**metin, ekran görüntüsü, diyagram, paralel içeren bir resim

Açıklama otomatik olarak oluşturulduNORTHWİND DATASET CAPSTONE PROJECT**

**ALTER TABLE employeeterritories**

**ADD CONSTRAINT fk\_employee\_id**

**FOREIGN KEY (employee\_id)**

**REFERENCES employees (employee\_id);**

**ALTER TABLE orders**

**ADD CONSTRAINT fk\_employee\_id\_orders**

**FOREIGN KEY (employee\_id)**

**REFERENCES employees (employee\_id);**

**ALTER TABLE order\_details**

**ADD CONSTRAINT fk\_order\_id**

**FOREIGN KEY (order\_id)**

**REFERENCES orders (order\_id);**

**ALTER TABLE customercustomerdemo**

**ADD CONSTRAINT fk\_customer\_id\_customerdemo**

**FOREIGN KEY (customer\_id)**

**REFERENCES customers (customer\_id);**

**ALTER TABLE orders**

**ADD CONSTRAINT fk\_customer\_id\_orders**

**FOREIGN KEY (customer\_id)**

**REFERENCES customers (customer\_id);**

**ALTER TABLE territories**

**ADD CONSTRAINT fk\_region\_id**

**FOREIGN KEY (region\_id)**

**REFERENCES region (region\_id);**

**ALTER TABLE products**

**ADD CONSTRAINT fk\_supplier\_id**

**FOREIGN KEY (supplier\_id)**

**REFERENCES suppliers (supplier\_id);**

**ALTER TABLE employeeterritories**

**ADD CONSTRAINT fk\_territory\_id**

**FOREIGN KEY (territory\_id)**

**REFERENCES territories (territory\_id);**

**ALTER TABLE order\_details**

**ADD CONSTRAINT fk\_product\_id**

**FOREIGN KEY (product\_id)**

**REFERENCES products (product\_id);**

**--Satış Performans Analizi**   
--Şirket yöneticileri, satış performansını detaylı bir şekilde görmek istiyor. Özellikle aşağıdaki bilgileri istemektedirler:

**--Toplam Satış Geliri**

**--En Çok Satılan Ürünler**

**--Ürün Bazında Toplam Gelir**

**--Aylık Satış Performansı**

**--En Çok Sipariş Alan Müşteriler**

**--Ortalama Sipariş Değeri**

**--En Çok Satış Yapılan Ülke**

**--1.Toplam satış geliri**

SELECT

ROUND (CAST (SUM (od.unit\_price \* od.quantity \* (1 - od.discount)) AS numeric), 2) AS total\_sales\_revenue

FROM

order\_details od;

**--2. En Çok Satılan Ürünler**

SELECT

p.product\_name,

SUM(od.quantity) AS total\_quantity\_sold

FROM

order\_details od

JOIN

products p ON od.product\_id = p.product\_id

GROUP BY

p.product\_name

ORDER BY

total\_quantity\_sold DESC

LIMIT 10;

**--3. Ürün Bazında Toplam Gelir**

SELECT

p.product\_name,

ROUND (SUM(od.unit\_price \* od.quantity)) AS product\_revenue

FROM

order\_details od

JOIN

products p ON od.product\_id = p.product\_id

GROUP BY

p.product\_name

ORDER BY

product\_revenue DESC;

**--4. Aylık Satış Performansı**

SELECT

EXTRACT (YEAR FROM o.order\_date) AS year,

EXTRACT (MONTH FROM o.order\_date) AS month,

SUM (od.unit\_price \* od.quantity) AS monthly\_sales\_revenue

FROM

orders o

JOIN

order\_details od ON o.order\_id = od.order\_id

GROUP BY

year, month

ORDER BY

year, month;

**--5. En Çok Sipariş Alan Müşteriler**

SELECT

c.customer\_id,

c.contact\_name,

COUNT(o.order\_id) AS total\_orders

FROM

customers c

JOIN

orders o ON c.customer\_id = o.customer\_id

GROUP BY

c.customer\_id, c.contact\_name

ORDER BY

total\_orders DESC

LIMIT 10;

**--6. Ortalama Sipariş Değeri**

SELECT

ROUND (SUM (od.unit\_price \* od.quantity \* (1 - od.discount))::NUMERIC / COUNT(o.order\_id), 2) AS avg\_order\_value

FROM

order\_details od

JOIN

orders o ON od.order\_id = o.order\_id;

**--7. En Çok Satış Yapılan Ülke**

SELECT

c.country,

SUM(od.unit\_price \* od.quantity \* (1 - od.discount)) AS total\_sales

FROM

customers c

JOIN orders o ON c.customer\_id = o.customer\_id

JOIN order\_details od ON o.order\_id = od.order\_id

GROUP BY

c.country

ORDER BY

total\_sales DESC

LIMIT 10;

**--MÜŞTERİ SEGMENTASYONU ANALİZİ**

--Şirket yöneticileri ve pazarlama ekibi, müşterilerin harcama alışkanlıklarını daha iyi anlayabilmek ve buna göre hedefli pazarlama stratejileri geliştirmek istiyor. Bu nedenle, müşteri segmentasyonu analizini talep etmektedirler. Analiz, farklı kriterlere göre müşterilerin sınıflandırılmasını içermektedir.

