Introduction

Congratulations! Our team voted that we should take your application to the next level, this is a very straightforward assignment, and you get to do it from the comfort of your home, perhaps with a nice cup of coffee.

About this assignment

We believe that this assignment should be done in 4 hours, however, we are not actually going to enforce a strict time limit, we understand that genius cannot be rushed 😊, however we need to be fair to other candidates. So, feel free to submit it by the indicated date as per the email. We will not grade based on the speed of submission but the quality of your work, mainly how cost-effective is your solution in terms of CPU, memory, bandwidth and other resources.

List of files with this assignment

1. This document you are reading now (Technical Assignment.PDF)
2. The file: `credentials.txt`, containing login details.

How to get started?

Okay, we have created a MySQL account for you, the login details are in the "`credentials.txt`" file. Please test your login and report any problems to this email: moe.almezaiyin@eightcap.com

Outcomes

1. A complete working solution that can be tested.
2. You will provide the source code of the solution.
3. You will provide documentation of the solution.

Assignment

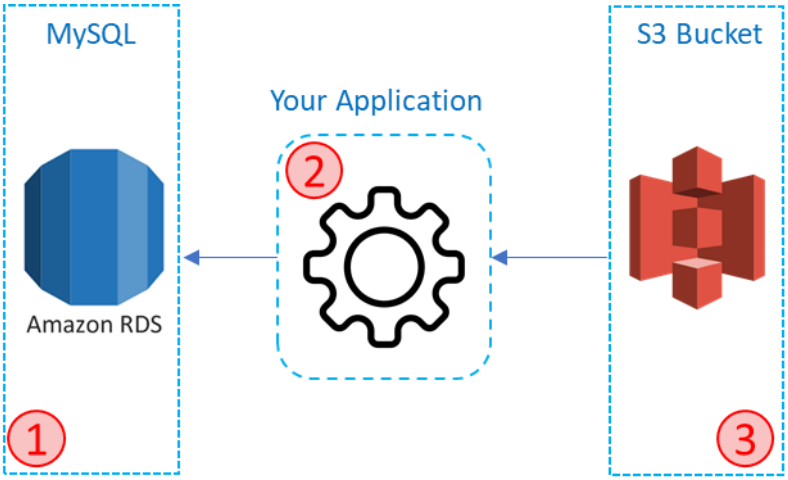
We are implementing a batch processing system that reads data from an S3 bucket, and populates a database with the results. We will not require you to code in any specific language, we accept Python, C#, Java, GO, or whatever. Make sure that the language you picked does indeed have an AWS SDK.

You can check the availability of the AWS SDKs from here:

https://aws.amazon.com/getting-started/tools-sdks/

This application can be a console application with no UI, it should be possible to start it from the command line, if you like, you can design a UI for the application, but it is an optional step.

Solution Design



Component	Configuration	Your work
1: Database	RDS MySQL v.5.7.x	Create the database as per the attached data model in this document, see below for the database design.
2: Application	Up to you, a desktop console application with no UI is the minimum accepted, if you like to build a nice UI/Web Interface it is your choice.	This application can be written with any language that has an AWS SDK. The application function is explained below.
3: S3 Bucket	An S3 Bucket containing a group of pictures	You will read data from this S3 bucket.

Database

Using the MySQL credentials, login and create tables to match this data model, each table is explained in detail below, also it is up to you to choose the correct indices that will make the queries run faster.

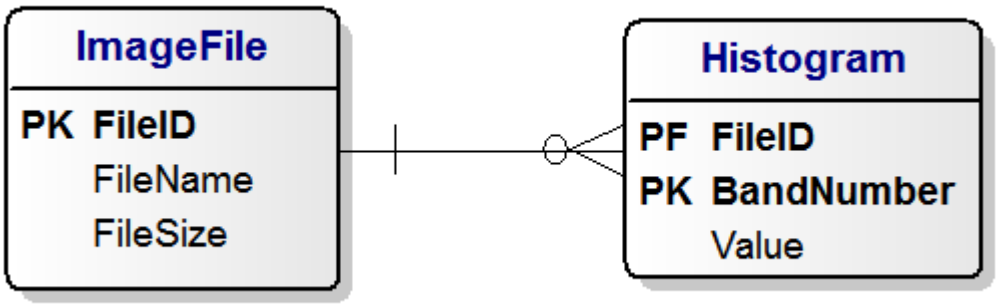


Table ImageFile

Contains the names of the JPEG files found in the S3 bucket (FileName), the byte size of each file (FileSize), the primary key is autoincement (FileId).

Table Histogram

The table fields: FileID (Primary/Foreign key of parent image file from the table ImageFile), BandNumber is between 0-255, the color index, Value is an integer representing the count of the frequency of that color.

This table contains a histogram of the image, the histogram is 256 bytes per any image, it is the count of the RGB colors, however the images are 24-bits, so you need to map from 24-bits (16777216 colors) to 8-bits (256 colors).

The easiest way is to use this reduction formula:
24-bits encode the RGB as 8-bits per channel (RRRRRRRRR GGGGGGGG BBBBBBBBB), we would map it to 8-bits (RRR GGG BB), the reduction formula is:

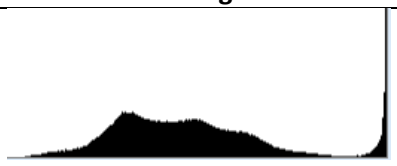
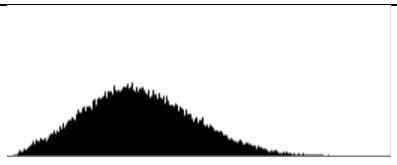
```
8BitColorByte = (Math.floor((red / 32)) << 5) + (Math.floor((green / 32)) << 2) + Math.floor((blue / 64));
```

To compute the histogram for the image, implement this algrothim:

1. For each image file in the directory
    a. Initialize the HistogramArray[256] to all zeros.
    b. For each pixel in the image do the following:
        i. Convert each pixel RGB to 8BitColorByte as per the reduction formula above.
        ii. HistogramArray[8BitColorByte]++
    c. Save the HistogramArray to the table Histogram for that image.

The histogram is the count of each color that appears in the image, you can read more about a histogram here: https://en.wikipedia.org/wiki/Image_histogram

Example of histograms

Image	Histogram
	
	

Application

The application can be a command line tool, it is optional to have a UI, the application will function as follows:

1. Connect to the S3 bucket. Using the AWS SDK library.
2. Connect to the MySQL database, delete all data from Pixels, and ImageFile tables.
3. Scan the S3 bucket for *.JPG files in the root directory.
4. For every JPG file found do the following:
    a. Read the size, name

   b. Insert a row in ImageFile.
   c. Download the file from S3
   d. Open it, read the pixels of the image using any popular image processing SDK or library.
   e. Compute the histogram as mentioned above.
   f. Insert the histogram data in the table Histogram.
   g. Print progress messages on the console when a file is finished
   h. Process next file, or finish if no more files are left.
5. Print a message at the end when done, showing how many files processed.

S3 Bucket

This is a read-only S3 bucket with a list of images, you can test access to this using a free tool like S3 Browser, you can download from here: http://s3browser.com/

Happy coding!

Image