**INTERNAL TABLOLAR**

**İNTERNAL TABLOLAR (INTERNAL TABLES)**

İnternal tablolar verileri ABAP programı çalıştığı sürece tablo şeklinde hafızada saklarlar. Veriler satır, satır saklanır. En çok veritabanı tablosundan alınan verileri depolamak ve düzenlemek için kullanılırlar. Kayıtlar internal tabloya eklendikten sonra silme, değiştirme, ekleme, sıralama gibi işlemler yapılabilir.

· Veriler satırlar halinde tutulur.

· Her satır aynı yapıdadır.

· Veriler SAP tablolarından veya başka kaynaklardan alınabilir.

· Hafızada oluşturulan veri nesneleridir. ABAP programı çalıştığı sürece hafızada saklanırlar.

İnternal Tablonun Veri Tipi (Data Type)

İnternal tabloları tanımlamak için veri tipleri kullanılır. Veri tipleri ABAP Dictionary içerisinde tanımlı bir tablo tipi veya TYPES ve DATA anahtar sözcükleri kullanılarak tanımlanabilir.

İnternal tablo veri tipini aşağıdaki bilgiler belirler.

1. Satir tipi (row type),

2. Tablo kategorisi (table category)

3. Tablo anahtarı (table key)

1. Satır Tipi (line type): İnternal tablonun satır tipidir. Genelde bir structure satır tipi olarak belirtilir. Structure’ ın her bir bileşeni (değişkeni) internal tablonun bir sütununa karşılık gelir.

2. Tablo Kategorisi: İnternal tablonun nasıl yönetileceğini ve kayıtlara nasıl erişileceği bilgisidir. 3 internal tablo kategorisi vardır.

a. Standart (standard)

b. Sıralı (sorted)

c. Karışık (Hashed)

3. Tablo Anahtarı (key): Tablodaki satırların anahtar alanlarını belirler. İki çeşit anahtar vardır.

a. Birincil anahtarlar (primary keys)

b. İkincil anahtarlar (secondary keys)

Tüm internal tablolar birincil anahtara sahiptir. Birincil anahtar standart veya kullanıcı tanımlı olabilir. Tablo kategorisine göre anahtar alanlar unique veya non-unique olarak tanımlanabilirler. Unique tanımda aynı değere sahip anahtar alanları olan birden fazla kayıt olamaz.

**İnternal Tablo Kategorileri**

Üç tip dâhilî tablo kategorisi vardır.

1. Standart (standard)

2. Sıralı (sorted)

3. Karışık (hashed)

**Standart (Standard) İnternal Tablo**

· Kayıtlara indeks numarası veya anahtar alanlar kullanılarak erişilir.

· Anahtar alanlarla kayıta erişim için geçen süre tablodaki kayıt sayısı ile orantılıdır.

· Standart tablonun anahtarları tekrarlanabilir (non-unique) yapıdadır. Unique anahtar tanımlanamaz.

· Dâhilî tabloya kayıt eklenirken unique anahtar (unique key) kontrolü yapılmadığı için standart tablolar çok hızlı doldurulurlar.

· Erişim için ikincil anahtar tanımlanabilir.

**Sıralı (Sorted) Dâhilî Tablo**

· Daima birincil anahtar alanlara (primary table key) göre sıralıdırlar. İndekslemede yapılır. Dâhilî tablolardaki kayıtlara indeks numarası veya anahtar alanlar kullanılarak erişilir.

· Anahtar alanlarla kayda erişim için geçen süre tablodaki kayıt sayısına göre logaritmik orantılıdır.

· Sıralı tablonun anahtarları eşsiz (unique) veya tekrarlanabilir (non-unique) yapıdadır. Unique anahtar tanımlanamaz.

**Karışık (Hashed) Dâhilî Tablo**

· Doğrusal indeks yapısı yoktur.

· Birincil ve ikincil anahtar alanlar ile erişilebilirler.

· Kayda erişim zamanı birincil anahtara göre erişimine göre sabit ve tablodaki kayıt sayısına bağlı değildir.

