Table of Contents

Introduction:

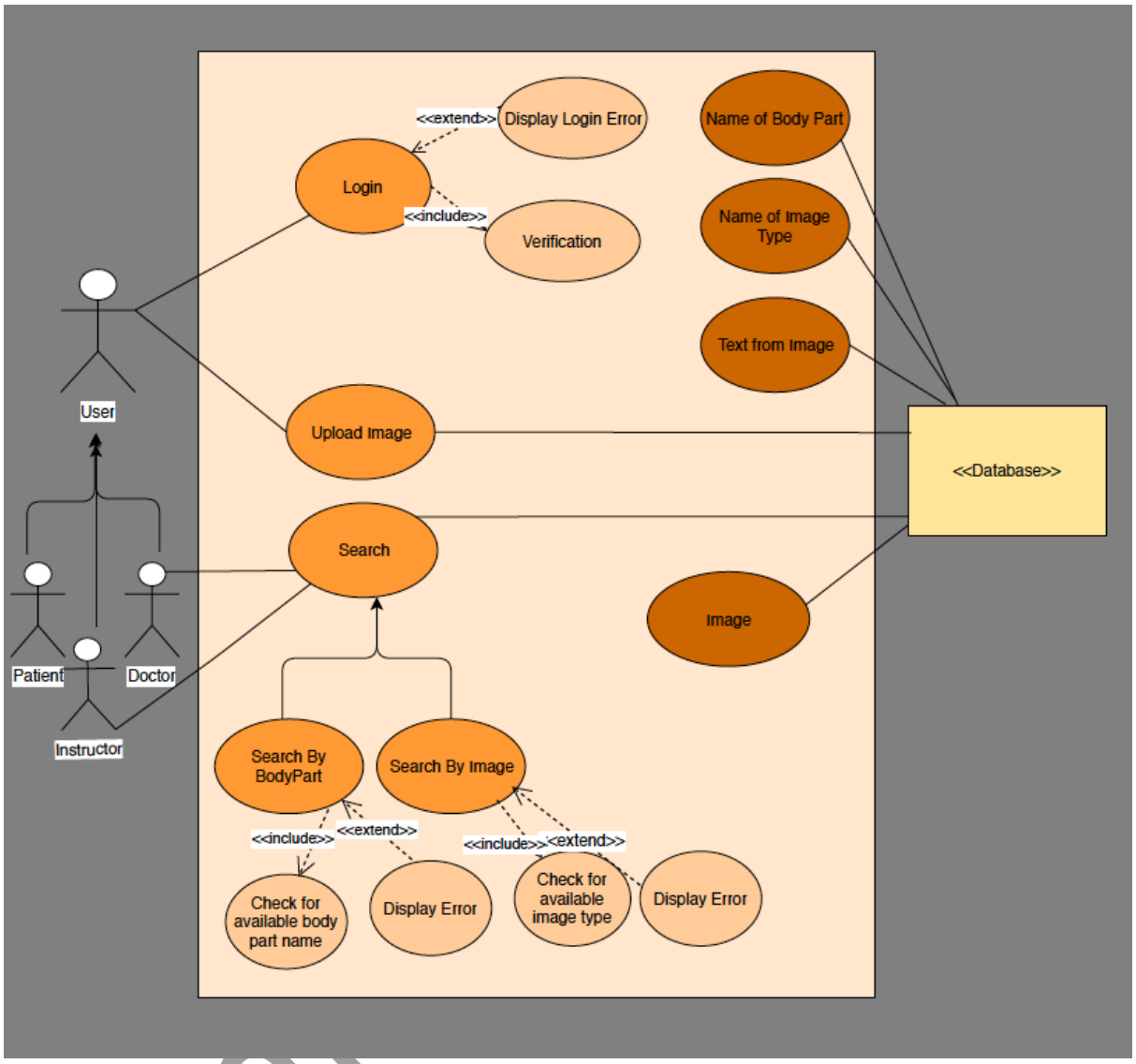
Imaging has become an essential component in many fields of medical and laboratory research and clinical practice. Radiologists identify and quantify tumors from MRI and CT scans; and neuroscientists detect regional metabolic brain activity from PET and functional MRI scans. Analysis of these diverse image types requires sophisticated computerized quantification and visualization tools. Until recently, 3D visualization of images and quantitative analysis could only be performed using expensive UNIX workstations and customized software. Today, much of the visualization and analysis can be performed on an inexpensive desktop computer equipped with the appropriate graphics hardware and software using machine learning algorithms. This project introduces an extensible, , general-purpose image processing and visualization program specifically designed to meet the needs of an Internet-linked medical research community. Using this analysis tool, researcher and clinicians at remote sites can easily share research data and analyses, thereby enhancing their ability to study, diagnose, monitor and treat medical disorders.

Motivation for project:

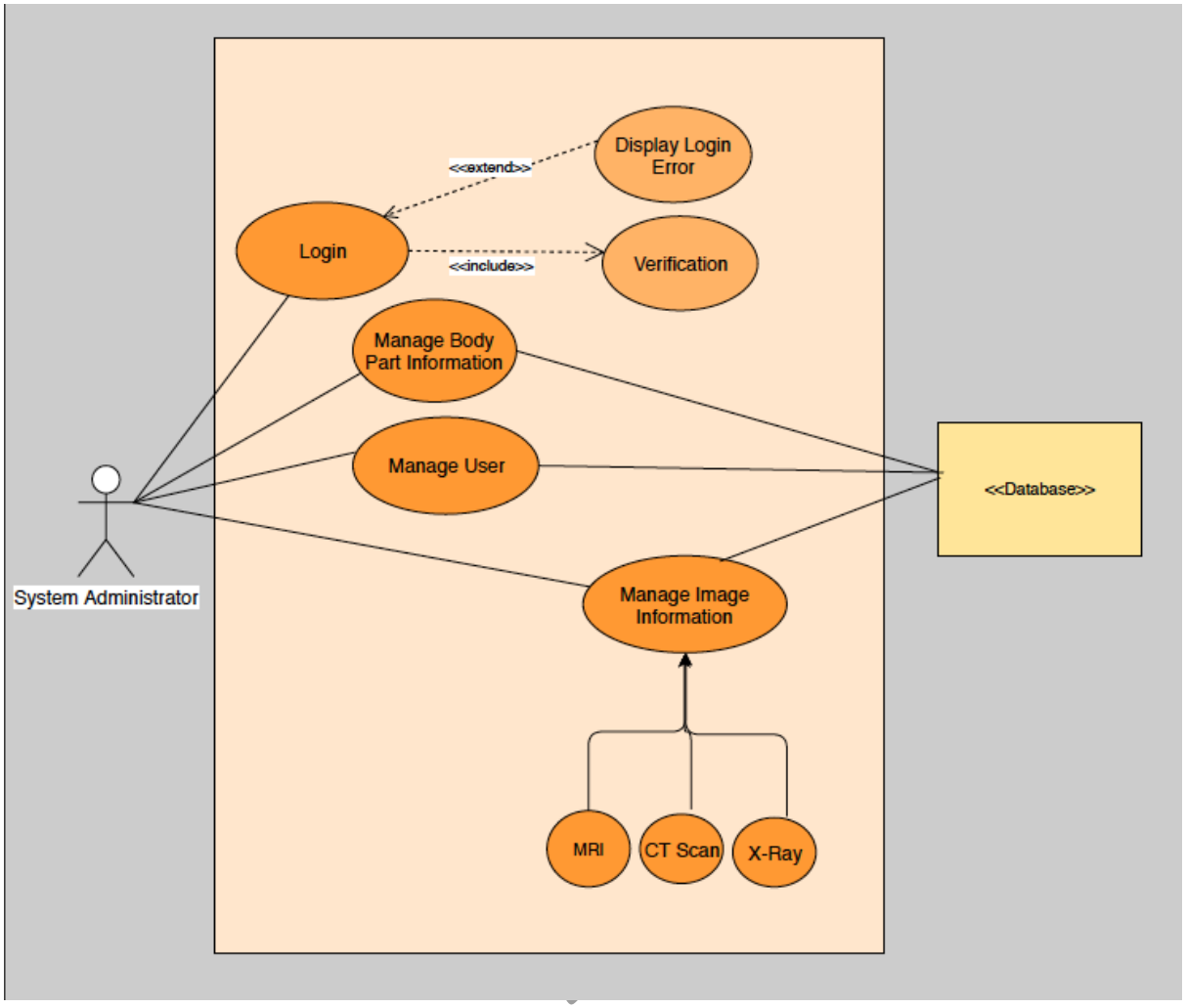
Time is lost between patients getting their body-part scanned, having their images analyzed by doctors, and scheduling a follow-up appointment. We propose a system to process images in real-time and allow people to seek & manage the treatment system.
The need for a comprehensive and automated method of disease screening has long been recognized, and there have been previous efforts for image classification, pattern recognition, and machine learning. With photos of different types of scans as input, the goal of this capstone is to create a new model, ideally resulting in realistic clinical potential.

Use Case Diagram:

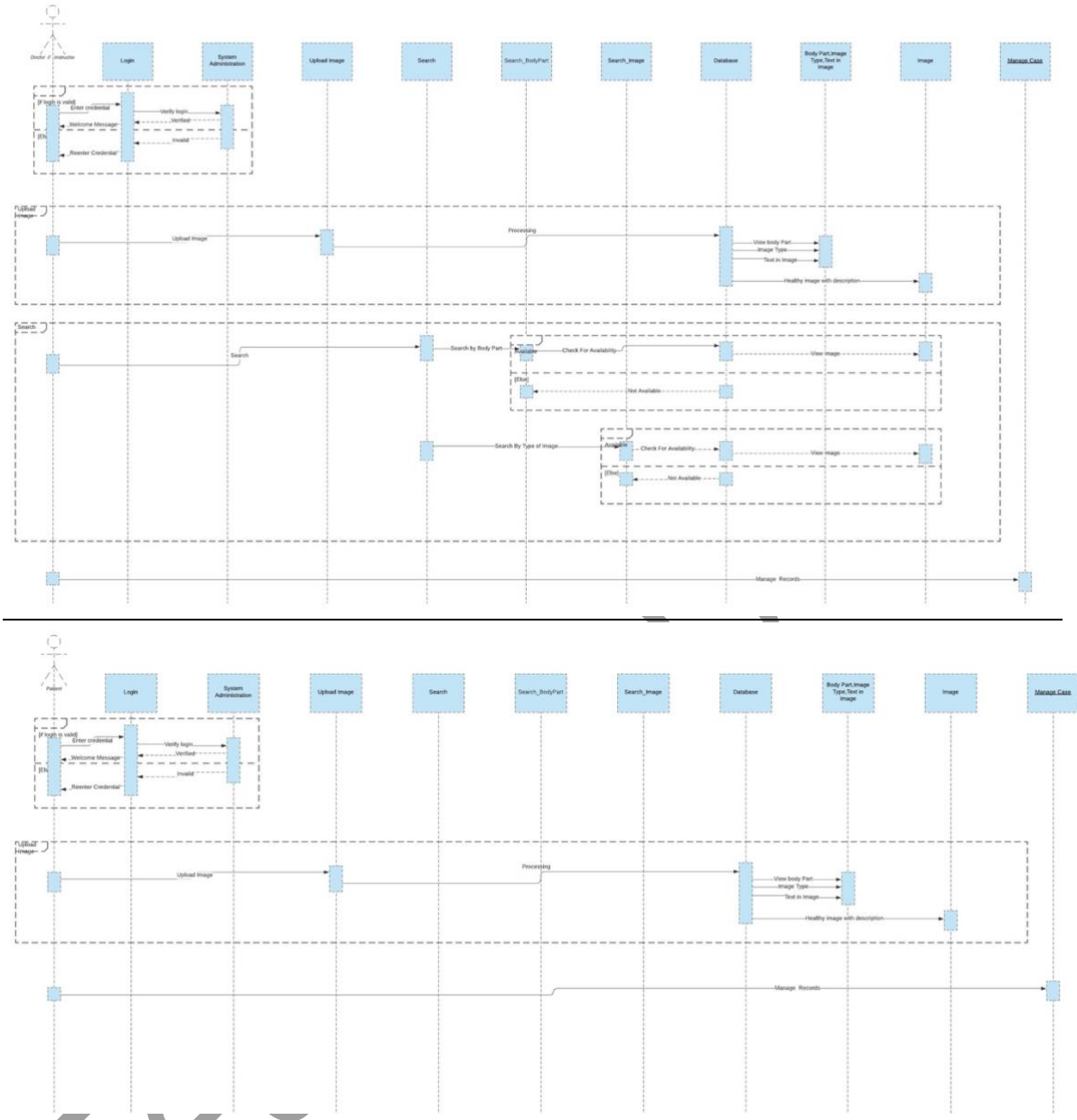
1



2


Display Login Error
<<extend>>
Login
<<include>>
Verification
Manage Body Part Information
Manage User
<<Database>>
System Administrator
Manage Image Information
MRI
CT Scan
X-Ray

Sequence Diagram:



Above sequence diagram represent the interaction between doctor & system and instructor & system when a particular use case is executed.
The vertical dash line represents the lifeline of an entity or object. A lifeline with an entity element represents system data.
The solid bar in a sequence diagram is called an active bar, placed on the lifeline which indicates that an object is active during an interaction between two objects. The length of the bar indicates the duration of the objects staying active.
A horizontal solid arrow is used to convey a message in a sequence diagram. And horizontal dash line arrow is used to give a response to that message.

Machine Learning Models:

Steps to follow: -

- Data Pre-processing
- Algorithm to choose
- Training of Model
- Validation of Model
- Testing of the Model

Data Pre-Processing:

The initial step in every machine learning model is to find out appropriate dataset. This is a much important step for any machine learning model.

One problem we faced that there was no readily available dataset. So, we started gathering the images from various sites for the body type (Brain, Knee, Shoulder) and image type (CT, MRI, X-ray). So, we gathered around 500 images in total as our image dataset.

These images were in medical format i.e. (. Dicom files), so we had to convert it into (.jpeg or .png) format, so we can use image processing libraries such as OpenCV in our model. Therefore, we converted these images into (.jpeg files) through online convertor.

After getting all the images in (.jpeg), we distributed these images into three folders according to Body type that are Brain, Knee and Shoulder. We did same process for images of different image types i.e. MRI, X-ray and CT scan.

The next step is to divide this data into three different folders, as there are three processes aligned with machine learning model that are Training, Validation and Testing.

In Training, we trained our model with the initial data available. Next, we do Validation, where we can evaluate the training process. Last step is testing, where we randomly give the images (not provided before in training and validation) to check the efficiency of the model.

Machine Learning Algorithms:

After getting appropriate dataset, next important step is to find which algorithm will give better accuracy for your project. Here, we are implementing the image processing on the dataset containing medical images, so the algorithm which we are looking for should give good accuracy in the process of image processing.

There are plenty of algorithms with which we can design machine learning model to implement image processing.

The machine learning algorithms are as follows:

- Logistic Regression: - Logistic Regression is used to estimate discrete values such as Binary Values 0/1, Yes/No, True/False depending on the input dataset of independent variables. So, this algorithm is nothing but used to predict the probability of occurrence

of an event by fitting data into logit function. So, because it is giving the probability values, so output values are between 0 to 1.

Mathematics behind the logit function: -

Odds = P / (1-P) = (probability of event Occurrence / Probability of not event Occurrence)

Ln (Odds) = ln (P / (1-P))

Logit (P) = ln (P / (1-P)) = b0 + b1X1 + b2X2 + b3X3…. + bkXk

- Decision Tree: - It is supervised learning algorithm used for classification problems. The algorithm works on the principle of splitting the population data into two or more sets. This is based on the distinct grouping of most significant data attributes.

- SVM (Support Vector Machine)**:** - In Support Vector Machine algorithm, there is plotting of each data item in n-dimensional space where n is number of features, with the value of each feature being the value of coordinate. Sometimes, algorithm uses the differentiating regression line to divide the data into distinct groups according to data sets available.

- kNN (k -Nearest Neighbors): - It is used for both Regression and Classification problems. It is very simple algorithm that stores all available cases and classifiers new cases by a majority vote of its k neighbors. The class is assigned to the class which is most common to its K nearest neighbors measured by a distance function. These functions can be any distance functions such as Euclidean, Manhattan, Minkowski and Hamming. Ideally, if the value of k is 1, then that case simply gets assigned to that nearest neighbor.

- Random Forest: - Random Forest is a collection of decision trees. So, to classify the objects based on attributes, each tree gives a classification votes, and then forest chooses the classification with most votes.

- Naïve Bayes: - Naïve Bayes classifier works on the principle of having the presence of a feature in a class is irrespective of the presence of any other feature. This algorithm is particularly useful on vary large data sets (Text classification) and calculates the probability of events using conditional probability.

  Probability: - P(c|x) = [P(x|c) X P(c)] / P(x)

  P(c|X) = P(x1|c) x P(x2|c) x P(x3|c) …. x P(xn|c) x P(c)

  Where, P(c|x) is Posterior Probability of target given predictor

  P(x|c) is likelihood probability of predictor using class

  P(c) is class prior probability

  P(x) is Predictor Prior Probability

<u>Training with these Algorithms:</u>

Here, we have written the python code to implement training of the model using above algorithms.
The basic step of any code is to import all required packages that are important for processing. We have also used several packages such as Numpy, PIL, scikit-learn, Keras, TensorFlow, OpenCV and imutils etc.

- NumPy: For numerical processing with Python.
- PIL: A simple image processing library.
- scikit-learn: Contains the machine learning algorithms
- Keras and TensorFlow: For deep learning packages and implementations
- OpenCV: To do Image Processing
- imutils: package of image processing/computer vision convenience functions
- argparse: To parse the arguments

```python
# import the necessary packages
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from sklearn.linear_model import LinearRegression
from PIL import Image
from imutils import paths
import numpy as np
import argparse
import os

def extract_color_stats(image):
    (R, G, B) = image.split()
    features = [np.mean(R), np.mean(G), np.mean(B), np.std(R),
        np.std(G), np.std(B)]
    return features

# construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-d", "--dataset", type=str, default="Final",
    help="path to directory containing the 'Dataset_path' dataset")
ap.add_argument("-m", "--model", type=str, default="mlp",
    help="type of python machine learning model to use")
args = vars(ap.parse_args())

```

Once we have imported the required packages, next thing we can do is to get the arguments. We have used argument parser here to get arguments like --dataset (path where our dataset is stored) and then --model (The Python machine learning model to employ.). These arguments will get from command prompt through user. We can use default values too, as we have set here for model and dataset.

Next, we have made a dictionary that defines the suite of models with their default implemented function. So, every time user pass model argument, this function will come to picture.

Then we have retrieved the images from dataset using for loop. To store these images with their respective labels, we have used two lists named Data [] and labels []. Also, for every image we have introduced RGB (Red, Green, Blue) function that calculates the RGB mean values for each image. Here, we have used the function LabelEncoder() to transform the textual labels into numbers.

Once we get all the image data, then we have called the train and validation split, in which we have divided the data into 75% for training and 25% for testing.

```
models = {"knn": KNeighborsClassifier(n_neighbors=1),
    "naive_bayes": GaussianNB(),
    "logit": LogisticRegression(solver="lbfgs", multi_class="auto"),
    "svm": SVC(kernel="linear"),
    "decision_tree": DecisionTreeClassifier(),
    "random_forest": RandomForestClassifier(n_estimators=100),}

print("[INFO] extracting image features...")
imagePaths = paths.list_images(args["dataset"])
data = []
labels = []

for imagePath in imagePaths:
    image = Image.open(imagePath)
    features = extract_color_stats(image)
    data.append(features)
    label = imagePath.split(os.path.sep)[-2]
    labels.append(label)

le = LabelEncoder()
labels = le.fit_transform(labels)

(trainX, testX, trainY, testY) = train_test_split(data, labels,
    test_size=0.25)

print("[INFO] using '{}' model".format(args["model"]))
model = models[args["model"]]
model.fit(trainX, trainY)

print("[INFO] evaluating...")
predictions = model.predict(testX)
print(classification_report(testY, predictions,
    target_names=le.classes_))
```

The next and final step is to train and evaluate the model. In python, to do any functionalities on the model we can use predefined variable as model. Then by using model.fit (), we have fit the data into model for training.

At last, we have evaluated the model on the testing set and then print a classification report on the terminal.

Results on Training

There are certain parameters with which we can check accuracy or efficiency of the model. parameters are as follows:

1. True Positives (TP) - These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes.
2. True Negatives (TN) - These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no.
3. False Positives (FP) – When actual class is no and predicted class is yes. E.g. if actual class says this passenger did not survive but predicted class tells you that this passenger will survive.
4. False Negatives (FN) – When actual class is yes but predicted class in no. E.g. if actual class value indicates that this passenger survived and predicted class tells you that passenger will die.

Accuracy - Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations.
Accuracy = TP+TN/TP+FP+FN+TN

Precision - Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.
Precision = TP/TP+FP

Recall (Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class
Recall = TP/TP+FN

F1 score - F1 Score is the weighted average of Precision and Recall.
F1 Score = 2*(Recall * Precision) / (Recall + Precision)

- results on training data using Naive-Bayes:

```
PS C:\Users\prathmesh\Desktop\python-machine-learning> python .\classification_images.py
[INFO] extracting image features...
[INFO] using 'naive_bayes' model
[INFO] evaluating...
              precision    recall  f1-score   support

          ct       0.58      0.86      0.69        22
         mri       0.77      0.47      0.59        36
        xray       0.77      0.89      0.83        19

    accuracy                           0.69        77
   macro avg       0.71      0.74      0.70        77
weighted avg       0.72      0.69      0.68        77
```

- results on training data using Logistic Regression:

```
PS C:\Users\prathmesh\Desktop\python-machine-learning> python .\classification_images.py
[INFO] extracting image features...
[INFO] using 'logit' model
C:\Users\prathmesh\p\lib\site-packages\sklearn\linear_model\logistic.py:947: ConvergenceWarning: lbfgs failed to converge. Increase the number of iterations.
  "of iterations.", ConvergenceWarning)
[INFO] evaluating...
              precision    recall  f1-score   support

          ct       0.96      0.96      0.96        25
         mri       0.88      0.88      0.88        34
        xray       0.83      0.83      0.83        18

    accuracy                           0.90        77
   macro avg       0.89      0.89      0.89        77
weighted avg       0.90      0.90      0.90        77
```

- results on training data using KNN

```
PS C:\Users\prathmesh\Desktop\python-machine-learning> python .\classification_images.py
[INFO] extracting image features...
[INFO] using 'knn' model
[INFO] evaluating...
              precision    recall  f1-score   support

          ct       0.89      0.96      0.92        25
         mri       0.94      0.86      0.90        37
        xray       0.88      0.93      0.90        15

    accuracy                           0.91        77
   macro avg       0.90      0.92      0.91        77
weighted avg       0.91      0.91      0.91        77
```

- results on training data using Decision Trees:

```
PS C:\Users\prathmesh\Desktop\python-machine-learning> python .\classification_images.py
[INFO] extracting image features...
[INFO] using 'decision_tree' model
[INFO] evaluating...
              precision    recall  f1-score   support

          ct       0.96      0.96      0.96        28
         mri       0.76      0.93      0.84        30
        xray       0.92      0.58      0.71        19

    accuracy                           0.86        77
   macro avg       0.88      0.83      0.84        77
weighted avg       0.87      0.86      0.85        77
```

- results on training data using Random Forest

```
PS C:\Users\prathmesh\Desktop\python-machine-learning> python .\classification_images.py
[INFO] extracting image features...
[INFO] using 'random_forest' model
[INFO] evaluating...
              precision    recall  f1-score   support

          ct       0.96      1.00      0.98        23
         mri       0.97      0.87      0.92        38
        xray       0.79      0.94      0.86        16

    accuracy                           0.92        77
   macro avg       0.91      0.94      0.92        77
weighted avg       0.93      0.92      0.92        77
```

- results on training data using Support Vector Machine

```
PS C:\Users\prathmesh\Desktop\python-machine-learning> python .\classification_images.py
[INFO] extracting image features...
[INFO] using 'svm' model
[INFO] evaluating...
              precision    recall  f1-score   support

          ct       0.96      0.96      0.96        25
         mri       0.92      0.92      0.92        37
        xray       0.87      0.87      0.87        15

    accuracy                           0.92        77
   macro avg       0.92      0.92      0.92        77
weighted avg       0.92      0.92      0.92        77
```


CONVOLUTIONAL NEURAL NETWORK: -

Convolutional Neural Network (CNN) is a special architecture of artificial neural networks. The main goal of the image classification is to accept the input data with its class, then learn the image features and attributes, and at last determine the picture in the image. For this, image get passed through a series of such as Convolutional, nonlinear, pooling layers and fully connected layers and then generates output.

The Convolutional Layer is the first layer of Convolutional Neural Network. Image gets stored in the memory in matrix of pixels that is the value of the intensity at that point. Then, computer selects small moving matrix, which is called as 'Filter'. And then filter produces convolution, that is moves along the input image. The filter's task is to multiply the pixel values of small matrix and then add them. This number gets stored in next layer.

The second layer is Nonlinear Layer, which has activation function. The functions bring nonlinear property in the model, without which network is unable to model for response variables.

The Pooling Layer is third layer followed by nonlinear layer. It is used to perform down sampling operation by changing its height and width. Due to this, image volume gets reduced. That means if some of the features is already have been identified through previous convolutional operations, then those features get excluded through this process, as those features are not required for further classification process.

After these layers, it is required to have Fully Connected Layer at last. Through this layer, we ger N dimensional vector, where N is the number of classes from which model selects desired class.

Training the model using Convolutional Neural Network: -

First step of training is to import the necessary packages. So here, we are importing the packages such as Numpy, Imutils, Argparse, Pickle, OS, PIL, Sklearn and Keras.

```python
# import the necessary packages
from keras.models import Sequential
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
from keras.layers.core import Activation
from keras.layers.core import Flatten
from keras.layers.core import Dense
from keras.optimizers import Adam
from sklearn.preprocessing import LabelBinarizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from PIL import Image
from imutils import paths
import numpy as np
import argparse
import pickle
import os
```

Next, we have constructed the argument parser through argparse library to parse the arguments such as

dataset (to specify path to directory containing the dataset), model (to specify path to output trained model), label (to specify output label binarizer). By doing this, we can call these values through the command prompt while executing the model. Otherwise, we can give the default values for the same.

```python
# construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-d", "--dataset", type=str, default="final",
    help="path to directory containing the 'final' dataset")
ap.add_argument("-m", "--model", required=True,
    help="path to output trained model")
ap.add_argument("-l", "--label-bin", required=True,
    help="path to output label binarizer")
args = vars(ap.parse_args())
```

We are storing the dataset directory path into variable ‘imagePaths’ for future reference. Also, we are initializing the lists for image and label storage. Then, we are iterating the for loop over image folder to grab all the images. Then we have resized the image into 32X32 pixels and scaled it into pixel intensities of range [0, 1] by dividing with 255. Then we are storing resized images and respective labels into data list and labels list respectively.
After this we have encoded the labels by converting them from strings to integers with the help of the function LabelBinarizer.

```
print("[INFO] loading images...")
imagePaths = paths.list_images(args["dataset"])
data = []
labels = []

for imagePath in imagePaths:

    image = Image.open(imagePath)
    image = np.array(image.resize((32, 32))) / 255.0
    data.append(image)

    label = imagePath.split(os.path.sep)[-2]
    labels.append(label)

lb = LabelBinarizer()
labels = lb.fit_transform(labels)
```

Then we have performed a training and testing split by using 70% of the data for training and 25% for evaluation/ validation. Validation is very much important step to evaluate the Algorithm performance.

```
(trainX, testX, trainY, testY) = train_test_split(np.array(data),
    np.array(labels), test_size=0.30)
```

After pre-processing of the data and model, we have defined our Convolutional Neural Network architecture. Here, we have used Keras to do modelling part, as it is an efficient way to describe the model. In Keras, we don't have to declare any weights or bias variables unlike other libraries like TensorFlow.

