**ERD ŞEMALARI**

ALTER TABLE employeeterritories

ADD CONSTRAINT fk\_employee\_territory

FOREIGN KEY (employee\_id)

REFERENCES employees(employee\_id);

ALTER TABLE order\_details

ADD CONSTRAINT fk\_order

FOREIGN KEY (order\_id)

REFERENCES orders(order\_id);

ALTER TABLE order\_details

ADD CONSTRAINT fk\_product

FOREIGN KEY (product\_id)

REFERENCES products(product\_id);

ALTER TABLE orders

ADD CONSTRAINT pk\_order

PRIMARY KEY (order\_id);

ALTER TABLE orders

ADD CONSTRAINT fk\_customer\_order

FOREIGN KEY (customer\_id)

REFERENCES customers(customer\_id);

ALTER TABLE products

ADD CONSTRAINT pk\_product

PRIMARY KEY (product\_id);

ALTER TABLE products

ADD CONSTRAINT fk\_supplier\_product

FOREIGN KEY (supplier\_id)

REFERENCES suppliers(supplier\_id);

ALTER TABLE suppliers

ADD CONSTRAINT pk\_supplier

PRIMARY KEY (supplier\_id);

ALTER TABLE categories

ADD CONSTRAINT pk\_category

PRIMARY KEY (category\_id);

ALTER TABLE products

ADD CONSTRAINT fk\_category\_product

FOREIGN KEY (category\_id)

REFERENCES categories(category\_id);

ALTER TABLE orders

ADD CONSTRAINT fk\_employee\_order

FOREIGN KEY (employee\_id)

REFERENCES employees(employee\_id);

ALTER TABLE orders

ADD CONSTRAINT fk\_shipper\_order

FOREIGN KEY (ship\_via)

REFERENCES shippers(shipper\_id);

ALTER TABLE employeeterritories

ADD CONSTRAINT fk\_territory\_employee

FOREIGN KEY (territory\_id)

REFERENCES territories(territory\_id);

ALTER TABLE customercustomerdemo

ADD CONSTRAINT fk\_customer

FOREIGN KEY (customer\_id)

REFERENCES customers(customer\_id);

ALTER TABLE customercustomerdemo

ADD CONSTRAINT fk\_customer\_type

FOREIGN KEY (customer\_type\_id)

REFERENCES customerdemographics(customer\_type\_id);

**Sipariş Analizi**

**Kargo detayları**

SELECT

EXTRACT(YEAR FROM o.shipped\_date) AS shipped\_years,

TO\_CHAR(o.shipped\_date, 'Month') AS shipped\_months,

s.shipper\_id,

s.company\_name AS cargo\_name,

COUNT(o.order\_id) AS toplam\_urun\_adedi

FROM

orders AS o

JOIN

shippers AS s ON o.ship\_via = s.shipper\_id

WHERE

o.shipped\_date IS NOT NULL

AND s.shipper\_id IS NOT NULL

GROUP BY

EXTRACT(YEAR FROM o.shipped\_date),

TO\_CHAR(o.shipped\_date, 'Month'),

s.shipper\_id,

s.company\_name

ORDER BY

toplam\_urun\_adedi DESC;

**Şirketin Yıllık Ciro Performansının Analizi**

SELECT

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

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

FROM

orders o

JOIN

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

GROUP BY

EXTRACT(YEAR FROM o.order\_date)

ORDER BY

total\_price DESC;

**Aylık Ciro Analizi**

SELECT

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

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

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

FROM

orders AS o

JOIN

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

GROUP BY

EXTRACT(MONTH FROM o.order\_date),

EXTRACT(YEAR FROM o.order\_date)

ORDER BY

total\_price DESC;

**En Çok Satan ve En Çok Gelir Getiren Ürünlerin Analizi**

SELECT

SUM(od.quantity) AS total\_quantity,

p.product\_name

FROM

products p

JOIN

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

GROUP BY

p.product\_name

ORDER BY

total\_quantity DESC;

WITH total\_price\_per\_month AS (

SELECT

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

TO\_CHAR(o.order\_date, 'Month') AS month,

ROUND(SUM(od.quantity \* od.unit\_price)) AS total\_price,

p.product\_name

FROM

products p

JOIN

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

JOIN

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

GROUP BY

EXTRACT(YEAR FROM o.order\_date),

TO\_CHAR(o.order\_date, 'Month'),

p.product\_name

)

SELECT

product\_name,

ROUND(SUM(total\_price)) AS total\_revenue

FROM

total\_price\_per\_month

GROUP BY

product\_name

ORDER BY

total\_revenue DESC;

**En Çok Satış Yapılan Kategori ve Bu Kategoriye Ait Ürünler**

SELECT

cat.category\_name,

p.product\_name,

COUNT(od.order\_id) AS sales\_count

FROM

products AS p

JOIN

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

JOIN

categories AS cat ON p.category\_id = cat.category\_id

GROUP BY

cat.category\_name,

p.product\_name

ORDER BY

sales\_count DESC;

SELECT

COUNT(\*) AS category\_count

FROM

categories;

**Farklı Kategorilerdeki Ürünlerin Ortalama Fiyatı ve Toplam Satış Tutarı**

SELECT

ct.category\_name AS category,

p.product\_name AS product,

ROUND(AVG(od.unit\_price)) AS average\_price,

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

FROM

products AS p

JOIN

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

JOIN

categories AS ct ON p.category\_id = ct.category\_id

GROUP BY

ct.category\_name,

p.product\_name

ORDER BY

average\_price DESC;

