








## ABAP HASH TABLE NEDİR?

ABAP Hash Table, ABAP programlarında kullanılan bir tür iç tablodur. Bu tablolar, hash işlevini kullanarak gerekli tablo kaydını elde etmek için kullanılır.

- Hash tablo, benzersiz bir karma anahtarı kullanarak içeride saklanan satırlara sahip bir iç tablo kategorisidir.

### Kullanım Alanları:

- **Veri Okuma:** Tek bir belge okunacaksa, örneğin bir malzeme bilgisi, **hashed tablo** kullanmak mantıklı olabilir.
- Birden fazla belge okunacaksa, örneğin siparişler, **hashed tablo** kullanmak önerilmez. Çünkü **hashed tablolar**, çok detaylı bilgileri okuduğundan performans olarak hızlı olmayabilir.
- **Veri İşleme:** Malzeme oluşturma, güncelleme gibi işlemler için de **hashed tablo** kullanılabilir.
- Performans Optimizasyonu: ABAP Hashed Table, yüksek sayıda satırın iç tablolarda depolandığı durumlarda performans kayıplarını önlemek için kullanılabilir.

	Index Tables		Hashed Table
Table Kind	STANDARD TABLE	SORTED TABLE	HASHED TABLE
Index Access 			
Key Access 			
Key Uniqueness	NON-UNIQUE	UNIQUE/NON-UNIQUE	UNIQUE
Use in	Mainly Index Access	Mainly Key Access	Only Key Access

Şekil 1. İnternal Tabloların Türleri

**Default Key:** ABAP'ta “varsayılan anahtar” terimi genellikle bir tablonun birincil anahtarını ifade eder. Varsayılan anahtar, SELECT, READ, MODIFY, DELETE gibi işlemleri veya bir tablodaki verilere erişmeyi veya bunları değiştirmeyi içeren diğer işlemleri gerçekleştirmek için varsayılan anahtar kullanılır.

**Non-Unique:** customer\_id, order\_id, ve product\_id alanlarından oluşan bir anahtar, yani bir kombinasyon, tablonun bir anahtarıysa ve bu alanların birleşimi benzersiz olmayacak şekilde birden fazla kayıda sahipse, bu durumda bu anahtar non-unique olarak kabul edilir. Bu durumda aynı müşteri, aynı sipariş ve aynı ürün için farklı miktarlarda kayıtlar bulunabilir.

**Unique Key:** “Unique key” (benzersiz anahtar), bir veritabanı tablosundaki belirli bir alan veya alan grubunun, her bir kayıt için eşsiz ve tekrarlanamayan bir değer içermesi gereken bir kısıtlamadır. Yani, bir alanın veya alan grubunun unique key olarak belirlenmesi, tablodaki her kaydın bu alan veya alan grubu tarafından benzersiz bir şekilde tanımlanması gerektiği anlamına gelir.

Bu benzersizlik kısıtlaması, genellikle veritabanı tasarımında kullanılır ve çoğu zaman birincil anahtar (primary key) olarak belirlenir. Birincil anahtar, tablo içindeki her kaydı benzersiz bir şekilde tanımlamak için kullanılır. Bunun yanı sıra, bir tabloda

birincil anahtarın dışında benzersiz olması istenen diğer alanlar için de unique key kısıtlamaları tanımlanabilir.

Hash table, verileri hızlı bir şekilde aramak ve erişmek için kullanılan bir veri yapısıdır. Hash tabloları, anahtar (key) ve değer çiftlerini içerir ve bu çiftler arasında bir eşleme sağlar. Hash tablosu, hızlı arama ve erişim sağlamak amacıyla özel bir hash fonksiyonu kullanır.

