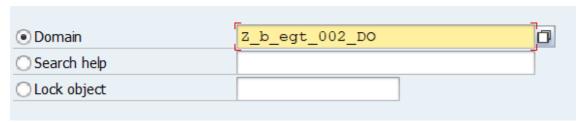
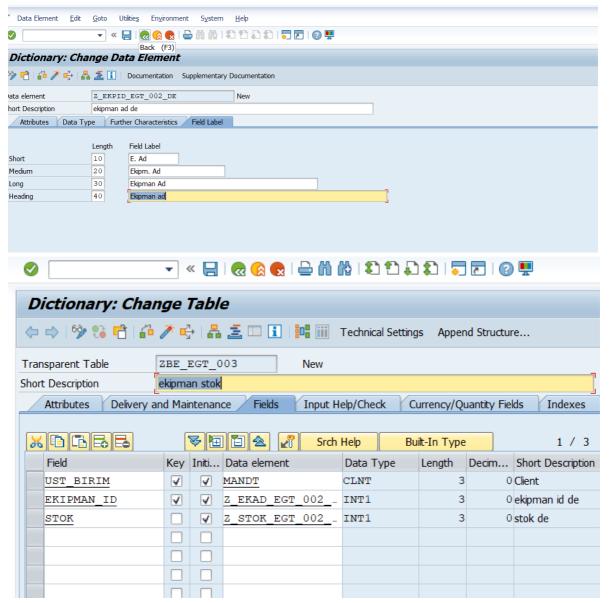
FOR ALL ENTRIES IN

Çekilmiş veya içeride oluşturmuş olduğumuz zaten elimizde olan bir data var, bu datalarla beraber db'e çoklu sorgu atmak istiyoruz. Aslında iki db joinlemeyip bir db ve bir tane internal table ile eşleştirme yapıyoruz. Şimdi ise ilk önce tablo datası çekip internal table atayacaz, bu table ile ise join yapılacak.



En başta iki farklı tablo yaparak işe başlıyoruz. Domainden itibaren başlatımı yapıyoruz. Domainlerin tamamlanmasından sonra ise gelen kısım elbette data element alanı olmuş oluyor. Onları da teker teker oluşturduktan sonra table oluşturma kısmına geçiş yapıyoruz.

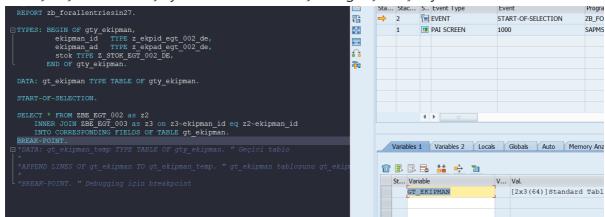


İki farklı tablo data element ve domainleriyle birlikte oluşturulmuş oldu. 002 adlı tablo id ve ad tutarken 003 adlı tablo ise yine id ve stok tutar. Bu noktadan sonra yapılacak şey for all entries adımlarına devam etmek olacak.

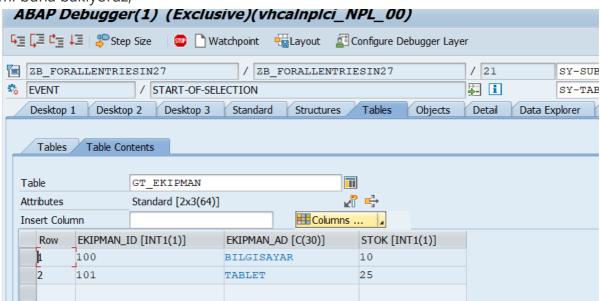
En başta tabloda veri tutmamız gerektiğinden girişleri yapıyoruz, her tablo içinde üçer atama yaptıktan sonra se38 ile kodumuza dönüş yapıyoruz.



Birkaç alıştırma ile başlayarak INNER JOIN işlemi gerçekleştiriyoruz;



Debug üzerinde kontrolünü yaparken değerlerin joinlenmesi yapılmış mı yapılmamış mı buna bakıyoruz;



Görüldüğü gibi ID'si eşleşen tüm verileri alarak gt_ekipman'a aktardı.

