

# 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 ataycaz, bu table ile ise join yapılacaktır.

<input checked="" type="radio"/> Domain	<input type="text" value="z_b_egt_002_DO"/>
<input type="radio"/> Search help	<input type="text"/>
<input type="radio"/> Lock object	<input type="text"/>

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.

Dictionary: Change Data Element

data element: Z\_EKPID\_EGT\_002\_DE New

Short Description: ekipman ad de

Attributes Data Type Further Characteristics Field Label

	Length	Field Label
Short	10	E. Ad
Medium	20	Ekipm. Ad
Long	30	Ekipman Ad
Heading	40	Ekipman ad

Dictionary: Change Table

Transparent Table: ZBE\_EGT\_003 New

Short Description: ekipman stok

Attributes Delivery and Maintenance Fields Input Help/Check Currency/Quantity Fields Indexes

Field	Key	Initi...	Data element	Data Type	Length	Decim...	Short Description
UST_BIRIM	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3	0	Client
EKIPMAN_ID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Z_EKAD_EGT_002_...	INT1	3	0	ekipman id de
STOK	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Z_STOK_EGT_002_...	INT1	3	0	stok de
	<input type="checkbox"/>	<input type="checkbox"/>					
	<input type="checkbox"/>	<input type="checkbox"/>					
	<input type="checkbox"/>	<input type="checkbox"/>					
	<input type="checkbox"/>	<input type="checkbox"/>					

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.

Data Browser: Table ZBE\_EGT\_003 Select Entries 3

Table: ZBE\_EGT\_003

Displayed Fields: 3 of 3 Fixed Columns: [2] List Width 0250

	UST_BIRIM	EKIPMAN_ID	STOK
<input type="checkbox"/>	001	100	10
<input type="checkbox"/>	001	101	25
<input type="checkbox"/>	001	103	44

Birkaç alıştırma ile başlayarak INNER JOIN işlemi gerçekleştiriyoruz;

```
REPORT zb_forallentriesin27.

TYPES: BEGIN OF gty_ekipman,
        ekipman_id TYPE z_ekpid_egt_002_de,
        ekipman_ad TYPE z_ekpad_egt_002_de,
        stok TYPE z_stok_egt_002_de,
      END OF gty_ekipman.

DATA: gt_ekipman TYPE TABLE OF gty_ekipman.

START-OF-SELECTION.

SELECT * FROM ZBE_EGT_002 as z2
      INNER JOIN ZBE_EGT_003 as z3 on z3-ekipman_id eq z2-ekipman_id
      INTO CORRESPONDING FIELDS OF TABLE gt_ekipman.

BREAK-POINT.

"DATA: gt_ekipman_temp TYPE TABLE OF gty_ekipman. " Geçici tablo
"APPEND LINES OF gt_ekipman TO gt_ekipman_temp. " gt_ekipman tablosunu gt_ekipman_temp'e ekliyoruz
"BREAK-POINT. " Debugging için breakpoint
```

Sta...	Stac...	S...	Event Type	Event	Progra
2			EVENT	START-OF-SELECTION	ZB_FO
1			PAI SCREEN	1000	SAPMS

St...	Variable	V...	Val.
	GT_EKIPMAN		[2x3 (64)] Standard Tabl

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

**ABAP Debugger(1) (Exclusive)(vhcalnplci\_NPL\_00)**

Step Size | Watchpoint | Layout | Configure Debugger Layer

ZB\_FORALLENTRIESIN27 / ZB\_FORALLENTRIESIN27 / 21 SY-SUB

EVENT / START-OF-SELECTION SY-TAB

Desktop 1 | Desktop 2 | Desktop 3 | Standard | Structures | Tables | Objects | Detail | Data Explorer

Tables | Table Contents

Table: GT\_EKIPMAN

Attributes: Standard [2x3(64)]

Insert Column: Columns ...

Row	EKIPMAN_ID [INT1(1)]	EKIPMAN_AD [C(30)]	STOK [INT1(1)]
1	100	BILGISAYAR	10
2	101	TABLET	25

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 yedirebilmek 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 geçişi internal table oluşturduk fakat içeriğini boş bıraktık.

**ABAP Debugger(1) (Exclusive)(vhcalnplci\_NPL\_00)**

Step Size | Watchpoint | Layout | Configure Debugger Layer

ZB\_FORALLENTRIESIN27 / ZB\_FORALLENTRIESIN27 / 29 SY-SUBR

EVENT / START-OF-SELECTION SY-TAB

Desktop 1 Desktop 2 Desktop 3 Standard Structures Tables Objects Detail Data Explorer

Tables Table Contents

Table GT\_TABLE3\_TMP

Attributes Standard [3x3(8)]

Insert Column Columns ...

Row	UST_BIRIM [C(3)]	EKIPMAN_ID [INT1(1)]	STOK [INT1(1)]
1	001	100	10
2	001	101	25
3	001	103	44

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 boşsa bunun içerisine 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.