**İşleyiş mantığı şu adımlardan oluşur:**

1. Hash Fonksiyonu Kullanımı: Her anahtar (key), bir hash fonksiyonu tarafından bir hash değerine dönüştürülür. Bu hash değeri genellikle sabit uzunluktadır.
2. İndeksleme: Oluşturulan hash değeri, tablonun boyutuna uygun bir indekse dönüştürülür. Bu, tablonun içindeki bir hücreyi belirler.
3. Çakışma Kontrolü: İki farklı anahtarın aynı hash değerine sahip olması durumunda (çakışma), bu durumu çözmek için çeşitli yöntemler kullanılır. Bu yöntemler arasında zincirleme (chaining), açık adresleme (open addressing), ikinci bir hash fonksiyonu kullanma gibi teknikler bulunabilir.
4. Erişim ve Arama: Veriye erişim veya arama yaparken, anahtarın hash değeri alınarak bu değere karşılık gelen indekse gidilir ve değer alınır. Eğer çakışma durumu söz konusuysa, çakışma çözümleme stratejileri kullanılarak doğru değer bulunur.

**Hash tablolarının avantajları şunlardır:**

- Hızlı Erişim: Hash tabloları, sabit bir süre içinde veriye erişim sağlar, çünkü hash fonksiyonları genellikle sabit bir sürede çalışır.
- Benzersiz Anahtarlar: Her anahtarın bir hash değeriyle eşleştirilmesi, her anahtarın benzersiz olmasını sağlar.

Ancak, çakışmalar ve çakışma çözümleme stratejileri gibi konulara dikkat edilmelidir. Çakışmaların sayısı ve çözümleme stratejisi, hash tablosunun performansını etkileyebilir.

**Hash tablolarının bazı dezavantajları şunlardır:**

1. Çakışma Sorunları: Çakışmalar, iki veya daha fazla anahtarın aynı hash değerine sahip olması durumudur. Bu durum, çakışma çözümleme stratejileriyle ele alınabilir, ancak yanlış bir çözümleme stratejisi seçilirse veya çakışma sıklığı çok yüksekse performans sorunlarına neden olabilir.
2. Bellek Tüketimi: Hash tabloları, genellikle tüm olası indekslere sahip bir dizi kullanır. Bu durum, gereksiz bellek tüketimine neden olabilir, özellikle tablonun boyutu çok büyükse ve verilerin çoğu kullanılmıyorsa.
3. İşlem Maliyeti: Hash fonksiyonlarının hesaplanması ve çakışma çözümleme stratejilerinin uygulanması, ek işlem maliyetine yol açabilir. Özellikle büyük veri setleriyle çalışırken bu işlem maliyeti artabilir.
4. Sıralama Sorunları: Hash tabloları genellikle sıralı bir yapı sunmazlar. Verilerin belirli bir sıra içinde saklanması veya erişilmesi gerekiyorsa, hash tabloları uygun bir seçenek olmayabilir.
5. Bilgi Kaybı: Hash tabloları genellikle bir anahtarın hash değeriyle ilişkilendirilmiş bir değeri saklar. Ancak, bu yapı bazen orijinal anahtarın bilgisini kaybetmek anlamına gelebilir. Bu durum, bazı uygulamalarda orijinal anahtar bilgisinin korunmasının gerektiği durumlarda sorunlara yol açabilir.

```

8  TABLES : mara.
9
10 SELECT-OPTIONS : so_matnr FOR mara-matnr.
11
12 □ TYPES : BEGIN OF st_mara,
13     matnr TYPE mara-matnr,
14     mtart TYPE mara-mtart,
15     matkl TYPE mara-matkl,
16 END OF st_mara.
17
18 data ls_mara type st_mara.
19
20 DATA lt_standard_tab TYPE TABLE OF st_mara WITH DEFAULT KEY. "kendi idsi      ***
21                                     "----->
22 DATA lt_standard_tab1 TYPE standard TABLE OF st_mara WITH DEFAULT KEY. "kendi idsi**
23
24 DATA lt_sorted_tab TYPE SORTED TABLE OF st_mara WITH UNIQUE KEY matnr. "matnrden unique key
25
26 DATA lt_sorted_tab1 TYPE SORTED TABLE OF st_mara WITH NON-UNIQUE KEY matnr. "matnrden unique OLMAYAN key
27
28 DATA lt_hash_tab TYPE HASHED TABLE OF st_mara WITH UNIQUE KEY matnr. "matnrden unique key
29
30 " DATA lt_hash_tab TYPE HASHED TABLE OF st_mara WITH NON-UNIQUE KEY matnr. "SYNTAX ERROR.
31
32 START-OF-SELECTION.
33
34     select matnr mtart matkl from mara into table lt_standard_tab
35     where matnr in so_matnr.
36
37     select matnr mtart matkl from mara into table lt_sorted_tab
38     where matnr in so_matnr.
39
40     select matnr mtart matkl from mara into table lt_hash_tab
41     where matnr in so_matnr.
42
43 *     READ TABLE lt_hash_tab into data(ls_mara) index 12. "index access yoktur
44     READ TABLE lt_hash_tab into ls_mara with TABLE KEY matnr ='12'.
45     READ TABLE lt_hash_tab into ls_mara with TABLE KEY matnr ='EWMS4-11'.
46
47 BREAK-POINT.
48     cl_demo_output=>display( ls_mara ).

