İçerik

[A. Modellerin Otomasyonu Sırasında Alınan Genel Hatalar 2](#_Toc465932730)

[1. Yetersiz Disk Alanı Hataları 2](#_Toc465932731)

[2. Syntax Hataları 2](#_Toc465932732)

[a. Kapatılmayan Parantez 2](#_Toc465932733)

[b. Degisken Adinda İzin Verilmeyen Karakter Kullanılması 3](#_Toc465932734)

[c. Virgül Kullanımının Unutulması ya da Yanlışlıkla Virgülün Silinmesi 3](#_Toc465932735)

[d. Değişken İsminde Boşluk Bulunması 4](#_Toc465932736)

[3. Akışın Yeniden Düzenlenmesi ile İlgili Hatalar 5](#_Toc465932737)

[a. Input Olarak Kullanılan Tablonun Bulunanamaması 5](#_Toc465932738)

[b. İstenilen Değişkenin Belirtilen İnput Tabloda Bulunanaması 6](#_Toc465932739)

[4. Kullanılan Makrolar ile İlgili Hatalar 6](#_Toc465932740)

[a. Makronun Oluşturulmamış Olması ya da Makro İsminin Yanlış Belirtilmiş Olması 6](#_Toc465932741)

[b. Formatın Yanlış Belirtilmesi 7](#_Toc465932742)

[c. Macro-Mend İçerisinde Makro Oluşturulamaması 8](#_Toc465932743)

[d. Split Columns Prosedürünün Kullanılmasından Kaynaklanan Hatalar 9](#_Toc465932744)

[5. Appendix 13](#_Toc465932745)

# Modellerin Otomasyonu Sırasında Alınan Genel Hatalar

Bu döküman, modellerin otomatize edilmesi amacıyla yeniden düzenlenerek tek akış haline getirilmesi ve gerekli yerlerde makro değişkenler oluşturularak akışın içerisine yerleştirilmesi sırasında yapılan genel hataları örnekler ile göstermek ve bu hatalara çözüm aranırken kolaylık sağlamak amacıyla oluşturulmuştur. Ayrıca arkaplanda çalıştırılması için otomatizasyonu sağlanan akışlar, kullanıcı manuel olarak kendi profilinde çalıştırdığında hata almasa bile, sistem farklılıklarından dolayı arkaplanda çalıştırıldığında hata verebilmektedir. Bu nedenle akışların arkaplanda çalıştırılması sırasında alınan genel hatalara da dökümanda yer verilmiştir.

## Yetersiz Disk Alanı Hataları

Kullandiginiz libraryde yeterli alan olmamasi durumunda aldiginiz hatalardir. Genelde icinde ‘insufficient space’ tarzi bir ifade bulunur. Uyariyi farkli sekillerde verebilir. İki örneğini aşağıda görebilirsiniz.

|  |
| --- |
| ERROR: Insufficient space in file WORK.CARD\_SGLK\_ISLM\_ADET.DATA. |
| ERROR: File WORK.CARD\_SGLK\_ISLM\_ADET.DATA is damaged. I/O processing did not complete. |
|  |
| ERROR: Insufficient space in file WORK.'SASTMP-000000028'n.UTILITY. |
| ERROR: File WORK.'SASTMP-000000028'n.UTILITY is damaged. I/O processing did not complete. |

## Syntax Hataları

Yazim hatalarindan kaynaklanır. Kapatılmayan parantez, değişkenler arasında virgül kullanılması gereken yerlerde virgülün kullanılmaması, kullanılan statementların sonuna ‘;’ konulmasının unutulması, degisken ve tablo isimlerinde kabul edilmeyen karakterlerin (nokta gibi) kullanılması gibi sebeplerden bu hatayı verir. Genellikle ‘Syntax Error’ ifadesi bulunur.

Ek olarak, kendi kullanıcınız ile bir kodu çaılştırırken hata almasanız da değişken isimlerinde kullanılan türkçe karakterler ve boşluklar aynı akış arkaplanda sasschedule user’ı üzerinden çalıştırıldığında hata verir.

### Kapatılmayan Parantez

|  |
| --- |
| 32               WHERE (t1.SPR\_REF\_NO = t2.SPR\_REF\_NO AND t1.MUST\_NO = t3.MUST\_NO) AND (t1.SPR\_DRM\_KOD = 'GOT' AND t1.SPR\_YIL IN |
| 33                    ( |
| 34                    &CRY\_1. |
| 35                    ) AND t1.SPR\_AY IN |
| 36                    ( |
| 37                    '06', |
| 38                    '09', |
| 39                    '12', |
| 40                    '03' |
| 41                    ) AND t3.FINSOFT\_MUST\_NO NOT IS MISSING AND ( t3.FINSOFT\_MUST\_NO > 0 or t3.FINSOFT\_MUST\_NO <0) |
| 42               GROUP BY t3.FINSOFT\_MUST\_NO, |
| \_\_\_\_\_ |
| 79 |
| ERROR 79-322: Expecting a ). |
|  |
| 43                        t1.SPR\_YIL, |
| 44                   t1.SPR\_AY; |

