**VERİTABANI ERİŞİMLERİ**

SAP üzerindeki bütün uygulama verileri ilişkisel veritabanı tablolarda tutulur. Tablolarına ulaşmak için SQL dili kullanılır. SAP iki tür SQL erişimine izin verir.

· Native SQL

· Open SQL

**NATIVE SQL**

Native Sql kullanılarak direkt olarak belirli bir veritabanı sistemine erişilebilir. Erişilen veritabanı üzerinde SQL komutları çalıştırılır. Kullanılan SQL komutları erişilen veritabanı sistemine özgüdür. Komutlar veritabanı sistemlerine (Oracle, MS SQL,…) göre farklılık gösterir. ABAP kodunda SQL komutları EXEC SQL ve ENDEXEC ABAP ifadeleri arasına yazılır.

EXEC SQL.

kod\_blogu

ENDEXEC.

**OPEN SQL**

SAP’ nin kurulu olduğu veritabanında işlem yapmak için kullanılır. Kullanılan ifadelerle tablolara kayıt ekleme, değiştirme veya silme işlemleri yapılabilir.

Open SQL veritabanına direkt olarak erişemez. Tablolara ulaşmadan önce veritabanı arayüz katmanına erişir. Arayüz katmanında OPEN SQL komutları erişilen veritabanı sistemine özgü SQL komutlara çevrilir. Veritabanı arayüz katmanı Open SQL’i SAP’ nin kurulduğu veritabanı sistemine komutları gönderir.Bu kullanım şekliyle Open SQL platform bağımsız bir veritabanı erişimi sağlar.

**OPEN SQL İFADELERİ**

Bazı Open SQL ifadeleri şunlardır.

|  |  |
| --- | --- |
| **Open SQL Komutu** | **Tanım** |
| SELECT | Veritabanı tablosundan verileri okur |
| INSERT | Veritabanı tablosuna kayıt ekler. |
| UPDATE | Veritabanı tablosundaki kayıtları değiştir. |
| MODIFY | Veritabanı tablosundaki kayıtları değiştirir, kayıtlar mevcut değilse ekler. |
| DELETE | Veritabanı tablosundan kayıt siler. |

Open SQL ifadeleri çalıştırıldıktan sonra SY-SUBRC ve SY-DBCNT sistem değişkenleri değişir.

|  |  |
| --- | --- |
| **Sistem Değişkeni** | **Anlamı** |
| sy-subrc | Yapılan işlem başarılı ise sıfır (0), değil ise sıfır dışında bir değer atanır. |
| sy-dbcnt | İşlem yapılan kayıt sayısı atanır. |

**SELECT**

Bir veya daha fazla tablodan veri okumak için kullanılır. Okunan veriler bir veri nesnesine (çalışma alanı veya internal tabloya) atanır. Kullanımı aşağıdaki gibidir.

SELECT sonuc

FROM kaynak

INTO|APPENDING hedef

[[FOR ALL ENTRIES IN internal\_tablo] WHERE sql\_kosulu]

[GROUP BY grup] [HAVING grup\_kosulu]

[ORDER BY siralama\_anahtari].

Kod\_blogu

[ENDSELECT].

SELECT ifadesi okunan verileri hedefe aktarırken bir döngü oluşturur. Bu döngü SELECT ifadesi ile başlar ENDSELECT ifadesi ile sonlanır. Her okunan kayıt hedef ile belirtilen veri nesnesine kopyalanır.

Döngü kullanımı olmadan da kayıtlar veri nesnelerine aktarılabilir. Bu kullanım türünde ENDSELECT kullanılmaz. Bu kullanımlar şunlardır.

· INTO … TABLE veya APPENDING … TABLE kullanıldığı ifadeler. PACKAGE SIZE eki kullanılmaz ise okunan kayıtlar direkt olarak internal tabloya aktarılır.

· SINGLE eki kullanılarak tablodan tek kayıt okunacak ise.

· Tablodan çekilecek alanların \* kullanıldığı ve GROUP BY ekinin kullanılmadığı ifadeler.

sonuc: Okunacak kayıt sayısı veya alan bilgilerini içerir. Kullanım şekillerine göre kayıt sayısı veya alanlar farklılık gösterir.

