|  |  |
| --- | --- |
|  | |
| Türk Ekonomi Bankası | |
| Credit Risk | |
| Operational & ETL | |
| Requirements | |
| Prepared by | |
|  | |
|  | |
| Author: | Muhammad Usman Shahrukh |
| Reviewed By: |  |
| Version: | 0.2 |
| Date Issued: | Aug 09, 2017 |
| Location: | Islamabad, Pakistan |

Version History

| Version No. | Date | Changed By | Changes Made |
| --- | --- | --- | --- |
| 0.1 | Aug 1, 2017 | Muhammad Usman Shahrukh | Initial Draft |
| 0.2 | Aug 9, 2017 | Muhammad Usman Shahrukh | Added tables such as Account Status, Event and Customer interaction.  Modified a few existing tables. |

**Distribution List**

|  |  |
| --- | --- |
| Name | Company |
| Cenk AKIN | TEB |
| Muge ZEREN YÜKSEL | TEB |
| Gizem ÖZDEN | TEB |
| Ali Özer ÇETİN | TEB |
| Özgür KAYNAR | Komtas |
| Esat DUMLUPINAR | Komtas |

Contents

[Türk Ekonomi Bankası 0](#_Toc521598941)

[Credit Risk 0](#_Toc521598942)

[Operational & ETL 0](#_Toc521598943)

[Requirements 0](#_Toc521598944)

[1 Purpose 9](#_Toc521598945)

[2 Databases 9](#_Toc521598946)

[3 Base Tables 9](#_Toc521598947)

[3.1 Customer Base 9](#_Toc521598948)

[3.1.1 Table Name 9](#_Toc521598949)

[3.1.2 Attributes 10](#_Toc521598950)

[3.1.3 Extraction Logic 10](#_Toc521598951)

[3.1.4 Aggregation Strategy/Grain 10](#_Toc521598952)

[3.1.5 Primary Index 10](#_Toc521598953)

[3.2 Merchant Segmentation Table -Daily 10](#_Toc521598954)

[3.2.1 Table Name 10](#_Toc521598955)

[3.2.2 Attributes 10](#_Toc521598956)

[3.2.3 Extraction Logic 11](#_Toc521598957)

[3.2.4 Grain 11](#_Toc521598958)

[3.2.5 Primary Index 11](#_Toc521598959)

[3.3 Customer Card Information 12](#_Toc521598960)

[3.3.1 Table Name 12](#_Toc521598961)

[3.3.2 Attributes 12](#_Toc521598962)

[3.3.3 Extraction Logic 12](#_Toc521598963)

[3.3.4 Grain 12](#_Toc521598964)

[3.3.5 Primary Index 13](#_Toc521598965)

[3.4 Related Supplementary Credit Cards 13](#_Toc521598966)

[3.4.1 Table name 13](#_Toc521598967)

[3.4.2 Attributes 13](#_Toc521598968)

[3.4.3 Extraction Logic 13](#_Toc521598969)

[3.4.4 Grain 13](#_Toc521598970)

[3.4.5 Primary index 13](#_Toc521598971)

[3.5 Customer Information 14](#_Toc521598972)

[3.5.1 Table Name 14](#_Toc521598973)

[3.5.2 Attributes 14](#_Toc521598974)

[3.5.3 Extraction Logic 14](#_Toc521598975)

[3.5.4 Grain 15](#_Toc521598976)

[3.5.5 Primary Index 15](#_Toc521598977)

[3.6 Customer Monthly Information 15](#_Toc521598978)

[3.6.1 Table Name 15](#_Toc521598979)

[3.6.2 Attributes 15](#_Toc521598980)

[3.6.3 Extraction Logic 16](#_Toc521598981)

[3.6.4 Grain 17](#_Toc521598982)

[3.6.5 Primary Index 17](#_Toc521598983)

[3.7 Statement Information 17](#_Toc521598984)

[3.7.1 Table name 17](#_Toc521598985)

[3.7.2 Attributes 17](#_Toc521598986)

[3.7.3 Extraction Logic 18](#_Toc521598987)

[3.7.4 Grain 18](#_Toc521598988)

[3.7.5 Primary index 19](#_Toc521598989)

[3.8 Product Target Table 19](#_Toc521598990)

[3.8.1 Table name 19](#_Toc521598991)

[3.8.2 Attributes 19](#_Toc521598992)

[3.8.3 Extraction Logic 19](#_Toc521598993)

[3.8.4 Grain 19](#_Toc521598994)

[3.8.5 Primary index 20](#_Toc521598995)

[3.9 Bureau Data 20](#_Toc521598996)

[3.9.1 Table name 20](#_Toc521598997)

[3.9.2 Attributes 20](#_Toc521598998)

[3.9.3 Extraction Logic 20](#_Toc521598999)

[3.9.4 Grain 20](#_Toc521599000)

[3.9.5 Primary index 20](#_Toc521599001)

[3.10 Employment Information 21](#_Toc521599002)

[3.10.1 Table name 21](#_Toc521599003)

[3.10.2 Attributes 21](#_Toc521599004)

[3.10.3 Extraction Logic 21](#_Toc521599005)

[3.10.4 Grain 21](#_Toc521599006)

[3.10.5 Primary index 21](#_Toc521599007)

[3.11 Model Result Table 21](#_Toc521599008)

[3.11.1 Table name 21](#_Toc521599009)

[3.11.2 Attributes 22](#_Toc521599010)

[3.11.3 Extraction Logic 22](#_Toc521599011)

[3.11.4 Primary index 22](#_Toc521599012)

[3.12 Employment Employee Information 22](#_Toc521599013)

[3.12.1 Table name 22](#_Toc521599014)

[3.12.2 Attributes 22](#_Toc521599015)

[3.12.3 Extraction Logic 23](#_Toc521599016)

[3.12.4 Grain 23](#_Toc521599017)

[3.12.5 Primary index 23](#_Toc521599018)

[3.13 Money Transfer 23](#_Toc521599019)

[3.13.1 Table name 23](#_Toc521599020)

[3.13.2 Attributes 23](#_Toc521599021)

[3.13.3 Extraction Logic 23](#_Toc521599022)

[3.13.4 Grain 24](#_Toc521599023)

[3.13.5 Primary index 24](#_Toc521599024)

[3.13.6 Secondary index 24](#_Toc521599025)

[3.14 Overdraft Status 24](#_Toc521599026)

[3.14.1 Table name 24](#_Toc521599027)

[3.14.2 Attributes 24](#_Toc521599028)

[3.14.3 Extraction Logic 25](#_Toc521599029)

[3.14.4 Grain 25](#_Toc521599030)

[3.14.5 Primary index 25](#_Toc521599031)

[3.15 Insurance Information 25](#_Toc521599032)

[3.15.1 Table name 25](#_Toc521599033)

[3.15.2 Attributes 25](#_Toc521599034)

[3.15.3 Extraction Logic 25](#_Toc521599035)

[3.15.4 Grain 26](#_Toc521599036)

[3.15.5 Primary index 26](#_Toc521599037)

[3.16 Restructured Payments 26](#_Toc521599038)

[3.16.1 Table name 26](#_Toc521599039)

[3.16.2 Attributes 26](#_Toc521599040)

[3.16.3 Extraction Logic 26](#_Toc521599041)

[3.16.4 Grain 27](#_Toc521599042)

[3.16.5 Primary index 27](#_Toc521599043)

[3.17 Product Ownership 27](#_Toc521599044)

[3.17.1 Table name 27](#_Toc521599045)

