**Predicts - Ad Match**

This site documents Predicts Project Ad Match component

Predicts project Ad Match component is aimed at enhancing sales productivity by recommending properties for customers. When a sales person is engaged with a given customer, the system suggests properties that the customer has not yet advertised on, but might be of interest to the customer. The system analyzes customer's advertisement history; matches it with other customers' advertisement history; finds other similar advertisers and then finds out which properties they have advertised on and might be useful to the given advertiser.

Predicts uses Apache Mahout libraries for generating recommendations. It takes data from various sources such as YCRM, UAD. Its User Interface is based on Ace Framework.

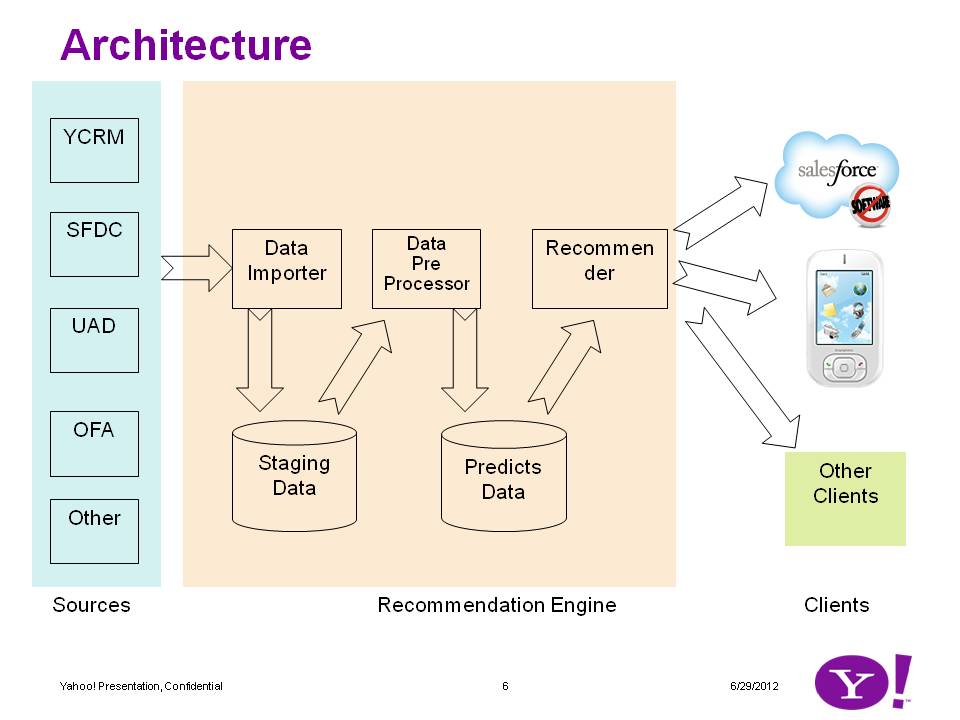
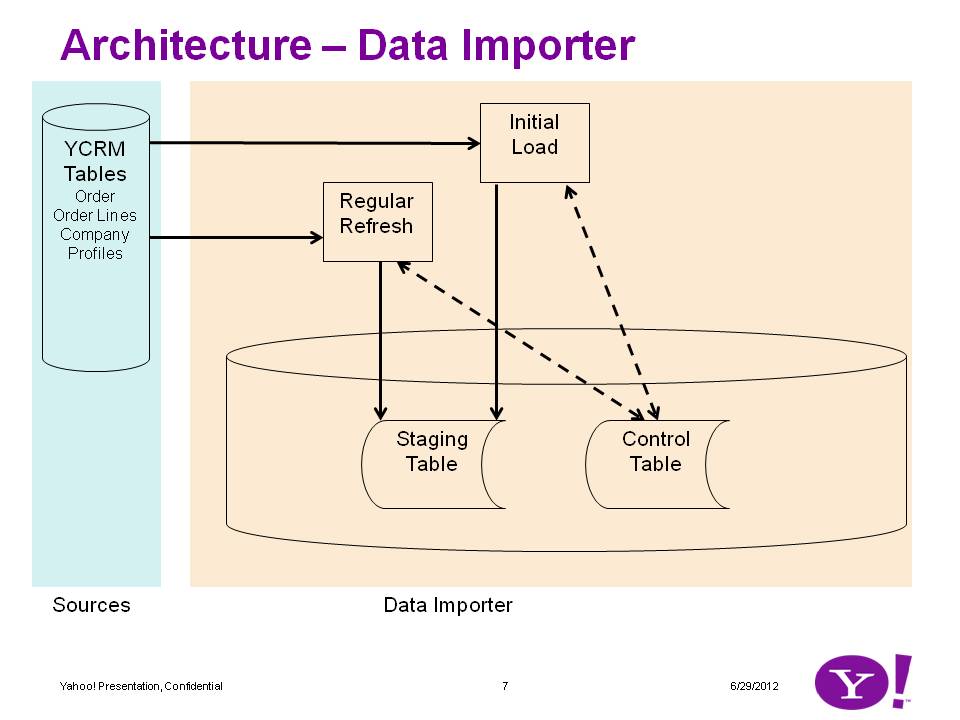
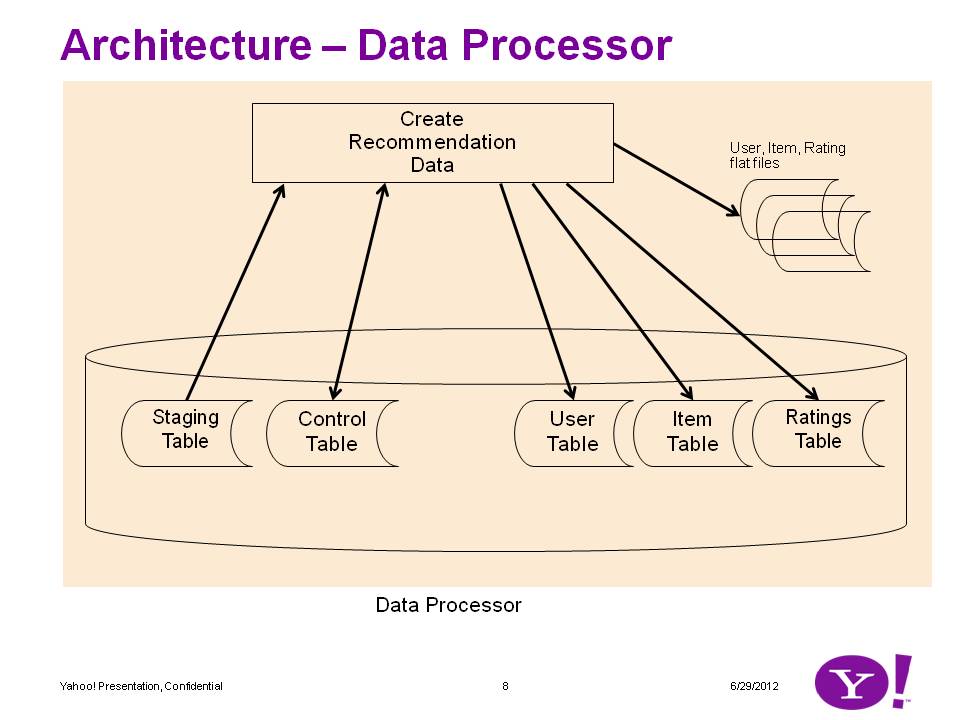
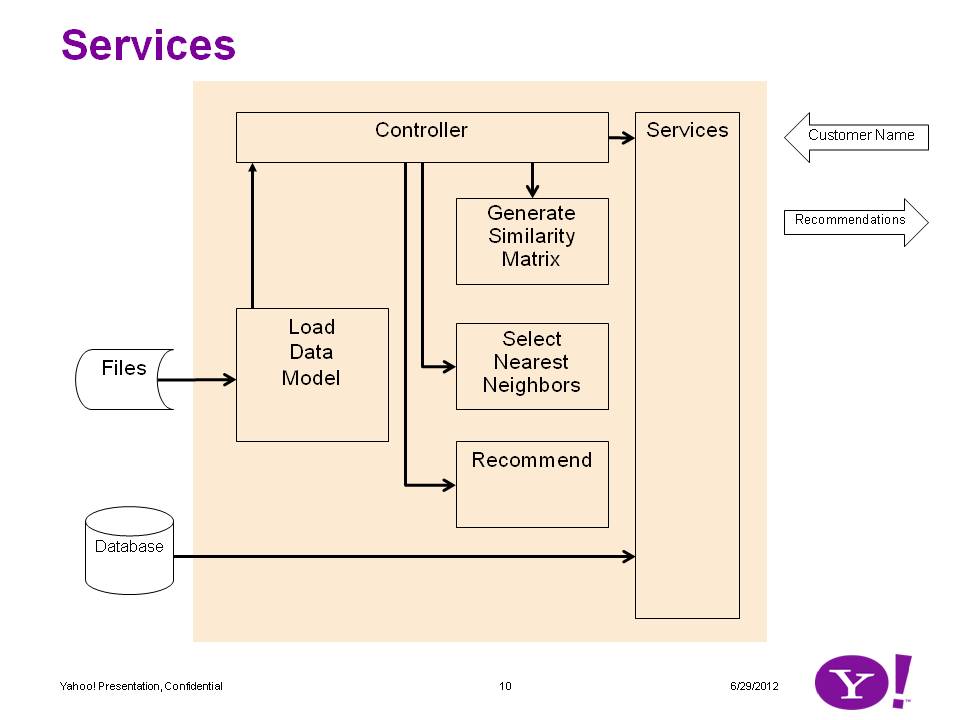
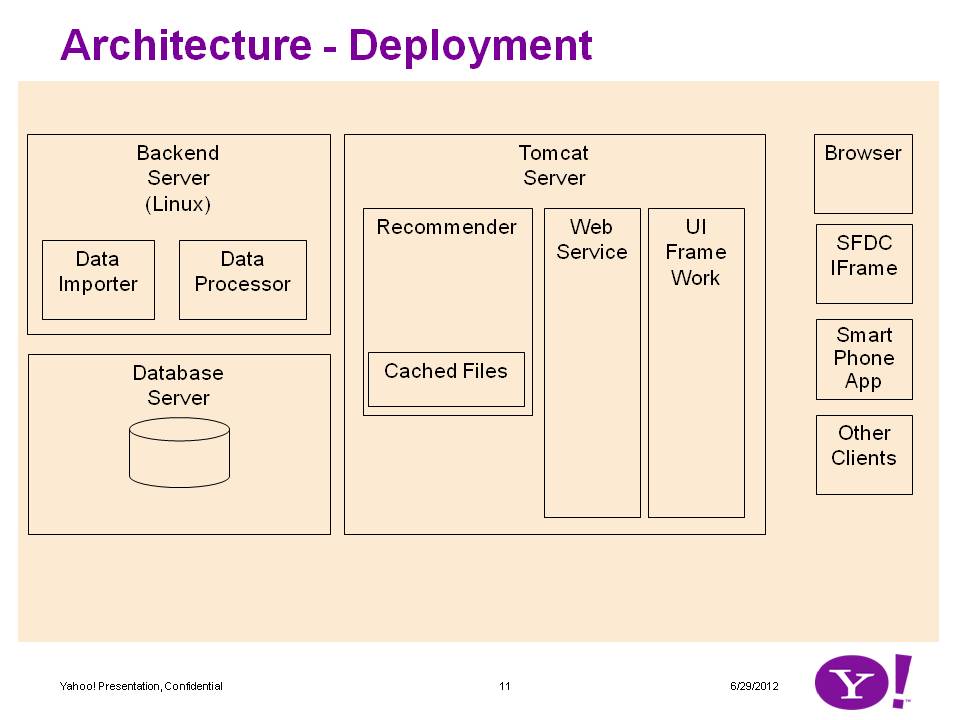
User Interface uses services provided by Predicts Services module. The services module provides services such as 'get similar companies', 'get recommendations', 'get company details', 'get company list' (for auto complete) etc. Services module has been implemented using Yahoo Web Services which runs on Spring framework and returns JSON result in REST based web services.

