

1.HASH TABLO NEDİR?

ABAP dilinde "hash tablo" terimi, bir veri yapısı türünü ifade eder. Hash tablolar, belirli bir anahtar değeri kullanarak veri hızlı bir şekilde erişmek için tasarlanmış veri yapılarıdır. Bu tür tablolar, anahtar değerine dayalı olarak veri erişimini optimize eder ve performans avantajları sağlar.

ABAP dilinde, hash tabloları "HASHED TABLE" anahtar kelimesiyle tanımlanır. Örnek bir ABAP kodu ile hash tablosu tanımı şu şekilde olabilir:

```
DATA: BEGIN OF hash_table OCCURS 0,
```

```
    key   TYPE string,
```

```
    value TYPE string,
```

```
END OF hash_table.
```

```
DATA: wa TYPE hash_table.
```

* Hash tablosuna veri ekleme

```
wa-key = 'anahtar_degeri'.
```

```
wa-value = 'deger'.
```

```
INSERT wa INTO TABLE hash_table.
```

* Hash tablosundan veri okuma

```
READ TABLE hash_table WITH KEY key = 'anahtar_degeri' TRANSPORTING NO  
FIELDS.
```

```
IF sy-subrc = 0.
```

```
    WRITE: / 'Değer:', wa-value.
```

```
ELSE.
```

```
    WRITE: / 'Belirtilen anahtar değeri bulunamadı.'.
```

```
ENDIF.
```

2. TABLE ve VIEW

ABAP dilinde "view" ve "table" (tablo) terimleri, veri saklama ve erişme konseptlerini ifade eden iki farklı kavramı temsil eder. İşte bu iki kavram arasındaki temel farklar:

1. **Tablo (Table):**

- Tablo, genellikle bir veritabanında verileri saklamak için kullanılan yapıdır.
- Her bir tablo, bir veritabanı nesnesidir ve belirli bir veri modelini temsil eder.
- Bir tablo, satırlar ve sütunlardan oluşan bir veri kümesini içerir. Satırlar, belirli bir konsepti veya nesneyi temsil eden kayıtları içerirken, sütunlar her bir kaydın özelliklerini ifade eden alanlardır.
- ABAP dilinde, tablolar "TABLES" veya "DATA" anahtar kelimeleri kullanılarak tanımlanır.

2. **View:**

- View, bir veya daha fazla tablonun belirli alanlarını içeren ve bu tablolara daha kolay erişim sağlayan bir yapıdır.
- View, genellikle veri modelini basitleştirmek veya belirli bir kullanım senaryosuna uygun hale getirmek amacıyla kullanılır.
- View'lar, genellikle JOIN işlemleri kullanılarak birden çok tabloyu birleştirip belirli bir işlevsel görünümü temsil eder.
- View'lar, tablolara bir arayüz sağlar ve genellikle veritabanında karmaşık sorgulara olan ihtiyacı azaltır.
- ABAP dilinde, view'lar "DATA" ve "TYPES" anahtar kelimeleri kullanılarak tanımlanır.

Örneğin, bir müşteri siparişi veritabanı modelini ele alalım:

- **Tablo:** "Müşteri" tablosu, müşteri bilgilerini içerir.
- **View:** "Sipariş Bilgisi" view'u, müşteri bilgilerini ve bu müşterinin verdiği sipariş bilgilerini birleştirir.

Kısacası, tablolar verileri saklamak için kullanılırken, view'lar bu verilere erişimi basitleştirmek, özelleştirmek veya optimize etmek amacıyla kullanılır.

BKPF TABLOSUNA Z İLİ ALAN EKLEME

1. **Z alanını BKPF tablosuna eklemek:**

- Z alanını eklemek için SAP GUI'de Data Dictionary'yi kullanabilirsiniz. Bu örnekte, SAP GUI kullanılarak eklemeyi gösteriyorum.
- Transaction Code (TCode): **SE11** (Data Dictionary)
- **BKPF** tablosunu seçin ve "Change" butonuna tıklayın.
- "Append Structure" seçeneğini kullanarak yeni bir yapı ekleyin ve Z alanını ekleyin.

- Yapıyı kaydedin.

2. Doküman Oluşturma:

- Belge oluşturmak için **FB01** veya diğer uygun transaksiyon kodlarını kullanabilirsiniz. Bu örnekte, **FB01** kullanılacaktır.

REPORT ZBKPF_DOCUMENT_CREATION.

DATA: wa_bkpf TYPE BKPF,

wa_bseg TYPE BSEG,

lv_document_number TYPE BKPF-BELNR.

* Z alanına değer atama

wa_bkpf-z_field = 'Z_VALUE'.