### Degisken Adinda İzin Verilmeyen Karakter Kullanılması

|  |
| --- |
| 41                   t2.Ihracat, |
| 42                   t3.Teminat\_Mektubu, |
| 43                   t3.DBS\_Taahhut\_Riski, |
| 44                   t3.Diger\_Gar.\_Ve\_Kefaletler, |
| \_ |
| 22 |
| 76 |
| ERROR 22-322: Syntax error, expecting one of the following: a quoted string, !, !!, &, \*, \*\*, +, ',', -, /, <, <=, <>, =, >, >=, ?, |
| AND, AS, BETWEEN, CONTAINS, EQ, EQT, FORMAT, FROM, GE, GET, GT, GTT, IN, INFORMAT, INTO, IS, LABEL, LE, LEN, LENGTH, |
| LET, LIKE, LT, LTT, NE, NET, NOT, NOTIN, OR, TRANSCODE, ^, ^=, |, ||, ~, ~=. |
|  |
| ERROR 76-322: Syntax error, statement will be ignored. |
|  |
| 45                   t3.Akreditif, |
| 46                   t3.Kabul\_Kredi, |

### Virgül Kullanımının Unutulması ya da Yanlışlıkla Virgülün Silinmesi

|  |
| --- |
| 31 t1.KREDI\_TARIHI, |
| 32 t1.FAIZ\_BULMA\_GUNU, |
| 33 t1.SEKTOR\_KODU |
| 34 t2.SEKTOR\_ADI LABEL="Sektor\_Kod\_Aciklamalari" AS Sektor\_Kod\_Aciklamalari, |
| \_\_ |
| 22 |
| 76 |
| ERROR 22-322: Syntax error, expecting one of the following: !, !!, &, \*, \*\*, +, ',', -, /, <, <=, <>, =, >, >=, ?, AND, BETWEEN, |
| CONTAINS, EQ, EQT, GE, GET, GT, GTT, LE, LET, LIKE, LT, LTT, NE, NET, OR, ^=, |, ||, ~=. |
|  |
| ERROR 76-322: Syntax error, statement will be ignored. |
|  |
| 35 t1.DURUM\_KOD, |
| 36 t1.KREDI\_TUTARI, |
| 37 t1.KANAL, |

\*\*\* Bu örnekte de olduğu gibi sas bazı durumlarda hata verdiğinde bir alt satırı işaretlemiş olabilir. Görüldüğü gibi Sektor\_Kod\_Aciklamalari ile biten satırın sonunda virgül kullanılmıştır. Fakat SEKTOR\_KODU ile biten bir üst satırın sonunda virgül kullanılmıştır. Bu nedenle hata alındığı zaman altını üstünü de kontrol etmek faydalı olacaktır.

### Değişken İsminde Boşluk Bulunması

|  |
| --- |
| 1407 /\* SUM\_of\_SON\_LIMIT \*/ |
| 1408 (SUM(t1.SON\_LIMIT)) FORMAT=22. AS SUM\_of\_SON\_LIMIT, |
| 1409 /\* AVG DISTINCT\_of\_SON\_LIMIT \*/ |
| 1410 (AVG(DISTINCT(t1.SON\_LIMIT))) FORMAT=22. AS 'AVG DISTINCT\_of\_SON\_LIMIT'n, |
| 1411 /\* MAX\_of\_SON\_KARAR \*/ |
| 1412 (MAX(t1.SON\_KARAR)) AS MAX\_of\_SON\_KARAR, |
| 1413 /\* MIN\_of\_SON\_KARAR \*/ |
| 1414 (MIN(t1.SON\_KARAR)) AS MIN\_of\_SON\_KARAR, |
| 1415 /\* MAX\_of\_Basvuru\_tipi \*/ |
| 1416 (MAX(t1.Basvuru\_tipi\_aciklama)) AS MAX\_of\_Basvuru\_tipi, |
| 1417 /\* MIN\_of\_Basvuru\_tipi \*/ |
| 1418 (MIN(t1.Basvuru\_tipi\_aciklama)) AS MIN\_of\_Basvuru\_tipi |
| 1419 FROM WORK.QUERY\_FOR\_NY\_CREDITCARD\_APP t1 |
| 1420 WHERE t1.DONEM IN |
| 1421 ( |
| 1422 "&PM\_1u." |
| 1423 ) |
| 1424 GROUP BY t1.ILK7, |
| 1425 t1.DONEM |
| 1426 ORDER BY MAX\_of\_SON\_KARAR DESC; |
| ERROR: The value AVG DISTINCT\_of\_SON\_LIMIT is not a valid SAS name. |

\*\*\* Bu hatayı kendi kullanıcınız ile kodu çalıştırdığınızda almayabilirsiniz. Fakat nihai amaç akışların otomatize edilerek arkaplanda çalıştırılması olduğu için bu kısımları da düzenlemeliyiz. Değişken ismi tırnak içinden çıkarılarak boşluklar ‘\_’ ile değiştirilerbilir. Aynı işlem türkçe karakter içeren değişkenler için de yapılmalıdır.

**Önemli Not 1:** Burada ornek olarak kullanılan kod içinde hata veren değişken, “(AVG(DISTINCT(t1.SON\_LIMIT))) FORMAT=22. AS 'AVG DISTINCT\_of\_SON\_LIMIT'n”, sorgu içerisinde oluşturulan bir değişken. Yani sorgu içerisinde, input tablo (WORK.QUERY\_FOR\_NY\_CREDITCARD\_APP) içindeki ‘SON\_LIMIT’ değişkeni kullanılarak ‘AVG DISTINCT\_of\_SON\_LIMIT’ değişkeni türetilmiş. Dolayısıyla ismini istediğimiz gibi değiştirdiğimizde sıkıntı yaratmayacaktır. Fakat her zaman durum bu olmayabilir. Eğer türkçe karakter veya boşluk içeren karakter sorgu içerisinde türetilmemiş ve input tablodan direk geliyor ise izleyeceğimiz prosedür daha farklı olacak. O tarz bir durumda izlenecek prosedürü Appendix A kısmında bulabilirsiniz.

