# **Healthcare Analytics Engine**

This project is a sample Healthcare Analytics Engine designed to process and analyze patient records. It uses Python with an Object-Oriented approach for data processing, a MySQL database for data storage, and is designed to be deployed on an AWS EC2 instance.

## **Table of Contents**

1. [Project Overview](#rshrbob10vyd)
2. [Architectural Diagram](#9mrtxmbo03o6)
3. [Setup Instructions](#y2gg6mo4cwnl)
   * [AWS EC2 Setup](#igcnymihz6dy)
   * [MySQL Server Setup](#tcrk2c9w28ew)
   * [Python Environment Setup](#cap3v2o5zmyx)
4. [Database Schema](#29ngmfwi5sdu)
5. [Running the Application](#oi1cxakpf9p4)
6. [Project Structure](#kzgcc4b4kru)

## **Project Overview**

This engine is designed to:

* **Manage Patient Records**: Store and manage a large volume of patient records in a normalized relational database.
* **Efficient Querying**: Utilize strategic indexing to ensure high-speed data retrieval.
* **Modular & Scalable Code**: Employ Object-Oriented Programming (OOP) principles in Python for modular, scalable, and maintainable code.

## **Architectural Diagram**

+-------------------+ +-------------------------+ +---------------------+  
| | | | | |  
| data\_importer.py |----->| main\_analysis.py |----->| MySQL Database |  
| (Reads CSV) | | (OOP Python Scripts) | | (on AWS EC2) |  
| | | | | |  
+-------------------+ +-------------------------+ +---------------------+  
 | ^  
 | | (Processes Data)  
 | (Loads Data) |  
 +----------------------------+

## **Setup Instructions**

### **1. AWS EC2 Setup**

1. **Launch an EC2 Instance**:
   * Go to the AWS Management Console and navigate to EC2.
   * Launch a new instance. An Amazon Linux 2 or Ubuntu Server AMI is recommended. A t2.micro instance is sufficient for this sample project.
   * Configure security groups to allow SSH (port 22) from your IP and MySQL/Aurora (port 3306) from within the security group.
2. **Connect to your Instance**:
   * Use SSH to connect to your newly created EC2 instance.  
     ssh -i "your-key.pem" ec2-user@your-ec2-public-dns

### **2. MySQL Server Setup**

1. **Install MySQL Server** on your EC2 instance:
   * For **Ubuntu**:  
     sudo apt update  
     sudo apt install mysql-server  
     sudo mysql\_secure\_installation
   * For **Amazon Linux 2**:  
     sudo yum update -y  
     sudo amazon-linux-extras install lamp-mariadb10.2-php7.2 -y  
     sudo systemctl start mariadb  
     sudo systemctl enable mariadb  
     sudo mysql\_secure\_installation
2. **Create Database and User**:
   * Log in to MySQL: sudo mysql -u root -p
   * Create the database and a dedicated user for the application.  
     CREATE DATABASE health\_analytics;  
     CREATE USER 'py\_user'@'localhost' IDENTIFIED BY 'password';  
     GRANT ALL PRIVILEGES ON health\_analytics.\* TO 'py\_user'@'localhost';  
     FLUSH PRIVILEGES;  
     EXIT;

### **3. Python Environment Setup**

1. **Install Python and Pip**:
   * Python 3 is usually pre-installed. You can check with python3 --version.
   * Install pip if it's not present:  
     sudo apt update  
     sudo apt install python3-pip # For Ubuntu  
       
     or  
     sudo yum install python3-pip # For Amazon Linux
2. **Create a Virtual Environment** (Recommended):  
   python3 -m venv venv  
   source venv/bin/activate
3. **Install Required Python Libraries**:
   * A requirements.txt file is provided. Use pip to install the dependencies.  
     pip install -r requirements.txt
   * The primary dependency is mysql-connector-python.

## **Database Schema**

The database health\_analytics contains the following tables:

* patients: Stores basic patient information.
* visits: Records each patient visit.
* diagnoses: Contains diagnosis information linked to a visit.

Indexes are created on patients(last\_name), visits(patient\_id), and diagnoses(visit\_id) to speed up common queries.

## **Running the Application**

1. **Clone the Project**:
   * Clone this repository onto your EC2 instance.
2. **Setup the Database**:
   * Execute the schema.sql script to create the tables and indexes.  
     mysql -u py\_user -p health\_analytics < schema.sql
3. **Load the Data**:
   * Run the data importer script to load the sample patient data from patients.csv.  
     python data\_importer.py
4. **Run the Main Analysis**:
   * Execute the main application script to perform data analysis.  
     python main\_analysis.py

## **Project Structure**

.  
├── main\_analysis.py # Main script for data analysis  
├── data\_importer.py # Script to import CSV data into MySQL  
├── db\_config.py # Database configuration  
├── healthcare/ # Python package for analytics logic  
│ ├── \_\_init\_\_.py  
│ ├── database.py # Handles all database interactions  
│ ├── patient.py # Patient data model  
│ └── analysis.py # Data processing and analysis logic  
├── schema.sql # SQL script for database schema  
├── patients.csv # Sample patient data  
└── requirements.txt # Python package requirements