Please click on following links to get more information on overall architecture

* [Architecture](http://sp2-crmarch-001.ads.pool.corp.sp2.yahoo.com:4080/predictsdoc/predictsArchitecture.html)
* [Data Model](http://sp2-crmarch-001.ads.pool.corp.sp2.yahoo.com:4080/predictsdoc/predictsDataModel.html)
* [Back End Processing](http://sp2-crmarch-001.ads.pool.corp.sp2.yahoo.com:4080/predictsdoc/predictsBackEndProcessing.html)
* [Web Services](http://sp2-crmarch-001.ads.pool.corp.sp2.yahoo.com:4080/predictsdoc/predictsWebServices.html)
* [Home](http://sp2-crmarch-001.ads.pool.corp.sp2.yahoo.com:4080/predictsdoc/index.html)
* This page documents Predicts Project architecture.

### Introduction

### Architecture

* Predicts architecture is divided into three layers: User Interface, Services, Data. User Interface layer is concerned about rendering the recommendations in an engaging manner and interacting with services layer to get and set data. Services layer uses data layer to read and write data that User Interface layer needs.   
  
* The image describes the three main components. Currently only YCRM source is ready. We hope that UAD source will be ready in the week of 7/1/12.
* There are two sets of tables in Predicts. First set of tables is the staging tables. These tables serve as a staging platform for incoming data. This helps isolate the second set of tables. Second set of tables consists of predicts tables. These tables contain data required by the algorithm.
* There are several components that process the data to make it usable by recommender engine. A component Data Importer imports data from the source (such as YCRM) and stores the data into Staging table. Second component Data Processor process this data and populates predicts tables. Recommender uses predicts tables to generate recommendation. Note: Apache Mahout library as of now does not work with database. Therefore we have built an interface through flat files.   
  
* Data Importer has two roles: Initial load, regular refresh.
* Initial load queries the source for data starting from a certain date in the past and gets data until system date. It updates a control table with the last import date.
* Regular refresh queries last processing date from the control table and then uses that as a start date to pull data from source table. It updates the control table with the last import date.
* As of 7/3/12 the importer works with YCRM. As soon as UAD schema becomes available we will build an importer to pull data from that source.   
  
* Data processor pulls data from the staging tables and writes data to predicts tables. As noted above, Predicts involves three essential tables. User table contains data about customers. Item table contains data about properties. Rating table contains data about which customer purchased which property. The source for these is a flat staging table. The data procerror reads the staging table and then populates the three tables.
* User table - Processor queries if a customer with the same name and address exists. If it doesn't exist, it creates the user. It matches the customer's name and address to ensure that if there are duplicate records they can be taken care of. YCRM contains some customers whose name and address is exact same but for some reason there are two or more records. The data processor recognizes this and inserts only one customer.
* Item table - Processor queries if an item with same propPath exists. If there is none, it enters a record.
* Rating table - Processor queries the table to see if for same customer and item a record exists. If yes, then it increments aggregateSum column and updates the rating record. If not then it adds a record to rating.
* All the above queries include Project Name. For YCRM data the name is PREDICTSYCRM. This allows us to isolate data from various sources into their own 'buckets'. When UAD data becomes avaialble then it will be under a project PREDICTSUAD.
* The processor uses control table to record when it processed last.
* NOTE: There is another program that reads data from predicts tables and generates data files. As noted above currently the database libraries of Apache Mahout are not working. Therefore current version of Predicts Recommender works from flat files.   
  
* There are two tpes of services: algorithm based services and database based services.
* Algorithm based services include recommendation, getting similar companies, getting similar items. These services work with a recommendation infrastructure. The infrastructure consists of a controller which loads the user, item and rating data into memory. As a request comes, the controller generates similarity matrix and generates similar companies (neighborhood) and then based on that generates recommendation.
* Database based services include getting company details, getting item details, getting a list of companies based on input string etc. These services go directly to the database to do these operations.   
  
* Deployment architecture diagram shows various subsystems. They need not be on separate computer.
* Backend Server contains the data importer, data processor and the program that generates flat files. Currently these are run manually. But in steady state these will be scheduled using cron.
* Database server is a regular Oracle RDBMS.
* Tomcat server contains the recommender classes. It runs off of flat files that are stored with the package. Web services layer calls recommender for algorithm based services and database for database based services. UI framework uses services.
* Clients use UI framework. Currently we are using only browsers such as IE, Chrome, Firefox. But in future the UI can be embedded into various other clients. Since the interface is through web services (JSON based) there is no special deployment needed at the client.
* [Home](http://sp2-crmarch-001.ads.pool.corp.sp2.yahoo.com:4080/predictsdoc/index.html)
* This site documents Predicts Project data schema.   
  [Predicts Tables](http://sp2-crmarch-001.ads.pool.corp.sp2.yahoo.com:4080/predictsdoc/predictsDataModel.html#predictsTables)   
  [YCRM Staging Table](http://sp2-crmarch-001.ads.pool.corp.sp2.yahoo.com:4080/predictsdoc/predictsDataModel.html#YCRMStagingTable)   
  [UAD Staging Tables](http://sp2-crmarch-001.ads.pool.corp.sp2.yahoo.com:4080/predictsdoc/predictsDataModel.html#UADStagingTables)

### Predicts Tables

* [Back to Top](http://sp2-crmarch-001.ads.pool.corp.sp2.yahoo.com:4080/predictsdoc/predictsDataModel.html#top)

### Predicts Tables

* Predicts tables are used by the algorithm and the database based services. There are three tables.
* User - Table that contains customer data
* Item - Table that contains data about properties which are purchased by customers
* Rating - Table that contains the association of customer to properties
* describe customer
* Name Null Type
* -------------- -------- -------------
* ID NOT NULL NUMBER(15)
* CUSTOMERNAME VARCHAR2(100) - Name
* CUSTOMERID VARCHAR2(50) - Unique id such as Siebel Row Id
* ADDRSTREET1 VARCHAR2(200)
* ADDRSTREET2 VARCHAR2(100)
* ADDRSTREET3 VARCHAR2(100)
* ADDRCITY VARCHAR2(50)
* ADDRSTATE VARCHAR2(10)
* ADDRPROVINCE VARCHAR2(50)
* ADDRCOUNTRY VARCHAR2(30)
* ADDRPOSTALCODE VARCHAR2(30)
* CATEGORY VARCHAR2(50)
* PARENTCATEGORY VARCHAR2(30)
* DESCRIPTION VARCHAR2(400)
* PROJECT VARCHAR2(100) - Used to collect a set of data such as YCRM, UAD
* CREATEDON DATE
* CREATEDBY VARCHAR2(30)
* DELETEDFLAG VARCHAR2(1)
* CUSTOMERSTATUS VARCHAR2(20)
* describe item
* Name Null Type
* ---------------- -------- -------------
* ID NOT NULL NUMBER(15)
* TYPE VARCHAR2(100) - PROPERTY, TARGETING etc.
* ITEMNAME VARCHAR2(255) - Property Path (not URL)
* POSITION VARCHAR2(255) - LREC, N, SKY etc
* SPACEID VARCHAR2(255) - Numeric Space Id
* ROOTPROPERTYNAME VARCHAR2(255) - Top property such as Sport, Finance, OMG etc
* CPM VARCHAR2(255) - Not used in 1.0
* COMSCORE VARCHAR2(255) - Not available in YCRM
* CLICKTHROUGHRATE VARCHAR2(255) - Not available in YCRM
* DESCRIPTION VARCHAR2(400)
* PROJECT VARCHAR2(100) - Used to collect a set of data such as YCRM, UAD
* CREATEDON TIMESTAMP(6)
* CREATEDBY VARCHAR2(30)
* DELETEDFLAG VARCHAR2(1)
* ITEMSTATUS VARCHAR2(20)
* describe rating
* Name Null Type
* ------------ -------- -------------
* ID NOT NULL NUMBER(15)
* USERID NUMBER(15) - Customer Id
* ITEMID NUMBER(15) - Property Id
* RATINGTYPE VARCHAR2(20) - If BOOL, it has 1.0 otherwise it can vary
* RATING NUMBER(15,2) - Value of rating
* AGGREGATESUM NUMBER(15,2) - Total times a customer bought this property
* PROJECT VARCHAR2(100) - Used to collect a set of data such as YCRM, UAD
* CREATEDON DATE
* CREATEDBY VARCHAR2(30)
* DELETEDFLAG VARCHAR2(1)
* RATINGSTATUS VARCHAR2(20)