· Okunacak verileri tanımlar. Bir veya daha fazla kayıt olabilir. SINGLE eki kullanılır ise tablodan tek kayıt okunur.

· Birden fazla kayıt içeren okuma işlemlerinde DISTINCT eki aynı olan satırlar eler.

· \* kullanılarak tüm sütunların seçilmesi sağlanır.

· Seçilecek alanlar birer birer belirtilebilir. Alan isimleri yazılır. AS eki kullanılarak okunan alan isimleri değiştirilebilir. Alanlarla ilgiil yığın fonksiyonlar ve DISTINCT eki kullanılabilir.

· Parantez içerisinde bir değişken ile dinamik olarak alanlar belirtilebilir.

kaynak: ABAP Dictonary’ de tanımlı tablo veya görünüm olabilir. Aşağıdaki ekler kullanılır.

· veritb\_tablosu [AS takma\_ad] ile tablo ismine bir takma ad tanımlanır.

· Join ifadesi ile birden fazla tablodan veya görünümden veri okunabilir.

· (veritb\_secimleri) [AS takma\_ad] parantez kullanılarak tablo veya görünümden kayıt seçim ifadeleri veri nesnesi ile belirtilebilir.

· CLIENT SPECIFIED: Otomatik üst birim seçimini devre dışı bırakıp, istenilen üst birimden veri çekmek için kullanılır. Parametre kullanılır ise WHERE koşulunda üst birim belirtilebilir.

· UP TO n ROWS: Sonuç kümesindeki kayıt sayısını sınırlar. n değeri için i tipinde bir tamsayı belirtilmelidir.

· … BYPASSING BUFFER: Kayıtların tamponlama (SAP buffering) kullanılmadan direkt veritabanından okunmasını sağlar.

hedef: Kaydın aktarım bilgisini tanımlar. Bir çalışma alanı, veri nesnesi veya bir internal tablo olabilir. Kullanım şekilleri aşağıdaki gibidir.

· INTO [CORRESPONDING FIELDS OF] calisma\_alani: Kayıtlar çalışma alanına aktarılır. INTO kullanılmış ise tablo alanları ve çalışma alanı alanları aynı olmalıdır. CORRESPONDING FIELDS OF kullanılarak tablo ve çalışma alanı alanları farklı olsa bile uygun bileşenler çalışma alanına aktarılır.

· INTO (veri\_nesnesi1, veri\_nesnesi2, … ): Tablodan seçilen alanlar veri nesnelerine aktarır.

· INTO|APPENDING [CORRESPONDING FIELDS OF] TABLE internal\_tablo [PACKAGE SIZE n]: INTO ile internal tablodaki kayıtlar silinir ve yeni kayıtlar eklenir. APPENDING ile mevcut kayıtlar korunur yenileri eklenir. PACKAGE SIZE n eki ile kayıtlar internal tabloya n ile belirtilen paket büyüklükleri ile eklenir. Bu ek ENDSELECT anahtar kelimesi ile kullanılır.

[FOR ALL ENTRIES IN internal\_tablo] WHERE sql\_kosulu … ..: İnternal tablodaki tüm kayıtlar için sorgulama yapılır. İnternal tablonun alanları WHERE ifadesinde koşul olarak kullanılabilir.

GROUP BY { {alan1 alan2 …} | (sutun\_bilgileri) } … .: Belirtilen sütunlarda aynı olan kayıtları bir satır kayıt olacak şekilde gruplar. Alanlar teker teker veya sutun\_bilgileri veri nesnesine atanarak dinamik olarak belirtilebilir.

HAVING sql\_kosulu… .: Gruplanmış kayıt sayısını sınırlamak için kullanılır.

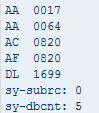
ORDER BY: Okunan kayıtları belirtilen alanlara göre sıralar. PRIMARY KEY eki ile tek bir tablodan okunan kayıtları anahtar alanlarına göre sıralar. Alan isimleri teker teker belirtilerek de sıralama işlemi yapılabilir. Teker teker belirtildiğinde ASCENDING veya DESCENDING ekleri ile azalan veya artan sıralama yapılabilir. Parantez içerisinde sıralama bilgileri veri nesnesinde belirtilerek de sıralama yapılabilir.

|  |  |
| --- | --- |
| **sy-subrc** | **Anlamı** |
| 0 | Kayıt okuma başarılı ve bütün değerler ABAP veri nesnesine aktarıldı veya SELECT ENDSELECT ifadeleri arasında en az bir kayıt işlendi. |
| 4 | Kayıt bulunamadı. |
| 8 | WHERE ifadesinde anahtar alanlar tam belirtilmedi ve FOR UPDATE eki kullanıldı. |

sy-dbcnt sistem değişkeni aktarılan kayıt sayısını alır.