· Karışık tablonun anahtarları tekrar edilemez (unique) yapıdadır.

**Not:**Standart ve sıralı dâhilî tablolara indeks numaraları ile erişilebildiği için INDEX tabloları olarak da adlandırılırlar.

**İnternal Tablo Tipi**

İnternal tablo tipi aşağıdaki şekilde tanımlanabilir.

TYPES veri\_tipi { {TYPE tablo\_tipi OF [REF TO] tip}

| {LIKE tablo\_tipi OF veri\_nesnesi} }

[tablo\_anahtarlari] [INITIAL SIZE n].

tip: TYPES anahtar sözcüğü ile tanımlanmış veri tipidir.

tablo\_tipi: İnternal tablonun kategorisidir. Aşağıdaki kategoriler kullanılabilir.

· [STANDARD] TABLE

· SORTED TABLE

· HASHED TABLE

· ANY TABLE

· INDEX TABLE

veri\_nesnesi: ABAP Dictionary’ de tanımlı veri nesnesidir.

Tablo\_anahtarlari: Tablo tipinin tablo anahtarlarını tanımlamak için kullanılır. Kullanımı aşağıdaki şekildedir.

· WITH key

· [ WITH ikinci\_anahtar1 ] [ WITH ikincil\_anahtar2 ] …

· {WITH|WITHOUT} FURTHER SECONDARY KEYS

Ekler:

key: Birincil tablo anahtarı tanımlamak için kullanılır.

ikincil\_anahtar1: 15 adete kadar ikincil tablo anahtarı tanımlanabilir.

{WITH|WITHOUT} FURTHER SECONDARY KEYS: WITH ile ikincil anahtarlar dikkate alınarak dahil edilir.

INITIAL SIZE n: Hafıza yönetimi ve sıralama için kullanılır. 0 değeri verilir ise hafıza yönetimi otomatik olarak tayin edilir.

**DÂHİLÎ TABLO DOLDURMA İFADELERİ**

**INSERT**

İnternal tabloya kayıt eklemek için kullanılır. Kullanımı aşağıdaki şekildedir.

INSERT eklenecek\_satir INTO tablo\_pozisyonu [sonuc].

eklenecek\_satir ile belirtilen satırı, tablo\_pozisyonu ile belirtilen pozisyona ekler.

**eklenecek\_satir**

Eklenecek kayıttır. 3 çeşit kayıt olabilir.

1. calisma\_alani

2. INITIAL LINE

3. LINES OF jtab [FROM idx1] [TO idx2] [USING KEY keyname]

1. calisma\_alani: Yapısal veri nesnesine atanan tek kayıtlık yapıdır.

2. INITIAL LINE: Tek kayıtlık boş bir satır eklemek için kullanılır.

3. LINES OF diger\_internal [FROM idx1] [TO idx2] [USING KEY anahtar\_alan]: Başka bir internal tablonun tüm satırlarını veya belirtilen satırlarını eklemek için kullanılır.

**itab\_pozisyon**

Tablonun hangi pozisyonuna satırların ekleneceğini belirtir. Kullanım şekilleri aşağıdakilerden birisi olabilir.

… {TABLE internal\_tablo}  
| { internal\_tablo INDEX indeks}  
| { internal\_tablo } … .

TABLE internal\_tablo: Aşağıdaki tablo kategorilerine göre eklenecek satır pozisyonu belirlenir.

1. Standart tablo: Birincil anahtar dikkate alınmadan her satır tablonun sonuna eklenir.

2. Sıralı tablo: Anahtar alanı değerine göre dâhilî tabloya sıralanmış satır yerleştirilir.

3. Karışık tablo: Birincil anahtar değerleri için hash yönetimi tarafından internal tabloya yeni satır eklenir.

itab INDEX indeks: Standart ve sıralı tablolar için çalışır. Eklenecek satırlar indeks ile belirtilen satırdan önce eklenir.

internal\_tablo: Sadece döngü içerisinde aynı tablo kullanılırken ve USING\_KEY kullanılmadığında kullanılabilir. Eklenecek her satır döngüdeki geçerli satırdan önce eklenir.