* BKPF tablosuna veri ekleme

INSERT INTO BKPF VALUES wa_bkpf.

* Oluşturulan belgenin numarasını alma

SELECT SINGLE BELNR INTO lv_document_number FROM BKPF WHERE Z_FIELD = 'Z_VALUE'.

IF sy-subrc = 0.

WRITE: / 'Belge Numarası:', lv_document_number.

ELSE.

WRITE: / 'Belge oluşturulurken hata oluştu.'.

ENDIF.

Bu örnekte, **Z_FIELD** adlı Z alanına bir değer atanmış ve ardından **BKPF** tablosuna bu değerle birlikte bir kayıt eklenmiştir. Daha sonra, oluşturulan belge numarası **lv_document_number** değişkenine alınmıştır.

bapi nedir batch input nedir

1. **BAPI (Business Application Programming Interface):**

- BAPI, SAP sistemi içindeki işlemleri çağırmak ve dış sistemlerle etkileşimde bulunmak için kullanılan bir programlama arayüzüdür.
- Standart SAP işlemlerine ve veri nesnelere standart bir arabirim sağlar.
- Harici uygulamaların SAP sistemine entegrasyonunu kolaylaştırır. BAPI'lar aracılığıyla, harici sistemler SAP'deki işlemleri çağırabilir veya SAP sistemine veri gönderebilir.
- BAPI'lar, genellikle SAP Fonksiyon Modülleri (Function Modules) olarak uygulanır.

2. **Batch Input:**

- Batch Input, büyük miktarda veriyi SAP sistemine otomatik olarak girmek veya işlemek için kullanılan bir yöntemdir.
- SAP GUI üzerinden manuel bir kullanıcı girişi gibi davranır, ancak bu işlem otomatiktir.
- Önceden belirlenmiş bir veri seti, özel bir yükleme programı aracılığıyla sisteme yüklenir ve SAP sistemi tarafından işlenir.
- Örneğin, SAP Transaction (TCode) kayıtlarını büyük bir dosyadan içe aktarmak amacıyla kullanılabilir.
- Batch Input, genellikle özel programlar veya SAP standart programları aracılığıyla uygulanır.

Her iki konsept de SAP sistemlerinde veri alışverişi ve işlemleri kolaylaştırmak amacıyla kullanılır, ancak temelde farklıdırlar. BAPI, programlar arası iletişimi sağlayan bir arabirimdir, Batch Input ise otomatik veri girişi ve işleme için kullanılan bir yöntemdir. Her ikisi de SAP sisteminin etkileşimli ve otomatik özelliklerini destekleyerek iş süreçlerini iyileştirmeye yardımcı olur.

JOB NEDİR?

ABAP dilinde, SAP sisteminde belirli görevleri otomatikleştirmek için "job" adı verilen zamanlanmış veya planlanmış arka plan görevleri oluşturabilirsiniz. SAP Job'ları, belirli bir zaman diliminde veya belirli bir olayın gerçekleşmesi durumunda çalışan programları içerebilir

Job Definition (İş Tanımı):

- İlk olarak, bir iş tanımı (job definition) oluşturmanız gerekiyor. İş tanımı, SAP sistemi içindeki bir işin özelliklerini (örneğin, program adı, çalışma periyodu, başlama zamanı) içerir.

Job Program (İş Programı):

- Ardından, iş tanımınızı gerçekleştiren bir iş programı oluşturmanız gerekiyor. İş programınız, belirli bir görevi gerçekleştiren ABAP kodunu içerir.

Job Variant (İş Varyantı):

- İş programınızın çalıştırılma parametrelerini içeren bir iş varyantı oluşturmanız gerekebilir.

--

Job'ı Planlama:
<ul style="list-style-type: none">İş tanımını, iş programını ve iş varyantını oluşturduktan sonra, işi planlamak için SAP'nin arka plan iş yönetimi araçlarını kullanabilirsiniz (SM37 Transaction Code).

REPORT ZDAILY_JOB_DEFINITION.

DATA: jobname TYPE variant VALUE 'DAILY_JOB',

stepname TYPE variant VALUE 'STEP_1',

report TYPE variant VALUE 'ZDAILY_JOB_PROGRAM'.

CALL FUNCTION 'JOB_OPEN'

EXPORTING

jobname = jobname

stepname = stepname

action = 'CHECK'

release = 'X'

EXCEPTIONS

others = 1.

IF sy-subrc = 0.

CALL FUNCTION 'JOB_SUBMIT'

EXPORTING

jobcount = jobname

stepcount = stepname

jobname = report

EXCEPTIONS

others = 1.

ENDIF.

REPORT ZDAILY_JOB_PROGRAM.

WRITE 'Günlük iş çalıştı!'.

FATMA AKIN