

# Dağıtık Sistemlerde Veri Tutarlılığı: CAP Teoremi, ACID ve BASE

Modern veri tabanı sistemlerinde performans, tutarlılık ve kullanılabilirlik arasında bir denge kurmak gerekmektedir. Bu bağlamda **CAP Teoremi**, **ACID** ve **BASE** ilkeleri, özellikle büyük ölçekli sistemler ve dağıtık veritabanları için kritik öneme sahiptir.

## 1. CAP Teoremi (Brewer'in Teoremi)

CAP Teoremi, 2000 yılında Eric Brewer tarafından ortaya atılmıştır. Dağıtık bir sistemde **aynı anda üç temel özellikten yalnızca ikisinin** sağlanabileceğini savunur:

- C – Consistency (Tutarlılık):** Tüm düğümler (nodes) her zaman aynı veriye sahip olur.
- A – Availability (Erişilebilirlik):** Tüm istekler mutlaka cevap alır (başarılı ya da hata ile).
- P – Partition Tolerance (Bölünme Toleransı):** Ağ bölünmelerine rağmen sistem çalışmaya devam eder.

### *Teorem Gereği:*

Bir dağıtık sistem **aynı anda sadece iki özelliği tam anlamıyla sağlayabilir**, üçüncüden ödün verilmelidir.

Sistem Türü	Sağlanan Özellikler	Örnek
CA Sistemi	Consistency + Availability	Tek sunuculu ilişkisel veritabanları (ör: PostgreSQL)
CP Sistemi	Consistency + Partition Tolerance	HBase
AP Sistemi	Availability + Partition Tolerance	CouchDB, Cassandra

Gerçek dünya uygulamalarında çoğu sistem, ihtiyaca göre tutarlılıktan ya da erişilebilirlikten kısmen feragat eder.

## 2. ACID Özellikleri

ACID, geleneksel ilişkisel veri tabanlarında işlemlerin güvenilirliğini sağlamak amacıyla geliştirilmiş 4 temel özelliği ifade eder:

### 1. A – Atomicity (Atomiklik)

İşlem ya tamamen gerçekleşir ya da hiç gerçekleşmez. Yarım kalmış işlemler sistemde iz bırakmaz.

### 2. C – Consistency (Tutarlılık)

İşlem öncesi ve sonrası veri tabanı tanımlı kurallara uygun durumdadır. Veri bütünlüğü korunur.

### 3. I – Isolation (Yalıtım)

Aynı anda gerçekleşen işlemler birbirini etkilemez, sanki sırasıyla yapılmış gibi davranılır.

### 4. D – Durability (Kalıcılık)

Tamamlanan işlemlerin sonucu kalıcıdır, sistem çökse bile veri kaybı yaşanmaz.

ACID, banka sistemleri gibi yüksek güvenlik gerektiren ortamlarda tercih edilir.

#### Örnek:

```
BEGIN;  
UPDATE HESAPLAR SET bakiye = bakiye - 100 WHERE id = 1;  
UPDATE HESAPLAR SET bakiye = bakiye + 100 WHERE id = 2;  
COMMIT;
```

Transfer işlemi ya tamamen gerçekleşir ya da başarısız olursa her şey geri alınır (rollback yapılır).

## 3. BASE Yaklaşımı (Eventually Consistent)

BASE, NoSQL sistemlerin ACID'e esnek bir alternatifidir. Özellikle büyük veri ve dağıtık sistemlerde performans odaklı çözümler için kullanılır. BASE, şu kavramları içerir:

### 1. B – Basically Available (Temel Eriřilebilirlik)

Sistem çoęu zaman kullanılabilir durumdadır, bazı durumlarda gecikme olabilir.

### 2. A – Soft state (Yumuřak Durum)

Veritabanının durumu zamanla deęiřebilir. Sabit ve kesin veri garantisi yoktur.

### 3. SE – Eventually Consistent (Sonunda Tutarlılık)

Veriler anlık olarak tutarsız olabilir fakat zamanla sistem kendini senkronize eder.

BASE sistemlerde hız ve erişilebilirlik ön plandadır, verinin %100 anlık doğruluęu ikinci planda kalabilir.

#### Örnek:

- Sosyal medya yorumlarının hemen gösterilmesi ama birkaç saniye sonra güncellenmesi.
- Amazon'un ürün stok bilgilerinin gecikmeli güncellenmesi.

## ACID vs BASE Karşılaştırması

Özellik	ACID	BASE
Tutarlılık	Kesin tutarlılık (strict)	Zamanla tutarlılık (eventual)
Performans	Düşük	Yüksek
Kullanım Alanı	Bankacılık, finans	Büyük veri, sosyal ağlar
Dayanıklılık	Yüksek	Esnek

## Sonuç

Veri tabanı sistemlerinde seçim yapılırken **performans, tutarlılık ve erişilebilirlik** gibi kriterler dikkate alınmalıdır. Geleneksel sistemlerde ACID, modern büyük ölçekli sistemlerde ise BASE ilkeleri ve CAP Teoremi, tasarımın temel yapı taşlarını oluşturur.