### Key Concepts

* Project - This is a top level container which will be used to collect various data sets.
* CPM and Position - These are placement level fields. They don't make sense at property level. Although they were included in item table for predicts, they will not be used in version 1 and may be factored out.
* Root Property Name - This is the top level property such as Sports, Finance, OMG! etc. In YCRM there is a field X\_PROPERTY which has some good values but not all the values are good. We are hoping that when UAD data becomes available the quality of this field will become better.

### YCRM Staging Table

* [Back to the Top](http://sp2-crmarch-001.ads.pool.corp.sp2.yahoo.com:4080/predictsdoc/predictsDataModel.html#top)
* This table contains customer and property information taken from YCRM.
* Some columns are not used now but may be used in future. Some columns were included for debugging purpose. These are not actively used in recommendation but help investigate if something needs investigation.
* describe stagingycrm
* Name Null Type
* -------------------- -------- --------------
* LINEID NOT NULL VARCHAR2(15)
* LINETYPE VARCHAR2(10) - Debug field
* ORDERNUMBER VARCHAR2(30) - Debug field
* CUSTOMERID VARCHAR2(15)
* CUSTOMER VARCHAR2(100)
* ADDRSTREET1 VARCHAR2(200)
* ADDRSTREET2 VARCHAR2(100)
* ADDRSTREET3 VARCHAR2(100)
* ADDRCITY VARCHAR2(50)
* ADDRSTATE VARCHAR2(10)
* ADDRPROVINCE VARCHAR2(50)
* ADDRCOUNTRY VARCHAR2(30)
* ADDRPOSTALCODE VARCHAR2(30)
* CATEGORY VARCHAR2(50)
* PARENTCATEGORY VARCHAR2(30)
* MARKET VARCHAR2(30) - Future use
* RUBYLINEID VARCHAR2(15) - Debug field
* APTLINEID VARCHAR2(30) - Debug field
* PROPPATH VARCHAR2(255)
* POSITION VARCHAR2(255) - Future use
* SPACEID VARCHAR2(255)
* ROOTPROPERTYNAME VARCHAR2(255)
* CPM VARCHAR2(255) - Future use
* ORDERSTATUS VARCHAR2(30)
* MEDIANETAMOUNT NUMBER(22,7) - Future use
* MEDIAGROSSAMOUNT NUMBER(22,7) - Future use
* AUDIENCETARGETING VARCHAR2(2000) - Future use
* CONTENTTARGETING VARCHAR2(2000) - Future use
* ADATTRIBUTETARGETING VARCHAR2(2000) - Future use
* FREQUENCYTARGETING VARCHAR2(2000) - Future use
* RUBYTARGETING VARCHAR2(2000) - Future use
* LINECREATEDON DATE
* LINEUPDATEDON DATE
* CREATEDON TIMESTAMP(6)
* CREATEDBY VARCHAR2(30)
* PROCESSEDON DATE - Once line is processed it is stamped so line is not processed again
* PROCESSEDBY VARCHAR2(30) - Future use
* PROCESSEDSTATUS VARCHAR2(20) - Future use
* PROCESSMESSAGE VARCHAR2(200) - Future use
* ARCHIVEDFLAG VARCHAR2(1) - N for active records
* DELETEDFLAG VARCHAR2(20) - N for active records
* DATASOURCE VARCHAR2(20) - Tells if data came from YCRM, UAD etc
* ENVIRONMENT VARCHAR2(20) - Tells if data came from production, QA, etc
* PROJECT VARCHAR2(100) - Collects all the data into one bucket PREDICTSYCRM holds YCRM sourced data