**Yıllara Göre En Çok Satan Ürünler**

SELECT

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

p.product\_name AS product,

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

FROM

orders AS o

JOIN

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

JOIN

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

GROUP BY

EXTRACT(YEAR FROM o.order\_date),

p.product\_name

ORDER BY

total\_price DESC;

**En Çok Satış Yapılan Ülke**

SELECT

o.ship\_country AS country,

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

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

FROM

orders o

JOIN

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

GROUP BY

o.ship\_country

ORDER BY

total\_orders DESC;

**En Çok Getiri Sağlayan Ülke**

SELECT

o.ship\_country AS country,

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

FROM

orders o

JOIN

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

GROUP BY

o.ship\_country

ORDER BY

revenue DESC;

**Müşteri Dağılımı ve Bölge Analizi**

SELECT

c.country AS country,

c.city AS city,

COUNT(c.customer\_id) AS customer\_count

FROM

customers c

GROUP BY

c.country,

c.city

ORDER BY

customer\_count DESC;

**RFM Analizi ile Müşteri Segmentasyonu**

WITH recency\_data AS (

SELECT

customer\_id AS customer,

(SELECT MAX(order\_date) FROM orders) - MAX(order\_date) AS recency

FROM

orders

GROUP BY

customer\_id

),

frequency\_data AS (

SELECT

customer\_id,

COUNT(order\_id) AS total\_orders

FROM

orders

GROUP BY

customer\_id

),

monetary\_data AS (

SELECT

o.customer\_id,

ROUND(SUM(od.unit\_price \* od.quantity)) AS total\_spent

FROM

order\_details od

JOIN

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

GROUP BY

o.customer\_id

),

recency\_scores AS (

SELECT

customer,

recency,

NTILE(5) OVER (ORDER BY recency) AS recency\_score

FROM

recency\_data

),

frequency\_scores AS (

SELECT

customer\_id,

total\_orders,

NTILE(5) OVER (ORDER BY total\_orders DESC) AS frequency\_score

FROM

frequency\_data

),

monetary\_scores AS (

SELECT

customer\_id,

total\_spent AS monetary,

NTILE(5) OVER (ORDER BY total\_spent DESC) AS monetary\_score

FROM

monetary\_data

),

rfm\_scores AS (

SELECT

r.customer AS customer\_id,

r.recency,

f.total\_orders AS frequency,

m.monetary,

r.recency\_score,

f.frequency\_score,

m.monetary\_score,

CONCAT(r.recency\_score::text, f.frequency\_score::text) AS rfm\_score

FROM

recency\_scores r

JOIN

frequency\_scores f ON r.customer = f.customer\_id

JOIN

monetary\_scores m ON r.customer = m.customer\_id

)

SELECT

customer\_id,

recency,

frequency,

monetary,

recency\_score,

frequency\_score,

monetary\_score,

rfm\_score,

CASE

WHEN rfm\_score ~ '^[1-2][1-2]$' THEN 'Hibernating'

WHEN rfm\_score ~ '^[1-2][3-4]$' THEN 'At\_risk'

WHEN rfm\_score ~ '^[1-2]5$' THEN 'Cant\_loose'

WHEN rfm\_score ~ '^3[1-2]$' THEN 'About\_the\_sleep'

WHEN rfm\_score = '33' THEN 'Need\_attention'

WHEN rfm\_score ~ '^[3-4][4-5]$' THEN 'Loyal\_customers'

WHEN rfm\_score = '41' THEN 'Promising'

WHEN rfm\_score = '51' THEN 'New\_customers'

WHEN rfm\_score ~ '^[4-5][2-3]$' THEN 'Potential\_loyalists'

WHEN rfm\_score ~ '^5[4-5]$' THEN 'Champions'

ELSE 'Other'

END AS customer\_segment

FROM

rfm\_scores;

**Satış Performansı Değerlendirmesi**

WITH employee\_prices AS (

SELECT

e.first\_name || ' ' || e.last\_name AS full\_name,

e.title AS employee\_title,

p.product\_name AS product,

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

COUNT(od.quantity) AS total\_quantity,

ROUND(AVG(od.unit\_price)) AS average\_price,

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

FROM

orders AS o

JOIN

employees AS e ON o.employee\_id = e.employee\_id

JOIN

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

JOIN

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

GROUP BY

e.first\_name, e.last\_name, e.title, p.product\_name, EXTRACT(YEAR FROM o.order\_date)

ORDER BY

total\_revenue DESC

)

SELECT

full\_name,

employee\_title,

product,

order\_year,

total\_quantity,

average\_price,

total\_revenue

FROM

employee\_prices;

**Çalışanlarımız Bize Hangi Bölgeden Destek Veriyor**

SELECT

e.country AS country\_name,

e.city AS city\_name,

COUNT(e.employee\_id) AS employee\_count

FROM

employees e

GROUP BY

e.country, e.city

ORDER BY

employee\_count DESC;

**Çalışanlarımızın Yaş Ortalaması**

WITH employee\_age AS (

SELECT

CONCAT(first\_name, ' ', last\_name) AS employee\_name,

EXTRACT(YEAR FROM CURRENT\_DATE) - EXTRACT(YEAR FROM birth\_date) AS age

FROM

employees

)

SELECT

employee\_name,

age,

CASE

WHEN age BETWEEN 20 AND 30 THEN '20-30'

WHEN age BETWEEN 31 AND 40 THEN '30-40'

WHEN age BETWEEN 41 AND 50 THEN '40-50'

WHEN age BETWEEN 51 AND 60 THEN '50-60'

ELSE '60+'

END AS age\_group

FROM

employee\_age;