First step in initializing the convolutional neural network model is to define which model we are going to use. We have declared the Sequential model as it allows us to easily stack Sequential layers of the network in the order from first layer(input) to last layer(output).

The first part of the convolutional neural network is 2D Convolutional Layer. First argument passed through convolutional layer is output channels, which we have declared to 8. Second Argument is kernel size, which is nothing but a size of moving window. we have set the kernel size to 3X3. Then we are declaring the input size as 32X32X3, which tells the network that the size of image is width 32 and height 32 with RGB channels 3. Declaration input shape is only required of the first layer- as Keras model is good enough to work out the size of the tensors flowing through the model from there. Next is Activation Function, that is ReLU in this case. ReLU stands for rectified linear unit. Next, we have added the 2D Max pooling layer. We have specified the pooling layer size as 2X2. and same in case for strides that is 2X2. This is all about one convolutional layer. Then we have added more two more layers with output channels 16, 32.

Now we are flattening the output through Fully Connected Layers. Next part of the fully connected layer is Dense layer, where we have specified 3 nodes with Activation function that soft-max classification. SoftMax it's a function, not a loss. It squashes a vector in the range (0, 1) and all the resulting elements add up to 1. It is used for multi-class classification.

```
# define our Convolutional Neural Network architecture
model = Sequential()
model.add(Conv2D(8, (3, 3), padding="same", input_shape=(32, 32, 3)))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Conv2D(16, (3, 3), padding="same"))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Conv2D(32, (3, 3), padding="same"))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Flatten())
model.add(Dense(3))
model.add(Activation("softmax"))
```

Here, we have used the Adam Optimizer to optimize and, we used categorical crossentropy as a loss function. It is a **SoftMax activation** plus a **Cross-Entropy loss**. The choice of loss function is very much important, so that can be used to estimate the loss of the model so that the weights can be updated to reduce the loss on the next evaluation. If we use categorical crossentropy loss, we will train a CNN to output a probability over the classes for each image. It is used for multi-class classification.
Then, we have fit the model with training and validation data. We also passed two arguments, which are epochs (number of time training happens) and batch size (Number of images passed during one epoch). We have set epochs to 25, and then batch size to 32.

```
# train the model using the Adam optimizer
print("[INFO] training network...")
opt = Adam(lr=1e-3, decay=1e-3 / 50)
model.compile(loss="categorical_crossentropy", optimizer=opt,
    metrics=["accuracy"])
H = model.fit(trainX, trainY, validation_data=(testX, testY),
    epochs=25, batch_size=32)
```

Then, we have written the code for predict method to evaluate the model.

```
# evaluate the network
print("[INFO] evaluating convolutioal neural network...")
predictions = model.predict(testX, batch_size=32)
print(classification_report(testY.argmax(axis=1),
    predictions.argmax(axis=1), target_names=lb.classes_))
```

At last, we have saved the model using Pickle library. For this, we have already passed the arguments such as model and label, to get an idea where are we storing this model.

```
80  #save the model and label binarizer to disk
81  print("[INFO] serializing network and label binarizer...")
82  model.save(args["model"])
83  f = open(args["label_bin"], "wb")
84  f.write(pickle.dumps(lb))
85  f.close()
```

Results on Training Data using Convolutional Neural Network:-

```
Epoch 23/25
214/214 [==============================] - 0s 764us/step - loss: 0.0111 - accuracy: 1.0000 - val_loss: 0.0520 - val_accuracy: 0.9783
Epoch 24/25
214/214 [==============================] - 0s 718us/step - loss: 0.0099 - accuracy: 1.0000 - val_loss: 0.0554 - val_accuracy: 0.9783
Epoch 25/25
214/214 [==============================] - 0s 722us/step - loss: 0.0088 - accuracy: 1.0000 - val_loss: 0.0499 - val_accuracy: 0.9783
[INFO] evaluating convolutioal neural network...
              precision    recall  f1-score   support

          ct       0.97      1.00      0.98        32
         mri       0.98      0.98      0.98        43
        xray       1.00      0.94      0.97        17

    accuracy                           0.98        92
   macro avg       0.98      0.97      0.98        92
weighted avg       0.98      0.98      0.98        92
```

Why Convolutional Neural Network?

Convolutional Neural Networks are nothing but efficient automatic feature extractors. CNN uses adjacent pixel values to effectively down sample the image first by convolution and then uses a prediction layer at the end. While, if we use other algorithms, there is possibility of losing a spatial interaction between pixels. Because there is no transfer of learning is occurring. On other hand, CNNs will work more efficiently if you train them with more and more samples. We can compare this fact comparing the training and evaluating accuracies of the above algorithms with the accuracies of Convolutional Neural Network. The highest training accuracy, we got on training is on SVM (96%, 92%, 86%), but in case of CNN, we got accuracy (97%, 98%, 100%). So, we can conclude that for image classification, we can use Convolutional Neural Network algorithm.

TESTING ON MODEL

Here, it is a code for testing of the model, where we have passed the images on which we want to do testing. For that, we have passed arguments such as image path, training model path. Once we are done with pre-processing of the image such as resizing and flattening (No need to do this if we have trained the model with Convolutional Neural Network, as the fully connected layer in the CNN already does this part), we have written the code for prediction. To do image operations and analysis, we have used OpenCV library. And at last, we have displayed the results on the image with the percentage.

```python
from keras.models import load_model
import argparse
import pickle
import cv2
from PIL import Image
import numpy as np

# construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--image", required=True,
    help="path to input image we are going to classify")
ap.add_argument("-m", "--model", required=True,
    help="path to trained Keras model")
ap.add_argument("-l", "--label-bin", required=True,
    help="path to label binarizer")
ap.add_argument("-f", "--flatten", type=int, default=0,
    help="whether or not we should flatten the image")
args = vars(ap.parse_args())

image = cv2.imread(args["image"])
output = image.copy()

image = cv2.resize(image, (32, 32))
image = image.astype("float") / 255.0

if args["flatten"] > 0:
    image = image.flatten()
    image = image.reshape((1, image.shape[0]))
else:
    image = image.reshape((1, image.shape[0], image.shape[1],
        image.shape[2]))

# load the model and label binarizer
print("[INFO] loading network and label binarizer...")
model = load_model(args["model"])
lb = pickle.loads(open(args["label_bin"], "rb").read())

# make a prediction on the image
preds = model.predict(image)

# find the class label index with the largest corresponding
# probability
i = preds.argmax(axis=1)[0]
label = lb.classes_[i]

# draw the class label + probability on the output image
text = "{}: {:.2f}%".format(label, preds[0][i] * 100)
cv2.putText(output, text, (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 0.7,
    (0, 0, 255), 2)

# show the output image
cv2.imshow("Image", output)
cv2.waitKey(0)
```

Text Detection and Recognition:

Our next objective of the project is to identify the text written on the medical image, to get an information such body alignment, image attributes etc. For image analysis, we have used

python library OpenCV and with that for Text detection and extraction purpose, we have used EAST Text detection algorithm.

Why we chose EAST detector?

- Image/sensor noise: Sensor noise from a handheld camera is typically higher than that of a traditional scanner.
- Viewing angles: Natural scene text can naturally have viewing angles that are not parallel to the text, making the text harder to recognize.
- Blurring and Lighting conditions
- Resolution: Not all cameras are created equal — we may be dealing with cameras with sub-par resolution.
- Unknown layout: We cannot use any a priori information to give our algorithms "clues" as to where the text resides.

OpenCV's text detector implementation of EAST is quite robust, capable of localizing text even when it's blurred, reflective, or partially obscured

Data preprocessing

- All images were scaled down to 256 by 256. Despite taking longer to train, the detail present in photos of this size is much greater then at 128 by 128.
- Inverting the photo to preserve the relative contrast between the elements

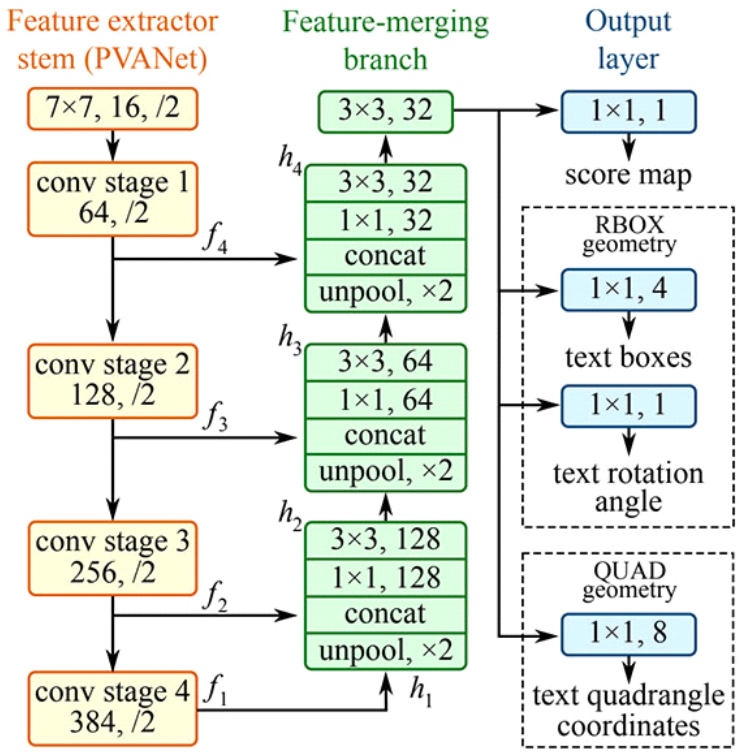


Figure 1: The structure of the EAST text detection Fully-Convolutional Network

Implementation:

Step 1:

To begin, we import our required packages and modules. Notably we import NumPy, OpenCV
We then proceed to the command line arguments:

- --image: The path to our input image.

- --east: The EAST scene text detector model file path.

```
# import the necessary packages
from imutils.object_detection import non_max_suppression
import numpy as np
import argparse
import time
import cv2

# construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--image", type=str,
    help="path to input image")
ap.add_argument("-east", "--east", type=str,
    help="path to input EAST text detector")
ap.add_argument("-c", "--min-confidence", type=float, default=0.5,
    help="minimum probability required to inspect a region")
```

Step 2:
Loading the image and resizing it for EAST Text detector.

```
# load the input image and grab the image dimensions
image = cv2.imread(args["image"])
orig = image.copy()
(H, W) = image.shape[:2]

# set the new width and height and then determine the ratio in change
# for both the width and height
(newW, newH) = (args["width"], args["height"])
rW = W / float(newW)
rH = H / float(newH)

# resize the image and grab the new image dimensions
image = cv2.resize(image, (newW, newH))
(H, W) = image.shape[:2]
```

Step 3:
Load the OpenCV's EAST text detector and load the neural network into memory using cv2.dnn.readNet by passing the path to the EAST detector.

```
# load the pre-trained EAST text detector
print("[INFO] loading EAST text detector...")
net = cv2.dnn.readNet(args["east"])

# construct a blob from the image and then perform a forward pass of
# the model to obtain the two output layer sets
blob = cv2.dnn.blobFromImage(image, 1.0, (W, H),
    (123.68, 116.78, 103.94), swapRB=True, crop=False)
start = time.time()
net.setInput(blob)
(scores, geometry) = net.forward(layerNames)
end = time.time()
```

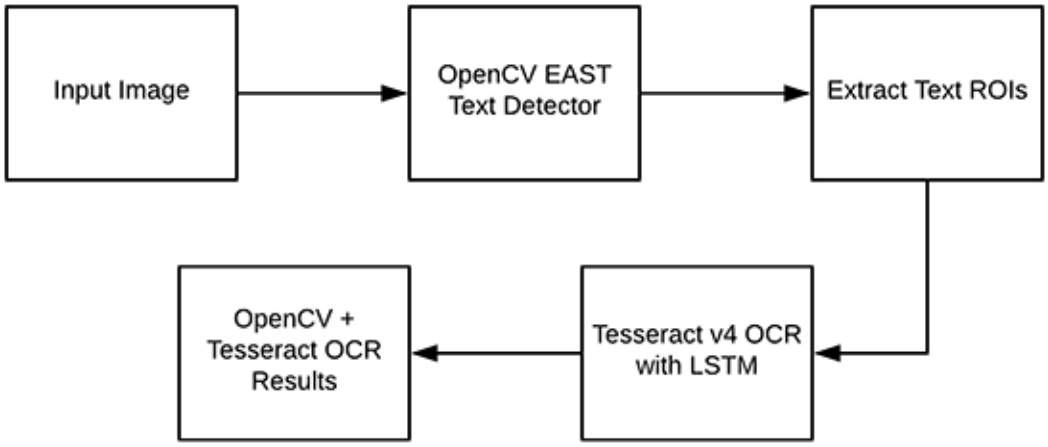
Step 4:
Extract our scores and geometry data

OCR and Text Recognition

Once we have detected the text regions with OpenCV, we'll then extract each of the text ROIs and pass them into Tesseract, enabling us to build an entire OpenCV OCR pipeline!
Requirements:

1) Numpy
2) TensorFlow
3) Tesseract

Understanding the flow



Implementation:

Step 1:

Start the decode function that does –

- Uses a deep learning-based text detector to detect (not recognize) regions of text in an image.
- The text detector produces two arrays, one containing the probability of a given area containing text, and another that maps the score to a bounding box location in the input image.

```
def decode_predictions(scores, geometry):
    # grab the number of rows and columns from the scores volume, then
    # initialize our set of bounding box rectangles and corresponding
    # confidence scores
    (numRows, numCols) = scores.shape[2:4]
    rects = []
    confidences = []
```

Step 2:

Preprocess using the EAST Text Detector to get the Text ROI's

Step 3:

we continue this process for other ROIs at the top of the loop and display/print the results of the text extracted:

```
# sort the results bounding box coordinates from top to bottom
results = sorted(results, key=lambda r:r[0][1])

# loop over the results
for ((startX, startY, endX, endY), text) in results:
    # display the text OCR'd by Tesseract
    print("OCR TEXT")
    print("========")
    print("{}\n".format(text))

    # strip out non-ASCII text so we can draw the text on the image
    # using OpenCV, then draw the text and a bounding box surrounding
    # the text region of the input image
    text = "".join([c if ord(c) < 128 else "" for c in text]).strip()
    output = orig.copy()
    cv2.rectangle(output, (startX, startY), (endX, endY),
        (0, 0, 255), 2)
    cv2.putText(output, text, (startX, startY - 20),
        cv2.FONT_HERSHEY_SIMPLEX, 1.2, (0, 0, 255), 3)

    # show the output image
    cv2.imshow("Text Detection", output)
    cv2.waitKey(0)
```

Limitations and Drawbacks:

It's important to understand that no OCR system is perfect!

There is no such thing as a perfect OCR engine, especially in real-world conditions.

And furthermore, expecting 100% accurate Optical Character Recognition is simply unrealistic.

As we found out, our OpenCV OCR system worked in well in some images, it failed in others.

There are two primary reasons we will see our text recognition pipeline fail:

- The text is skewed/rotated.
- The font of the text itself is not like what the Tesseract model was trained on.

Storing the extracted text in Database

```
def insertBLOB(path, label3, label4, text3, text4):
    print("Inserting BLOB into result table")
    try:
        connection = mysql.connector.connect(host='localhost',
                                             database='Medical_Imaging',
                                             user=       ,
                                             password=  )

        cursor = connection.cursor()
```

We are using mysql.connector to connect to our mysql database through python script

```
def convertToBinaryData(filename):
    # Convert digital data to binary format
    with open(filename, 'rb') as file:
        binaryData = file.read()
    return binaryData
```

Above snippet is used to convert the image which is passed in arguments into BLOB(Binary Large Objects) which will be stored

```
def insertBLOB(path, label3, label4, text3, text4):
    print("Inserting BLOB into result table")
    try:
        connection = mysql.connector.connect(host='localhost',
                                             database='Medical_Imaging',
                                             user='root',
                                             password='')

        cursor = connection.cursor()

        empPicture = convertToBinaryData(path)
        #file = convertToBinaryData(biodataFile)
        sql_insert_blob_query = "INSERT INTO result (Image, Body_Part_Name, Imaging_Technique_Name, Accuracy_By_Body_Part, Accuracy_By_Technique) VALUES
        (%s,%s,%s,%s, %s)"
        # Convert data into tuple format
        insert_blob_tuple = (empPicture, label3, label4,text3,text4)
        result = cursor.execute(sql_insert_blob_query, insert_blob_tuple)
        connection.commit()
        print("Image and file inserted successfully as a BLOB into mri_image table", result)

    except mysql.connector.Error as error:
        print("Failed inserting BLOB data into MySQL table {}".format(error))

    finally:
        if (connection.is_connected()):
            cursor.close()
            connection.close()
            print("MySQL connection is closed")

insertBLOB("C://Users//prathmesh//Desktop//Final_Project//example.jpg",""+label1, ""+label2, ""+str(text1), ""+str(text2))
```

Above snippet is used to insert the extracted text and the imaged passed in argument.

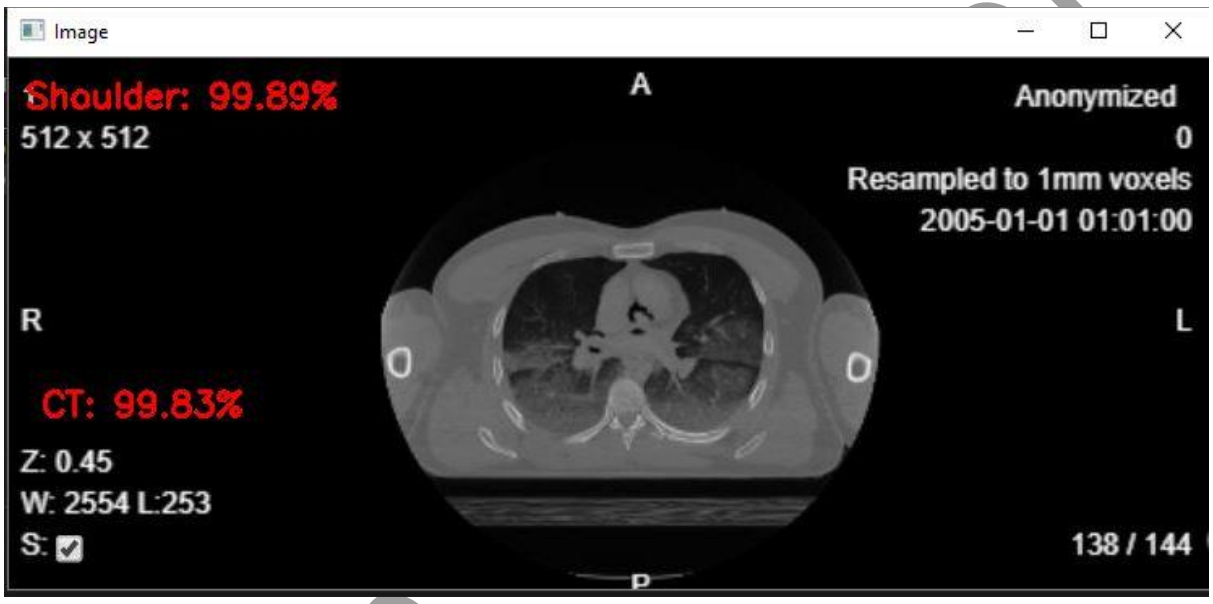
Label1: Body part extracted

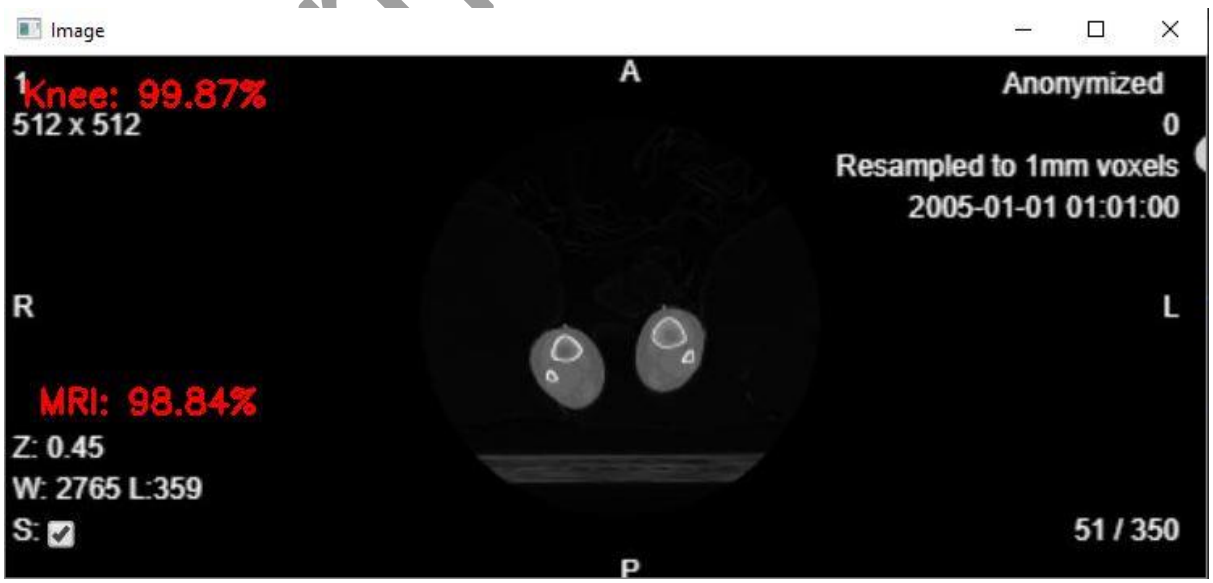
Label2: Imaging Technique

Text1: accuracy by which the body part is identified

Text2: accuracy by which the imaging technique is identified

Results





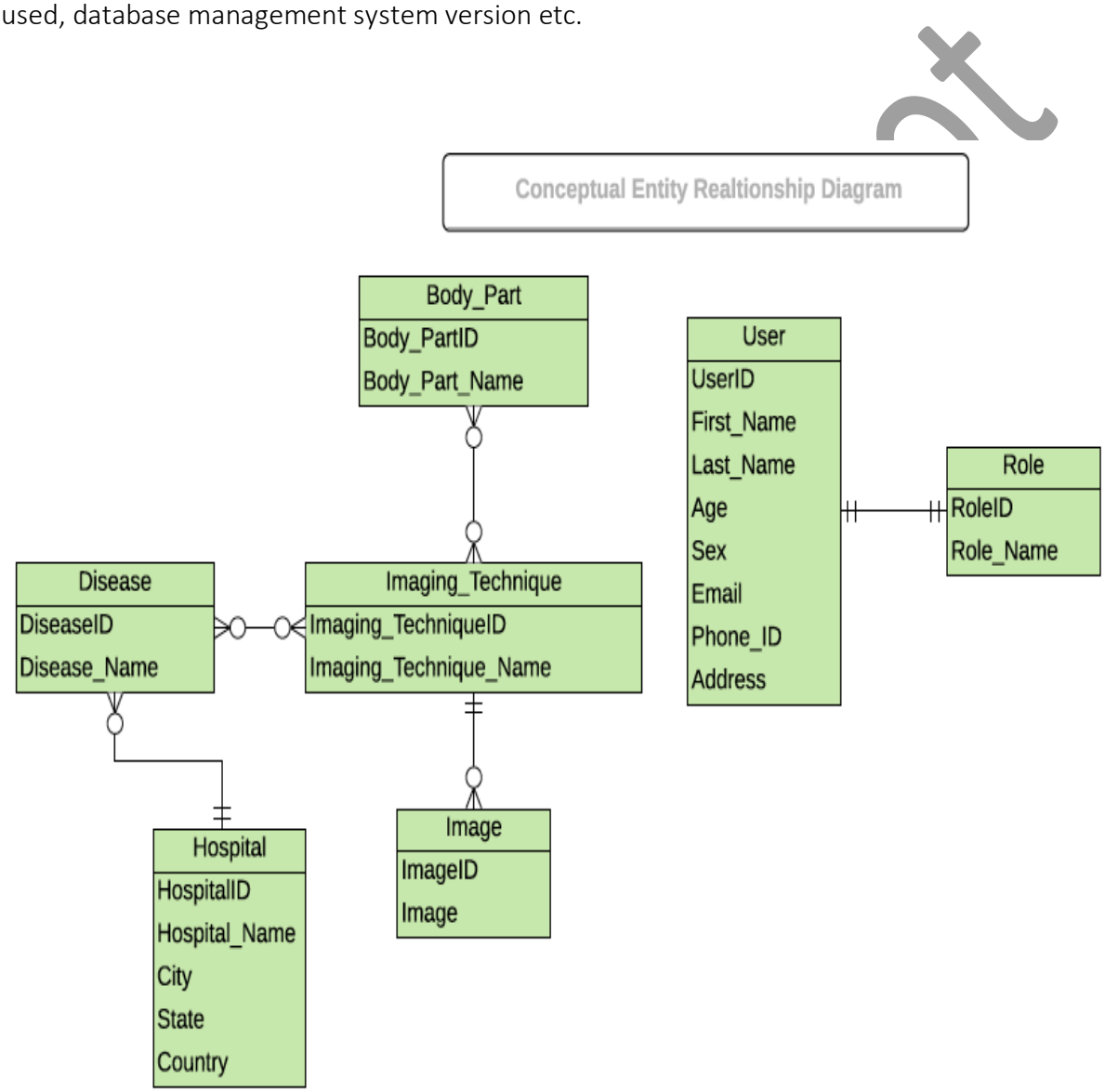
```
Using TensorFlow backend.
------------------------------------------------
--Machine Learning Model and Text recognition--
------------------------------------------------
[INFO] loading network and label binarizer...
2019-12-10 21:06:52.310812: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2
[INFO] loading EAST text detector...
------------------------------------------------
------------Text Recognition Results---------
------------------------------------------------
['a9 83%', '', 'Anonymized', '99.89%', '2005-01-01 01:01:00', '2554 1:253', 'CK eee?', 'Resampled to 1m', 'CT: 9', '512x!', 'Z045', 'iki', '"RU Un Tt', 'x 512']
------------------------------------------------
-------Inserting Results into Database---------
------------------------------------------------
Inserting BLOB into result table
Image and file inserted successfully as a BLOB into mri_image table None
MySQL connection is closed
------------------------------------------------
----------------Description--------------------
------------------------------------------------
[('A shoulder computed tomography scan or (CT or CAT scan) creates cross-sectional images of the shoulder using specialized X-ray cameras. \nThis scan can help doctors see the bones
and soft tissues in the shoulder in order to detect abnormalities. \nThe CT scan may also help identify tumors and blood clots. A CT scan can be performed with or without contrast
e. \nThe contrast material helps your doctor analyze important vessels and structures. \nIt also allows them to identify abnormalities that cannot be seen without the dye.',)]
```

ResultID	Image	Body_Part_Name	Imaging_Technique_Name	Accuracy_By_Body_Part	Accuracy_By_Technique
1	BLOB	Brain	CT	Brain: 99.96%	CT: 100.00%
2	BLOB	Knee	CT	Knee: 89.57%	CT: 71.39%
3	BLOB	Brain	CT	Brain: 99.96%	CT: 100.00%
4	BLOB	Shoulder	CT	Shoulder: 99.89%	CT: 99.83%
5	BLOB	Knee	MRI	Knee: 99.87%	MRI: 98.84%

CONCEPTUAL DESIGN:

Initially, based on the requirements of the project, a conceptual design is made with few attributes and relations. While developing and implementing the database model, new attributes will be added as per the requirements and some of the redundant are deleted. When all the entity- relationships are well defined, we'll go ahead with the implementation of the project by developing database model. Conceptual ERD is the simplest model among all.