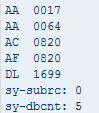
Örnek: SELECT ve ENDSELECT ifadeleri arasında SPFLI tablosunun ilk 5 kaydını yazdıran örnek.

DATA gs\_spfli TYPE spfli.  
SELECT \* FROM spfli INTO gs\_spfli UP TO 5 ROWS.  
WRITE:/ gs\_spfli-carrid, gs\_spfli-connid.  
ENDSELECT.  
WRITE: / ‘sy-subrc:’, sy-subrc LEFT-JUSTIFIED,  
/ ‘sy-dbcnt:’, sy-dbcnt LEFT-JUSTIFIED.



Örnek: SPFLI tablosundaki ilk 5 kaydı gt\_spfli internal tablosuna kopyalayarak ekrana yazan örnek.

DATA: gt\_spfli TYPE STANDARD TABLE OF spfli,  
gs\_spfli TYPE spfli.  
SELECT \* FROM spfli UP TO 5 ROWS  
INTO CORRESPONDING FIELDS OF TABLE gt\_spfli.  
LOOP AT gt\_spfli INTO gs\_spfli.  
WRITE:/ gs\_spfli-carrid, gs\_spfli-connid.  
ENDLOOP.  
WRITE: / ‘sy-subrc:’, sy-subrc LEFT-JUSTIFIED,  
/ ‘sy-dbcnt:’, sy-dbcnt LEFT-JUSTIFIED.



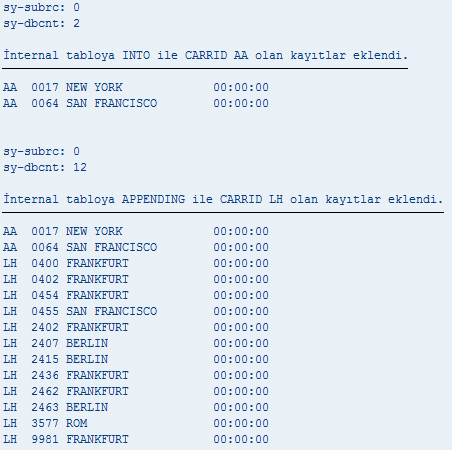
Örnek: SINGLE ile SPFLI tablosundan CARRID alanı AA ve CONNID alanı 0017 olan tek kaydı seçen ve ekrana yazdıran örnek.

DATA gs\_spfli TYPE spfli.  
SELECT SINGLE \* FROM spfli INTO gs\_spfli WHERE carrid = ‘AA’ AND  
connid = ‘0017’.  
WRITE:/ gs\_spfli-carrid, gs\_spfli-connid, gs\_spfli-cityfrom.  
WRITE: / ‘sy-subrc:’, sy-subrc LEFT-JUSTIFIED,  
/ ‘sy-dbcnt:’, sy-dbcnt LEFT-JUSTIFIED.

http://iuyanik.com/abaptr/wp-content/uploads/2013/11/img_527566d8bd874.png

Örnek: INTO VE APPENDING kullanımını gösteren örnek. SPFLI tablosundan CARRID alanı AA olan kayıtlar internal tabloya eklenir. Sadece CARRID, CONNID ve CITYFROM alanları internal tabloya eklenir. Ardından APPENDING ile internal tablodaki kayıtlara ek olarak CARRID alanı LH olan kayıtlar ilave edilir ve ekrana yazdırılır. Ekrana yazdırılan alanlar içerisinde DEPTIME alanıda vardır. Bu alan seçimde belirtilmediği için internal tablo eklenmediği için ekrana alanın varsayılan değeri yazdırılır. SKIP satır atlamak için, ULINE yatay çizgi çizmek için kullanılmıştır.

