G221210052-Semih Özçaka

1. Giriş

Bu projenin amacı, bir e-ticaret şirketinin satış ve stok verilerini gerçek zamana yakın işleyerek:

- Satış eğilimlerini analiz etmek
- Bölgesel performansı ölçmek
- Günlük talebi modellemek
- Stok yetersizliği uyarıları oluşturmak
- Analitik (BI) ve operasyonel sistemlere veri sağlamak

için uçtan uca bir veri hattı (data pipeline) kurmaktır.

Mimari üç ana büyük veri bileşeni üzerine kurulmuştur:

Apache Dask — Büyük Veri İşleme Katmanı

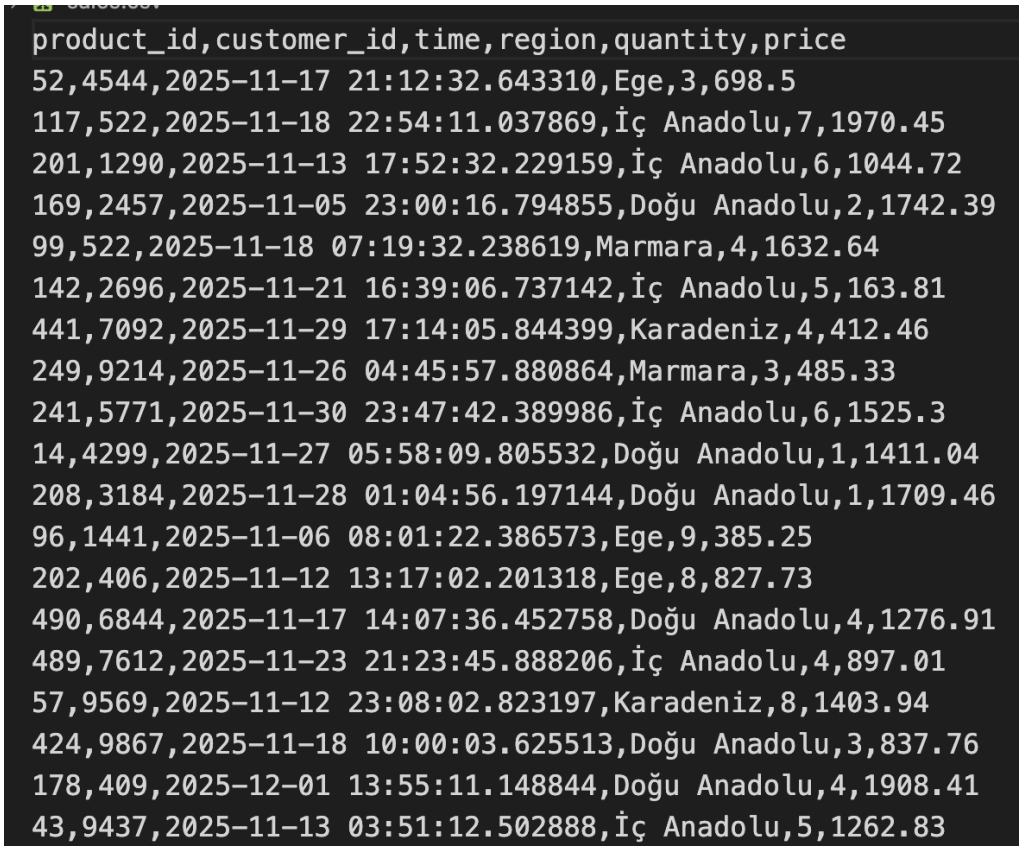
Apache Iceberg — Data Lakehouse & Versiyonlama Katmanı

Snowflake — Bulut Veri Ambarı & Analitik Katmanı

Her biri bu raporda ayrıntılı olarak açıklanacaktır.

