

## SYST Structure Alanları Nelerdir? Ne İşe Yarar?

SAP platformunda ABAP programlama dilinde kullanılan SYST yapısı, sistemle ilgili bilgileri içeren bir dizi alanı içerir. Bu alanlar, ABAP programları içinde kullanılarak sistem durumu, kullanıcı bilgileri ve çeşitli diğer önemli bilgiler hakkında bilgi sağlar. SYST yapısındaki önemli alanlar ve görevleri:

1. **SYST-TCODE:** İlgili işlem kodunu içerir. Bu alan, hangi SAP işlem koduyla çalıştığınızı belirtir.
2. **SYST-CPROG:** Çağrılan programın adını içerir. Bu alan, çalıştırılan programın ismini sağlar.
3. **SYST-TCOMM:** SAP transaksyonunun tipini belirler. Örneğin, 'SM37' gibi bir işlem kodu bir transaksyonu temsil eder.
4. **SYST-CUCOOKIE:** Kullanıcı çerezini içerir. Bu alan, kullanıcı oturumunun benzersiz bir tanımlayıcısını sağlar.
5. **SYST-SUBRC:** Alt programın dönüş kodunu içerir. 0 başarılı bir işlemi, 4 ise başarısız bir işlemi temsil eder.
6. **SYST-COMMIT:** Bir COMMIT WORK komutunun gerçekleşip gerçekleşmediğini belirler. 'X' ise bir COMMIT WORK gerçekleşti, boş ise gerçekleşmedi.
7. **SYST-DBDAT:** Sistem tarihini içerir. Bu alan, sistemdeki geçerli tarihi sağlar.
8. **SYST-DBTIM:** Sistem saatinin değerini içerir. Bu alan, sistemdeki geçerli saati sağlar.
9. **SYST-MSGID, SYST-MSGTY, SYST-MSGNO, SYST-MSGTY:** Son alınan hata veya bilgi mesajına ait bilgileri içerir. MSGID, MSGTY ve MSGNO ile birlikte kullanılır.
10. **SYST-CALLSTACK:** Çağrı yığını içerir. Bu alan, ABAP programında bir CALL FUNCTION veya PERFORM çağrısı yaptığınızda yığın içindeki bilgileri sağlar.
11. **SYST-MSGV1, SYST-MSGV2, SYST-MSGV3, SYST-MSGV4:** Hata veya bilgi mesajındaki değişken değerlerini içerir. MSGID, MSGTY ve MSGNO ile birlikte kullanılır.

## Open SQL sum,avg,max,min,count işlemleri

ABAP programlama dilinde, Open SQL kullanılarak veritabanı işlemleri gerçekleştirilebilir. Bu işlemler arasında SUM, AVG, MAX, MIN ve COUNT gibi toplu işlemler önemli bir yer tutar. İşlemlerin Open SQL içindeki kullanımları:

### **SUM:**

SUM, belirli bir sütundaki değerlerin toplamını hesaplamak için kullanılır.

```
SELECT SUM(ekonomik_birim) INTO lv_toplam  
FROM s_economic_unit  
WHERE company_code = '100'.
```

### **AVG:**

AVG, belirli bir sütundaki değerlerin ortalamasını hesaplamak için kullanılır.

```
SELECT AVG(salary) INTO lv_ortalama  
FROM employee  
WHERE department = 'IT'.
```

### **MAX:**

MAX, belirli bir sütundaki en büyük değeri bulmak için kullanılır.

### **MIN:**

MIN, belirli bir sütundaki en küçük değeri bulmak için kullanılır.

```
SELECT MIN(order_date) INTO lv_min_tarih  
FROM sales_orders  
WHERE customer_id = '67890'.
```

### **COUNT:**

COUNT, belirli bir koşula uyan satırların sayısını bulmak için kullanılır.

## Open SQL , insert, update, delete, modify Komutları

```
REPORT ZMK_SQL_EXAMPLE.
```

```
DATA: lv_customer TYPE string,  
      lv_country TYPE string.
```

```
* INSERT
```

```
lv_customer = 'NewCustomer'.
```

```
lv_country = 'Germany'.
```

```
INSERT INTO customers
```

```
  (customer_name, country)
```

```
VALUES
```

```
  (lv_customer, lv_country).
```

```
* UPDATE
```

```
lv_customer = 'UpdatedCustomer'.
```

```
lv_country = 'France'.
```

```
UPDATE customers
```

```
  SET customer_name = lv_customer
```

```
  WHERE country = 'Germany'.
```

```
* DELETE
```

```
DELETE FROM customers
```

```
  WHERE customer_name = 'ObsoleteCustomer'.
```

```
* MODIFY
```

```
lv_customer = 'ModifiedCustomer'.
```

```
lv_country = 'Italy'.
```

```
MODIFY customers
```

```
  FROM lv_customer
```

```
  WHERE country = 'France'.
```

```
* Display the result
```

```
SELECT * FROM customers INTO TABLE @DATA(result).
```

```
LOOP AT result INTO DATA(customer).
```

```
  WRITE: / 'Customer:', customer-customer_name,
```

```
          'Country:', customer-country.
```

```
ENDLOOP.
```