DATA: gt\_spfli TYPE STANDARD TABLE OF spfli,  
gs\_spfli TYPE spfli.  
SELECT carrid connid cityfrom  
FROM spfli  
INTO CORRESPONDING FIELDS OF TABLE gt\_spfli  
WHERE carrid = ‘AA’.  
WRITE: / ‘sy-subrc:’, sy-subrc LEFT-JUSTIFIED,  
/ ‘sy-dbcnt:’, sy-dbcnt LEFT-JUSTIFIED.  
SKIP.  
WRITE / ‘İnternal tabloya INTO ile CARRID AA olan kayıtlar eklendi.’.  
ULINE /1(58).  
LOOP AT gt\_spfli INTO gs\_spfli.  
WRITE:/ gs\_spfli-carrid,   gs\_spfli-connid,  
gs\_spfli-cityfrom, gs\_spfli-deptime.  
ENDLOOP.  
SKIP 2.  
SELECT carrid connid cityfrom  
FROM spfli  
APPENDING CORRESPONDING FIELDS OF TABLE gt\_spfli  
WHERE carrid = ‘LH’.  
WRITE: / ‘sy-subrc:’, sy-subrc LEFT-JUSTIFIED,  
/ ‘sy-dbcnt:’, sy-dbcnt LEFT-JUSTIFIED.  
SKIP.  
WRITE /  ‘İnternal tabloya APPENDING ile CARRID LH olan kayıtlar eklendi.’.  
ULINE /1(63).  
LOOP AT gt\_spfli INTO gs\_spfli.  
WRITE:/ gs\_spfli-carrid,   gs\_spfli-connid,  
gs\_spfli-cityfrom, gs\_spfli-deptime.  
ENDLOOP.



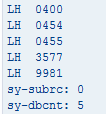
Örnek: SPFLI tablosundaki ilk 5 kayıda göre SFLIGHT tablosundan okuma yapan örnek.

DATA: gt\_spfli   TYPE STANDARD TABLE OF spfli,  
gt\_sflight TYPE STANDARD TABLE OF sflight,  
gs\_sflight TYPE sflight.  
SELECT \* FROM spfli UP TO 5 ROWS  
INTO CORRESPONDING FIELDS OF TABLE gt\_spfli.  
IF gt\_spfli[] IS NOT INITIAL.  
SELECT \* FROM sflight  
INTO CORRESPONDING FIELDS OF TABLE gt\_sflight  
FOR ALL ENTRIES IN gt\_spfli  
WHERE carrid = gt\_spfli-carrid AND  
connid = gt\_spfli-connid.  
ENDIF.  
LOOP AT gt\_sflight INTO gs\_sflight.  
WRITE:/ gs\_sflight-carrid, gs\_sflight-connid.  
ENDLOOP.  
WRITE: / ‘sy-subrc:’, sy-subrc LEFT-JUSTIFIED,  
/ ‘sy-dbcnt:’, sy-dbcnt LEFT-JUSTIFIED.



Örnek: INNER JOIN eki kullanılarak SPFLI tablosundaki CARRID alanı LH değerine sahip kayıtları SFLIGHT tablusundan GT\_SFLIGHT internal tablosuna ekleyen ve ekrana yazdıran örnek.

DATA: gt\_spfli   TYPE STANDARD TABLE OF spfli,  
gt\_sflight TYPE STANDARD TABLE OF sflight,  
gs\_sflight TYPE sflight.  
SELECT \* FROM spfli AS a  
INNER JOIN sflight AS b ON ( a~carrid = b~carrid AND  
a~connid = b~connid )  
INTO CORRESPONDING FIELDS OF TABLE gt\_sflight  
WHERE a~carrid = ‘LH’.  
LOOP AT gt\_sflight INTO gs\_sflight.  
WRITE:/ gs\_sflight-carrid, gs\_sflight-connid.  
ENDLOOP.  
WRITE: / ‘sy-subrc:’, sy-subrc LEFT-JUSTIFIED,  
/ ‘sy-dbcnt:’, sy-dbcnt LEFT-JUSTIFIED.



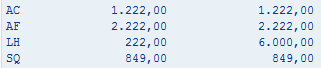
Örnek: DISTINCT kullanım örneği.