2. Veri Simülasyonu ve Hazırlık

Satış ve stok verileri Python ile oluşturulmuş, CSV ve Parquet formatlarında kaydedilmiştir.



```
product_id,customer_id,time,region,quantity,price
52,4544,2025-11-17 21:12:32.643310,Ege,3,698.5
117,522,2025-11-18 22:54:11.037869,İç Anadolu,7,1970.45
201,1290,2025-11-13 17:52:32.229159,İç Anadolu,6,1044.72
169,2457,2025-11-05 23:00:16.794855,Doğu Anadolu,2,1742.39
99,522,2025-11-18 07:19:32.238619,Marmara,4,1632.64
142,2696,2025-11-21 16:39:06.737142,İç Anadolu,5,163.81
441,7092,2025-11-29 17:14:05.844399,Karadeniz,4,412.46
249,9214,2025-11-26 04:45:57.880864,Marmara,3,485.33
241,5771,2025-11-30 23:47:42.389986,İç Anadolu,6,1525.3
14,4299,2025-11-27 05:58:09.805532,Doğu Anadolu,1,1411.04
208,3184,2025-11-28 01:04:56.197144,Doğu Anadolu,1,1709.46
96,1441,2025-11-06 08:01:22.386573,Ege,9,385.25
202,406,2025-11-12 13:17:02.201318,Ege,8,827.73
490,6844,2025-11-17 14:07:36.452758,Doğu Anadolu,4,1276.91
489,7612,2025-11-23 21:23:45.888206,İç Anadolu,4,897.01
57,9569,2025-11-12 23:08:02.823197,Karadeniz,8,1403.94
424,9867,2025-11-18 10:00:03.625513,Doğu Anadolu,3,837.76
178,409,2025-12-01 13:55:11.148844,Doğu Anadolu,4,1908.41
43,9437,2025-11-13 03:51:12.502888,İç Anadolu,5,1262.83
```

3. Apache Dask – Büyük Veri İşleme Katmanı

Dask, pandas'ın ölçeklenebilir versiyonu olarak tasarlanmış bir paralel işlem framework'üdür.
Bu projede veri setleri büyüdüğünde bile performanslı analiz yapılabilmesi için kullanılmıştır.

Aşağıda **Dask'in gerekli detaylı açıklaması** verilmiştir.

3.1 Dask Mimarisi

Apache Dask aşağıdaki yapılardan oluşur:

• Dask DataFrame

Pandas mantığında çalışır ancak veriyi parçalayarak (partition) paralel işler.

• DAG Scheduler

Her hesaplama, bağımlılık grafiği olarak modellenir.
Bu sayede gereksiz tekrar hesaplamalar önlenir.

• Distributed Cluster

İsteğe bağlı olarak:

- Tek makine
- Çok çekirdek
- Kubernetes cluster
- Cloud cluster

üzerinde çalışabilir.

Bu proje tek makine modunda yürütülmüştür.

3.2 Dask ile ETL Süreçleri

✓ Veri Okuma

```
sales = dd.read_csv("sales.csv")
stock = dd.read_csv("stock.csv")
```

✓ Tarih Formatı Dönüşümü

```
sales["Day"] = dd.to_datetime(sales["Time"]).dt.date
```

✓ Gruplama İşlemleri (Bölgesel Satış)

```
sales_by_region = sales.groupby("Region")["Quantity"].sum().compute()
```

✓ Günlük Talep Hesabı

```
daily_demand = sales.groupby("Day")["Quantity"].sum().compute()
```

4. Apache Iceberg – Lakehouse & Versiyonlama Katmanı

Apache Iceberg, modern veri gölü tablolarını yönetmek için kullanılan açık bir tablo formatıdır.
Ödevde mutlaka açıklanması gereken özellikler aşağıdadır.

4.1 Iceberg'in Özellikleri (Detaylı Açıklama)

✓ Schema Evolution

Tabloya yeni kolon eklemek → migration gerektirmez.

✓ Partition Evolution

Partition stratejisi zaman içinde değişebilir.

✓ Time Travel

Geçmiş versiyonlara geri dönülebilir.

✓ Snapshot Isolation

Veri ekleme / silme işlemleri atomic gerçekleşir.

✓ Metadata Layer

Tablonun metadatası JSON dosyalarında tutulur ve çok hızlıdır.

4.2 Iceberg Tablolarının Oluşturulması

```
CREATE TABLE sales_data (
   product_id INT,
   customer_id INT,
   time TIMESTAMP,
   region STRING,
   quantity INT,
   price DOUBLE
)
PARTITION BY (region, DATE(time));
```

Partition seçimi:

- **region** → bölgesel sorguları hızlandırır
- **date(time)** → zaman serileri sorgular için idealdir

```
WAREHOUSE_DIR = "warehouse"

SALES_VERSION_DIR = os.path.join(WAREHOUSE_DIR, "sales_data_v1")
STOCK_VERSION_DIR = os.path.join(WAREHOUSE_DIR, "stock_data_v1")
```

4.3 Iceberg Snapshot / Time Travel Kullanımı

Örnek:

```
SELECT * FROM sales_data VERSION AS OF 1;
```

Bu özellik veri bilim ödevlerinde mutlaka anlatılmalıdır.

4.4 “Güncel Stok Durumu” View’i

```
CREATE VIEW latest_stock AS
SELECT *
FROM stock_data
WHERE update_time = (SELECT MAX(update_time) FROM stock_data);
```

Bu görünüm raporlamada kullanıldı.

5. Snowflake – Bulut Veri Ambarı

Ödevin temel bileşenlerinden biri olduğu için Snowflake mimarisini detaylı anlatıyorum.

