

## Giriş

Bu belge, projemiz için oluşturduğumuz nihai veritabanı mimarisi ve Row Level Security (RLS) politikalarının tam bir özetini sunmaktadır. Her bir kural, projenin özel iş mantığına ve kullanıcı rollerine göre dikkatlice tasarlanmıştır.

### Bölüm 1: Merkezi ve Temel Tablolar

#### 1.1. user\_profiles (Kullanıcı Profilleri)

Bu tablo, tüm kullanıcıların rollerini ve temel bilgilerini barındıran sistemin kalbidir.

- **Politika 1: Rol bazlı profil görüntüleme kuralları**

- **Komut:** SELECT
- **Hedef Rol:** authenticated
- **Mantık:** Bir profilin görülmesine aşağıdaki koşullardan **biri** doğruysa izin verilir:
  1. Kullanıcı kendi profilini görüntülüyorsa.
  2. Görüntüleyen kullanıcının rolü admin veya superadmin ise.
  3. Görüntülenen profilin rolü teacher ise (herkesin öğretmenleri listeleyebilmesi için).
  4. Görüntüleyen bir teacher ise ve baktığı profil, henüz bir öğretmene atanmamış bir student ise.
- **SQL Kodu:**

```
(
  (auth.uid() = user_id) OR
  ((SELECT role FROM public.user_profiles WHERE user_id = auth.uid()) IN
  ('admin'::user_role, 'superadmin'::user_role)) OR
  (role = 'teacher'::user_role) OR
  (
    ((SELECT role FROM public.user_profiles WHERE user_id = auth.uid()) =
    'teacher'::user_role) AND
    (role = 'student'::user_role) AND
    (NOT (EXISTS (SELECT 1 FROM public.teacher_student_links WHERE
    (teacher_student_links.student_id = user_profiles.user_id))))
  )
)
```

- **Politika 2: Kullanıcılar kendi profilini güncelleyebilir**

- **Komut:** UPDATE
- **Hedef Rol:** authenticated
- **Mantık:** Bir kullanıcı sadece kendi user\_id'sine sahip olan satırı güncelleyebilir.
- **SQL Kodu:**

(auth.uid() = user\_id)

## Bölüm 2: İçerik ve Atama Yönetimi

### 2.1. content (Quizler ve Ders İçerikleri)

Bu tablo, öğretmenler ve adminler tarafından oluşturulan tüm eğitim materyallerini içerir.

- **Politika 1: Öğretmenler ve Adminler quiz oluşturabilir**
  - **Komut:** INSERT
  - **Hedef Rol:** authenticated
  - **Mantık:** Sadece rolü teacher veya superadmin olan kullanıcılar yeni içerik ekleyebilir.
  - **SQL Kodu:**  
((SELECT role FROM public.user\_profiles WHERE (user\_profiles.user\_id = auth.uid())) IN ('teacher'::user\_role, 'superadmin'::user\_role))
- **Politika 2: Quiz'i sadece oluşturanlar veya Adminler güncelleyebilir**
  - **Komut:** UPDATE
  - **Hedef Rol:** authenticated
  - **Mantık:** Bir içeriği, sadece onu oluşturan kişi (creator\_id) veya rolü superadmin olan bir kullanıcı güncelleyebilir.
  - **SQL Kodu:**  
(  
    (creator\_id = auth.uid()) OR  
    ((SELECT role FROM public.user\_profiles WHERE (user\_profiles.user\_id = auth.uid())) = 'superadmin'::user\_role)  
)
- **Politika 3: Kullanıcılar ilgili quizleri görebilir**
  - **Komut:** SELECT
  - **Hedef Rol:** authenticated
  - **Mantık:** Bir içeriği, onu oluşturan kişi, bir superadmin veya o içeriğin kendisine (quiz\_assignments üzerinden) atandığı bir öğrenci görebilir.
  - **SQL Kodu:**  
(  
    (creator\_id = auth.uid()) OR  
    ((SELECT role FROM public.user\_profiles WHERE (user\_profiles.user\_id = auth.uid())) = 'superadmin'::user\_role) OR

```

(EXISTS (
  SELECT 1 FROM public.quiz_assignments
  WHERE (
    (quiz_assignments.quiz_id = content.id) AND
    (
      (quiz_assignments.assigned_to_user_id = auth.uid()) OR
      (quiz_assignments.assigned_to_grade = (SELECT grade FROM
public.user_profiles WHERE (user_profiles.user_id = auth.uid()))
    )
  )
))
)

```

## 2.2. quiz\_assignments (Quiz Atamaları)

Bu tablo, hangi quizin kime atandığını yönetir.