**Önemli Not 2:** Akış içerisinde bir sorguda bu şekilde isim değişikliği yaptığımızda, türetilen değişkenin ismini değiştirmiş oluyoruz. Bu örnek üzerinden gidersek, ‘AVG DISTINCT\_of\_SON\_LIMIT' yerine ‘AVG\_DISTINCT\_of\_SON\_LIMIT’ değişkenini oluşturmuş oluyoruz. Eğer akışın devamında aynı değişken kullanılıyorsa, yani başka bir sorguda bu kısımda uretilen tabloyu input tablo olarak kullanıyor ve ‘AVG DISTINCT\_of\_SON\_LIMIT' değişkenini çekiyor ise, biz adını değiştirdiğimizden dolayı değişkeni bulamayacak ve hata verecektir. Bu nedenle akışın devamındaki sorgularda da değişken ismini aynı şekilde değiştirmek zorundayız. Aksi takdirde, aşağıdaki şekilde bir hata alırız:

|  |
| --- |
| 1283       IFN(t1.‘AVG DISTINCT\_of\_SON\_LIMIT'n is missing,0,t1.'Döviz Kredileri'n,0), |
| 1284       IFN(t1.Dovize\_Endeksli is missing,0,t1.Dovize\_Endeksli,0) as Bakiye\_dvze\_kredi, |
| 1355       FROM WORK.MUSTERI\_SAHIPLIK\_5 t1; |
| ERROR: Column ‘AVG DISTINCT\_of\_SON\_LIMIT' could not be found in the table/view identified with the correlation name T1. |

## Akışın Yeniden Düzenlenmesi ile İlgili Hatalar

Bir projenin (.egp uzantılı dosya) tek akış (.sas) haline getirilmesi sırasında, projedeki sorguların doğru sırayla akış içerisine alınmamasından kaynaklanan hatalardır.

### Input Olarak Kullanılan Tablonun Bulunanamaması

|  |
| --- |
| 3850             FROM WORK.QUERY\_FOR\_MUSTERI\_ |
| 3851       t1 |
| 3852     LEFT JOIN WORK.POS\_HACIM t2 ON (t1.Yil = t2.Yil) AND (t1.Ay = t2.Ay) AND (t1.MUSTERI\_NO = t2.Customer); |
| ERROR: File WORK.QUERY\_FOR\_MUSTERI\_.DATA does not exist. |
| NOTE: PROC SQL set option NOEXEC and will continue to check the syntax of statements. |
| 3853       QUIT; |
| NOTE: The SAS System stopped processing this step because of errors. |
| NOTE: PROCEDURE SQL used (Total process time): |
| real time           0.07 seconds |
| cpu time            0.09 seconds |

\*\*\* Buradaki hata, sorgu içerisinde kullanılan input tablolardan WORK.QUERY\_FOR\_MUSTERI\_ tablosunun oluşturulmamış olmasıdır. Bunun farklı nedenleri olabilir:  
- Modeli proje ağacından(.egp) tek akışa çevirirken ilgili tablonun oluşturulduğu sorguyu akış içerisine almayı unutmak ya da o sorguyu hata aldığımız sorgunun ilerisine yerleştirmiş olmak. Bu noktada yapılması gereken akış içerisinde ‘QUERY\_FOR\_MUSTERI\_’ şeklinde aratmak eğer yok ise egp dosyasından tablonun oluştuğu sorguyu aramaktır.  
Not: Eğer akış içerisinde QUERY\_FOR\_MUSTERI\_ tablosunun oluşturulduğu sorgu varsa ve hata alınan sorgudan geride yer alıyorsa, akış çalıştırılırken o kısım çalıştırılmamış veya çalışmış ve hata vermiş ama farkedilmemiştir. Bu durumda yapılması gereken QUERY\_FOR\_MUSTERI\_ tablosunun oluşturulduğu sorgunun tekrar çalıştırılıp logunun dikkatlice incelenmesidir.  
- Eğer yukarıdaki adım problemi çözmüyorsa yanlışlıkla tablo adının değiştirildiği düşünülebilir. Mesela bu örnek üzerinden gidecek olursak, örneğin aslında QUERY\_FOR\_MUSTERI\_SAHIPLIK veya QUERY\_FOR\_MUSTERI\_&PM\_1u. vs seklinde olması geken tablo isminin sonu yanlışlıkla silinmiş olabilir. Dolayısıyla akış içerisinde kullanılan benzer isimdeki tablolar kontrol edilmelidir.

### İstenilen Değişkenin Belirtilen İnput Tabloda Bulunanaması

|  |
| --- |
| 2511 t3.CE\_LA\_Loan\_TotLoaned\_&PM\_1u. |
| 2512 FROM WORK.CUSTLIST\_LOANAPP\_F t1 |
| 2513 LEFT JOIN WORK.LOANAPPRENAME t2 ON (t1.MUS\_NO = t2.MUS\_NO) |
| 2514 LEFT JOIN WORK.F\_CUSTEXP\_03\_LOANAPP\_STD t3 ON (t1.MUS\_NO = t3.MUS\_NO); |
| 2516 QUIT; |
| ERROR: Column t3.CE\_LA\_Loan\_TotLoaned\_&PM\_1u. could not be found in the table/view identified with the correlation name T1. |

\*\*\* Buradaki hata T3 olarak belirtilen WORK.F\_CUSTEXP\_03\_LOANAPP\_STD tablosunda istenilen CE\_LA\_Loan\_TotLoaned\_&PM\_1u. değişkeninin bulunamamasıdır. Yapılması gereken ilk olarak belirtilen tablodan başlayarak akışın geriye doğru takip edilmesi ve ilgili değişkenin kontrol edilmesidir. Ayrıca hata değişken isminin yanlış yazılmasından da kaynaklanmış olabilir.