[3.17.2 Attributes 27](#_Toc521599046)

[3.17.3 Extraction Logic 27](#_Toc521599047)

[3.17.4 Grain 28](#_Toc521599048)

[3.17.5 Primary index 28](#_Toc521599049)

[3.18 Account Status 28](#_Toc521599050)

[3.18.1 Table name 28](#_Toc521599051)

[3.18.2 Attributes 28](#_Toc521599052)

[3.18.3 Extraction Logic 28](#_Toc521599053)

[3.18.4 Grain 28](#_Toc521599054)

[3.18.5 Primary index 28](#_Toc521599055)

[3.19 Customer Interaction 29](#_Toc521599056)

[3.19.1 Table name 29](#_Toc521599057)

[3.19.2 Attributes 29](#_Toc521599058)

[3.19.3 Extraction Logic 29](#_Toc521599059)

[3.19.4 Grain 29](#_Toc521599060)

[3.19.5 Primary index 29](#_Toc521599061)

[3.20 Customer Events 29](#_Toc521599062)

[3.20.1 Table name 29](#_Toc521599063)

[3.20.2 Attributes 30](#_Toc521599064)

[3.20.3 Extraction Logic 30](#_Toc521599065)

[3.20.4 Grain 30](#_Toc521599066)

[3.20.5 Primary index 30](#_Toc521599067)

[3.21 Lookup Tables 30](#_Toc521599068)

[3.21.1 MERCH 30](#_Toc521599069)

[3.21.2 CURR 31](#_Toc521599070)

[3.21.3 LOC 31](#_Toc521599071)

[3.21.4 CARD\_TYPE 31](#_Toc521599072)

[3.21.5 CUST\_TYPE 31](#_Toc521599073)

[3.21.6 CARD\_SEG 31](#_Toc521599074)

[3.21.7 OCCUP 32](#_Toc521599075)

[3.21.8 EMPLNT 32](#_Toc521599076)

[3.21.9 CITY 32](#_Toc521599077)

[3.21.10 DIST 32](#_Toc521599078)

[3.21.11 MN\_BRNCH 32](#_Toc521599079)

[3.21.12 PROD 32](#_Toc521599080)

[3.21.13 MDL\_TYPE 33](#_Toc521599081)

[3.21.14 TYPE\_OF\_PMT 33](#_Toc521599082)

[3.21.15 TYPE\_OF\_TRNSFR 33](#_Toc521599083)

[3.21.16 INSR\_TYPE 33](#_Toc521599084)

[3.21.17 STS 33](#_Toc521599085)

[3.21.18 CHNL 34](#_Toc521599086)

[3.21.19 INTRACTN\_RSN 34](#_Toc521599087)

# Purpose

The purpose of this document is to share the Operational and ETL requirements for Credit Risk Scoring. The following sections describe in detail the users, databases and tables that need to be created and tested/verified as they will serve as the base for this project.

# Databases

Following two databases will be created

|  |  |  |
| --- | --- | --- |
| **Database Name** | **Space Required** | **Purpose** |
| <database name>\_CRSC | 1.5 TB | Database to be used for all credit risk related analysis |
| <database name>\_METADATA | 1GB | Database to be used for TDWM metadata  \*Incase Teradata Warehouse Miner is used. |

Table 1 - Databases

# Base Tables

This section describes the details of the base tables which are required for Behavior Segmentation. Following are the properties common to all the tables except lookups

|  |  |
| --- | --- |
| **Primary Index** | CARD\_ID |
| **Type of Traffic** | Credit Transactions |
| **Inclusions** | \*Needs to be filled |
| **Exclusions** | \*Needs to be filled |
| **Run-date** | \*Needs to be filled |
| **Duration of Analysis** | \*Needs to be filled |

Table 2 - Common Properties

## Customer Base

This table will contain information about base selection.

**DON’T HAVE IT**

### Table Name

CUSTOMER\_BASE\_CR

### Attributes

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| CUST\_ID | VARCHAR(20) | Unique identifier for each TEB customer who has an account or is availing TEB's products |
| CARD\_ID | VARCHAR(20) | Unique identifier for the credit card that belongs to TEB's customer |
| STMT\_DT | DATE | Statement date |
| TGT | BYTEINT | Default/Non-Default |

Table 3 – CUSTOMER\_BASE\_BA

### Extraction Logic

Following steps represent extraction logic:

* Extract a list of 2 Years active customers as on run-date and get following information about each subscriber
  + Card ID
  + Statement Date
  + Target Label (Default/Non-Default)

### Aggregation Strategy/Grain

The grain of the table will be CARD\_ID

### Primary Index

CARD\_ID

## Merchant Segmentation Table -Daily

This table will contain information on a daily for credit card transactions and corresponding merchants.

**Table created with stated columns below.**

### Table Name

**MERCHANT\_SEGMENT\_TABLE**

### Attributes

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| CUST\_ID | NUMERIC(8,0) | Unique identifier for each TEB customer who has an account or is availing TEB's products |
| CARD\_ID | VARCHAR(19) | Unique identifier for the credit card that belongs to TEB's customer |
| TRX\_DATE | DATE | Date of transaction for the card |
| MERCHANT\_ID | VARCHAR(4) | Refer to REF\_MERCHANTID\_ENG and REF\_MERCHANTCAT lookup tables to get the Merchant where the card was used |
| CURRENCY | VARCHAR(3) | the currency in which the transaction was made (TL : 949) |
| ECOMMERCE\_FLAG | INTEGER | 1 represents e-commerce transaction. 0 otherwise |
| CITY | VARCHAR(20) | City in which the transaction was made |
| COUNTRY | VARCHAR(3) | Country in which the transaction was made |
| POSTPONED\_PAY\_FLAG | VARCHAR(1) | 0 represents the payment of the transaction is postponed. 0 otherwise |
| LATE\_INSTALLEMENT\_FLG | INTEGER | 1 represents the payment of the transaction is splitted into installements afterward |
| OTC | NUMERIC(4,0) | OTC and OTS define trx type together (detailed grouping) |
| OTS | NUMERIC(4,0) | OTC and OTS define trx type together (detailed grouping) |
| ISLEMKODU | VARCHAR(2) | ISLEMKODU and ISLEMSUBKODU define trx type together (high level grouping) |
| ISLEMSUBKODU | VARCHAR(2) | ISLEMKODU and ISLEMSUBKODU define trx type together (high level grouping) |
| TRX\_AMT\_TL | NUMERIC(28,6) | Transaction amount in Turkish Lira |
| TRX\_AMNT\_FX | NUMERIC(28,6) | Transaction amount in case of any other currency |
| TRX\_AMT\_DOLLAR | NUMERIC(28,6) | Transaction amount in case of dollar |
| TRX\_CNT | BIGINT | The number of times that card was used i.e. the number of transactions |

### Extraction Logic

Followings steps represent extraction logic:

* For all the customers in CUSTOMER\_BASE\_CR table, for each credit card, collect following information
  + Merchant
  + Date/Time of Transaction
  + Currency
  + E-Commerce Flag
  + Location of the transaction

Information/ rules for all these can be found in the lookup tables

* For each unique combination of above mentioned fields, calculate the following
  + Amount of Transactions in Turkish Lira
  + Amount of Transaction in Foreign Currency if it exists
  + Count of Transactions
* OTC, OTS, ISLEMKODU, ISLEMSUBKODU explanations are in ref\_cc\_trx table in Aster. (03\_01\_lp\_merchant\_data excel in 00\_Reference folder)
* To get expenses from this table, the filter below should be used.
  + ((ISLEMKODU ='BI' and ISLEMSUBKODU ='IN') or

(ISLEMKODU ='BK' and ISLEMSUBKODU in ('EW','MW','SA','WS')) or ISLEMKODU IN ('SA','NA'))

* 3 txn\_name should be check if we should include these txn types in amnt\_total\_spend calculation.

|  |
| --- |
| Avans Mil Hizmet Bedeli |
| Avans Mil İhlal Bedeli |
| Avans Mil Kullanım |

### Grain

First 14 columns will form the grain for this table. We need a separate row for each unique combination of these 14 columns.

