Paper 113-25

Production-Strength Data Mining Solution for Terabyte-Class Relational Data Warehouses

Michael Nichols, John Zhao, John David Campbell
IXL, Inc.

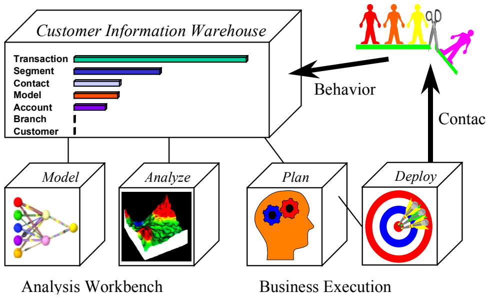
RME Purpose and Processes

Profitability for leading corporations depends on increasing revenue while containing advertising and marketing costs. In an effort to acquire a complete customer profile, many organizations have created large, customer-centric relational databases. After compiling and centralizing customer data, companies are faced with the challenge of translating volumes of customer data into knowledge that can be acted upon. To maintain a competitive edge, marketing analysts need to be able to access and interpret customer information accurately and efficiently.

With Tessera's Rapid Modeling Environment (RME), companies can develop applications quickly so that marketing analysts can access and interpret customer information on an ad-hoc basis. Tessera's RME improves an organization’s overall analytic capabilities such as segmentation, profiling, and scoring. Unlike data mining tools on the market today, Tessera's RME lets analysts perform scheduled production scoring against the full customer base. In addition, analysts can capture generated scores and write them back to the marketing database.

Knowledge-Based Marketing

Increased competition is driving organizations to leverage information gained about their current and prospective customer base so that they can market to them more effectively. Companies perform a complex multi-step process to mine customer information from data warehouses. Marketers use the results from these tools to develop and deploy marketing campaigns. The company tracks the response rates from these campaigns and feeds back this information into the customer database. The company then uses the enhanced customer data to refine its models and develop new marketing campaigns. Using measurable information to develop and hone marketing strategies is the essence of knowledge-based marketing.



Building the Warehouse

Many organizations have invested a lot of time and money developing complex customer information warehouses that contain terabytes of data extracted from their primary operations systems data. Typically, the data includes base customer and account information, transactions, channel usage, promotional information, and external demographics. Data warehousing capabilities have grown tremendously in the past decade to support marketing efforts. Major Relational Database Management Systems (RDBMS) providers have customized their database engines to support Online Analytic Processing (OLAP) operations efficiently. In addition, many new products greatly facilitate the Extraction, Transformation, and Loading (ETL) of legacy system data into a centralized information warehouse. These new technologies have enabled companies to build larger and more complex warehouses, which, in turn has made it harder to effectively extract and mine the information necessary to improve marketing and overall profitability.

Mining the Data

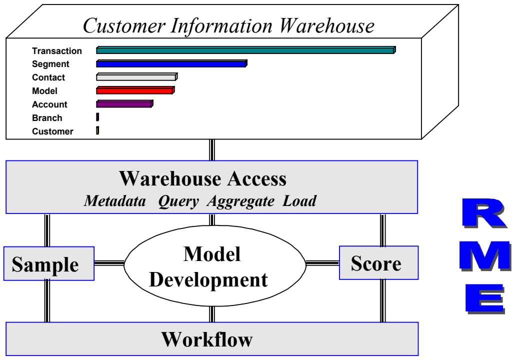
Historically, the development and application of predictive models for customer marketing is a multi-stepped process that takes months. The emergence of data mining packages has improved the value and timeliness of core model development over traditional manual development. Now, companies can apply a variety of sophisticated techniques (CART, CHAID, Neural Networks, Logistic Regression, etc.) to a single set of data to produce differing models whose predictive performance can be projected and compared. Marketers can choose the best model form to predict behavior for the full customer base. However, these mining tools do not improve the acquisition and integration of the rich data sets that drive model development. Most packages expect to start with a single flat file. Those tools that support links to DBMS environments still expect the data to be pre-processed into several tables made up of a single row per customer. This constraint conflicts with the relational database structure of customer information warehouses.

Acting on Marketing Information

Having formulated a predictive model, the challenge is to use the model to predict behavior for the full target population and then use the resulting scores to focus marketing programs. Traditionally, this step has involved time-consuming and error-prone re-coding of the scoring algorithm into a mainframe environment where it is combined with operational applications to compute scores and generate lists of target customers for marketing actions. The new data mining tools do little to change this process because they neither address scheduled production scoring against the full customer base, nor capture generated scores and write them back to the marketing database.

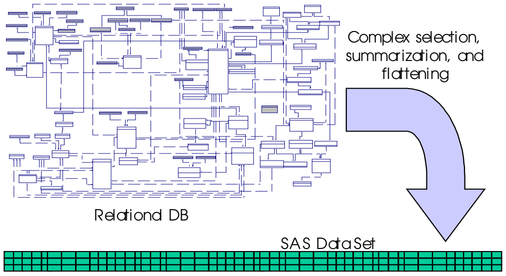
Using Tessera's RME to Build and Execute Models

Without a flexible extraction mechanism to get appropriate data from the warehouse, traditional mining algorithms extract little information of value. Without a production scoring facility, the developed models generate few targets for new marketing activity and provide little return on the marketing investment. Tessera’s RME focuses on these pre- and post-modeling activities to enable the organization to successfully mine customer data and take action on the identified information. Its metadata-driven modules support complex extraction of data from the customer information warehouse, sampling of extracted data sets, registration of developed scoring algorithms, and scheduled application of the production models to the appropriate customer base. In addition, it captures the scores and writes them back to the information warehouse. The modeling organization can choose the tools used in the core model development process.



Extracting Data for Modeling

The modeling process requires diverse data to uncover informational relationships. While traditional customer information warehouses capture the raw data necessary for quantitative analysis, the raw data is in fairly normalized relational structures. However, marketing scientists need wide, de-normalized data structures for analysis. Typically, data is distributed among tens to hundreds of database tables. The creation of a composite customer profile view to integrate this diverse data involves retrieving and aggregating data from many different tables within an RDBMS. The different sets of customer data need to be pre-processed or transformed to prepare them for analysis.



The two major transformations are *aggregation*, where multiple records of data per customer are combined into a single factual metric per customer, and *flattening*, where multiple records per customer are reverse pivoted into multiple columns. In addition, a system must support complex *filtering* criteria to include only the appropriate data in each computation and transformation. When selecting data from different parts of the information warehouse, analysts supply separate SQL statements, which must be merged after all aggregations and flattening, to produce a single analysis data set.

Sophisticated analysts apply these three types of operations to data throughout the database to retrieve and transform it into the appropriate data set for exploratory analysis. The RME provides a graphical user interface (GUI) wherein users can develop applications that support aggregation, flattening, and complex filtering. The RME Extraction module uses information about the database schema to guide and manage a user's selections where appropriate.

Selection and Aggregation

Marketing scientists select data fields from the information warehouse to produce analysis data sets with one record per customer. Any selected field from a table with multiple rows of data per customer requires an aggregation operator to reduce the data to a single value per customer. This information is merged with data from other tables to produce a singe composite row per customer. The RME Extraction interface uses its metadata to manage this relationship and automatically forces the specification of aggregation operators when appropriate. The RME supports many standard aggregation operators based on SQL column operators such as MIN, MAX, SUM, and AVG. For example, a common aggregation is the result of a query of transaction detail data that sums the number and amount of a certain type of transaction across all accounts for a customer. Another common aggregation takes the maximum value of the transaction dates per customer to produce a “recency” metric.

