**Threat Actor Profiling (MySQL)**

**Table1: apt\_group**

* Added URL Column
* Associated Group Column

**Table2: apt\_group\_technique**

* group\_id
* technique\_id

**Changed both above datatypess to VARCHAR from int**

**Table3: Table software\_used (Added)**

CREATE TABLE software\_used (

id VARCHAR(255) PRIMARY KEY,

name VARCHAR(255),

reference VARCHAR(255),

techniques VARCHAR(255)

);

**Table4: sub\_id (Added)**

CREATE TABLE sub\_id (

id VARCHAR(255) PRIMARY KEY,

name VARCHAR(255)

);

**Table5: procedure\_example (Added)**

CREATE TABLE procedure\_example (

id VARCHAR(255) PRIMARY KEY,

name VARCHAR(255),

description Varchar(255),

reference varchar(255)

);

**Code of the Pipeline for the SQL:**

|  |
| --- |
| # Define your item pipelines here  #  # Don't forget to add your pipeline to the ITEM\_PIPELINES setting  # See: https://docs.scrapy.org/en/latest/topics/item-pipeline.html  import re  import scrapy  # useful for handling different item types with a single interface  from itemadapter import ItemAdapter  import mysql.connector  import scrapy.item  # class TutorialPipeline:  #     def process\_item(self, item, spider):  #         return item  # group pipeline  MYSQL\_SETTINGS = {      'host': 'localhost',      'port': 3306,      'database': 'etiapt',      'user': 'root',      'password': '7777',#7777:1234  }  class GroupTable (scrapy.Item):      MittreName=scrapy.Field()      GroupName=scrapy.Field()      Summary=scrapy.Field()      AssociatedGroups=scrapy.Field()      Url=scrapy.Field()    class TechniquesTable(scrapy.Item):      ID=scrapy.Field()      Use=scrapy.Field()      Domain=scrapy.Field()      References=scrapy.Field()      SubId=scrapy.Field()      GroupId=scrapy.Field()  class SoftwareTable(scrapy.Item):      ID=scrapy.Field()      Name=scrapy.Field()      References=scrapy.Field()      Techniques=scrapy.Field()  class CompainsTable(scrapy.Item):      ID=scrapy.Field()      Name=scrapy.Field()      FirstSeen=scrapy.Field()      LastSeen=scrapy.Field()      References=scrapy.Field()      Techniques=scrapy.Field()  class SubTechniques(scrapy.Item):      ID=scrapy.Field()      Name=scrapy.Field()    class ProcedureExamples(scrapy.Item):      Id=scrapy.Field()      Name=scrapy.Field()      Description=scrapy.Field()  class Mitigations(scrapy.Item):      ID=scrapy.Field()      Mitigation=scrapy.Field()      Description=scrapy.Field()  class Detections(scrapy.Item):      ID=scrapy.Field()      DataSource=scrapy.Field()      DataComponent=scrapy.Field()      Detects=scrapy.Field()  # class AptReferences(scrapy.Item):  #     ID=scrapy.Field()  #     Url=scrapy.Field()  class MySQLPipeline:      def open\_spider(self, spider):          self.conn = mysql.connector.connect(\*\*MYSQL\_SETTINGS)          self.cursor = self.conn.cursor()          print("conection est ")      def close\_spider(self, spider):          self.conn.close()      def process\_item(self, item, spider):          if isinstance(item, GroupTable):              try:                  sql = "INSERT INTO apt\_group (mitre\_name, group\_name, summary, associated\_groups, group\_url) VALUES (%s, %s, %s, %s, %s)"                    values = (item.get('MittreName'), item.get('GroupName'), item.get('Summary'), item.get('AssociatedGroups'), item.get('Url'))                  self.cursor.execute(sql, values)                  self.conn.commit()              except mysql.connector.Error as err:                  if err.errno == 1062:  # MySQL error code for duplicate entry                      print("------------Duplicate entry found for the provided values in apt\_group table.--------------")                  else:                      print("--------------An error occurred:-----------------", err)                      print("An error occurred:", err)          elif isinstance(item,TechniquesTable):              try:                    sql = "INSERT INTO apt\_group\_techniques ( techniques\_id, description, domain\_name,sub\_id ) VALUES (%s, %s,%s,%s)"                  values = ( item.get('ID'), item.get('Use'), item.get('Domain'),item.get('SubId'))                  # sqlref="INSERT INTO apt\_references (reference\_id, reference\_link) VALUES (%s, %s)"                  # Using regex to find all URLs in the string                  input\_string=item.get('References')                  technique=item.get('ID')                  links = re.findall(r'https?://(?:[-\w.]|(?:%[\da-fA-F]{2}))+', input\_string)                  print("linkes are :",technique)                  for link in links:                      query = ("INSERT INTO apt\_references (reference\_link, apt\_group\_techniques\_techniques\_id) VALUES (%s,%s) ")                      ref\_values=(link,technique)                      try:                          self.cursor.execute(query, ref\_values)                          print("added",link,technique)                      except:                          print("Failed")                  self.cursor.execute(sql, values)                  self.conn.commit()              except mysql.connector.Error as err:                  if err.errno == 1062:  # MySQL error code for duplicate entry                      print("------------Duplicate entry found for the provided values in apt\_group table.