Database Bootcamp – Ödev 1

Bu ödevde, verilen SQL sorularına uygun sorgular yazıldı ve her birinin açıklaması detaylandırıldı.

1. Soru: Çalışanların sadece FirstName, LastName ve Salary bilgilerini getiren bir SQL sorgusu yazınız.

SQL Sorgusu:

SELECT firstname, lastname, salary

FROM employees;



Bu sorguda, employees tablosundan SELECT sorgusu ile firstname, lastname ve salary sütunlarını seçtim.

FROM ifadesi, bu verilerin hangi tablodan çekileceğini belirtir. Ayrıca, ihtiyaca göre belirli sütunları ayrı ayrı sorgulayarak da veri çekebiliriz.

SELECT firstname FROM employees

SELECT lastname FROM employees

SELECT salary FROM employees

Gibi sorgularla yalnızca istenen sütun da tablodan çekilebilir.

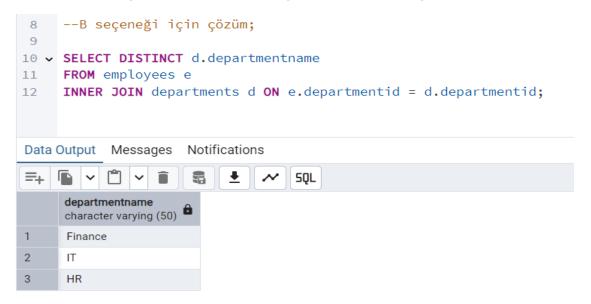
2. Soru: Çalışanların çalıştıkları departmanları benzersiz olarak listeleyen bir SQL sorgusu yazınız.

SQL Sorgusu:

SELECT DISTINCT d.departmentname

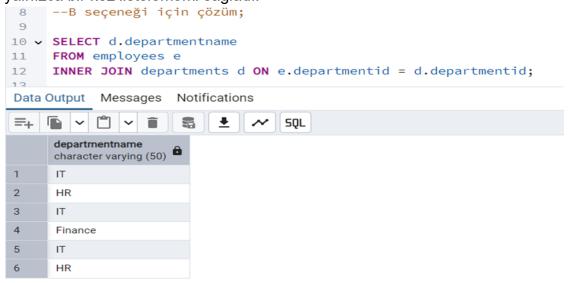
FROM employees e

INNER JOIN departments d ON e.departmentid = d.departmentid;



Bu sorguda, employees ve departments tablolarını **INNER JOIN** ile departmentid sütunu üzerinden birleştirdim. **INNER JOIN** tercih etmemin sebebi, yalnızca eşleşen satırları getirmek istemem ve bu işlemin **JOIN** işlemine kıyasla daha düşük maliyetli olmasıdır.

Daha sonra, çalışanların bağlı olduğu departmentname sütununu sorguya dahil ettim. **DISTINCT** ifadesi, tekrar eden departman isimlerini engelleyerek her departmanı yalnızca bir kez listelememi sağladı.



Böylece, sistemde kayıtlı ve en az bir çalışanı bulunan benzersiz departmanların listesini elde ettim. Eğer **DISTINCT** kullanmasaydım, employees tablosunda yer alan tüm çalışanların departmanları tekrar edecek şekilde listelenecekti.

3. Soru: Sadece IT departmanında çalışanların bilgilerini getiren bir SQL sorgusu yazınız.

SQL Sorgusu:

SELECT * FROM employees e

INNER JOIN departments d **ON** e.departmentid = d.departmentid

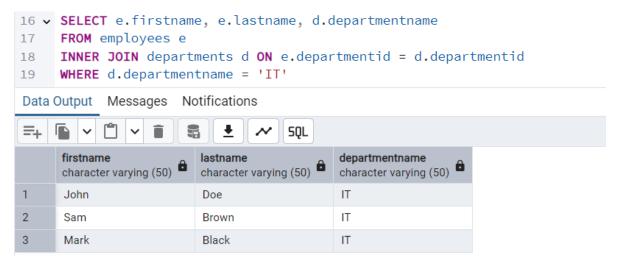
WHERE d.departmentname = 'IT'



Bu sorguda, employees tablosundaki tüm sütunları seçtim ve **INNER JOIN** kullanarak departments tablosuyla departmentid sütunu üzerinden birleştirdim.

