

-- Soru 1: Tum musterilerin listesini alin.

```
SELECT * FROM Customers;
```

-- Cikti Aciklamasi: Musteri bilgileri listelenir.

-- Aciklama: Customers tablosundaki tum kayitlar getirilir.

-- Soru 2: Almanya'daki musterileri listeleyin.

```
SELECT CompanyName, ContactName, Country FROM Customers WHERE Country = 'Germany';
```

-- Cikti Aciklamasi: Almanya'da bulunan musterilerin bilgileri doner.

-- Aciklama: WHERE kosulu ile filtreleme yapilir. Lokasyon bazli analizde faydali olur.

-- Soru 3: Mevcut urunlerin adlarini ve fiyatlarini listeleyin.

```
SELECT ProductName, UnitPrice FROM Products;
```

-- Cikti Aciklamasi: Tum urunlerin isimleri ve fiyatları listelenir.

-- Aciklama: Urunlerin fiyatlandırma bilgileri raporlanabilir. Maliyet analizine ilk adimdir.

-- Soru 4: Sadece sirket isimlerini listeleyin.

```
SELECT CompanyName FROM Customers;
```

-- Cikti Aciklamasi: Tum musterilerin sirket isimleri doner.

-- Aciklama: Musteri adi odakli analiz yapılabilir.

-- Soru 5: Urunleri fiyata gore azalan sirayla listeleyin.

```
SELECT ProductName, UnitPrice FROM Products ORDER BY UnitPrice DESC;
```

-- Cikti Aciklamasi: Urunler pahalidan ucuza siralanir.

-- Aciklama: Urun fiyat analizleri ve kampanya planlamasi icin uygundur.

-- Soru 6: Kategori isimlerini benzersiz olarak getirin.

```
SELECT DISTINCT CategoryName FROM Categories;
```

-- Cikti Aciklamasi: Tekil kategori isimleri doner.

-- Aciklama: Kategori cesitliliği görüntülenebilir.

-- Soru 7: Telefon numarasi belirtilmis musterileri listeleyin.

```
SELECT ContactName, Phone FROM Customers WHERE Phone IS NOT NULL;
```

-- Cikti Aciklamasi: Telefon bilgisi olan musteriler doner.

-- Aciklama: Iletisim bilgisi tam olan musteriler filtrelendir.

-- Soru 8: Ilk 10 urunu ada gore listeleyin.

```
SELECT TOP 10 ProductName FROM Products ORDER BY ProductName;
```

-- Cikti Aciklamasi: Alfabetik sirayla ilk 10 urun listelenir.

-- Aciklama: Hizli urun gozden gecirme icin faydali.

-- Soru 9: Stokta olmayan urunleri bulun.

```
SELECT ProductName FROM Products WHERE UnitsInStock = 0;
```

-- Cikti Aciklamasi: Stokta kalmamis urunler gosterilir.

-- Aciklama: Stok yonetimi ve tedarik planlamasi icin onemlidir.

-- Soru 10: Urun fiyatlarini ortalama fiyatla karsilastirin.

```
SELECT ProductName, UnitPrice FROM Products WHERE UnitPrice > (SELECT AVG(UnitPrice) FROM Products);
```

-- Cikti Aciklamasi: Ortalamanin ustundeki fiyatli urunler listelenir.

-- Aciklama: Fiyat segment analizi icin kullanilir.

-- Soru 11: 20 adetten fazla stokta olan urunleri gosterin.

```
SELECT ProductName, UnitsInStock FROM Products WHERE UnitsInStock > 20;
```

-- Cikti Aciklamasi: Yuksek stoklu urunler gorunur.

-- Aciklama: Stok fazlasi yonetimi icin kullanislidir.

-- Soru 12: Gonderici firmalari alfabetik sirayla listeleyin.

```
SELECT CompanyName FROM Shippers ORDER BY CompanyName ASC;
```

-- Cikti Aciklamasi: Kargo firmalari isim sirasina gore gelir.

-- Aciklama: Kargo firmalarinin yonetimi icin baslangic listesi olusturulabilir.

-- Soru 13: Her siparisin urun adedini listeleyin.

```
SELECT OrderID, SUM(Quantity) AS TotalItems FROM OrderDetails GROUP BY OrderID;
```

-- Cikti Aciklamasi: Siparislerdeki toplam urun adedi gosterilir.

-- Aciklama: Siparis yogunlugu olculebilir.

-- Soru 14: Her calisanin aldigi siparis sayisini bulun.

```
SELECT e.FirstName, e.LastName, COUNT(o.OrderID) AS OrderCount  
FROM Employees e  
JOIN Orders o ON e.EmployeeID = o.EmployeeID  
GROUP BY e.FirstName, e.LastName;
```

-- Cikti Aciklamasi: Her calisanin kac siparis aldigi listelenir.