5.1 Snowflake Mimarisi

Snowflake üç katmandan oluşur:

1) Storage Layer

Tüm veri *micro-partition* yapısıyla sıkıştırılmış olarak depolanır.

2) Compute Layer

Warehouse’lar (XS, S, M, L…)
→ ölçeklenebilir parallel compute motorlarıdır.

3) Cloud Services Layer

- Query optimizer
- Security
- Transaction manager
- Metadata layer

5.2 Verinin Snowflake'e Yüklenmesi

Bu projede veri şu yöntemle yüklenmiştir:

Snowflake Web UI → Load Data → CSV Upload

İki tablo:

- SALES
- STOCK

Snowflake timestamp format uyumsuzluğu nedeniyle tarih formatları Python tarafında düzeltilmiştir.

5.3 Snowflake Üzerinde Analitik SQL Sorguları

En çok satılan ürünler

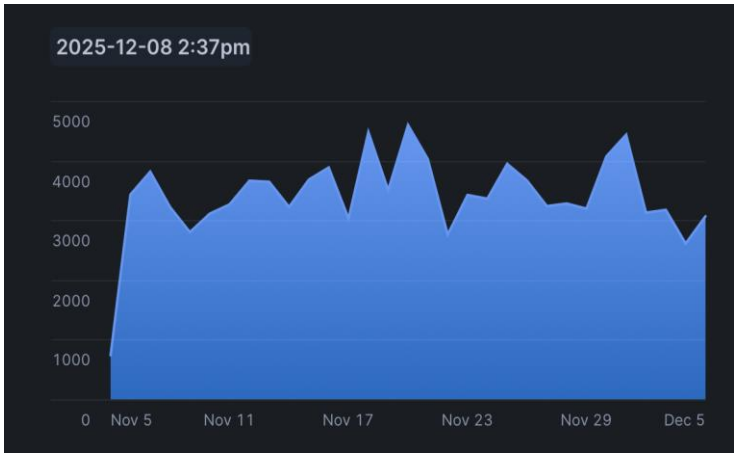
```
SELECT product_id, SUM(quantity)
FROM sales
GROUP BY product_id
ORDER BY 2 DESC;
```

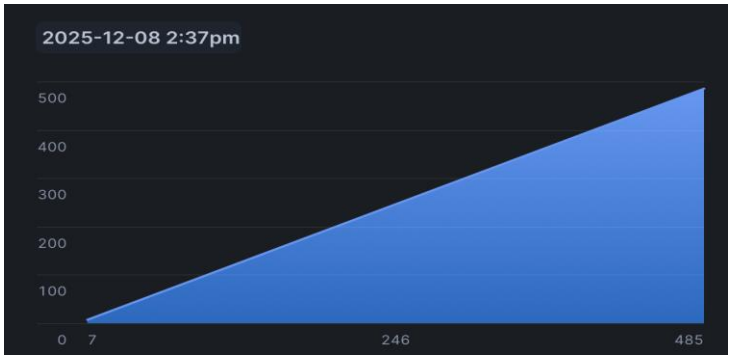
Bölge bazlı eğilim

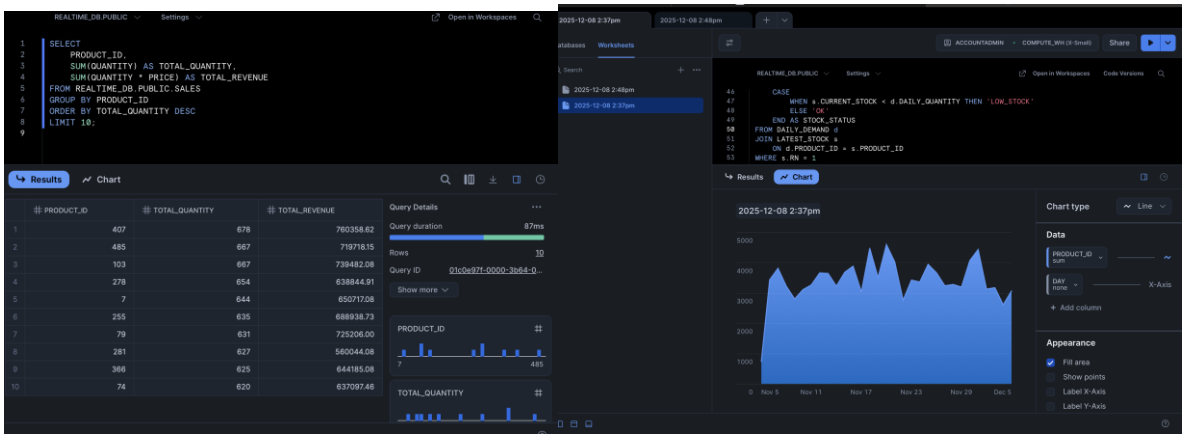
```
SELECT region, DATE(time), SUM(quantity)
FROM sales
GROUP BY region, DATE(time);
```