### UAD Staging Tables

* [Back to the top](http://sp2-crmarch-001.ads.pool.corp.sp2.yahoo.com:4080/predictsdoc/predictsDataModel.html#top)
* desc fact\_comcore
* Name Null Type
* -------- ---- ------
* SPACE\_ID NUMBER
* COMSCORE NUMBER
* desc fact\_revn\_book\_traf
* Name Null Type
* -------------------------- ---- ------
* SPACE\_ID NUMBER
* TRAF\_DLVR\_IMPRESSIONS\_PRNT NUMBER
* TRAF\_DLVR\_CLICKS\_PRNT NUMBER
* TRAF\_DLVR\_CONVERSIONS\_PRNT NUMBER
* TRAF\_DLVR\_IMPRESSIONS\_CHLD NUMBER
* TRAF\_DLVR\_CLICKS\_CHLD NUMBER
* TRAF\_DLVR\_CONVERSIONS\_CHLD NUMBER
* TRAF\_DLVR\_IMPRESSIONS NUMBER
* TRAF\_DLVR\_CLICKS NUMBER
* TRAF\_DLVR\_CONVERSIONS NUMBER
* TRAF\_DLVR\_CLICKTROUGH\_PRNT NUMBER
* TRAF\_DLVR\_CLICKTROUGH\_CHLD NUMBER
* TRAF\_DLVR\_CLICKTROUGH NUMBER
* desc lkp\_advertiser
* Name Null Type
* ------------------ -------- ------------------
* ADVERTISER\_ID NOT NULL NUMBER
* CRM\_ADVERTISER\_ID NOT NULL VARCHAR2(15 CHAR)
* ADVERTISER\_NAME VARCHAR2(300 CHAR)
* ADV\_PRNT\_CATEGORY VARCHAR2(30 CHAR)
* ADV\_ADDR\_LINE\_1 VARCHAR2(300 CHAR)
* ADV\_ADDR\_LINE\_2 VARCHAR2(300 CHAR)
* ADV\_ADDR\_LINE\_3 VARCHAR2(300 CHAR)
* ADV\_ADDR\_CITY VARCHAR2(255 CHAR)
* ADV\_ADDR\_STATE VARCHAR2(90 CHAR)
* ADV\_ADDR\_ZIPCODE VARCHAR2(30 CHAR)
* ADV\_ADDR\_COUNTRY VARCHAR2(30 CHAR)
* ADV\_CURRENCY\_ID NUMBER
* ADV\_CURRENCY\_CODE VARCHAR2(3 CHAR)
* ADV\_TIME\_ZONE\_ID NUMBER
* ADV\_TIME\_ZONE\_NAME VARCHAR2(64 CHAR)
* ADV\_ACTIVE\_FLAG CHAR(1 CHAR)
* SRC\_CREATED\_DATE DATE
* SRC\_UPDATED\_DATE DATE
* CREATED\_DATE NOT NULL DATE
* CREATED\_BY NOT NULL VARCHAR2(30 CHAR)
* UPDATED\_DATE DATE
* UPDATED\_BY VARCHAR2(30 CHAR)
* desc lkp\_content
* Name Null Type
* ---------------------- -------- ------------------
* CONTENT\_ID NOT NULL NUMBER
* SPACE\_ID NOT NULL NUMBER
* SPACE\_ID\_NAME VARCHAR2(4000)
* SPACE\_ID\_PATH VARCHAR2(4000)
* PROPERTY\_ID NOT NULL NUMBER
* PROPERTY\_NAME VARCHAR2(512 CHAR)
* PROP\_SHORT\_NAME VARCHAR2(10 CHAR)
* PROPERTY\_PATH VARCHAR2(512 CHAR)
* PROP\_OBSOLETE\_FLAG VARCHAR2(1)
* TOP\_PROPERTY\_ID NOT NULL NUMBER
* TOP\_PROPERTY\_NAME VARCHAR2(512 CHAR)
* TOP\_PROP\_SHORT\_NAME VARCHAR2(10 CHAR)
* TOP\_PROPERTY\_PATH VARCHAR2(512 CHAR)
* PROP\_MAIN\_COUNTRY\_ID NOT NULL NUMBER
* PROP\_MAIN\_COUNTRY\_CODE VARCHAR2(3 CHAR)
* CONTENT\_FLOOR\_PRICE NUMBER
* SRC\_CREATED\_DATE DATE
* SRC\_CREATED\_BY VARCHAR2(200 CHAR)
* SRC\_UPDATED\_DATE DATE
* SRC\_UPDATED\_BY VARCHAR2(200 CHAR)
* CREATED\_DATE NOT NULL DATE
* CREATED\_BY NOT NULL VARCHAR2(30 CHAR)
* UPDATED\_DATE DATE
* UPDATED\_BY VARCHAR2(30 CHAR)
* desc lkp\_currency
* Name Null Type
* ---------------- -------- ------------------
* CURRENCY\_ID NOT NULL NUMBER
* CURRENCY\_CODE NOT NULL VARCHAR2(3 CHAR)
* CURRENCY\_NAME NOT NULL VARCHAR2(100 CHAR)
* DATA\_SOURCE\_ID NOT NULL NUMBER
* DATA\_SOURCE\_CODE NOT NULL VARCHAR2(10 CHAR)
* SRC\_CREATED\_DATE DATE
* SRC\_CREATED\_BY VARCHAR2(200 CHAR)
* SRC\_UPDATED\_DATE DATE
* SRC\_UPDATED\_BY VARCHAR2(200 CHAR)
* CREATED\_DATE NOT NULL DATE
* CREATED\_BY NOT NULL VARCHAR2(30 CHAR)
* UPDATED\_DATE DATE
* UPDATED\_BY VARCHAR2(30 CHAR)
* desc lkp\_order
* Name Null Type
* ----------------------- -------- -------------------
* ORDER\_ID NOT NULL NUMBER
* CRM\_ORDER\_ID VARCHAR2(15 CHAR)
* UAD\_ORDER\_ID NUMBER
* CRM\_UAD\_ORDER\_ID NUMBER
* CRM\_ORDER\_NUM VARCHAR2(30 CHAR)
* CRM\_ORDER\_DESC VARCHAR2(255 CHAR)
* UAD\_ORDER\_NAME VARCHAR2(1000 CHAR)
* ORD\_STATUS\_ID NOT NULL NUMBER
* ORD\_STATUS\_CODE VARCHAR2(40 CHAR)
* ORD\_START\_DATE DATE
* ORD\_START\_DATE\_ADV\_TZ DATE
* ORD\_END\_DATE DATE
* ORD\_END\_DATE\_ADV\_TZ DATE
* ORD\_COMMITTED\_DATE DATE
* ORD\_CANCELLED\_DATE DATE