Dimensional Flattening

Dimensional flattening is an extension of aggregation to produce multiple metrics per customer differentiated by a dimensional entity such as time periods, transaction codes, or product categories. For example, summing the transaction amounts per customer per month for the last three months produces multiple monthly result rows per customer that are transposed into a fixed set of three monthly amount sums per customer.

Sequential Flattening

When original source values are desired without aggregation of multiple source rows for each target identifier, it is necessary to transpose a number of the source rows into one row of multiple sets of columns. For example, on a per customer basis, the order dates, IDs and amounts could be retrieved from the order table for the first two, last five and biggest three orders. These extractions, sorting, truncations and transposing are denoted as sequence flattening.

Filtering

Typically, filtering is the constraint of rows from a source table for data extraction, aggregation, and flattening. Filtering can be simple such as “where transaction amount is greater than 5,000” or “where transaction code is D." Filters may be more complex, such as “where the transaction amount is greater than 5,000 and the transaction code is D.” In addition, filtering constraints may be derived from a link to another table. For example, the filtering constraint may be “where the transaction occurred at a store in Massachusetts.” This requires a link between the transaction source table and the store source table, where each table has the store identifier.

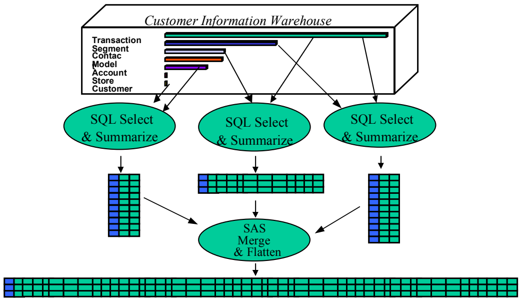
Merging

The retrieval of aggregated fields from several tables must be conducted independently from each other. This multiple SQL approach avoids unintended Cartesian products in the results that will corrupt calculated values. The aggregation of result sets to the customer level enables all result sets to be merged on the common customer identifier. The merged data set is the consolidated data set that the analyst specified.

The RME Extraction Module

The RME Extraction engine provides a GUI to guide the user through the selection of fields, flattening, filtering, and merging. The selections are saved in an extract specification file on the RME Client (as defined by the RME architecture below), and can be used later or as a starting point for a different extraction problem. When the analyst is satisfied with the specification, the RME Extraction Module moves the specification data to the RME Server (as defined by the RME architecture below), where it generates and executes the appropriate, multiple SQL statements to retrieve the data from the information warehouse. The RME Server generates and executes SAS code to transform, flatten, and merge the results into the desired analysis data set in a SAS format on the RME Server platform.

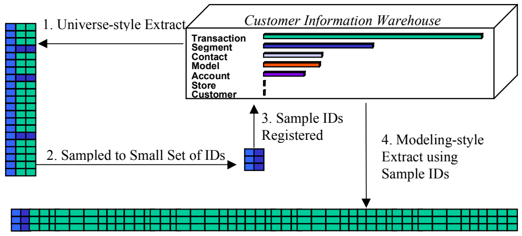
This functionality enables analysts to quickly define and create data sets with hundreds of output fields or variables that address the marketing question at hand. The speed of specification and generation encourages an iterative approach to data acquisition to ensure that the right data from the information warehouse goes into the modeling analysis.



Sampling

The analysis of customer data to detect trends and relationships that are useful for predicting customer behavior requires only a fraction of the customer base. Hence, a common approach to model development involves defining a target universe and a statistically appropriate sample. Analysts create a complex set of analysis variables only for this sample set of customers. Upon completion of the modeling development process, the analysts apply the resulting model to the full target universe for production scoring.

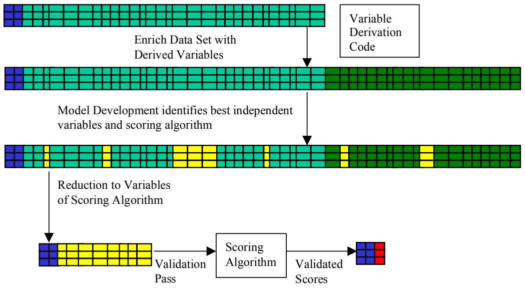
Tessera's RME Sampling Module works with the Extraction Module to facilitate this sampling process. It also enables the defined sample sets to be preserved and shared within the Information Warehouse. Using the Extraction Module, a data set is created with a small number of variables and many rows. This data set (the "universe-style extract") reflects the entire target population. The Sampling Module provides GUI elements to guide the user through defining and executing either a simple or stratified sampling of the universe data set. The Sampling Module also supports registering the set of sample customer identifiers. The registration process loads the set of sample customer identifiers and metadata about the sample (such as the sampling method) back into the Information Warehouse. Using the registered sample of customers as filtering criteria, the analysts specify a rich set of variables for the sample population. The resulting data set allows analysts to begin exploratory data analysis and model development. This data set is called the modeling-style extract.


1. Universe-style Extract
Customer Information Warehouse
Transaction
Segment
Contact
Model
Account
Store
Customer
3. Sample IDs Registered
2. Sampled to Small Set of IDs
4. Modeling-style Extract using Sample IDs

Developing Models

After selecting the model-style extract, marketing analysts determine the modeling technique and tools to develop the analytical model. SAS-based statistical analysis and other data mining techniques provide the modeling techniques and tools. The set of raw variables in the model-style extract is supplemented with derived variables including dummy variables and linearized transformations of other variables. Various techniques are applied to produce a model and an associated scoring algorithm. This model and scoring algorithm provide a "good" prediction of customer behavior.

The scoring algorithm is encoded using a number of raw and derived variables and a set of coefficients or weights as determined by the training data set. The algorithm is validated against a set of records and then judged acceptable.

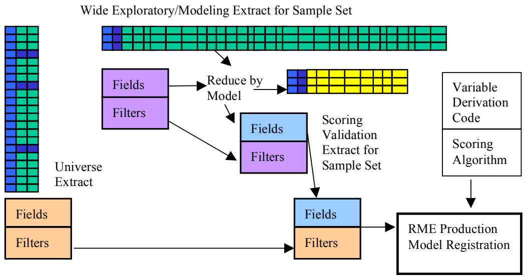

Variable Derivation Code
Enrich Data Set with Derived Variables
Model Development identifies best independent variables and scoring algorithm
Reduction to Variables of Scoring Algorithm
Validation Pass
Scoring Algorithm
Validated Scores

Customer Scoring

When marketing analysts use the developed models to generate scores and create better targets for marketing actions, the company begins to realize the return on its marketing investments. Tessera's RME provides functions to enable model management, scheduled production score generation, and administration of the generated scores.

Model Registration

When the model-scoring algorithm has been validated and the reduced extract variable list has been defined, analysts modify filtering criteria for the reduced extract specification to encompass the original target universe. The production extract specification and the scoring algorithm code are registered through the RME Administration Module. Analysts can insert date indicators in the place of absolute dates where appropriate in the extract specification to allow periodic execution of the model without re-coding the dates each time. To complete the production scoring specification, analysts specify how the scores are quantiled (deciling, percentiling, etc.) prior to loading them into the customer information warehouse.

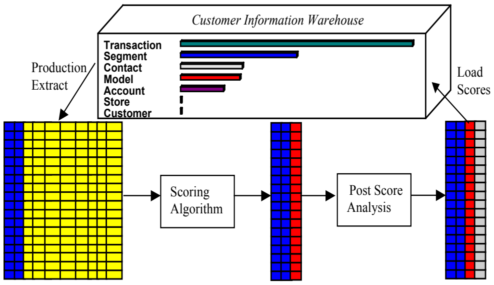


Scheduling

Analysts can configure the RME Administration Module to provide mechanisms for both immediate and deferred score computation for registered models. The RME Administration Module supports the selection and editing of the deferred processing queue on the RME Server.