-- Aciklama: JOIN ve GROUP BY kullanilarak calisan performanslari degerlendirilebilir

-- Soru 15: 1997 yilinda verilen siparisleri listeleyin.

```
SELECT * FROM Orders  
WHERE YEAR(OrderDate) = 1997;
```

-- Cikti Aciklamasi: 1997 yilina ait siparisler doner.

-- Aciklama: Zaman bazli analizde kullanilabilir.

-- Soru 16: Urunleri kategoriye gore gruplayin ve ortalama fiyatlarını hesaplayın.

```
SELECT c.CategoryName, AVG(p.UnitPrice) AS AvgPrice
FROM Categories c
JOIN Products p ON c.CategoryID = p.CategoryID
GROUP BY c.CategoryName;
```

-- Çıktı Açıklaması: Kategori bazlı ortalama ürün fiyatları gösterilir.

-- Açıklama: Ürün gruplarında fiyat trendi analiz edilebilir.

-- Soru 17: Her çalışanın gönderdiği son siparisi bulun.

```
SELECT o.*
FROM Orders o
INNER JOIN (
  SELECT EmployeeID, MAX(OrderDate) AS LastOrder
  FROM Orders
  GROUP BY EmployeeID
) latest ON o.EmployeeID = latest.EmployeeID AND o.OrderDate = latest.LastOrder;
```

-- Çıktı Açıklaması: Çalışanların en son islediği siparişler döner.

-- Açıklama: Aktivite düzeyini görmek için kullanılır.

-- Soru 18: 1996 ve 1997 yılları arasındaki siparişlerin toplam tutarını hesaplayın.

```
SELECT SUM(od.UnitPrice * od.Quantity) AS TotalRevenue
FROM Orders o
JOIN OrderDetails od ON o.OrderID = od.OrderID
WHERE YEAR(OrderDate) BETWEEN 1996 AND 1997;
```

-- Çıktı Açıklaması: Belirtilen yıllar arasındaki toplam gelir döner.

-- Açıklama: Dönemsel gelir analizi yapılabilir.

-- Soru 19: Her ürün için minimum, maksimum ve ortalama sipariş miktarını hesaplayın.

```
SELECT ProductID, MIN(Quantity) AS MinQty, MAX(Quantity) AS MaxQty, AVG(Quantity) AS AvgQty
FROM OrderDetails
GROUP BY ProductID;
```

-- Çıktı Açıklaması: Ürün bazlı sipariş dağılımı döner.

-- Açıklama: Ürün taleplerindeki değişkenliği analiz etmede kullanılır.

-- Soru 20: Her siparis için siparis sirasini tarihe görebelirleyin.

```
SELECT CustomerID, OrderID, OrderDate,  
       ROW_NUMBER() OVER (ORDER BY OrderDate) AS OrderSeq_ROW,  
       RANK() OVER (ORDER BY OrderDate) AS OrderSeq_RANK,  
       DENSE_RANK() OVER (ORDER BY OrderDate) AS OrderSeq_DENSE_RANK  
FROM Orders;
```

-- Cikti Aciklamasi: Musteri bazli siparis siralamasi elde edilir.

-- Aciklama: Musteri davranislarini incelemek için idealdir.

-- Soru 21: Her calisanin isleme aldigi ilk siparisi bulun.

```
SELECT o.*  
FROM Orders o  
JOIN (  
    SELECT EmployeeID, MIN(OrderDate) AS FirstOrder  
    FROM Orders  
    GROUP BY EmployeeID  
) firsts ON o.EmployeeID = firsts.EmployeeID AND o.OrderDate = firsts.FirstOrder;
```

-- Cikti Aciklamasi: Calisanların ilk siparis kayitlari doner.

-- Aciklama: Calisma gecmisi ve musteriler kazanim sureci analiz edilebilir.

-- Soru 22: Her siparis için kargo teslim suresini hesaplayin.

```
SELECT OrderID, DATEDIFF(DAY, OrderDate, ShippedDate) AS DeliveryTime  
FROM Orders  
WHERE ShippedDate IS NOT NULL;
```

-- Cikti Aciklamasi: Teslimat suresi gun olarak gosterilir.

-- Aciklama: Lojistik surec analizi için önemli bir metriktir.