* ORD\_HOUSE\_AD\_FLAG VARCHAR2(1 CHAR)
* ORD\_RMX\_Y\_BILL\_FLAG VARCHAR2(1 CHAR)
* ORD\_MME\_IO\_TYPE\_CODE VARCHAR2(2 CHAR)
* ORD\_BILL\_TERMS\_ID NOT NULL NUMBER
* ORD\_BILL\_TERMS\_CODE VARCHAR2(40 CHAR)
* ORD\_INVC\_METHOD\_ID NOT NULL NUMBER
* ORD\_INVC\_METHOD\_CODE VARCHAR2(40 CHAR)
* ORD\_GROSS\_COST NUMBER
* ORD\_NET\_COST NUMBER
* ORD\_CURRENCY\_ID NOT NULL NUMBER
* ORD\_CURRENCY\_CODE VARCHAR2(3 CHAR)
* USD\_CORP\_CURR\_EXCH\_RATE NUMBER
* USD\_PLAN\_CURR\_EXCH\_RATE NUMBER
* ORD\_TIME\_ZONE\_ID NOT NULL NUMBER
* ORD\_TIME\_ZONE\_NAME VARCHAR2(64 CHAR)
* ADVERTISER\_ID NOT NULL NUMBER
* CRM\_SRC\_CREATED\_DATE DATE
* CRM\_SRC\_UPDATED\_DATE DATE
* UAD\_SRC\_CREATED\_DATE DATE
* UAD\_SRC\_UPDATED\_DATE DATE
* CREATED\_DATE NOT NULL DATE
* CREATED\_BY NOT NULL VARCHAR2(30 CHAR)
* UPDATED\_DATE DATE
* UPDATED\_BY VARCHAR2(30 CHAR)
* desc lkp\_placement\_line
* Name Null Type
* ----------------------------- -------- -------------------
* LINE\_ID NOT NULL NUMBER
* ORDER\_ID NUMBER
* CRM\_ORDER\_ITEM\_ID VARCHAR2(15 CHAR)
* UAD\_ORDER\_LINE\_ID NUMBER
* CRM\_ORDER\_ITEM\_LINE\_NUM NUMBER(10)
* CRM\_ORD\_ITEM\_DESC\_TEXT VARCHAR2(250 CHAR)
* UAD\_ORDER\_LINE\_NAME VARCHAR2(255 CHAR)
* LINE\_STATUS\_ID NUMBER
* LINE\_STATUS\_CODE VARCHAR2(40 CHAR)
* LINE\_START\_DATE DATE
* LINE\_START\_DATE\_ADV\_TZ DATE
* LINE\_END\_DATE DATE
* LINE\_END\_DATE\_ADV\_TZ DATE
* LINE\_EXPIRATION\_DATE DATE
* LINE\_ACTIVATED\_DATE DATE
* LINE\_COMMITTED\_DATE DATE
* LINE\_CANCELLED\_DATE DATE
* LINE\_DELETED\_DATE DATE
* LINE\_DELETED\_FLAG VARCHAR2(1 CHAR)
* LINE\_BUY\_TYPE\_ID NUMBER
* LINE\_BUY\_TYPE\_CODE VARCHAR2(30 CHAR)
* LINE\_PRIC\_TYPE\_ID NUMBER
* LINE\_PRIC\_TYPE\_CODE VARCHAR2(40 CHAR)
* LINE\_REVN\_MODEL\_ID NUMBER
* LINE\_REVN\_MODEL\_CODE VARCHAR2(40 CHAR)
* LINE\_MME\_REVN\_TYPE\_CODE VARCHAR2(30)
* LINE\_DLVR\_MODEL\_ID NUMBER
* LINE\_DLVR\_MODEL\_CODE VARCHAR2(40 CHAR)
* LINE\_INV\_CLS\_ID NUMBER
* LINE\_INV\_CLS\_CODE VARCHAR2(10 CHAR)
* LINE\_PRODUCT\_ID NUMBER
* LINE\_PRODUCT\_CODE VARCHAR2(30)
* LINE\_GROSS\_FLOOR\_PRICE\_AC\_RC NUMBER
* LINE\_GROSS\_FLOOR\_PRICE\_CRM\_QI NUMBER
* LINE\_GROSS\_FLOOR\_PRICE\_CRM\_OI NUMBER
* LINE\_NET\_TARG\_PRICE NUMBER
* LINE\_GROSS\_COST NUMBER
* LINE\_CURRENCY\_ID NUMBER
* LINE\_CURRENCY\_CODE VARCHAR2(3 CHAR)
* ADV\_CORP\_CURR\_EXCH\_RATE NUMBER
* USD\_CORP\_CURR\_EXCH\_RATE NUMBER
* USD\_PLAN\_CURR\_EXCH\_RATE NUMBER
* LINE\_TIME\_ZONE\_ID NUMBER
* LINE\_TIME\_ZONE\_NAME VARCHAR2(64 CHAR)
* BOOK\_CONTENT\_ID NUMBER
* BOOK\_AD\_POSITION\_ID NUMBER
* BOOK\_AD\_POSITION\_NAME VARCHAR2(3000 CHAR)
* BOOK\_AD\_POS\_SP\_ID\_FLOOR\_PRICE NUMBER
* BOOK\_AD\_POS\_SP\_ID\_ORD\_CURR\_ID NUMBER
* BOOK\_AD\_POS\_SP\_ID\_ORD\_CURR\_CD VARCHAR2(3 CHAR)
* CRM\_SRC\_CREATED\_DATE DATE
* CRM\_SRC\_UPDATED\_DATE DATE
* UAD\_SRC\_CREATED\_DATE DATE
* UAD\_SRC\_UPDATED\_DATE DATE
* CREATED\_DATE NOT NULL DATE
* CREATED\_BY NOT NULL VARCHAR2(30 CHAR)
* UPDATED\_DATE DATE
* UPDATED\_BY VARCHAR2(30 CHAR)
* desc lkp\_time\_zone
* Name Null Type
* ------------------- -------- -----------------
* TIME\_ZONE\_ID NOT NULL NUMBER
* TIME\_ZONE\_NAME NOT NULL VARCHAR2(64 CHAR)
* OFFSET\_FROM\_UTC\_MIN NUMBER(38)
* CREATED\_DATE NOT NULL DATE
* CREATED\_BY NOT NULL VARCHAR2(30 CHAR)
* UPDATED\_DATE DATE
* UPDATED\_BY VARCHAR2(30 CHAR)
* <.pre>
* [Home](http://sp2-crmarch-001.ads.pool.corp.sp2.yahoo.com:4080/predictsdoc/index.html) Top   
  [Data Processing](http://sp2-crmarch-001.ads.pool.corp.sp2.yahoo.com:4080/predictsdoc/predictsBackEndProcessing.html#dataProcessing)   
  [Queries](http://sp2-crmarch-001.ads.pool.corp.sp2.yahoo.com:4080/predictsdoc/predictsBackEndProcessing.html#queries)