Production Scoring

When a production scoring process is initiated or started through the queue, the process is executed entirely on the RME Server and the DBMS Server hosting the Information Warehouse. The production extract specification is retrieved from the registered model directory and the relative dates are evaluated for the current date of the information warehouse. The specification drives the generation and execution of the production extract and produces a result set of the variables necessary for the scoring algorithm of the particular target universe of customers. The extracted SAS data set is then passed to the registered scoring algorithm code resulting in a SAS data set of scores. The scores data is quantiled and normalized and then loaded back into the Information Warehouse. The system produces a production scoring summary log file.

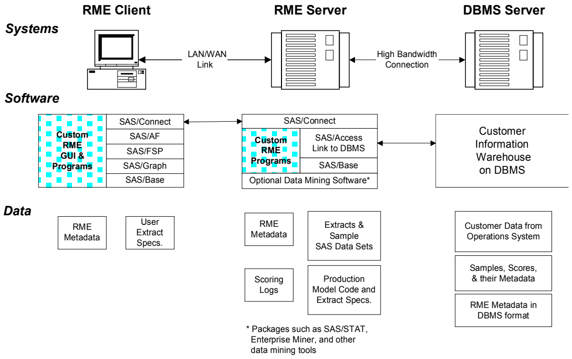


Score Administration

The execution of production scoring runs generates increasing volumes of score data per customer. The RME Administrator Module maintains summary information about which scores have been generated historically and which are still active in the scores portion of the Customer Information Warehouse. In addition, this module supports the deletion or archiving of obsolete scores from the scores tables.

The RME System Architecture

The RME is a three-tiered distributed application that is layered on an existing Customer Information Warehouse hosted on an RDBMS. The user interacts with the RME Client application, which operates on Windows PCs using SAS core software. Processing functions are executed on the RME Server component, which is generally on a UNIX server platform that may or may not be distinct from the RDBMS platform. The RME Client communicates with the RME Server using TCP/IP and TELNET protocols and functions remotely over a WAN as well as through LAN access. The RME Server must be connected to the DBMS Server through a fairly high bandwidth connection to support the movement of the extracted SQL results from the DBMS environment into the SAS environment and to support the loading of samples and scores generated in the SAS environment into the DBMS.


RME Client
RME Server
DBMS Server
Systems
LAN/WAN Link
High Bandwidth Connection
Software
Custom RME GUI & Programs
SAS/Connect
SAS/AF
SAS/FSP
SAS/Graph
SAS/Base
SAS/Connect
Custom RME Programs
SAS/Access Link to DBMS
SAS/Base
Optional Data Mining Software*
Customer Information Warehouse on DBMS
Data
RME Metadata
User Extract Specs.
RME Metadata
Extracts & Sample SAS Data Sets
Scoring Logs
Production Model Code and Extract Specs.
Customer Data from Operations System
Samples, Scores, & their Metadata
RME Metadata in DBMS format
* Packages such as SAS/STAT, Enterprise Miner, and other data mining tools

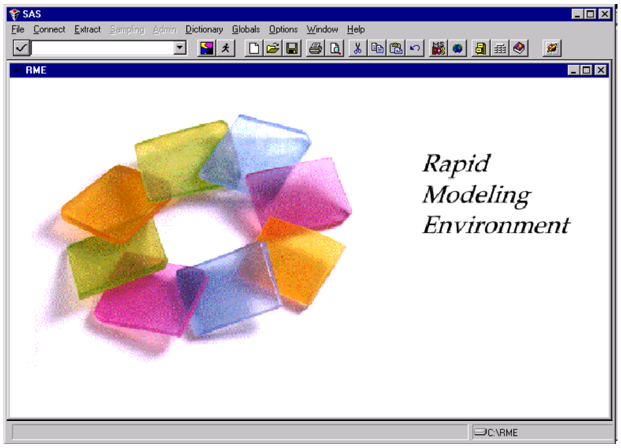
Metadata

The RME modules all use a series of metadata data sets in SAS to control GUI presentation, SQL and SAS code generation, and production scoring. This metadata is based on the physical schema of the Customer Information Warehouse database and the business relationships of the data within the database. Populating this metadata is a principal activity in integrating the RME for an organization. These metadata data sets are managed on the RME Server and are automatically distributed to the RME Client PCs when the client first connects after any update. The local cache of metadata on the RME Client enables users to formulate and edit extract specifications without an active connection to the RME Server. When the connection is established, the user can use the previously defined extract specification to generate and execute code and to create the extract file that will reside on the RME Server.

RME Tool Elements and Operation

The RME is a SAS-based multi-tiered analysis framework. The SAS AF-based GUI is an extension of the analyst's SAS desktop where RME functions have been attached through main menu extensions for each set of module:

- Connect – for signon/signoff to server and related options
- Extract – for specifying and executing single table queries and multi-table/multi-SQL extracts from the database
- Sampling – Nth, Random, Numerical and Categorical Stratified Random and Registration of Sample IDs back into database
- Admin – for listing and managing sample ID sets, model scores, modeling extracts and programs, and scheduling scoring runs
- Dictionary – for presenting data dictionary of tables and fields of database available through RME.

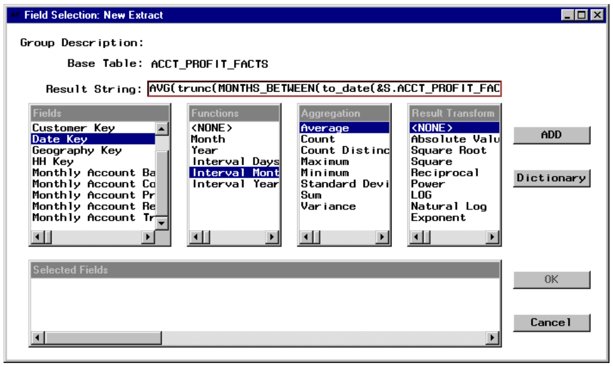


Rapid Modeling Environment

Extraction Overview

Multi-table Extracts are targeted to a specific output data set resolution -- one observation per household, individual, account, store, state, product, etc. Extract specifications are composed of one or more variable groups each defining a separate SQL. Each generated SQL may join multiple tables to define a set of output fields or transformed and aggregated field expressions that have a particular filtering scope and flattening strategy. Each variable group results in a PROC SQL clause that creates a temporary data set of SQL results. These results are then transformed, if appropriate, by generated SAS code to transpose or flatten multiple rows to a single row per target entity identifier (HH_ID, INDV_ID, ST_CD, etc.). The multiple flattened intermediate result sets are then merged on the common target entity key. The merge of the intermediate results are supported with Boolean AND, OR, and EXCEPT connectors and parentheses.

Variable group definitions begin with the selection of a base table from which to draw data and a decision on whether to aggregate rows when the base table is below the granularity of the target entity. Then field expressions are selected that which include a rich set of source field transformations, aggregations and aggregate transforms.



