

## **SAP BAPI(Business Application Programming Interface) Nedir?**

SAP BAPI, standart nesneleri okuma, oluşturma, güncelleme, silme ve birçok farklı görev için kullanılan bir fonksiyondur. Örnek verecek olur isek, SAP BAPI fonksiyonu ile SAP Sisteminde Malzeme oluşturabilir, güncelleyebilir veya bir malzemeye ait tüm ana verileri okuyabiliriz. Verdiğimiz örnekte yer alan her işlem için ayrı BAPI'ler kullanılmalıdır. BAPI işlem kodundan BAPI'lerin tüm özelliklerini inceleyebiliriz.

## **SAP BAPI'yi Nerede Kullanmalıyız?**

SAP Sistemlerinde yaptığınız geliştirmelerde veri okuma işlemlerinde eğer tek bir belge okunacak ise BAPI kullanmamız mantıklı olur. Ancak birden fazla belgeyi, örneğin sipariş okuyor isek BAPI kullanmamız çok önerilmez. Çünkü BAPI'ler birçok detay bilgiyi okuduğundan performans olarak vereceği dönüş hızları çok hızlı olmayacağından dolayı kullanıcılar için tatmin edici olmayabilir.

## **Batch Input Nedir?**

Batch Input, SAP sistemine toplu veri girişi yapmak için kullanılan bir yöntemdir. Bu yöntemde, dış sistemlerden veya kaynaklardan toplu veri alınır ve SAP sistemindeki veri alanlarına kaydedilir. Bu işlem, kullanıcı girişi olmadan yapılabilir ve otomatikleştirilmiş veri girişi için kullanışlıdır. Ancak kullanımı karmaşık olabilir ve veri uyumsuzlukları gibi sorunlara neden olabilir.

## **View Nedir?**

SAP ABAP'ta "View", bir veya daha fazla tablodan veri alarak bu verilere bir bakış açısı veya pencere sunan bir veritabanı nesnesidir. Bu, tablolar arasında ilişkili verileri bir araya getirerek kullanıcılara belirli bir bakış açısı sağlar.

**View'lar, genellikle aşağıdaki amaçlar için kullanılır:**

- **Veri Gizleme:** Belirli kullanıcı gruplarına veya rollerine belirli sütunları veya veri kümelerini göstermeme yeteneği. Hassas verilerin korunmasını sağlar.
- **Farklı Perspektifler:** Aynı veri setini farklı açılardan görmeyi sağlar. Örneğin, bir müşteri siparişleri tablosundan müşteri bilgileriyle birlikte ürün detaylarına erişmek gibi.
- **Karmaşık Sorguların Basitleştirilmesi:** Birçok tabloyu birleştirmek ve karmaşık sorguları basit bir şekilde sunmak için kullanılabilir. Bu sayede kullanıcılar, birden fazla tablodan veri almaksızın gereksinimlerini karşılayabilir.

View'lar, veritabanında depolanan verilere erişmek için kullanılır ve bu verileri işlemek, raporlamak veya belirli işlevleri gerçekleştirmek için kullanılabilir. Bu şekilde, veriye erişim ve işleme süreçleri daha düzenli ve daha kolay yönetilebilir hale gelir.

## **View ve SQL Cümlecikleri Arasındaki Farklar Nelerdir?**

View, bir veya birden fazla tabloyu veya diğer view'leri temsil eden sanal bir tablodur. Bu, belirli bir işlem veya sorgu için veri görünümü sağlar.

View'ler, bir kez tanımlandıktan sonra tekrar kullanılabilir. Bir sorgu olarak değil, adlandırılmış bir yapı olarak saklanır ve veriye dinamik bir şekilde erişimi sağlar.

Normal SQL cümlecikleri, belirli bir işlem veya sorgu için SQL dilinde ifade edilen komutlardır. SELECT, INSERT, UPDATE, DELETE gibi temel SQL komutları içerir.

Bu cümlecikler, genellikle veritabanında anlık işlemler için kullanılır. Belirli bir sorguyu yazıp çalıştırdıktan sonra sonuçlar elde edilir ve bu cümlecikler bir işlem olarak saklanmaz.

#### Farklar:

- **Yeniden Kullanılabilirlik:** View'ler, bir kez tanımlandıktan sonra tekrar kullanılabilir. Normal SQL cümlecikleri, genellikle bir kez çalıştırıldıktan sonra sonuçlar elde edilir ve bu cümlecikler tekrar kullanılmaz.
- **Tanımlama ve Saklama:** View'ler, bir adlandırılmış yapı olarak saklanırken, normal SQL cümlecikleri anlık işlemler için kullanılır ve genellikle saklanmaz.
- **Görünüm ve Yapılandırma:** View'ler, belirli bir veri görünümünü veya yapılandırmayı temsil ederken, normal SQL cümlecikleri belirli bir işlemi veya sorguyu ifade eder.