DATA: gt\_spfli TYPE STANDARD TABLE OF spfli,  
gs\_spfli TYPE spfli.  
SELECT DISTINCT cityto  
FROM spfli  
INTO CORRESPONDING FIELDS OF TABLE gt\_spfli  
WHERE cityfrom = ‘BERLIN’.  
LOOP AT gt\_spfli INTO gs\_spfli.  
WRITE gs\_spfli-cityto.  
ENDLOOP.

http://iuyanik.com/abaptr/wp-content/uploads/2013/11/img_527567391433e.png

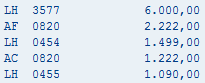
Örnek: SFLIGHT tablosundaki CARRID alanına göre minimum ve maksimum değerlerini yazan örnek.

DATA: BEGIN OF gs\_carrid,  
carrid TYPE sflight-carrid,  
min    TYPE sflight-price,  
max    TYPE sflight-price,  
END OF gs\_carrid.  
SELECT carrid MIN( price ) MAX( price )  
FROM sflight  
INTO (gs\_carrid-carrid, gs\_carrid-min, gs\_carrid-max)  
GROUP BY carrid.  
WRITE: / gs\_carrid-carrid, gs\_carrid-min, gs\_carrid-max.  
ENDSELECT.



Örnek: SFLIGHT tablosundaki ilk 5 kaydı GT\_SFLIGHT internal tablosuna koplayan ve PRICE alanına göre azalan şekilde sıralayan örnek.

DATA: gt\_sflight TYPE STANDARD TABLE OF sflight,  
gs\_sflight TYPE sflight.  
SELECT \* FROM sflight UP TO 5 ROWS  
INTO CORRESPONDING FIELDS OF TABLE gt\_sflight  
ORDER BY price DESCENDING.  
LOOP AT gt\_sflight INTO gs\_sflight.  
WRITE:/ gs\_sflight-carrid, gs\_sflight-connid, gs\_sflight-price.  
ENDLOOP.



**INSERT**

Veritabanı tablosuna kayıt eklemek için kullanılır.

INSERT { {INTO hedef VALUES calisma\_alani}

| { hedef FROM calisma\_alani|{TABLE internal\_tablo} } }.

hedef: Kayıtların ekleneceği tablo veya görünümdür. Tablo ismi direkt olarak belirtilebilir veya parantez içerisinde değişken ile belirtilebilir. CLIENT SPECIFIED otomatik üst birim seçimini devre dışı bırakır. Kayıt ekleme işlemi için yeni belirtilen üst birim kullanılır.

VALUES’ dan sonra veya FROM’ dan sonra çalışma alanı ismi yazarak, çalışma alanındaki kayıt eklenebilir veya TABLE ile belirtilen internal tablodaki kayıtlar eklenebilir. TABLE’ dan sonra ACCEPTING DUPLICATE KEYS eki kullanılarak veritabanı tablosunda kayıt daha önce mevcut ise hata alınmaması sağlanır. Sadece veritabanı tablosunda mevcut olmayan kayıtlar eklenir.

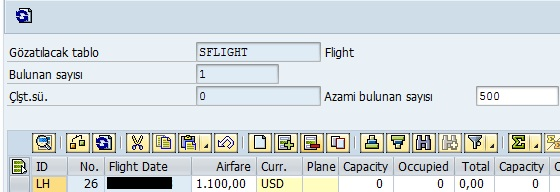
Örnek: GS\_SFLIGHT çalışma alanındaki kaydı SFLIGHT tablosuna ekleyen örnek.

DATA gs\_sflight TYPE sflight.  
CLEAR gs\_sflight.  
gs\_sflight-carrid   = ‘LH’.  
gs\_sflight-connid   = ‘0026’.  
gs\_sflight-fldate   = sy-datum.  
gs\_sflight-price    = 1100.  
gs\_sflight-currency = ‘USD’.  
INSERT into sflight values gs\_sflight.  
COMMIT WORK AND WAIT.

WRITE: / ‘sy-subrc:’, sy-subrc LEFT-JUSTIFIED,  
/ ‘sy-dbcnt:’, sy-dbcnt LEFT-JUSTIFIED.

http://iuyanik.com/abaptr/wp-content/uploads/2013/11/img_5275677652688.png

Eklenen kayıtları SE16N işlem kodu ile görüntüleyebilirsiniz.



