

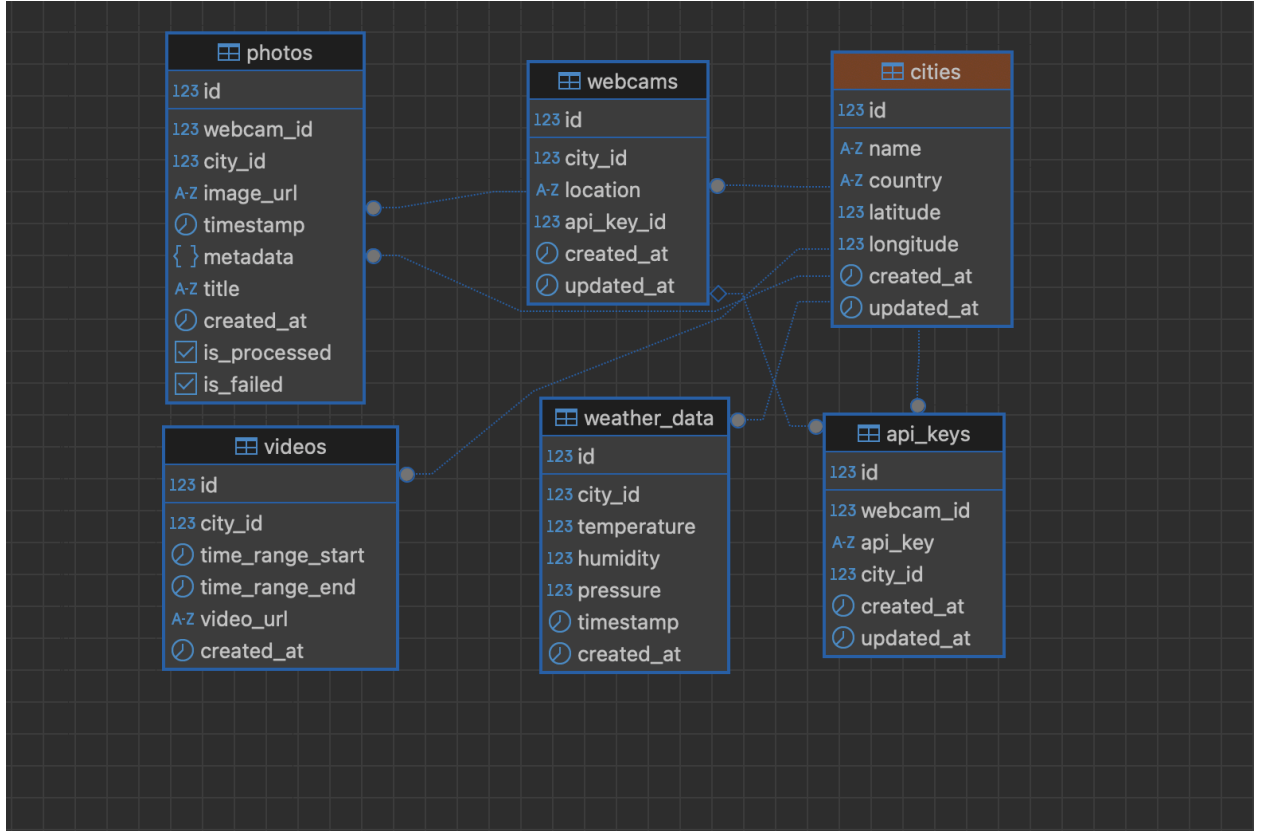
## CourseProject Döküman

### S3 Bucket Servisi:

S3 üzerinde course-project adında bir bucket oluşturulmuştur.

### RDS Servisi:

Rds Üzerinde **courseproject** adında bir tabanı oluşturulmuştur.



Tablo isimlerinden de görüldüğü üzere fotoğraflar, hava durumu verileri, api keyler ve işaret ettiği webcam lar, fotoğraflardan üretilen videolar ve şehirler burada tutulmaktadır.

### CloudWatch Event Rules ile Schedule İşlemi:

CloudWatch event rules üzerinde 60 dk da dir çalışacak schedule oluşturuldu ve video-almbda lambda servisine bağlandı.

## Api Gateway Servisler:

Lambdaya bağı 2 api gateway servisi bulunmaktadır. Bunlar aşağıdaki webcams-lambda ve api-lambda fonksiyonlarına bağı olup, rest api oluşturmaktadır.