View'ler ve normal SQL cümlecikleri, farklı amaçlar için kullanılır. View'ler veri görünümlerini oluşturmak ve kullanmak için kullanışlıdır, normal SQL cümlecikleri ise anlık işlemler için sorgular oluşturmak ve veri manipülasyonu yapmak için kullanılır.

#### Hash Tablosu Nedir?

Hash tabloları, hızlı erişim için optimize edilmiş veri yapılarıdır çünkü veriye erişimde sabit bir zaman karmaşıklığına ( $O(1)$ ) sahip olabilirler. Bu hızlı erişim, bir anahtarın doğrudan bir dizi indekse dönüştürülerek ilgili veriye erişimi sağlar. İşte bunun bazı nedenleri:

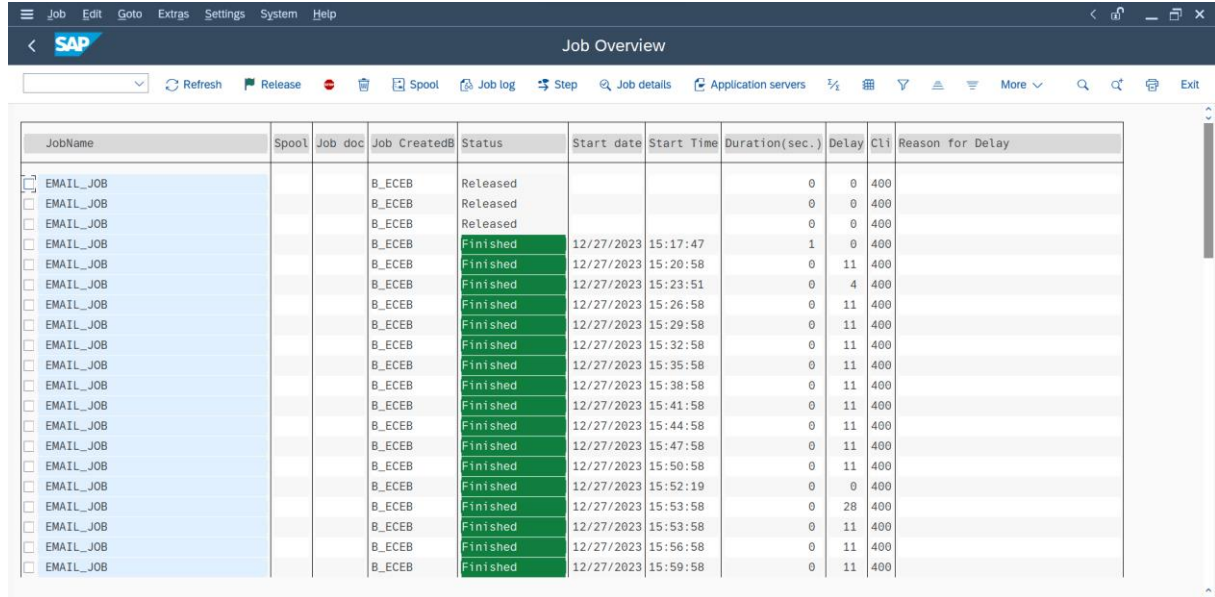
- **Hash Fonksiyonları:** Anahtarlar, bir hash fonksiyonu aracılığıyla bir dizi indekse dönüştürülür. İyi bir hash fonksiyonu, genellikle çakışmaları (collision) en aza indirger, yani farklı anahtarlar aynı indekse eşlenmez. Bu sayede her anahtar için benzersiz bir indeks belirlenir ve veriye erişim hızlı olur.
- **Sabit Zaman Karmaşıklığı ( $O(1)$ ):** İyi bir hash fonksiyonu kullanıldığında, veri boyutu artsa bile erişim süresi değişmez. Çünkü her anahtar, hash fonksiyonu tarafından doğrudan indekse dönüştürülerek ilgili veriye erişilir.
- **Hafızadaki Doğrudan Erişim:** Hash tablolar, bellekte doğrudan indeks üzerinden erişim sağlar. Bu, diğer veri yapılarına göre (örneğin, linked list gibi) bellek üzerinde dolaşma gereksinimini ortadan kaldırır, bu da daha hızlı erişim anlamına gelir.

Ancak, hash tablolarının hızlı erişim avantajı, iyi bir hash fonksiyonunun kullanılması ve çakışmaların minimize edilmesi ile sağlanır. Eğer çok sayıda çakışma olursa (yani, birçok anahtar aynı indekse düşerse), bu durum hızlı erişimi azaltabilir ve performansı düşürebilir. Bu yüzden, iyi bir hash fonksiyonu seçmek ve çakışmaları yönetmek önemlidir.

