

ABAP SİSTEM ALANLARI

SAP ABAP'da "sistem alanları" genellikle "sy" (System) öneki ile başlayan ve programın çalışma zamanında çeşitli sistem durumu bilgilerini içeren özel değişkenler veya yapılar olarak ifade edilir. Bu sistem alanları, programın çalışma sırasında sistem durumu, kullanıcı bilgileri, mesajlar ve diğer çeşitli bilgiler hakkında bilgi sağlamak için kullanılır. Sistem içerisinde bulunan bazı alanlar ve açıklamaları aşağıdaki gibidir.

SYST_SUBRC:

Aşağıda kodlamasını gösterdiğim bu işlemin çalışma şekli şudur; işlem sonunda 0 dönüşü olursa işlem başarılı, 4 dönüşü olursa uyarı, 8 dönüşü olursa hatalı işlem olduğunu gösterir. Aşağıdaki kodda sistem çalışması başarılı olursa yani 'sy-subrc = 0' ekrana 'Sistem çalıştı' çıktısını verir.

```
IF sy-subrc eq 0.  
    WRITE : 'Sistem çalıştı.'.  
ENDIF.
```

SYST_LANGU:

Bu structure ile kullanıcının sisteme hangi dil ile giriş yaptığını tespit etme ve ekrana yazdırma işlemini yapabiliriz. Kullanımı şu şekildedir:

```
if sy-langu EQ 'EN'.  
  
    write: 'İngilizce giriş yapıldı.'.  
  
else.  
  
    write: 'Başka bir dil ile giriş yapıldı.'.  
  
endif.
```

SYST_DATUM:

Bu değişken, programın çalışma zamanında o anki tarihi temsil eder. Yıl, ay ve gün bilgilerini içerir. Programlarda belirli bir tarihi kontrol etmek, kaydetmek veya ekranlarda görüntülemek amacıyla kullanılabilir. Ayrıca, bu değişkenin formatı, dil ortamına bağlı olarak değişebilir. Örneğin 'EN' ülkesi için belirli bir format kullanılabilir ve kullanımı şu şekildedir:

```
sy-datum COUNTRY = 'EN'
```

SYST_INDEX:

DO ve WHILE döngülerindeki döngü geçişlerinin sayısını içerir. Bu, DO veya WHILE döngüsünün içinde kaç kez geçildiğini, şu anda hangi geçişte olduğunu takip etmek için kullanılır. Bu değer, mevcut döngü geçişi dahil olmak üzere geçmiş tüm döngü geçişlerini içerir. SY-TABIX ile karıştırılmamalıdır.

SYST_TABIX:

Bu değişken, bir iç tablonun içindeki geçerli satırın indeksini tutar. Yani, iç tablo üzerinde yapılan işlemlerde hangi satırda olduğumuzu belirlemek için kullanılır.

-**APPEND** ifadesi bir iç tabloya yeni bir satır eklerken, SY-TABIX değişkeni, eklenen son satırın indeksini içerir. Bu sayede eklenen satırın pozisyonunu takip edebiliriz.

-**LOOP AT** ifadesi içinde döngü yapılırken, her döngü geçişinde SY-TABIX, mevcut döngü geçişinin indeksini içerir ve iç tablo üzerinde dolaşırken her geçişte o anki satırın indeksini gösterir.

-**READ TABLE** ifadesi ile bir iç tablonun belirli bir satırı okunduğunda, SY-TABIX, okunan satırın indeksini içerir.

SYST_UCOMM:

Bu değişken, kullanıcının ekranda gerçekleştirdiği son kullanıcı işlem kodunu (User Command) içerir. Kullanıcının yaptığı bir işlemi (örneğin, bir düğmeye tıklama veya bir menü seçeneğini seçme) takip etmek için kullanılır.

SYST_TCODE:

Bu değişken, mevcut işlem kodunu içerir. İşlem kodu, kullanıcının çalıştırdığı program veya işlemleri temsil eden bir koddur. Örnek kullanımı şu şekildedir:

```
DATA: lv_tcode TYPE sy-tcode.
```

```
lv_tcode = sy-tcode.
```

```
WRITE: / 'Mevcut İşlem Kodu:', lv_tcode.
```

Bu kodun ekran çıktısı:

Eğitim Program 1

Mevcut İşlem Kodu: SE38

SYST_HOST:

Bu değişken, programın çalıştığı SAP sisteminin ana bilgisayar (host) adını içerir. Yani, programın çalıştığı fiziksel veya sanal sunucunun adını temsil eder. Özellikle dağıtık sistemlerde veya farklı ortamlarda çalışan programların, çalıştıkları sunucunun bilgisine ihtiyaç duyduğu durumlarda kullanılır. Ayrıca, loglama veya izleme amaçlarıyla programın çalıştığı sunucunun bilgisini elde etmek için de kullanılabilir. Örnek kullanımı şu şekildedir:

```
REPORT zhd_p01.  
  
DATA: lv_host TYPE sy-host.  
  
lv_host = sy-host.  
  
WRITE: / 'Mevcut Ana Bilgisayar:', lv_host.
```

