Table	Table Description
DWCUVC01	Control Table View 01. This control table maintains a number of parameters that the Data Warehouse uses to control various processes. In particular it contains the Dimension Table Loading mechanism. Dimension tables can be loaded and searched in 3 different ways. 1) Load the whole table 2) Load 100 records maximum 3) Read the dimension table record directly from the dimension table every time using SQL. This table also provides the option for Date_To management on the Time Variant 3NF data. If this field is set to MINUS1 then the update programs for the Time Variant 3NF data will subtract one day from the Date_To field of the previous record when inserting time variant data.
DWCUVC02	Control Table View 02. This table controls the aggregation levels of various fact tables. It contains records with the fact table name and 10 integers (one for each dimension) specifying the aggregation levels.
DWCUV03	Control Table View 03. This table controls the monthly processing of the data. It contains fields of last_month, this_month and next_month.
DWCUV04	This is the Aggregate Keys Allocation Table. This table maintains the next keys to be allocated to a dimension at all levels. It provides the ability to allocate key ranges for dimensions.
DWCUVCCT	This is the Audit Control Table. Each program writes into this table it's name, the field name that it is auditing, the total number of rows processed and a time stamp of when the record was written. So for example in an update and insert program there will be at least three entries. The number of records read from the input file, the number of records inserted and the number of records updated. Allowance has been made for, and a future release will contain, integer totals and dollar total so that hash totals can be traced through the system
DWDWVMSG	The Messages Table. This table contains all the messages issued by the Data Warehouse Generator.
DWDWVDBG	The Debug Control Table. This table is read by all programs to set the level of debugging that is applied to each program as it runs. This allows the dynamic changing of debugging levels without having to recompile the code. A useful feature when developing code. This table contains the program name and an integer to set the level of debugging.
DWCUVD01	Dimension Table 01. This dimension table is set up as the time dimension table as Time is nearly always the first dimension for any Star Schema.
DWCUVD02 – DWCUVD04	Dimension Tables 02 - 04. These are the other 3 dimension tables. They contain the dimensions by which you want to analyse the Data.
DWCUVFD1	Detail Fact Table 001. This is the detailed level fact table. It contains 4 integer keys and some data.
DWCUVFS1	Summary Fact Table 001. This table stores summaries of the detail fact table. The purpose of having the detail and the summary in physically different tables is that I often store real keys at the back of the detail table which cannot be stored on the summary records.

That is, 11 tables that define how to load a star schema. I do have other code for 3NF and Time Variant data that I hope to publish some time. However, I have seen much less interest in such code.

The experience of years is embedded in these tables, I hope you can make use of it.

10. WHAT PROGRAMS EXIST?

The actual programs that exist, their functions and names are available on request. Currently there are thirty Cobol programs that are available.



Given the 5 stages of data processing that must be performed to place a record into the data warehouse one can reasonably expect at least 5 pieces of code for each star. The programs have naming standards:

DWXXYYNN

DW – Data Warehouse

XX – Name of star being processed

YY – Type of program

NN – Number of that type of program

The initial suite of code that is delivered for a star is as follows:

- DWCURP1 = DW - CUsomer Star – Reformat Program 1
- DWCUAT1 = DW - CUsomer Star – Attribute Program 1
- DWCUAG1 = DW - CUsomer Star – Aggregate Program 1
- DWCUCL1 = DW - CUsomer Star – Consolidate Program 1
- DWCULD1 = DW - CUsomer Star – Load Program 1
- DWCUUP1 = DW - CUsomer Star – Update Program 1

The attribution program calls dimension table lookup programs called

DWCUDL01 – NN - Dimension Lookup 1 through to N

So for an 8 dimension fact table there are 14 pieces of cobol code plus code that is standard function code for writing messages etc.

All this code is generated for each star. This code generation dramatically reduces the time and effort to write the code to support large star schema data warehouses.

The names of the programs and listings in the downloadable zip file are as follows. They are documented in the order they are executed in.

Table	Table Description
DWCURP1	Reformat Program. This is simply a program that will accept a file, reformat it to some other output format. Quite useful as a sample program.
DWCUXR1	eXtRact program. This is a simple program to extract data from a table. Useful as a sample program. Also, it creates the ability to set up a view and then extract data from the view using this extract program.
DWCUAT1	ATtribution program. This is the heart of the star schema generation process. The attribution process takes the real keys passed into the program, performs lookups on the dimension tables, and returns all the lookup keys to the fact record. These keys are then attached to the fact record and written out to the output file for aggregation.
DWCUDL01	Dimension Load program 01. Each dimension table is looked up by using a dimension load and lookup program. You can see that the lookup can use SQL, binary array search or array of 100 elements sequential search. Cobol did not support the allocation of memory at run time hence there was no way to count the number of rows in the dimension table and to load them into the memory of the machine.
DWCUDL02	Dimension Load program 02.
DWCUDL03	Dimension Load program 03.
DWCUDL04	Dimension Load program 04.
DWCUAG1	AGgregation program. This program reads in the attributed transactions and produced a file which is a collection of all the aggregates that are defines in the aggregate control table DWCUVC02. The secret to aggregation was to process the input file as many times as there were summary levels to create moving the aggregate keys forward in the record before the sort/sum.
DWCUCL1	ConsoLidation program. Having produced the set of records that are required for the summary records we then need to determine which summary records already exist and add the incoming records to these records which already exist. This is the process of enabling incremental updates to summary tables. The summary records that do not have pre-existing rows in the target fact table are simply written to a separate output file.
DWCULD1	LoaD program. This is a program that will load the detailed fact records into the detailed fact table. It assumes that each record will be new. However, if a unique index clash occurs it will update the record.
DWCUUP1	UpDate program. This is a program that will update the summary fact table. It uses flag written out by AG1 to determine whether to perform an insert or an update. If the flag proves to be in accurate it will issue an error message.
DWCUDM01	Dimension Maintenance program.
DWCUDM04	Dimension Maintenance program.
DWDWCNTL	CoNTroL record lookup program. This is a program to lookup the control record to determine what processing should be performed in certain circumstances.
DWDWDEBUG	DeBUG program. This program is called at the start of every program to lookup the debugging options for the program. This means that the debugging behavior of any program can be changes without a recompile. Very useful for testing or re-running a production program to determine what happened.
DWDWMSGs	MeSsaGeS program. This is the program that is called to write messages to the message log. At any time a message needs to be issued this routine is called. This is much more effective than printing messages to standard output. The only thing you need to be aware of is that if you are going to issue a rollback you need to issue the rollback and checkpoint prior to issuing the message.
DWLOGON	LOGON program. To access an oracle database the program must provide a userid and password in a logon call. This program performs this logon function for oracle.
DWHU101	This is a very old program that updates a time variant table. I thought I would include it just in case anyone is looking for such a piece of code.

11. **CURRENT LIMITS?**

This section documents the current limitations on the Data Warehouse Generator.

Dimension Tables

One detail level key and nine aggregate keys on all dimension tables including the time dimension tables.

Commit Processing

Currently there are no commit points built into the processing of the update programs. The update programs commit at the end of the program. The update programs either work or they don't work and they are restarted from the beginning.

Number of Fields on 3NF Tables

Most of the 3NF tables have very few fields on them. Usually an integer key, an integer field, a dollar field (15,2) and a character field. They have been implemented this way because it is very simple to add more columns once you already have one of each defined. And if you are reading this you are a very smart person and you don't need us to do simple work for you...:-)