```

Şekil 2. Örnek Kod Bloğu

- Satır 44'de matnr değerini 12 verince data fetchlenmiyor çünkü matnr alanında 12 değeri olmadığı için bunu key value olarak hash table üzerinde kullanılmamış oluyor.
- Öte yandan EWMS4-11 değeri ise matnr alanında bir kayıt olduğu için bunu hash table üzerinde kullanabiliriz ve ilgili satırı çekebiliriz.

Table: MARA  
Displayed Fields: 16 of 312 Fixed Columns: 2 List With

MANDT	MATNR	ERSDA	CREATED_AT_TIME	ER
400	CM-FL-V00	10/19/2022	00:37:53	BP
400	CM-FL-V01	10/19/2022	00:35:19	BP
400	CM-MLFL-KM-VXX	10/19/2022	00:35:40	BP
400	EWMS4-01	10/19/2022	00:46:08	BP
400	EWMS4-02	10/19/2022	00:46:08	BP
400	EWMS4-03	10/19/2022	01:01:31	BP
400	EWMS4-10	10/19/2022	00:46:08	BP
400	EWMS4-11 key 8	10/19/2022	00:46:08	BP
400	EWMS4-20	10/19/2022	00:52:02	BP
400	EWMS4-40	10/19/2022	00:46:08	BP
400	EWMS4-41	10/19/2022	00:46:08	BP
400	EWMS4-42	10/19/2022	00:46:08	BP
400	EWMS4-50	10/19/2022	01:02:34	BP
400	EWMS4-502	10/19/2022	01:02:34	BP
400	EWMS4-503	10/19/2022	01:02:34	BP
400	EWMS4-601	10/19/2022	01:02:34	BP
400	EWMS4-CAR00	10/19/2022	00:46:08	BP
400	EWMS4-PAL00	10/19/2022	00:46:08	BP

Şekil 3. İNDEX Değeri 8

Tables Table Contents

Table LT\_HASH\_TAB

Attributes Hashed [338x3(106)]

Insert Column Columns ...

Row	MATNR [C(40)]	MTART [C(4)]	MATKL [C(9)]
124	FG-FL-MV-V01	FERT	L004
125	FG-FL-MV-V02	FERT	L004
126	EWMS4-01	HAWA	L001
127	EWMS4-02	HAWA	L001
128	EWMS4-10	HAWA	L001
129	EWMS4-11 index 129	HAWA	L001
130	EWMS4-40	HAWA	L001
131	EWMS4-41	HAWA	L001
132	EWMS4-42	HAWA	L001
133	EWMS4-CAR00	VERP	
134	EWMS4-PAL00	VERP	

Şekil 4. İNDEX Değeri 129

- Şekil 4.de görüldüğü gibi kullandığımız key'in ana tablodaki index değeriyle internal tablodaki index değeri arasında bir ilişki bulunamamaktadır.
- **Hash Key:** Tablo satırlarına atamanın bir karma algoritma kullanılarak yönetildiği dahili bir tablonun benzersiz tablo anahtarı. Bir hash anahtarı, hashlenmiş bir tablonun birincil tablo anahtarıdır ve her tür internal tabloya ikincil tablo anahtarı olarak atanabilir. Bir hash anahtarı kullanarak dahili bir tabloya erişim sırasında, tablo girişlerinin sayısına bakılmaksızın yanıt süresi sabittir.