Bu kodun ekran çıktısı:

```
Egitim Program 1  
  
Mevcut Ana Bilgisayar: idesapl
```

SYST_UNAME:

Bu değişken, programın çalıştığı anki SAP sisteminde oturum açan kullanıcının adını temsil eder. Bu değişken, programlar içinde oturum açan kullanıcının kimliğine erişim sağlamak için kullanılır. Kullanıcının kimliği, genellikle çeşitli yetki kontrolleri, kullanıcıya özel işlemler ve raporlama gibi durumlarda kullanılır. Örnek kullanımı şu şekildedir:

```
REPORT zhd_p01.  
  
DATA: lv_username TYPE sy-uname.  
  
lv_username = sy-uname.  
  
WRITE: / 'Mevcut Kullanıcı Adı:', lv_username.
```

Bu kodun ekran çıktısı:

```
Egitim Program 1  
  
Mevcut Kullanıcı Adı: B_HAMZAD
```

SYST_UZEIT:

Bu değişken, mevcut SAP sisteminin saat dilimine göre zamanını içerir. Programın çalıştığı anki sistem zamanını, saat ve dakika cinsinden temsil eder. SY-UZEIT, genellikle programın çalıştığı zamanı belirlemek, işlemlerin zaman damgasını almak veya belirli bir sürecin ne kadar sürede gerçekleştiğini takip etmek gibi durumlar için kullanılır. Bu değişken, zamanla ilgili işlemlerde kullanıcılara esneklik ve kontrol sağlar. Örnek kullanımı şu şekildedir:

```
DATA: lv_system_time TYPE sy-zeit.
```

```
lv_system_time = sy-zeit.
```

```
WRITE: / 'Mevcut Sistem Zamanı:', lv_system_time.
```

Ekran çıktısı:

Eğitim Program 1

Mevcut Sistem Zamanı: 16:24:39

OPEN SQL FONKSİYONLARI

Bu yazımda Open SQL sum, avg, max, min ve count fonksiyonlarının açıklamasını yapacağım. Bu fonksiyonlar genellikle 'SELECT' ifadesinin bir parçası olarak kullanılır ve belirli bir koşulu karşılayan satırlar üzerinde işlem yapar. Bunları ayrı ayrı kısaca açıklayacak olursak:

- 1. SUM:** Bu fonksiyon, belirli bir sütundaki değerlerin toplamını hesaplar. Kullanımı:
"Select Sum (AlanAdı) From TabloAdı." şeklindedir.
- 2. AVG:** Bu fonksiyon, belirli bir sütundaki değerlerin ortalamasını hesaplar. Boş değerler (NULL) hesaplamada dikkate alınmaz. Kullanımı:
"Select AVG (AlanAdı) From TabloAdı." Şeklindedir.
- 3. MAX ve MIN:** Bu fonksiyonlar belirtilen sütundaki değerlerin en büyük ve en küçük değerini sırasıyla döndürür. Kullanımı:
"Select Max (AlanAdı) From TabloAdı."
"Select Min (AlanAdı) From TabloAdı." Şeklindedir.
- 4. COUNT:** Bu fonksiyon, belirli bir koşulu karşılayan satır satır sayısını döndürür. Kullanımı:
"Select COUNT (*) From TabloAdı Where koşul." Şeklindedir.

OPEN SQL KOMUTLARI VE KULLANIMI

Open Sql komutlarını çalıştırmak için personel tablosu oluşturarak üzerinde işlemleri gerçekleştirdim ve kodlar üzerinde yorum satırı halinde yapılan işlemi açıklamaya çalıştım.

SELECT:

```
DATA : gv_persid type zhd_persid_de,  
        gv_persad TYPE zhd_persad_de,  
        gv_perssoyad TYPE zhd_perssoyad_de,  
        gv_perscins TYPE zhd_perscins_de,  
        gs_pers_t TYPE zhd_pers_t,  
        gt_pers_t TYPE TABLE OF zhd_pers_t.
```

**Tablo çektiğimiz ve tabloya attığımız için INTO TABLE kullandık:*

```
select * FROM zhd_pers_t  
        INTO TABLE gt_pers_t.
```

**Koşula göre tablodan veri çekmek (Id numarası 1 olan veriyi çekmek):*

```
select * FROM zhd_pers_t  
        INTO TABLE gt_pers_t  
        WHERE pers_id eq 1.
```

**Structure çekmek için:*

```
SELECT SINGLE * FROM zhd_pers_t  
        INTO gs_pers_t.
```

**Tek bir kolon çekmek için(PERS_ID Kolon Adı):*

```
SELECT SINGLE PERS_ID FROM zhd_pers_t  
        INTO gv_persid.
```