Bu kısa bir join özeti şeklindeydi ve şimdi asıl yapmak istediğimiz alana geliyoruz;

```
DATA: gt_table2 TYPE TABLE OF zbe_egt_002.

START-OF-SELECTION.

SELECT * FROM zbe_egt_002 INTO TABLE gt_table2.
BREAK-POINT.
```

Bu şekilde bir yapı oluşturduktan sonra zbe_egt_002 değerleri gt_table2 içerisinde bulunmuş oluyor. Fakat buradan zbe_egt_003 değerleri ulaşmaya çalışıyoruz. Normalde tablo3'e bağlamak için parametre ya da select options olsaydı "where"

kullanabilirdik ama burada bu internal table olduğu için mümkün olmuyor. Bunu yedirebilemek için ise şu şekilde bir kullanım yapıyoruz;

```
DATA: gt_table2 TYPE TABLE OF zbe_egt_002,
    gt_table3 type table of zbe_egt_003,
    gt_table3_tmp type table of zbe_egt_003.

START-OF-SELECTION.

SELECT * FROM zbe_egt_002 INTO TABLE gt_table2.
SELECT * FROM zbe_egt_003 INTO TABLE gt_table3_tmp.

SELECT * FROM zbe_egt_003 INTO TABLE gt_table3
FOR ALL ENTRIES IN gt_table2 WHERE ekipman_id EQ gt_table2-ekipman_id.
```

Burada önemli olan for all entries in dedikten sonra yazacağımız tablonun boş olmamasıdır. Tablo içeriği kesinlikle dolu olmalı. Test edebilmek adına;

```
TA: gt_table2 TYPE TABLE OF zbe_egt_002,
    gt_table2_tmp TYPE TABLE OF zbe_egt_002,
    gt_table3 type table of zbe_egt_003,
    gt_table3_tmp type table of zbe_egt_003.

ART-OF-SELECTION.

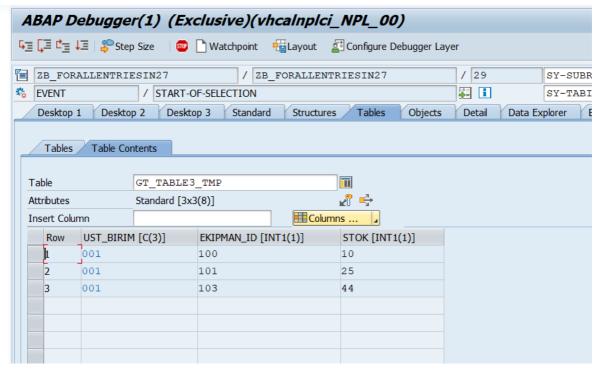
SELECT * FROM zbe_egt_002 INTO TABLE gt_table2.

SELECT * FROM zbe_egt_003 INTO TABLE gt_table3
    FOR ALL ENTRIES IN gt_table2 WHERE ekipman_id EQ gt_table2-ekipman_id.

SELECT * FROM zbe_egt_003 INTO TABLE gt_table3_tmp
    FOR ALL ENTRIES IN gt_table2_tmp WHERE ekipman_id EQ gt_table2_tmp-ekipman_id

BREAK-POINT.
```

gt_table2_tmp adında bir geçiçi internal table oluşturduk fakat içini boş bıraktık.



Ve yukarıda da görüldüğü gibi for all entries in olayının hiçbir hükmü kalmamış oldu. Direkt olarak ne kadar veri varsa hepsini aldı. İşte bu nedenden dolayı da table'in boş olmaması büyük önem taşımaktadır.

Araya tablonun dolu mu boş mu olduğunu anlayan bir yapı kurabiliriz, bunu yaparsak kod daha sağlıklı işlemiş olacak;

```
IF gt_table2 IS NOT INITIAL. "gt_table2 bossa bunun icerisine girmeyecek.
    SELECT * FROM zbe_egt_003 INTO TABLE gt_table3
    FOR ALL ENTRIES IN gt_table2 WHERE ekipman_id EQ gt_table2-ekipman_id.
ENDIF.
```

Yukarıda görüldüğü haliyle IF yapısı kurulduğunda daha sağlıklı bir for all entries in işlemi gerçekleşmiş olur. Eğer gt_table2 boşsa bu bloğun içerisine girmez, değilse girecek.