Field Selection: New Extract

Group Description:

Base Table: ACCT_PROFIT_FACTS

Result String: AVG(trunc(MONTHS_BETWEEN(to_date(&S.ACCT_PROFIT_FAC

Fields: Customer Key, Date Key, Geography Key, HH Key, Monthly Account Ba, Monthly Account Co, Monthly Account Pr, Monthly Account Re, Monthly Account Tr

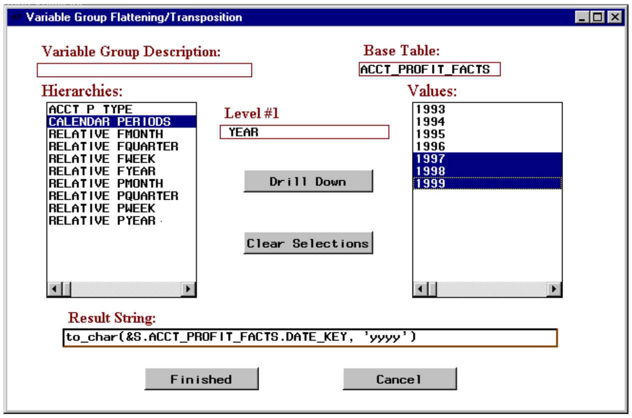
Functions: <NONE>, Month, Year, Interval Days, Interval Mont, Interval Year

Aggregation: Average, Count, Count Distinc, Maximum, Minimum, Standard Devi, Sum, Variance

Result Transform: <NONE>, Absolute Valu, Square Root, Square, Reciprocal, Power, LOG, Natural Log, Exponent

Selected Fields

Dimensional flattening supports the selection values from dimension hierarchies to control grouping operations for the aggregation operators. For example, sums of order amounts per customer could be flattened by year, month, product category, purchase channel, payment method, etc.



Variable Group Flattening/Transposition

Variable Group Description:

Base Table: ACCT_PROFIT_FACTS

Hierarchies: ACCT P TYPE, CALENDAR PERIODS, RELATIVE FMONTH, RELATIVE FQUARTER, RELATIVE FWEEK, RELATIVE FYEAR, RELATIVE PMONTH, RELATIVE PQUARTER, RELATIVE PWEEK, RELATIVE PYEAR

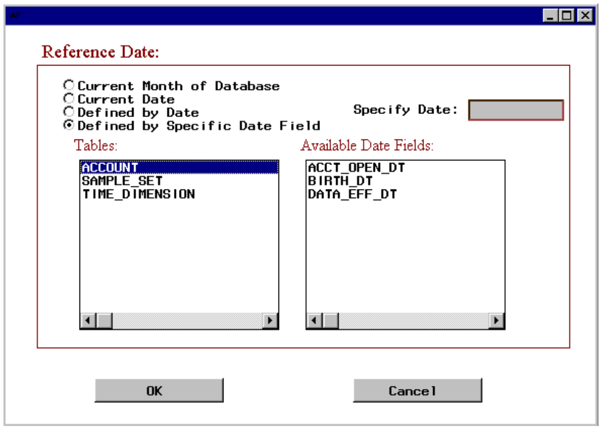
Level #1: YEAR

Values: 1993, 1994, 1995, 1996, 1997, 1998, 1999

Result String: to_char(&S.ACCT_PROFIT_FACTS.DATE_KEY, 'yyyy')

The dimensional flattening can be driven by dimensional tables in the data model in the fashion of a star schema, or the dimensional values can be equation-driven using a field in a fact tables and RME metadata entries to hold the formulas. This approach enables the RME to support relative time dimensions where a date field such as order date drives the grouping of order records and aggregates of other order fields such as sum of order amount into time periods relative to a user-defined time reference.

The selection of a time reference will be from available date fields. Selecting a time reference such as first purchase date would enable order amount sums to be computed per customer per month for the first *n* months of each customer's purchase history. This temporally aligns all customers' history to a common set of initial time periods. This mechanism equally supports days, quarters, or years; it also supports relative periods prior to or after a defined reference date.



Reference Date:

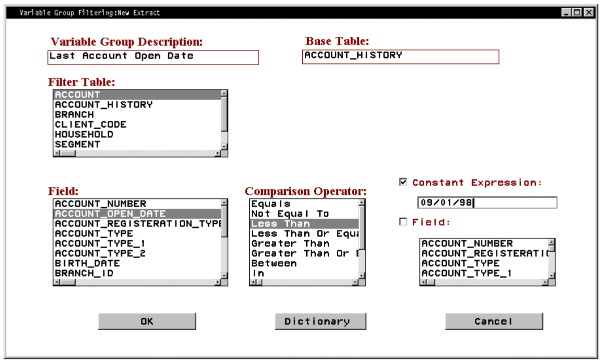
○ Current Month of Database
○ Current Date
○ Defined by Date
● Defined by Specific Date Field

Specify Date:

Tables: ACCOUNT, SAMPLE_SET, TIME_DIMENSION

Available Date Fields: ACCT_OPEN_DT, BIRTH_DT, DATA_EFF_DT

Filtering of source rows is supported for a rich set of comparison operators and allows criteria to join to many other tables in the database. A similar structure supports results filtering on aggregated field expressions through generated SQL “having clauses”.



Variable Group Filtering:New Extract

Variable Group Description: Last Account Open Date

Base Table: ACCOUNT_HISTORY

Filter Table: ACCOUNT, ACCOUNT_HISTORY, BRANCH, CLIENT_CODE, HOUSEHOLD, SEGMENT

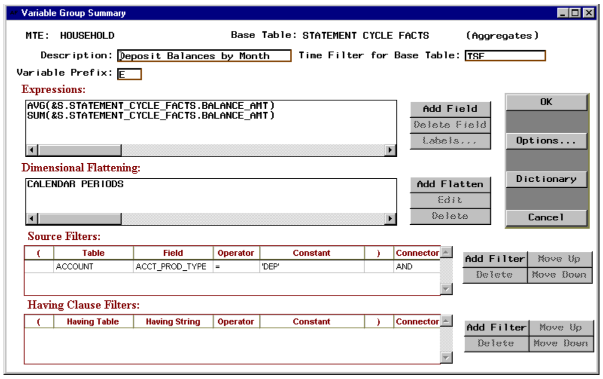
Field: ACCOUNT_NUMBER, ACCOUNT_OPEN_DATE, ACCOUNT_REGISTERATION_TYPE, ACCOUNT_TYPE, ACCOUNT_TYPE_1, ACCOUNT_TYPE_2, BIRTH_DATE, BRANCH_ID

Comparison Operator: Equals, Not Equal To, Less Than, Less Than Or Equa, Greater Than, Greater Than Or E, Between, In

☑ Constant Expression: 09/01/98

☐ Field: ACCOUNT_NUMBER, ACCOUNT_REGISTERATI, ACCOUNT_TYPE, ACCOUNT_TYPE_1

Once field expressions, flattening, and filtering specifications have been entered, a variable group is fully defined. Other variable groups can be defined using the same base table, similar fields, flattening, and filters to provide the desired data from the database.



Variable Group Summary

MTE: HOUSEHOLD Base Table: STATEMENT CYCLE FACTS (Aggregates)

Description: Deposit Balances by Month Time Filter for Base Table: TSF

Variable Prefix: E

Expressions:

AVG(&S.STATEMENT_CYCLE_FACTS.BALANCE_AMT)
SUM(&S.STATEMENT_CYCLE_FACTS.BALANCE_AMT)

Dimensional Flattening:

CALENDAR PERIODS

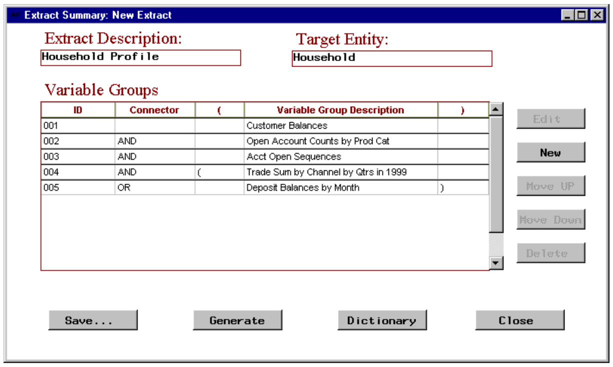
Source Filters:

(	Table	Field	Operator	Constant	)	Connector
	ACCOUNT	ACCT_PROD_TYPE	=	'DEP'		AND

Having Clause Filters:

(	Having Table	Having String	Operator	Constant	)	Connector

Once the specification and integration of variable groups are defined, then the extract request is submitted to the RME Server.