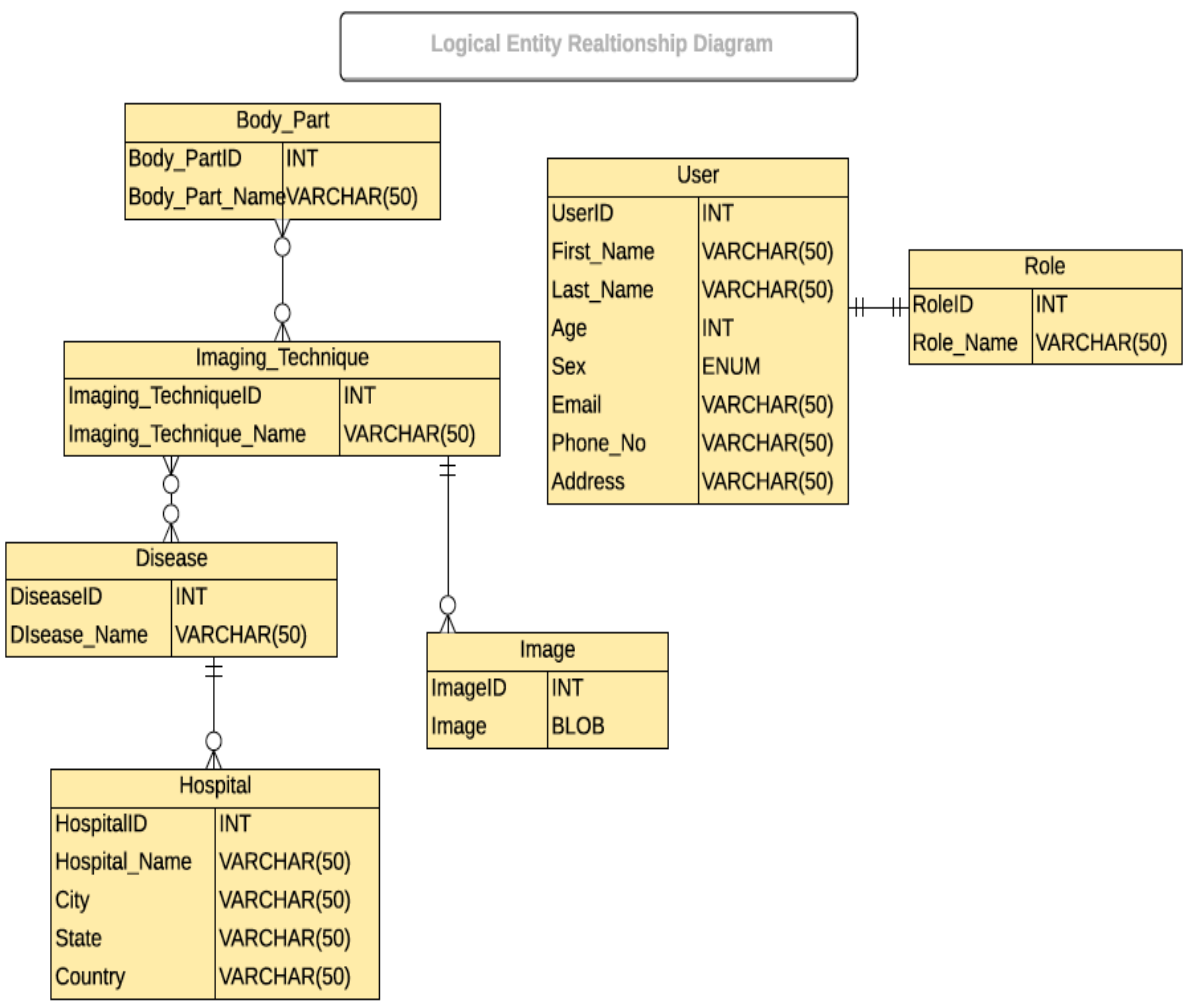
Below is the model developed with platform independent i.e. irrespective of the technology used, database management system version etc.



LOGICAL DESIGN:

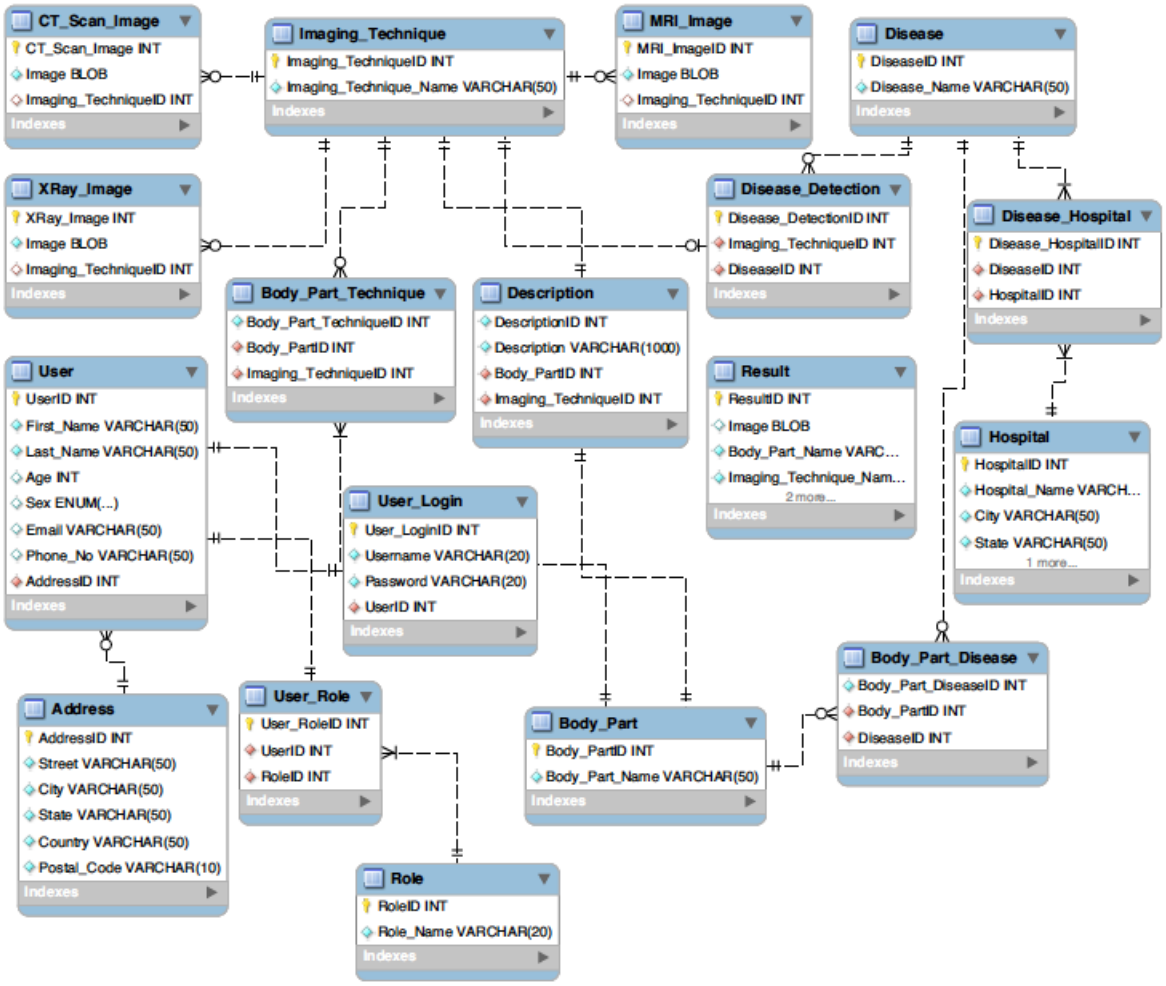
Once the requirements are finalized, a Business Requirements Document (BRD) is published i.e. document detailing all the information regarding entities, relationships between entities, keys, constraints etc. It is more complex than conceptual model in that column types are set to aid business analysis. Setting of column types is optional and it has nothing to do with database creation yet.

Below is the diagram that shows the logical design developed based on the tables created.



PHYSICAL DATABASE DESIGN:

Physical Database design represents the blueprint of a relational database. It represents how data should be structured and related in a specific database management system, so it is important to consider the convention and restriction of the DBMS when we design a physical ERD. We are using the MySQL for the creation and development of the database objects based on the logical model developed with the accurate use of data type for entity columns. We add primary keys, foreign keys and constraints to the design.



Handling Strong Entities:

Imaging_Technique (Imaging_TechniqueID, Imaging_Technique_Name)

Disease (DiseaseID, Disease_Name)

Body_Part (Body_PartID, Body_Part_Name)

Address (AddressID, Street, City, State, Country, Postal_Code)

User (UserID, First_Name, Last_Name, Age, Sex, Email, Phone_No, AddressID)

User_Login (User_LoginID, Username, Password, UserID)

Role (RoleID, Role_Name)

Hospital (HospitalID, Hospital_Name, City, State, Country)

Weak Entities:

Disease_Detection (Disease_DetectionID, Imaging_TechniqueID, DiseaseID)

Body_Part_Disease (Body_Part_DiseaseID, Body_PartID, DiseaseID)

Body_Part_Technique (Body_Part_TechniqueID, Body_PartID, Imaging_TechniqueID)

User_Role (UserRoleID, RoleID, UserID)

Disease_Hospital (Disease_HospitalID, DiseaseID, HospitalID)

MRI_Image (MRI_ImageID, Image, Imaging_TechniqueID)

CT_Scan_Image (CT_Scan_ImageID, Image, Imaging_TechniqueID)

XRay_Image (XRay_ImageID, Image, Imaging_TechniqueID)

DATABASE CREATION:

Medical Imaging:

In this project, we are creating a database named Medical_Imaging. Medical imaging is the technique and process of creating visual representations of the interior of a body for clinical analysis and medical intervention, as well as visual representation of the function of some body parts. It shows internal structures hidden by the skin and bones, as well as help to diagnose and treat disease. The database contains fields related to medical imaging like types of imaging techniques, diseases detected by the techniques, best hospitals, body parts, and images. The databases also have additional information of the user, user login, user role, and the address.

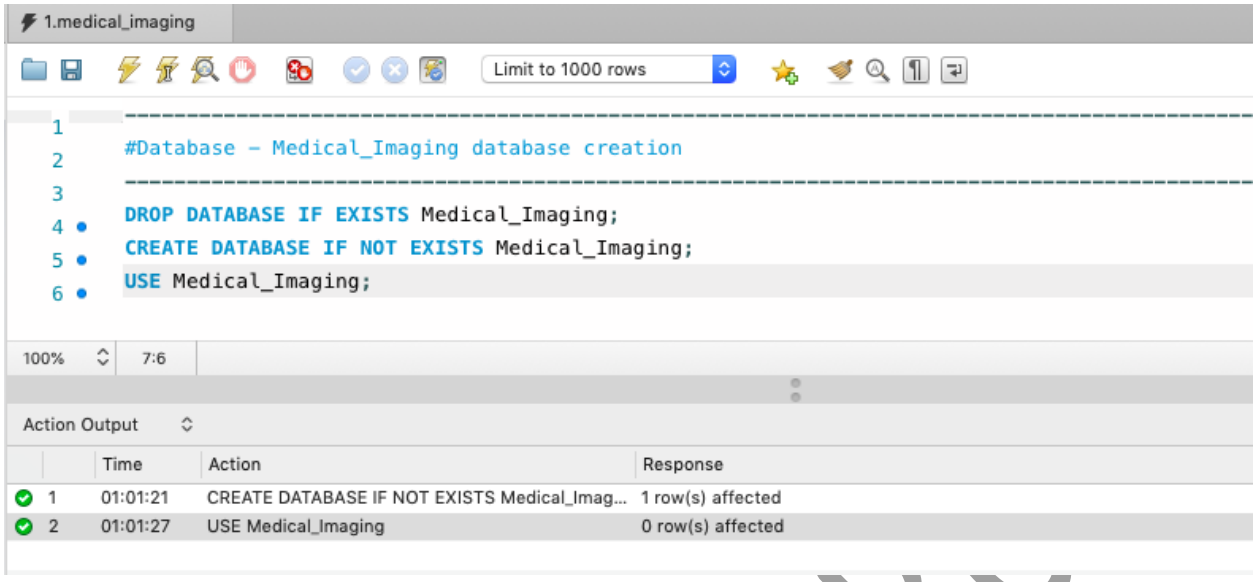


TABLE CREATION:

1. Imaging Technique: This table contains all the imaging techniques used in medical imaging to diagnose disease in humans. Imaging techniques help to understand the causes of an illness and ensure that the diagnosis is accurate. These techniques include x-rays, computed tomography (CT) scans, magnetic resonance imaging (MRI) etc. For this table's identity, we used a PRIMARY KEY constraint on the Imaging_TechniqueID column. We also used other constraints like AUTO_INCREMENT and NOT NULL.

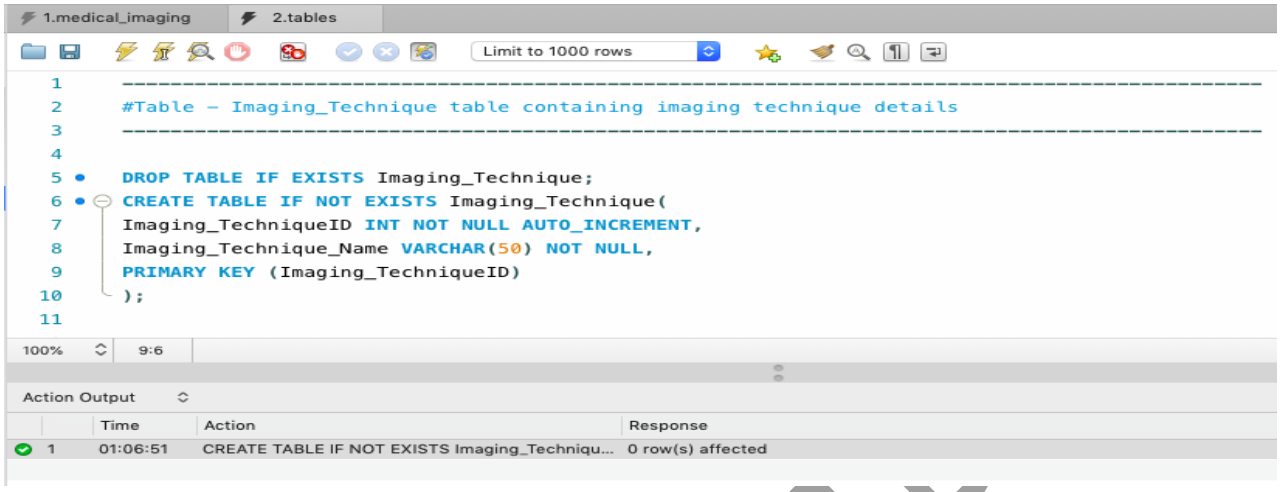
   X-rays are the most common and readily available diagnostic imaging tool which can be used as first tool to diagnose problem before going to do more sophisticated tests.

   Computed tomography (CT) is an imaging technique that combines computer technology with x-rays to produce a more detailed, cross-sectional image of body. By the help of this scan, we can see the size, shape, and position of structures that are deep inside your body, such as organs, tissues, or tumors.

Magnetic resonance imaging (MRI) is diagnostic imaging technique that produces cross-sectional images of your body. MRI works without radiation. The MRI tool uses magnetic fields and computer to take high-resolution pictures of your bones and soft tissues.

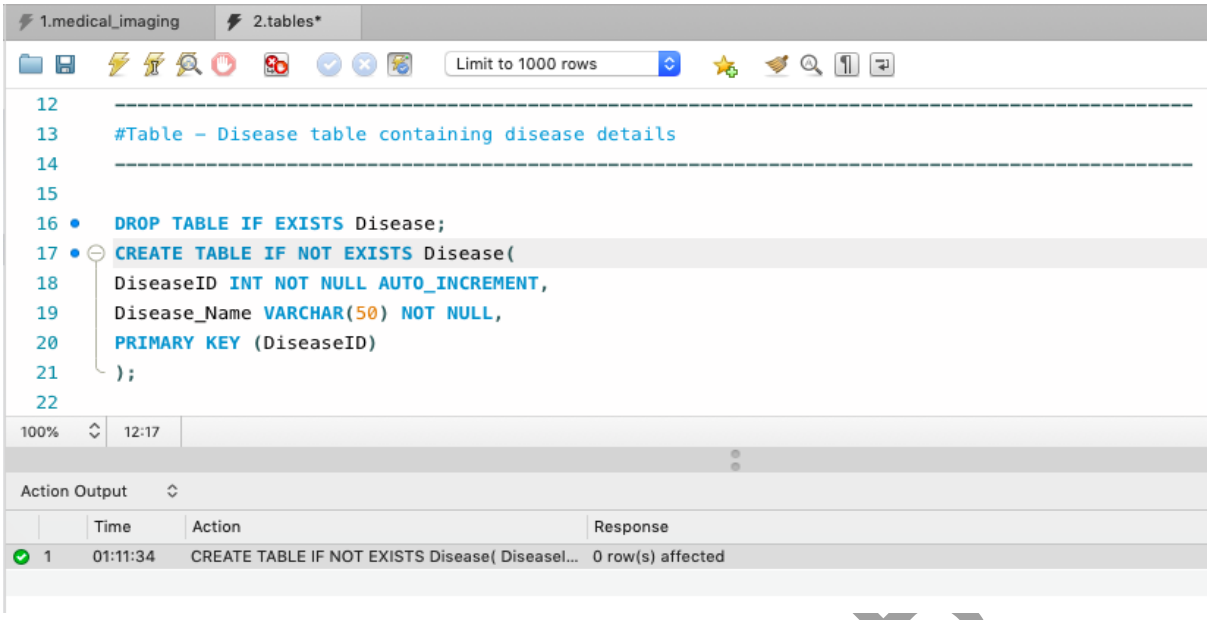
```
-----------------------------------------------------------------------------------------
#Table - Imaging_Technique table containing imaging technique details
-----------------------------------------------------------------------------------------

DROP TABLE IF EXISTS Imaging_Technique;
CREATE TABLE IF NOT EXISTS Imaging_Technique(
Imaging_TechniqueID INT NOT NULL AUTO_INCREMENT,
```

```
Imaging_Technique_Name VARCHAR (50) NOT NULL,
PRIMARY KEY (Imaging_TechniqueID)
);
```



2. Disease: This table includes all the disease which can be diagnosed by the above imaging techniques. For this table's identity, we used a PRIMARY KEY constraint on the DiseaseID column. We also used other constraints like AUTO_INCREMENT and NOT NULL.

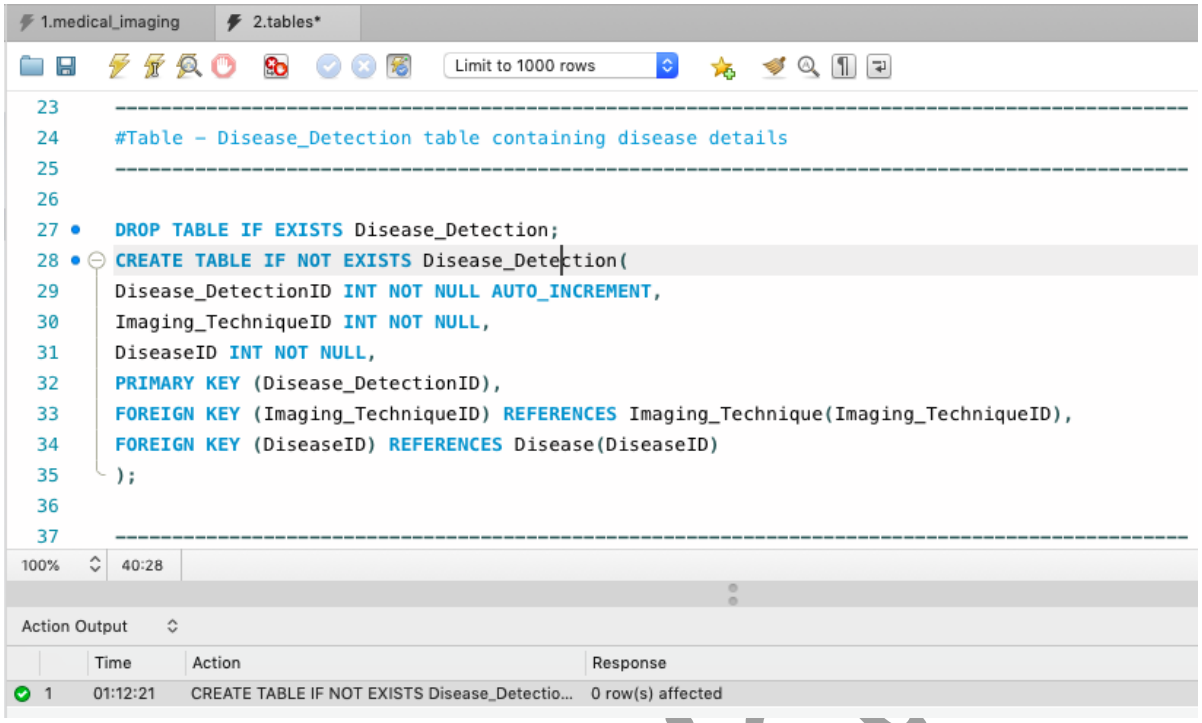
```
-------------------------------------------------------------------------------------------
#Table - Disease table containing disease details
-------------------------------------------------------------------------------------------
DROP TABLE IF EXISTS Disease;
CREATE TABLE IF NOT EXISTS Disease (
DiseaseID INT NOT NULL AUTO_INCREMENT,
Disease_Name VARCHAR (50) NOT NULL,
PRIMARY KEY (DiseaseID)
);
```



3. Disease_Detection: This table contains the disease associated with a particular imaging technique. Here the primary key is Disease_DetectionID. This table includes one column from Imaging_Technique table and another column from Disease table. We used FOREIGN KEY constraint here for Imaging_TechniqueID and DiseaseID and also used AUTO_INCREMENT and NOT NULL constraints.

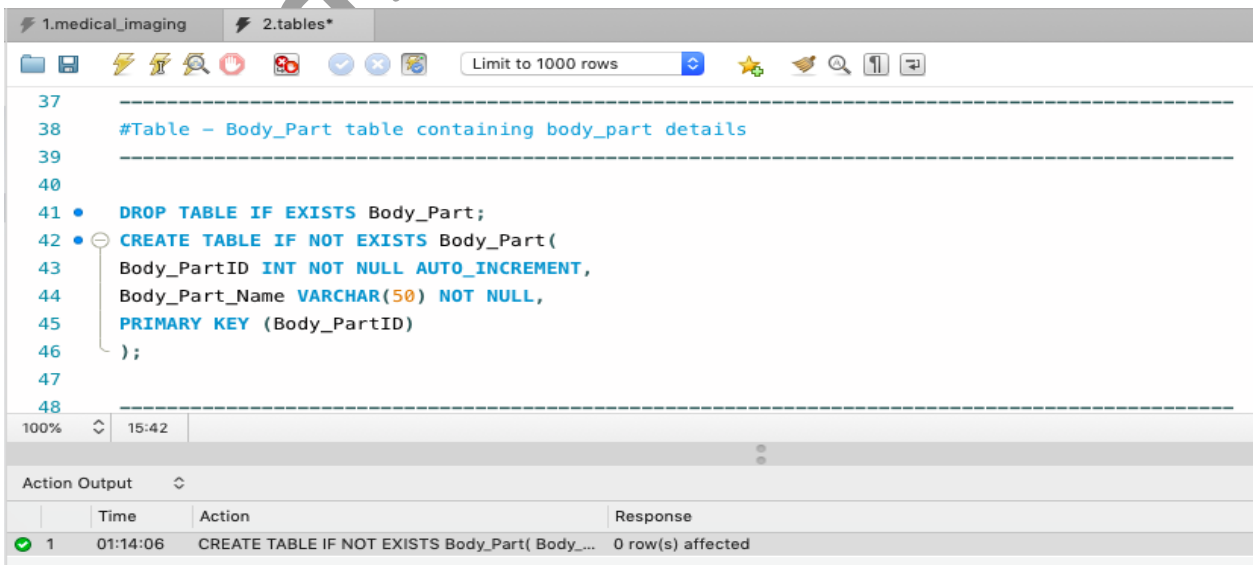
#Table - Disease_Detection table containing disease details

```
DROP TABLE IF EXISTS Disease_Detection;
CREATE TABLE IF NOT EXISTS Disease_Detection(
Disease_DetectionID INT NOT NULL AUTO_INCREMENT,
Imaging_TechniqueID INT NOT NULL,
DiseaseID INT NOT NULL,
PRIMARY KEY (Disease_DetectionID),
FOREIGN KEY (Imaging_TechniqueID) REFERENCES
Imaging_Technique(Imaging_TechniqueID),
FOREIGN KEY (DiseaseID) REFERENCES Disease (DiseaseID)
);
```



4. Body_Part: This table contains the body part name. Here the primary key is Body_PartID and constraints used are AUTO_INCREMENT and NOT NULL.

```
-----------------------------------------------------------------------------------------
#Table - Body_Part table containing body_part details
-----------------------------------------------------------------------------------------

DROP TABLE IF EXISTS Body_Part;
CREATE TABLE IF NOT EXISTS Body_Part(
Body_PartID INT NOT NULL AUTO_INCREMENT,
Body_Part_Name VARCHAR(50) NOT NULL,
PRIMARY KEY (Body_PartID)
);
```

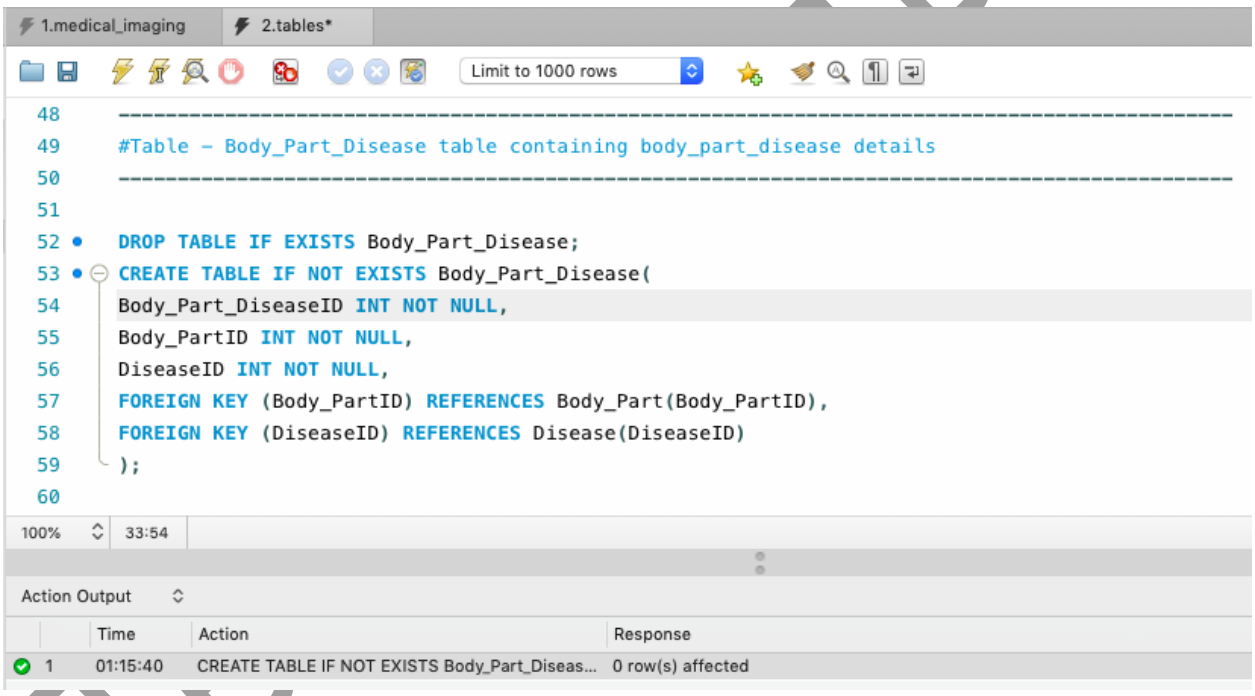


5. Body_Part_Disease: This table incorporates the DiseaseID and Body_PartID as a foreign key which gives us links between disease associated with the particular body part. Here PRIMARY KEY constraint used on Body_Part_DiseaseID column with NOT NULL constraint.