**--1. Toplam Harcamaya Göre Müşteri Segmentasyonu**

SELECT

c.customer\_id,

c.company\_name,

SUM(od.unit\_price \* od.quantity \* (1 - od.discount))::NUMERIC AS total\_spent

FROM

customers c

JOIN

orders o ON c.customer\_id = o.customer\_id

JOIN

order\_details od ON o.order\_id = od.order\_id

GROUP BY

c.customer\_id, c.company\_name

ORDER BY

total\_spent DESC

LIMIT 10;

**--2. Sipariş Sıklığına Göre Segmentasyon**

SELECT

c.customer\_id,

c.company\_name,

COUNT(o.order\_id) AS total\_orders

FROM

customers c

JOIN

orders o ON c.customer\_id = o.customer\_id

GROUP BY

c.customer\_id, c.company\_name

ORDER BY

total\_orders DESC;

**--3. Ortalama Sipariş Değerine Göre Segmentasyon**

SELECT

c.customer\_id,

c.company\_name,

ROUND(SUM(od.unit\_price \* od.quantity \* (1 - od.discount))::NUMERIC / COUNT(o.order\_id), 2) AS avg\_order\_value

FROM

customers c

JOIN

orders o ON c.customer\_id = o.customer\_id

JOIN

order\_details od ON o.order\_id = od.order\_id

GROUP BY

c.customer\_id, c.company\_name

ORDER BY

avg\_order\_value DESC;

**--4. En Aktif Müşterilerin Analizi (Son Sipariş Tarihine Göre)**

SELECT

c.customer\_id,

c.company\_name,

MAX(o.order\_date) AS last\_order\_date

FROM

customers c

JOIN

orders o ON c.customer\_id = o.customer\_id

GROUP BY

c.customer\_id, c.company\_name

ORDER BY

last\_order\_date DESC

LIMIT 10;

**--5. Coğrafi Bölgeye Göre Müşteri Dağılımı**

SELECT

c.country,

COUNT(DISTINCT c.customer\_id) AS total\_customers,

SUM(od.unit\_price \* od.quantity \* (1 - od.discount))::NUMERIC AS total\_spent

FROM

customers c

JOIN

orders o ON c.customer\_id = o.customer\_id

JOIN

order\_details od ON o.order\_id = od.order\_id

GROUP BY

c.country

ORDER BY

total\_spent DESC;

**--6. Son 6 Ay İçinde Sipariş Vermeyen Müşteriler**

SELECT

c.customer\_id,

c.company\_name

FROM

customers c

LEFT JOIN

orders o ON c.customer\_id = o.customer\_id

AND o.order\_date >= (SELECT MAX(order\_date) - INTERVAL '6 months' FROM orders)

GROUP BY

c.customer\_id, c.company\_name

HAVING

COUNT(o.order\_id) = 0;

**TEDARİKÇİ PERFORMANSI ANALİZİ**

--Şirket yöneticileri ve tedarik zinciri ekibi, tedarikçilerle olan iş ilişkilerini iyileştirmek ve tedarik performansını değerlendirmek için aşağıdaki bilgileri istemektedir:

**--Toplam Tedarik Miktarına Göre Tedarikçi Performansı**

**--Tedarikçiye Göre Toplam Satış Geliri**

**--Tedarikçi Başına Ortalama Ürün Fiyatı**

**--En Fazla Satılan Ürünleri Sağlayan Tedarikçiler**

**--Tedarikçiye Göre Sipariş Sıklığı**

**--Tedarikçiye Göre Son Sipariş Tarihi**

**--1. Toplam Tedarik Miktarına Göre Tedarikçi Performansı**

SELECT

s.supplier\_id,

s.company\_name,

SUM(od.quantity) AS total\_quantity\_supplied

FROM

suppliers s

JOIN

products p ON s.supplier\_id = p.supplier\_id

JOIN

order\_details od ON p.product\_id = od.product\_id

GROUP BY

s.supplier\_id, s.company\_name

ORDER BY

total\_quantity\_supplied DESC;

**--2. Tedarikçiye Göre Toplam Satış Geliri**

SELECT

s.supplier\_id,

s.company\_name,

SUM(od.unit\_price \* od.quantity \* (1 - od.discount))::NUMERIC AS total\_sales\_revenue

FROM

suppliers s

JOIN

products p ON s.supplier\_id = p.supplier\_id

JOIN

order\_details od ON p.product\_id = od.product\_id

GROUP BY

s.supplier\_id, s.company\_name

ORDER BY

total\_sales\_revenue DESC;