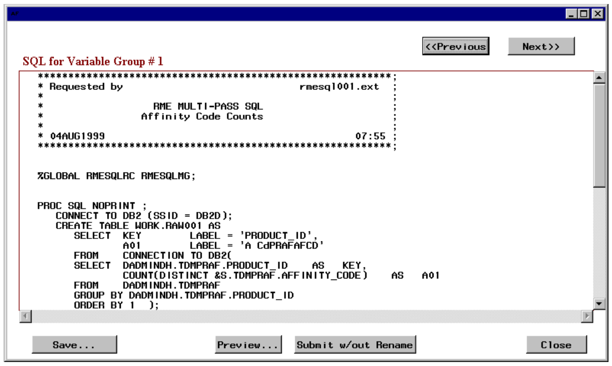
Extract Description: Household Profile

Target Entity: Household

Variable Groups

ID	Connector	(	Variable Group Description	)
001			Customer Balances	
002	AND		Open Account Counts by Prod Cat	
003	AND		Acct Open Sequences	
004	AND	(	Trade Sum by Channel by Qtrs in 1999	
005	OR		Deposit Balances by Month	)

The specifications are saved locally as a structured text file for subsequent editing or exchange with other analysts. When the extract is to be executed, the SQL elements are first generated. The specification is passed from the RME client to the server, parsed into different temporary SAS data sets and then the data sets are manipulated by complex SAS macros to generate SQL fragments. The macros heavily utilize SAS SQL to manipulate the user selections and RME metadata to create the proper fragments. These character strings are eventually assembled into one SQL statement per variable group. Each SQL is wrapped within a SAS PROC SQL construct that creates a temporary SAS table to hold the intermediate results from the DBMS. The generated PROC SQL code is downloaded to the RME client and displayed to the user for validation.



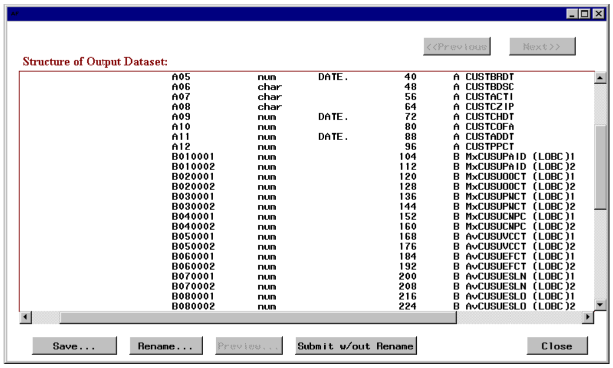
SQL for Variable Group # 1

```
**********************************************************;
* Requested by                              rmesql001.ext  ;
*                                                          ;
*                     RME MULTI-PASS SQL                   ;
*                   Affinity Code Counts                   ;
*                                                          ;
* 04AUG1999                                         07:55  ;
**********************************************************;


%GLOBAL RMESQLRC RMESQLMG;


PROC SQL NOPRINT ;
   CONNECT TO DB2 (SSID = DB2D);
   CREATE TABLE WORK.RAW001 AS
      SELECT  KEY         LABEL = 'PRODUCT_ID',
              A01         LABEL = 'A CdPRAFAFCD'
      FROM    CONNECTION TO DB2(
      SELECT  DADMINDH.TDMPRAF.PRODUCT_ID    AS   KEY,
              COUNT(DISTINCT &S.TDMPRAF.AFFINITY_CODE)    AS   A01
      FROM    DADMINDH.TDMPRAF
      GROUP BY DADMINDH.TDMPRAF.PRODUCT_ID
      ORDER BY 1   );
```

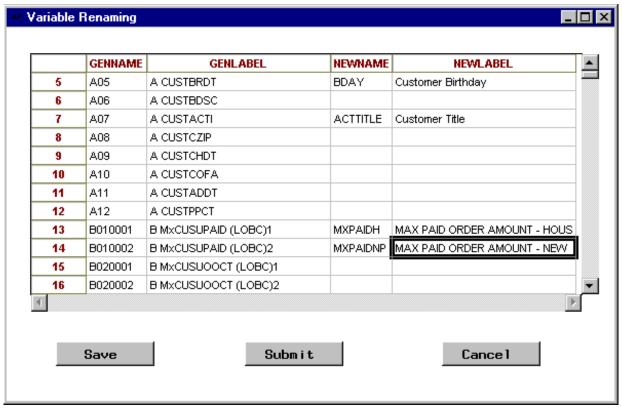
A preview option causes a simplified version of the macros for flattening and merging to process the generated metadata to construct a no-observation data set that will identically match the structure of the result data set that would be generated using this extract specification. This enables the user to confirm that the desired output variables are included in the single consolidated result data structure.



Structure of Output Dataset:

```
                 A05        num      DATE.      40     A CUSTBRDT
                 A06        char                48     A CUSTBDSC
                 A07        char                56     A CUSTACTI
                 A08        char                64     A CUSTCZIP
                 A09        num      DATE.      72     A CUSTCHDT
                 A10        num                 80     A CUSTCOFA
                 A11        num      DATE.      88     A CUSTADDT
                 A12        num                 96     A CUSTPPCT
                 B010001    num                104     B MxCUSUPAID (LOBC)1
                 B010002    num                112     B MxCUSUPAID (LOBC)2
                 B020001    num                120     B MxCUSUOOCT (LOBC)1
                 B020002    num                128     B MxCUSUOOCT (LOBC)2
                 B030001    num                136     B MxCUSUPNCT (LOBC)1
                 B030002    num                144     B MxCUSUPNCT (LOBC)2
                 B040001    num                152     B MxCUSUCNPC (LOBC)1
                 B040002    num                160     B MxCUSUCNPC (LOBC)2
                 B050001    num                168     B AvCUSUVCCT (LOBC)1
                 B050002    num                176     B AvCUSUVCCT (LOBC)2
                 B060001    num                184     B AvCUSUEFCT (LOBC)1
                 B060002    num                192     B AvCUSUEFCT (LOBC)2
                 B070001    num                200     B AvCUSUESLN (LOBC)1
                 B070002    num                208     B AvCUSUESLN (LOBC)2
                 B080001    num                216     B AvCUSUESLO (LOBC)1
                 B080002    num                224     B AvCUSUESLO (LOBC)2
```

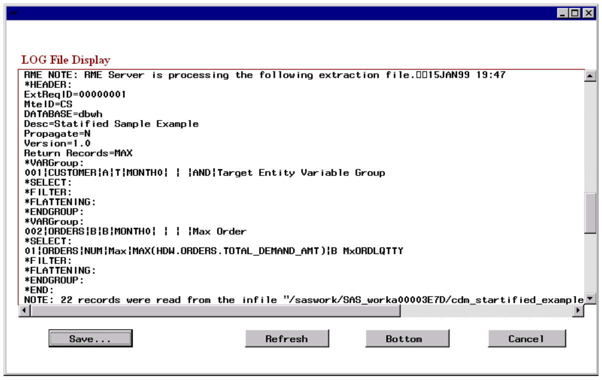
From here the user can optionally define remapping from the generated variables' names to a corporate or individual set of standard variable names and labels. This variable renaming is applied after SQL extraction, flattening and merging, immediately before the consolidated data is delivered to the user's data library.