UPDATE:

```
DATA : gv_persid type zhd_persid_de,  
        gv_persad TYPE zhd_persad_de,  
        gv_perssoyad TYPE zhd_perssoyad_de,  
        gv_perscins TYPE zhd_perscins_de,  
        gs_pers_t TYPE zhd_pers_t,  
        gt_pers_t TYPE TABLE OF zhd_pers_t.
```

**Tablomda var olan 1 numaralı personel Id sahibi 'HAMZA' nın adını 'METE' ile değiştirme işlemi. Kodun çalıştığını da 'WRITE' kodunun çalıştığından anlamış olduk.*

```
UPDATE zhd_pers_t SET PERS_AD = 'METE'  
        WHERE pers_id eq 1.
```

```
WRITE : 'Update komutu çalıştırıldı.'.
```

INSERT:

```
DATA : gv_persid type zhd_persid_de,  
       gv_persad TYPE zhd_persad_de,  
       gv_perssoyad TYPE zhd_perssoyad_de,  
       gv_perscins TYPE zhd_perscins_de,  
       gs_pers_t TYPE zhd_pers_t,  
       gt_pers_t TYPE TABLE OF zhd_pers_t.
```

```
gs_pers_t-pers_id = 3.  
gs_pers_t-pers_ad = 'FURKAN'.  
gs_pers_t-pers_soyad = 'KO'.  
gs_pers_t-pers_cins = 'E'.  
INSERT zhd_pers_t FROM gs_pers_t.  
WRITE : 'Insert komutu çalıştırıldı.'.
```

*Insert komutu ile tabloya yeni veri girişi yapıldı. Bu işlem oluşturduğum structure'da gerekli alanları doldurarak yapıldı.

DELETE:

```
DATA : gv_persid type zhd_persid_de,  
       gv_persad TYPE zhd_persad_de,  
       gv_perssoyad TYPE zhd_perssoyad_de,  
       gv_perscins TYPE zhd_perscins_de,  
       gs_pers_t TYPE zhd_pers_t,  
       gt_pers_t TYPE TABLE OF zhd_pers_t.
```

```
DELETE FROM zhd_pers_t WHERE pers_id eq 2.  
WRITE : 'Delete komutu çalıştırıldı.'.
```

*Id'si 2 olan satırı tablodan silmek için Delete komutu yukarıdaki gibi çalışmaktadır.

MODIFY:

**Update ve insert bileşimi şeklinde çalışıyor denilebilir. Aşağıdaki gibi çalıştırırsak insert gibi ekleme yapacaktır.*

```
gs_pers_t-pers_id = 4.  
gs_pers_t-pers_ad = 'Ali'.  
gs_pers_t-pers_soyad = 'Veli'.  
gs_pers_t-pers_cins = 'E'.
```

```
MODIFY zhd_pers_t FROM gs_pers_t.  
WRITE : 'Modify komutu çalıştırıldı.'.
```

Daha öncesinde tabloda bu bilgilerde personel bulunmadığı için direk ekleme yaptı. Kodun çalıştıktan sonraki tablo görseli aşağıdaki gibidir (Görsel A.).

```

gs_pers_t-pers_id = 4.
gs_pers_t-pers_ad = 'Ali'.
gs_pers_t-pers_soyad = 'Sağlam'.
gs_pers_t-pers_cins = 'E'.

MODIFY zhd_pers_t FROM gs_pers_t.
WRITE : 'Modify komutu çalıştırıldı.'.

```

Bu kodlar çalıştırıldığına daha önceden var olan kısımlarda değişiklik yokken Soyad kısmı değiştiği için sadece orada değişiklik yaparak 'Veli' olan kısım 'Sağlam' olmuştur. (Görsel B.)

	MANDT	PERS_ID	PERS_AD	PERS_SOYAD
<input type="checkbox"/>	400	1	METE	DEMİR
<input type="checkbox"/>	400	3	FURKAN	KO
<input type="checkbox"/>	400	4	Ali	Veli

Görsel A.

	MANDT	PERS_ID	PERS_AD	PERS_SOYAD
<input type="checkbox"/>	400	1	METE	DEMİR
<input type="checkbox"/>	400	3	FURKAN	KO
<input type="checkbox"/>	400	4	Ali	Sağlam

Görsel B.

KAYNAKÇA

- <https://answers.sap.com/questions/2565537/sy-langu-how-to-compare.html>
 - <https://blogs.sap.com/2017/01/13/logon-language-sy-langu-and-rfc/>
 - [what is SY-DATUM | SAP Community](#)
 - [ABAP System Fields - ABAP Keyword Documentation \(sap.com\)](#)
 - [sy-tabix,sy-index | SAP Community](#)
 - [Detecting hard-coded user names in ABAP \(daniel-berlin.de\)](#)