Stok alarmı

```
CASE WHEN current_stock < daily_quantity THEN 'LOW_STOCK'
```

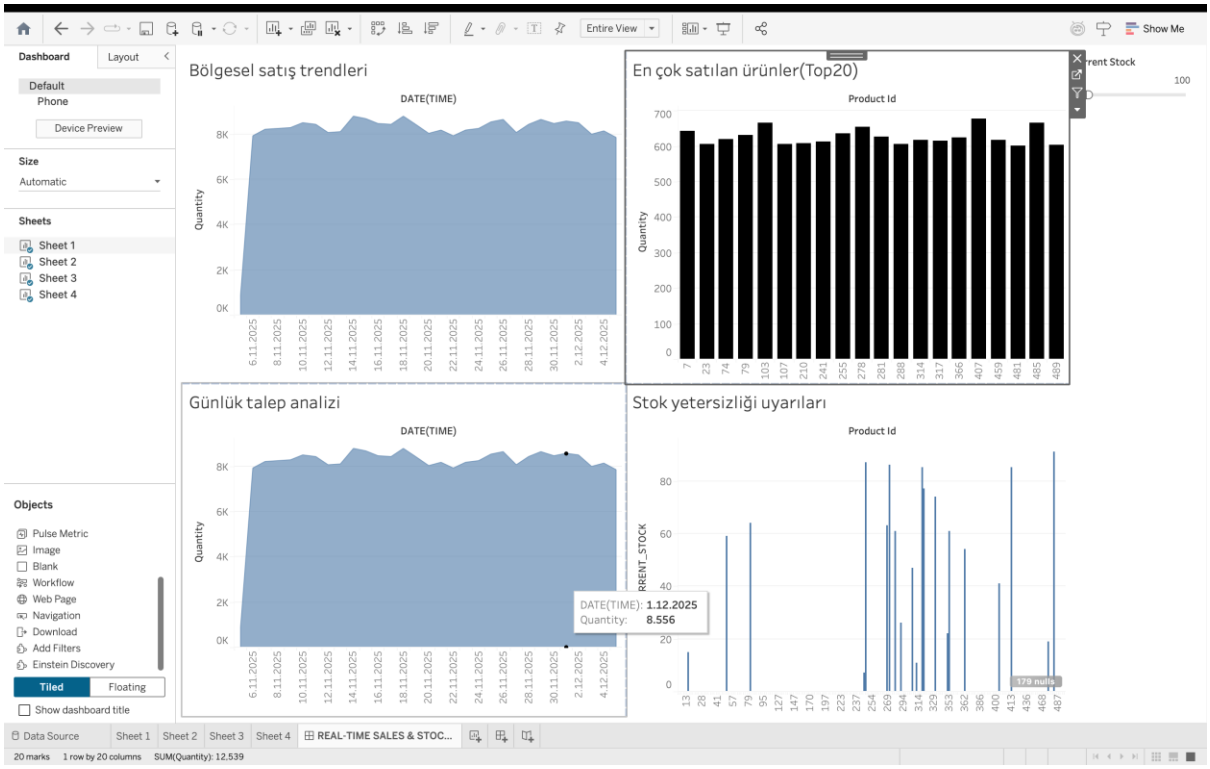






6. Tableau Dashboard Tasarımı

Üç kaynaktan gelen veri işlenip Tableau’da dört ana grafik ile dashboard oluşturulmuştur:



6.1 Bölgesel Satış Trendleri

✓ Line chart
✓ Zaman serisi
✓ Region renk ayrımı

6.2 Günlük Talep Analizi

✓ Area chart
✓ Talep eğilimi çıkarımı

6.3 En Çok Satılan Ürünler

✓ Bar chart
✓ Product_ID – SUM(Quantity)

6.4 Stok Yetersizliği Uyarıları

✓ Bar chart
✓ Filtre: Current_Stock < 100

7. Sonuç ve Değerlendirme

Bu proje ile modern bir e-ticaret firmasının ihtiyaç duyacağı **uçtan uca veri altyapısı** başarılı şekilde simüle edilmiştir.

Elde edilen çıktılar:

- Dask ile paralel büyük veri işlemleri yapılmıştır.
- Iceberg ile veri gölüne time-travel özellikli tablo yapısı kurulmuştur.
- Snowflake'e veri aktarılmış ve OLAP analizleri yapılmıştır.
- Tableau ile canlı dashboard geliştirilmiştir.

Bu yapılar birlikte çalışarak modern **Lakehouse + Cloud DW + BI** mimarisini temsil etmektedir.