Örnek: Internal tablodan kayıt ekleyen örnek.

Örnek: SFLIGHT tablosuna GT\_SFLIGHT internal tablosundaki kayıtları ekleyen örnek.

DATA: gt\_sflight TYPE STANDARD TABLE OF sflight,  
gs\_sflight TYPE sflight.  
CLEAR gs\_sflight.  
gs\_sflight-carrid   = ‘AA’.  
gs\_sflight-connid   = ‘0026’.  
gs\_sflight-fldate   = sy-datum.  
gs\_sflight-price    = 1000.  
gs\_sflight-currency = ‘USD’.  
APPEND gs\_sflight TO gt\_sflight.  
CLEAR gs\_sflight.  
gs\_sflight-carrid   = ‘AF’.  
gs\_sflight-connid   = ‘0026’.  
gs\_sflight-fldate   = sy-datum.  
gs\_sflight-price    = 1200.  
gs\_sflight-currency = ‘USD’.  
APPEND gs\_sflight TO gt\_sflight.  
INSERT sflight FROM TABLE gt\_sflight.  
COMMIT WORK AND WAIT.  
WRITE: / ‘sy-subrc:’, sy-subrc LEFT-JUSTIFIED,  
/ ‘sy-dbcnt:’, sy-dbcnt LEFT-JUSTIFIED.

http://iuyanik.com/abaptr/wp-content/uploads/2013/11/img_527567974ccb0.png

**UPDATE**

Veritabanı tablosundaki kayıtları güncellemek için kullanılır. Kullanımı aşağıdaki gibidir.

UPDATE hedef kaynak.

hedef: Kayıtların güncelleneceği tablo veya görünümdür. Tablo ismi direkt olarak belirtilebilir veya parantez içerisinde değişken ile belirtilebilir. CLIENT SPECIFIED otomatik üst birim seçimini devre dışı bırakır. Kayıt güncelleme işlemi için yeni belirtilen üst birim kullanılır.

kaynak: Değişecek kayıtları belirtir. SET sonrası değişecek alanlar belirtilebilir. SET ile WHERE kullanılarak hangi kayıtların değişeceği belirtilir. Kayıtları güncellemenin bir diğer yolu FROM kullanılarak çalışma alanı veya internal tablo kullanmaktır. Çalışma alanı kullanılarak tablodaki bir kaydın güncellenmesi sağlanabilir. Tablodaki değişecek kaydın anahtar alanları çalışma alanında tanımlanmalıdır. İnternal tablodan güncelleme yapmak için FROM dan sonra TABLE kullanılmalıdır. İnternal tablodaki kayıtlar teker teker güncelleme için kullanılır.

Örnek: SFLIGHT tablosundaki WHERE koşulunda belirtilen koşula göre bulunan kayıtların PRICE alanını güncelleyen örnek.

UPDATE sflight SET price = 6000  
WHERE carrid = ‘LH’   AND  
connid = ‘0026’ AND  
fldate = sy-datum.  
COMMIT WORK AND WAIT.  
WRITE: / ‘sy-subrc:’, sy-subrc LEFT-JUSTIFIED,  
/ ‘sy-dbcnt:’, sy-dbcnt LEFT-JUSTIFIED.

http://iuyanik.com/abaptr/wp-content/uploads/2013/11/img_527567a39d8f4.png

Örnek: SFLIGHT tablosundaki kaydın PRICE alanını GS\_SFLIGHT çalışma alanındaki kayda göre güncelleyen örnek.

DATA gs\_sflight TYPE sflight.  
CLEAR gs\_sflight.  
gs\_sflight-carrid   = ‘LH’.  
gs\_sflight-connid   = ‘0026’.  
gs\_sflight-fldate   = sy-datum.  
gs\_sflight-price    = 3000.  
gs\_sflight-currency = ‘USD’.  
UPDATE sflight FROM gs\_sflight.  
COMMIT WORK AND WAIT.  
WRITE: / ‘sy-subrc:’, sy-subrc LEFT-JUSTIFIED,  
/ ‘sy-dbcnt:’, sy-dbcnt LEFT-JUSTIFIED.

http://iuyanik.com/abaptr/wp-content/uploads/2013/11/img_527567ae21fde.png

Örnek: SFLIGHT tablosundaki kayıtlarıın PRICE alanını GT\_SFLIGHT internal tablosundaki kayıtlara göre güncelleyen örnek.