### Primary Index

CARD\_ID

## Customer Card Information

This table will contain information about card information for the customer (Latest Snapshot)

**Table created with stated columns below.**

### Table Name

**CUST\_CARD\_INFO**

### Attributes

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| CUST\_ID | NUMERIC(8,0) | Unique identifier for each TEB customer who has an account or is availing TEB's products |
| CARD\_ID | VARCHAR(20) | Unique identifier for the credit card that belongs to TEB's customer |
| DATE | DATE | Date of the end of the month |
| PRIMSUPPVIRT\_TYPE | VARCHAR(20) | Card type (VIRT: Virtual, SUPP: Supplementary, PRIM: Primary) |
| BIN | VARCHAR(6) | Card type (detailed grouping – explanations are in 00\_Referece folder - 03\_01\_Card\_BIN excel) |
| Card\_Segment | VARCHAR(20) | Credit Card Segments (Business, Platinum etc.) |
| DT\_OF\_ISSUE | DATE | Date of issue (for each card ) |
| DT\_OF\_CLSR | DATE | Date of closure of the card |
| Curr\_Limit | NUMERIC(19,4) | Current CC limit |
| CARD\_TYPE | VARCHAR(20) | Card type (Visa, Troy, Mastercard) |
| Last\_Trx\_Date | DATE | Date of last CC transaction |
| Statement\_Date | DATE | Date of CC statement |
| Cash\_Limit | NUMERIC(19,4) | Current CC cash limit |
| Yearly\_Nonpayment\_Count | NUMERIC(3,0) | Unpaid statements month count in a year |
| Full\_Payment\_Stat | VARCHAR(1) | Full paid statement status |
| DaysPastDue | NUMERIC(8,0) | Days past due date |
| valid\_EH | VARCHAR(1) | To filter valid credit cards |
| registered\_YN | VARCHAR(1) | To filter valid credit cards |
| cc\_status | VARCHAR(2) | To filter valid credit cards |
| followup\_status | VARCHAR(2) | To filter valid credit cards |
| cc\_substatus | VARCHAR(3) | To filter valid credit cards |
| valid\_f | INTEGER | valid\_EH='E' AND  registered\_YN ='Y' AND  cc\_status ='N' AND  followup\_status ='N' AND  cc\_substatus is null |
| DT\_OF\_ACTV | DATE | Date of activation (for each card) |
| CC\_Close\_Reason | VARCHAR(18) | Refers to Ref\_CC\_Close\_Reason Table. |
| CC\_Close\_Subreason | VARCHAR(100) | CC Close Sub-Reason |
| Legal\_Followup | NUMERIC(28,6) | Legal followup flag |
| Last\_Payment\_Channel | VARCHAR(40) | Channel of last cc payment. Refers to Ref\_Last\_Payment\_Channel Table |
| Acq\_Channel | VARCHAR(30) | CC Acquisition channel. Refers to Ref\_Acq\_Channel Table |
| Product | VARCHAR(255) | Product type of credit card (detailed) |
| Brand | VARCHAR(5) | Credit Card brand (Bonus, World) |

Table 4 – CUST\_CARD\_INFO\_CR

### Extraction Logic

Followings steps represent extraction logic:

* For all the Customers in CUSTOMER\_BASE\_CR table, for Credit Card, collect following information
  + PrimSuppVirt Type (Primary, Supplementary or Virtual)
  + Card Segment (Business, Platinum etc.)
  + Date of issue of the first card (for each card)
  + Date of activation for the latest card (for each card)
  + Date of closure of the card

Information/ rules for all these can be found in the lookup tables

* To filter valid credit cards, this filter should be used and added as “valid\_f” field:

valid\_EH='E' AND

registered\_YN ='Y' AND

cc\_status ='N' AND

followup\_status ='N' AND /\* non-default customers \*/

cc\_substatus is null

Note that all information should be as of run-date.

### Grain

First 2 columns will form the grain for this table. We need a separate row for each unique combination of these 2 columns.

### Primary Index

CARD\_ID

## Related Supplementary Credit Cards

This table will contain information about relationship of Supplementary Credit Cards to the parent Credit Card

**Table created with stated columns below.**

### Table name

SUP\_CC\_RLT

### Attributes

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| CUST\_ID | NUMERIC(8) | Unique identifier for each TEB customer who has an account or is availing TEB's products |
| CARD\_ID | VARCHAR(20) | Unique identifier for the credit card that belongs to TEB's customer |
| DATE | TIMESTAMP (without time zone) | Monthly date (end of the each month except january. For January, it contains the data for 1st and 31st ) |
| SUPP\_CARD\_ID | VARCHAR(20) | Unique identifier for the supplementary credit card that belongs to TEB's customer |
| Supp\_Card\_Cust\_ID | NUMERIC(8) | Unique identifier for each TEB customer who has an account or is availing TEB's products |
| valid\_EH | VARCHAR(1) | To filter valid credit cards |
| registered\_YN | VARCHAR(1) | To filter valid credit cards |
| cc\_status | VARCHAR(2) | To filter valid credit cards |
| followup\_status | VARCHAR(2) | To filter valid credit cards |
| cc\_substatus | VARCHAR(3) | To filter valid credit cards |
| valid\_f | INTEGER | valid\_EH='E' AND  registered\_YN ='Y' AND  cc\_status ='N' AND  followup\_status ='N' AND  cc\_substatus is null |

Table 5 – SUP\_CC\_RLT\_CR

### Extraction Logic

Followings steps represent extraction logic:

* For all the subscribers in CUSTOMER\_BASE\_CR table, for each Credit Card, collect following information
  + Related supplementary credit card

Information/ rules for all these can be found in the lookup tables

* To filter valid credit cards, this filter should be used as “valid\_f” field:

valid\_EH='E' AND

registered\_YN ='Y' AND

cc\_status ='N' AND

followup\_status ='N' AND /\* non-default customers \*/

cc\_substatus is null

Note that all information should be as of run-date.

### Grain

First 3 columns will form the grain for this table. We need a separate row for each unique combination of these 3 columns.

### Primary index

CARD\_ID

## Customer Information

**Table created for 3 time periods of the model with stated columns below.**

FILTER:

Customer\_type = Real and Business\_Line in (Individual&Affluent) or Portfolio\_Type in (Individual& Affluent)

These filters shows that customer is in target population which is real individual customers.

### Table Name

RISK.CUST\_INFO\_AUG17

RISK.CUST\_INFO\_OCT17

RISK.CUST\_INFO\_DEC17

Attributes kontrol edilmedi.