#### SAP JOB

1. SE38 transaksyon koduyla program ekranına giriş yaptım.
2. Standard program RSWUWFML2'yi çalıştırdım.
3. İş öğeleri için bildirimleri gönderme ekranına geldim.
4. Zaten önceden kaydedilmiş olanı "Save as variant" olarak seçtim.
5. Variant Attributes ekranına ulaştım.
6. Yeni variant ismini "Etest" olarak belirledim.
7. SM36 transaksyon koduyla Define Job ekranına geçiş yaptım.
8. İş adını "email\_job" olarak belirledim.
9. Kullanıcıyı "sap\_wfirt" olarak ayarladım ve kaydettim.

10. Start time ekranında "Period values" seçeneğini tıkladım.
11. Options bölümünden "Others"'ı seçip "Days"'i işaretledim ve günlük olarak çalışması için "1" yazdım.
12. Define job ekranına geri dönüp "Job selection"'ı tıkladım.
13. Scheduled'ı seçtim. (Diğer seçenekler zaten seçiliydi.)
14. Job Overview'da detayları kontrol ettim.



The screenshot shows the SAP Job Overview screen. The table displays the following data:

JobName	Spool	Job doc	Job CreatedB	Status	Start date	Start Time	Duration(sec.)	Delay	CLI	Reason for Delay
EMAIL_JOB			B_ECEB	Released			0	0	400	
EMAIL_JOB			B_ECEB	Released			0	0	400	
EMAIL_JOB			B_ECEB	Released			0	0	400	
EMAIL_JOB			B_ECEB	Finished	12/27/2023	15:17:47	1	0	400	
EMAIL_JOB			B_ECEB	Finished	12/27/2023	15:20:58	0	11	400	
EMAIL_JOB			B_ECEB	Finished	12/27/2023	15:23:51	0	4	400	
EMAIL_JOB			B_ECEB	Finished	12/27/2023	15:26:58	0	11	400	
EMAIL_JOB			B_ECEB	Finished	12/27/2023	15:29:58	0	11	400	
EMAIL_JOB			B_ECEB	Finished	12/27/2023	15:32:58	0	11	400	
EMAIL_JOB			B_ECEB	Finished	12/27/2023	15:35:58	0	11	400	
EMAIL_JOB			B_ECEB	Finished	12/27/2023	15:38:58	0	11	400	
EMAIL_JOB			B_ECEB	Finished	12/27/2023	15:41:58	0	11	400	
EMAIL_JOB			B_ECEB	Finished	12/27/2023	15:44:58	0	11	400	
EMAIL_JOB			B_ECEB	Finished	12/27/2023	15:47:58	0	11	400	
EMAIL_JOB			B_ECEB	Finished	12/27/2023	15:50:58	0	11	400	
EMAIL_JOB			B_ECEB	Finished	12/27/2023	15:52:19	0	0	400	
EMAIL_JOB			B_ECEB	Finished	12/27/2023	15:53:58	0	28	400	
EMAIL_JOB			B_ECEB	Finished	12/27/2023	15:53:58	0	11	400	
EMAIL_JOB			B_ECEB	Finished	12/27/2023	15:56:58	0	11	400	
EMAIL_JOB			B_ECEB	Finished	12/27/2023	15:59:58	0	11	400	

## FIELD Alan Ekleme

SAP standart tablolara yeni bir Z ile başlayan alan eklemek genellikle önerilmez çünkü SAP, standart tabloların değiştirilmesini önermez ve güncelleme sırasında sorunlara neden olabilir. Ancak, bazı durumlarda SAP'nin sağladığı belli uzantı mekanizmalarını kullanarak bu tür değişiklikleri yapabilirsiniz.

Bir tabloya Z ile başlayan yeni bir alan eklemek istiyorsanız, genellikle şu adımları takip edersiniz:

1. **Append yapısı kullanma:** SAP, birçok standart tablo için "append" yapıları sağlar. Bu yapıları kullanarak standart tabloya kendi alanlarınızı ekleyebilirsiniz. Öncelikle, SAP'nin bu tablo için sunduğu append yapılarını kontrol etmelisiniz.
2. **Customizasyon veya Enhancement Framework kullanma:** SAP, Enhancement Framework gibi mekanizmalarla standart tabloların özelleştirilmesine izin verir. Bu çerçeveleri kullanarak, standart bir tabloya ek alanlar ekleyebilirsiniz.
3. **BADI veya User Exit kullanma:** Bazı durumlarda, standart bir tabloya alan eklemek için Business Add-Ins (BADI) veya User Exits gibi özelleştirme noktalarını kullanabilirsiniz.

Ece Nur BALKAN