WEEK 3

**🔹 TRIGGER Nedir?**

**✅ Tanım:**

**Trigger**, SQL Server'da (ve diğer veritabanlarında) çalışan, **otomatik olarak tetiklenen** bir **veritabanı nesnesidir**.

* Bir tabloya **INSERT**, **UPDATE** veya **DELETE** işlemi uygulandığında çalışır.
* **“Bir şey olursa, otomatik olarak şu işlemi yap.”** mantığıyla çalışır.

**💡 Nerede Kullanılır?**

Senin örneğinle açıklayalım:  
**Bir satış yapıldığında**, yani Satislar tablosuna yeni bir kayıt eklendiğinde:

✅ **Stok miktarı otomatik düşsün**  
✅ **Log kaydı otomatik olarak tutulsun**  
✅ **İndirim hesaplaması yapılsın (örnek: kampanya)**  
✅ **Tahsilat tablosuna veri yazılsın**

Bunları sen yazmasan bile **trigger kendi yapar.**

**🧠 Trigger Türleri**

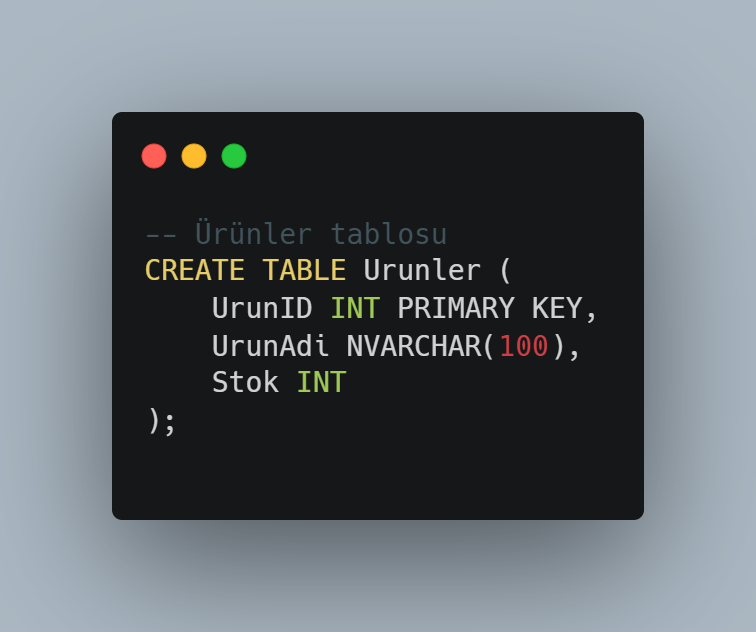
| **Tür** | **Açıklama** |
| --- | --- |
| AFTER INSERT | Veri eklendikten sonra tetiklenir |
| AFTER UPDATE | Veri güncellendikten sonra tetiklenir |
| AFTER DELETE | Veri silindikten sonra tetiklenir |
| INSTEAD OF | Normal işlemi iptal eder, kendi işlemini yapar |