\*\*\*\*\* Bu hata ayın ilk günlerinde çalıştırılan kodlarda bazı kaynak tabloların henüz dolmamış olmasından da kaynaklanabilir. Örneğin Churn.Network\_Degree\_All ve Churn.Interest\_Rates geç dolmaktadır. Dolayısı ile değişken isminde periyod makrosu kullanılan durumlarda, genellikle bir ay öncesi kullanılıyor, istenilen değişken ayın ilk günlerinde daha eklenmemiş olur ve bu hata alınır.

## Kullanılan Makrolar ile İlgili Hatalar

Makroların oluşturulmamış olması, makro isminin yanlış yazılması, makro formatının yanlış belirtilmesi gibi sebeplerden kaynaklanabileceği gibi direk olarak makronun yanlış oluşturulmasından da kaynaklanabilir. Bu nedenle makro ile ilgili hata aldığımız zaman ilk olarak “%put &(makro-adı).;” şeklinde makronun doğruluğunu kontrol etmeliyiz.

### Makronun Oluşturulmamış Olması ya da Makro İsminin Yanlış Belirtilmiş Olması

|  |
| --- |
| 83 FROM O\_C\_OPR.BOSS\_ARAMAKOD t1 |
| 84 LEFT JOIN O\_C\_OPR.BOSS\_SEKTOR t2 ON (t1.SEKTOR\_KODU = t2.SEKTOR\_KOD) |
| 85 LEFT JOIN WORK.DURUM\_KOD t3 ON (t1.DURUM\_KOD = t3.Durum\_Kod) |
| 86 LEFT JOIN WORK.ACILAN\_ONONAYLI\_KREDILER\_F t4 ON (t1.TALEP\_NO = t4.TALEP\_NO) |
| 87 LEFT JOIN O\_C\_OPR.BOSS\_ADK\_BASVURU t5 ON (t1.TALEP\_NO = t5.TALEP\_NO) |
| 88 WHERE t1.RZ\_TARIHI BETWEEN "&M\_4B."dt AND "&M\_1EHE."dt AND t1.MUSTERI\_NO NOT IS MISSING |
| WARNING: Apparent symbolic reference M\_4B not resolved. |
| WARNING: Apparent symbolic reference M\_1EHE not resolved. |
| 89 ORDER BY t1.MUSTERI\_NO, |
| 90 t1.RZ\_TARIHI; |

\*\*\* Burada alınan hata, sasın makroyu tanımamasından kaynaklanmaktadır. Yani bu isimlerde makro akış içerisinde daha önce tanımlanmamıştır. Makroların oluşturulduğu kısım akış çalıştırılırken atlanmış olabilir ya da makro ismi yanlış yazılmış olabilir. Akış içerisinde makroların oluşturulduğu kısım kontrol edilmelidir.

### Formatın Yanlış Belirtilmesi

|  |
| --- |
| 25 if KST\_TRH="&M\_1Ed."dt; |
| NOTE: Line generated by the macro variable "M\_1ED". |
| 25 "30SEP2016 |
| \_\_\_\_\_\_\_\_\_\_ |
| 77 |
| ERROR: Invalid date/time/datetime constant "30SEP2016"dt. |
| ERROR 77-185: Invalid number conversion on "30SEP2016"dt. |

\*\*\* Burada alınan hata “30SEP2016” formatında oluşturulan &M\_1Ed makrosunun “&M\_1Ed.”dt şeklinde ‘date-time’ formatında kullanılmak istenmesinden kaynaklanmaktadır. Eğer ‘date-time’ formatında bir makro kullanılması gerekiyorsa ona göre yeni bir makro oluşturulmalı ve o kullanılmalıdır. Eğer ‘date’ formatında bir makro kullanılması yeterli ise hata alan kısım “&M\_1Ed.”d şeklinde değiştirilmelidir.

**Önemli Not:** Makroların oluşturulduğu kısımlar dikkatli incelenmelidir. Çünkü makro oluşturulurken genellikle hata vermez. Bu nedenle yanlış makro oluşturulduğu farkedilmez.

|  |
| --- |
| data \_null\_; |
|  |
| call symput ('PM1', put(INTNX('month', "&sysdate"d , 1, 'beginning'), yymmn6.)); |
| call symput ('PM0', put(INTNX('month', "&sysdate"d , 0, 'beginning'), yymmn6.)); |
| call symput ('PM\_1', put(INTNX('month', "&sysdate"d , -1, 'beginning'), yymmn6.)); |
| call symput ('PM\_2', put(INTNX('month', "&sysdate"d , -2, 'beginning'), yymmn6.)); |
| call symput ('PM\_3', put(INTNX('month', "&sysdate"d , -3, 'beginning'), yymmn6.)); |
| call symput ('PM\_4', put(INTNX('month', "&sysdate"d , -4, 'beginning'), yymm6.)); |
| call symput ('PM\_5', put(INTNX('month', "&sysdate"d , -5, 'beginning'), yymmn6.)); |
| call symput ('M\_4B', cat(put(INTNX('month', "&sysdate"d , -4, 'beginning'), DATE9.), ":00:00:00")); |
| call symput ('M\_1EHE', cat(put(INTNX('month', "&sysdate"d , -1, 'end'), DATE9.), ":23:59:59")); |
|  |
| run; |

