Java -jar jenkins.war

Select the item

Custom workspace , build triggers

H/15 \* \* \* \*

Every 15 minutes job should run.

This is so called job should schedule.

Next ..

Build ..Windows batch command

cd tests

pt.test --html=report.html -v

--junitxml = "results.xml"

Post build action Email Notifications..

[SMTP.gmail.com](http://smtp.gmail.com/)

465.

Continuous Integration purpose

**Jenkins** is an open source automation tool written in Java with plugins built for Continuous Integration purpose. **Jenkins** is used to build and test your software projects continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build

**Git** is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

git config --global user.name "Priya24it"

git config --global user.email priya.son24@gmail.com

git add \*

git commit -m "first commit"

Step1:

git remote add origin https://github.com/priya24it/dbtasks

git push -u origin master

Step2:

git add README.md

git add \*

git commit -m "first commit"

Xlsxwriter

excelWriter = pd.ExcelWriter(**'MDMBusinessData.xlsx'**)

dfMain.to\_excel(excelWriter, sheet\_name=ListofTableNames[i])  
workbook = excelWriter.book  
worksheet = excelWriter.sheets[ListofTableNames[i]]  
  
format = workbook.add\_format({**'text\_wrap'**: **True**})  
*# Setting the format but not setting the column width.*worksheet.set\_column(**'C:C'**, 15, format)  
worksheet.set\_column(**'D:D'**, 15, format)  
worksheet.set\_column(**'E:E'**, 15, format)

**import**SQLConnection **as**con  
**import**pandas **as**pd  
**import**pandas.io.sql **as**SQL  
  
excelWriter = pd.ExcelWriter(**'MDMBusinessData.xlsx'**)  
  
TablesData = pd.read\_excel(**'MDMBackupTables.xlsx'**,sheet\_name=**'MDMBusinessTables'**)  
TablesData = pd.DataFrame(TablesData)  
ListofTableNames = TablesData[**'TableNames'**].values.tolist()  
CountofTables = len(TablesData[**'SQLQuery'**].values.tolist())  
  
  
ListofIDs = pd.read\_excel(**'MDMBackupTables.xlsx'**,sheet\_name=**'ListofIDs'**)  
ListofIDs = pd.DataFrame(ListofIDs)  
ListofIDs = ListofIDs[**'ID'**].values.tolist()  
ListofIDs = str(ListofIDs)  
ListofIDs = ListofIDs.replace(**'['**,**''**)  
ListofIDs = ListofIDs.replace(**']'**,**''**)  
  
print(CountofTables)  
*#i= 0***for**i **in**range(CountofTables):  
    print(i)  
    sql =TablesData[**'SQLQuery'**][i] + **' and grid in ('**+ListofIDs+**')'***#sql = TablesData[i]+' and grid in ('+ListofIDs+')'*print(sql)  
    f= open(**'sql.txt'**,**'a'**)  
    f.write(str(sql))  
    dfMainExcel = SQL.read\_sql(sql,con.conn)  
    dfMain = pd.DataFrame(dfMainExcel)  
    print(dfMain)  
    dfMain.to\_excel(excelWriter, sheet\_name=ListofTableNames[i])  
    workbook = excelWriter.book  
    worksheet = excelWriter.sheets[ListofTableNames[i]]  
  
    format = workbook.add\_format({**'text\_wrap'**: **True**})  
    *# Setting the format but not setting the column width.*worksheet.set\_column(**'C:C'**, 15, format)  
    worksheet.set\_column(**'D:D'**, 15, format)  
    worksheet.set\_column(**'E:E'**, 15, format)  
    worksheet.set\_column(**'F:F'**, 15, format)  
    worksheet.set\_column(**'G:G'**, 15, format)  
    worksheet.set\_column(**'H:H'**, 15, format)  
    worksheet.set\_column(**'I:I'**, 15, format)  
    worksheet.set\_column(**'A:A'**, 15, format)  
    worksheet.set\_column(**'B:B'**, 15, format)  
  
f.close()  
excelWriter.save()