### Data Processing

* [Back to the top](http://sp2-crmarch-001.ads.pool.corp.sp2.yahoo.com:4080/predictsdoc/predictsBackEndProcessing.html#top)
* This page describes the back end processes that ensure the data is obtained, processed and made available to the algorithm.
* YCRM Data Processing: Data from YCRM S\_ORDER\_ITEM is queried and moved to STAGINGYCRM table. Program MoveYCRMDataToYCRMStagingTable does this processing.
* **MoveYCRMDataToYCRMStagingTable** - This process moves data from YCRM tables to STAGINGTCRM table. It runs four queries one by one and gets data from YCRM. We are not running all the four queries as 'union all' because it takes a lot of time on the database side. We split the queries in four sub queries. See [below](http://sp2-crmarch-001.ads.pool.corp.sp2.yahoo.com:4080/predictsdoc/predictsBackEndProcessing.html#queries) for details on the queries.
* **MoveStagingYCRMDataToPredictsTables** - This program reads a record from stagingycrm table. Then it creates a record in customer table. It first checks if a customer with same name and address exists. If yes, then it doesn't add a new record. It stores the record's Id. Then it does the same for item. After that it writes to ratings table. Before entering a record into rating table, it checks if a rating exists for the customer and item. If it does then it increments aggregateSum column and saves the record. If none exists then it adds the record.
* **MovePredictsDataToFiles** - This program would not be necessary once the database connectivity from Apache Mahout is resolved. Until then this program generates files. It reads ratings table and generates flat files. Same with customer and item.   
  It generates four types of rating files   
    ratingBOOLUserItem - This is a boolean rating file for user's preference for item   
    ratingBOOLItemUser - This is a boolean rating file for item's preference for user. In version 1 we don't use it.   
    ratingFREQUserItem - This is a frequency rating file for user's preference for item   
    ratingFREQItemUser - This is a frequency rating file for item's preference for user. In version 1 we don't use it.
* Frequency rating files are generated using the aggregateSum column of rating table. The aggregate sum tells how many times a customer has purchased a property. We are hoping to use this mertics to enhance the rating. We will imply the rating as if customer gave the rating. Higher the number of times a customer purchased a property, higher the rating. As of 7/3/12 this is not generating great result because most of the customers still have purchased only 1 which takes this to Boolean rating. This needs some more research.