**APPEND**

İnternal tabloya bir veya birden fazla kayıt ekler. Birincil tablo indeksine bağlı olarak son kayıttan sonra yeni bir kayıt eklenir. Kullanımı aşağıdaki şekildedir.

APPEND eklenecek\_satir TO internal\_tablo [SORTED BY bilesen] [sonuc].

**eklenecek\_satir**

INSERT komutuyla aynıdır.

Aşağıdaki tablo kategorilerine göre eklenecek satır pozisyonu belirlenir.

1. Standart tablo: İnternal tablo içeriği kontrol edilmeden direkt olarak eklenir.

2. Sıralı tablo: Satırlar sıralama sırasına uyuyorsa ve birincil tablo anahtarına göre çift kayıt oluşturulmazsa eklenir.

3. Karışık tablo: Satır eklenemez.

sy-tabix sistem değişkeni son eklenen satır numarasını alır.

**COLLECT**

Bir çalışma alanının içeriğini bir satır olarak dâhilî tabloya ekler ve sayısal bileşenlerini aynı birincil tablo anahtarına sahip kayıtlarla toplar. Dâhilî tablo birincil anahtara göre aşağıdaki şekilde taranır. Kullanımı aşağıdaki şekildedir.

COLLECT calisma\_alani INTO internal\_tablo [sonuc].

Standart tablo için COLLECT kullanıma uygun değildir ve kullanılmamalıdır. Sıralı ve karışık tablolarda kullanılabilir.

Bir İnternal Tablo İçeriğini Diğerine Kopyalama

Bir internal tablonun içeriğini başka bir internal tabloya kopyalamak için aşağıdaki yapı kullanılabilir.

<internal\_tablo1>[] = <internal\_tablo2>[].

İnternal tablo isimden sonra gelen köşeli parantezler ([]) tablo içeriğini ifade eder. İnternal tabloların tipleri aynı olmalıdır.

**İnternal Tablodan Verileri Okumak**

Bir internal tablodan veriler üç şekilde okunabilir.

1. READ TABLE

2. LOOP AT, ENDLOOP ifadeleri arasında birden fazla kayıt sırasıyla okunabilir.

3. AT

**1. READ TABLE**

İnternal tablonun bir satırını okur. Kullanımı aşağıdaki şekildedir.

**READ TABLE internal\_tablo { table\_key**

**| free\_key**

**| index } result.**

tablo\_anahtari: Tablo anahtarı kullanılarak arama yapmak için kullanılır. Aramada kullanılacak anahtarlar çalışma alanında belirtilir veya TABLE KEY ile teker teker belirtilebilir. Kullanımı aşağıda şekildedir.

… { FROM calisma\_alani [USING KEY anahtar\_alan] }

| { WITH TABLE KEY [anahtar\_alan COMPONENTS]

{bilesen\_ismi1|(isim1)} = veri\_nesnesi1

{bilesen\_ismi2|(isim2)} = veri\_nesnesi2

… } … .

USING KEY anahtar\_alan: Aranacak kaydın anahtar alanları çalışma alanı içerisinde belirtilir.

WITH TABLE KEY [keyname COMPONENTS]: Aranacak kaydın anahtar alanların isimleri teker teker yazılarak belirtilir.

Free\_key: Anahtar alanı bağımsız arama yapmak için kullanılır. İki şekilde kullanımı vardır.

… WITH KEY { bilesen1 = veri\_nesnesi1 bilesen2 = veri\_nesnesi2 … [BINARY SEARCH] }

| { keyname COMPONENTS bilesen1 = veri\_nesnesi1 bilesen2 = veri\_nesnesi2 … } … .

bilesen1 = veri\_nesnesi1 bilesen2 = veri\_nesnesi2: Aranacak anahtar alanlar tek tek belirtilir.