**--3. Tedarikçi Başına Ortalama Ürün Fiyatı**

SELECT

s.supplier\_id,

s.company\_name,

ROUND(AVG(p.unit\_price)::NUMERIC, 2) AS avg\_product\_price

FROM

suppliers s

JOIN

products p ON s.supplier\_id = p.supplier\_id

GROUP BY

s.supplier\_id, s.company\_name

ORDER BY

avg\_product\_price DESC;

**--4. En Fazla Satılan Ürünleri Sağlayan Tedarikçiler**

SELECT

s.supplier\_id,

s.company\_name,

p.product\_id,

p.product\_name,

SUM(od.quantity) AS total\_quantity\_sold

FROM

suppliers s

JOIN

products p ON s.supplier\_id = p.supplier\_id

JOIN

order\_details od ON p.product\_id = od.product\_id

GROUP BY

s.supplier\_id, s.company\_name, p.product\_id, p.product\_name

ORDER BY

total\_quantity\_sold DESC

LIMIT 10;

**--5. Tedarikçiye Göre Sipariş Sıklığı**

SELECT

s.supplier\_id,

s.company\_name,

COUNT(DISTINCT o.order\_id) AS total\_orders

FROM

suppliers s

JOIN

products p ON s.supplier\_id = p.supplier\_id

JOIN

order\_details od ON p.product\_id = od.product\_id

JOIN

orders o ON od.order\_id = o.order\_id

GROUP BY

s.supplier\_id, s.company\_name

ORDER BY

total\_orders DESC;

**--6. Tedarikçiye Göre Son Sipariş Tarihi**

SELECT

s.supplier\_id,

s.company\_name,

MAX(o.order\_date) AS last\_order\_date

FROM

suppliers s

JOIN

products p ON s.supplier\_id = p.supplier\_id

JOIN

order\_details od ON p.product\_id = od.product\_id

JOIN

orders o ON od.order\_id = o.order\_id

GROUP BY

s.supplier\_id, s.company\_name

ORDER BY

last\_order\_date DESC;

**--SİPARİŞ TESLİMAT PERFORMANSI ANALİZİ**

--Şirket yönetimi ve lojistik ekibi, sipariş teslimat süreçlerini iyileştirmek ve teslimat performansını değerlendirmek için aşağıdaki bilgilere ihtiyaç duymaktadır:

**--Ortalama Teslimat Süresi**

**--Geç Teslim Edilen Siparişler**

**--En Uzun Teslimat Süreli Ürünler**

**--En Başarılı Satış Temsilcisi**

**--1. Ortalama Teslimat Süresi**

SELECT

c.customer\_id,

c.company\_name,

COUNT(o.order\_id) AS total\_orders,

SUM(od.unit\_price \* od.quantity \* (1 - od.discount))::NUMERIC AS total\_spent,

ROUND(SUM(od.unit\_price \* od.quantity \* (1 - od.discount))::NUMERIC / COUNT(o.order\_id), 2) AS avg\_order\_value,

ROUND(AVG(o.shipped\_date - o.order\_date), 2) AS avg\_delivery\_days

FROM

customers c

JOIN

orders o ON c.customer\_id = o.customer\_id

JOIN

order\_details od ON o.order\_id = od.order\_id

WHERE

o.shipped\_date IS NOT NULL

GROUP BY

c.customer\_id, c.company\_name;

**--2. Geç Teslim Edilen Siparişler**

SELECT

o.order\_id,

o.order\_date,

o.shipped\_date,

(o.shipped\_date - o.order\_date) :: INT AS delivery\_days

FROM

orders o

WHERE

o.shipped\_date IS NOT NULL

AND (o.shipped\_date - o.order\_date) :: INT > 7;

**--3. En Uzun Teslimat Süreli Ürünler**

SELECT

p.product\_id,

p.product\_name,

ROUND(AVG((o.shipped\_date - o.order\_date) :: INT), 2) AS avg\_delivery\_days

FROM

products p

JOIN

order\_details od ON p.product\_id = od.product\_id

JOIN

orders o ON od.order\_id = o.order\_id

WHERE

o.shipped\_date IS NOT NULL

GROUP BY

p.product\_id, p.product\_name

ORDER BY

avg\_delivery\_days DESC

LIMIT 10;

**--4.En Başarılı Satış Temsilcisi**

SELECT

e.employee\_id,

e.first\_name,

e.last\_name,

SUM(od.unit\_price \* od.quantity \* (1 - od.discount))::NUMERIC AS total\_sales,

ROUND(AVG(o.shipped\_date - o.order\_date), 2) AS avg\_delivery\_days

FROM

employees e

JOIN

orders o ON e.employee\_id = o.employee\_id

JOIN

order\_details od ON o.order\_id = od.order\_id

WHERE

o.shipped\_date IS NOT NULL

GROUP BY

e.employee\_id, e.first\_name, e.last\_name

ORDER BY

total\_sales DESC

LIMIT 1;