Yukarıdaki kodun çalıştırılması durumunda logda yer alacak ifade şu şekildedir (Genelde makro tanımlanan yerlerde oluşacak log da buna benzerdir ve bize hiç bir bilgi vermez):

|  |
| --- |
| 23 data \_null\_; |
| 24 |
| 25 call symput ('PM1', put(INTNX('month', "&sysdate"d , 1, 'beginning'), yymmn6.)); |
| 26 call symput ('PM0', put(INTNX('month', "&sysdate"d , 0, 'beginning'), yymmn6.)); |
| 27 call symput ('PM\_1', put(INTNX('month', "&sysdate"d , -1, 'beginning'), yymmn6.)); |
| 28 call symput ('PM\_2', put(INTNX('month', "&sysdate"d , -2, 'beginning'), yymmn6.)); |
| 29 call symput ('PM\_3', put(INTNX('month', "&sysdate"d , -3, 'beginning'), yymmn6.)); |
| 30 call symput ('PM\_4', put(INTNX('month', "&sysdate"d , -4, 'beginning'), yymm6.)); |
| 31 call symput ('PM\_5', put(INTNX('month', "&sysdate"d , -5, 'beginning'), yymmn6.)); |
| 32 call symput ('M\_4B', cat(put(INTNX('month', "&sysdate"d , -4, 'beginning'), DATE9.), ":00:00:00")); |
| 33 call symput ('M\_1EHE', cat(put(INTNX('month', "&sysdate"d , -1, 'end'), DATE9.), ":23:59:59")); |
| 34 |
| 35 run; |
|  |
| NOTE: DATA statement used (Total process time): |
| real time 0.00 seconds |
| cpu time 0.00 seconds |

Fakat dikkatli bakıldığı takdirde görülebileceği gibi ‘PM\_4’ isminde olusturulan makronun formatı diğerlerinden farklı oluşturulmuştur ve istenilen ‘201606’ değeri yerine ‘16M06’ değerini verir. Bu örnek çok çeşitli şekillerde çoğaltılabilir. Makro oluşturulurken bu tarz durumlarda sas hata vermediği için dikkatli olunmalı ve oluşturulan her makronun doğruluğu “%put &(makro-adı).;” ile kontrol edilmelidir.

### Macro-Mend İçerisinde Makro Oluşturulamaması

|  |
| --- |
| **%macro imputeOutliers(data=, customer\_id=, out=, details=\_outlierdetails\_);** |
| %let name = %scan(&data, 1, '()'); |
| %let libname=%scan(&name, 1, '.'); |
| %let memname=%scan(&name, 2, '.'); |
|  |
| proc contents data=&data out=names(keep=name type format) noprint; |
| run; |
|  |
| proc sql noprint; |
| select name into :varnames seperated by " " |
| from names where |
| lowcase(name) ne lowcase("&customer\_id") |
| and format not like 'DATE%' and type = 1; |
| quit; |
| . |
| . |
| . |
| . |
| . |
| . |
| . |
| **%mend;** |
| **%imputeOutliers(data=WORK.BOND\_ALL, customer\_id=MUS\_NO, out=WORK.OUTLIER\_USED\_bond, details=\_detailsOUTLIER\_);** |

\*\*\* Bu ve benzeri durumlarda sasschedule kullanıcısı üzerinden arka planda çalıştırılan kodun logunda genellikle ‘No rows selected’ ifadesi görülür. Bu oluşturulmak istenen ‘varnames’ makrosuna herhangi bir değer tanımlanmadığı anlamına gelir ve akışın devamında bu makronun kullanıldığı yerlerde hata verir. Çözümü, makro tanımlanan yerin macro-mend dışına çıkarılmasıdır. Bunu yaparken dışarı çıkarılan yerdeki &data, &customer\_id, &out ve &details gibi yerlerin değiştirilmesi gerekmektedir. Örneğin bu örnekte &data geçen yerlere ‘WORK.BOND\_ALL’, &customer\_id kisimlarinin asağıdaki şekilde değiştirilmesi gerekmektedir.

|  |
| --- |
| proc contents data=WORK.BOND\_ALL out=names(keep=name type format) noprint; |
| run; |
|  |
| proc sql noprint; |
| select name into :varnames seperated by " " |
| from names where |
| lowcase(name) ne lowcase("MUS\_NO") |
| and format not like 'DATE%' and type = 1; |
| quit; |
|  |
| %macro imputeOutliers(data=, customer\_id=, out=, details=\_outlierdetails\_); |
| %let name = %scan(&data, 1, '()'); |
| %let libname=%scan(&name, 1, '.'); |
| %let memname=%scan(&name, 2, '.'); |
|  |
| . |
| . |
| . |
| . |
| . |
| . |
| . |
| %mend; |
| %imputeOutliers(data=WORK.BOND\_ALL, customer\_id=MUS\_NO, out=WORK.OUTLIER\_USED\_bond, details=\_detailsOUTLIER\_); |

### Split Columns Prosedürünün Kullanılmasından Kaynaklanan Hatalar

Otomatize edilmek istenen modellerin çoğunda aşağıdaki örnektekine benzer şekilde split columns prosedürü kullanılmıştır. Bu kısımlar, akış arkaplanda çalıştırıldığında hata almamıza neden olmaktadır.