[BINARY SEARCH]: Binary arama yapak için.

keyname COMPONENTS bilesen1 = veri\_nesnesi1 bilesen2 = veri\_nesnesi2: anahtar\_alan tablo anahtarı tanımlamak için kullanılır.

İndex: Satır numarası verilen kaydı okumak için kullanılır.

… INDEX indeks [USING KEY anahtar\_alan] … .

USING KEY anahtar\_alan:???

Result: Okunacak satırın aktarım davranışını belirtir.

· … INTO wa [transport\_options]

· … ASSIGNING <fs> [CASTING]

· … REFERENCE INTO dref

· … TRANSPORTING NO FIELDS

READ TABLE ifadesi sy-subrc, sy-tabix, sy-tfill, ve sy-tleng sistem değişkenlerini doldurur.

READ TABLE anahtar kelimesi kullanımı sonrası sy-subrc sistem değişkeni aşağıdaki şekilde değişir.

|  |  |
| --- | --- |
| sy-subrc | Anlamı |
| 0 | Arama başarılı. |
| 2 | Arama başarılı. Result içerisinde COMPARING kullanımına göre değişir. |
| 4 | Arama başarısız. |
| 8 | Arama başarısız. Eğer giriş binary arama ile bulundu ve tablonun sonuna geldi ise sy-tabix satır numarası + 1 olur. |

LOOP AT: İnternal tablodaki her kayıt için bir döngü bloğu çalıştırılır. LOOP ifadesi ile WHERE anahtar kelimesi kullanılarak döngü sayısı sınırlandırılabilir. Koşula uyan her kayıt sayısı kadar döngü oluşur. Kullanımı aşağıdaki şekildedir.

LOOP AT itab result [cond].

Kod\_blogu

ENDLOOP.

Result: Okunan satırın bilgi aktarma davranışı için kullanılır.

… { INTO wa }

| { ASSIGNING <fs> [CASTING] }

| { REFERENCE INTO dref }

| { TRANSPORTING NO FIELDS } … .

INTO calisma\_alani ile okunan kayıt bir çalışma alanına transfer edilir.

ASSIGNING <fieldsymbols> [CASTING] ile okunan kayıt bir field symbol’ e transfer edilir.

REFERENCE INTO veri\_referansi ile okunan kayıt bir referans nesnesine transfer edilir.

TRANSPORTING NO FIELDS ile kayıt herhangi bir veri nesnesine transfer edilmez. WHERE anahtar kelimesi ile birlikte kullanılabilir.

LOOP AT-ENDLOOP ifadesi sonrası sy-subrc sistem değişkeni aşağıdaki şekilde değişir.

|  |  |
| --- | --- |
| **sy-subrc** | **Anlamı** |
| 0 | Döngü en az bir defa çalıştı |
| 4 | Döngü çalışmadı |

sy-tfill ve sy-tleng alanları da dolar.

**AT**

Loop döngüsü içerisinde kontrol yapıları için kullanılır. AT ve ENDAT blokları kontrol kesmelerinde işlenen bildirimler tanımlar. Kontrol yapısı değiştiğinde, AT ifadesi hangi ifade bloğunun çalıştığı kontrol kesmesini belirler. Bu ifade blokları içerisinde SUM ifadesi sayısal değerlerin toplanmasında kullanılabilir.

LOOP AT itab result …

[AT FIRST.

…

ENDAT.]

[AT NEW comp1.

…

ENDAT.

[AT NEW comp2.

…

ENDAT.

[…]]]

[ … ]

[[[…]

AT END OF comp2.

…

ENDAT.]

AT END OF comp1.

…

ENDAT.]

[AT LAST.

…

ENDAT.]

ENDLOOP.

FIRST: Kontrol düzeyi dâhilî tablonun ilk satırında tanımlanır. Kontrol kesmesi ilk satır okunduğunda çalışır.

… {NEW}|{END OF} compi: Kontrol düzeyler compi de tanımlanan değeri aynı olan satır grubunun başlangıcında ve sonunda tanımlanır. Kontrol kesmeleri compi bileşeninin içeriği değiştiğinde veya başka bileşen compi’ nin solunda olan başka bir bileşen değiştiğinde çalışır.

