

**SQL (Structured Query Language)** ve **NoSQL (Not Only SQL)** veritabanları, farklı yaklaşımlara sahip olan iki temel veritabanı türüdür.

#### **Veri Modeli:**

**SQL:** İlişkisel modeli kullanır. Veri tabloları ve bu tablolar arasındaki ilişkiler temel alınır.

**NoSQL:** Çeşitli veri modelleri kullanabilir. Örneğin, belge tabanlı (Document), sütun tabanlı (Column), anahtar-değer tabanlı (Key-Value), graf tabanlı (Graph) gibi.

#### **Veri Yapısı:**

**SQL:** Tablolarda sütunlar ve satırlar kullanılarak yapılandırılan yapılara sahiptir.

**NoSQL:** Veri modeline bağlı olarak, belgeler, sütun aileleri, anahtar-değer çiftleri gibi farklı veri yapıları kullanılabilir.

#### **Veri İlişkileri:**

**SQL:** İlişkisel veritabanlarında sıkı ilişkiler tanımlanır. Bu nedenle verilerin düzgün bir şekilde ilişkilendirilmesi gereklidir.

**NoSQL:** Genellikle esnek veri yapılarına sahiptir ve ilişkisiz verileri daha iyi yönetebilir. Ancak bazı NoSQL veritabanları da ilişkisel verileri destekleyebilir.

#### **Ölçeklenebilirlik:**

**SQL:** Dikey ölçeklenebilirlik (scaling up) genellikle daha yaygındır, yani daha güçlü bir donanım ile mevcut veritabanının performansı artırılır.

**NoSQL:** Yatay ölçeklenebilirlik (scaling out) genellikle daha yaygındır, yani daha fazla sunucu ekleyerek veritabanının kapasitesi ve performansı artırılır.

#### **Sorgulama Dili:**

**SQL:** Yapılandırılmış sorgulama dili olan SQL kullanılır. Veritabanından veri çekmek, filtrelemek, birleştirmek için kullanılır.

**NoSQL:** Sorgulama dilleri veritabanı türüne göre değişebilir. Örneğin, belge tabanlı NoSQL veritabanları genellikle JSON benzeri formatlarda sorgulama yapar.

#### **Esneklik:**

**SQL:** Yapısal bir düzen gerektirir. Şema (schema) değişiklikleri zor olabilir.

**NoSQL:** Şema esnekliği daha yüksektir. Veri yapısında kolaylıkla değişiklik yapılabilir.

#### **Kullanım Senaryoları:**

**SQL:** Yapısal verilerin sıkı ilişkiler gerektirdiği durumlar için uygun olabilir. Örneğin finansal sistemler, ticaret siteleri vb.

**NoSQL:** Büyük, dinamik veya ilişkisiz veri kümesi olan projeler için daha uygundur. Örneğin büyük ölçekli sosyal ağlar, loglama sistemleri vb.

NoSQL veritabanlarının doğuşu, geleneksel ilişkisel veritabanlarının belirli senaryolarda yetersiz kalması veya uygun olmaması nedeniyle gerçekleşti.

#### **Esnek Veri Yapıları:**

Geleneksel SQL veritabanlarında veri şeması önceden tanımlanır ve bu şema doğrultusunda veri saklanır. Ancak bazen verinin yapısı önceden bilinmeyebilir veya değişebilir. NoSQL veritabanları, esnek veri yapıları sayesinde bu tür değişken veri yapılarını daha iyi yönetebilir.

#### **Büyük Veri ve Yüksek Hacimli Veri:**

Geleneksel veritabanları büyük veri külelerini işlemekte zorlanabilir. NoSQL veritabanları, yatay ölçeklenebilirlik ve paralel işleme yetenekleri sayesinde büyük veri külelerini daha etkili bir şekilde işleyebilir.

#### **Dağıtık ve Paralel İşleme:**

NoSQL veritabanları, genellikle dağıtık ve paralel işleme yetenekleriyle tasarlanmıştır. Bu, yüksek performans ve ölçeklenebilirlik avantajı sağlar.

#### **Yüksek Hız ve Düşük Gecikme Süresi:**

NoSQL veritabanları, bazı kullanım senaryolarında daha hızlı yanıt süreleri sağlayabilir. Özellikle gerçek zamanlı uygulamalar, önbellekleme gerektiren projeler veya hızlı veri akışı gerektiren sistemler için tercih edilebilir.

#### **İlişkisiz Veri:**

Bazı projelerde veriler arasında sıkı bir ilişki bulunmaz. Geleneksel SQL veritabanlarının bu tür durumlar için getirdiği karmaşıklık ve performans kayıpları, NoSQL veritabanlarının kullanılmasını teşvik edebilir.

NoSQL veritabanları aynı zamanda belirli projelerde tercih edilir:

### **Büyük Ölçekli Web Uygulamaları:**

Sosyal medya platformları, e-ticaret siteleri gibi büyük ölçekli uygulamalar, NoSQL veritabanlarının ölçeklenebilirlik ve hız avantajlarından yararlanabilir.

### **Sağlık İzleme ve IoT Uygulamaları:**

Sensörlerden gelen sürekli veri akışını yöneten uygulamalar, NoSQL veritabanları ile daha etkili bir şekilde işlenebilir.

### **Oyun Geliştirme:**

Çok oyunculu oyunlar, oyun istatistikleri ve diğer oyun verilerini hızlı ve ölçeklenebilir bir şekilde saklamak için NoSQL veritabanlarını tercih edebilir.

### **Log ve Olay İzleme:**

Büyük ölçekli loglama sistemleri, NoSQL veritabanlarını kullanarak hızlı bir şekilde logları saklayabilir ve analiz edebilir.

### **Grafik Veri ve Sosyal Ağlar:**

Grafik verilerini (örneğin, arkadaşlık ilişkileri) saklamak ve sorgulamak için graf veritabanları kullanılabilir.