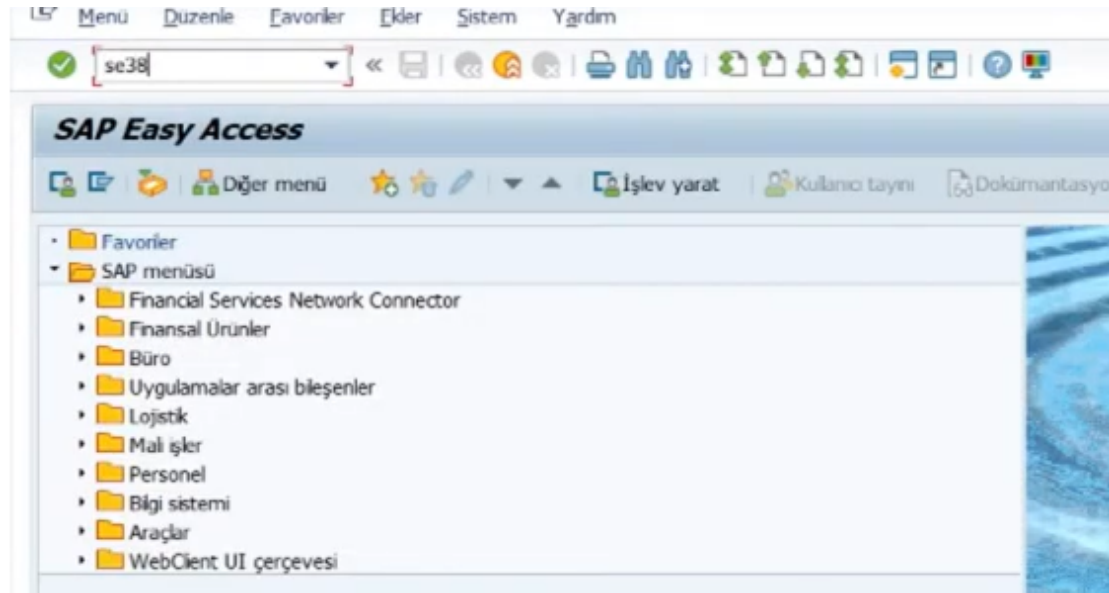


03.03.2023 | SAP ABAP EGITIMI 4 - OPEN SQL KOMUTLARI *

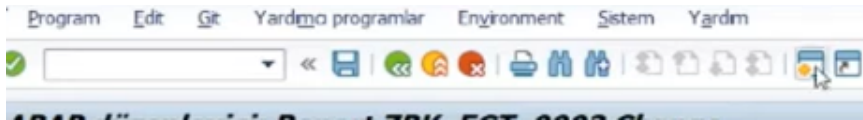
Eğitim içerisinde görülecek olanlar;

<input type="checkbox"/>	Variable
<input type="checkbox"/>	Structure
<input type="checkbox"/>	Table
<input type="checkbox"/>	Select
<input type="checkbox"/>	Update
<input type="checkbox"/>	Insert
<input type="checkbox"/>	Delete
<input type="checkbox"/>	Modify

SARI alanla gösterilmiş kısım tabloyu anlatmak için gösterilmiş alandır, tablo orasıdır. Mavi kısımlar, yani satırlar ise structure yapılarıdır, structure'ların içerisinde kalan tek hücrelik alanlar olan yeşiller ise variable olarak geçmektedir.



se38 ile program oluşturuyoruz.



Farenin üzerinde olduğu alan yeni bir section açar, yeni bir sayfa açar. Yeni sayfa açtıktan sonra se11 TCode girerek ZBK_PERS_t tablosunu açıyoruz.

Dictionary: Display Table

Transparent Table: ZBK_PERS_T Active

Short Description: Personellerin Verileri

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

Alan adı	Key	Initi...	Data element	Veri tipi	Length	Decim...	Short Description
MANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3		0 Üst birim
PERS_ID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ZBK_PERSI_DE	INT1	3		0 Personel Id
PERS_AD	<input type="checkbox"/>	<input type="checkbox"/>	ZBK_PERSAD_DE	CHAR	30		0 Personel Ad
PERS_SOYAD	<input type="checkbox"/>	<input type="checkbox"/>	ZBK_PERSSOYAD_DE	CHAR	40		0 Personel Soyad
PERS_CINS	<input type="checkbox"/>	<input type="checkbox"/>	ZBK_PERSCINS_DE	CHAR	1		0 Personel Cinsiyeti