#Table - Body_Part_Disease table containing body_part_disease details

---

```
DROP TABLE IF EXISTS Body_Part_Disease;
CREATE TABLE IF NOT EXISTS Body_Part_Disease(
Body_Part_DiseaseID INT NOT NULL,
Body_PartID INT NOT NULL,
DiseaseID INT NOT NULL,
FOREIGN KEY (Body_PartID) REFERENCES Body_Part(Body_PartID),
FOREIGN KEY (DiseaseID) REFERENCES Disease (DiseaseID)
);
```



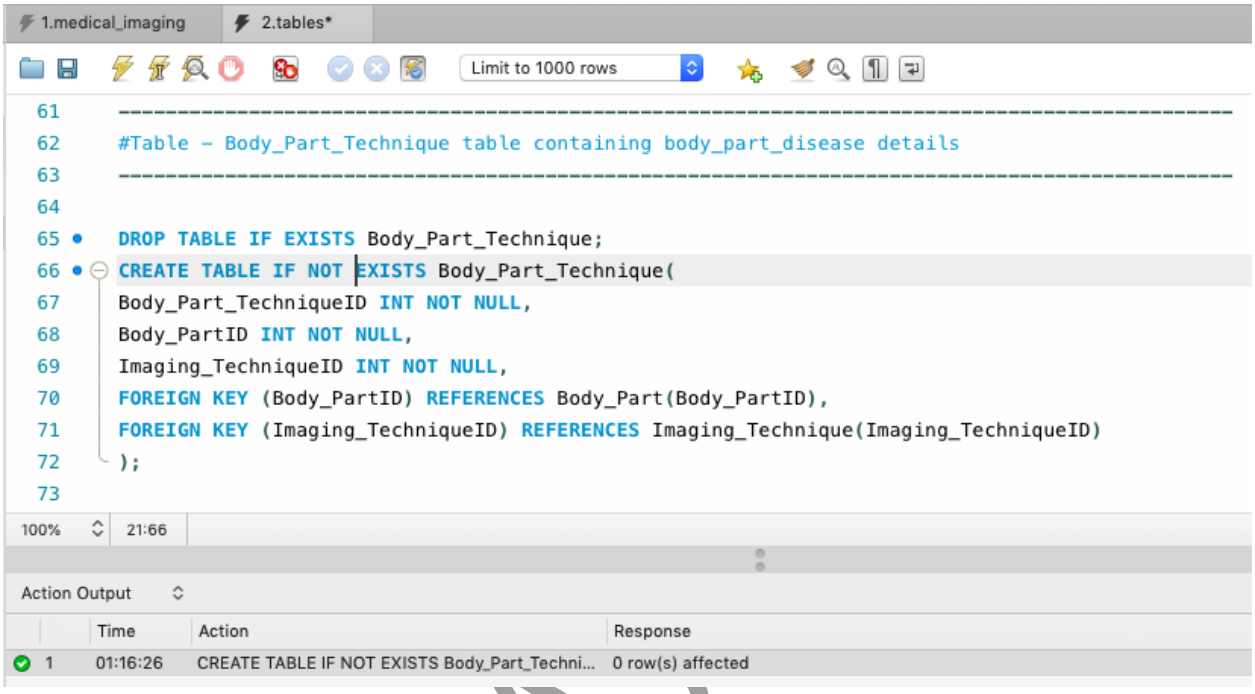
6. Body_Part_Technique: This table incorporates the Body_PartID and Imaging_TechniqueID as a foreign key which gives us links between technique associated with a particular body part. Here PRIMARY KEY constraint used on Body_Part_TechniqueID column with NOT NULL constraint.

#Table - Body_Part_Technique table containing body_part_disease details

---

```
DROP TABLE IF EXISTS Body_Part_Technique;
CREATE TABLE IF NOT EXISTS Body_Part_Technique(
Body_Part_TechniqueID INT NOT NULL,
Body_PartID INT NOT NULL,
```

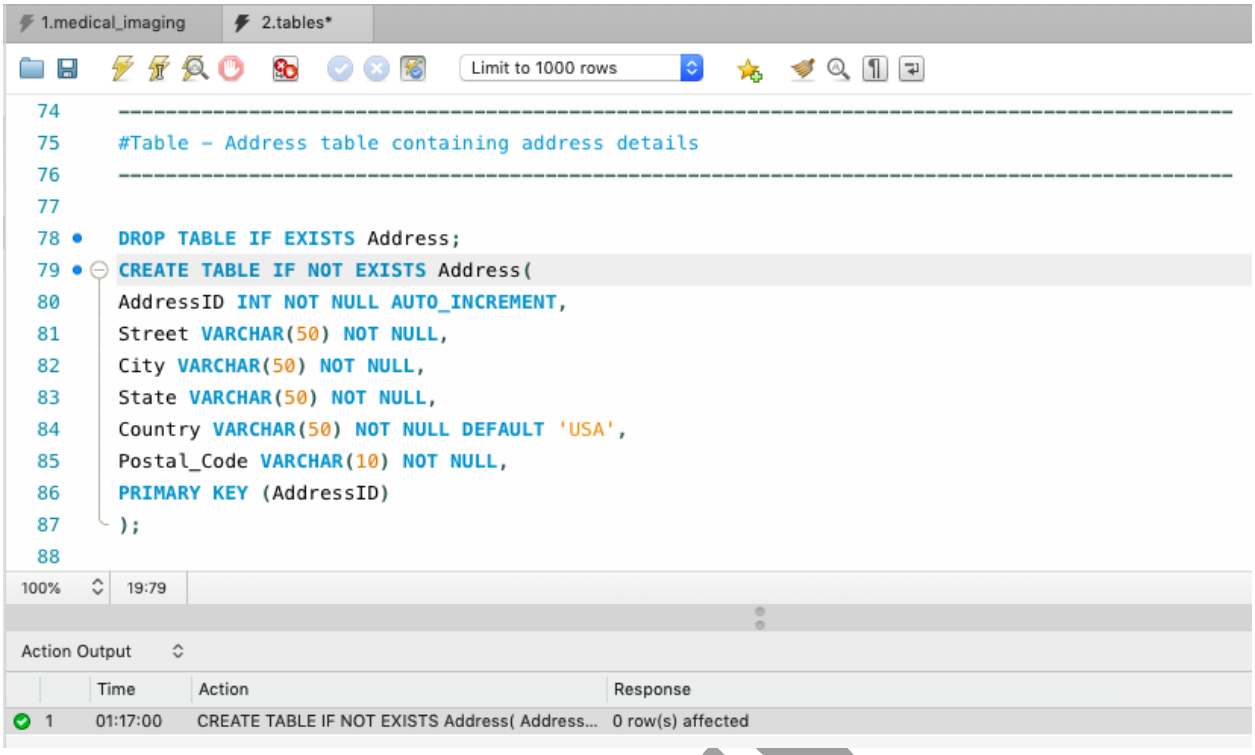
Imaging_TechniqueID INT NOT NULL,
FOREIGN KEY (Body_PartID) REFERENCES Body_Part(Body_PartID),
FOREIGN KEY (Imaging_TechniqueID) REFERENCES
Imaging_Technique(Imaging_TechniqueID)
);



7. Address: This table includes the attribute for an address like street, city, state, country, postal code. Here we used AUTO_INCREMENT, NOT NULL, DEFAULT and PRIMARY KEY constraints.

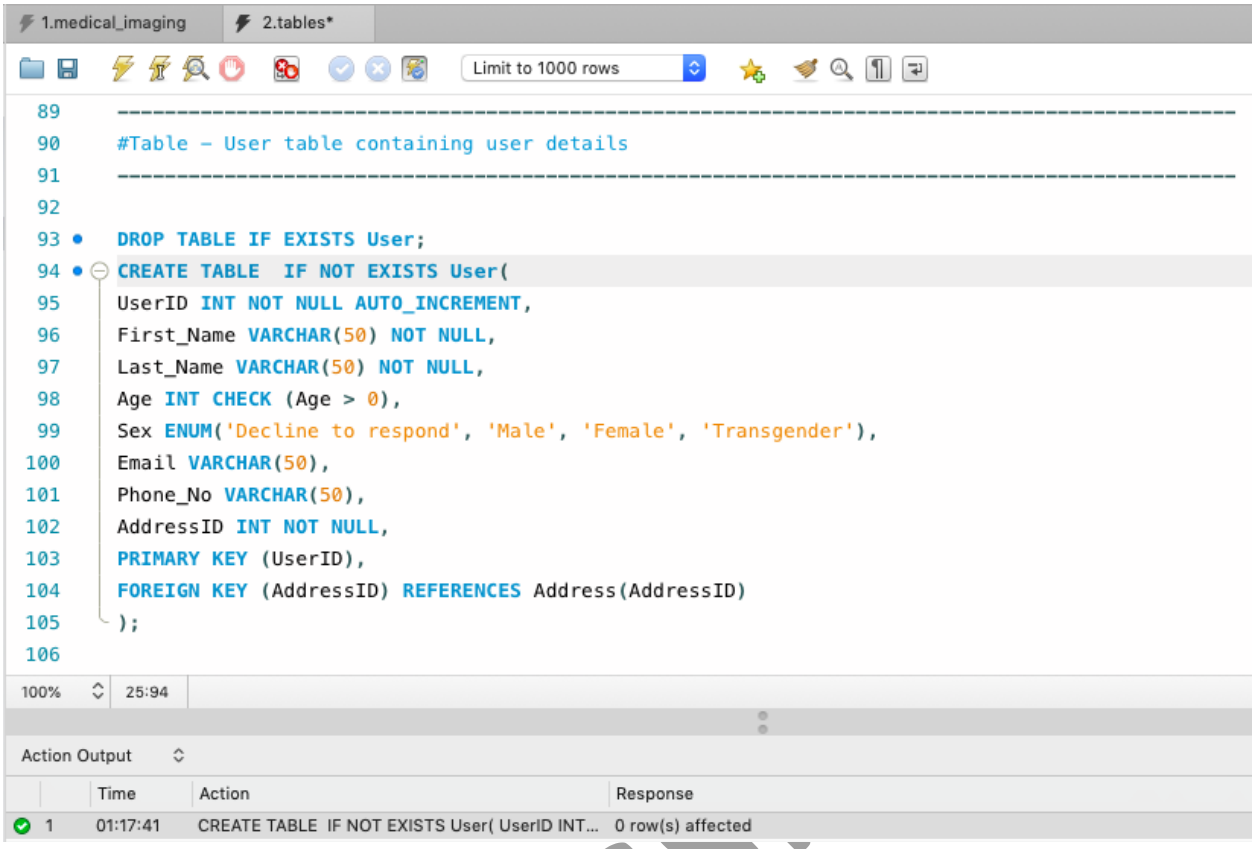
#Table - Address table containing address details

DROP TABLE IF EXISTS Address;
CREATE TABLE IF NOT EXISTS Address(
AddressID INT NOT NULL AUTO_INCREMENT,
Street VARCHAR(50) NOT NULL,
City VARCHAR(50) NOT NULL,
State VARCHAR(50) NOT NULL,
Country VARCHAR(50) NOT NULL DEFAULT 'USA',
Postal_Code VARCHAR(10) NOT NULL,
PRIMARY KEY (AddressID)
);



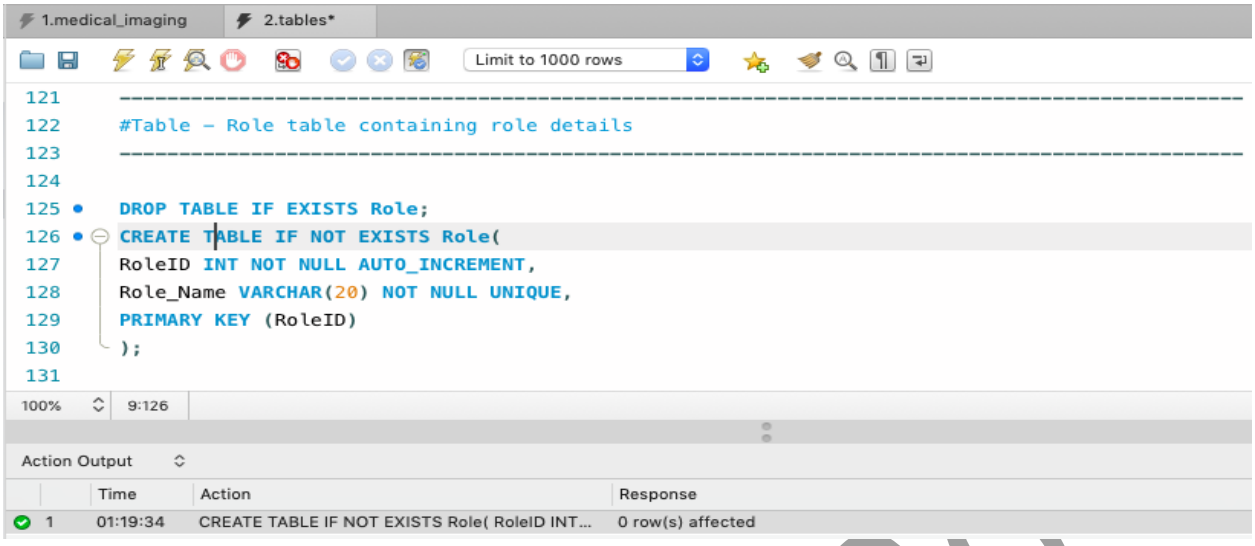
8. User: This table contains all the information of the user like first name, last name, age, sex, email, phone number, address, etc. In this table, we used PRIMARY KEY constraint on UserID column and FOREIGN KEY constraint on AddressID column. We also used AUTO_INCREMENT, NOT NULL and CHECK constraints.

```
----------
#Table - User table containing user details
----------

DROP TABLE IF EXISTS User;
CREATE TABLE IF NOT EXISTS User
(
UserID INT NOT NULL AUTO_INCREMENT,
First_Name VARCHAR (50) NOT NULL,
Last_Name VARCHAR (50) NOT NULL,
Age INT CHECK (Age > 0),
Sex ENUM('Decline to respond', 'Male', 'Female', 'Transgender'),
Email VARCHAR(50),
Phone_No VARCHAR(50),
AddressID INT NOT NULL,
PRIMARY KEY (UserID),
FOREIGN KEY (AddressID) REFERENCES Address(AddressID)
);
```



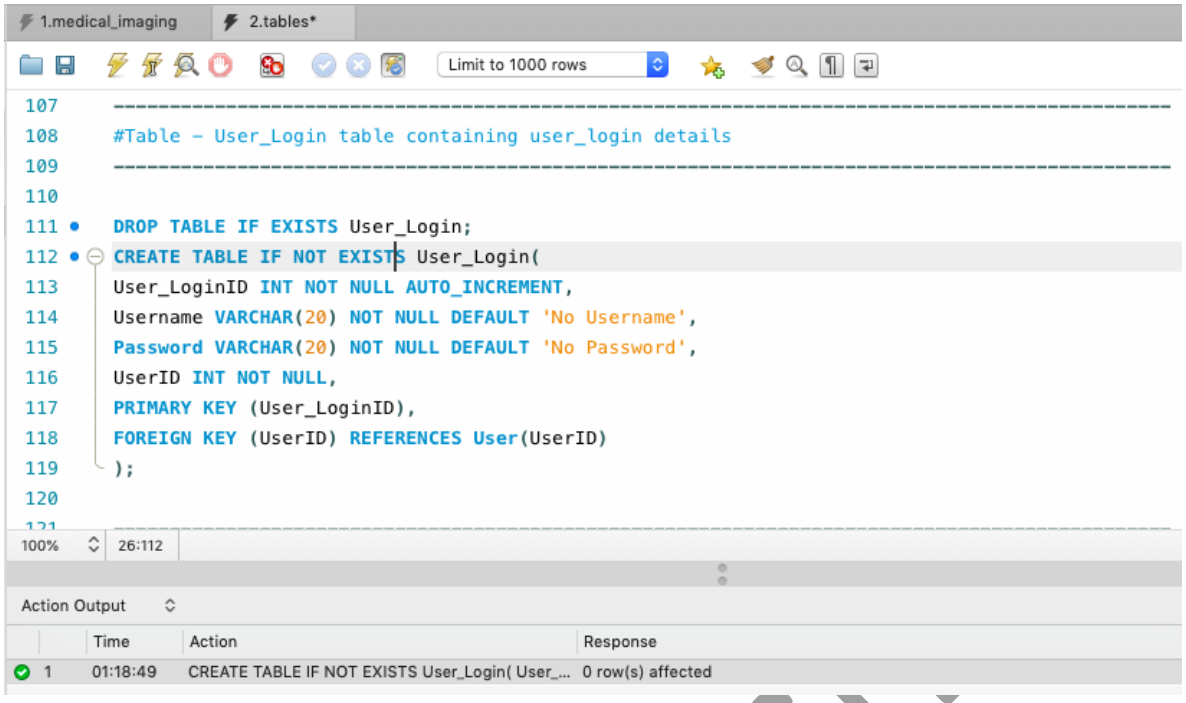
9. Role: Role table includes all the possible role information with a primary key RoleID. This table has AUTO_INCREMENT, NOT NULL, UNIQUE, and PRIMARY KEY constraints.

```
-----------------------------------------------------------------------------------------
#Table - Role table containing role details
-----------------------------------------------------------------------------------------

DROP TABLE IF EXISTS Role;
CREATE TABLE IF NOT EXISTS Role (
RoleID INT NOT NULL AUTO_INCREMENT,
Role_Name VARCHAR (20) NOT NULL UNIQUE,
PRIMARY KEY (RoleID) );
```



10. User_Login: User_Login table is made for users' login by creating a username and password. Here User_LoginID is the primary key and UserID is the foreign key. We are using AUTO_INCREMENT, NOT NULL, DEFAULT, PRIMARY KEY, and FOREIGN KEY constraints.

```
-----------------------------------------------------------------------------------------
#Table - User_Login table containing user_login details
-----------------------------------------------------------------------------------------

DROP TABLE IF EXISTS User_Login;
CREATE TABLE IF NOT EXISTS User_Login(
User_LoginID INT NOT NULL AUTO_INCREMENT,
Username VARCHAR (20) NOT NULL DEFAULT 'No Username',
Password VARCHAR (20) NOT NULL DEFAULT 'No Password',
UserID INT NOT NULL,
PRIMARY KEY (User_LoginID),
FOREIGN KEY (UserID) REFERENCES User (UserID)
);
```



11. User_Role: This table gives the user information with their role in the database. This table includes one column from the User table and another column from the Role table. We also used FOREIGN KEY constraint here for UserID and RoleID and also used AUTO_INCREMENT and NOT NULL constraints.

```
-------------------------------------------------------------------------------------------
#Table - User_Role table containing user_role details
-------------------------------------------------------------------------------------------

DROP TABLE IF EXISTS User_Role;
CREATE TABLE IF NOT EXISTS User_Role(
User_RoleID INT NOT NULL AUTO_INCREMENT,
UserID INT NOT NULL,
RoleID  INT NOT NULL,
PRIMARY KEY (User_RoleID)
);
```

```
133    #Table – User_Role table containing user_role details
134    ----------------------------------------------------------------------------------------------
135
136 •  DROP TABLE IF EXISTS User_Role;
137 •  CREATE TABLE IF NOT EXISTS User_Role(
138    User_RoleID INT NOT NULL AUTO_INCREMENT,
139    UserID INT NOT NULL,
140    RoleID  INT NOT NULL,
141    PRIMARY KEY (User_RoleID)
142    );
143
144 •  ALTER TABLE User_Role
145    ADD CONSTRAINT fk_role_ur
146    FOREIGN KEY(RoleID)
147    REFERENCES Role(RoleID);
148
149 •  ALTER TABLE User_Role
150    ADD CONSTRAINT fk_user_ur
151    FOREIGN KEY(UserID)
152    REFERENCES User(UserID);
```

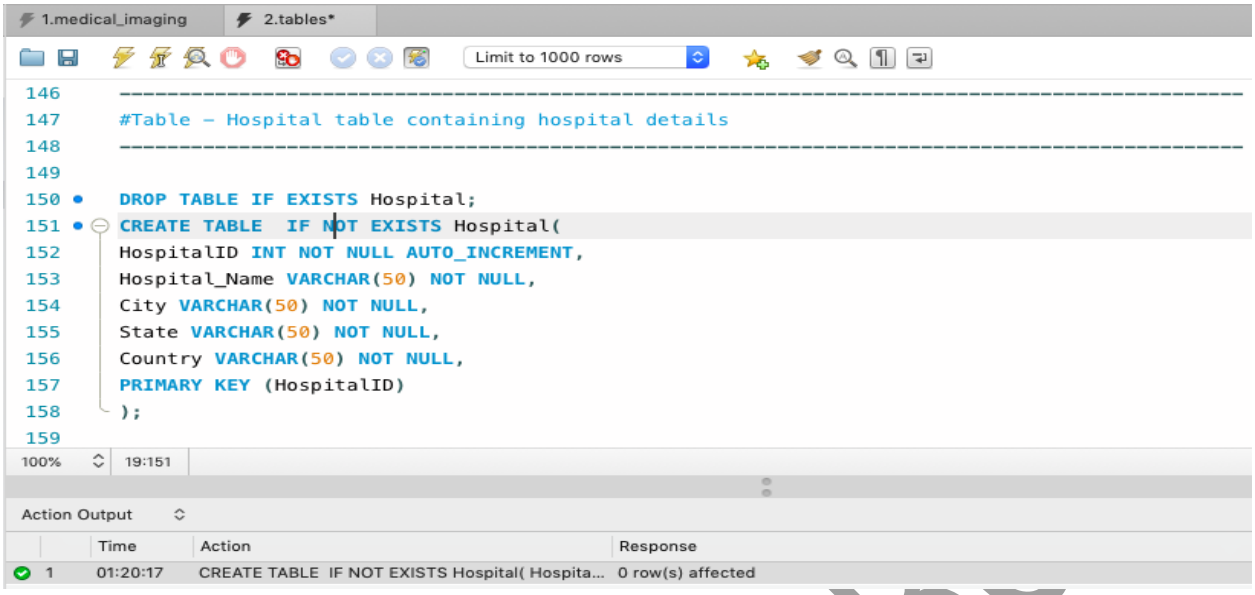
100% 2:149

Action Output

	Time	Action	Response
1	14:46:57	CREATE TABLE IF NOT EXISTS User_Role(User_RoleID INT NOT NULL AUTO_INC...	0 row(s) affected
2	14:47:00	ALTER TABLE User_Role ADD CONSTRAINT fk_role_ur FOREIGN KEY(RoleID) REF...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
3	14:47:06	ALTER TABLE User_Role ADD CONSTRAINT fk_user_ur FOREIGN KEY(UserID) REF...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

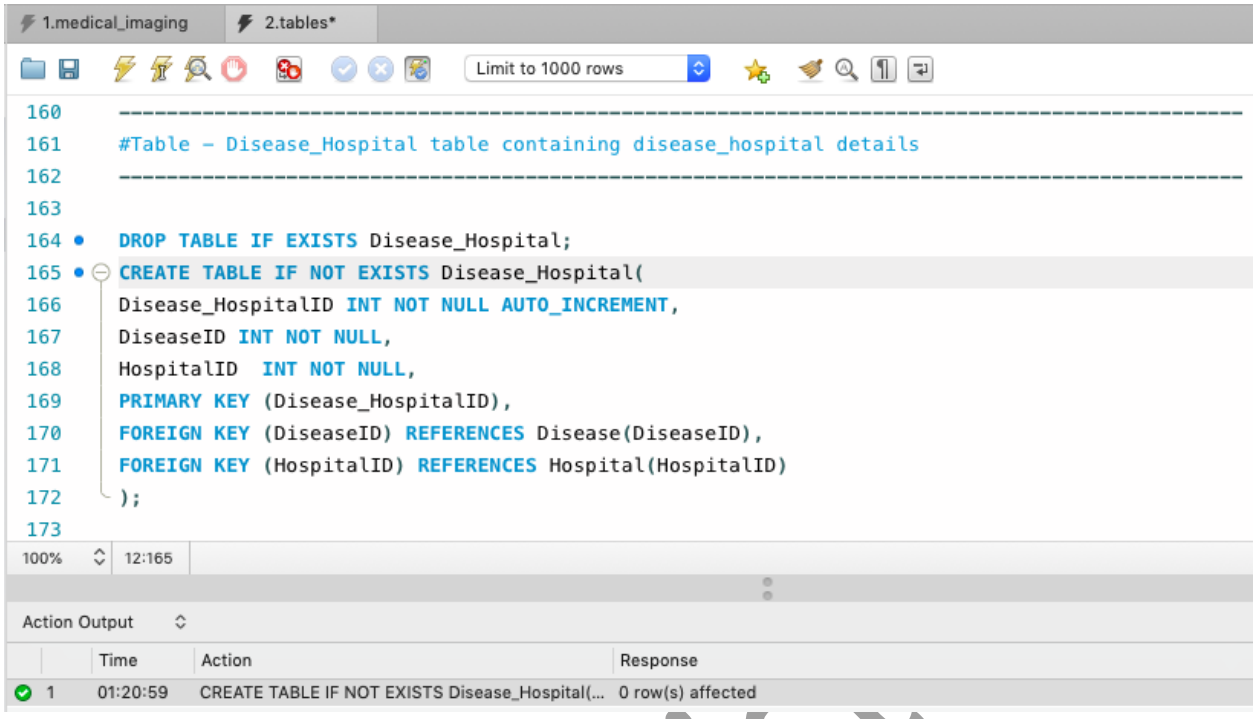
12. Hospital: This table has a list of best available hospitals. At the time of creation, we used AUTO_INCREMENT, NOT NULL, and PRIMARY KEY constraints.

```
----------------------------------------------------------------------------------------------
#Table - Hospital table containing hospital details
----------------------------------------------------------------------------------------------

DROP TABLE IF EXISTS Hospital;
CREATE TABLE IF NOT EXISTS Hospital (
HospitalID INT NOT NULL AUTO_INCREMENT,
Hospital_Name VARCHAR (50) NOT NULL,
City VARCHAR (50) NOT NULL,
State VARCHAR (50) NOT NULL,
Country VARCHAR (50) NOT NULL,
PRIMARY KEY (HospitalID));
```



13. Disease_Hospital: This table gives the hospital information for a particular disease in the database. This table includes one column from the Disease table and another column from the Hospital table. We used FOREIGN KEY constraint here for DiseaseID and HospitalID and also used AUTO_INCREMENT and NOT NULL constraints.

```
-----------------------------------------------------------------------------------------
#Table - Disease_Hospital table containing disease_hospital details
-----------------------------------------------------------------------------------------

DROP TABLE IF EXISTS Disease_Hospital;
CREATE TABLE IF NOT EXISTS Disease_Hospital(
Disease_HospitalID INT NOT NULL AUTO_INCREMENT,
DiseaseID INT NOT NULL,
HospitalID INT NOT NULL,
PRIMARY KEY (Disease_HospitalID),
FOREIGN KEY (DiseaseID) REFERENCES Disease (DiseaseID),
FOREIGN KEY (HospitalID) REFERENCES Hospital (HospitalID)
);
```



14. CT_Scan_Image: This table has listed all the images detected by CT Scan. At the time of table creation, we used AUTO_INCREMENT, NOT NULL, NULL DEFAULT NULL, and PRIMARY KEY constraints.

```
----------
#Table - CT_Scan_Image table containing CT Scan image details
----------

DROP TABLE IF EXISTS CT_Scan_Image;
CREATE TABLE IF NOT EXISTS CT_Scan_Image(
CT_Scan_Image INT NOT NULL AUTO_INCREMENT,
Image BLOB NOT NULL,
Imaging_TechniqueID INT NULL DEFAULT NULL,
PRIMARY KEY (CT_Scan_Image),
FOREIGN KEY (Imaging_TechniqueID) REFERENCES
Imaging_Technique(Imaging_TechniqueID)

);
```

```
182  -----------------------------------------------------------------------------------------------
183  #Table - CT_Scan_Image table containing CT Scan image details
184  -----------------------------------------------------------------------------------------------
185
186  DROP TABLE IF EXISTS CT_Scan_Image;
187  CREATE TABLE IF NOT EXISTS CT_Scan_Image(
188  CT_Scan_Image INT NOT NULL AUTO_INCREMENT,
189  Image BLOB NOT NULL,
190  Imaging_TechniqueID INT NULL DEFAULT NULL,
191  PRIMARY KEY (CT_Scan_Image),
192  FOREIGN KEY (Imaging_TechniqueID) REFERENCES Imaging_Technique(Imaging_TechniqueID)
193
194  );
195
```

