

TURK STUDENT COMMUNITY

PostgreSQL ile Çevrimiçi Eğitim Platformu Veritabanı Tasarımı Projesi

Bu projede, PostgreSQL veritabanı yönetim sistemini kullanarak bir "Çevrimiçi Eğitim Platformu" için veri tabanı kuracaksınız. Bu platformda üyeler kayıt olabilecek, eğitimlere katılabilecek, bu eğitimlerden sertifika kazanabilecek ve kullanıcılar blog tarafında paylaşım yaparak kendilerine ait profil sayfalarında seviye atlayabileceklerdir. Proje, veritabanı tasarımı prensiplerini ve PostgreSQL kullanımını pratik bir senaryo üzerinden deneyimlemenizi amaçlamaktadır.

1. Veritabanı Şeması Tasarımı

Aşağıdaki temel işlevleri destekleyecek tablolar tasarlayın:

- Üyeler (Members):** Üye bilgilerini (kullanıcı adı (VARCHAR(50) UK), e-posta (VARCHAR(100) UK), şifre (VARCHAR(255)), kayıt tarihi (TIMESTAMP), ad (VARCHAR(50)), soyad (VARCHAR(50)) vb.) saklayacak bir tablo oluşturun. Bu tabloda bir Birincil Anahtar (PK) (INTEGER veya BIGINT) tanımlayın.
- Eğitimler (Courses):** Platformdaki eğitimlerin bilgilerini (adı (VARCHAR(200)), açıklaması (TEXT), başlangıç tarihi (DATE), bitiş tarihi (DATE), eğitmen bilgisi (VARCHAR(100)) vb.) saklayacak bir tablo oluşturun. Bu tabloda bir Birincil Anahtar (PK) (INTEGER veya BIGINT) tanımlayın.
- Kategoriler (Categories):** Eğitim kategorilerini (yapayzeka (VARCHAR(100)), blokzincir (VARCHAR(100)), siber güvenlik (VARCHAR(100)) vb.) yönetebilecek bir tablo oluşturun. Bu tabloda bir Birincil Anahtar (PK) (INTEGER veya SMALLINT) tanımlayın. Eğitimler ile kategoriler arasında bir ilişki kurun (Yabancı Anahtar (FK) (INTEGER veya SMALLINT) kullanılarak).
- Katılımlar (Enrollments):** Kullanıcıların hangi eğitimlere katıldığını takip edebilecek bir ara tablo oluşturun. Bu tablo, kullanıcılar (INTEGER veya BIGINT FK) ve eğitimler (INTEGER veya BIGINT FK) tabloları arasındaki çok-çok ilişkisini yönetmeli ve her iki tabloya ait Yabancı Anahtarları (FK) içermelidir. Bu tabloda bir Birincil Anahtar (PK) (INTEGER veya BIGINT) da tanımlanabilir (örneğin, otomatik artan bir ID). İsteğe bağlı olarak katılım tarihi (TIMESTAMP) gibi bilgiler de eklenebilir.
- Sertifikalar (Certificates):** Tamamlanan eğitimler için sertifika bilgilerini (sertifika kodu (VARCHAR(100) UK), veriliş tarihi (DATE) vb.) saklayacak bir tablo oluşturun. Bu tabloda bir Birincil Anahtar (PK) (INTEGER veya BIGINT) tanımlayın.
- Sertifika Atamaları (CertificateAssignments):** Hangi kullanıcının (INTEGER veya BIGINT FK) hangi sertifika (INTEGER veya BIGINT FK) aldığını ilişkilendirecek bir tablo oluşturun (Yabancı Anahtarlar (FK) kullanarak üyeler ve sertifikalar tablolarına bağlanılmalıdır). Bu tabloda bir Birincil Anahtar (PK) (INTEGER veya BIGINT) tanımlayın ve isteğe bağlı olarak alım tarihi (DATE) gibi bilgiler ekleyebilirsiniz.
- Blog Gönderileri (BlogPosts):** Kullanıcıların blog gönderilerini (başlık (VARCHAR(255)), içerik (TEXT), yayın tarihi (TIMESTAMP), yazar bilgisi (INTEGER veya BIGINT FK) vb.) saklayacak bir tablo oluşturun. Bu tablo, üyeler tablosu ile ilişkilendirilmelidir (Yabancı Anahtar (FK) kullanılarak) ve bir Birincil Anahtar (PK) (INTEGER veya BIGINT) içermelidir.

