

10. Ders - ETL

15 Eylül 2025 Pazartesi 09:16

- **Veri Söleşmesi -> Staging** : verilerin hedef sisteme yüklenmek üzere hazırlandığı, temizleme, normalleştirme ve dönüştürme işlemlerinden geçtiği geçici bir çalışma alanıdır.
- Araçların şase numarası, servis id'si vs bunlar Staging kapsamında işlenebilen veriler olabilir diye örnekledik

ISO 3779/3780:

Konu: Araç Kimlik Numarası (VIN - Vehicle Identification Number)

Amaç: Dünyadaki para birimlerini temsil eden üç harfli alfabetik (örn: USD, EUR, TRY) ve üç rakamlı sayısal kodları tanımlar. Üretici, model, motor tipi, üretim yeri gibi bilgileri standart bir formatta şifreler.

ISO 4217:

Konu: Para birimi kodları

Amaç: Dünyadaki para birimlerini temsil eden üç harfli alfabetik (örn: USD, EUR, TRY) ve üç rakamlı sayısal kodları tanımlar. Finansal işlemlerde ve uluslararası ticarete karışıklığı önler.

- **İncele** : <https://learn.microsoft.com/tr-tr/azure/synapse-analytics/sql/overview-architecture>

Veri Ambarı (ETL) Katmanları :

ETL'in içinde Staging, Core ve Mart olmak üzere akışlarımız bulunmaktadır:

1. Staging (Evreleme)

- Ham verinin kaynak sistemlerden (DB, API, log, vs.) olduğu gibi, hiç değiştirilmeden çekildiği ilk katman.
- Amaç: Kaynak sistemleri ETL sürecinin yükünden korumak. Hızlıca veriyi almak. Hiçbir veriyi kaybetmeden izlenebilir bir şekilde kaynağı depolaması. Farklı kaynaklardan gelen verinin heterojenliğini korur.

Özellikler:

- Genellikle geçici (temporary) tablolardır.
- Veri kalitesi kontrolleri yapılmaz.
- Yapı, kaynak sistemle aynıdır.
- Veri tipleri gevşek tutulur. (string / text)
- Audit sütunları eklenir. (load_id, src_system) -> [ETL'de audit sütunları, verinin yüklenme sürecini izlemek için eklenen kontrol alanlarıdır. Genelde kim yükledi, ne zaman yüklendi, hangi kaynaktan geldi, kaçınıcı sürüm gibi bilgileri tutar. Bu sayede veri hattı şeffaf, izlenebilir ve hatalar kolayca takip edilebilir.]

2. Core (Çekirdek / Integration)

- İş kurallarının uygulandığı, temizlenen, birleştirilen ve standartlaştırılan katman.
- Farklı kaynaklardan gelen veriyi normalize etmek standartize etmek.
- Veri belli iş kuralları ve data contract seviyesinde normalize edilir.

```
1 vehicle_vin = X12234555555555555555 ISO 3779/3780
2 sale_currency = (USD) (TL) ISO4217
```

- Amaç: Tek doğru kaynak (single source of truth) yaratmak.

Özellikler:

- Veri temizleme (data cleansing) burada olur.
- Farklı kaynaklardan gelen veriler entegre edilir.
- Tarihsel veri saklama (history) bu katmanda başlar.
- Genellikle "normalize" bir yapıdadır.
- Tip dönüşümleri, standartlaşma, SCD (Slowly Changing Dimension), as-of join(?)

3. Mart (Data Mart / Sunum)

- Kullanıcıya ve raporlama araçlarına hazır verinin sunulduğu son katman.
- Amacı iş zekası, raporlama ve AI Analitik katmanları için optimize edilmiş özet tablolar sunmak.
- Amaç: Belirli bir departmanın veya iş biriminin (Örn: Finans, Satış) ihtiyaçlarına yönelik performanslı sorgulama ve raporlama.

Özellikler:

- Genellikle yıldız veya kartalı şema gibi dimension modelleri kullanılır (Fact ve Dimension tabloları).
- Sorgu performansı için optimize edilmiştir.
- Kullanıcılar doğrudan bu katmandaki verileri kullanır.
- İş odaklı metrikler (ortalama satış fiyatı, il 12 ay servis maliyeti, en çok değişen parçalar)
- **Periyodik snapshot** : Haftalık, günlük özet tablolar

Basit Analoji:

- ◇ Staging: Çiğ malzemelerin mutfağa getirildiği yer.
- ◇ Core: Malzemelerin yıkanıp, doğranıp, pişirildiği tezgah.
- ◇ Mart: Servis tabağı. Yemeğin kullanıcıya sunulduğu nihai halidir.

SQL sadece veri değil, AI için bir API

(BURAYA NOT EKLE)

```
1 CREATE TABLE core_sales(  
2   sale_id,  
3   customer_id,  
4   dealer_code,  
5   sale_time_utc,  
6   vehicle_vin,  
7   price_usd  
8 );  
9  
10 COMMENT ON COLUMN core_sales.sale_time_utc IS  
11 COMMENT ON COLUMN core_sales.price_usd IS  
12 CHECK (price_usd > 0) -- bu yapı sql şemamızın API gibi okunmasını sağlıyor artık
```

Self-Healing ETL

Self-Healing ETL (Kendini iyileştiren ETL), olağan dışı bir durum, hata veya veri kaynağındaki beklenmedik bir değişiklikle karşılaştığında, *insan müdahalesi olmadan bu sorunu tespit eden*, tanımlayan ve çözmeye çalışan (veya en azından sistemi çalışır durumda tutan) otomatik ETL süreçlerini ifade eder.

Geleneksel ETL'den farkı, "fail and notify" (başarısız ol ve bildir) yerine "detect and recover" (tespit et ve kurtar) mantığıyla çalışmasıdır.

```
SELECT * FROM ops.runlog WHERE load_id = '2025-09-15_sales'  
  
DELETE FROM mart_daily  
WHERE sale_date = '2025-09-15';  
  
INSERT INTO mart_daily  
SELECT sale_date, COUNT(*) AVF(price_usd)  
FROM core_sales  
WHERE sale_date = '2025-09-15'  
AND price_usd < 500000  
GROUP BY sale_date;
```

Explainability-by Design

Explainability-by-Design (EbD), bir YZ(Yapay Zeka)/MÖ(Makine Öğrenmesi) modelinin geliştirme sürecinin en başından itibaren açıklanabilir olmasını hedefleyen bir tasarım felsefesidir.

Modeli önce karmaşık bir şekilde kurup, sonradan "neden bu sonucu verdi?" sorusunu cevaplamaya çalışmak yerine, açıklanabilirliği projenin bir gereksinimi ve ayrılmaz bir parçası haline getirir.

Türkçede "Tasarımdan İtibaren Açıklanabilirlik" veya "İçsel Açıklanabilirlik" olarak düşünülebilir.

```
SELECT sale_id, vehicle_vin, price_usd FROM core_sales
WHERE customer_region = 'Samandıra' AND payment = 'loan'
AND sale_date BETWEEN '2025-08-15' AND '2025-09-15'
ORDER BY price_usd DESC
LIMIT 10;
```