100% 15:187

Action Output

	Time	Action	Response
1	20:13:51	CREATE TABLE IF NOT EXISTS CT_Scan_Image(CT_Scan_Image INT NOT N...	0 row(s) affected

15. MRI_Image: This table has listed all the images detected by MRI. At the time of table creation, we used AUTO_INCREMENT, NOT NULL, NULL DEFAULT NULL, and PRIMARY KEY constraints.

```
-----------------------------------------------------------------------------------------------
#Table - MRI_Image table containing MRI image details
-----------------------------------------------------------------------------------------------

DROP TABLE IF EXISTS MRI_Image;
CREATE TABLE IF NOT EXISTS MRI_Image(
MRI_ImageID INT NOT NULL AUTO_INCREMENT,
Image BLOB NOT NULL,
Imaging_TechniqueID INT NULL DEFAULT NULL,
PRIMARY KEY (MRI_ImageID),
FOREIGN KEY (Imaging_TechniqueID) REFERENCES
Imaging_Technique(Imaging_TechniqueID)
);
```

```
196     -----------------------------------------------------------------------------------------------
197     #Table - MRI_Image table containing MRI image details
198     -----------------------------------------------------------------------------------------------
199
200 •   DROP TABLE IF EXISTS MRI_Image;
201 •   CREATE TABLE IF NOT EXISTS MRI_Image(
202     MRI_ImageID INT NOT NULL AUTO_INCREMENT,
203     Image BLOB NOT NULL,
204     Imaging_TechniqueID INT NULL DEFAULT NULL,
205     PRIMARY KEY (MRI_ImageID),
206     FOREIGN KEY (Imaging_TechniqueID) REFERENCES Imaging_Technique(Imaging_TechniqueID)
207     );
208
```

100% 17:201

Action Output

	Time	Action	Response
1	20:14:27	CREATE TABLE IF NOT EXISTS MRI_Image(MRI_ImageID INT NOT NULL AU...	0 row(s) affected

16. XRay_Image: This table has listed all the images detected by X-Ray. At the time of table creation, we used AUTO_INCREMENT, NOT NULL, NULL DEFAULT NULL, and PRIMARY KEY constraints.

```
-----------------------------------------------------------------------------------------------
#Table - XRay_Image table containing Xray image details
-----------------------------------------------------------------------------------------------

DROP TABLE IF EXISTS XRay_Image;
CREATE TABLE IF NOT EXISTS XRay_Image(
XRay_Image INT NOT NULL AUTO_INCREMENT,
Image BLOB NOT NULL,
Imaging_TechniqueID INT NULL DEFAULT NULL,
PRIMARY KEY (XRay_Image),
FOREIGN KEY (Imaging_TechniqueID) REFERENCES
Imaging_Technique(Imaging_TechniqueID)
);
```

```
209     -----------------------------------------------------------------------------------------------
210     #Table - XRay_Image table containing Xray image details
211     -----------------------------------------------------------------------------------------------
212
213 •   DROP TABLE IF EXISTS XRay_Image;
214 •   CREATE TABLE IF NOT EXISTS XRay_Image(
215     XRay_Image INT NOT NULL AUTO_INCREMENT,
216     Image BLOB NOT NULL,
217     Imaging_TechniqueID INT NULL DEFAULT NULL,
218     PRIMARY KEY (XRay_Image),
219     FOREIGN KEY (Imaging_TechniqueID) REFERENCES Imaging_Technique(Imaging_TechniqueID)
```

100% 16:214

Action Output

	Time	Action	Response
1	20:18:16	CREATE TABLE IF NOT EXISTS XRay_Image(XRay_Image INT NOT NULL AU...	0 row(s) affected

17. Result:

This table contains the result from the machine model after performing the query. Here we created ResultID as a PRIMARY KEY and we also used other constraint NOT NULL. Column of tis tables gives the accuracy of machine model.

```
-----------------------------------------------------------------------------------------
#Table - Result table containing result from machine learning model details
-----------------------------------------------------------------------------------------

DROP TABLE IF EXISTS Result;
CREATE TABLE IF NOT EXISTS Result(
ResultID INT NOT NULL AUTO_INCREMENT,
Image BLOB,
Body_Part_Name VARCHAR (50) NOT NULL,
Imaging_Technique_Name VARCHAR (50) NOT NULL,
Accuracy_By_Body_Part VARCHAR (50) NOT NULL,
Accuracy_By_Technique VARCHAR (50) NOT NULL,
PRIMARY KEY (ResultID)
);
```

18. Description:

This table have the description of all imaging technique performance on the body part and how it treats the disease. Here DescriptionID is the primary key. If we don't have any description about any image or bodypart then it will give No description by default.

```
-----------------------------------------------------------------------------------------
#Table - Description table containing technique work on bodypart details
-----------------------------------------------------------------------------------------

DROP TABLE IF EXISTS Description;
CREATE TABLE IF NOT EXISTS Description (
DescriptionID INT NOT NULL,
Description VARCHAR (1000) NOT NULL DEFAULT 'No Description',
Body_Part VARCHAR (50) NOT NULL,
Technique_Name VARCHAR (50) NOT NULL,
Body_PartID INT NOT NULL,
Imaging_TechniqueID INT NOT NULL,
FOREIGN KEY (Body_PartID) REFERENCES Body_Part(Body_PartID),
FOREIGN KEY (Imaging_TechniqueID) REFERENCES
Imaging_Technique(Imaging_TechniqueID)
```

);

Alter Table:

```
ALTER TABLE `Medical_Imaging`.`User_Login`
CHANGE COLUMN `Password` `Password` BLOB NOT NULL ;
```

ALTER TABLE User MODIFY Phone_No VARCHAR(50) NULL DEFAULT NULL;

ALTER TABLE User MODIFY Sex ENUM ('Decline to respond', 'Male', 'Female', 'Transgender') NULL DEFAULT NULL;

UPDATE User

SET First_Name = 'Nancy', Last_Name = 'Gyle'

WHERE UserID = 11;

DELETE FROM Disease

WHERE DiseaseID = 31;

CONSTRAINTS:

The MySQL constraint is used to define what values can be stored in columns. The purpose of using constraints is to enforce the integrity of a database. It can be classified into two types - column level and table level. The column level constraints can apply only to one column whereas table level constraints are applied to the entire table. The constraint is declared at the time of creating a table. We used the following constraints:

1. NOT NULL
2. CHECK
3. UNIQUE
4. DEFAULT
5. AUTO_INCREMENT
6. PRIMARY KEY
7. FOREIGN KEY

TRIGGER:

The MySQL trigger is a database object that is associated with a table. It will be activated automatically when a specified change operation is performed on a specified table. Triggers are useful for tasks such as enforcing business rules, validating input data, and keeping an audit trail. We created triggers to manage and monitor tables during insert, update or delete.

Advantage of Trigger-

- Triggers are used to check the integrity of data.
- Triggers handle errors from the database layer.
- Triggers give another way to run scheduled tasks. By using triggers, we don't have to wait for the scheduled events to run because the triggers are invoked automatically before or after a change is made to the data in a table.
- Triggers can be useful for auditing the data changes in tables.

1. Address_BEFORE_INSERT

```
DROP TRIGGER IF EXISTS `Medical_Imaging`.`Address_BEFORE_INSERT`;

DELIMITER $$
USE `Medical_Imaging`$$
CREATE DEFINER = CURRENT_USER TRIGGER `Medical_Imaging`.`Address_BEFORE_INSERT`
BEFORE INSERT ON `Address`
FOR EACH ROW
BEGIN
IF NEW.AddressID = '' THEN
SIGNAL SQLSTATE'45000';
END IF;
END$$
DELIMITER ;
```

2. Body_Part_BEFORE_INSERT

```
DROP TRIGGER IF EXISTS `Medical_Imaging`.`Body_Part_BEFORE_INSERT`;

DELIMITER $$
USE `Medical_Imaging`$$
CREATE DEFINER = CURRENT_USER TRIGGER `Medical_Imaging`.`Body_Part_BEFORE_INSERT`
BEFORE INSERT ON `Body_Part`
FOR EACH ROW
BEGIN
IF NEW.Body_partID = '' THEN
SIGNAL SQLSTATE'45000';
END IF;
END$$
DELIMITER ;
```

3. Body_Part_Technique_BEFORE_INSERT

```
DROP TRIGGER IF EXISTS `Medical_Imaging`.`Body_Part_Technique_BEFORE_INSERT`;

DELIMITER $$
USE `Medical_Imaging`$$
CREATE DEFINER = CURRENT_USER TRIGGER `Medical_Imaging`.`Body_Part_Technique_BEFORE_INSERT`
BEFORE INSERT ON `Body_Part_Technique`
FOR EACH ROW
BEGIN
IF NEW.Body_PartID = '' THEN
SIGNAL SQLSTATE '45000';
END IF;
END$$
DELIMITER ;
```

4. Disease_BEFORE_INSERT

```
DROP TRIGGER IF EXISTS `Medical_Imaging`.`Disease_BEFORE_INSERT`;

DELIMITER $$
USE `Medical_Imaging`$$
CREATE DEFINER = CURRENT_USER TRIGGER `Medical_Imaging`.`Disease_BEFORE_INSERT`
BEFORE INSERT ON `Disease`
FOR EACH ROW
BEGIN
IF NEW.DiseaseID = '' THEN
 SET NEW.DiseaseID = NULL;
 ELSEIF
 NEW.Disease_Name = '' THEN
 SET NEW.Disease_Name = NULL;
 END IF;
END$$
DELIMITER ;
```

5. Disease_Hospital_BEFORE_INSERT

```
DROP TRIGGER IF EXISTS `Medical_Imaging`.`Disease_Hospital_BEFORE_INSERT`;

DELIMITER $$
USE `Medical_Imaging`$$
CREATE DEFINER = CURRENT_USER TRIGGER `Medical_Imaging`.`Disease_Hospital_BEFORE_INSERT`
BEFORE INSERT ON `Disease_Hospital`
FOR EACH ROW
BEGIN
IF NEW.DiseaseID = '' THEN
SIGNAL SQLSTATE '45000';
END IF;
END$$
DELIMITER ;
```

6. Hospital_BEFORE_INSERT

```
DROP TRIGGER IF EXISTS `Medical_Imaging`.`Hospital_BEFORE_INSERT`;

DELIMITER $$
USE `Medical_Imaging`$$
CREATE DEFINER = CURRENT_USER TRIGGER `Medical_Imaging`.`Hospital_BEFORE_INSERT`
BEFORE INSERT ON `Hospital`
FOR EACH ROW
BEGIN
IF NEW.HospitalID = '' THEN
SIGNAL SQLSTATE'45000';
END IF;
END$$
DELIMITER ;
```

7. Imaging_Technique_BEFORE_INSERT

```
DROP TRIGGER IF EXISTS `Medical_Imaging`.`Imaging_Technique_BEFORE_INSERT`;

DELIMITER $$
USE `Medical_Imaging`$$
CREATE DEFINER = CURRENT_USER TRIGGER `Medical_Imaging`.`Imaging_Technique_BEFORE_INSERT`
BEFORE INSERT ON `Imaging_Technique`
FOR EACH ROW
BEGIN
IF NEW.Imaging_TechniqueID = '' THEN
SIGNAL SQLSTATE'45000';
END IF;
END$$
DELIMITER ;
```

8. Role_BEFORE_INSERT

```
DROP TRIGGER IF EXISTS `Medical_Imaging`.`Role_BEFORE_INSERT`;

DELIMITER $$
USE `Medical_Imaging`$$
CREATE DEFINER = CURRENT_USER TRIGGER `Medical_Imaging`.`Role_BEFORE_INSERT`
BEFORE INSERT ON `Role`
FOR EACH ROW
BEGIN
IF NEW.RoleID = '' THEN
SIGNAL SQLSTATE '45000';
END IF;
END$$
DELIMITER ;
```

9. User_BEFORE_INSERT

```
DROP TRIGGER IF EXISTS `Medical_Imaging`.`User_BEFORE_INSERT`;

DELIMITER $$
USE `Medical_Imaging`$$
CREATE DEFINER = CURRENT_USER TRIGGER `Medical_Imaging`.`User_BEFORE_INSERT`
BEFORE INSERT ON `User`
FOR EACH ROW
BEGIN
IF NEW.UserID = '' THEN
SIGNAL SQLSTATE'45000';
END IF;
END$$
DELIMITER ;
```

10. User_BEFORE_INSERT_NULL

```
DROP TRIGGER IF EXISTS `Medical_Imaging`.`User_BEFORE_INSERT_NULL`;

DELIMITER $$
USE `Medical_Imaging`$$
CREATE DEFINER = CURRENT_USER TRIGGER `Medical_Imaging`.`User_BEFORE_INSERT_NULL`
BEFORE INSERT ON `User`
FOR EACH ROW
BEGIN
 IF NEW.Phone_No = '' THEN
 SET NEW.Phone_No = NULL;
 ELSEIF
 NEW.Email = '' THEN
 SET NEW.Email = NULL;
 END IF;
END$$
DELIMITER ;
```

11. User_Login_BEFORE_INSERT

```
DROP TRIGGER IF EXISTS `Medical_Imaging`.`User_Login_BEFORE_INSERT`;

DELIMITER $$
USE `Medical_Imaging`$$
CREATE DEFINER = CURRENT_USER TRIGGER `Medical_Imaging`.`User_Login_BEFORE_INSERT`
BEFORE INSERT ON `User_Login`
FOR EACH ROW
BEGIN
IF NEW.UserID = '' THEN
SIGNAL SQLSTATE '45000';
END IF;
END$$
DELIMITER ;
```

12. insert_user_vw (AFTER_INSERT_ON Trigger)

```
1   DROP TRIGGER IF EXISTS `Medical_Imaging`.`insert_user_vw`;
2
3   DELIMITER $$
4   USE `Medical_Imaging`$$
5   CREATE DEFINER = CURRENT_USER TRIGGER `Medical_Imaging`.`insert_user_vw`
6   AFTER INSERT ON `User_Role`
7   FOR EACH ROW
8   BEGIN
9       INSERT INTO user_with_instructor_role_vw
10      (First_Name, Last_Name, Age, Sex, Email, Phone_No, Role_Name)
11      SELECT u.First_Name, u.Last_Name, u.Age, u.Sex, u.Email, u.Phone_No, r.Role_Name
12      FROM User u
13      INNER JOIN User_Role ur
14      INNER JOIN Role r
15      WHERE u.USerID = ur.USerID
16      AND ur.RoleID = r.RoleID
17      AND Role_Name = 'Instructor';
18  END$$
19  DELIMITER ;
```

VIEW:

The MySQL view is a simple to select statement that gets the inner join result and the view is always going to be up to date whenever we run the view statement. We can also update the table through the view. If we update any value in a column in the view, then it automatically updates the table. So, view not only gives us the latest data, but it also allows us to put data in.
There are some restrictions when we update the view, updateable the view can't include:

- Aggregate functions,
- GROUP BY clause,
- HAVING Clause,
- UNION Clause,
- DISTINCT,
- LEFT and RIGHT JOIN
- Subqueries

For a view to be updatable, there must be a one-to-one relationship between the rows in the view and the rows in the underlying table. If we change in the structure of the table then it breaks the view.

<u>View 1:</u>

1. The following view gives the list of all diseases that can be detected by MRI imaging technique.

```
USE `Medical_Imaging`;
CREATE  OR REPLACE VIEW `disease_mri` AS
SELECT D.DiseaseID, D.Disease_Name, I.Imaging_Technique_Name
FROM Imaging_Technique AS I, Disease_Detection AS DD, Disease AS D
WHERE DD.Imaging_TechniqueID = I.Imaging_TechniqueID
AND DD.DiseaseID = D.DiseaseID
AND Imaging_Technique_Name = 'MRI';;
```

```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `disease_mri` AS
    SELECT
        `D`.`DiseaseID` AS `DiseaseID`,
        `D`.`Disease_Name` AS `Disease_Name`,
        `I`.`Imaging_Technique_Name` AS `Imaging_Technique_Name`
    FROM
        ((`imaging_technique` `I`
        JOIN `disease_detection` `DD`)
        JOIN `disease` `D`)
    WHERE
        ((`DD`.`Imaging_TechniqueID` = `I`.`Imaging_TechniqueID`)
            AND (`DD`.`DiseaseID` = `D`.`DiseaseID`)
            AND (`I`.`Imaging_Technique_Name` = 'MRI'))
```

This following query gives the result from the view ‘disease_mri’

```
SELECT * FROM disease_mri;
```

100% 27:1

Result Grid Filter Rows: Search Export:

DiseaseID	Disease_Name	Imaging_Technique_Name
3	Coma	MRI
4	Alzheimers disease	MRI
5	Brain tumor	MRI
6	Multiple sclerosis	MRI
8	Venous malformations	MRI
10	Subdural hematoma	MRI
11	Degenerative diseases	MRI
12	Hypoxic encephalopathy	MRI
13	Encephalomyelitis	MRI

disease_mri 1

Action Output

	Time	Action	Response
1	20:30:20	SELECT * FROM disease_mri LIMIT 0, 1000	22 row(s) returned

View 2:

2. The following view gives the list of all diseases that can be detected by CT_Scan imaging technique.

```
USE `Medical_Imaging`;
CREATE  OR REPLACE VIEW `disease_ctscan` AS
SELECT D.DiseaseID, D.Disease_Name, I.Imaging_Technique_Name
FROM Imaging_Technique AS I, Disease_Detection AS DD, Disease AS D
WHERE DD.Imaging_TechniqueID = I.Imaging_TechniqueID
AND DD.DiseaseID = D.DiseaseID
AND Imaging_Technique_Name = 'CT Scan';;
```

View 3:

3. The following view gives the list of all diseases that can be detected by X-Ray imaging technique.

```
USE `Medical_Imaging`;
CREATE  OR REPLACE VIEW `disease_xray` AS
SELECT D.DiseaseID, D.Disease_Name, I.Imaging_Technique_Name
FROM Imaging_Technique I, Disease_Detection DD, Disease D
WHERE DD.Imaging_TechniqueID = I.Imaging_TechniqueID
AND DD.DiseaseID = D.DiseaseID
AND Imaging_Technique_Name = 'X-Ray';;
```

View 4:

4. The following view list all the user information who works as an instructor.

```
USE `Medical_Imaging`;
CREATE  OR REPLACE VIEW `user_with_instructor_role_vw` AS
SELECT u.First_Name, u.Last_Name, u.Age, u.Sex, u.Email, u.Phone_No, r.Role_Name
FROM User u
INNER JOIN User_Role ur
INNER JOIN Role r
WHERE u.UserID = ur.UserID
AND ur.RoleID = r.RoleID
AND r.Role_Name = 'Instructor';;
```

STORED PROCEDURE:

MySQL stored the procedure is an executable database object that contains one or more SQL statements. A procedure has a name, a parameter list, and SQL statements.
Advantage of the stored procedure-

- Stored procedures are fast. So, if we have a repetitive task that requires checking, looping, multiple statements, and no user interaction, we can do it with a single call to a procedure that's stored on the server.
- Stored procedures are portable, and it will write in SQL. So, we can run on every platform that MySQL runs on, without obliging you to install an additional runtime-environment package or set permissions for program execution in the operating system.
- Stored procedures can restrict and control access to a database. In this way, it can prevent both accidental errors and malicious damage in the database.

Here we created following stored procedure:

STORED PROCEDURE 1:

1. This stored procedure is to verify user login in the database. If users' login credential is matched with the database, it shows 0 as a result otherwise 1 as a result.

```
USE `Medical_Imaging`;
DROP procedure IF EXISTS `verify_user_login_proc`;

DELIMITER $$
USE `Medical_Imaging`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `verify_user_login_proc`
(IN user_name VARCHAR(20), IN password VARCHAR(20))
BEGIN
    SELECT IF(count(*) > 0, 0, 1) AS Result
    FROM user_login
    WHERE Username = user_name
    AND Password = password;
END$$

DELIMITER ;
```

STORED PROCEDURE 2:

2. Stored procedure add_user is created to add a new user to the table. Once the user inserts all the fields for the user table in the database, this procedure shows the success message.

```
USE `Medical_Imaging`;
DROP procedure IF EXISTS `add_user`;

DELIMITER $$
USE `Medical_Imaging`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `add_user`(IN userid INT, IN fname VARCHAR(50),
BEGIN
DECLARE message_text VARCHAR(50);
INSERT INTO User (UserID, First_Name, Last_Name, Age, Sex, Email, Phone_No, AddressID)
SELECT @userid, @fname, @lname, @age, @sex, @emailid, @phone, @addressid;
SET @message_text = 'User added successfully';
SELECT @message_text;
##LIMIT 1
COMMIT;
END$$

DELIMITER ;
```

Here we inserting details to create a new user in the table:

```
SET FOREIGN_KEY_CHECKS = 0;
set @userid = 15, @fname = 'Nidhi', @lname = 'Goyal', @age = 24, @sex = 'Female', @emailid = 'goyal.ni@husky.neu.edu',
@phone = '(201)626-0670', @addressid = 21 ;
CALL add_user (@userid, @fname, @lname, @age, @sex, @emailid, @phone, @addressid);
```

100% 83:14

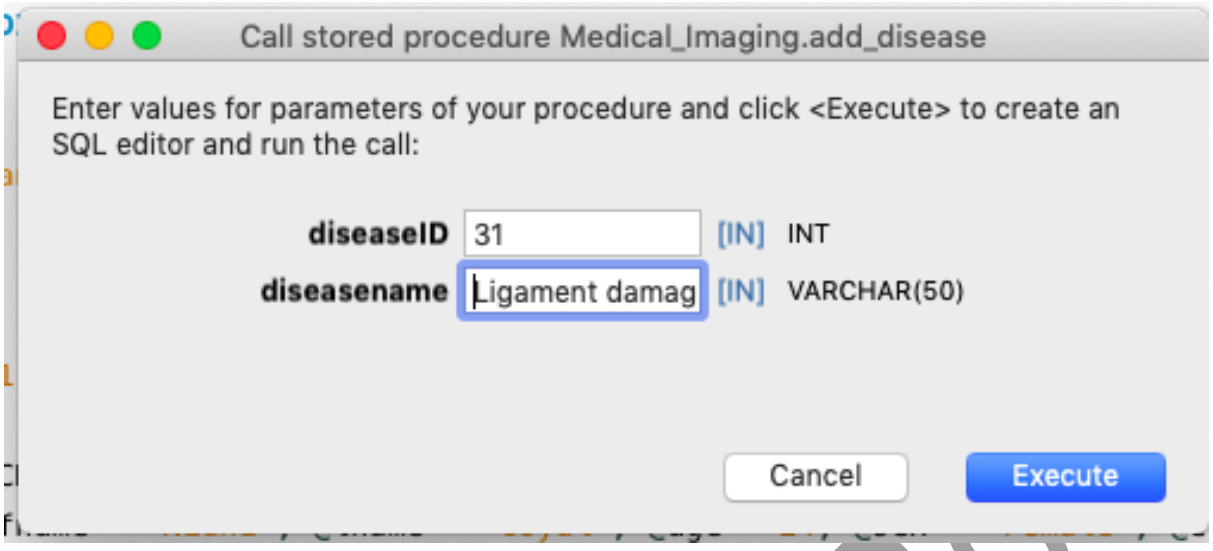
Result Grid Filter Rows: Search Export:

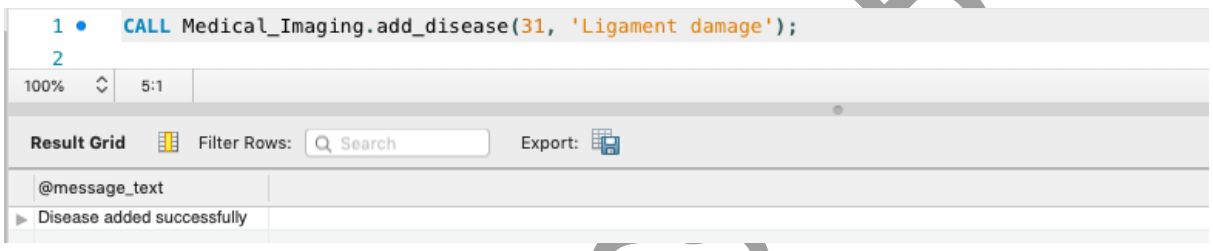
@message_text
User added successfully

STORED PROCEDURE 3:

3. This stored procedure is to use inserting value in the Disease table in the database.

```
USE `Medical_Imaging`;
DROP procedure IF EXISTS `add_disease`;

DELIMITER $$
USE `Medical_Imaging`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `add_disease`(IN diseaseID INT, IN diseasename VARCHAR(50))
BEGIN
DECLARE message_text VARCHAR(50);
INSERT INTO Disease (DiseaseID, Disease_Name)
SELECT @diseaseid, @diseasename;
SET @message_text = 'Disease added successfully';
SELECT @message_text;
##LIMIT 1

END$$

DELIMITER ;
```





STORED PROCEDURE 4:

4. When we want add new body part in the database, we can use add_body_part stored procedure. We only need to insert values for all fields.

```
USE `Medical_Imaging`;
DROP procedure IF EXISTS `add_body_part`;

DELIMITER $$
USE `Medical_Imaging`$$
CREATE PROCEDURE `add_body_part` (IN bodypartID INT, IN bodypartname VARCHAR(50))
BEGIN
DECLARE message_text VARCHAR(50);
INSERT INTO Body_Part (Body_PartID, Body_Part_Name)
SELECT @bodypartID, @bodypartname;
SET @message_text = 'New BodyPart added successfully';
SELECT @message_text;
##LIMIT 1
END$$

DELIMITER ;
```

STORED PROCEDURE 5:

5. This stored procedure is to use inserting value in the Disease table in the database.

```
USE `Medical_Imaging`;
DROP procedure IF EXISTS `add_imaging_technique`;

DELIMITER $$
USE `Medical_Imaging`$$
CREATE PROCEDURE `add_imaging_technique` (IN techniqueID INT, IN techniquename VARCH
BEGIN
DECLARE message_text VARCHAR(50);
INSERT INTO Imaging_Technique (Imaging_TechniqueID, Imaging_Technique_Name)
SELECT @techniqueID, @techniquename;
SET @message_text = 'New Technique added successfully';
SELECT @message_text;
##LIMIT 1
END$$

DELIMITER ;
```

STORED PROCEDURE 6:

6. By using this procedure we can easily search techniques for a particular disease.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `search_technique_for_disease`(IN diseasename VARCHAR(50))
BEGIN
SELECT Imaging_Technique_Name
FROM Imaging_Technique I
WHERE Imaging_TechniqueID IN (
    SELECT Imaging_TechniqueID
    FROM Disease_Detection DD
    JOIN Disease D
    WHERE I.Imaging_TechniqueID = DD.Imaging_TechniqueID
    AND DD.DiseaseID = D.DiseaseID
    AND Disease_Name = diseasename);
END
```

STORED FUNCTION:

MySQL stored function is a set of SQL statements that perform some operation and return a single value. It's an in-built function in MySQL, it can be called from within a MySQL statement. This helps improve the readability and maintainability of the procedural code.

Here we created following stored functions:

STORED FUNCTION 1:

1. This stored function performs on different imaging techniques in the database and gives the technique performance. We have Disease and Imaging_Technique table. By joining both the table we get the Disease_Detection table which takes DiseaseID and

Imaging_TechniqueID. With the help of this table, we count the total number of diseases detected from different techniques and trying to understand which technique is more works on different diseases.