# VERİ YÖNETİMİNDE OLAP, OLTP VE BÜYÜK VERİ (BIG DATA)

Veri yönetimi, yalnızca verinin saklanması ve işlenmesiyle sınırlı değildir. Aynı zamanda verinin nasıl işlendiği, analiz edildiği ve karar destek sistemlerine nasıl katkı sunduğu da önemlidir. Bu bağlamda, **OLAP (Online Analytical Processing)** ve **OLTP (Online Transaction Processing)** gibi iki temel işlem türü, veri tabanı sistemlerinde farklı amaçlara hizmet ederken; **Big Data (Büyük Veri)** kavramı ise bu sistemlerin üzerinde çalıştığı geniş veri evrenini tanımlar.

## 1. OLTP (Online Transaction Processing) – Çevrimiçi İşlem İşleme

**OLTP**, gerçek zamanlı veri işleme sistemidir ve özellikle **günlük operasyonel işlemler** için kullanılır. Bankacılık sistemleri, e-ticaret siteleri, müşteri işlemleri gibi sürekli veri giriş/çıkışının olduğu sistemlerde OLTP kullanılır.

### Özellikleri:

- Yüksek işlem hacmi (transaction throughput)
- Anlık veri güncelleme (insert, update, delete işlemleri)
- Düşük gecikme süresi
- Normalize veri yapısı (veri tekrarını en aza indirme)
- Kullanıcı odaklıdır (müşteri, personel işlemleri)

### Örnek:

Bir e-ticaret sitesinde müşterinin sepete ürün eklemesi, sipariş vermesi ve ödeme yapması bir OLTP sürecidir.

```
INSERT INTO Orders (order_id, customer_id, total_amount) VALUES  
(1001, 25, 349.90);
```

## 2. OLAP (Online Analytical Processing) – Çevrimiçi Analitik İşleme

**OLAP**, veri üzerinde derinlemesine analiz yapılmasına olanak tanır. İş zekâsı, raporlama ve karar destek sistemlerinde kullanılır. Veri genellikle **veri ambarlarından (data warehouse)** alınır.

### Özellikleri:

- Düşük işlem hacmi, yüksek veri hacmi
- Kompleks sorgular (çok boyutlu analizler)
- Genellikle read-only işlemler
- Denormalize veri yapısı (performans için tekrar veriler olabilir)
- Yönetici ve analist odaklıdır

### Örnek:

Bir mağazanın son 5 yıldaki satışlarını bölge, ürün türü ve sezona göre analiz etmesi OLAP işlemine örnektir.

```
SELECT region, product_type, SUM(sales)
FROM Sales_Data
GROUP BY region, product_type;
```

## 3. Big Data (Büyük Veri)

**Big Data**, geleneksel veri tabanı yönetim sistemleriyle işlenemeyecek kadar büyük, hızlı ve çeşitli veri kümelerini ifade eder. Sosyal medya, IoT cihazları, mobil uygulamalar gibi kaynaklardan sürekli akan veriler büyük veri kapsamında değerlendirilir.

### Büyük Verinin 5V Özelliği:

1. **Volume (Hacim):** Terabaytlarca veri
2. **Velocity (Hız):** Anlık veri akışı ve güncelleme
3. **Variety (Çeşitlilik):** Yapılandırılmış (tablo), yapılandırılmamış (video, ses), yarı yapılandırılmış (JSON)
4. **Veracity (Doğruluk):** Verinin güvenilirliği ve doğruluğu
5. **Value (Değer):** Veriden çıkarılabilecek anlamlı bilgi

## Büyük Veri Teknolojileri:

- Hadoop
- Spark
- NoSQL (MongoDB, Cassandra)
- Kafka

## Örnek:

Twitter üzerinde saniyede binlerce gönderinin analiz edilmesi, spam içeriklerin tespiti ve kullanıcı eğilimlerinin çıkarılması büyük veri işleme örneğidir.

## OLAP – OLTP Karşılaştırması:

Özellik	OLTP	OLAP
Amaç	İşlem işleme	Veri analizi, karar desteği
Kullanıcı	Son kullanıcı (müşteri)	Yöneticiler, analistler
Veri yapısı	Normalize	Denormalize
Sorgular	Basit, kısa	Karmaşık, uzun
İşlem sayısı	Yüksek	Düşük
Veri hacmi	Az (günlük veriler)	Çok (tarihsel veriler)

## SONUÇ

OLTP ve OLAP sistemleri, farklı amaçlara hizmet eden veri işleme modelleridir. OLTP; hızlı, güvenilir ve anlık işlemler için tasarlanırken, OLAP büyük veri kümeleri üzerinden anlamlı analizler yapılmasını sağlar. Öte yandan Big Data kavramı, bu sistemlerin üzerinde çalıştığı çok büyük ve çeşitli veri kaynaklarını temsil eder. Kurumlar, doğru teknolojiyi doğru bağlamda kullanarak veri üzerinden stratejik kararlar alabilir hale gelmektedir.

**Bu bölüm, BTK Akademi kaynakları ve modern veri sistemleri dokümantasyonları referans alınarak Arda Karadağ tarafından hazırlanmıştır.**