|  |
| --- |
| /\* ------------------------------------------------------------------- |
| Code generated by SAS Task |
|  |
| Generated on: Tuesday, June 07, 2016 at 2:10:03 PM |
| By task: 10 |
|  |
| Input Data: SASApp:WORK.BONOLAR |
| Server: SASApp |
| ------------------------------------------------------------------- \*/ |
|  |
| %\_eg\_conditional\_dropds(WORK.HB\_MTD\_MAX\_TUTAR\_TRH, |
| WORK.TMP0TempTableInput, |
| WORK.TMP1TempLabelText); |
| /\* ------------------------------------------------------------------- |
| Sort data set SASApp:WORK.BONOLAR |
| ------------------------------------------------------------------- \*/ |
|  |
| PROC SQL; |
| CREATE VIEW WORK.TMP0TempTableInput AS |
| SELECT T.MTD\_MAX\_TUTAR\_TRH, T.DONEM, T.MUS\_NO |
| FROM WORK.BONOLAR as T |
| ; |
| QUIT; |
|  |
| /\* ------------------------------------------------------------------- |
| To prevent log warnings, only process the EG\_LABEL statement when necessary. |
| ------------------------------------------------------------------- \*/ |
| %LET \_\_EG\_DROP\_LABEL\_\_=; |
| %LET \_\_EG\_LABEL\_OPT\_\_=; |
| PROC CONTENTS DATA=WORK.BONOLAR OUT=WORK.TMP1TempLabelText(KEEP=LABEL) NOPRINT; |
| RUN; |
|  |
| DATA \_NULL\_; |
| SET WORK.TMP1TempLabelText(WHERE=(LABEL NE "")); |
| CALL SYMPUT("\_\_EG\_DROP\_LABEL\_\_", "\_\_EG\_LABEL\_\_"); |
| CALL SYMPUT("\_\_EG\_LABEL\_OPT\_\_", "LABEL=\_\_EG\_LABEL\_\_"); |
| STOP; |
| RUN; |
|  |
| PROC TRANSPOSE DATA=WORK.TMP0TempTableInput |
| OUT=WORK.HB\_MTD\_MAX\_TUTAR\_TRH(LABEL="Split WORK.BONOLAR" |
| DROP=\_\_EG\_TYPE\_\_ &\_\_EG\_DROP\_LABEL\_\_ ) |
| NAME=\_\_EG\_TYPE\_\_ |
| &\_\_EG\_LABEL\_OPT\_\_ |
| PREFIX=HB\_MTD\_MAX\_TUTAR\_TRH\_ |
| ; |
| BY MUS\_NO; |
|  |
| ID DONEM; |
| VAR MTD\_MAX\_TUTAR\_TRH; |
|  |
| /\* ------------------------------------------------------------------- |
| End of task code |
| ------------------------------------------------------------------- \*/ |
| RUN; QUIT; |
| %\_eg\_conditional\_dropds(WORK.TMP0TempTableInput, |
| WORK.TMP1TempLabelText); |
| TITLE; FOOTNOTE; |

Bu kodlar, Sas tarafından otomatik olarak oluşturulan kodlardır ve içerisinde kullanılan ara tablolar bütün split columns prosedürlerinde aynıdır. Burada hata almamızın sebebi ise kırmızı ile belirtilen alanlardaki makroların WORK.TMP0TempTableInput.VIEW’ı work üzerinden kaldıramamasından kaynaklanır. Arkaplanda çalıştırılan akışta bu benzeri bir çok kısım bulunur ve akışın çalıştırılması sırasında, ilk split columns prosedürü sonrasında oluşan WORK.TMP0TempTableInput.VIEW, sonraki split columns prosedürlerinde aynı isimdeki view’ların oluşturulmasını engeller ve dolayısıla split columns prosedürü sonunda elde edilmek istenen tablo alınamaz.

Manuel olarak kullanıcı profile üzerinden çalıştırıldığı zaman, aynı isimde daha önceden oluşturulmuş olan ‘.VIEW’ların üzerine yazılmasına izin verirken arkaplanda izin vermez. Bu nedenle arkaplanda çalıştırılması için düzenlenen bütün akışlarda bu kısımlarda değişiklik yapmak gerekmektedir.

|  |
| --- |
| %macro \_eg\_conditional\_dropds1(table); |
|  |
| %if %sysfunc(exist(&table)) %then %do; |
| proc sql; |
| drop table &table; |
| quit; |
| %end; |
| %mend; |
|  |
| %macro \_eg\_conditional\_dropds2(table1, table2); |
|  |
| %if %sysfunc(exist(&table1)) %then %do; |
| proc sql; |
| drop view &table1; |
| drop table &table1; |
| quit; |
| %end; |
| %if %sysfunc(exist(&table2)) %then %do; |
| proc sql; |
| drop table &table2; |
| quit; |
| %end; |
|  |
| proc sql; |
| drop view &table1; |
| quit; |
|  |
| %mend; |
|  |
| %macro \_eg\_conditional\_dropds3(table1, table2, table3); |
|  |
| %if %sysfunc(exist(&table1)) %then %do; |
| proc sql; |
| drop table &table1; |
| quit; |
| %end; |
| %if %sysfunc(exist(&table2)) %then %do; |
| proc sql; |
| drop view &table2; |
| drop table &table2; |
| quit; |
| %end; |
| %if %sysfunc(exist(&table3)) %then %do; |
| proc sql; |
| drop table &table3; |
| quit; |
| %end; |
|  |
| proc sql; |
| drop view &table2; |
| quit; |
|  |
| %mend; |

Bu durumda yapılması gereken yukarıdaki kodun akışın en başına yerleştirilerek yeni makrolar tanımlanması ve ilgili kısımlarda WORK.TMP0TempTableInput.VIEW library üzerinden kaldırılacak şekilde makroların güncellenmesidir. Burada dikkat edilmesi gereken, oluşturulan makroların, parantez içerisinde belirtilen tablo sayısına göre isimlendirilmesidir. Dolayısıyla yapılması gereken düzenleme aşağıdaki gibi olmalıdır.