…

LAST: Kontrol düzeyi dâhilî tablonun son satırında tanımlanır. Kontrol kesmesi son satır okunduğunda çalışır.

**İnternal Tablo İçeriğini Değiştirme**

**MODIFY**

İnternal tablodaki bir veya daha fazla kaydı değiştirmek için kullanılır. Kayıtlara indeks numarası veya anahtar alanları ile ulaşılır. Kullanımı aşağıdaki şekildedir.

MODIFY { itab\_line | itab\_lines }.

Anahtar alanların içeriğini değiştirme ile ilgili aşağıda belirtilen sınırlamalar vardır.

Sıralı ve hash tabloların anahtar alanları sadece okunabilir ve değiştirilmemelidir. Değiştirilmeye çalışılması hatalara neden olur.

İkincil tablo anahtarının, anahtar alanları ikincil indeks kullanıldığında okunabilir düzendedir. LOOP döngüleri ve USING KEY ekinden sonra ikincil indekste MODIFY kullanılan örnek içindir. Aksi durumda anahtar alanlar okunabilir düzende değildir.

MODIFY ifadesi sonrası sy-subrc sistem değişkeni aşağıdaki şekilde değişir.

|  |  |
| --- | --- |
| **sy-subrc** | **Anlamı** |
| 0 | En az bir kayıt değişti. |
| 4 | Belirtilen koşullara uygun kayıt bulunamadığından hiçbir kayıt değiştirilemedi. |

**DELETE**

İnternal tablodan bir veya daha fazla sayıda kayıt silmek için veya birbiri ardına sıralanmış çift satırları silmek kullanılır. Silinecek satırlar indeks numarası veya tablo anahtarları belirtilerek silinirler.

|  |  |
| --- | --- |
| **sy-subrc** | **Anlamı** |
| 0 | En az bir kayıt silindi. |
| 4 | Uygun satır bulunamadığından hiçbir kayıt silinemedi.  1. Kayıt tablo anahtarları ile aramada başarısız olduğundan,  2. belirtilen mantıksal ifade veya belirtilen kayıt indeksi tablo kayıt indeksinden büyük olduğundan  3. çift satırlar sıralı olmadığından bulunamadı. |

**SORT**

İnternal tabloyu sıralamak için kullanılır. Kullanımı aşağıdaki şekildedir.

SORT internal\_tablo [STABLE]

{ [ASCENDING|DESCENDING]

[AS TEXT]

[BY {comp1 [ASCENDING|DESCENDING] [AS TEXT]}

{ comp2 [ASCENDING|DESCENDING] [AS TEXT]} … ] }

| { [BY (otab)] }

STABLE: Kararlı sıralama için kullanılır. …

ASCENDING|DESCENDING: Sıralama yönünü belirtir. ASCENDING artan, DESCENDING azalan yönde sıralama yapar. Eğer bu ek kullanılmaz ise dâhilî tablo artan düzende sıralanır. BY ekinden sonra bileşen ismi belirtilerek belirtilen bileşenlere göre münferit olarak sıralama yapılabilir.

AS TEXT: text-like bileşenler için bölgesel code page e göre sıralama yapmak için kullanılır.

BY compi [ASCENDING|DESCENDING] [AS TEXT]: Kendisinden sonra belirtilen bileşenlere göre sıralama yapmak için kullanılır.

BY (otab): Dâhilî tabloyu, otab dâhilî tablosunda belirtilen bileşenlere göre sıralar.

Otab ABAP Dictionary’ de tanımlı ABAP\_SORTORDER\_TAB tipinde sıralı bir tablodur. Satır tipi ABAP\_SORTORDER’ dır. Bileşenler aşağıdaki gibidir.

NAME of type SSTRING:

sıralamada kullanılacak bileşenler.