### Attributes

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| CUST\_ID | NUMERIC(8) | Unique identifier for each TEB customer who has an account or is availing TEB's products |
| DT | TIMESTAMP (without time zone) | The end of the month (contains only 1 date) |
| OCC\_KEY | VARCHAR(4) | The occupation of the customer. Refers to Ref\_Occupation Table |
| SEX | VARCHAR(1) | Gender of the customer (K: Woman, E: Man) |
| EDU\_KEY | VARCHAR(3) | Education level of the customer. Refers to Ref\_Education Table |
| EMP\_KEY | VARCHAR(3) | Employment Detail. Refers to Ref\_Employment table |
| DOB | DATE | Date of birth |
| CUST\_ACQ\_DT | TIMESTAMP (without time zone) | Customer acquisition date |
| ADDR\_CITY\_KEY | VARCHAR(3) | The city where the bank can reach out to the customer |
| ADDR\_DIST\_KEY | VARCHAR(2) | The district where the bank can reach out to the customer |
| MARITAL\_STS | VARCHAR(1) | Marital status of the customer |
| SALARY\_CUST\_FLG | VARCHAR(1) | Customer has Salary account in TEB (E: Yes /H: No) |
| MN\_BRNCH\_KEY | NUMERIC(4) | The main branch for the customer |
| CUST\_TYPE\_KEY\_P | VARCHAR(3) | Customer Portfolio Type. Refer to REF\_CUST\_TYPE\_V2 lookup table |
| CUST\_TYPE\_KEY\_I | NUMERIC(3) | CustomerBUSINESS TYPE. Refer to REF\_CUST\_TYPE\_IV2 lookup table |
| INC | NUMERIC | Monthly income declared by the customer |
| CURR\_LIMIT | NUMERIC(19,4) | Credit limit for the credit card |

Table 6 – CUST\_INFO\_CR

### Extraction Logic

Followings steps represent extraction logic:

* For all the Customers in CUSTOMER\_BASE\_BA table, collect following information
  + Monthly income declared by the customer
  + Occupation
  + Gender
  + Employment
  + Date of Birth
  + Customer acquisition date
  + Home? city
  + Home? district
  + City of current address
  + District of current address
  + Main branch
  + Customer Type (Affluent, Mass etc.)
  + Credit Card Limit (The limit is shared amongst cards)

Information/ rules for all these can be found in the lookup tables. In ASTER;

"risk"."ref\_occupation"

"risk"."ref\_title" (bu neyin ref i anlamadım?)

"risk"."ref\_income"

"risk"."ref\_education"

"risk"."ref\_employment"

Note that all information should be as of run-date.

* Table should be reduced by applying a filter to select only Mass and Affluent Portfolio Customers.

### Grain

First column will form the grain for this table. We need a separate row for each customer.

### Primary Index

CUST\_ID

## Customer Monthly Information

Cust\_info is created. Prepared code will be updated in the same way with date.

FILTER:

Customer\_type = Real and Business\_Line in (Individual&Star) or Portfolio\_Type in (Individual&Star)

These filters shows that customer is in target population which is real individual customers.

### Table Name

RISK.CUST\_INFO\_MONTHLY

### Attributes

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| CUST\_ID | NUMERIC(8) | Unique identifier for each TEB customer who has an account or is availing TEB's products |
| DT | TIMESTAMP (without time zone) | The end of the month (contains only 1 date) |
| OCC\_KEY | VARCHAR(4) | The occupation of the customer. Refers to Ref\_Occupation Table |
| SEX | VARCHAR(1) | Gender of the customer (K: Woman, E: Man) |
| EDU\_KEY | VARCHAR(3) | Education level of the customer. Refers to Ref\_Education Table |
| EMP\_KEY | VARCHAR(3) | Employment Detail. Refers to Ref\_Employment table |
| DOB | DATE | Date of birth |
| CUST\_ACQ\_DT | TIMESTAMP (without time zone) | Customer acquisition date |
| ADDR\_CITY\_KEY | VARCHAR(3) | The city where the bank can reach out to the customer |
| ADDR\_DIST\_KEY | VARCHAR(2) | The district where the bank can reach out to the customer |
| MARITAL\_STS | VARCHAR(1) | Marital status of the customer |
| SALARY\_CUST\_FLG | VARCHAR(1) | Customer has Salary account in TEB (E: Yes /H: No) |
| MN\_BRNCH\_KEY | NUMERIC(4) | The main branch for the customer |
| CUST\_TYPE\_KEY\_P | VARCHAR(3) | Customer Portfolio Type. Refer to REF\_CUST\_TYPE\_V2 lookup table |
| CUST\_TYPE\_KEY\_I | NUMERIC(3) | CustomerBUSINESS TYPE. Refer to REF\_CUST\_TYPE\_IV2 lookup table |

Extraction Table 8 – CUST\_MNTH\_INFO\_CR

### Extraction Logic

Followings steps represent extraction logic:

* For all the Customers in CUSTOMER\_BASE\_CR table, in whichever month the relevant information changed, collect following information
  + Year Month
  + Monthly income declared by the customer
  + Occupation
  + Gender
  + Employment
  + Date of Birth
  + Customer acquisition date
  + Home? city
  + Home? district
  + City of current address
  + District of current address
  + Main branch
  + Customer Type (Affluent, Mass etc.)

Information/ rules for all these can be found in the lookup tables. In ASTER;

"risk"."ref\_occupation"

"risk"."ref\_title"

"risk"."ref\_income"

"risk"."ref\_education"

"risk"."ref\_employment"

Note that all information should be as of run-date.

* Table should be reduced by applying a filter to select only Mass and Affluent Portfolio Customers.
* Discussed about cc limit with Usman and decided not to add on this table. We have cc limit information on customer card information table.

### Grain

First column will form the grain for this table. We need a separate row for each customer.

### Primary Index

CUST\_ID

## Statement Information

Statement table for credit card information is created. Asset and loan informations will be added.

FILTER:

Customer\_type = Real and Business\_Line in (Individual&Affluen) or Portfolio\_Type in (Individual&Affluent)

These filters shows that customer is in target population which is real individual customers.