- **Politika 1: Öğretmenler ve Adminler quiz atayabilir**
  - **Komut:** INSERT
  - **Hedef Rol:** authenticated
  - **Mantık:** Sadece rolü teacher, admin veya superadmin olan kullanıcılar yeni atama yapabilir.
  - **SQL Kodu:**

```

((SELECT role FROM public.user_profiles WHERE (user_profiles.user_id =
auth.uid())) IN ('teacher'::user_role, 'admin'::user_role,
'superadmin'::user_role))

```
- **Politika 2: İlgili taraflar ve Adminler atamaları görebilir**
  - **Komut:** SELECT
  - **Hedef Rol:** authenticated
  - **Mantık:** Bir atamayı, atamayı yapan kişi, atamanın yapıldığı kişi veya rolü admin/superadmin olan bir kullanıcı görebilir.
  - **SQL Kodu:**

```

(
  (assigned_to_user_id = auth.uid()) OR
  (assigned_by = auth.uid()) OR
  ((SELECT role FROM public.user_profiles WHERE (user_profiles.user_id =
auth.uid())) IN ('admin'::user_role, 'superadmin'::user_role))
)

```

## Bölüm 3: Kullanıcı İlişkileri ve Başvurular

### 3.1. parent\_student\_links (Veli-Öğrenci Bağlantıları)

- **Politika 1: Öğrenciler veli daveti oluşturabilir**

- **Komut:** INSERT
- **Hedef Rol:** authenticated
- **Mantık:** Sadece rolü student olan bir kullanıcı, kendi adına (student\_id'si kendi user\_id'si ise) bir davet oluşturabilir.
- **SQL Kodu:**  
(  
((SELECT role FROM public.user\_profiles WHERE (user\_profiles.user\_id =  
auth.uid())) = 'student'::user\_role) AND  
(auth.uid() = student\_id)  
)

- **Politika 2: İlgili taraflar ve Adminler bağlantıyı görebilir**

- **Komut:** SELECT
- **Hedef Rol:** authenticated
- **Mantık:** Bir bağlantıyı, o bağlantıdaki veli, öğrenci veya rolü admin/superadmin olan bir kullanıcı görebilir.
- **SQL Kodu:**  
(  
(auth.uid() = parent\_id) OR  
(auth.uid() = student\_id) OR  
((SELECT role FROM public.user\_profiles WHERE (user\_profiles.user\_id =  
auth.uid())) IN ('admin'::user\_role, 'superadmin'::user\_role))  
)

### 3.2. teacher\_student\_links (Öğretmen-Öğrenci Bağlantıları)

- **Politika 1: Öğrenciler öğretmene başvurabilir**

- **Komut:** INSERT
- **Hedef Rol:** authenticated
- **Mantık:** Sadece rolü student olan bir kullanıcı, kendi adına bir başvuru oluşturabilir.
- **SQL Kodu:**  
(  
((SELECT role FROM public.user\_profiles WHERE (user\_profiles.user\_id =

```
auth.uid())) = 'student'::user_role) AND  
(auth.uid() = student_id)  
)
```

- **Politika 2: İlgili taraflar ve Adminler bağlantıyı görebilir**

- **Komut:** SELECT
- **Hedef Rol:** authenticated
- **Mantık:** Bir bağlantıyı, o bağlantıdaki öğretmen, öğrenci veya rolü admin/superadmin olan bir kullanıcı görebilir.
- **SQL Kodu:**  
(  
 (auth.uid() = teacher\_id) OR  
 (auth.uid() = student\_id) OR  
 ((SELECT role FROM public.user\_profiles WHERE (user\_profiles.user\_id =  
auth.uid())) IN ('admin'::user\_role, 'superadmin'::user\_role))  
)

- **Politika 3: Öğretmenler başvuru durumunu güncelleyebilir**

- **Komut:** UPDATE
- **Hedef Rol:** authenticated
- **Mantık:** Sadece rolü teacher olan bir kullanıcı, kendi taraf olduğu bir bağlantının durumunu güncelleyebilir.
- **SQL Kodu:**  
(  
 ((SELECT role FROM public.user\_profiles WHERE (user\_profiles.user\_id =  
auth.uid())) = 'teacher'::user\_role) AND  
 (auth.uid() = teacher\_id)  
)