	GENNAME	GENLABEL	NEWNAME	NEWLABEL
5	A05	A CUSTBRDT	BDAY	Customer Birthday
6	A06	A CUSTBDSC		
7	A07	A CUSTACTI	ACTTITLE	Customer Title
8	A08	A CUSTCZIP		
9	A09	A CUSTCHDT		
10	A10	A CUSTCOFA		
11	A11	A CUSTADDT		
12	A12	A CUSTPPCT		
13	B010001	B MxCUSUPAID (LOBC)1	MXPAIDH	MAX PAID ORDER AMOUNT - HOUS
14	B010002	B MxCUSUPAID (LOBC)2	MXPAIDNP	MAX PAID ORDER AMOUNT - NEW
15	B020001	B MxCUSUOOCT (LOBC)1		
16	B020002	B MxCUSUOOCT (LOBC)2		

At this point the user can elect to submit the extract specification to the RME server for execution. The SQL is regenerated in a different location and the post-SQL macros are invoked. These macros perform the appropriate dimensional or sequence flattening transpose operations per intermediate data set and finally the complex merge of the respective intermediate result sets. The process ends with a single data set delivered to the user's SAS data library, which can be tens to thousands of variables wide and tens to millions of observations deep.

The users can monitor the execution of the spawned extraction process through the log view window, which presents a list of completed and active extraction process SAS logs. Selecting the desired log from the list results in the download of the log file to the PC for display, avoiding the need for the user to run an explicit terminal session on the RME server to track job progress. The log presents the submitted extract specification text, the generated SQLs, the times and sizes for the intermediate results data sets, and the name and size of the persistent, consolidated output data set.



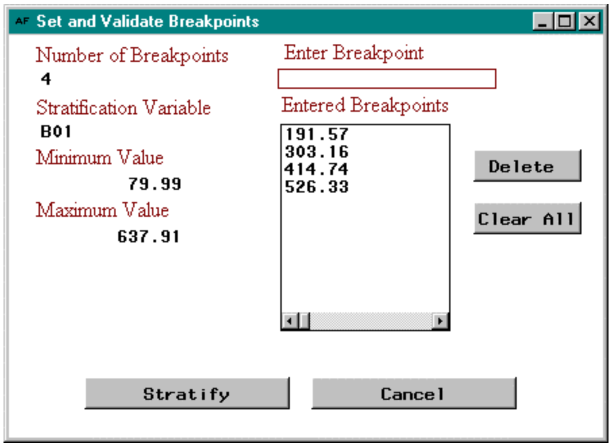
LOG File Display

```
RME NOTE: RME Server is processing the following extraction file.□□15JAN99 19:47
*HEADER:
ExtReqID=00000001
MteID=CS
DATABASE=dbwh
Desc=Statified Sample Example
Propagate=N
Version=1.0
Return Records=MAX
*VARGroup:
001¦CUSTOMER¦A¦T¦MONTH0¦ ¦ ¦AND¦Target Entity Variable Group
*SELECT:
*FILTER:
*FLATTENING:
*ENDGROUP:
*VARGroup:
002¦ORDERS¦B¦B¦MONTH0¦ ¦ ¦ ¦Max Order
*SELECT:
01¦ORDERS¦NUM¦Max¦MAX(HDW.ORDERS.TOTAL_DEMAND_AMT)¦B MxORDLQTTY
*FILTER:
*FLATTENING:
*ENDGROUP:
*END:
NOTE: 22 records were read from the infile "/saswork/SAS_worka00003E7D/cdm_startified_example
```

Stratification, Sampling and Sample Registration

To support analysis of millions of customers and thousands of variables, sampling must be employed, as discussed earlier. The extracted SAS data sets can be sampled using user-defined custom SAS code, but the RME provides a set of GUI screens and SAS code to support most standard sampling needs. The RME manages the registration of the IDs of the sample set and their load back into the DBMS where they are used to filter criteria in variable groups to create wide rich extracts for data mining analysis.

The RME supports simple Nth and simple uniform random sampling of extract data set as well as more sophisticated numerical and categorical stratification with random sampling within strata. Numerical stratification is driven by the user's selection definition of the number of desired strata and of an appropriate continuous variable, and then the assignment of breakpoint of the variable's values to define the strata. The breakpoint can be initialized with system-defined values and then edited to support the user's understanding of the control variable's distribution.



Number of Breakpoints: 4

Stratification Variable: B01

Minimum Value: 79.99

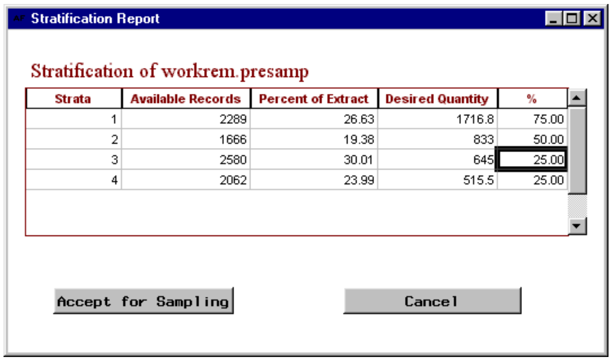
Maximum Value: 637.91

Enter Breakpoint:

Entered Breakpoints:

191.57
303.16
414.74
526.33

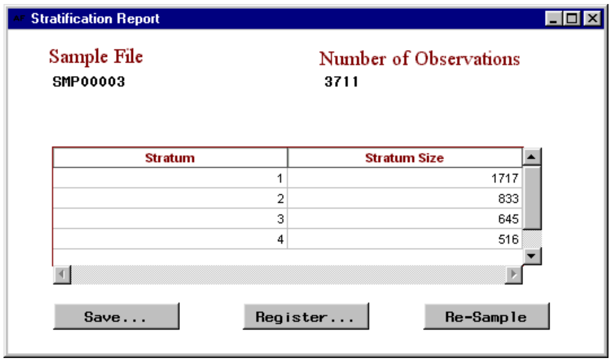
Once stratified, the user selects the number of samples per strata as either a percent or absolute number of observations. All defined strata do not have to be sampled and a complete stratum can be included in the sample data set.



Stratification of workrem.presamp

Strata	Available Records	Percent of Extract	Desired Quantity	%
1	2289	26.63	1716.8	75.00
2	1666	19.38	833	50.00
3	2580	30.01	645	25.00
4	2062	23.99	515.5	25.00

After sampling, a summary is displayed and the user is allowed to iterate back through prior steps until the desired stratified sample is achieved.

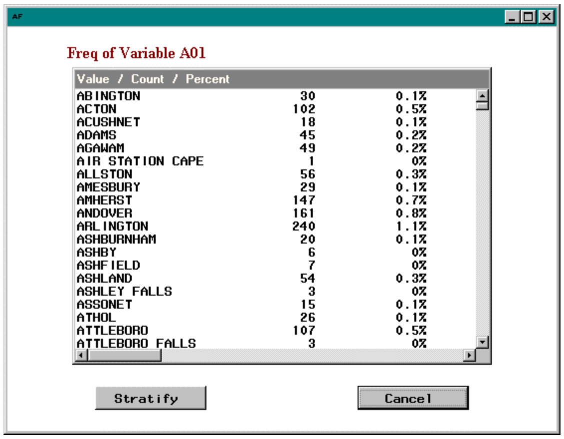


Sample File: SMP00003

Number of Observations: 3711

Stratum	Stratum Size
1	1717
2	833
3	645
4	516

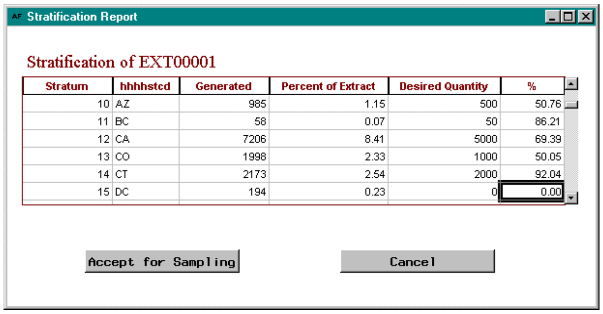
Similarly, a categorical variable can be used to define stratification. In this case, the number of distinct values of the selected control variable defines the number of levels. A frequency distribution is computed and displayed to allow the user to confirm the appropriateness of the control variable.