### Table name

STMNT\_INFO\_CR

### Attributes

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| CUST\_ID | NUMERIC(8) | Unique identifier for each TEB customer who has an account or is availing TEB's products |
| CARD\_ID | VARCHAR(20) | Unique identifier for the credit card that belongs to TEB's customer |
| AS\_OF\_DT | DATE | Monthly date (end of the each month) |
| STMNT\_DT | DATE | Credit card statement date |
| DUE\_DT | DATE | Credit card payment due date |
| LAST\_TXN\_AMT | NUMERIC | Last transaction amount |
| LAST\_TXN\_DT | DATE | Last transaction date |
| LMT | NUMERIC | Credit card limit |
| PREV\_STMT\_DT | DATE | Previous statement date |
| PREV\_DUE\_DT | DATE | Previous due date |
| NXT\_STMT\_DT | DATE | Next statement date |
| NXT\_DUE\_DT | DATE | Next due date |
| CL\_BRND\_TP\_NM | VARCHAR(50) | Card type (Visa, Troy, Mastercard) |
| CL\_BIN\_CLSS\_NM | VARCHAR(150) | Credit Card Segments (Business, Platinum etc.) |
| cust\_no | INTEGER | Customer cc account number? |
| CL\_CRD\_ST\_NM | VARCHAR(150) | Refers to REF\_ CL\_CRD\_ST\_NM table |
| NXT\_FEE\_DT | DATE |  |
| CL\_CC\_FLLW\_ST\_TP\_NM | VARCHAR(150) | (Normal or Null) |
| CL\_CC\_CST\_ST\_TP\_NM | VARCHAR(150) | (Normal, Null or VEFAT:Decease) |
| CL\_WORST\_XTRT\_PYMT\_ST\_TP\_NM | VARCHAR(150) | (Normal, Gecikme: Delay, Dönemsel Gecikme: Periodic Delay or Null) |
| CL\_CST\_XTRT\_PYMT\_ST\_TP\_NM | VARCHAR(150) | (Normal, Gecikme: Delay, Dönemsel Gecikme: Periodic Delay or Null) |
| CL\_WORST\_STMT\_VAL\_NM | VARCHAR(150) | Refers to REF\_ CL\_WORST\_STMT\_VAL\_NM Table |
| CL\_CC\_CST\_FLLW\_ST\_TP\_NM | VARCHAR(150) | Reference folder’inda 07\_01\_Statement\_LP excel’inde ref oluşturmaya başladım. Ingilizceleri bulamadım. Sen tamamlayabilir misin. Kodda datamart kodu içerisinde |
| DLQ\_STRT\_DT | DATE | ???? |
| DLQ\_DAY\_CNT | NUMERIC |  |
| CASH\_LMT | NUMERIC |  |
| RSK\_LVL | NUMERIC |  |
| MIN\_AMNT | NUMERIC(17,2) |  |
| MIN\_AMNT\_CASH | NUMERIC(17,2) |  |
| MIN\_AMNT\_OTHERS | NUMERIC(17,2) |  |
| MIN\_AMNT\_PRCH | NUMERIC(17,2) |  |
| MIN\_AMNT\_OVERLIMIT\_CASH | NUMERIC(17,2) |  |
| MIN\_AMNT\_OVERLIMIT\_PRCH | NUMERIC(17,2) |  |
| TOTAL\_AMNT | NUMERIC(17,2) |  |
| C\_LIMIT | NUMERIC(17,2) |  |
| CC\_LMT\_UTL | BYTEINT | Credit Card limit utilization |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| OVD\_LMT | INTEGER | Overdraft limit |
| OVD\_LMT\_UTL | BYTEINT | Overdraft limit utilization |
| TOT\_SPNT\_AMNT | DEC(18,2) | The amount for total transactions in the current statement |
| TOT\_TRX\_CNT | DEC(18,2) | The count of total transaction for the current statement |
| DAYS\_USD | INTEGER | Count of days was the credit card used |
| TOT\_EXPSR\_TEB | DEC(18,2) | Total amount for credit products enrolled by the customer |
| TOT\_ASST\_TEB | DEC(18,2) | Total assets of the customer that reside in TEB |
| DEP\_TEB | DEC(18,2) | Sum of amount in deposit accounts |
| MIN\_PMT\_DUE | DEC(18,2) | Minimum payment due |
| INSTLMNT | BYTEINT | Installments left on the Credit card |
| INTRST\_AMNT | DEC(18,2) | Interest amount |
| INTRST\_FEE\_AMNT | DEC(18,2) | Fees related to interest amount |
| CHRGS\_AMT | DEC(18,2) | Charges related in interest amount |
| PRNCPL\_AMT | DEC(18,2) | Principal amount |

Table 9 – STMNT\_INFO\_CR

### Extraction Logic

Followings steps represent extraction logic:

* For all the customers in CUSTOMER\_BASE\_CR table, for each card and statement month, collect following information
  + Statement date
  + Payment due date
  + Last used date of the card. If the card was not used in the current statement month, the field will be NULL.
  + Credit Card limit
  + Credit Card limit utilization
  + Overdraft limit of the account
  + Overdraft limit utilization
  + Minimum payment due

Information/ rules for all these can be found in the lookup tables

* For each unique combination of above mentioned fields, calculate the following
  + Total spend amount in the statement month
  + Total transactions in the statement month
  + Number of days the card was used
  + Total amount of credit products (Credit Card limit, Loan amount, Mortgage amount)
  + Total Asset at TEB (Deposit, Savings, Investments, Mutual funds)
  + Deposits at TEB
  + Number of installments left on the credit card
  + Interest amount
  + Interest fee
  + Interest charges
  + Principal amount

Note that all information should be as of run-date.

### Grain

First 3 columns will form the grain for this table. We need a separate row for each unique combination of these 3 columns.

### Primary index

CARD\_ID

## Product Target Table

Ready for Loans .

Reference table for PROD\_KEY is created.

### Table name

PRODUCT\_TARGET\_CREDIT

### Attributes

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| CUST\_ID | VARCHAR(20) | Unique identifier for each TEB customer who has an account or is availing TEB's products |
| PROD\_KEY | BYTEINT | Refer to the Product lookup table |
| PRODUCT\_CLASS | VARCHAR(39) | Type of the loan product |
| DT(due\_date) | DATE | Daily date |
| DAYS\_PST\_DUE | BYTEINT | Days past due. Increment everyday if the minimum payment has not been paid by the due date |
| PMT\_AMT | DEC(18,2) | Amount paid on the particular day |
| DUE\_AMT | DEC(18,2) | Amount due for the month |
| MIN\_DUE\_PMT | DEC(18,2) | Minimum due amount(Not ready) |
| Payment\_No | NUMERIC(20,0) | Unique identifier for each payment |
| Credit\_No | NUMERIC(8,0) | Unique identifier for each credit |
| Payment\_Status | VARCHAR(20) | Status of payment (Paid, UnPaid, etc.) |
| Principal\_Amt | NUMERIC(17,2) | Amount of principal capital |
| Amt\_of\_Int | NUMERIC(17,2) | Amount of interest |
| Due\_Date | TIMESTAMP (without time zone) | Last payment day of installment |
| Branch\_No | NUMERIC(4,0) | Branch number of credit account |
| Trx\_Date | TIMESTAMP (without time zone) | Payment day |
| Trx\_Exc\_Rate | NUMERIC(14,5) | Exchange rate on the payment day |
| Payment\_Type | VARCHAR(50) |  |
| Def\_Int\_Amt | NUMERIC(28,6) | Amount of default interest |
| Cancellation\_Status | VARCHAR(1) | Status of cancellation |
| Cancellation\_Date | TIMESTAMP (without time zone) | Cancellation Day |
| Bank\_Product\_Type\_ID | NUMERIC(8,0) | Credit type ID |
| Product\_ID | NUMERIC(8,0) | Unique identifier for Credit Account No |
| Account\_No | NUMERIC(8,0) | Credit Account No |
| Credit\_Status | NUMERIC(1,0) | Refer to the Credit Status lookup table |
| Total\_Paid\_Installment | NUMERIC(3,0) | Number of paid installments |
| Total\_Remaining\_Installment | NUMERIC(3,0) | Number of remaning installments |
| Early\_Paid\_Installment | NUMERIC(3,0) | Early paid installments |
| Final\_Installment\_Due\_Date | TIMESTAMP (without time zone) |  |
| Early\_Payment\_Date | TIMESTAMP (without time zone) | Early Closed credits payment date |
| Worst\_Payment\_Perfomance | NUMERIC(2,0) | Worst payment performance of the related credit |
| Recent\_Payment\_Perfomance | NUMERIC(2,0) | Recent payment performance of the related credit |
| Total\_Default\_Payment | NUMERIC(3,0) | Number of default payments |
| Str\_Default | VARCHAR(400) | String identifier for default payments |
| Total\_Remaining\_Principal\_Amt | NUMERIC(17,2) | Remaining amount of principal capital of credit |
| Total\_Paid\_Installment\_Amt | NUMERIC(17,2) | Amount of paid installments |
| Remaining\_Installment\_Amt | NUMERIC(17,2) | Amount of remaining installments |
| Number\_of\_Remaining\_Installment | NUMERIC(3,0) | Number of remaining installments |
| Total\_Installment\_Amt | NUMERIC(17,2) | Amount of total installments |
| Monthly\_Installment\_Amt | NUMERIC(17,2) | Monthly amount of installment |
| Currency\_Code | VARCHAR(3) | Currency Code |
| Openning\_Date | TIMESTAMP (without time zone) | Openning day of credit |
| Closing \_Date | TIMESTAMP (without time zone) | Closing day of credit |
| Follow\_Up\_Status | VARCHAR(1) | Follow Up status |
| New\_Customer\_Flag | VARCHAR(1) |  |
| Follow\_Up\_Date | TIMESTMAP (without time zone) | Follow Up Day |
| Follow\_Up\_Principal\_Amt | NUMERIC(13,8) | Amount of principal capital of follow up credit |
| Used\_Interest\_Rate | NUMERIC(7,4) |  |
| Current\_Interest\_Rate | NUMERIC(7,4) |  |