Yine programa dönüp gv tanımlamalarıyla TYPE'ları tablodan alıp değişken(variable) tanımlamalarını yapıyoruz.

```
DATA: gv_persid TYPE zbk_persi_de,
      gv_persad TYPE zbk_persad_de,
      gv_perssoyad TYPE zbk_perssoyad_de,
      gv_perscins TYPE zbk_perscins_de.
```

Bunun ardından gelecek olan ise structure geliyor, onun tanımı da bu şekilde yapılır;

```
gs_pers_t TYPE ZBK_PERS_T.
```

Tablo adını kopyalayıp bu sefer gs ile başlatıyoruz, akıllı kodlama mantığıyla yapman her zaman daha iyi ve kolay okunabilir.

Tablo tanımı:

```
gt_pers_t TYPE TABLE OF zbk_pers_t.
```

Görüldüğü gibi bunu da ekleme yaparak TYPE değil TYPE TABLE OF diyerek tablo adını yanına yazarak tanımlamayı gerçekleştiriyoruz.

Bunları tanımladıktan sonra pekala nasıl kullanılır? Bunları kullanmak içinde SQL sorgularını kullanırız, temelde 5 tane vardır;

SELECT, UPDATE, INSERT, DELETE, MODIFY

SELECT;

Burada tablo çekiliyor;

```
SELECT * FROM zbk_pers_t
INTO TABLE gt_pers_t.]
```

Burada structure çekiliyor;

```
SELECT SINGLE * FROM zbk_pers_t
INTO gs_pers_t.]
```

Burada ise tek bir kolon çekiliyor;

```
SELECT SINGLE pers_id FROM zbk_pers_t  
INTO gv_persid.
```

Spesifik bir koşul vererek çekmek için ise şu şekilde bir kodlama yapıyoruz, örneğin burada pers_id değeri 1'e eşit olan durumları getirmesini istemiş oluyoruz;

```
SELECT * FROM zbk_pers_t  
INTO TABLE gt_pers_t  
WHERE pers_id EQ 1.
```

UPDATE;

Burada örnek olarak id'si 1 olan personelin adını HAKAN olarak değiştiriyoruz;

```
UPDATE zbk_pers_t SET pers_ad = 'HAKAN'  
WHERE pers_id EQ 1.
```

INSERT;

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

DELETE;

Örnek olarak pers_id'si 2 olan satırı komple sildik.

```
DELETE FROM zbk_pers_t WHERE pers_id EQ 2.  
WRITE: 'Delete komutu çalıştırıldı!'.
```

MODIFY;

Update ve Insert'n birleşimi şeklinde çalışır. Keyleri ortak bir path varsa diğer kolonların hepsini update edip yeni bir satır eklemeyecek fakat from kısmından sonra gs_pers_t'de verdiğim key zbk_pers_t'de yoksa onu yeni bir satır olarak ekleyecek. Yani o keyde satır varsa update, satır yoksa insert olarak çalışır.

```
gs_pers_t-pers_id = 4.  
gs_pers_t-pers_ad = 'Gülçin'.  
gs_pers_t-pers_soyad = 'SOYLEMEZ'.  
gs_pers_t-pers_cins = 'K'.  
MODIFY zbk_pers_t FROM gs_pers_t.  
WRITE: 'Modify komutu çalıştırıldı!'.
```

Pers Id	Pers Ad	Pers Soyad	P. Cins
3	FURKAN	SOYLEMEZ	E
4	Gülçin	SOYLEMEZ	K
1	HAKAN	KOCAASLAN	E

Görüldüğü gibi pers_id = 4 olan bir değer olmadığı için bunu insert etti. Fakat şöyle yapacak olursak;

```
gs_pers_t-pers_id = 4.  
gs_pers_t-pers_ad = 'Gülçin'.  
gs_pers_t-pers_soyad = 'KOCAASLAN'.  
gs_pers_t-pers_cins = 'K'.  
  
MODIFY zbk_pers_t FROM gs_pers_t.  
  
WRITE: 'Modify komutu çalıştırıldı!'.
```

ID 4'e denk bir tablo değeri olduğu için Gülçin Söylemez kodun çalıştırılmasıyla beraber Gülçin Kocaaslan olmuş oldu. Yani duruma göre değişiklik gösteren ve o durumla ilgili kısa yöntem şeklinde de düşünülebilir.