**🛒 E-Ticaret Örneği: Satış Yapıldığında Stok Düşsün**

**🎯 Senaryo:**

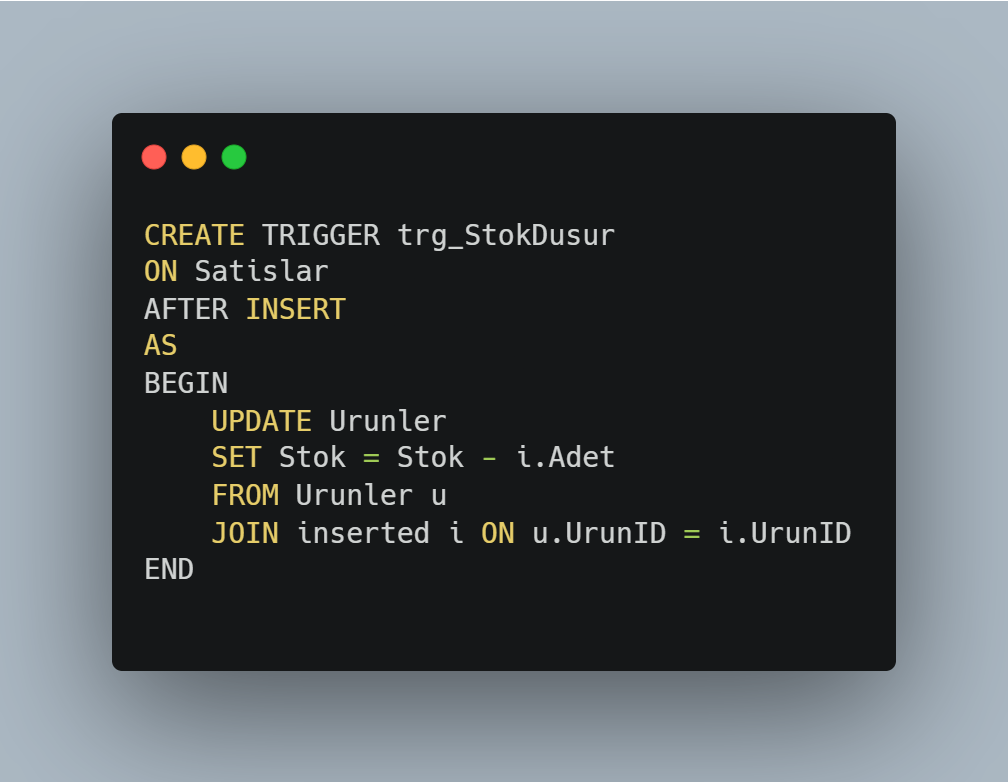
* Satislar tablosuna kayıt eklendiğinde
* Urunler tablosundaki ilgili ürünün **stok miktarı azalsın**

**📦 İlgili Tablolar (Basit Haliyle)**





**⚙️ Trigger Oluşturma**

****

**🧾 Ne Oldu?**

* inserted → yeni eklenen satış satırlarını temsil eder (SQL'de özel tablo).
* i.Adet → satılan ürün adeti kadar stok düşürülür.
* Bu trigger, **her yeni satışta otomatik olarak çalışır**.

**✅ Bonus: Neden Trigger Kullanılır?**

| **Avantaj** | **Açıklama** |
| --- | --- |
| Otomasyon | Belirli işlemler otomatik yapılır |
| Veri tutarlılığı | Hatalı stok, eksik kayıt önlenir |
| Kod karmaşası azalır | Tüm işlemler SQL tarafında |
| Tek noktadan kontrol | Tüm sistem trigger'a bağlı çalışır |

**🔹 Session Nedir?**

**✅ Tanım:**

**Session**, bir kullanıcının **web uygulamasına yaptığı geçici ziyareti (oturumu)** temsil eder.  
Kullanıcının **geçici verilerini** sunucuda saklamak için kullanılır.

📌 **Kısa haliyle:**

Session = “Kullanıcı giriş yaptıktan sonra verilerini geçici olarak saklamak.”

**🧠 Ne İçin Kullanılır?**

* Kullanıcının **oturumunu açık tutmak**
* **Giriş yapan kullanıcıyı hatırlamak**
* Sepet, tercih, rol bilgisi gibi geçici verileri saklamak

**🧱 Session Nerede Saklanır?**

Veri, çoğunlukla **sunucuda (server)** saklanır.  
Kullanıcının tarayıcısında sadece **Session ID** adlı küçük bir çerez (cookie) tutulur.

**🔐 Session Ne Zaman Biter?**

* Tarayıcı kapatılınca (varsayılan)
* Belirli bir süre kullanılmazsa (örneğin 20 dakika)
* Manuel olarak Session.Clear() ya da Session.Abandon() yapılırsa

**🔧 ASP.NET Core'da Session Kullanımı**

**1. Startup.cs veya Program.cs ayarları:**

builder.Services.AddSession();

app.UseSession();

**2. Veri ekleme:**

HttpContext.Session.SetString("kullaniciAdi", "baran");

**3. Veri okuma:**

string ad = HttpContext.Session.GetString("kullaniciAdi");

**🔄 Session mı? Cookie mi?**

| **Özellik** | **Session** | **Cookie** |
| --- | --- | --- |
| Nerede saklanır? | Sunucuda | Kullanıcının tarayıcısında |
| Veri boyutu | Büyük veri saklanabilir | Küçük veri (max 4 KB) |
| Güvenlik | Daha güvenlidir | Dış müdahaleye açık olabilir |
| Kullanım süresi | Tarayıcı kapanınca silinebilir | Süresi uzatılabilir (expire) |

**🛒 E-Ticaret Örneği:**

Kullanıcı sepete ürün ekledi → Bu ürünler session'da tutulur → Sipariş verince veritabanına yazılır.