Table 10 – PROD\_TGT\_CR

### Extraction Logic

Followings steps represent extraction logic:

* For all the customer in CUSTOMER\_BASE\_CR table, for each product, collect following information
  + Date (Every day of the year)
  + Days past due
  + Amount of payment made
  + Payment due
  + Minimum due payment for the month

Information/ rules for all these can be found in the lookup tables

Note that all information should be as of run-date.

### Grain

First 3 columns will form the grain for this table. We need a separate row for each unique combination of these 3 columns. It is a daily snapshot table.

### Primary index

CUST\_ID

## Bureau Data

We created view instead of table. You can find all columns in 09\_01\_Bureau\_Data excel in 00\_Reference folder.

If you filter SAS\_OUTPUT in Category column, you find Behavior score of the risk model ☺ (I guess)

If you filter KKB, you find bureau data variables.

### Table name

BUREAU\_CR

### Attributes

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| CUST\_ID | VARCHAR(20) | Unique identifier for each TEB customer who has an account or is availing TEB’s products |
| CARD\_LMT | DEC(18,2) | Aggregation of credit card limit for all banks |
| TOT\_EXPSR | DEC(18,2) | Aggregation of credit product from all banks |
| RSK\_SCORE | BYTEINT | Risk score |
| LOAN\_CNT | BYTEINT | Count loans from all banks (doesn’t exist) |
| PMT\_PRFMNC | BYTEINT | Payment performance |

Table 11 – BUREAU\_CR

### Extraction Logic

Followings steps represent extraction logic:

* For each customer, collect the following information
  + Card limit overall
  + Total exposure overall
  + Risk score
  + Count of all loans from all the banks against the customer
  + Payment performance

All information in this table will be from the perspective of the Bureau.

Note that all information might not be as of run-date.

### Grain

First column will form the grain for this table. We need a separate row for each customer.

### Primary index

CUST\_ID

## Employment Information

Table created with stated columns below.

### Table name

EMP\_INFO

### Attributes

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| EMP\_ID | BYTEINT | Refer to the lookup table |
| CREDIT\_SC | BYTEINT | Credit score |
| SALARY\_ACCT\_CNT | INTEGER | Number of salary account in TEB for the employer |
| AVG\_SALARY | DEC(18,2) | Average salary that the customer gives out |
| TOP10PCT\_AVG\_SALARY | DEC(18,2) | Average salary for the top 10 percent paid employees |
| TOP10PCT\_MED\_SALARY | DEC(18,2) | Median salary for the top 10 percent paid employees |

Table 12 – EMPLMNT\_CR

### Extraction Logic

Followings steps represent extraction logic:

* For each employer, collect following information
  + Credit score
  + Number of salary accounts
  + Average salary
  + Average of the top 10 percent of the salaries
  + Median of the top 10 percent of the salaries

Note that all information should be as of run-date.

### Grain

First column will form the grain for this table. We need a separate row for each employer ID.

### Primary index

EMPLYR\_ID

## Model Result Table

### Table name

STAGING.POV\_DWD\_SAS\_KY\_OPT\_SCORE

### Attributes

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| TARIH\_ID | DATE | Date when the model was run |
| CMUSNO | VARCHAR(20) | Unique identifier for each TEB customer who has an account or is availing TEB's products |
| PRODUCT\_ID | BYTEINT | Refer to the Model lookup table REF\_MODEL\_RESULT |
| OFFERCODE | BYTEINT | ID of the model |
| SCORE | BYTEINT | Score of the customer |

Table 13 – MDL\_RSLT\_CR

### Extraction Logic

Followings steps represent extraction logic:

* For each customer and model, collect following information
  + PRODUCT\_ID : Model name
  + Date when the model is run
  + Score of the customer for the model
* You can find model type information in REF\_MODEL\_RESULT table. We need to a colums on this table for product grouping

You can find reference excel “11\_Model\_Result\_Ref” in O:\bbank\03\_CRM\_BOLUMU\001\_Analitik\_CRM\17\_Big\_Data\20\_PreDefaultRisk\00\_Reference folder.

Note that all information should be as of run-date.

### Primary index

CUST\_ID

## Employment Employee Information

Tax is N/A. Salary amount is divided into 5 brackets (at 2500 TL intervals)

### Table name

RISK.EMPLOYER\_EMPLOYEE

### Attributes

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| EMPLYR\_KEY | BYTEINT | Refer to the lookup EMPLYR |
| CUST\_ID | VARCHAR(20) | Unique identifier for each TEB customer who has an account or is availing TEB's products |
| YR\_MNTH | DATE | Year month (First day of month) |
| SALARY\_AMT | DEC(18,2) | Salary of the customer |
| TAX | DEC(2,2) | Tax rate |

Table 14 - EMPLMNT\_EMPLYE\_CR

### Extraction Logic

Followings steps represent extraction logic:

* For each customer, collect following information
  + Employer name
  + Year month
  + Salary paid
  + Tax paid by customer (in terms of %)

Note that all information should be as of run-date.

### Grain

First 2 columns will form the grain for this table. We need a separate row for each unique combination of these 2 columns.

### Primary index

CUST\_ID

## Money Transfer

Not Ready.

### Table name

MNY\_TRNSFR\_CR

### Attributes

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| cust\_id | NUMERIC(8,0) | Unique identifier for each TEB customer sent or received the money |
| Trn\_no | NUMERIC(20,0) | Unique identifier of money transfer transaction |
| Trn\_date | DATE | Date of transaction |
| SNDR\_TEB\_CUST\_FLG | VARCHAR(1) | Flag to signify if the receiver is a TEB customer(could not be created) |
| RCVR\_TEB\_CUST\_FLG | VARCHAR(1) | Flag to signify if the sender is a TEB customer(could not be created) |
| TYPE\_OF\_TRANSFER | BYTEINT | The type of transfer i.e. inter-bank, intra-bank etc. |
| TYPE\_OF\_TRNSFR\_KEY | BYTEINT | Monthly, Rental, Educational Payment etc. |
| Trn\_amt | NUMERIC(17,2) | Amount transferred |
| currency\_code | VARCHAR(3) | Currency code of the transaction |
| counter\_bankname | VARCHAR(75) | For EFT’s the field shows the incoming or outgoing bank |
| rec\_acc\_no | NUMERIC(8,0) | Account number of the reciever |
| sndr\_acc\_no | NUMERIC(8,0) | Account number of the sender |
| TRN\_PRD | VARCHAR(16) | Used product(account or credit card) for transaction |
| periodic\_payment\_yn | VARCHAR(1) | The trx is periodic or not flag |
| salary\_payment\_yn | VARCHAR(1) | The trx is salary payment or not flag |
| lost\_eft\_yn | VARCHAR(1) | For outgoing EFT’s the field shows the receiver side is a TEB customer or not |
| coll\_payment\_yn | VARCHAR(1) | Shows whether the transaction is collective or not |

Table 15 – MNY\_TRNSFR\_CR

### Extraction Logic

Followings steps represent extraction logic:

* For each money transfer activity, collect following information
  + ID of the sender
  + ID of the receiver
  + Date of transaction
  + Sender flag to be ‘Y’ if the entity is a TEB Customer
  + Receiver flag to be ‘Y’ if the entity is a TEB Customer
  + Type of payment
  + Type of transfer
* For each unique combination of above mentioned fields, calculate the following
  + Amount of transfer

Note that all information should be as of run-date.