2. Birincil ve Yabancı Anahtarlar:

- I. Her tabloda bir **Birincil Anahtar (PK)** tanımlayın. Uygun veri tipini seçin (**INTEGER, BIGINT, SMALLINT** vb.).
- II. Tablolar arasındaki ilişkileri doğru bir şekilde kurmak için **Yabancı Anahtarları (FK)** kullanın. Yabancı anahtar sütunlarının veri tiplerinin, ilişkili oldukları birincil anahtar sütunlarının veri tipleriyle aynı olmasına dikkat edin.
- III. Gerekli alanlarda veri tekliğini sağlamak için **Tekil Anahtarlar (UK)** tanımlayın (örneğin, kullanıcı adı (**VARCHAR**), e-posta (**VARCHAR**), sertifika kodu (**VARCHAR**)). Uygun uzunlukları belirtin.

3. SQL Dosyası Oluşturma

- I. Oluşturduğunuz tüm tabloları, sütun adlarını, uygun **veri tiplerini (INTEGER, BIGINT, VARCHAR, TEXT, DATE, TIMESTAMP, FLOAT, BOOLEAN vb.)**, **Birincil Anahtar (PK)**, **Yabancı Anahtar (FK)** ve **Tekil Anahtar (UK)** kısıtlamalarını tanımlayan bir .sql dosyası oluşturun. Gerekli durumlarda NOT NULL gibi kısıtlamaları da ekleyin.
- II. Bu .sql dosyası, veritabanının sıfırdan oluşturulabilmesini ve ilişkilerin kurulabilmesini sağlamalıdır.

4. Genel Şema Ekran Resmi

- I. Oluşturduğunuz tabloları ve aralarındaki **Birincil Anahtar (PK)** ve **Yabancı Anahtar (FK)** ilişkilerini (oklarla gösterilmiş şekilde) görsel olarak gösteren bir veritabanı şema diyagramının ekran görüntüsünü alın. Bu, veritabanı tasarımının anlaşılmasına yardımcı olacaktır. Şemada sütun adlarını ve veri tiplerini de göstermeye çalışın.

5. GitHub Deposu Oluşturma

- I. Projenizi tamamladıktan sonra, oluşturduğunuz .sql dosyasını ve genel şema ekran resmini içeren bir GitHub deposu oluşturun.
- II. Projenizi değerlendirebilmemiz için reponuzu **public** hale getirin.
- III. Projenizi tamamladıktan sonra, .sql dosyanızı ve genel şema ekran resminizi içeren GitHub repo bağlantınızı **[25.04.2025]** tarihine kadar saat 23:59'a kadar aşağıda yer alan formdaki gerekli bilgileri doldurarak gönderin.

6. Değerlendirme Kriterleri:

- I. Veritabanı tasarımının doğruluğu ve tutarlılığı.
- II. İhtiyaçlara uygun tabloların ve ilişkilerin modellenmesi.
- III. **Birincil Anahtar (PK)**, **Yabancı Anahtar (FK)** ve **Tekil Anahtar (UK)** kısıtlamalarının doğru ve etkin kullanımı.
- IV. Sütunlar için uygun **veri tiplerinin** seçilmesi.
- V. Oluşturulan .sql dosyasının hatasız çalışabilir olması.
- VI. Veritabanı şemasının anlaşılır ve doğru bir şekilde görselleştirilmesi.
- VII. GitHub kullanımının doğru olması (repo oluşturma, dosya yükleme, public yapma).

Proje Teslim Formu: <https://forms.gle/PSSWr4LD3ya7Z1W88>

Başarılar dileriz!