## Lambda Servisleri:

4 Adet lambda fonksiyonu bulunmaktadır

### 1) webcams-lambda

Bu lambda veritabanında bulunan api key ler ile istek kabul etmektedir. (Api Key kontrolu yapılmaktadır.) Geçerli api keyi olan isteklerle, hava durumu verisi (weather\_data) ve fotoğraf (photo yükleyebilir.) Postman de WebCams klasöründe bu servisler yer almaktadır. Yüklenen fotoğraflar s3 e kayıt edilmektedir. Görsel S3 üzerinde photos klasörüne eklenir.

### 2) picture-lambda

Bu lambda webcams-lambda tarafından her fotoğraf eklendiğinde tetiklenmektedir. Yani s3 ile bağlanmıştır. Webcams-lambda üzerinden s3 e her görsel eklendiğinde, eklenen görsel için bu lambda tetiklenir ve eklenen görseli resize ederek, s3 bucket içinde [processed/](#) klasörüne eklenir. Ayrıca eklenen resmin veritabanındaki url i de işlenmiş görsel url olarak değiştirilir.

### 3) video-lambda

Bu lambda cloudwatch rules schedule ile 60 dakikadır bir tetiklenir. Son eklenen 5 resize edilmiş görseli processed klasöründen çeker, ffmpeg kütüphanesi kullanarak bu görsellerden 15 saniyelik bir video oluşturur ve sonrasında bunu bucket içinde videos kısmına kayıt eder. Ayrıca kayıt ettiği bu görselin url ini de sonradan api üzerinden çekebilmek için veri tabanına kayıt eder.

### 4) api-lambda

Bu lambda postman üzerinde Api klasoru içinde ki istekleri barındırır. Şehirleri listeleme (search yapabilecek şekilde), eklenen fotoğrafları tarih ve city\_id ye göre listeleme, videoları tarih ve city\_id ye göre listeleme, hava durumu verilerini tarih ve city\_id ye göre listeleme servisleri bulunur. Bu servisler ile veritabanından bilgi çeker. Ayrıca burdan çekilen bilgilerin tamamı upstash üzerinden redis cloud da 1 saat saklanmaktadır. Ayrı filtrelere (city\_id, start\_date, end\_date) sahip aynı route a gelen istekler, dökümandaki gibi tekrar tekrar çekilmek yerine, redis üzerine yazılıp 1 saat içinde redis den çekilmektedir. (Önbellekleme mekanizması kurulmuştur.)

Not: api-lambda ve webcams-lambda fonksiyonları lambda olarak oluşturulup sonrasında api gateway üzerinden http isteklerine dönüştürülmüştür. Bu lambda için örneğin webcams daki /weather gibi route lar eklenerek restful mantığı giydirilmiştir.

Tüm lambda kodları nodejs dilinde yazılmış olup, picture-lambda da sharp kütüphanesi ile görseli resize, video-lambda da ise ffmpeg kütüphanesi video birleştirme yapılmaktadır. Ffmpeg kütüphanesi boyutundan dolayı lambda layer olarak eklenmiştir ve sonraki süreçte tekrar tekrar kullanılabilir.

## Nodejs Proje Kurulumu

Ek de bir last.zip dosyası ilettim. İçinde 4 adet lambda function için ayrı ayrı klasörler mevcuttur. Her klasörün içinde bir lambda.zip dosyası bulunuyor. Aws üzerinde lambda function oluşturup sonrasında upload zip sekmesinden içeri alınabilir. Video işleme için kullanılan ffmpeg layer i de layer.zip içindedir. Oda layer olarak projeye eklenebilir.

Lambdalar env kullanılmıştır. Lambda configden env variables kısmına aşağıdakiler eklenmelidir:

```
BUCKET_NAME=courseproject-bucket  
DATABASE_URL="postgres://postgres:ONIP3B6y98@courseproject.c92qa2260f4e.us-east-1.rds.amazonaws.com:5432  
"
```

Bunlar db connection için geçerlidir. (hesabın geri alınmaması durumunda bunları atıyorum, eğer lab tekrar açılırsa sizin hesapta bunlar zaten kurulu. Ek olarak lambdaları import falan etmenize gerek yok eklenmişler içini inceleyebilirsiniz.)

Sadece bir upstash hesabına ihtiyacınız var. Free olarak oluşturup sonrasında api lambdasında,

```
const redis = new Redis("rediss://d
```

Şu kısımda connection bilgis olarak kendi oluşturduğunuz upstash redis bağlantısını eklemeniz gerekmektedir. Bunu direk api-lambda function içinden de ekleyebilirsiniz.

Postman dökümanını da ilettim ek olarak. (iki adet. Biri api için (get istekleri), diğeri webcam için. Webcam için api key authentication bulunuyor. )