### Grain

First 3 columns will form the grain for this table. We need a separate row for each unique combination of these 3 columns.

### Primary index

SNDR\_ID

### Secondary index

RCVR\_ID

## Overdraft Status

Table created with stated columns below.

### Table name

RISK.OVD\_STS\_CR\_DAILY (daily based)- Variables pointed with D

RISK.OVD\_STS\_CR (monthly based)-Variables pointed with M

Attributes ler değiştirildi

### Attributes

ovd\_lmt\_utl\_avg\_monthly numeric(22,2)

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| CUST\_ID (D,M) | numeric(8,0) | Unique identifier for each TEB customer who has an account or is availing TEB's products |
| YR\_MNTH (D,M) | DATE | Year month |
| OVD\_LMT (M) | numeric(22,2) | Overdraft limit |
| ovd\_lmt\_utl\_avg\_monthly (M) | numeric(22,2) | Monthly average of overdraft |
| OVD\_ DAILY\_BLNC (D) | NUMERIC(28,6) | Overdraft balance |
| STATUS (D) | VARCHAR(1) | OPEN ACCOUNT: A |
| OVD\_LMT\_UTL (M) | numeric(22,2) | Overdraft limit utilization |

Table 16 – OVD\_STS\_CR

### Extraction Logic

Followings steps represent extraction logic:

* For each customer, collect following information
  + Year month
  + Account (not available)
  + Overdraft limit
  + Overdraft limit utilization

Note that all information should be as of run-date.

### Grain

First 2 columns will form the grain for this table. We need a separate row for each unique combination of these 2 columns.

### Primary index

CUST\_ID

## Insurance Information

Table created with stated columns below.

### Table name

RISK.INSR\_INFO\_CR

### Attributes

Attributes ler değiştirildi

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| CUST\_ID | numeric(8,0), | Unique identifier for each TEB customer who has an account or is availing TEB's products |
| INSR\_TYPE\_KEY | character varying(254), | Insurance type code |
| DT | timestamp without time zone | Daily date for each insurance type for the customer |
| AMT\_DUE\_DT | timestamp without time zone | Due Date for the insurance |
| dt\_of\_issue | timestamp without time zone | Policy creation date |
| Policy\_no | numeric | Policy number |
| pol\_amt\_dt | timestamp without time zone | Minimum payment amount |
| PMT\_AMT | numeric | total amount |
| paid\_pmt\_amt | numeric | paid amount |
| pmt\_due\_amt | numeric | remaining amount |
| pol\_stat | character | policy status (new:YENI, renewYENILEME, changed:ZEYL, cancelled:IPTAL) |
| PMT\_DUE\_AMT | DEC(18,2) | The payment amount due |

Table 17 – INSR\_INFO\_CR

### Extraction Logic

Followings steps represent extraction logic:

* For each customer, type of insurance and day, collect following information
  + Amount paid/Payment amount
  + Payment amounts due

Note that all information should be as of run-date.

### Grain

First 3 columns will form the grain for this table. We need a separate row for each unique combination of these 3 columns. This will be a daily snapshot table.

### Primary index

CUST\_ID

## Restructured Payments

Creating. Perşembe gunu gizem değiştirecek

### Table name

RSTRD\_PMT\_CR

### Attributes

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| CUST\_ID | VARCHAR(20) | Unique identifier for each TEB customer who has an account or is availing TEB’s products |
| PROD\_KEY | BYTEINT | Refer to the lookup table PROD |
| RSTRD\_DT | DATE | The restructuring date |
| YR\_MNTH | DATE | The month in which the payments against restructured amounts paid |
| RSTRD\_AMT | DEC(18,2) | The restructured amount |
| OUTSTD\_AMT | DEC(18,2) | The amount which is left to be paid in that particular month |
| PAID\_AMT | DEC(18,2) | The amount paid in that month |
| AMT\_DUE | DEC(18,2) | The amount that was due to be paid after restructuring |
| ORIG\_AMT | DEC(18,2) | The original amount which was later restructured |

Table 18 – RSTRD\_PMT\_CR

### Extraction Logic

Followings steps represent extraction logic:

* For each customer and credit product, collect following information
  + Restructuring date
  + Month of the year in which the payment is made after restructuring
  + Restructured amount
  + Outstanding amount
  + Amount paid
  + Amount due
  + Original amount

Note that all information should be as of run-date.

### Grain

First 4 columns will form the grain for this table. We need a separate row for each unique combination of these 4 columns.

### Primary index

CUST\_ID

## Product Ownership

Ready for loan products. Other products (cc, deposit) will be added.

Ready for loan and assets. Asset’lerin count’uyla ilgili sıkıntı var. anıl baksın der gizem

Reference table in ASTER : "risk"."ref\_bankaurunid\_updated\_v2"

### Table name

PROD\_OWNSHP

### Attributes

Data type kontrol et

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| CUST\_ID | VARCHAR(20) | Unique identifier for each TEB customer who has an account or is availing TEB’s products |
| PROD\_KEY | BYTEINT | Refer to the lookup table PROD |
| YR\_MNTH | DATE | Year Month |
| BALNC | DEC(18,2) | Balance of the credit product left to be paid |
| PROD\_CNT | INTEGER | Count of products |

Table 19 – PROD\_OWNSHP

### Extraction Logic

Followings steps represent extraction logic:

* For each customer and credit product, collect following information
  + Year Month
  + Balance
* For each unique combination of above mentioned fields, including product, calculate the following
  + Count of Products

Note that all information should be as of run-date.

### Grain

First 3 columns will form the grain for this table. We need a separate row for each unique combination of these 3 columns. This is a monthly table(historical).

### Primary index

CUST\_ID

## Account Status

Ready for specified products and customers’ state (live,closed,cancelled,target).

Customer’s State Variable : D\_REF\_MUS\_DRM

FILTER:

Customer\_type = Real and Business\_Line in (Individual&Star) or Portfolio\_Type in (Individual&Star)

These filters shows that customer is in target population which is real individual customers.

Hangi ürünler için konulduğunun excel’i var benim yaptığım. O İngilizceye çevrilip içeri koyulcak

### Table name

ACCNT\_STS

### Attributes

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| CUST\_ID | VARCHAR(20) | Unique identifier for each TEB customer who has an account or is availing TEB’s products |
| PROD\_NAME | character varying(100) | Refer to the lookup table PROD |
| tarih\_id | DATE | Year Month |
| ownership | integer | Ownership of product as 1 or 0 |
| activity | integer | Activity of products as 1 or 0 |
| STS\_KEY | BYTEINT | Refer to lookup table STS |

Table 20 – ACCNT\_STS

### Extraction Logic

Followings steps represent extraction logic:

* For each customer and credit product, collect following information
  + Year Month
  + Account Status

Note that all information should be as of run-date.