WHERE koşulunda yalnızca departmentname değeri **"IT"** olan satırların getirilmesini sağladım.

Ayrıca, tablo adlarını **employees e** ve **departments d** olarak kısaltarak kullandım. Bu sayede, departments.departmentname yerine d.departmentname gibi daha sade bir yazım kullanarak sorguyu daha okunaklı ve kullanışlı hale getirdim.



Burada ise yalnızca **IT** departmanında çalışan kişilerin adını ve soyadını tabloda görmek istedim, böylece sorgunun daha sade ve anlaşılır olmasını sağladım.

Bunu sağlamak için, önceki örnekte olduğu gibi **SELECT** sorgusunu kullandım ve yalnızca ihtiyacım olan sütunları seçtim.

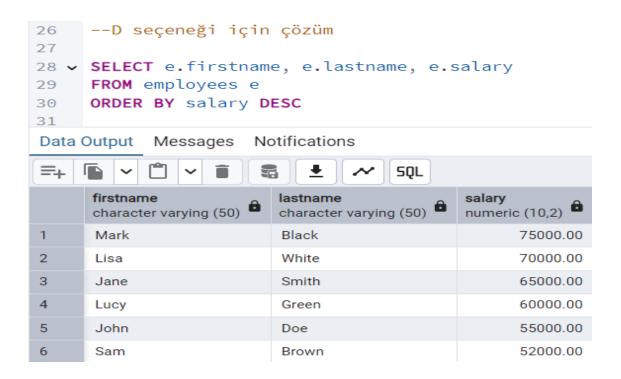
4. Soru: Çalışanları maaşlarına göre büyükten küçüğe sıralayan bir SQL sorgusu yazınız.

SQL Sorgusu:

SELECT e.firstname, e.lastname, e.salary

FROM employees e

ORDER BY salary DESC



Bu sorguda, employees tablosundan firstname, lastname ve salary sütunlarını seçtim. **ORDER BY** ifadesi, sorgu sonucundaki verileri belirli bir sütuna göre sıralamak için kullanılır.

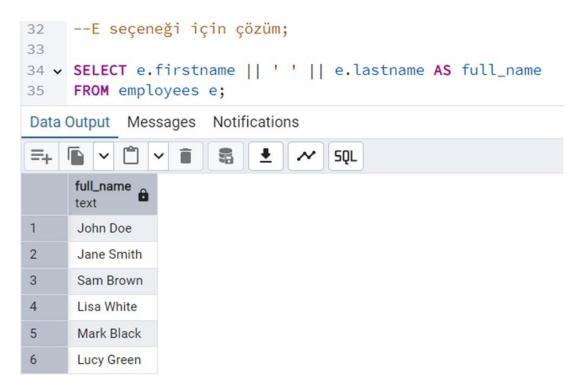
Bu örnekte, sıralamanın salary sütununa göre yapılması gerektiği için **ORDER** BY salary ifadesini kullandım. Maaşların büyükten küçüğe sıralanmasını sağlamak için **DESC (Azalan sıralama)** ifadesini ekledim.

5. Soru: Çalışanların FirstName ve LastName alanlarını birleştirerek, tam adlarını içeren yeni bir sütun oluşturan bir SQL sorgusu yazınız.

SQL Sorgusu:

SELECT e.firstname | | ' ' | | e.lastname **AS** full_name

FROM employees e;



Bu sorguda, çalışanların **firstname** ve **lastname** sütunlarını birleştirerek tam isimlerini içeren **full_name** adında yeni bir sütun oluşturdum. **CONCAT** fonksiyonu, MySQL gibi veritabanlarında metinleri birleştirmek için kullanılır. **PostgreSQL** gibi sistemlerde ||''|| operatörü kullanılsa da **Tuncay hocamız**, MySQL'de **CONCAT** fonksiyonunun kullanılması gerektiğini belirtti.

Ayrıca, '' ifadesini ekleyerek ad ve soyad arasında bir boşluk olmasını sağladım. **AS full_name** ifadesiyle de oluşturduğum yeni sütuna anlamlı bir isim verdim.

Bu ödevde, farklı SQL sorgularını kullanarak belirlenen kriterlere uygun veri çekme işlemleri gerçekleştirildi. SQL'in temel sorgulama yapıları olan SELECT, JOIN, WHERE, DISTINCT ve ORDER BY gibi ifadeler pratik edildi.