```
CREATE DEFINER=`root`@`localhost` FUNCTION `technique_performance`() RETURNS varchar(50) CHARSET utf8mb4
    READS SQL DATA DETERMINISTIC
Begin
 DECLARE stringValue VARCHAR(50);
 DECLARE Total__By_MRI INT;
 DECLARE Total__By_CT_SCAN INT;
 DECLARE Total__By_XRAY INT;

 SET Total__By_MRI = (SELECT COUNT(d.DiseaseID) FROM Disease d
    JOIN Disease_Detection dd JOIN Imaging_Technique i
    WHERE d.DiseaseID = dd.DiseaseID
    AND DD.Imaging_TechniqueID = i.Imaging_TechniqueID AND i.Imaging_Technique_Name = 'MRI');

SET Total__By_CT_SCAN = (SELECT COUNT(d.DiseaseID) FROM Disease d
    JOIN Disease_Detection dd JOIN Imaging_Technique i
    WHERE d.DiseaseID = dd.DiseaseID
    AND DD.Imaging_TechniqueID = i.Imaging_TechniqueID AND i.Imaging_Technique_Name = 'CT Scan');

SET Total__By_XRAY = (SELECT COUNT(d.DiseaseID) FROM Disease d
    JOIN Disease_Detection dd JOIN Imaging_Technique i
    WHERE d.DiseaseID = dd.DiseaseID
    AND DD.Imaging_TechniqueID = i.Imaging_TechniqueID AND i.Imaging_Technique_Name = 'X-Ray');

 IF Total__By_MRI > Total__By_CT_SCAN OR Total__By_MRI > Total__By_XRAY THEN
 SET stringValue = ('MRI detected more disease than CT Scan and X-Ray'); ELSE
 SET stringValue = ('MRI detected less disease than CT Scan and X-Ray');
 END IF;
 RETURN stringValue;
 END
```

STORED FUNCTION 2:

2. body_part_max_image

```
CREATE DEFINER=`root`@`localhost` FUNCTION `body_part_max_image`() RETURNS
varchar(50) CHARSET utf8mb4
   READS SQL DATA
   DETERMINISTIC
BEGIN
 DECLARE stringValue VARCHAR(50);
 DECLARE Total__Brain_Image INT;
```

```
 DECLARE Total__Knee_Image INT;
 DECLARE Total__Shoulder_Image INT;

 SET Total__Brain_Image = (SELECT COUNT(B.Body_PartID) FROM Body_Part B
   JOIN Body_Part_Technique BT JOIN MRI_Image I
   WHERE B.Body_PartID = BT.Body_PartID
   AND BT.Imaging_TechniqueID = I.Imaging_TechniqueID AND Body_Part_Name = 'Brain');

SET Total__Knee_Image = (SELECT COUNT(B.Body_PartID) FROM Body_Part B
   JOIN Body_Part_Technique BT JOIN CT_Scan_Image I
   WHERE B.Body_PartID = BT.Body_PartID
   AND BT.Imaging_TechniqueID = I.Imaging_TechniqueID AND Body_Part_Name = 'Knee');

SET Total__Shoulder_Image = (SELECT COUNT(B.Body_PartID) FROM Body_Part B
   JOIN Body_Part_Technique BT JOIN XRay_Image I
   WHERE B.Body_PartID = BT.Body_PartID
   AND BT.Imaging_TechniqueID = I.Imaging_TechniqueID AND Body_Part_Name =
'Shoulder');

IF (Total__Brain_Image > Total__Knee_Image) AND (Total__Brain_Image >
Total__Shoulder_Image) THEN
SET stringValue = 'BRAIN';
ELSEIF Total__Brain_Image > Total__Knee_Image THEN
SET stringValue = 'BRAIN';
ELSEIF Total__Brain_Image > Total__Shoulder_Image THEN
SET stringValue = 'BRAIN';

ELSEIF (Total__Knee_Image > Total__Brain_Image) AND (Total__Knee_Image >
Total__Shoulder_Image) THEN
SET stringValue = 'KNEE';
ELSEIF Total__Knee_Image > Total__Brain_Image THEN
SET stringValue = 'KNEE';
ELSEIF Total__Knee_Image > Total__Shoulder_Image THEN
SET stringValue = 'KNEE';

ELSEIF (Total__Shoulder_Image > Total__Brain_Image) AND (Total__Shoulder_Image >
Total__Knee_Image) THEN
SET stringValue = 'KNEE';
ELSEIF Total__Shoulder_Image > Total__Knee_Image THEN
SET stringValue = 'KNEE';
ELSEIF Total__Shoulder_Image > Total__Shoulder_Image THEN
SET stringValue = 'KNEE';
ELSE
SET stringValue = 'ALL ARE EQUAL';
```

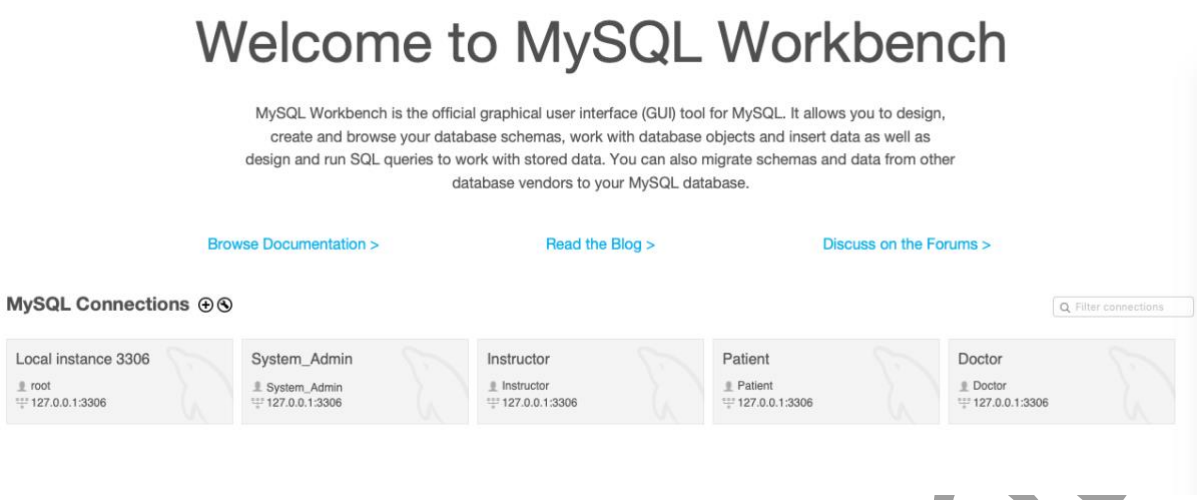
END IF;
RETURN stringValue;

END

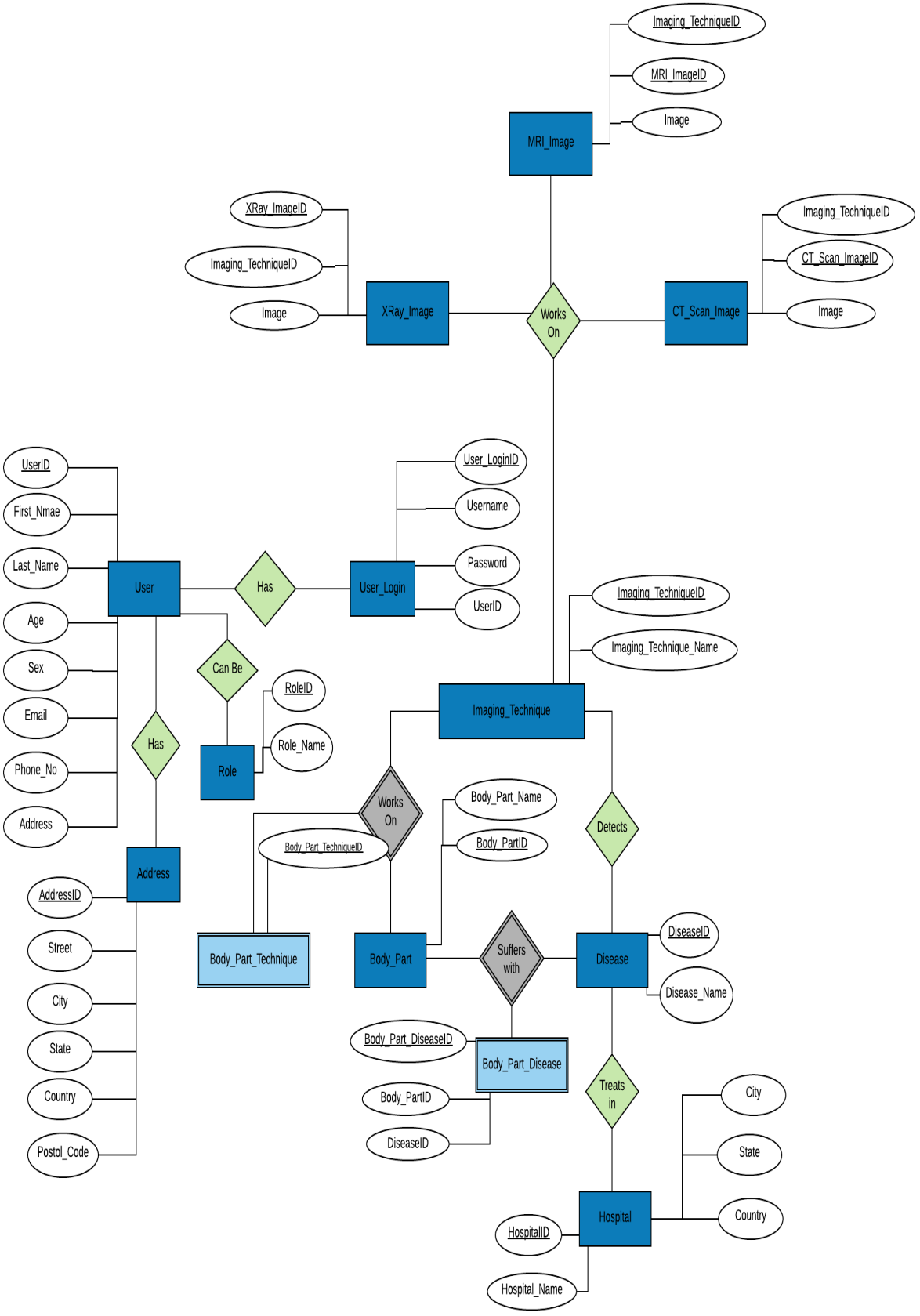
USER PRIVILEGES:

In the Medical_Imaging database, we assigned four user role which are System_Admin, Instructor, Patient, and Doctor. This section grants the user's various permission levels in order to ensure data security.

```
CREATE USER 'System_Admin' IDENTIFIED BY 'admin';

GRANT ALL ON Medical_Imaging.* TO 'System_Admin';
-------------------------------------------------------------------------------
CREATE USER 'Instructor' IDENTIFIED BY 'instructor';

GRANT Execute ON PROCEDURE Medical_Imaging.add_disease TO 'Instructor';
GRANT SELECT ON Medical_Imaging.MRI_Image TO 'Instructor';
GRANT SELECT ON Medical_Imaging.CT_Scan_Image TO 'Instructor';
GRANT SELECT ON Medical_Imaging.xray_Image TO 'Instructor';
GRANT SELECT ON Medical_Imaging.Body_Part TO 'Instructor';
GRANT SELECT ON Medical_Imaging.Imaging_Technique TO 'Instructor';
GRANT SELECT ON Medical_Imaging.Hospital TO 'Instructor';

-------------------------------------------------------------------------------
CREATE USER 'Patient' IDENTIFIED BY 'patient';

GRANT SELECT ON Medical_Imaging.MRI_Image TO 'Patient';
GRANT SELECT ON Medical_Imaging.CT_Scan_Image TO 'Patient';
GRANT SELECT ON Medical_Imaging.xray_Image TO 'Patient';
GRANT Execute ON PROCEDURE Medical_Imaging.add_user TO 'Patient';
-------------------------------------------------------------------------------
CREATE USER 'Doctor' IDENTIFIED BY 'doctor';

GRANT SELECT ON Medical_Imaging.MRI_Image TO 'Doctor';
GRANT SELECT ON Medical_Imaging.CT_Scan_Image TO 'Doctor';
GRANT SELECT ON Medical_Imaging.xray_Image TO 'Doctor';
GRANT Execute ON  PROCEDURE Medical_Imaging.add_body_part TO 'Doctor';
GRANT Execute ON  PROCEDURE Medical_Imaging.add_imaging_technique TO 'Doctor';
-------------------------------------------------------------------------------
FLUSH PRIVILEGES;
```



E-R TO RELATIONSHIP MAPPING:



Subqueries:

1. List all types of imaging techniques used in diagnose disease in human.

```
SELECT Imaging_Technique_Name
FROM Imaging_Technique I
INNER JOIN Disease_Detection DD
INNER JOIN Disease D
WHERE I. Imaging_TechniqueID = DD.Imaging_TechniqueID
AND D. DiseaseID = DD.DiseaseID
GROUP BY Imaging_Technique_Name;
```

2. List disease associated with MRI, CT Scan and X-Ray.

```
SELECT Disease_Name, Imaging_Technique_Name
FROM Imaging_Technique I
INNER JOIN Disease_Detection DD
INNER JOIN Disease D
WHERE I.Imaging_TechniqueID = DD.Imaging_TechniqueID
AND D.DiseaseID = DD.DiseaseID
AND Imaging_Technique_Name IN ('MRI', 'CT Scan', 'X-Ray');
```

3. List body part name on which imaging technique can perform.

```
SELECT Body_Part_Name
FROM Imaging_Technique I
INNER JOIN Body_Part B
INNER JOIN Body_Part_Technique BI
WHERE I.Imaging_TechniqueID = BI.Imaging_TechniqueID
AND B. Body_PartID = BI.Body_PartID
GROUP BY Body_Part_Name;
```

4. List all MRI images of brain.

```
SELECT M.Image, B.Body_Part_Name
FROM MRI_Image M
INNER JOIN Body_Part_Technique BI
INNER JOIN Body_Part B
WHERE M.Imaging_TechniqueID = BI.Imaging_TechniqueID
AND BI.Body_PartID = B.Body_PartID
AND B.Body_Part_Name = 'Brain';
```

Future expansions:

- Expanding this model for other type scans (like PET scan,) and for different body parts.
- Collaborating with a hospital to get accurate image dataset to train the model even better in-turn to provide a better health care facility

- Enhancing the features of the proposed system to enable/ predict the type and severity of the disease by providing the image of the scan.
- 3D-modeling of the disease and treatment plan.

MySQL Code:

```
-- MySQL dump 10.13  Distrib 8.0.17, for macos10.14 (x86_64)

--

-- Host: 127.0.0.1    Database: Medical_Imaging

-- ------------------------------------------------------

-- Server version         8.0.17


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;

/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;

/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;

/*!50503 SET NAMES utf8 */;

/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;

/*!40103 SET TIME_ZONE='+00:00' */;

/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;

/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;

/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;

/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;


--

-- Table structure for table `address`

--


DROP TABLE IF EXISTS `address`;

/*!40101 SET @saved_cs_client     = @@character_set_client */;

/*!50503 SET character_set_client = utf8mb4 */;
```

```
CREATE TABLE `address` (
  `AddressID` int(11) NOT NULL AUTO_INCREMENT,
  `Street` varchar(50) NOT NULL,
  `City` varchar(50) NOT NULL,
  `State` varchar(50) NOT NULL,
  `Country` varchar(50) NOT NULL DEFAULT 'USA',
  `Postal_Code` varchar(10) NOT NULL,
  PRIMARY KEY (`AddressID`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;


--
-- Dumping data for table `address`
--


LOCK TABLES `address` WRITE;
/*!40000 ALTER TABLE `address` DISABLE KEYS */;
INSERT INTO `address` VALUES (1,'2 Street','Baltimore','Maryland','USA','21201'),(2,'2
Street','Jersey','New Jersey','USA','07302'),(3,'23 Street','New York City','New
York','USA','10001'),(4,'Pike Street','Seattle','Washington','USA','05432');
/*!40000 ALTER TABLE `address` ENABLE KEYS */;
UNLOCK TABLES;


--
-- Table structure for table `body_part`
--


DROP TABLE IF EXISTS `body_part`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
```

```
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `body_part` (
  `Body_PartID` int(11) NOT NULL AUTO_INCREMENT,
  `Body_Part_Name` varchar(50) NOT NULL,
  PRIMARY KEY (`Body_PartID`)
) ENGINE=InnoDB AUTO_INCREMENT=104 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `body_part`
--

LOCK TABLES `body_part` WRITE;
/*!40000 ALTER TABLE `body_part` DISABLE KEYS */;
INSERT INTO `body_part` VALUES (4,'Heart'),(101,'Brain'),(102,'Knee'),(103,'Shoulder');
/*!40000 ALTER TABLE `body_part` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `body_part_disease`
--

DROP TABLE IF EXISTS `body_part_disease`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `body_part_disease` (
  `Body_Part_DiseaseID` int(11) NOT NULL,
  `Body_PartID` int(11) NOT NULL,
  `DiseaseID` int(11) NOT NULL,
```

```
  KEY `Body_PartID` (`Body_PartID`),
  KEY `DiseaseID` (`DiseaseID`),
  CONSTRAINT `body_part_disease_ibfk_1` FOREIGN KEY (`Body_PartID`) REFERENCES `body_part` (`Body_PartID`),
  CONSTRAINT `body_part_disease_ibfk_2` FOREIGN KEY (`DiseaseID`) REFERENCES `disease` (`DiseaseID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `body_part_disease`
--

LOCK TABLES `body_part_disease` WRITE;
/*!40000 ALTER TABLE `body_part_disease` DISABLE KEYS */;
INSERT INTO `body_part_disease` VALUES (1,101,1),(2,101,2),(3,101,3),(4,101,4),(5,101,5),(6,101,6),(7,101,7),(8,101,8),(9,101,9),(10,101,10),(11,101,11),(12,101,12),(13,101,13),(14,101,14),(15,101,15),(16,102,15),(17,103,15),(18,101,16),(19,101,17),(20,102,18),(21,103,18),(22,102,19),(23,103,19),(24,101,20),(25,102,20),(26,103,20),(27,101,23),(28,102,23),(29,103,23),(30,101,24),(31,102,24),(32,103,24),(33,101,25),(34,101,26),(35,101,28),(36,102,29),(37,103,29);
/*!40000 ALTER TABLE `body_part_disease` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `body_part_technique`
--

DROP TABLE IF EXISTS `body_part_technique`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `body_part_technique` (
```

```
  `Body_Part_TechniqueID` int(11) NOT NULL,
  `Body_PartID` int(11) NOT NULL,
  `Imaging_TechniqueID` int(11) NOT NULL,
  KEY `Body_PartID` (`Body_PartID`),
  KEY `Imaging_TechniqueID` (`Imaging_TechniqueID`),
  CONSTRAINT `body_part_technique_ibfk_1` FOREIGN KEY (`Body_PartID`) REFERENCES `body_part` (`Body_PartID`),
  CONSTRAINT `body_part_technique_ibfk_2` FOREIGN KEY (`Imaging_TechniqueID`) REFERENCES `imaging_technique` (`Imaging_TechniqueID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `body_part_technique`
--

LOCK TABLES `body_part_technique` WRITE;
/*!40000 ALTER TABLE `body_part_technique` DISABLE KEYS */;
INSERT INTO `body_part_technique` VALUES (1,101,1),(2,101,2),(3,101,7),(4,102,1),(5,102,2),(6,102,7),(7,103,1),(8,103,2),(9,103,7);
/*!40000 ALTER TABLE `body_part_technique` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `ct_scan_image`
--

DROP TABLE IF EXISTS `ct_scan_image`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
```

```
CREATE TABLE `ct_scan_image` (
  `CT_Scan_Image` int(11) NOT NULL AUTO_INCREMENT,
  `Image` blob NOT NULL,
  `Imaging_TechniqueID` int(11) DEFAULT NULL,
  PRIMARY KEY (`CT_Scan_Image`),
  KEY `Imaging_TechniqueID` (`Imaging_TechniqueID`),
  CONSTRAINT `ct_scan_image_ibfk_1` FOREIGN KEY (`Imaging_TechniqueID`) REFERENCES
`imaging_technique` (`Imaging_TechniqueID`)
) ENGINE=InnoDB AUTO_INCREMENT=176 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;


--
-- Dumping data for table `ct_scan_image`
--


LOCK TABLES `ct_scan_image` WRITE;
/*!40000 ALTER TABLE `ct_scan_image` DISABLE KEYS */;
INSERT INTO `ct_scan_image` VALUES (1,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_1.jpg',7),(2,_bin
ary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_2.jpg',7),(3,_bin
ary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_3.jpg',7),(4,_bin
ary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_4.jpg',7),(5,_bin
ary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_5.jpg',7),(6,_bin
ary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_6.jpg',7),(7,_bin
ary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_7.jpg',7),(8,_bin
ary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_8.jpg',7),(9,_bin
ary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_9.jpg',7),(10,_bi
```

nary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_10.jpg',7),(11,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_11.jpg',7),(12,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_12.jpg',7),(13,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_13.jpg',7),(14,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_14.jpg',7),(15,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_15.jpg',7),(16,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_16.jpg',7),(17,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_17.jpg',7),(18,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_18.jpg',7),(19,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_19.jpg',7),(20,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_20.jpg',7),(21,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_21.jpg',7),(22,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_22.jpg',7),(23,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_23.jpg',7),(24,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_24.jpg',7),(25,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_25.jpg',7),(26,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_26.jpg',7),(27,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_27.jpg',7),(28,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_28.jpg',7),(29,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_29.jpg',7),(30,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_30.jpg',7),(31,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_31.jpg',7),(32,_

binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_32.jpg',7),(33,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_33.jpg',7),(34,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_34.jpg',7),(35,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_35.jpg',7),(36,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_36.jpg',7),(37,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_37.jpg',7),(38,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_38.jpg',7),(39,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_39.jpg',7),(40,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_40.jpg',7),(41,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_41.jpg',7),(42,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_42.jpg',7),(43,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_43.jpg',7),(44,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_44.jpg',7),(45,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_44.jpg',7),(46,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_46.jpg',7),(47,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_47.jpg',7),(48,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_48.jpg',7),(49,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_49.jpg',7),(50,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_50.jpg',7),(51,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_51.jpg',7),(52,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_52.jpg',7),(53,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_53.jpg',7),(54,_

binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_54.jpg',7),(55,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_55.jpg',7),(56,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_56.jpg',7),(57,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_57.jpg',7),(58,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_58.jpg',7),(59,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_59.jpg',7),(60,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_60.jpg',7),(61,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_61.jpg',7),(62,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_62.jpg',7),(63,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_63.jpg',7),(64,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_64.jpg',7),(65,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_65.jpg',7),(66,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_66.jpg',7),(67,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_67.jpg',7),(68,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_68.jpg',7),(69,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_69.jpg',7),(70,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_70.jpg',7),(71,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_71.jpg',7),(72,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_72.jpg',7),(73,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_73.jpg',7),(74,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_74.jpg',7),(75,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_75.jpg',7),(76,_

binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_76.jpg',7),(77,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_77.jpg',7),(78,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_78.jpg',7),(79,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_79.jpg',7),(80,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_80.jpg',7),(81,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_81.jpg',7),(82,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_82.jpg',7),(83,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_83.jpg',7),(84,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_84.jpg',7),(85,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_85.jpg',7),(86,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_86.jpg',7),(87,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_87.jpg',7),(88,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_88.jpg',7),(89,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_89.jpg',7),(90,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_90.jpg',7),(91,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_91.jpg',7),(92,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_92.jpg',7),(93,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_93.jpg',7),(94,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_94.jpg',7),(95,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_95.jpg',7),(96,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_96.jpg',7),(97,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_97.jpg',7),(98,_

binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_98.jpg',7),(99,_
binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_99.jpg',7),(100,
_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_100.jpg',7),(101
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_101.jpg',7),(102
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_102.jpg',7),(103
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_103.jpg',7),(104
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_104.jpg',7),(105
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_105.jpg',7),(106
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_106.jpg',7),(107
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_107.jpg',7),(108
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_108.jpg',7),(109
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_109.jpg',7),(110
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_110.jpg',7),(111
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_111.jpg',7),(112
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_112.jpg',7),(113
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_113.jpg',7),(114
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_114.jpg',7),(115
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_115.jpg',7),(116
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_116.jpg',7),(117
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_117.jpg',7),(118
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_118.jpg',7),(119
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_119.jpg',7),(120

,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_120.jpg',7),(121
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_121.jpg',7),(122
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_122.jpg',7),(123
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_123.jpg',7),(124
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_124.jpg',7),(125
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_125.jpg',7),(126
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_126.jpg',7),(127
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_127.jpg',7),(128
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_128.jpg',7),(129
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_129.jpg',7),(130
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_130.jpg',7),(131
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_131.jpg',7),(132
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_132.jpg',7),(133
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_133.jpg',7),(134
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_134.jpg',7),(135
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_135.jpg',7),(136
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_136.jpg',7),(137
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_137.jpg',7),(138
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_138.jpg',7),(139
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_139.jpg',7),(140
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_140.jpg',7),(141
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_141.jpg',7),(142

,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_142.jpg',7),(143
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_143.jpg',7),(144
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_144.jpg',7),(145
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_145.jpg',7),(146
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_146.jpg',7),(147
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_147.jpg',7),(148
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_148.jpg',7),(149
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_149.jpg',7),(150
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_150.jpg',7),(151
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_151.jpg',7),(152
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_152.jpg',7),(153
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_153.jpg',7),(154
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_154.jpg',7),(155
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_155.jpg',7),(156
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_156.jpg',7),(157
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_157.jpg',7),(158
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_158.jpg',7),(159
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_159.jpg',7),(160
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_160.jpg',7),(161
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_161.jpg',7),(162
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_162.jpg',7),(163
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_163.jpg',7),(164

```
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_164.jpg',7),(165
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_165.jpg',7),(166
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_166.jpg',7),(167
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_167.jpg',7),(168
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_168.jpg',7),(169
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_169.jpg',7),(170
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_170.jpg',7),(171
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_171.jpg',7),(172
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_172.jpg',7),(173
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_173.jpg',7),(174
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_174.jpg',7),(175
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//CT//CT_175.jpg',7);

/*!40000 ALTER TABLE `ct_scan_image` ENABLE KEYS */;

UNLOCK TABLES;


--

-- Table structure for table `Description`

--


DROP TABLE IF EXISTS `Description`;

/*!40101 SET @saved_cs_client     = @@character_set_client */;

/*!50503 SET character_set_client = utf8mb4 */;

CREATE TABLE `Description` (

  `DescriptionID` int(11) NOT NULL,

  `Description` varchar(1000) NOT NULL DEFAULT 'No Description',
```

```
  `Body_Part` varchar(50) NOT NULL,
  `Technique_Name` varchar(50) NOT NULL,
  `Body_PartID` int(11) NOT NULL,
  `Imaging_TechniqueID` int(11) NOT NULL,
  KEY `Body_PartID` (`Body_PartID`),
  KEY `Imaging_TechniqueID` (`Imaging_TechniqueID`),
  CONSTRAINT `description_ibfk_1` FOREIGN KEY (`Body_PartID`) REFERENCES `body_part` (`Body_PartID`),
  CONSTRAINT `description_ibfk_2` FOREIGN KEY (`Imaging_TechniqueID`) REFERENCES `imaging_technique` (`Imaging_TechniqueID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `Description`
--