DESCENDING of the type CHAR of a length 1:

Bileşenin sıralama yönünü belirtmek için kullanılır. Boş ise artan düzende sıralama yapılır. Eğer “X” değeri var ise azalan düzende sıralama yapılır.

ASTEXT of type CHAR with length 1:

Eğer “X” değeri bileşen için tanımlanmış ise, sıralama AS TEXT eki kullanılarak yapılır.

**İnternal Tabloda Arama İşlemleri**

**FIND IN TABLE**

İnternal tablo içerisinde bir örneğe göre bayt-karakter dizisi veya metin karakter dizisine göre arama yapar. İnternal tablo ikincil tablo anahtarı olmayan standart tablo olmalıdır.

FIND [{FIRST OCCURRENCE}|{ALL OCCURRENCES} OF] pattern

IN TABLE internal\_tablo [tablo\_araligi]

[IN {CHARACTER|BYTE} MODE]

[find\_options].

tablo\_araligi: Dâhilî tablodaki arama aralığını daraltmak için kullanılır.

IN {CHARACTER|BYTE} MODE: Satırın karakter-benzeri (character-like) veya byte-benzeri (byte-like) olup olmadığını belirtir.

|  |  |
| --- | --- |
| **sy-subrc** | **Anlamı** |
| 0 | Arama aralığında, aranan örnek en az bir defa bulundu |
| 4 | Arama aralığında aranan örnek bulunamadı |

**REPLACE IN TABLE**

Dâhilî tablo örnekte verilen byte-karakter dizisi veya metin karakter dizisine göre satır satır aranır ve bulunan bölge belirtilen veri nesnesi ile değiştirilir.

|  |  |
| --- | --- |
| **sy-subrc** | **Anlamı** |
| 0 | Aranan dizi, belirtilen içerik ile değiştirildi ve tüm sonuç tablo satırında mevcut |
| 2 | Aranan dizi, belirtilen içerik ile değiştirildi ve değişikliğin sonucu en az bir tablo satırda kesildi |
| 4 | Aranan dizi bulunamadı |
| 8 | Örnek veya yeni veri nesnesi çevrilebilir double-byte karakter içermiyor. |

**İNTERNAL TABLO İŞLEMLERİ**

**REFRESH**

İnternal tablonun içeriğini (tüm kayıtları) siler. Eğer internal tablonun başlık satırı (header line) varsa, başlık satırı temizlenmez.

REFRESH <dâhilî\_tablo>.

**CLEAR**

Dâhilî tablonun başlık satırını (header line) temizler.

CLEAR <dâhilî\_tablo>.

Dâhili tablo içerisindeki tüm kayıtları silmek için aşağıdaki şekilde kullanılabilir.

CLEAR <dâhilî\_tablo>[].

**FREE**

Dâhilî tablo içerisindeki tüm kayıtları siler ve internal tablonun hafızada işgal ettiği alanı serbest bırakır.

**DESCRIBE TABLE**

İnternal tabloya ait bilgileri elde etmek için kullanılır.

DESCRIBE TABLE <dâhilî\_tablo> [LINES <degisken>].

İnternal tablonun kaç kayıttan oluştuğu gibi bilgiler elde edilebilir. Kullanımı sonrası bazı sistem değişkenleri aşağıdaki şekilde değişir.

· SY-TFILL: İnternal tablonun kayıt sayısı.

· SY-TLENG: İnternal tablonun kayıt genişliği.

**DELETE ADJACENT DUPLICATE ENTRIES FROM**

İnternal tablodaki ardışık olarak sıralanmış aynı değere sahip kayıtları siler.

DELETE ADJACENT DUPLICATE ENTRIES FROM <dâhilî\_tablo>

                            [COMPARING <alan1> <alan2> ... |ALL FIELDS].

· COMPARING <alan1> <alan2>: Belirtilen alan değerlerine göre karşılaştırma yaparak kayıtlar silinir.

· COMPARING ALL FIELDS: İnternal tablonun tüm alan değerleri karşılaştırılarak kayıtlar silinir.

COMPARING ALL FIELDS varsayılan seçenektir.