### Grain

First 3 columns will form the grain for this table. We need a separate row for each unique combination of these 3 columns. This is a historical table, it will only capture the transition of status against the customer and product, whenever it occurs.

### Primary index

CUST\_ID

## Customer Interaction

Discussed with Usman and decided not to create it. We have some channel interaction events in zeren.event\_list table.

### Table name

CUST\_INTRACTN

### Attributes

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| CUST\_ID | VARCHAR(20) | Unique identifier for each TEB customer who has an account or is availing TEB’s products |
| INTRACTN\_DATE | DATE | Interaction Date |
| CHNL\_KEY | BYTEINT | Refer to lookup table CHNL |
| INTRACTN\_RSN\_KEY | BYTEINT | Refer to INTRACTN\_RSN lookup table |
| LOC\_KEY | BYTEINT | Refer to LOC lookup table |
| INTRACTN\_CNT | INTEGER | Count of interaction |

Table 21 – CUST\_INTRACTN

### Extraction Logic

Followings steps represent extraction logic:

* For each customer, collect following information
  + Interaction Date
  + Channel for Interaction e.g. ATM, Phone, SMS, Branch
  + Interaction Reason e.g. Complaint, Info, Withdrawal
  + Location of interaction
* For each unique combination of above mentioned fields, calculate the following
  + Count of Interactions

Note that all information should be as of run-date.

### Grain

First 4 columns will form the grain for this table. We need a separate row for each unique combination of these 4 columns. This is a transactional table.

### Primary index

CUST\_ID

## Customer Events

You can find event list in the excel “Event\_List” in the folder below:

O:\bbank\03\_CRM\_BOLUMU\001\_Analitik\_CRM\17\_Big\_Data\19\_EVENT\_CHECK

### Table name

ZEREN.EVENT\_LIST

### Attributes

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| cmusno | VARCHAR(20) | cmusno |
| event\_id | VARCHAR(5) | Unique event id (5 digits) |
| event\_date | DATETIME | Event Date in timestamp format. If there is no time info, 00:00:00 is used. |
| domain\_type | VARCHAR(30) | Related product or service info. |
| event1 | VARCHAR(90) | Event name.(ex: credit card open) |
| event2 | VARCHAR(90) | Event detail.(ex: supplementary credit card open) |
| event3 | VARCHAR(90) | Necessity of this field will be decided later. |
| channel\_name | VARCHAR(90) | Channel information if exists. |

Table 22 – CUST\_EVENT

### Extraction Logic

Followings steps represent extraction logic:

* For each customer, collect following information
  + Event Domain
  + Event Name
  + Event Detail
  + Channel Name

Note that all information should be as of run-date.

### Grain

First 4 columns will form the grain for this table. We need a separate row for each unique combination of these 4 columns. This is a transactional table.

### Primary index

CMUSNO

## Lookup Tables

Following are the lookup tables

### MERCH

|  |  |  |
| --- | --- | --- |
| **MERCH\_KEY** | **MERCH\_NM** | **DESC** |
| 1 |  |  |
| 2 |  |  |

Table 23 - MERCH

### CURR

|  |  |  |
| --- | --- | --- |
| **CURR\_KEY** | **CURR\_NM** | **DESC** |
| 1 |  |  |
| 2 |  |  |

Table 24 - CURR

### LOC

|  |  |  |
| --- | --- | --- |
| **LOC\_KEY** | **LOC\_NM** | **DESC** |
| 1 |  |  |
| 2 |  |  |

Table 25 - LOC

### CARD\_TYPE

|  |  |  |
| --- | --- | --- |
| **CARD\_TYPE\_KEY** | **CARD\_TYPE\_NAME** | **DESC** |
| 1 |  |  |
| 2 |  |  |

Table 26– CARD\_TYPE

### CUST\_TYPE

|  |  |  |
| --- | --- | --- |
| **CUST\_TYPE\_KEY** | **CUST\_TYPE\_NM** | **DESC** |
| 1 |  |  |
| 2 |  |  |

Table 27 – CUST\_TYPE

### CARD\_SEG

|  |  |  |
| --- | --- | --- |
| **CARD\_SEG\_KEY** | **CARD\_SEG\_NM** | **DESC** |
| 1 |  |  |
| 2 |  |  |

Table 28– CARD\_SEG

### OCCUP

|  |  |  |
| --- | --- | --- |
| **OCCUP\_KEY** | **OCCUP\_NM** | **DESC** |
| 1 |  |  |
| 2 |  |  |

Table 29 - OCCUP

### EMPLNT

|  |  |  |
| --- | --- | --- |
| **EMPLNT\_KEY** | **EMPLNT\_NM** | **DESCR** |
| 1 |  |  |
| 2 |  |  |

Table 30 - EMPNT

### CITY

|  |  |  |
| --- | --- | --- |
| **CITY\_KEY** | **CITY\_NM** | **DESCR** |
| 1 |  |  |
| 2 |  |  |

Table 31 - CITY

### DIST

|  |  |  |
| --- | --- | --- |
| **DIST\_KEY** | **DIST\_NM** | **DESC** |
| 1 |  |  |
| 2 |  |  |

Table 32 - DIST

### MN\_BRNCH

|  |  |  |
| --- | --- | --- |
| **MN\_BRNCH\_KEY** | **MN\_BRNCH\_NM** | **DESC** |
| 1 |  |  |
| 2 |  |  |

Table 33 – MN\_BRNCH

### PROD

|  |  |  |
| --- | --- | --- |
| **PROD\_KEY** | **PROD\_NM** | **DESC** |
| 1 |  |  |
| 2 |  |  |

Table 34 - PROD

### MDL\_TYPE

|  |  |  |
| --- | --- | --- |
| **MDL\_TYPE\_KEY** | **MDL\_TYPE\_NM** | **DESC** |
| 1 |  |  |
| 2 |  |  |

Table 35 - MDL\_TYPE

### TYPE\_OF\_PMT

|  |  |  |
| --- | --- | --- |
| **TYPE\_OF\_PMT\_KEY** | **TYPE\_OF\_PMT\_NM** | **DESC** |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |

Table 36 – TYPE\_OF\_PMT

### TYPE\_OF\_TRNSFR

|  |  |  |
| --- | --- | --- |
| **TYPE\_OF\_TRNSFR\_KEY** | **TYPE\_OF\_TRNSFR\_NM** | **DESC** |
| 1 |  |  |
| 2 |  |  |

Table 37 – TYPE\_OF\_TRNSFR

### INSR\_TYPE

|  |  |  |
| --- | --- | --- |
| **INSR\_TYPE\_KEY** | **INSR\_TYPE\_NM** | **DESC** |
| 1 |  |  |
| 2 |  |  |

Table 38 – INSR\_TYPE

### STS

|  |  |  |
| --- | --- | --- |
| **STS\_KEY** | **STS\_NM** | **DESC** |
| 1 |  |  |
| 2 |  |  |
|  |  |  |

Table 39 – STS

### CHNL

|  |  |  |
| --- | --- | --- |
| **CHNL\_KEY** | **CHNL\_NM** | **DESC** |
| 1 |  |  |
| 2 |  |  |

Table 40 – CHNL

### INTRACTN\_RSN

|  |  |  |
| --- | --- | --- |
| **INTRACTN\_RSN\_KEY** | **INTRACTN\_RSN \_NM** | **DESC** |
| 1 |  |  |
| 2 |  |  |

Table 41 – INTRACTN\_RSN