--------------")                  else:                      print("--------------An error occurred:-----------------", err)                      print("An error occurred:", err)          elif isinstance(item,SoftwareTable):              try:                  sql = "INSERT INTO software\_used( id, name,techniques ) VALUES ( %s, %s,%s)"                  values = ( item.get('ID'), item.get('Name'), item.get('References',),item.get('Techniques'))                  links = re.findall(r'https?://(?:[-\w.]|(?:%[\da-fA-F]{2}))+', item.get('References'))                    print(links)                  for link in links:                   software\_id=item.get('ID')                   query = ("INSERT INTO apt\_references (reference\_link,software\_used\_software\_Id) VALUES (%s, %s)")                   self.cursor.execute(query, (link,software\_id))                  self.cursor.execute(sql, values)                  self.conn.commit()              except mysql.connector.Error as err:                  if err.errno == 1062:# MySQL error code for duplicate entry                      print("------------Duplicate entry found for the provided values in apt\_group table.--------------")                  else:                      print("--------------An error occurred:-----------------", err)                      print("An error occurred:", err)          elif isinstance(item,CompainsTable):              try:                  sql = "INSERT INTO CompainsTable ( id, name, reference,techniques ) VALUES (%s, %s, %s,%s)"                  values = ( item.get('ID'),  item.get('Name'), item.get('References',),item.get('Techniques'))                  self.cursor.execute(sql, values)                  self.conn.commit()              except mysql.connector.Error as err:                  if err.errno == 1062:  # MySQL error code for duplicate entry                      print("------------Duplicate entry found for the provided values in apt\_group table.--------------")                  else:                      print("--------------An error occurred:-----------------", err)                      print("An error occurred:", err)          elif isinstance(item,SubTechniques):              try:                  sql = "INSERT INTO sub\_id ( id, name) VALUES (%s, %s)"                  values = ( item.get('ID'), item.get('Name'))                  self.cursor.execute(sql, values)                  self.conn.commit()              except mysql.connector.Error as err:                  if err.errno == 1062:  # MySQL error code for duplicate entry                      print("------------Duplicate entry found for the provided values in apt\_group table.--------------")                  else:                      print("--------------An error occurred:-----------------", err)                      print("An error occurred:", err)          elif isinstance(item,ProcedureExamples):              try:                  sql = "INSERT INTO procedure\_example ( id, name,description,reference) VALUES (%s, %s,%s,%s)"                  values = ( item.get('ID'), item.get('Name'),item.get('Description'),item.get('Reference'))                  self.cursor.execute(sql, values)                  self.conn.commit()              except mysql.connector.Error as err:                  if err.errno == 1062:  # MySQL error code for duplicate entry                      print("------------Duplicate entry found for the provided values in apt\_group table.--------------")                  else:                      print("--------------An error occurred:-----------------", err)                      print("An error occurred:", err)          elif isinstance(item,Mitigations):              try:                  sql = "INSERT INTO mitigations ( id, name,description,reference) VALUES (%s, %s,%s,%s)"                  values = ( item.get('ID'), item.get('Name'),item.get('Description'),item.get('Reference'))                  self.cursor.execute(sql, values)                  self.conn.commit()              except mysql.connector.Error as err:                  if err.errno == 1062:  # MySQL error code for duplicate entry                      print("------------Duplicate entry found for the provided values in apt\_group table.--------------")                  else:                      print("--------------An error occurred:-----------------", err)                      print("An error occurred:", err)          return item    #group\_name varchar(255)  # mitre\_name varchar(255)  # summary longtext  # created\_date datetime  # modified\_date |