LOCK TABLES `Description` WRITE;
/*!40000 ALTER TABLE `Description` DISABLE KEYS */;
INSERT INTO `Description` VALUES (1,'Magnetic resonance imaging (MRI) of the head is a painless, \nnoninvasive test that produces detailed images of your brain and brain stem. \nAn MRI machine creates the images using a magnetic field and radio waves. \nThis test is also known as a brain MRI or a cranial MRI. \nYou will go to a hospital or radiology center to take a head MRI.','Brain','MRI',101,2),(2,'Magnetic resonance imaging (MRI) of the knee uses a powerful magnetic field, \nradio waves and a computer to produce detailed pictures of the structures within the knee joint. \nIt is typically used to help diagnose or evaluate pain, weakness, swelling or bleeding in and around the joint. \nKnee MRI does not use ionizing radiation, and it can help determine whether you require surgery.','
Knee','MRI',102,2),(3,'Magnetic resonance imaging (MRI) of the shoulder uses a powerful magnetic field, \nradio waves and a computer to produce detailed pictures of the bones, tendons, \nmuscles and blood vessels within the shoulder joint. It is primarily used to assess injuries.','Shoulder','MRI',103,2),(4,'A CT of the brain is a noninvasive diagnostic imaging procedure that uses\n special X-rays measurements to produce horizontal, or axial, images (often called slices) of the brain. \n Brain CT scans can provide more detailed information about brain tissue and brain structures than standard X-rays of the head,\n thus providing
```

more data related to injuries and/or diseases of the brain.','Brain','CT',101,7),(5,'A computed tomography (CT) scan is a type of X-ray that shows cross-sectional images of a specific area on your body. \nFor example, a CT scan of your knee would help doctors diagnose disease or inspect injuries on your knee. \nA CT scanner circles the body and sends images to a computer. The computer uses these images to make detailed pictures. \nThis allows doctors and trained technicians to see the muscles, tendons, ligaments, vessels, and bones that make up your knee.','Knee','CT',102,7),(6,'A shoulder computed tomography scan or (CT or CAT scan) creates cross-sectional images of the shoulder using specialized X-ray cameras. \nThis scan can help doctors see the bones and soft tissues in the shoulder in order to detect abnormalities. \nThe CT scan may also help identify tumors and blood clots. A CT scan can be performed with or without contrast dye. \nThe contrast material helps your doctor analyze important vessels and structures. \nIt also allows them to identify abnormalities that cannot be seen without the dye.','Shoulder','CT',103,7),(7,'A skull X-ray is an imaging test doctors use to examine the bones of the skull, \nincluding the facial bones, the nose, and the sinuses.avg an easy, quick, \nand effective method that has been used for decades to help doctors view the area that houses your most vital organ â” your brain.','Brain','Xray',101,1),(8,'A standard X-ray is a simple test in which an X-ray beam \n(a form of electromagnetic radiation) is passed through the knee to create a two-dimensional picture of the bones that form the joint. \nYour doctor can use X-rays to view bone spurs. \nBony overgrowths at the joint are a sign of osteoarthritis.','Knee','Xray',102,1),(9,'A standard X-ray is a simple test in which an X-ray beam (a form of electromagnetic radiation) \nis passed through the shoulder to create a two-dimensional picture of the bones that form the joint. \nYour doctor can use X-rays to view: joint space.','Shoulder','Xray',103,1);

/*!40000 ALTER TABLE `Description` ENABLE KEYS */;

UNLOCK TABLES;

--

-- Table structure for table `disease`

--

DROP TABLE IF EXISTS `disease`;

/*!40101 SET @saved_cs_client = @@character_set_client */;

/*!50503 SET character_set_client = utf8mb4 */;

CREATE TABLE `disease` (

`DiseaseID` int(11) NOT NULL AUTO_INCREMENT,

`Disease_Name` varchar(50) NOT NULL,

PRIMARY KEY (`DiseaseID`)

```
) ENGINE=InnoDB AUTO_INCREMENT=32 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

/*!40101 SET character_set_client = @saved_cs_client */;


--

-- Dumping data for table `disease`

--


LOCK TABLES `disease` WRITE;

/*!40000 ALTER TABLE `disease` DISABLE KEYS */;

INSERT INTO `disease` VALUES (1,'HIV infection'),(2,'Stroke'),(3,'Coma'),(4,'Alzheimers
disease'),(5,'Brain tumor'),(6,'Multiple sclerosis'),(7,'Aneurysms'),(8,'Venous
malformations'),(9,'Hemorrhage'),(10,'Subdural hematoma'),(11,'Degenerative
diseases'),(12,'Hypoxic
encephalopathy'),(13,'Encephalomyelitis'),(14,'Hydrocephalus'),(15,'Bone
injuries'),(16,'Traumatic brain injury'),(17,'Dementia'),(18,'Bone tumor'),(19,'Bone
fractures'),(20,'Cancer'),(21,'Liver masses'),(22,'Emphysema'),(23,'Internal
injuries'),(24,'Infection'),(25,'Cystic fibrosis'),(26,'Headache'),(27,'Abdominal Aortic
Aneurysm'),(28,'Epilepsy'),(29,'Arthritis'),(30,'Osteoporosis'),(31,'Ligament damage');

/*!40000 ALTER TABLE `disease` ENABLE KEYS */;

UNLOCK TABLES;


--

-- Temporary view structure for view `disease_ctscan`

--


DROP TABLE IF EXISTS `disease_ctscan`;

/*!50001 DROP VIEW IF EXISTS `disease_ctscan`*/;

SET @saved_cs_client     = @@character_set_client;

/*!50503 SET character_set_client = utf8mb4 */;

/*!50001 CREATE VIEW `disease_ctscan` AS SELECT

 1 AS `DiseaseID`,
```

```
 1 AS `Disease_Name`,
 1 AS `Imaging_Technique_Name`*/;
SET character_set_client = @saved_cs_client;


--
-- Table structure for table `disease_detection`
--


DROP TABLE IF EXISTS `disease_detection`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `disease_detection` (
  `Disease_DetectionID` int(11) NOT NULL AUTO_INCREMENT,
  `Imaging_TechniqueID` int(11) NOT NULL,
  `DiseaseID` int(11) NOT NULL,
  PRIMARY KEY (`Disease_DetectionID`),
  KEY `Imaging_TechniqueID` (`Imaging_TechniqueID`),
  KEY `DiseaseID` (`DiseaseID`),
  CONSTRAINT `disease_detection_ibfk_1` FOREIGN KEY (`Imaging_TechniqueID`)
REFERENCES `imaging_technique` (`Imaging_TechniqueID`),
  CONSTRAINT `disease_detection_ibfk_2` FOREIGN KEY (`DiseaseID`) REFERENCES `disease`
(`DiseaseID`)
) ENGINE=InnoDB AUTO_INCREMENT=52 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;


--
-- Dumping data for table `disease_detection`
--
```

```
LOCK TABLES `disease_detection` WRITE;

/*!40000 ALTER TABLE `disease_detection` DISABLE KEYS */;

INSERT INTO `disease_detection` VALUES
(1,3,1),(2,7,2),(3,1,2),(4,7,3),(5,2,3),(6,2,4),(7,9,4),(8,2,5),(9,2,6),(10,7,7),(11,7,8),(12,2,8),(13,7,9),(14,2,10),(15,2,11),(16,2,12),(17,2,13),(18,2,14),(19,4,14),(20,7,14),(21,2,15),(22,2,16),(23,7,16),(24,2,17),(25,7,17),(26,1,18),(27,2,18),(28,7,18),(29,1,19),(30,2,19),(31,7,20),(32,1,21),(33,2,21),(34,4,21),(35,7,21),(36,7,22),(37,1,23),(38,2,23),(39,7,23),(40,2,24),(41,7,24),(42,1,25),(43,2,25),(44,7,25),(45,2,26),(46,7,26),(47,4,27),(48,2,28),(49,1,29),(50,2,29),(51,1,30);

/*!40000 ALTER TABLE `disease_detection` ENABLE KEYS */;

UNLOCK TABLES;


--

-- Table structure for table `disease_hospital`

--


DROP TABLE IF EXISTS `disease_hospital`;

/*!40101 SET @saved_cs_client     = @@character_set_client */;

/*!50503 SET character_set_client = utf8mb4 */;

CREATE TABLE `disease_hospital` (

  `Disease_HospitalID` int(11) NOT NULL AUTO_INCREMENT,

  `DiseaseID` int(11) NOT NULL,

  `HospitalID` int(11) NOT NULL,

  PRIMARY KEY (`Disease_HospitalID`),

  KEY `DiseaseID` (`DiseaseID`),

  KEY `HospitalID` (`HospitalID`),

  CONSTRAINT `disease_hospital_ibfk_1` FOREIGN KEY (`DiseaseID`) REFERENCES `disease` (`DiseaseID`),

  CONSTRAINT `disease_hospital_ibfk_2` FOREIGN KEY (`HospitalID`) REFERENCES `hospital` (`HospitalID`)

) ENGINE=InnoDB AUTO_INCREMENT=126 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

/*!40101 SET character_set_client = @saved_cs_client */;
```

```
--
-- Dumping data for table `disease_hospital`
--

LOCK TABLES `disease_hospital` WRITE;
/*!40000 ALTER TABLE `disease_hospital` DISABLE KEYS */;
INSERT INTO `disease_hospital` VALUES
(111,2,1),(112,3,1),(113,4,1),(114,5,3),(115,6,4),(116,7,1),(117,8,5),(118,9,6),(119,10,7),(120,
18,1),(121,19,8),(122,20,9),(123,28,1),(124,29,6),(125,30,10);
/*!40000 ALTER TABLE `disease_hospital` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Temporary view structure for view `disease_mri`
--

DROP TABLE IF EXISTS `disease_mri`;
/*!50001 DROP VIEW IF EXISTS `disease_mri`*/;
SET @saved_cs_client     = @@character_set_client;
/*!50503 SET character_set_client = utf8mb4 */;
/*!50001 CREATE VIEW `disease_mri` AS SELECT
 1 AS `DiseaseID`,
 1 AS `Disease_Name`,
 1 AS `Imaging_Technique_Name`*/;
SET character_set_client = @saved_cs_client;

--
-- Temporary view structure for view `disease_xray`
--
```

```
DROP TABLE IF EXISTS `disease_xray`;
/*!50001 DROP VIEW IF EXISTS `disease_xray`*/;
SET @saved_cs_client     = @@character_set_client;
/*!50503 SET character_set_client = utf8mb4 */;
/*!50001 CREATE VIEW `disease_xray` AS SELECT
 1 AS `DiseaseID`,
 1 AS `Disease_Name`,
 1 AS `Imaging_Technique_Name`*/;
SET character_set_client = @saved_cs_client;


--
-- Table structure for table `hospital`
--


DROP TABLE IF EXISTS `hospital`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `hospital` (
  `HospitalID` int(11) NOT NULL AUTO_INCREMENT,
  `Hospital_Name` varchar(50) NOT NULL,
  `City` varchar(50) NOT NULL,
  `State` varchar(50) NOT NULL,
  `Country` varchar(50) NOT NULL,
  PRIMARY KEY (`HospitalID`)
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;


--
```

```
-- Dumping data for table `hospital`

--


LOCK TABLES `hospital` WRITE;

/*!40000 ALTER TABLE `hospital` DISABLE KEYS */;

INSERT INTO `hospital` VALUES (1,'Mayo Clinic
Hospital','Rochester','Minnesota','USA'),(2,'Mayo Clinic
Hospital','Jacksonville','Florida','USA'),(3,'Mayo Clinic
Hospital','Phoenix','Arizona','USA'),(4,'Cleveland Clinic Neurological
Institute','Cleveland','Ohio','USA'),(5,'Mount Sinai Hospital','New York City','New
York','USA'),(6,'The Johns Hopkins Hospital','Baltimore','Maryland','USA'),(7,'Neurological
Surgery P.C.','Rockville','New York','USA'),(8,'Arkansas Surgical Hospital','North Little
Rock','Arkansas','USA'),(9,'H. Lee Moffitt Cancer Center & Research
Institute','Tampa','Florida','USA'),(10,'Brigham and Womens
Hospital','Boston','Massachusetts','USA');

/*!40000 ALTER TABLE `hospital` ENABLE KEYS */;

UNLOCK TABLES;


--

-- Table structure for table `imaging_technique`

--


DROP TABLE IF EXISTS `imaging_technique`;

/*!40101 SET @saved_cs_client     = @@character_set_client */;

/*!50503 SET character_set_client = utf8mb4 */;

CREATE TABLE `imaging_technique` (

  `Imaging_TechniqueID` int(11) NOT NULL AUTO_INCREMENT,

  `Imaging_Technique_Name` varchar(50) NOT NULL,

  PRIMARY KEY (`Imaging_TechniqueID`)

) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

/*!40101 SET character_set_client = @saved_cs_client */;
```

```
--

-- Dumping data for table `imaging_technique`

--


LOCK TABLES `imaging_technique` WRITE;

/*!40000 ALTER TABLE `imaging_technique` DISABLE KEYS */;

INSERT INTO `imaging_technique` VALUES (1,'X-Ray'),(2,'MRI'),(3,'Nuclear
medicine'),(4,'Ultrsound'),(5,'Elastography'),(6,'Photoacoustic imaging'),(7,'CT
Scan'),(8,'Echocardiography'),(9,'Functional near-infrared spectroscopy'),(10,'Magnetic
Particle Imaging');

/*!40000 ALTER TABLE `imaging_technique` ENABLE KEYS */;

UNLOCK TABLES;


--

-- Table structure for table `mri_image`

--


DROP TABLE IF EXISTS `mri_image`;

/*!40101 SET @saved_cs_client     = @@character_set_client */;

/*!50503 SET character_set_client = utf8mb4 */;

CREATE TABLE `mri_image` (

  `MRI_ImageID` int(11) NOT NULL AUTO_INCREMENT,

  `Image` blob NOT NULL,

  `Imaging_TechniqueID` int(11) DEFAULT NULL,

  PRIMARY KEY (`MRI_ImageID`),

  KEY `Imaging_TechniqueID` (`Imaging_TechniqueID`),

  CONSTRAINT `mri_image_ibfk_1` FOREIGN KEY (`Imaging_TechniqueID`) REFERENCES
`imaging_technique` (`Imaging_TechniqueID`)

) ENGINE=InnoDB AUTO_INCREMENT=176 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

/*!40101 SET character_set_client = @saved_cs_client */;
```

--

-- Dumping data for table `mri_image`

--

LOCK TABLES `mri_image` WRITE;

/*!40000 ALTER TABLE `mri_image` DISABLE KEYS */;

INSERT INTO `mri_image` VALUES (1,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_1.jpg',2),(2,
_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_2.jpg',2),(3,
_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(4,
_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(5,
_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(6,
_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(7,
_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(8,
_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(9,
_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(1
0,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(1
1,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(1
2,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(1
3,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(1
4,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(1
5,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(1
6,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(1
7,_binary

'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(18,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(19,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(20,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(21,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(22,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(23,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(24,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(25,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(26,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(27,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(28,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(29,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(30,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(31,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(32,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(33,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(34,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(35,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(36,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(37,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(38,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(39,_binary

'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(40,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(41,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(42,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(43,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(44,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(45,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(46,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(47,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(48,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(49,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(50,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//brain//MRI//brain_3.jpg',2),(51,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_1.jpg',2),(52,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_2.jpg',2),(53,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_3.jpg',2),(54,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_4.jpg',2),(55,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_5.jpg',2),(56,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_6.jpg',2),(57,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_7.jpg',2),(58,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_8.jpg',2),(59,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_9.jpg',2),(60,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_10.jpg',2),(61,_binary

'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_11.jpg',2),(62,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_12.jpg',2),(63,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_13.jpg',2),(64,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_14.jpg',2),(65,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_15.jpg',2),(66,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_16.jpg',2),(67,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_17.jpg',2),(68,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_18.jpg',2),(69,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_19.jpg',2),(70,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_20.jpg',2),(71,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_21.jpg',2),(72,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_22.jpg',2),(73,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_23.jpg',2),(74,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_24.jpg',2),(75,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_25.jpg',2),(76,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_26.jpg',2),(77,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_27.jpg',2),(78,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_28.jpg',2),(79,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_29.jpg',2),(80,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_30.jpg',2),(81,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_31.jpg',2),(82,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_32.jpg',2),(83,_binary

'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_33.jpg',2),(84,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_34.jpg',2),(85,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_35.jpg',2),(86,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_36.jpg',2),(87,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_37.jpg',2),(89,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_38.jpg',2),(90,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_39.jpg',2),(91,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_40.jpg',2),(92,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_41.jpg',2),(93,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_42.jpg',2),(94,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_43.jpg',2),(95,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_44.jpg',2),(96,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_45.jpg',2),(97,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_46.jpg',2),(98,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_47.jpg',2),(99,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_48.jpg',2),(100,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//Knee//MRI//knee_49.jpg',2),(101,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_1.jpg',2),(102,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_2.jpg',2),(103,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_3.jpg',2),(104,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_4.jpg',2),(105,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_5.jpg',2),(106,_binary

'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_6.jpg',2),(107,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_7.jpg',2),(108,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_8.jpg',2),(109,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_9.jpg',2),(110,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_10.jpg',2),(111,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_11.jpg',2),(112,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_12.jpg',2),(113,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_13.jpg',2),(114,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_14.jpg',2),(115,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_15.jpg',2),(116,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_16.jpg',2),(117,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_17.jpg',2),(118,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_18.jpg',2),(119,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_19.jpg',2),(120,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_20.jpg',2),(121,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_21.jpg',2),(122,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_22.jpg',2),(123,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_23.jpg',2),(124,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_24.jpg',2),(125,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_25.jpg',2),(126,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_26.jpg',2),(127,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_27.jpg',2),(128,_binary

'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_28.jpg',2),(129,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_29.jpg',2),(130,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_30.jpg',2),(131,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_31.jpg',2),(132,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_32.jpg',2),(133,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_33.jpg',2),(134,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_34.jpg',2),(135,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_35.jpg',2),(136,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_36.jpg',2),(137,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_37.jpg',2),(138,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_38.jpg',2),(139,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_39.jpg',2),(140,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_40.jpg',2),(141,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_41.jpg',2),(142,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_42.jpg',2),(143,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_43.jpg',2),(144,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_44.jpg',2),(145,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_45.jpg',2),(146,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_46.jpg',2),(147,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_47.jpg',2),(148,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_48.jpg',2),(149,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_49.jpg',2),(150,_binary

'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_50.jpg',2),(151,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_51.jpg',2),(152,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_52.jpg',2),(153,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_53.jpg',2),(154,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_54.jpg',2),(155,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_55.jpg',2),(156,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_56.jpg',2),(157,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_57.jpg',2),(158,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_58.jpg',2),(159,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_59.jpg',2),(160,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_60.jpg',2),(161,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_61.jpg',2),(162,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_62.jpg',2),(163,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_63.jpg',2),(164,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_64.jpg',2),(165,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_65.jpg',2),(166,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_66.jpg',2),(167,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_67.jpg',2),(168,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_68.jpg',2),(169,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_69.jpg',2),(170,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_70.jpg',2),(171,_binary
'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_71.jpg',2),(172,_binary

'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_72.jpg',2),(173,_binary 'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_73.jpg',2),(174,_binary 'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_74.jpg',2),(175,_binary 'C://Users//nares//OneDrive//Desktop//dmdd//All_images//shoulder//MRI//shoulder_75.jpg',2);

/*!40000 ALTER TABLE `mri_image` ENABLE KEYS */;

UNLOCK TABLES;

--

-- Table structure for table `new`

--

DROP TABLE IF EXISTS `new`;

/*!40101 SET @saved_cs_client = @@character_set_client */;

/*!50503 SET character_set_client = utf8mb4 */;

CREATE TABLE `new` (

`id` int(11) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

/*!40101 SET character_set_client = @saved_cs_client */;

--

-- Dumping data for table `new`

--

LOCK TABLES `new` WRITE;

/*!40000 ALTER TABLE `new` DISABLE KEYS */;

/*!40000 ALTER TABLE `new` ENABLE KEYS */;

UNLOCK TABLES;

```
--
-- Table structure for table `Result`
--

DROP TABLE IF EXISTS `Result`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `Result` (
  `ResultID` int(11) NOT NULL AUTO_INCREMENT,
  `Image` blob,
  `Body_Part_Name` varchar(50) NOT NULL,
  `Imaging_Technique_Name` varchar(50) NOT NULL,
  `Accuracy_By_Body_Part` varchar(50) NOT NULL,
  `Accuracy_By_Technique` varchar(50) NOT NULL,
  PRIMARY KEY (`ResultID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `Result`
--

LOCK TABLES `Result` WRITE;
/*!40000 ALTER TABLE `Result` DISABLE KEYS */;
/*!40000 ALTER TABLE `Result` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `role`
```

```
--


DROP TABLE IF EXISTS `role`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `role` (
  `RoleID` int(11) NOT NULL AUTO_INCREMENT,
  `Role_Name` varchar(20) NOT NULL,
  PRIMARY KEY (`RoleID`),
  UNIQUE KEY `Role_Name` (`Role_Name`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;


--
-- Dumping data for table `role`
--


LOCK TABLES `role` WRITE;
/*!40000 ALTER TABLE `role` DISABLE KEYS */;
INSERT INTO `role` VALUES (4,'Doctor'),(2,'Instructor'),(3,'Patient'),(1,'System Admin');
/*!40000 ALTER TABLE `role` ENABLE KEYS */;
UNLOCK TABLES;


--
-- Table structure for table `user`
--


DROP TABLE IF EXISTS `user`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
```

```
/*!50503 SET character_set_client = utf8mb4 */;

CREATE TABLE `user` (

  `UserID` int(11) NOT NULL AUTO_INCREMENT,

  `First_Name` varchar(50) NOT NULL,

  `Last_Name` varchar(50) NOT NULL,

  `Age` int(11) DEFAULT NULL,

  `Sex` enum('Decline to respond','Male','Female','Transgender') DEFAULT NULL,

  `Email` varchar(50) DEFAULT NULL,

  `Phone_No` varchar(50) DEFAULT NULL,

  `AddressID` int(11) NOT NULL,

  PRIMARY KEY (`UserID`),

  KEY `AddressID` (`AddressID`),

  CONSTRAINT `user_ibfk_1` FOREIGN KEY (`AddressID`) REFERENCES `address`
(`AddressID`),

  CONSTRAINT `user_chk_1` CHECK ((`Age` > 0))

) ENGINE=InnoDB AUTO_INCREMENT=102 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

/*!40101 SET character_set_client = @saved_cs_client */;


--

-- Dumping data for table `user`

--


LOCK TABLES `user` WRITE;

/*!40000 ALTER TABLE `user` DISABLE KEYS */;

INSERT INTO `user` VALUES (11,'Nancy','Gyle',26,'Female','ng@gmail.com','(201) 456
6578',2),(12,'Sam','Crush',21,'Male','sc@gmail.com','(673) 234
1238',1),(13,'Micheal','Johnson',26,'Male','mj@gmail.com','(673) 113
1238',3),(14,'Mike','Crush',45,'Transgender','mc@gmail.com','(224) 776
8767',4),(15,'Nidhi','Goyal',24,'Female','goyal.ni@husky.neu.edu','(201)626-0670',21);

/*!40000 ALTER TABLE `user` ENABLE KEYS */;
```

```
UNLOCK TABLES;
/*!50003 SET @saved_cs_client      = @@character_set_client */ ;
/*!50003 SET @saved_cs_results     = @@character_set_results */ ;
/*!50003 SET @saved_col_connection = @@collation_connection */ ;
/*!50003 SET character_set_client  = utf8mb4 */ ;
/*!50003 SET character_set_results = utf8mb4 */ ;
/*!50003 SET collation_connection  = utf8mb4_0900_ai_ci */ ;
/*!50003 SET @saved_sql_mode       = @@sql_mode */ ;
/*!50003 SET sql_mode              =
'ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERRO
R_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;
/*!50003 CREATE*/ /*!50017 DEFINER=`root`@`localhost`*/ /*!50003 TRIGGER
`user_BEFORE_INSERT_NULL` BEFORE INSERT ON `user` FOR EACH ROW BEGIN
IF NEW.Phone_No = ' ' THEN
SET NEW.Phone_No = NULL;
ELSEIF
NEW.Email =' ' THEN
SET NEW.Email = NULL;
END IF;
END */;;
DELIMITER ;
/*!50003 SET sql_mode              = @saved_sql_mode */ ;
/*!50003 SET character_set_client  = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection  = @saved_col_connection */ ;


--
-- Table structure for table `User_Login`
--
```

```
DROP TABLE IF EXISTS `User_Login`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `User_Login` (
  `User_LoginID` int(11) NOT NULL AUTO_INCREMENT,
  `Username` varchar(20) NOT NULL DEFAULT 'No Username',
  `Password` blob NOT NULL,
  `UserID` int(11) NOT NULL,
  PRIMARY KEY (`User_LoginID`),
  KEY `UserID` (`UserID`),
  CONSTRAINT `user_login_ibfk_1` FOREIGN KEY (`UserID`) REFERENCES `user` (`UserID`)
) ENGINE=InnoDB AUTO_INCREMENT=1006 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;


--
-- Dumping data for table `User_Login`
--


LOCK TABLES `User_Login` WRITE;
/*!40000 ALTER TABLE `User_Login` DISABLE KEYS */;
INSERT INTO `User_Login` VALUES (1001,'ng@gmail.com',_binary
'ng@123',11),(1002,'sc@gmail.com',_binary 'sc@111',12),(1003,'mj@gmail.com',_binary
'mj@111',13),(1004,'mc@gmail.com',_binary
'mc@111',14),(1005,'goyal.ni@husky.neu.e',_binary '?ß»œ
´¶\Ù\ï ¤\n¢',15);
/*!40000 ALTER TABLE `User_Login` ENABLE KEYS */;
UNLOCK TABLES;
/*!50003 SET @saved_cs_client      = @@character_set_client */ ;
/*!50003 SET @saved_cs_results     = @@character_set_results */ ;
```

```
/*!50003 SET @saved_col_connection = @@collation_connection */ ;

/*!50003 SET character_set_client  = utf8mb4 */ ;

/*!50003 SET character_set_results = utf8mb4 */ ;

/*!50003 SET collation_connection  = utf8mb4_0900_ai_ci */ ;

/*!50003 SET @saved_sql_mode       = @@sql_mode */ ;

/*!50003 SET sql_mode              =
'ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERRO
R_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION' */ ;

DELIMITER ;;

/*!50003 CREATE*/ /*!50017 DEFINER=`root`@`localhost`*/ /*!50003 TRIGGER
`User_Login_BEFORE_INSERT` BEFORE INSERT ON `user_login` FOR EACH ROW BEGIN

IF NEW.UserID = ' ' THEN

SIGNAL SQLSTATE '45000';

END IF;

END */;;

DELIMITER ;

/*!50003 SET sql_mode              = @saved_sql_mode */ ;

/*!50003 SET character_set_client  = @saved_cs_client */ ;

/*!50003 SET character_set_results = @saved_cs_results */ ;

/*!50003 SET collation_connection  = @saved_col_connection */ ;


--

-- Table structure for table `user_role`

--


DROP TABLE IF EXISTS `user_role`;

/*!40101 SET @saved_cs_client     = @@character_set_client */;

/*!50503 SET character_set_client = utf8mb4 */;

CREATE TABLE `user_role` (

  `User_RoleID` int(11) NOT NULL AUTO_INCREMENT,
```

```
  `UserID` int(11) NOT NULL,
  `RoleID` int(11) NOT NULL,
  PRIMARY KEY (`User_RoleID`),
  KEY `fk_role_ur` (`RoleID`),
  KEY `fk_user_ur` (`UserID`),
  CONSTRAINT `fk_role_ur` FOREIGN KEY (`RoleID`) REFERENCES `role` (`RoleID`),
  CONSTRAINT `fk_user_ur` FOREIGN KEY (`UserID`) REFERENCES `user` (`UserID`)
) ENGINE=InnoDB AUTO_INCREMENT=105 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;


--
-- Dumping data for table `user_role`
--


LOCK TABLES `user_role` WRITE;
/*!40000 ALTER TABLE `user_role` DISABLE KEYS */;
INSERT INTO `user_role` VALUES (101,11,1),(102,13,2),(103,14,3),(104,12,4);
/*!40000 ALTER TABLE `user_role` ENABLE KEYS */;
UNLOCK TABLES;
/*!50003 SET @saved_cs_client      = @@character_set_client */ ;
/*!50003 SET @saved_cs_results     = @@character_set_results */ ;
/*!50003 SET @saved_col_connection = @@collation_connection */ ;
/*!50003 SET character_set_client  = utf8mb4 */ ;
/*!50003 SET character_set_results = utf8mb4 */ ;
/*!50003 SET collation_connection  = utf8mb4_0900_ai_ci */ ;
/*!50003 SET @saved_sql_mode       = @@sql_mode */ ;
/*!50003 SET sql_mode              =
'ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERRO
R_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION' */ ;
```