Freq of Variable A01

Value / Count / Percent		
ABINGTON	30	0.1%
ACTON	102	0.5%
ACUSHNET	18	0.1%
ADAMS	45	0.2%
AGAWAM	49	0.2%
AIR STATION CAPE	1	0%
ALLSTON	56	0.3%
AMESBURY	29	0.1%
AMHERST	147	0.7%
ANDOVER	161	0.8%
ARLINGTON	240	1.1%
ASHBURNHAM	20	0.1%
ASHBY	6	0%
ASHFIELD	7	0%
ASHLAND	54	0.3%
ASHLEY FALLS	3	0%
ASSONET	15	0.1%
ATHOL	26	0.1%
ATTLEBORO	107	0.5%
ATTLEBORO FALLS	3	0%

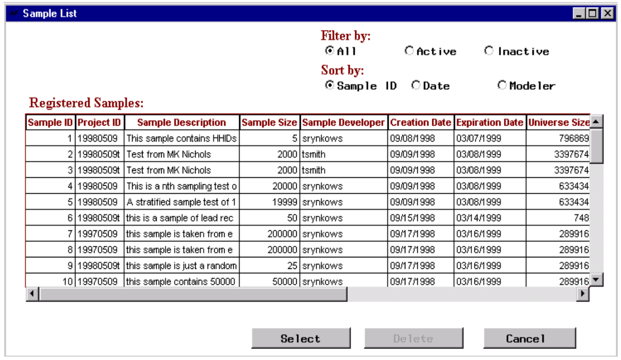
Once a control variable is accepted, the data set is stratified and then the user again determines the level of sampling for each stratum. Here the categorical variable value is also displayed per stratum.



Stratification of EXT00001

Stratum	hhhhstcd	Generated	Percent of Extract	Desired Quantity	%
10	AZ	985	1.15	500	50.76
11	BC	58	0.07	50	86.21
12	CA	7206	8.41	5000	69.39
13	CO	1998	2.33	1000	50.05
14	CT	2173	2.54	2000	92.04
15	DC	194	0.23	0	0.00

The modeling target entity IDs from the sample set and the strata indicator can be saved and submitted to the RME administrator for load into the appropriate sample table in the DBMS. This makes the pool of sample IDs available for use by the submitting analyst or the whole analysis organization on continuing basis. The RME user community can use the Admin options to list the sample ID sets currently active in the samples' tables on the database.



Filter by: ● All ○ Active ○ Inactive

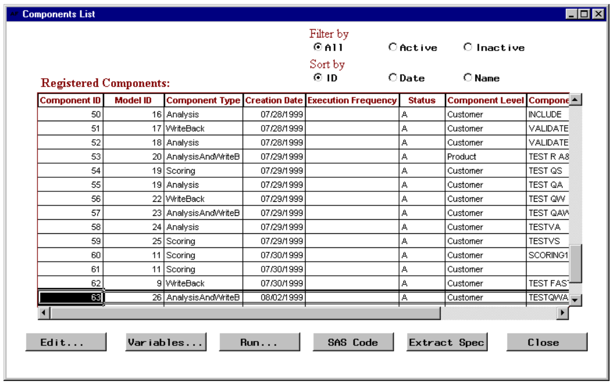
Sort by: ● Sample ID ○ Date ○ Modeler

Registered Samples:

Sample ID	Project ID	Sample Description	Sample Size	Sample Developer	Creation Date	Expiration Date	Universe Size
1	19980509	This sample contains HHIDs	5	srynkows	09/08/1998	03/07/1999	796869
2	19980509t	Test from MK Nichols	2000	tsmith	09/09/1998	03/08/1999	3397674
3	19980509t	Test from MK Nichols	2000	tsmith	09/09/1998	03/08/1999	3397674
4	19980509	This is a nth sampling test o	20000	srynkows	09/09/1998	03/08/1999	633434
5	19980509	A stratified sample test of 1	19999	srynkows	09/09/1998	03/08/1999	633434
6	19980509t	this is a sample of lead rec	50	srynkows	09/15/1998	03/14/1999	748
7	19970509	this sample is taken from e	200000	srynkows	09/17/1998	03/16/1999	289916
8	19970509	this sample is taken from e	200000	srynkows	09/17/1998	03/16/1999	289916
9	19980509t	this sample is just a random	25	srynkows	09/17/1998	03/16/1999	289916
10	19970509	this sample contains 50000	50000	srynkows	09/17/1998	03/16/1999	289916

Scoring and Analysis Components

Combining a scoring universe extract specification with a scoring code produces a scoring component that is registered in the Admin section and supports scheduled execution with load-back of quantiled scores to the database. Other components can also be registered including an analysis component that runs a provided extract specification followed by user-defined SAS analysis or reporting code. A third type of component is termed a write-back component because it runs an extract and then writes back the target entity IDs from the extract into the database to a special temporary ID table. The contents of the temporary ID table would be typically used as filtering criteria in a scoring or analysis component. The final type of component is an analysis with write back that executes an extract, runs the user-defined analysis program and then writes back the IDs output from the analysis program.



Components List

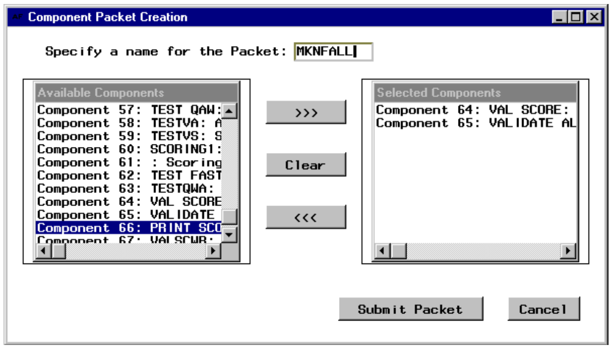
Filter by ● All ○ Active ○ Inactive

Sort by ● ID ○ Date ○ Name

Registered Components:

Component ID	Model ID	Component Type	Creation Date	Execution Frequency	Status	Component Level	Compone
50	16	Analysis	07/28/1999		A	Customer	INCLUDE
51	17	WriteBack	07/28/1999		A	Customer	VALIDATE
52	18	Analysis	07/28/1999		A	Customer	VALIDATE
53	20	AnalysisAndWriteB	07/29/1999		A	Product	TEST R A&
54	19	Scoring	07/29/1999		A	Customer	TEST QS
55	19	Analysis	07/29/1999		A	Customer	TEST QA
56	22	WriteBack	07/29/1999		A	Customer	TEST QW
57	23	AnalysisAndWriteB	07/29/1999		A	Customer	TEST QAW
58	24	Analysis	07/29/1999		A	Customer	TESTVA
59	25	Scoring	07/29/1999		A	Customer	TESTVS
60	11	Scoring	07/30/1999		A	Customer	SCORING1
61	11	Scoring	07/30/1999		A	Customer	
62	9	WriteBack	07/30/1999		A	Customer	TEST FAS
63	26	AnalysisAndWriteB	08/02/1999		A	Customer	TESTQWA

These components are registered by the RME Administrator and can then be executed individually or combined in a multi-step packet of components. The RME Administrator can execute components or packets on an immediate or on a scheduled basis.