DATA: gt\_sflight TYPE STANDARD TABLE OF sflight,  
gs\_sflight TYPE sflight.  
CLEAR gs\_sflight.  
gs\_sflight-carrid   = ‘AA’.  
gs\_sflight-connid   = ‘0026’.  
gs\_sflight-fldate   = sy-datum.  
gs\_sflight-price    = 4000.  
gs\_sflight-currency = ‘USD’.  
APPEND gs\_sflight TO gt\_sflight.

CLEAR gs\_sflight.  
gs\_sflight-carrid   = ‘AF’.  
gs\_sflight-connid   = ‘0026’.  
gs\_sflight-fldate   = sy-datum.  
gs\_sflight-price    = 5000.  
gs\_sflight-currency = ‘USD’.  
APPEND gs\_sflight TO gt\_sflight.  
MODIFY sflight FROM TABLE gt\_sflight.  
COMMIT WORK AND WAIT.  
WRITE: / ‘sy-subrc:’, sy-subrc LEFT-JUSTIFIED,  
/ ‘sy-dbcnt:’, sy-dbcnt LEFT-JUSTIFIED.

http://iuyanik.com/abaptr/wp-content/uploads/2013/11/img_527567bb8c922.png

**MODIFY**

Veritabanı tablosundaki kayıtları değiştirir. Kayıtlar mevcut değil ise ekler. Kullanımı aşağıdaki gibidir.

MODIFY hedef FROM kaynak.

hedef: Kayıtların güncelleneceği tablo veya görünümdür. Tablo ismi direkt olarak belirtilebilir veya parantez içerisinde değişken ile belirtilebilir. CLIENT SPECIFIED otomatik üst birim seçimini devre dışı bırakır. Kayıt güncelleme işlemi için yeni belirtilen üst birim kullanılır.

kaynak: Değişecek/eklenecek kayıtları belirtir. Kayıtları güncellemek için çalışma alanı veya internal tablo kullanılabilir. Çalışma alanı kullanılarak tablodaki bir kaydın güncellenmesi sağlanabilir. Tablodaki değişecek kaydın anahtar alanları çalışma alanında tanımlanmalıdır. İnternal tablodan güncelleme yapmak için FROM dan sonra TABLE kullanılmalıdır. İnternal tablodaki kayıtlar teker teker güncelleme için kullanılır.

http://iuyanik.com/abaptr/wp-content/uploads/2013/11/img_527567ce93418.png

Örnek: SFLIGHT tablosundaki kaydın PRICE alanını GS\_SFLIGHT çalışma alanındaki kayda göre güncelleyen örnek.

DATA gs\_sflight TYPE sflight.  
CLEAR gs\_sflight.  
gs\_sflight-carrid   = ‘LH’.  
gs\_sflight-connid   = ‘0026’.  
gs\_sflight-fldate   = sy-datum.  
gs\_sflight-price    = 2000.  
gs\_sflight-currency = ‘USD’.  
MODIFY sflight FROM gs\_sflight.  
COMMIT WORK AND WAIT.  
WRITE: / ‘sy-subrc:’, sy-subrc LEFT-JUSTIFIED,  
/ ‘sy-dbcnt:’, sy-dbcnt LEFT-JUSTIFIED.

http://iuyanik.com/abaptr/wp-content/uploads/2013/11/img_527567da6a577.png

Örnek: SFLIGHT tablosundaki kayıtlarıın PRICE alanını GT\_SFLIGHT internal tablosundaki kayıtlara göre değiştiren örnek.