```
DELIMITER ;;
/*!50003 CREATE*/ /*!50017 DEFINER=`root`@`localhost`*/ /*!50003 TRIGGER
`insert_user` AFTER INSERT ON `user_role` FOR EACH ROW BEGIN
INSERT INTO user_with_instructor_role_vw
(First_Name, Last_Name, Age, Sex, Email, Phone_No, Role_Name)
SELECT u.First_Name, u.Last_Name, u.Age, u.Sex, u.Email, u.Phone_No, r.Role_Name
FROM User u
INNER JOIN User_Role ur
INNER JOIN Role r
WHERE u.UserID = ur.UserID
AND ur.RoleID = r.RoleID
AND r. Role_Name = 'Instructor';
END */;;
DELIMITER ;
/*!50003 SET sql_mode          = @saved_sql_mode */ ;
/*!50003 SET character_set_client  = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection  = @saved_col_connection */ ;


--
-- Temporary view structure for view `user_with_instructor_role_vw`
--


DROP TABLE IF EXISTS `user_with_instructor_role_vw`;
/*!50001 DROP VIEW IF EXISTS `user_with_instructor_role_vw`*/;
SET @saved_cs_client     = @@character_set_client;
/*!50503 SET character_set_client = utf8mb4 */;
/*!50001 CREATE VIEW `user_with_instructor_role_vw` AS SELECT
 1 AS `First_Name`,
 1 AS `Last_Name`,
```

```
 1 AS `Age`,
 1 AS `Sex`,
 1 AS `Email`,
 1 AS `Phone_No`,
 1 AS `Role_Name`*/;
SET character_set_client = @saved_cs_client;


--
-- Table structure for table `xray_image`
--


DROP TABLE IF EXISTS `xray_image`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `xray_image` (
  `XRay_Image` int(11) NOT NULL AUTO_INCREMENT,
  `Image` blob NOT NULL,
  `Imaging_TechniqueID` int(11) DEFAULT NULL,
  PRIMARY KEY (`XRay_Image`),
  KEY `Imaging_TechniqueID` (`Imaging_TechniqueID`),
  CONSTRAINT `xray_image_ibfk_1` FOREIGN KEY (`Imaging_TechniqueID`) REFERENCES
`imaging_technique` (`Imaging_TechniqueID`)
) ENGINE=InnoDB AUTO_INCREMENT=121 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;


--
-- Dumping data for table `xray_image`
--
```

LOCK TABLES `xray_image` WRITE;

/*!40000 ALTER TABLE `xray_image` DISABLE KEYS */;

INSERT INTO `xray_image` VALUES (1,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_1.jpg',1),(2,
_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_2.jpg',1),(3,
_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_3.jpg',1),(4,
_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_4.jpg',1),(5,
_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_5.jpg',1),(6,
_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_6.jpg',1),(7,
_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_7.jpg',1),(8,
_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_8.jpg',1),(9,
_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_9.jpg',1),(10
,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_10.jpg',1),(1
1,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_11.jpg',1),(1
2,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_12.jpg',1),(1
3,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_13.jpg',1),(1
4,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_14.jpg',1),(1
5,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_15.jpg',1),(1
6,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_16.jpg',1),(1
7,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_17.jpg',1),(1
8,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_18.jpg',1),(1
9,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_19.jpg',1),(2
0,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_20.jpg',1),(2
1,_binary

'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_21.jpg',1),(22,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_22.jpg',1),(23,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_23.jpg',1),(24,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_24.jpg',1),(25,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_25.jpg',1),(26,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_26.jpg',1),(27,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_27.jpg',1),(28,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_28.jpg',1),(29,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_29.jpg',1),(30,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_30.jpg',1),(31,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_31.jpg',1),(32,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_32.jpg',1),(33,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_33.jpg',1),(34,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_34.jpg',1),(35,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_35.jpg',1),(36,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_36.jpg',1),(37,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_37.jpg',1),(38,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_38.jpg',1),(39,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_39.jpg',1),(40,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_40.jpg',1),(41,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_41.jpg',1),(42,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_42.jpg',1),(43,_binary

'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_43.jpg',1),(44,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_44.jpg',1),(45,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_45.jpg',1),(46,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_46.jpg',1),(47,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_47.jpg',1),(48,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_48.jpg',1),(49,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_49.jpg',1),(50,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_50.jpg',1),(51,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_51.jpg',1),(52,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_52.jpg',1),(53,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_53.jpg',1),(54,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_54.jpg',1),(55,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_55.jpg',1),(56,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_56.jpg',1),(57,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_57.jpg',1),(58,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_58.jpg',1),(59,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_59.jpg',1),(60,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_60.jpg',1),(61,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_61.jpg',1),(62,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_62.jpg',1),(63,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_63.jpg',1),(64,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_64.jpg',1),(65,_binary

'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_65.jpg',1),(66,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_66.jpg',1),(67,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_67.jpg',1),(68,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_68.jpg',1),(69,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_69.jpg',1),(70,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_70.jpg',1),(71,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_71.jpg',1),(72,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_72.jpg',1),(73,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_73.jpg',1),(74,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_74.jpg',1),(75,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_75.jpg',1),(76,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_76.jpg',1),(77,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_77.jpg',1),(78,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_78.jpg',1),(79,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_79.jpg',1),(80,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_80.jpg',1),(81,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_81.jpg',1),(82,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_82.jpg',1),(83,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_83.jpg',1),(84,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_84.jpg',1),(85,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_85.jpg',1),(86,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_86.jpg',1),(87,_binary

'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_87.jpg',1),(88,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_88.jpg',1),(89,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_89.jpg',1),(90,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_90.jpg',1),(91,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_91.jpg',1),(92,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_92.jpg',1),(93,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_93.jpg',1),(94,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_94.jpg',1),(95,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_95.jpg',1),(96,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_96.jpg',1),(97,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_97.jpg',1),(98,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_98.jpg',1),(99,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_99.jpg',1),(100,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_100.jpg',1),(101,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_101.jpg',1),(102,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_102.jpg',1),(103,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_103.jpg',1),(104,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_104.jpg',1),(105,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_105.jpg',1),(106,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_106.jpg',1),(107,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_107.jpg',1),(108,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_108.jpg',1),(109,_binary

'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_109.jpg',1),(
110,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_110.jpg',1),(
111,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_111.jpg',1),(
112,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_112.jpg',1),(
113,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_113.jpg',1),(
114,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_114.jpg',1),(
115,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_115.jpg',1),(
116,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_116.jpg',1),(
117,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_117.jpg',1),(
118,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_118.jpg',1),(
119,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_119.jpg',1),(
120,_binary
'C://Users//nares//OneDrive//Desktop//Final_image//Final_image//Xray//xray_120.jpg',1);

/*!40000 ALTER TABLE `xray_image` ENABLE KEYS */;

UNLOCK TABLES;

--

-- Dumping events for database 'Medical_Imaging'

--

--

-- Dumping routines for database 'Medical_Imaging'

--

/*!50003 DROP FUNCTION IF EXISTS `body_part_max_image` */;

/*!50003 SET @saved_cs_client = @@character_set_client */ ;

/*!50003 SET @saved_cs_results = @@character_set_results */ ;

/*!50003 SET @saved_col_connection = @@collation_connection */ ;

```
/*!50003 SET character_set_client  = utf8mb4 */ ;

/*!50003 SET character_set_results = utf8mb4 */ ;

/*!50003 SET collation_connection  = utf8mb4_0900_ai_ci */ ;

/*!50003 SET @saved_sql_mode       = @@sql_mode */ ;

/*!50003 SET sql_mode              =
'ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERRO
R_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION' */ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` FUNCTION `body_part_max_image`() RETURNS
varchar(50) CHARSET utf8mb4

    READS SQL DATA

    DETERMINISTIC

BEGIN

 DECLARE stringValue VARCHAR(50);

 DECLARE Total__Brain_Image INT;

 DECLARE Total__Knee_Image INT;

 DECLARE Total__Shoulder_Image INT;


 SET Total__Brain_Image = (SELECT COUNT(B.Body_PartID) FROM Body_Part B

    JOIN Body_Part_Technique BT JOIN MRI_Image I

    WHERE B.Body_PartID = BT.Body_PartID

    AND BT.Imaging_TechniqueID = I.Imaging_TechniqueID AND Body_Part_Name = 'Brain');


SET Total__Knee_Image = (SELECT COUNT(B.Body_PartID) FROM Body_Part B

    JOIN Body_Part_Technique BT JOIN CT_Scan_Image I

    WHERE B.Body_PartID = BT.Body_PartID

    AND BT.Imaging_TechniqueID = I.Imaging_TechniqueID AND Body_Part_Name = 'Knee');


SET Total__Shoulder_Image = (SELECT COUNT(B.Body_PartID) FROM Body_Part B

    JOIN Body_Part_Technique BT JOIN XRay_Image I
```

```
   WHERE B.Body_PartID = BT.Body_PartID
   AND BT.Imaging_TechniqueID = I.Imaging_TechniqueID AND Body_Part_Name =
'Shoulder');


IF (Total__Brain_Image > Total__Knee_Image) AND (Total__Brain_Image >
Total__Shoulder_Image) THEN
SET stringValue = 'BRAIN';
ELSEIF Total__Brain_Image > Total__Knee_Image THEN
SET stringValue = 'BRAIN';
ELSEIF Total__Brain_Image > Total__Shoulder_Image THEN
SET stringValue = 'BRAIN';


ELSEIF (Total__Knee_Image > Total__Brain_Image) AND (Total__Knee_Image >
Total__Shoulder_Image) THEN
SET stringValue = 'KNEE';
ELSEIF Total__Knee_Image > Total__Brain_Image THEN
SET stringValue = 'KNEE';
ELSEIF Total__Knee_Image > Total__Shoulder_Image THEN
SET stringValue = 'KNEE';


ELSEIF (Total__Shoulder_Image > Total__Brain_Image) AND (Total__Shoulder_Image >
Total__Knee_Image) THEN
SET stringValue = 'KNEE';
ELSEIF Total__Shoulder_Image > Total__Knee_Image THEN
SET stringValue = 'KNEE';
ELSEIF Total__Shoulder_Image > Total__Shoulder_Image THEN
SET stringValue = 'KNEE';
ELSE
SET stringValue = 'ALL ARE EQUAL';
 END IF;
 RETURN stringValue;
```

```
END ;;
DELIMITER ;
/*!50003 SET sql_mode              = @saved_sql_mode */ ;
/*!50003 SET character_set_client  = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection  = @saved_col_connection */ ;
/*!50003 DROP FUNCTION IF EXISTS `technique_performance` */;
/*!50003 SET @saved_cs_client      = @@character_set_client */ ;
/*!50003 SET @saved_cs_results     = @@character_set_results */ ;
/*!50003 SET @saved_col_connection = @@collation_connection */ ;
/*!50003 SET character_set_client  = utf8mb4 */ ;
/*!50003 SET character_set_results = utf8mb4 */ ;
/*!50003 SET collation_connection  = utf8mb4_0900_ai_ci */ ;
/*!50003 SET @saved_sql_mode       = @@sql_mode */ ;
/*!50003 SET sql_mode              =
'ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERRO
R_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` FUNCTION `technique_performance`() RETURNS
varchar(50) CHARSET utf8mb4
    READS SQL DATA
    DETERMINISTIC
Begin
 DECLARE stringValue VARCHAR(50);
 DECLARE Total__By_MRI INT;
 DECLARE Total__By_CT_SCAN INT;
 DECLARE Total__By_XRAY INT;

 SET Total__By_MRI = (SELECT COUNT(d.DiseaseID) FROM Disease d
    JOIN Disease_Detection dd JOIN Imaging_Technique i
```

```
   WHERE d.DiseaseID = dd.DiseaseID
   AND DD.Imaging_TechniqueID = i.Imaging_TechniqueID AND i.Imaging_Technique_Name
= 'MRI');
SET Total__By_CT_SCAN = (SELECT COUNT(d.DiseaseID) FROM Disease d
   JOIN Disease_Detection dd JOIN Imaging_Technique i
   WHERE d.DiseaseID = dd.DiseaseID
   AND DD.Imaging_TechniqueID = i.Imaging_TechniqueID AND i.Imaging_Technique_Name
= 'CT Scan');
SET Total__By_XRAY = (SELECT COUNT(d.DiseaseID) FROM Disease d
   JOIN Disease_Detection dd JOIN Imaging_Technique i
   WHERE d.DiseaseID = dd.DiseaseID
   AND DD.Imaging_TechniqueID = i.Imaging_TechniqueID AND i.Imaging_Technique_Name
= 'X-Ray');
 IF (Total__By_MRI > Total__By_CT_SCAN) AND (Total__By_MRI > Total__By_XRAY) THEN
 SET stringValue = 'MRI detected more disease than CT Scan AND X-Ray ';
 ELSEIF Total__By_MRI > Total__By_CT_SCAN THEN
 SET stringValue = 'MRI detected more disease than CT_Scan';
 ELSEIF Total__By_MRI > Total__By_XRAY THEN
 SET stringValue = 'MRI detected more disease than X-Ray';
 ELSE
 SET stringValue = 'MRI detected less disease than CT Scan and X-Ray';
 END IF;
 RETURN stringValue;
END ;;
DELIMITER ;
/*!50003 SET sql_mode              = @saved_sql_mode */ ;
/*!50003 SET character_set_client  = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection  = @saved_col_connection */ ;
/*!50003 DROP PROCEDURE IF EXISTS `add_body_part` */;
```

```
/*!50003 SET @saved_cs_client      = @@character_set_client */ ;

/*!50003 SET @saved_cs_results     = @@character_set_results */ ;

/*!50003 SET @saved_col_connection = @@collation_connection */ ;

/*!50003 SET character_set_client  = utf8mb4 */ ;

/*!50003 SET character_set_results = utf8mb4 */ ;

/*!50003 SET collation_connection  = utf8mb4_0900_ai_ci */ ;

/*!50003 SET @saved_sql_mode       = @@sql_mode */ ;

/*!50003 SET sql_mode              =
'ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERRO
R_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION' */ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `add_body_part`(IN bodypartID INT, IN
bodypartname VARCHAR(50))

BEGIN

DECLARE message_text VARCHAR(50);

INSERT INTO body_part (Body_PartID, Body_Part_Name)

SELECT @bodypartID, @bodypartname;

SET @message_text = 'New BodyPart added successfully';

SELECT @message_text;

##LIMIT 1

END ;;

DELIMITER ;

/*!50003 SET sql_mode              = @saved_sql_mode */ ;

/*!50003 SET character_set_client  = @saved_cs_client */ ;

/*!50003 SET character_set_results = @saved_cs_results */ ;

/*!50003 SET collation_connection  = @saved_col_connection */ ;

/*!50003 DROP PROCEDURE IF EXISTS `add_disease` */;

/*!50003 SET @saved_cs_client      = @@character_set_client */ ;

/*!50003 SET @saved_cs_results     = @@character_set_results */ ;

/*!50003 SET @saved_col_connection = @@collation_connection */ ;
```

```
/*!50003 SET character_set_client  = utf8mb4 */ ;

/*!50003 SET character_set_results = utf8mb4 */ ;

/*!50003 SET collation_connection  = utf8mb4_0900_ai_ci */ ;

/*!50003 SET @saved_sql_mode       = @@sql_mode */ ;

/*!50003 SET sql_mode              =
'ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERRO
R_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION' */ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `add_disease`(IN diseaseID INT, IN
diseasename VARCHAR(50))

BEGIN

DECLARE message_text VARCHAR(50);

INSERT INTO Disease (DiseaseID, Disease_Name)

SELECT @diseaseid, @diseasename;

SET @message_text = 'Disease added successfully';

SELECT @message_text;

##LIMIT 1


END ;;

DELIMITER ;

/*!50003 SET sql_mode              = @saved_sql_mode */ ;

/*!50003 SET character_set_client  = @saved_cs_client */ ;

/*!50003 SET character_set_results = @saved_cs_results */ ;

/*!50003 SET collation_connection  = @saved_col_connection */ ;

/*!50003 DROP PROCEDURE IF EXISTS `add_imaging_technique` */;

/*!50003 SET @saved_cs_client      = @@character_set_client */ ;

/*!50003 SET @saved_cs_results     = @@character_set_results */ ;

/*!50003 SET @saved_col_connection = @@collation_connection */ ;

/*!50003 SET character_set_client  = utf8mb4 */ ;

/*!50003 SET character_set_results = utf8mb4 */ ;
```

```
/*!50003 SET collation_connection  = utf8mb4_0900_ai_ci */ ;

/*!50003 SET @saved_sql_mode      = @@sql_mode */ ;

/*!50003 SET sql_mode             =
'ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERRO
R_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION' */ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `add_imaging_technique`(IN techniqueID
INT, IN techniquename VARCHAR(50))

BEGIN

DECLARE message_text VARCHAR(50);

INSERT INTO Imaging_Technique (Imaging_TechniqueID, Imaging_Technique_Name)

SELECT @techniqueID, @techniquename;

SET @message_text = 'New Technique added successfully';

SELECT @message_text;

##LIMIT 1

END ;;

DELIMITER ;

/*!50003 SET sql_mode             = @saved_sql_mode */ ;

/*!50003 SET character_set_client  = @saved_cs_client */ ;

/*!50003 SET character_set_results = @saved_cs_results */ ;

/*!50003 SET collation_connection  = @saved_col_connection */ ;

/*!50003 DROP PROCEDURE IF EXISTS `add_user` */;

/*!50003 SET @saved_cs_client     = @@character_set_client */ ;

/*!50003 SET @saved_cs_results    = @@character_set_results */ ;

/*!50003 SET @saved_col_connection = @@collation_connection */ ;

/*!50003 SET character_set_client  = utf8mb4 */ ;

/*!50003 SET character_set_results = utf8mb4 */ ;

/*!50003 SET collation_connection  = utf8mb4_0900_ai_ci */ ;

/*!50003 SET @saved_sql_mode      = @@sql_mode */ ;
```

```
/*!50003 SET sql_mode          =
'ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERRO
R_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION' */ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `add_user`(IN userid INT, IN fname
VARCHAR(50), IN lname VARCHAR(50), IN age INT, IN sex VARCHAR(10), IN emailid
VARCHAR(50), IN phone VARCHAR(50), IN addressid INT)

BEGIN

DECLARE message_text VARCHAR(50);

INSERT INTO User (UserID, First_Name, Last_Name, Age, Sex, Email, Phone_No, AddressID)

SELECT @userid, @fname, @lname, @age, @sex, @emailid, @phone, @addressid;

SET @message_text = 'User added successfully';

SELECT @message_text;

##LIMIT 1

COMMIT;

END ;;

DELIMITER ;

/*!50003 SET sql_mode          = @saved_sql_mode */ ;

/*!50003 SET character_set_client  = @saved_cs_client */ ;

/*!50003 SET character_set_results = @saved_cs_results */ ;

/*!50003 SET collation_connection  = @saved_col_connection */ ;

/*!50003 DROP PROCEDURE IF EXISTS `search_technique_for_disease` */;

/*!50003 SET @saved_cs_client     = @@character_set_client */ ;

/*!50003 SET @saved_cs_results    = @@character_set_results */ ;

/*!50003 SET @saved_col_connection = @@collation_connection */ ;

/*!50003 SET character_set_client  = utf8mb4 */ ;

/*!50003 SET character_set_results = utf8mb4 */ ;

/*!50003 SET collation_connection  = utf8mb4_0900_ai_ci */ ;

/*!50003 SET @saved_sql_mode      = @@sql_mode */ ;
```

```
/*!50003 SET sql_mode          =
'ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERRO
R_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION' */ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `search_technique_for_disease`(IN
diseasename VARCHAR(50))

BEGIN

SELECT Imaging_Technique_Name

FROM Imaging_Technique I

WHERE Imaging_TechniqueID IN (

        SELECT Imaging_TechniqueID

   FROM Disease_Detection DD

        JOIN Disease D

        WHERE I.Imaging_TechniqueID = DD.Imaging_TechniqueID

        AND DD.DiseaseID = D.DiseaseID

        AND Disease_Name = diseasename);

END ;;

DELIMITER ;

/*!50003 SET sql_mode          = @saved_sql_mode */ ;

/*!50003 SET character_set_client  = @saved_cs_client */ ;

/*!50003 SET character_set_results = @saved_cs_results */ ;

/*!50003 SET collation_connection  = @saved_col_connection */ ;

/*!50003 DROP PROCEDURE IF EXISTS `verify_user_login_proc` */;

/*!50003 SET @saved_cs_client     = @@character_set_client */ ;

/*!50003 SET @saved_cs_results    = @@character_set_results */ ;

/*!50003 SET @saved_col_connection = @@collation_connection */ ;

/*!50003 SET character_set_client  = utf8mb4 */ ;

/*!50003 SET character_set_results = utf8mb4 */ ;

/*!50003 SET collation_connection  = utf8mb4_0900_ai_ci */ ;

/*!50003 SET @saved_sql_mode      = @@sql_mode */ ;
```

```
/*!50003 SET sql_mode          =
'ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERRO
R_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION' */ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `verify_user_login_proc`(IN user_name
VARCHAR(1000), IN password VARCHAR(1000))

   READS SQL DATA

BEGIN

        SELECT IF(count(*) > 0, 0, 1) AS Result

   FROM user_login

   WHERE Username = user_name

   AND Password = password;

END ;;

DELIMITER ;

/*!50003 SET sql_mode          = @saved_sql_mode */ ;

/*!50003 SET character_set_client  = @saved_cs_client */ ;

/*!50003 SET character_set_results = @saved_cs_results */ ;

/*!50003 SET collation_connection  = @saved_col_connection */ ;


--

-- Final view structure for view `disease_ctscan`

--


/*!50001 DROP VIEW IF EXISTS `disease_ctscan`*/;

/*!50001 SET @saved_cs_client        = @@character_set_client */;

/*!50001 SET @saved_cs_results       = @@character_set_results */;

/*!50001 SET @saved_col_connection    = @@collation_connection */;

/*!50001 SET character_set_client     = utf8mb4 */;

/*!50001 SET character_set_results    = utf8mb4 */;

/*!50001 SET collation_connection     = utf8mb4_0900_ai_ci */;
```

```
/*!50001 CREATE ALGORITHM=UNDEFINED */

/*!50013 DEFINER=`root`@`localhost` SQL SECURITY DEFINER */

/*!50001 VIEW `disease_ctscan` AS select `d`.`DiseaseID` AS `DiseaseID`,`d`.`Disease_Name`
AS `Disease_Name`,`i`.`Imaging_Technique_Name` AS `Imaging_Technique_Name` from
((`imaging_technique` `i` join `disease_detection` `dd`) join `disease` `d`) where
((`dd`.`Imaging_TechniqueID` = `i`.`Imaging_TechniqueID`) and (`dd`.`DiseaseID` =
`d`.`DiseaseID`) and (`i`.`Imaging_Technique_Name` = 'CT Scan')) */;

/*!50001 SET character_set_client      = @saved_cs_client */;

/*!50001 SET character_set_results     = @saved_cs_results */;

/*!50001 SET collation_connection      = @saved_col_connection */;


--

-- Final view structure for view `disease_mri`

--


/*!50001 DROP VIEW IF EXISTS `disease_mri`*/;

/*!50001 SET @saved_cs_client          = @@character_set_client */;

/*!50001 SET @saved_cs_results         = @@character_set_results */;

/*!50001 SET @saved_col_connection     = @@collation_connection */;

/*!50001 SET character_set_client      = utf8mb4 */;

/*!50001 SET character_set_results     = utf8mb4 */;

/*!50001 SET collation_connection      = utf8mb4_0900_ai_ci */;

/*!50001 CREATE ALGORITHM=UNDEFINED */

/*!50013 DEFINER=`root`@`localhost` SQL SECURITY DEFINER */

/*!50001 VIEW `disease_mri` AS select `d`.`DiseaseID` AS `DiseaseID`,`d`.`Disease_Name` AS
`Disease_Name`,`i`.`Imaging_Technique_Name` AS `Imaging_Technique_Name` from
((`imaging_technique` `i` join `disease_detection` `dd`) join `disease` `d`) where
((`dd`.`Imaging_TechniqueID` = `i`.`Imaging_TechniqueID`) and (`dd`.`DiseaseID` =
`d`.`DiseaseID`) and (`i`.`Imaging_Technique_Name` = 'MRI')) */;

/*!50001 SET character_set_client      = @saved_cs_client */;

/*!50001 SET character_set_results     = @saved_cs_results */;

/*!50001 SET collation_connection      = @saved_col_connection */;
```

```
--
-- Final view structure for view `disease_xray`
--

/*!50001 DROP VIEW IF EXISTS `disease_xray`*/;
/*!50001 SET @saved_cs_client          = @@character_set_client */;
/*!50001 SET @saved_cs_results         = @@character_set_results */;
/*!50001 SET @saved_col_connection     = @@collation_connection */;
/*!50001 SET character_set_client      = utf8mb4 */;
/*!50001 SET character_set_results     = utf8mb4 */;
/*!50001 SET collation_connection      = utf8mb4_0900_ai_ci */;
/*!50001 CREATE ALGORITHM=UNDEFINED */
/*!50013 DEFINER=`root`@`localhost` SQL SECURITY DEFINER */
/*!50001 VIEW `disease_xray` AS select `d`.`DiseaseID` AS `DiseaseID`,`d`.`Disease_Name`
AS `Disease_Name`,`i`.`Imaging_Technique_Name` AS `Imaging_Technique_Name` from
((`imaging_technique` `i` join `disease_detection` `dd`) join `disease` `d`) where
((`dd`.`Imaging_TechniqueID` = `i`.`Imaging_TechniqueID`) and (`dd`.`DiseaseID` =
`d`.`DiseaseID`) and (`i`.`Imaging_Technique_Name` = 'X-Ray')) */;
/*!50001 SET character_set_client      = @saved_cs_client */;
/*!50001 SET character_set_results     = @saved_cs_results */;
/*!50001 SET collation_connection      = @saved_col_connection */;

--
-- Final view structure for view `user_with_instructor_role_vw`
--

/*!50001 DROP VIEW IF EXISTS `user_with_instructor_role_vw`*/;
/*!50001 SET @saved_cs_client          = @@character_set_client */;
/*!50001 SET @saved_cs_results         = @@character_set_results */;
/*!50001 SET @saved_col_connection     = @@collation_connection */;
```

```
/*!50001 SET character_set_client      = utf8mb4 */;

/*!50001 SET character_set_results     = utf8mb4 */;

/*!50001 SET collation_connection      = utf8mb4_0900_ai_ci */;

/*!50001 CREATE ALGORITHM=UNDEFINED */

/*!50013 DEFINER=`root`@`localhost` SQL SECURITY DEFINER */

/*!50001 VIEW `user_with_instructor_role_vw` AS select `u`.`First_Name` AS
`First_Name`,`u`.`Last_Name` AS `Last_Name`,`u`.`Age` AS `Age`,`u`.`Sex` AS `Sex`,`u`.`Email`
AS `Email`,`u`.`Phone_No` AS `Phone_No`,`r`.`Role_Name` AS `Role_Name` from ((`user` `u`
join `user_role` `ur`) join `role` `r`) where ((`u`.`UserID` = `ur`.`UserID`) and (`ur`.`RoleID` =
`r`.`RoleID`) and (`r`.`Role_Name` = 'Instructor')) */;

/*!50001 SET character_set_client      = @saved_cs_client */;

/*!50001 SET character_set_results     = @saved_cs_results */;

/*!50001 SET collation_connection      = @saved_col_connection */;

/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;


/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;

/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;

/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;

/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;

/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;


-- Dump completed on 2019-12-10 16:46:16
```


References:

- https://www.tcts.fpms.ac.be/publications/regpapers/2007/VS_cmtbg2007.pdf
- https://github.com/llSourcell/AI_in_Medicine_Clinical_Imaging_Classification#preprocessing
- https://www.youtube.com/watch?v=mWl45NkFBOc