**I. Üyeler (Members):**

CREATE TABLE members (

member\_id SERIAL PRIMARY KEY, -- Birincil Anahtar (otomatik artan)

username VARCHAR(50) UNIQUE NOT NULL, -- Kullanıcı adı (tekil ve boş olamaz)

email VARCHAR(100) UNIQUE NOT NULL, -- E-posta (tekil ve boş olamaz)

password VARCHAR(255) NOT NULL, -- Şifre (güvenlik için uzun tutuldu)

registration\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP, -- Kayıt tarihi (otomatik olarak şimdi)

first\_name VARCHAR(50) NOT NULL, -- Ad

last\_name VARCHAR(50) NOT NULL -- Soyad

);

* SERIAL → PostgreSQL'de otomatik artan tamsayı demektir. member\_id için ideal.
* UNIQUE → Aynı kullanıcı adı veya e-posta tekrar girilemez.
* VARCHAR(255) → Şifreler için uzun tutulur.
* DEFAULT CURRENT\_TIMESTAMP→ Kayıt tarihi otomatik olarak güncel zamanı alır.

**II. Eğitimler (Courses):**

CREATE TABLE courses (

course\_id SERIAL PRIMARY KEY, -- Birincil anahtar (otomatik artan)

course\_name VARCHAR(200) NOT NULL, -- Eğitim adı

description TEXT, -- Eğitim açıklaması

start\_date DATE NOT NULL, -- Başlangıç tarihi

end\_date DATE, -- Bitiş tarihi

instructor\_name VARCHAR(100) NOT NULL -- Eğitmen adı ve soyadı

);

* SERIAL → course\_id için otomatik artar.
* TEXT → Eğitim açıklamaları uzun olabileceği için TEXT kullanılır.
* DATE → Tarihler sadece gün/ay/yıl bilgisi taşıyacağı için DATE türünde tutulur.
* instructor\_name → Eğer ilerde eğitmenler için kullanılır.

**III. Kategoriler (Categories):**

CREATE TABLE categories (

category\_id SMALLINT PRIMARY KEY, -- Birincil anahtar

category\_name VARCHAR(100) NOT NULL -- Kategori adı

);

* SMALLINT performans açısından iyidir.

ALTER TABLE courses

ADD COLUMN category\_id SMALLINT,

ADD CONSTRAINT fk\_category

FOREIGN KEY (category\_id)

REFERENCES categories(category\_id)

ON DELETE SET NULL;

INSERT INTO categories (category\_id, category\_name) VALUES

(1, 'Yapay Zeka'),

(2, 'Blokzincir'),

(3, 'Siber Güvenlik');

* courses tablosuna category\_id eklenir.
* Bu sütun, categories tablosundaki category\_id alanına bağlanır.
* Bir kategori silinirse, o kategoriye bağlı eğitimlerin category\_id değeri NULL olur (ilişki kopar ama eğitim kalır).

**IV. Katılımlar (Enrollments):**

CREATE TABLE enrollments (

enrollment\_id BIGSERIAL PRIMARY KEY,

member\_id BIGINT NOT NULL,

course\_id BIGINT NOT NULL,

enrollment\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

CONSTRAINT fk\_member

FOREIGN KEY (member\_id)

REFERENCES members(member\_id)

ON DELETE CASCADE,

CONSTRAINT fk\_course

FOREIGN KEY (course\_id)

REFERENCES courses(course\_id)

ON DELETE CASCADE

);

* BIGSERIAL: Otomatik artan ID üretir.
* FOREIGN KEY (member\_id) → members tablosuna bağlı.
* FOREIGN KEY (course\_id) → courses tablosuna bağlı.
* ON DELETE CASCADE: Kullanıcı veya eğitim silinirse, bu tablo da ona bağlı verileri siler.
* DEFAULT CURRENT\_TIMESTAMP: Tarih otomatik atanır.

**V. Sertifikalar (Certificates):**

CREATE TABLE certificates (

certificate\_id BIGSERIAL PRIMARY KEY, -- Otomatik artan PRİMARY KEY

member\_id BIGINT NOT NULL, -- Sertifikayı alan kişi

course\_id BIGINT NOT NULL, -- Tamamlanan kurs

certificate\_code VARCHAR(100) UNIQUE NOT NULL, -- Benzersiz sertifika kodu

issue\_date DATE DEFAULT CURRENT\_DATE, -- Veriliş tarihi

CONSTRAINT fk\_cert\_member

FOREIGN KEY (member\_id)

REFERENCES members(member\_id)

ON DELETE CASCADE,

CONSTRAINT fk\_cert\_course

FOREIGN KEY (course\_id)

REFERENCES courses(course\_id)

ON DELETE CASCADE

);

* BIGSERIAL: Otomatik artan ID.
* UNIQUE: Sertifika kodları benzersiz olmalı.
* DEFAULT CURRENT\_DATE: Sertifika tarihi otomatik olarak o günü alır.
* FOREIGN KEY: members ve courses ile ilişkili hale getirir.