|  |
| --- |
| /\* ------------------------------------------------------------------- |
| Code generated by SAS Task |
|  |
| Generated on: Tuesday, June 07, 2016 at 2:10:03 PM |
| By task: 10 |
|  |
| Input Data: SASApp:WORK.BONOLAR |
| Server: SASApp |
| ------------------------------------------------------------------- \*/ |
|  |
| %\_eg\_conditional\_dropds3(WORK.HB\_MTD\_MAX\_TUTAR\_TRH, |
| WORK.TMP0TempTableInput, |
| WORK.TMP1TempLabelText); |
| /\* ------------------------------------------------------------------- |
| Sort data set SASApp:WORK.BONOLAR |
| ------------------------------------------------------------------- \*/ |
| . |
| . |
| /\* ------------------------------------------------------------------- |
| End of task code |
| ------------------------------------------------------------------- \*/ |
| RUN; QUIT; |
| %\_eg\_conditional\_dropds**2**(WORK.TMP0TempTableInput, |
| WORK.TMP1TempLabelText); |
| TITLE; FOOTNOTE; |

## Appendix

Akışlar içerisinde kaynak tablolardan çekilen ve değişken değerlerinde türkçe karakter içeren bazı değişkenler bulunmaktadır. Akışların devamında bu değişkenlerin proc transpose içinde kullanılması, değişken isimlerinde türkçe karakter olan tablolar yaratılmasına neden olmaktadır. Bu tarz durumlarda direk olarak değişken ismini değiştirmemiz mümkün olmadığı için aşağıdaki prosedürü izliyoruz.

Örnek olarak akışların birinde kullanılan ödeme çekleri tablosunu ele alalım:

|  |
| --- |
| proc sql; |
| create table odeme\_cekleri as |
| SELECT |
| t1.MUSTERI, |
| (datepart(t1.TARIH)) FORMAT=FINDFDD10. AS Tarih\_rpr, |
|  |
| year(datepart(t1.TARIH)) AS Yil, |
| (month(datepart(t1.TARIH))) AS Ay, |
|  |
| year((intnx('month',datepart(t1.TARIH),1))) AS Yil\_Next, |
| month((intnx('month',datepart(t1.TARIH),1))) AS Ay\_Next, |
|  |
| t1.ACIKLAMA\_4, |
| t1.ACIKLAMA\_2, |
| t1.ADET, |
|  |
| (intnx('month',datepart(t1.TARIH),1)-1) FORMAT=FINDFDD10. AS Lastday\_of\_month |
|  |
| FROM OFS\_CRM.SALESMAN\_NYM\_AYL\_DATA t1 |
| WHERE |
| t1.TARIH >= "&M\_13B."dt |
| and |
| t1.ACIKLAMA\_4 = 'Ödemeler' |
| and |
| t1.ACIKLAMA\_2 IN ('Bankamýz Çeki TL','Bankamýz Çeki YP','BLOKE ÇEK') |
| and |
| t1.ADET > 0 |
| and |
| t1.MUSTERI in (SELECT DISTINCT MUSTERI\_NO FROM WORK.KOBI\_MUSTERILERI) |
| AND |
| (CALCULATED Tarih\_rpr)=(CALCULATED Lastday\_of\_month) |
| ORDER BY t1.MUSTERI,Tarih\_rpr; |
|  |
|  |
| quit; |
|  |
| PROC SQL; |
| CREATE TABLE WORK.odeme\_cekleri\_2 AS |
| SELECT DISTINCT t1.MUSTERI, |
| t1.Yil, |
| t1.Ay, |
| t1.ACIKLAMA\_4, |
| t1.ACIKLAMA\_2, |
| /\* Urun\_Adet \*/ |
| (SUM(t1.ADET)) AS Urun\_Adet |
| FROM WORK.ODEME\_CEKLERI t1 |
| GROUP BY t1.MUSTERI, |
| t1.Yil, |
| t1.Ay, |
| t1.ACIKLAMA\_4, |
| t1.ACIKLAMA\_2; |
| QUIT; |

Görüldüğü üzere oluşturulan Odeme\_cekleri ve Odeme\_cekleri\_2 tablolarında OFS\_CRM.SALESMAN\_NYM\_AYL\_DATA tablosundan çekilen ACIKLAMA\_2 ve ACIKLAMA\_4 değişkenleri, türkçe karakter içeren değerler bulunmaktadır. Fakat şu an için bunlar değişken ismi olmadığından herhangi bir sıkıntı yaratmamaktadır.

|  |
| --- |
| PROC TRANSPOSE DATA=WORK.TMP0TempTableInput |
| OUT=WORK.odeme\_cekleri\_3(LABEL="Split WORK.ODEME\_CEKLERI\_2" |
| DROP=\_\_EG\_TYPE\_\_ &\_\_EG\_DROP\_LABEL\_\_ ) |
| NAME=\_\_EG\_TYPE\_\_ |
| &\_\_EG\_LABEL\_OPT\_\_ |
| ; |
| BY MUSTERI Yil Ay; |
|  |
| ID ACIKLAMA\_2; |
| VAR Urun\_Adet; |
|  |
| /\* ------------------------------------------------------------------- |
| End of task code |
| ------------------------------------------------------------------- \*/ |
| RUN; QUIT; |

Fakat akışın devamında kullanılan ‘Split Columns’ prosedüründe yer alan proc transpose işleminde ID kisminda ACIKLAMA\_2 değişkeni kullanılması, yeni oluşturulan Odeme\_cekleri\_3 tablosunda türkçe karakter içeren değişken isimleri oluşmasına neden olmaktadır.