Component Packet Creation

Specify a name for the Packet: MKNFALL

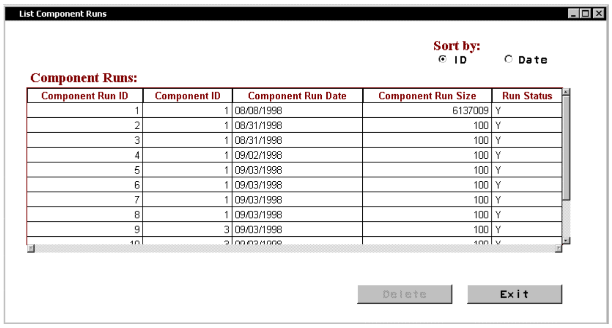
Available Components

```
Component 57: TEST QAW:
Component 58: TESTVA: A
Component 59: TESTVS: S
Component 60: SCORING1:
Component 61: : Scoring
Component 62: TEST FAST
Component 63: TESTQWA:
Component 64: VAL SCORE
Component 65: VALIDATE
Component 66: PRINT SCO
Component 67: VALSCUB:
```

Selected Components

```
Component 64: VAL SCORE:
Component 65: VALIDATE AL
```

Scoring components write scores to the database and the RME logs these new scores and makes then visible to users through a Component Run screen. These scores are then generally used in subsequent campaign selection processes and tools.



List Component Runs

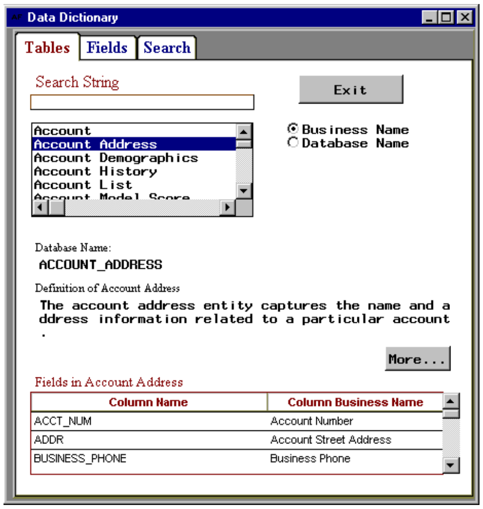
Sort by: ● ID ○ Date

Component Runs:

Component Run ID	Component ID	Component Run Date	Component Run Size	Run Status
1	1	08/08/1998	6137009	Y
2	1	08/31/1998	100	Y
3	1	08/31/1998	100	Y
4	1	09/02/1998	100	Y
5	1	09/03/1998	100	Y
6	1	09/03/1998	100	Y
7	1	09/03/1998	100	Y
8	1	09/03/1998	100	Y
9	3	09/03/1998	100	Y
10	3	09/03/1998	100	Y

Data Dictionary

A component of the RME metadata is detailed information about the database tables and fields presented to the users through the RME. The Tables panel displays the descriptive attributes of the selected table and the list of fields.



Data Dictionary

Search String

```
Account
Account Address
Account Demographics
Account History
Account List
Account Model Score
```

● Business Name ○ Database Name

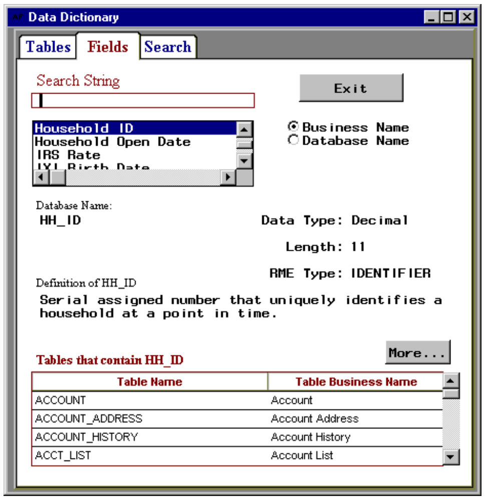
Database Name:
ACCOUNT_ADDRESS

Definition of Account Address
The account address entity captures the name and a ddress information related to a particular account .

Fields in Account Address

Column Name	Column Business Name
ACCT_NUM	Account Number
ADDR	Account Street Address
BUSINESS_PHONE	Business Phone

Selecting a specific field name displays the descriptive attributes of the selected field and the list of all tables containing a field of the same name.



Data Dictionary

Search String

```
Household ID
Household Open Date
IRS Rate
IXI Birth Date
```

● Business Name ○ Database Name

Database Name:
HH_ID

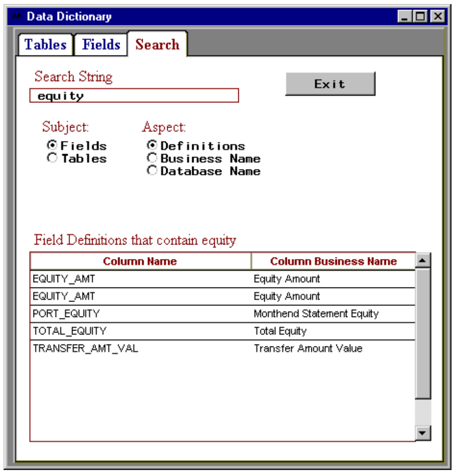
Data Type: Decimal
Length: 11
RME Type: IDENTIFIER

Definition of HH_ID
Serial assigned number that uniquely identifies a household at a point in time.

Tables that contain HH_ID

Table Name	Table Business Name
ACCOUNT	Account
ACCOUNT_ADDRESS	Account Address
ACCOUNT_HISTORY	Account History
ACCT_LIST	Account List

A generic search of database names, business names or definitions aids significantly in initial database discovery and understanding for new users. Selecting and returned table or field row jump to the respective panel of table or field details.



Data Dictionary

Search String: equity

Subject: ● Fields ○ Tables

Aspect: ● Definitions ○ Business Name ○ Database Name

Field Definitions that contain equity

Column Name	Column Business Name
EQUITY_AMT	Equity Amount
EQUITY_AMT	Equity Amount
PORT_EQUITY	Monthend Statement Equity
TOTAL_EQUITY	Total Equity
TRANSFER_AMT_VAL	Transfer Amount Value

Summary

The selection of a SAS-based infrastructure has enabled the RME to be quickly delivered to the marketing analysis departments of Fortune 500 companies. A core of SAS application code operates against a variety of DBMS engines under a variety of server operating systems, including a spectrum of UNIX systems, MVS-OS/390, and Windows NT. SAS Connect has enabled the RME Client to communicate effectively with servers across the country over standard telephone lines and modems, enabling a geographically dispersed group of analysts to share a central computing resource and database. The RME enables business analysts and modelers to access and leverage DBMS data warehouses in the powerful and familiar SAS environment.

Without Tessera's RME, complex iterative data mining is tedious, time-consuming, and costly. Tessera's RME creates an environment for knowledge-based marketing. The RME facilitates acquiring richer data to mine from the corporate and marketing data warehouses. The RME automates the mundane steps modelers are forced to perform. This automation reduces development time and lets users focus their energies on creating complex models and performing advanced analysis quickly and accurately.

Contact Information

IXL, Inc.
7 Audubon Road
Wakefield, MA 01880
(781) 245-9024
www.ixl.com

Michael K. Nichols
(781) 716-1048
mnichols@ixl.com

John Zhao
(781) 716-1029
jzhao@ixl.com

John David Campbell
(781) 716-1081
jcampbell@ixl.com