DATA: gt\_sflight TYPE STANDARD TABLE OF sflight,  
gs\_sflight TYPE sflight.  
CLEAR gs\_sflight.  
gs\_sflight-carrid   = ‘AA’.  
gs\_sflight-connid   = ‘0026’.  
gs\_sflight-fldate   = sy-datum.  
gs\_sflight-price    = 800.  
gs\_sflight-currency = ‘USD’.  
APPEND gs\_sflight TO gt\_sflight.  
CLEAR gs\_sflight.  
gs\_sflight-carrid   = ‘AF’.  
gs\_sflight-connid   = ‘0026’.  
gs\_sflight-fldate   = sy-datum.  
gs\_sflight-price    = 1500.  
gs\_sflight-currency = ‘USD’.  
APPEND gs\_sflight TO gt\_sflight.  
MODIFY sflight FROM TABLE gt\_sflight.  
COMMIT WORK AND WAIT.  
WRITE: / ‘sy-subrc:’, sy-subrc LEFT-JUSTIFIED,  
/ ‘sy-dbcnt:’, sy-dbcnt LEFT-JUSTIFIED.

http://iuyanik.com/abaptr/wp-content/uploads/2013/11/img_527567e60ac90.png

**DELETE**

Veritabanı tablosundan kayıtları siler. Kullanımı aşağıdaki gibidir.

DELETE { {FROM hedef [WHERE sql\_kosulu]}

| {hedef FROM kaynak} }.

hedef: Kayıtların silineceği tablodur. Tablo ismi direkt olarak belirtilebilir veya parantez içerisinde değişken ile belirtilebilir. CLIENT SPECIFIED otomatik üst birim seçimini devre dışı bırakır. Kayıt güncelleme işlemi için yeni belirtilen üst birim kullanılır. WHERE sql\_kosulu ile silinecek kayıtlar belirtilir.

kaynak: Silinecek kayıtları belirtir. Kayıtları silmek için çalışma alanı veya internal tablo kullanılabilir. Çalışma alanı kullanılarak tablodaki bir kaydın silinmesi sağlanabilir. Tablodaki silinecek kaydın anahtar alanları çalışma alanında tanımlanmalıdır. İnternal tablodan silmek için için FROM dan sonra TABLE kullanılmalıdır. İnternal tablodaki kayıtlar teker teker güncelleme için kullanılır.

Örnek: SFLIGHT tablosundaki FLDATE alanı günün tarihi olan kayıtları silen örnek.

DELETE FROM sflight WHERE fldate = sy-datum.  
COMMIT WORK AND WAIT.  
WRITE: / ‘sy-subrc:’, sy-subrc LEFT-JUSTIFIED,  
/ ‘sy-dbcnt:’, sy-dbcnt LEFT-JUSTIFIED.

http://iuyanik.com/abaptr/wp-content/uploads/2013/11/img_527567f5525e3.png

Örnek: Çalışma alanına göre SFLIGHT tablosundaki kayıtları silen örnek.

DATA gs\_sflight TYPE sflight.  
CLEAR gs\_sflight.  
gs\_sflight-carrid   = ‘LH’.  
gs\_sflight-connid   = ‘0026’.  
gs\_sflight-fldate   = sy-datum.  
DELETE sflight FROM gs\_sflight.  
COMMIT WORK AND WAIT.  
WRITE: / ‘sy-subrc:’, sy-subrc LEFT-JUSTIFIED,  
/ ‘sy-dbcnt:’, sy-dbcnt LEFT-JUSTIFIED.

Örnek: SFLIGHT tablosundaki kayıtları GT\_SFLIGHT internal tablosundaki kayıtlara göre silen örnek.

DATA: gt\_sflight TYPE STANDARD TABLE OF sflight,  
gs\_sflight TYPE sflight.  
CLEAR gs\_sflight.  
gs\_sflight-carrid   = ‘AA’.  
gs\_sflight-connid   = ‘0026’.  
gs\_sflight-fldate   = sy-datum.  
APPEND gs\_sflight TO gt\_sflight.  
CLEAR gs\_sflight.  
gs\_sflight-carrid   = ‘AF’.  
gs\_sflight-connid   = ‘0026’.  
gs\_sflight-fldate   = sy-datum.  
APPEND gs\_sflight TO gt\_sflight.  
DELETE sflight FROM TABLE gt\_sflight.  
COMMIT WORK AND WAIT.  
WRITE: / ‘sy-subrc:’, sy-subrc LEFT-JUSTIFIED,  
/ ‘sy-dbcnt:’, sy-dbcnt LEFT-JUSTIFIED.

http://iuyanik.com/abaptr/wp-content/uploads/2013/11/img_5275680266d4b.png