|  |
| --- |
| t4.'Bankamız Çeki TL'n, |
| t4.'Bankamız Çeki YP'n, |
| t4.'BLOKE ÇEK'n, |
| FROM WORK.MUSTERI\_SAHIPLIK\_4 t1 |
| LEFT JOIN WORK.NKRD\_3 t2 ON (t1.MUSTERI\_NO = t2.MUSTERI\_NO) AND (t1.Ay = t2.Ay) AND (t1.Yil = t2.Yil) |
| LEFT JOIN WORK.GNAKDI\_3 t3 ON (t1.MUSTERI\_NO = t3.MUSTERI\_NO) AND (t1.Ay = t3.Ay) AND (t1.Yil = t3.Yil) |
| LEFT JOIN WORK.ODEME\_CEKLERI\_3 t4 ON (t1.MUSTERI\_NO = t4.MUSTERI) AND (t1.Ay = t4.Ay) AND (t1.Yil = t4.Yil) |

Bunun sonucunda akışın devamında yukarıdaki gibi bir sorgu ile karşılaşıyoruz. Burada sorgu içerisinde türetilen değişkenlerde oldugu gibi direk olarak değişken ismini değiştiremiyoruz. Direk olarak ‘Bankamız Çeki TL’n ismini ‘Bankamiz\_Ceki\_TL’ olarak değiştirmemiz, öyle bir değişken olmadığından dolayı hata almamıza neden olacaktır. Bu nedenle bu tarz durumlarda akışı geriye doğru takip ederek değişkenler oluşturulmadan önce işlem yapıyoruz.

Bu örnekte türkçe karakter içeren değişkenlerin proc transpose kısmında kullanılan odeme\_cekleri\_2 tablosundaki ACIKLAMA\_2 değişkeninden geldiğini görüyoruz. Dolayısıyla odeme\_cekleri\_2 tablosu oluşturulmadan önce ACIKLAMA\_2 değişkeninin değerlerindeki türkçe karakterleri ve boşlukları kaldırmamız gerekiyor.

|  |
| --- |
| proc sql; |
| select distinct ACIKLAMA\_2 |
| from odeme\_cekleri; |
| quit; |

İlk adım olarak yukarıdaki sorgu ile odeme\_cekleri\_2 tablosuna odeme\_cekleri tablosundan gelen ACIKLAMA\_2 değişkeninin bütün değerlerini görüntülüyoruz.

|  |
| --- |
| data odeme\_cekleri\_11; |
| set odeme\_cekleri; |
| if ACIKLAMA\_2 = 'BLOKE ÇEK' then ACIKLAMA\_2 = 'BLOKE\_CEK'; |
| if ACIKLAMA\_2 = 'Bankamýz Çeki TL' then ACIKLAMA\_2 = 'Bankamiz\_Ceki\_TL'; |
| if ACIKLAMA\_2 = 'Bankamýz Çeki YP' then ACIKLAMA\_2 = 'Bankamiz\_Ceki\_YP'; |
| run; |

Ardından tespit ettiğimiz bütün türkçe karakter ve boşluk içeren değişken değerlerini yukarıdaki şekilde değiştiriyoruz. Bu sorguyu, odeme\_cekleri tablosu oluşturulduktan sonra, odeme\_cekleri\_2 tablosu oluşturulmadan önce olacak şekilde akışımıza yerleştiriyoruz.

|  |
| --- |
| PROC SQL; |
| CREATE TABLE WORK.odeme\_cekleri\_2 AS |
| SELECT DISTINCT t1.MUSTERI, |
| t1.Yil, |
| t1.Ay, |
| t1.ACIKLAMA\_4, |
| t1.ACIKLAMA\_2, |
| /\* Urun\_Adet \*/ |
| (SUM(t1.ADET)) AS Urun\_Adet |
| FROM WORK.ODEME\_CEKLERI\_11 t1 |
| GROUP BY t1.MUSTERI, |
| t1.Yil, |
| t1.Ay, |
| t1.ACIKLAMA\_4, |
| t1.ACIKLAMA\_2; |
| QUIT; |

Son adım olarak proc transpose’da kullandığımız odeme\_cekleri\_2 tablosunun oluşturulduğu sorguda input tabloyu yeni oluşturduğumuz tablo ile değiştiriyoruz.

Bu işlemlerin ardından, adını değiştirdiğimiz değişkenlerin akışın devamında kullanıldığı yerleri tespit ederek bütün sorgularda aynı değişikliği yapıyoruz :

|  |
| --- |
| t4.Bankamiz\_Ceki\_TL, |
| t4.Bankamiz\_Ceki \_YP, |
| t4.BLOKE\_CEK, |
| FROM WORK.MUSTERI\_SAHIPLIK\_4 t1 |
| LEFT JOIN WORK.NKRD\_3 t2 ON (t1.MUSTERI\_NO = t2.MUSTERI\_NO) AND (t1.Ay = t2.Ay) AND (t1.Yil = t2.Yil) |
| LEFT JOIN WORK.GNAKDI\_3 t3 ON (t1.MUSTERI\_NO = t3.MUSTERI\_NO) AND (t1.Ay = t3.Ay) AND (t1.Yil = t3.Yil) |
| LEFT JOIN WORK.ODEME\_CEKLERI\_3 t4 ON (t1.MUSTERI\_NO = t4.MUSTERI) AND (t1.Ay = t4.Ay) AND (t1.Yil = t4.Yil) |