### 3.3. teacher\_applications (Öğretmen Başvuruları)

- **Politika 1: Herkes öğretmenlik başvurusu yapabilir**

- **Komut:** INSERT
- **Hedef Rol:** public
- **Mantık:** Herkes (giriş yapmamışlar dahil) başvuru yapabilir.
- **SQL Kodu:** true

- **Politika 2: Adminler öğretmen başvurularını görebilir**

- **Komut:** SELECT
- **Hedef Rol:** authenticated

- **Mantık:** Sadece rolü admin veya superadmin olanlar başvuruları görebilir.
- **SQL Kodu:**  
`((SELECT role FROM public.user_profiles WHERE (user_profiles.user_id = auth.uid())) IN ('superadmin'::user_role, 'admin'::user_role))`
- **Politika 3: Adminler öğretmen başvurularını güncelleyebilir**
  - **Komut:** UPDATE
  - **Hedef Rol:** authenticated
  - **Mantık:** Sadece rolü admin veya superadmin olanlar başvuruları güncelleyebilir.
  - **SQL Kodu:**  
`((SELECT role FROM public.user_profiles WHERE (user_profiles.user_id = auth.uid())) IN ('superadmin'::user_role, 'admin'::user_role))`

#### Bölüm 4: Kullanıcıya Özel Veri Tabloları

Bu bölümdeki tüm tablolar (quiz\_attempts, user\_stats, user\_streaks, analytics\_events, point\_history, user\_badges) aynı temel güvenlik mantığını paylaşır.

##### 4.1. INSERT Politikaları

- **quiz\_attempts için:**
  - **Mantık:** Sadece rolü student olan kullanıcılar kendi adlarına yeni deneme ekleyebilir.
  - **SQL Kodu:**  
`(  
 ((SELECT role FROM public.user_profiles WHERE (user_profiles.user_id = auth.uid())) = 'student'::user_role) AND  
 (auth.uid() = student_id)  
 )`
- **Diğer tüm tablolar için (user\_stats, point\_history vb.):**
  - **Mantık:** Kullanıcıların doğrudan veri eklemesi engellenmiştir. Bu işlemler sadece Edge Function gibi güvenli bir arka plan süreciyle yapılabilir.
  - **SQL Kodu:** false

##### 4.2. SELECT Politikaları (Tüm 6 Tablo İçin Ortak Şablon)

- **Mantık:** Bir kaydı, o kaydın sahibi olan kullanıcı, o kullanıcının velisi, öğretmeni veya admin/superadmin rolündeki bir kullanıcı görebilir.
- **SQL Şablonu:** [TABLO\_ADı] kısmını ilgili tablonun adıyla değiştirerek kullanılır.

```
(
  (auth.uid() = user_id) OR
  ((SELECT role FROM public.user_profiles WHERE (user_profiles.user_id =
auth.uid())) IN ('admin'::user_role, 'superadmin'::user_role)) OR
  (EXISTS (
    SELECT 1 FROM public.parent_student_links
    WHERE (parent_student_links.parent_id = auth.uid()) AND
    (parent_student_links.student_id = [TABLO_ADI].user_id)
  )) OR
  (EXISTS (
    SELECT 1 FROM public.teacher_student_links
    WHERE (teacher_student_links.teacher_id = auth.uid()) AND
    (teacher_student_links.student_id = [TABLO_ADI].user_id)
  ))
)
Not: quiz_attempts tablosu için [TABLO_ADI].user_id yerine
[TABLO_ADI].student_id kullanılır.
```

## Bölüm 5: Genel Tablolar

### 5.1. daily\_leaderboard (Liderlik Tablosu)

- **Politika 1: Herkes liderlik tablosunu görebilir**
  - **Komut:** SELECT
  - **Hedef Rol:** public
  - **Mantık:** Herkes (giriş yapmamışlar dahil) liderlik tablosunu görebilir.
  - **SQL Kodu:** true
- **Politika 2: Client-side eklemeler engellendi**
  - **Komut:** INSERT
  - **Hedef Rol:** authenticated
  - **Mantık:** Kullanıcıların doğrudan liderlik tablosuna veri eklemesi engellenmiştir.
  - **SQL Kodu:** false