### Queries

* [Back to the top](http://sp2-crmarch-001.ads.pool.corp.sp2.yahoo.com:4080/predictsdoc/predictsBackEndProcessing.html#top)
* The importer reads from several tables. There are two sources APT and Ruby. For each there are two situations: advertiser and agency. Due to the fact that for agency orders, the actual advertiser name is stored in a different field, we have to use two queries. Therefore there are four queries as given below.
* **Source - APT, Advertiser order**
* select
* line.row\_id lineId, -- Unique id to identify this record - Using s\_order\_item's row\_id
* 'APT' lineType,
* ord.order\_num orderNumber, -- Not used in 1.0 For debug only
* company.row\_id customerId, -- Row Id of the customer
* company.name customer, -- Name of the customer
* addr.addr addrStreet1,
* addr.addr\_line\_2 addrStreet2,
* addr.addr\_line\_3 addrStreet3,
* addr.city addrCity,
* addr.state addrState,
* addr.province addrProvince,
* addr.zipcode addrPostalCode,
* addr.country addrCountry,
* company.x\_category category,
* company.x\_ycrm\_par\_cat parentCategory,
* ordx.attrib\_48 market, -- Not used in 1.0
* line.x\_ruby\_order\_line\_id rubyLineId,
* line.x\_yan\_plcmnt\_id aptLineId,
* line.x\_profile propPath, -- This is the property where advertisement is done
* line.x\_position position,
* line.x\_line\_id spaceid,
* line.x\_property rootPropertyName,
* line.x\_gross\_cpm cpm,
* ord.status\_cd orderStatus, -- Not used in 1.0
* linex.attrib\_15 mediaNetAmount, -- Not used in 1.0
* line.x\_gross\_amnt mediaGrossAmount, -- Not used in 1.0
* line.x\_yan\_adgrp\_prof audienceTargeting, -- Not used in 1.0
* line.x\_yan\_plcmnt\_prof contentTargeting, -- Not used in 1.0
* line.x\_yan\_ad\_attrib\_prof adAttributeTargeting, -- Not used in 1.0
* line.x\_yan\_freq\_prof frequencyTargeting, -- Not used in 1.0
* null rubyTargeting, -- Not used in 1.0
* line.created lineCreatedOn,
* line.last\_upd lineUpdatedOn
* from
* siebel.s\_order ord,
* siebel.s\_order\_x ordx,
* siebel.s\_order\_item line,
* siebel.s\_order\_item\_x linex,
* siebel.s\_org\_ext company,
* siebel.s\_addr\_org addr
* where
* line.created >= '01-Jan-2011' -- Start date
* and line.created <= sysdate -- End date is always today
* and line.order\_id = ord.row\_id -- Joins order header with order line
* and ord.row\_id = ordx.par\_row\_id -- Joins order header to extension table
* and line.row\_id = linex.par\_row\_id -- Joins order line to its extension table
* and line.prod\_id = '1-36D4KF' -- Ensures only media product is chosen
* and line.x\_profile not in ('/', '/site', '/mail') -- Exclude very common and generic sites
* and line.x\_profile is not null -- x\_profile contains the property path - So it must have value
* and line.par\_order\_item\_id is null -- Exclude children of auto flight - For auto flight header, the parent is null
* and ord.status\_cd in ('Committed', 'Completed', 'Exception', 'Running') -- Order status
* and ordx.attrib\_48 is not null -- Market
* and ord.accnt\_id = company.par\_row\_id -- This is advertiser query, so advertiser row id is in accnt\_id
* and ord.x\_agency\_id is null -- and agency\_id should be null
* and line.x\_ruby\_order\_line\_id is null -- This is APT query so Ruby line id should be null
* and line.x\_yan\_plcmnt\_id is not null -- This is APT query so APT Placement Id should NOT be null
* and company.pr\_addr\_id = addr.row\_id (+) -- Outer join to company's primary address
* **Source - APT, Agency order**
* select
* line.row\_id lineId,
* 'APT' lineType,
* ord.order\_num orderNumber,
* company.row\_id customerId,
* company.name customer,
* addr.addr addrStreet1,
* addr.addr\_line\_2 addrStreet2,
* addr.addr\_line\_3 addrStreet3,
* addr.city addrCity,
* addr.state addrState,
* addr.province addrProvince,
* addr.zipcode addrPostalCode,
* addr.country addrCountry,
* company.x\_category category,
* company.x\_ycrm\_par\_cat
* parentCategory,
* ordx.attrib\_48 market,
* line.x\_ruby\_order\_line\_id rubyLineId,
* line.x\_yan\_plcmnt\_id aptLineId,
* line.x\_profile propPath,
* line.x\_position position,
* line.x\_line\_id spaceid,
* line.x\_property rootPropertyName,
* line.x\_gross\_cpm cpm,
* ord.status\_cd orderStatus,
* linex.attrib\_15 mediaNetAmount,
* line.x\_gross\_amnt mediaGrossAmount,
* line.x\_yan\_adgrp\_prof audienceTargeting,
* line.x\_yan\_plcmnt\_prof contentTargeting,
* line.x\_yan\_ad\_attrib\_prof adAttributeTargeting,
* line.x\_yan\_freq\_prof frequencyTargeting,
* null rubyTargeting,
* line.created lineCreatedOn,
* line.last\_upd lineUpdatedOn
* from
* siebel.s\_order ord,
* siebel.s\_order\_x ordx,
* siebel.s\_order\_item line,
* siebel.s\_order\_item\_x linex,
* siebel.s\_org\_ext company,
* siebel.s\_addr\_org addr
* where
* line.created >= '01-Jan-2011'
* and line.created <= sysdate
* and line.order\_id = ord.row\_id
* and ord.row\_id = ordx.par\_row\_id
* and line.row\_id = linex.par\_row\_id
* and line.prod\_id = '1-36D4KF'
* and line.x\_profile not in ('/', '/site', '/mail')
* and line.x\_profile is not null
* and line.par\_order\_item\_id is null
* and ord.status\_cd in ('Committed', 'Completed', 'Exception', 'Running')
* and ordx.attrib\_48 is not null
* and ord.x\_agency\_id = company.par\_row\_id -- This is an agency order, so company name is in agency\_id (why? see next line)
* and ord.x\_agency\_id is not null -- Agency id is in accnt\_id, and company id is in agency\_id (go figure! :-)
* and line.x\_ruby\_order\_line\_id is null
* and line.x\_yan\_plcmnt\_id is not null
* and company.pr\_addr\_id = addr.row\_id (+)
* **Source - Ruby, Advertiser order**
* select
* line.row\_id lineId,
* 'RUBY' lineType,
* ord.order\_num orderNumber,
* company.row\_id customerId,
* company.name customer,
* addr.addr addrStreet1,
* addr.addr\_line\_2 addrStreet2,
* addr.addr\_line\_3 addrStreet3,
* addr.city addrCity,
* addr.state addrState,
* addr.province addrProvince,
* addr.zipcode addrPostalCode,
* addr.country addrCountry,
* company.x\_category category,
* company.x\_ycrm\_par\_cat parentCategory,
* ordx.attrib\_48 market,
* line.x\_ruby\_order\_line\_id rubyLineId,
* line.x\_yan\_plcmnt\_id aptLineId,
* line.x\_profile propPath,
* line.x\_position position,
* line.x\_line\_id spaceid,
* line.x\_property rootPropertyName,
* line.x\_gross\_cpm cpm,
* ord.status\_cd orderStatus,
* linex.attrib\_15 mediaNetAmount,
* line.x\_gross\_amnt mediaGrossAmount,
* line.x\_yan\_adgrp\_prof audienceTargeting,
* line.x\_yan\_plcmnt\_prof contentTargeting,
* line.x\_yan\_ad\_attrib\_prof adAttributeTargeting,
* line.x\_yan\_freq\_prof frequencyTargeting,
* prof.targeting\_string rubyTargeting,
* line.created lineCreatedOn,
* line.last\_upd lineUpdatedOn
* from
* siebel.s\_order ord,
* siebel.s\_order\_x ordx,
* siebel.s\_order\_item line,
* siebel.s\_order\_item\_x linex,
* siebel.s\_org\_ext company,
* siebel.s\_addr\_org addr,
* siebel.cx\_ycrm\_profile prof
* where
* line.created >= '01-Jan-2011'
* and line.created <= sysdate
* and line.order\_id = ord.row\_id
* and ord.row\_id = ordx.par\_row\_id
* and line.row\_id = linex.par\_row\_id
* and line.prod\_id = '1-36D4KF'
* and line.x\_profile not in ('/', '/site', '/mail')
* and line.x\_profile is not null
* and line.par\_order\_item\_id is null
* and ord.status\_cd in ('Committed', 'Completed', 'Exception', 'Running')
* and ordx.attrib\_48 is not null
* and ord.accnt\_id = company.par\_row\_id
* and ord.x\_agency\_id is null
* and line.x\_ruby\_order\_line\_id is not null
* and line.x\_yan\_plcmnt\_id is null
* and company.pr\_addr\_id = addr.row\_id (+)
* and line.x\_ycrm\_profile\_id = prof.row\_id (+) -- Outer join to Ruby Profile to get Ruby targeting
* **Source - Ruby, Agency order**
* select
* line.row\_id lineId,
* 'RUBY' lineType,
* ord.order\_num orderNumber,
* company.row\_id customerId,
* company.name customer,
* addr.addr addrStreet1,
* addr.addr\_line\_2 addrStreet2,
* addr.addr\_line\_3 addrStreet3,
* addr.city addrCity,
* addr.state addrState,
* addr.province addrProvince,
* addr.zipcode addrPostalCode,
* addr.country addrCountry,
* company.x\_category category,
* company.x\_ycrm\_par\_cat parentCategory,
* ordx.attrib\_48 market,
* line.x\_ruby\_order\_line\_id rubyLineId,
* line.x\_yan\_plcmnt\_id aptLineId,
* line.x\_profile propPath,
* line.x\_position position,
* line.x\_line\_id spaceid,
* line.x\_property rootPropertyName,
* line.x\_gross\_cpm cpm,
* ord.status\_cd orderStatus,
* linex.attrib\_15 mediaNetAmount,
* line.x\_gross\_amnt mediaGrossAmount,
* line.x\_yan\_adgrp\_prof audienceTargeting,
* line.x\_yan\_plcmnt\_prof contentTargeting,
* line.x\_yan\_ad\_attrib\_prof adAttributeTargeting,
* line.x\_yan\_freq\_prof frequencyTargeting,
* prof.targeting\_string rubyTargeting,
* line.created lineCreatedOn,
* line.last\_upd lineUpdatedOn
* from
* siebel.s\_order ord,
* siebel.s\_order\_x ordx,
* siebel.s\_order\_item line,
* siebel.s\_order\_item\_x linex,
* siebel.s\_org\_ext company,
* siebel.s\_addr\_org addr,
* siebel.cx\_ycrm\_profile prof
* where
* line.created >= '01-Jan-2011'
* and line.created <= sysdate
* and line.order\_id = ord.row\_id
* and ord.row\_id = ordx.par\_row\_id
* and line.row\_id = linex.par\_row\_id
* and line.prod\_id = '1-36D4KF'
* and line.x\_profile not in ('/', '/site', '/mail')
* and line.x\_profile is not null
* and line.par\_order\_item\_id is null
* and ord.status\_cd in ('Committed', 'Completed', 'Exception', 'Running')
* and ordx.attrib\_48 is not null
* and ord.x\_agency\_id = company.par\_row\_id
* and ord.x\_agency\_id is not null
* and line.x\_ruby\_order\_line\_id is not null
* and line.x\_yan\_plcmnt\_id is null
* and company.pr\_addr\_id = addr.row\_id (+)
* and line.x\_ycrm\_profile\_id = prof.row\_id (+)
* UAD Data Processing: To be done.