-- Soru 23: Musteri basina son 3 siparisin ortalama tutarini hesaplayin.

```
WITH RankedOrders AS (  
    SELECT o.CustomerID, o.OrderID, SUM(od.UnitPrice * od.Quantity) AS OrderTotal,  
    ROW_NUMBER() OVER (PARTITION BY o.CustomerID ORDER BY o.OrderDate DESC) AS rn  
    FROM Orders o  
    JOIN OrderDetails od ON o.OrderID = od.OrderID  
    GROUP BY o.CustomerID, o.OrderID, o.OrderDate)  
SELECT CustomerID, AVG(OrderTotal) AS AvgLast3  
FROM RankedOrders WHERE rn <= 3  
GROUP BY CustomerID;
```

-- Cikti Aciklamasi: Her musteriler için son 3 siparisin ortalama tutari doner.

-- Aciklama: Sadakat ve guncel harcama davranislarini olcmek için kullanilir.

-- Soru 24: CASE WHEN ile urun stok durumunu metinsel olarak gosterin.

```
SELECT ProductName,  
UnitsInStock,  
CASE  
WHEN UnitsInStock = 0 THEN 'Stokta yok'  
WHEN UnitsInStock < 20 THEN 'Az stok'  
ELSE 'Yeterli stok'  
END AS StokDurumu  
FROM Products;
```

-- Cikti Aciklamasi: Her urun icin stok seviyesine gore aciklama verilir.

-- Aciklama: Depo yonetimi ve uyarı sistemi icin kullanılabilir.

-- Soru 25: HAVING kullanarak yalnızca 1000 birimden fazla satan ürünleri gosterin.

```
SELECT ProductID, SUM(Quantity) AS ToplamAdet  
FROM OrderDetails  
GROUP BY ProductID  
HAVING SUM(Quantity) > 1000;
```

-- Cikti Aciklamasi: Sadece yuksek hacimli satis yapan urunler doner.

-- Aciklama: Stok planlamasi ve kampanya icin uygundur.

-- Soru 26: ROW_NUMBER ile her calisanin siparis siralamasini bulun.

```
SELECT EmployeeID, OrderID, OrderDate,  
ROW_NUMBER() OVER (PARTITION BY EmployeeID ORDER BY OrderDate) AS SiparisSira  
FROM Orders;
```

□ ROW_NUMBER(): Satırlara sıra numarası verir.

□ OVER: Bu sıralama işlemi nerede ve nasıl uygulanacak, onu belirtir.

□ PARTITION BY: Veriyi gruplara ayırır. Her grup içinde yeniden 1'den başlar.

-- Cikti Aciklamasi: Her calisanin siparislerine sirali numara verilir.

-- Aciklama: Zaman bazli calisan performansi analizi yapılabilir.

-- Soru 27: ROUND kullanarak siparis tutarlarını 1 ondalıkla gosterin.

```
SELECT OrderID, ROUND(SUM(UnitPrice * Quantity), 2) AS ToplamTutar  
FROM OrderDetails  
GROUP BY OrderID;
```

-- Cikti Aciklamasi: Her siparisin tutari 1 ondalik basamakla gosterilir.

-- Aciklama: Finansal raporlarda daha sade gosterim saglar.

-- Soru 28: FORMAT fonksiyonu ile satis tutarini para formatinda gosterin.

```
SELECT OrderID, FORMAT(SUM(UnitPrice * Quantity), 'C', 'en-US') AS ParaFormati
FROM OrderDetails
GROUP BY OrderID;
```

-- Cikti Aciklamasi: Siparis tutari USD formatinda doner (ornek: \$1,234.56).

-- Aciklama: Raporlar icin gorsel okunabilirliği artırır.

```
SELECT ProductName, FORMAT(UnitPrice, 'N', 'tr-TR') AS Fiyat_TR
FROM Products;
```

--Soru 29: Aynı satış adetlerine sahip ürünler için hem RANK() hem de DENSE_RANK() kullanarak farklarını karşılaştırın.

```
SELECT
    ProductID,
    SUM(Quantity) AS TotalSold,
    RANK() OVER (ORDER BY SUM(Quantity) DESC) AS RankOrder,
    DENSE_RANK() OVER (ORDER BY SUM(Quantity) DESC) AS DenseRankOrder
FROM OrderDetails
GROUP BY ProductID;
```

--Soru 30: Ürün adetlerinde trend analizi yapınız.

```
SELECT
    ProductID,
    SUM(Quantity) AS TotalSold,
    LAG(SUM(Quantity)) OVER (ORDER BY SUM(Quantity) DESC) AS OncekiSatis
FROM OrderDetails
GROUP BY ProductID;
```

```
SELECT
    ProductID,
    SUM(Quantity) AS TotalSold,
    LEAD(SUM(Quantity)) OVER (ORDER BY SUM(Quantity) DESC) AS SonrakiSatis
FROM OrderDetails
GROUP BY ProductID;
```