**VI. Sertifika Atamaları (CertificateAssignments):**

CREATE TABLE certificate\_assignments (

assignment\_id BIGSERIAL PRIMARY KEY, -- Otomatik artan birincil anahtar

member\_id BIGINT NOT NULL, -- Üye FOREIGN KEY

certificate\_id BIGINT NOT NULL, -- Sertifika FOREIGN KEY

assignment\_date DATE DEFAULT CURRENT\_DATE, -- Alım tarihi

CONSTRAINT fk\_ca\_member

FOREIGN KEY (member\_id)

REFERENCES members(member\_id)

ON DELETE CASCADE,

CONSTRAINT fk\_ca\_certificate

FOREIGN KEY (certificate\_id)

REFERENCES certificates(certificate\_id)

ON DELETE CASCADE

);

* BIGSERIAL: Otomatik artan ID.
* FOREIGN KEY:

1. member\_id: members tablosuna bağlı.
2. certificate\_id: certificates tablosuna bağlı.

* ON DELETE CASCADE: Bağlı kayıt silinirse buradaki kayıt da silinir.
* assignment\_date: Sertifikanın verildiği tarih, boş geçilirse otomatik olarak o gün atanır.

**VII. Blog Gönderileri (BlogPosts):**

CREATE TABLE blog\_posts (

post\_id BIGSERIAL PRIMARY KEY, -- Otomatik artan birincil anahtar

title VARCHAR(255) NOT NULL, -- Blog başlığı

content TEXT NOT NULL, -- Blog içeriği

publish\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP, -- Yayın tarihi

author\_id BIGINT NOT NULL, -- Yazarı (members tablosundan)

CONSTRAINT fk\_post\_author

FOREIGN KEY (author\_id)

REFERENCES members(member\_id)

ON DELETE CASCADE

);

* BIGSERIAL: Her blog gönderisine otomatik ID verir.
* author\_id: Blog gönderisini yazan kullanıcıyı temsil eder ve members tablosuna bağlıdır.
* DEFAULT CURRENT\_TIMESTAMP: Yayın tarihi boş bırakılırsa otomatik şu anki tarih atanır.
* ON DELETE CASCADE: Üye silinirse blog gönderileri de silinir.

**1. Birincil Anahtarlar (PK)**

Her tabloda **benzersiz** kayıtları tanımlamak için **PK** (Primary Key) kullandık:

| **Tablo** | **Birincil Anahtar** | **Veri Tipi** |
| --- | --- | --- |
| members | member\_id | BIGSERIAL |
| courses | course\_id | BIGSERIAL |
| categories | category\_id | SMALLSERIAL |
| enrollments | enrollment\_id | BIGSERIAL |
| certificates | certificate\_id | BIGSERIAL |
| certificate\_assignments | assignment\_id | BIGSERIAL |
| blog\_posts | post\_id | BIGSERIAL |

**2. Yabancı Anahtarlar (FK)**

İlişkisel veri yapısını kurmak için aşağıdaki FK’leri tanımladık:

| **Tabloda Yer Alan FK** | **Hangi Tablodaki PK'ye Bağlı** | **FK Veri Tipi** |
| --- | --- | --- |
| course\_id (courses) | category\_id (categories) | SMALLINT |
| member\_id (enrollments) | member\_id (members) | BIGINT |
| course\_id (enrollments) | course\_id (courses) | BIGINT |
| member\_id (certificates) | member\_id (members) | BIGINT |
| course\_id (certificates) | course\_id (courses) | BIGINT |
| member\_id (certificate\_assignments) | member\_id (members) | BIGINT |
| certificate\_id (certificate\_assignments) | certificate\_id (certificates) | BIGINT |
| author\_id (blog\_posts) | member\_id (members) | BIGINT |

**3. Tekil Anahtarlar (UK)**

Bazı verilerin tekrar etmemesi gerekir. Bunun için **UNIQUE** kısıtlamaları kullandık:

| **Tabloda Yer Alan Alan** | **UNIQUE (UK) Açıklama** |
| --- | --- |
| username (members) | Kullanıcı adları tektir |
| email (members) | E-posta adresleri benzersiz |
| certificate\_code (certificates) | Sertifika kodları benzersiz |