In this document, we will describe about different aspects of Stack Exchange Data Generation Process. In section 1, our Data Base Schema and then program's logic will be described. In section 2, program running manual is presented and in section 3, Data Collection details are presented.

1- Data Collection Description

This project is programmed to prepare a data set of User Profiles based on Stack Exchange Community Question Answering Log Files that are available Here. This project is categorized into some classes that are detailed below:

1-1- CreateTables Class

This class makes some tables according to Stack Exchange Data Base Schema that are available here. Tables summary in order is:

Table Schema

User Table

This table takes hold information about users and in this program we've selected some features as below fields:

UserId, siteName, PostTypeId, PostScore, PostViewCount, PostTags, PostAnswerCount, PostCommentCount, PostFavoriteCount, PostOwnerUserId, UserReputation, UserViews, UserUpVotes, UserDownVotes, PostCreationDate, UserCreationDate, UserDisplayName, UserLastAccessDate, UserWebsiteUrl, UserLocation, UserEmailHash, UserAge, UserAccountId

It should be noticed that "UserAccountId" field is common among all different sites of Stack Exchange Tables. It does mean if a individual user has some post activities(such as question or answer posts) in different sites, all can be retrieved by "UserAccountId" field.

Post Table

This table takes hold information about all kinds of users' posts but in this program we've just selected Question and Answer post Types; Posts Table Fields are:

id, PostId, siteName, PostTypeId, CreationDate, Score, ViewCount, OwnerUserId, Tags, AnswerCount, CommentCount, FavoriteCount.

It should be noticed that the "**OwnerUserId**" field refers to Users Table "**UserId**" field; additionally, we add "siteName" field to users table which specifies each (question/answer) post site.

UsersPostJoin Table

Per record of this table represents integrated information about an individual user post. UsersPostJoin Table Fields are:

id UserId, siteName, PostTypeId, PostScore, PostViewCount, PostTags, PostAnswerCount, PostCommentCount, PostFavoriteCount, PostOwnerUserId, UserReputation, UserViews, UserUpVotes, UserDownVotes, PostCreationDate, UserCreationDate, UserDisplayName, UserLastAccessDate, UserWebsiteUrl, UserLocation, UserEmailHash, UserAge, UserAccountId.

UserProfile Table

Finally, we purpose to have a table so that for each user, there be a whole archive of his activities. To do that, we have done some process in UsersPostJoin Table that will be described in Section 1-4. UsersPostJoin Table Fields are:

id, tag, userAccountId, UserReputation, UserAge, UserVots, postRate, userAnswers, userQuestions, questionRate, answerRate, normalizedEntropyMeasure, normalizedTopicEntropy, topicalReputation, UserCreationDate, UserLastAccessDate, UserDisplayName, UserWebsiteUrl, UserLocation, UserEmailHash.

1-2- ImportTables Class

At first we Downloaded the dump files of three famous Stack Exchange QAC:

1- StackOverflow

2- ServerFault

3- DataSciense

Each Dump File Contains Eight LOG Xml Files and each file is mapped to its corresponding table in Stack Exchange Data Base.

The program creates Users Table and Post Tables by parsing Users.xml and Posts.xml files that are already downloaded. Program runs in a while loop and takes **site name, users.xml and posts.xml address file** as inputs to import information into tables based on schema (which is described in Section 1-1). This trend is continuing till the user stops giving inputs to the program.

1-3- JoinTable Class

In this step, is turn to create a join table between Users & Posts Table to yields a table so that per record represents integrated information about an individual user post; So new information are inserted into userspostsjoin table. It's necessary that for each site (in this program, we mean stackoverflow.com,serverfault.com and datascience.com) calculate AVERAGE VALUE OF "REPUTAION" & "VIEWCOUNT" features **separately** then initialize appropriate variables based on these average values for each site. To calculate the Average reputation and viewCount, you

can make some queries in online Stack Exchange Data Explorer which is available here.
Then per a site, we make a join query between users and posts table and filters users' posts that satisfy our conditions; In the following, Filter Conditions are outlined:

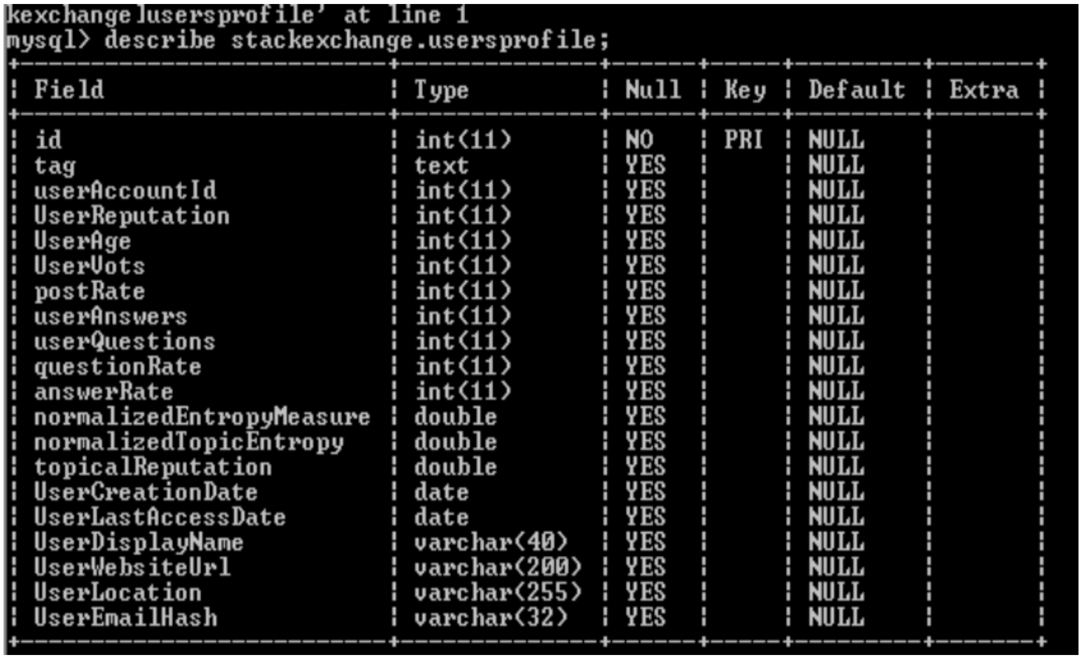
- The posts which are GREATER than Average Score.
- Or the posts which are GREATER than Average ViewCount AND GREATER than Average ViewCount .
- Or Post of users whose their Reputation is GREATER than Average Reputation.

1-4- UserProfile Class

In last step, we are seeking have an archive log for each individual user. As described In Section 1-1 we need some features that are not directly in userspostsjoin table and consequently we need some processes to obtain our desired features; these features are:

- **User Answers**
- **User Questions**
- **Question Rate**
- **Answer Rate**
- **Post Rate**
- **Normalized Entropy Measure**
- **Normalized Topic Entropy**
- **Topical Reputation**

more information can be accessed in [1]; Besides, per a user, we provide a list of tags which are refer him among different Stack exchange Sites. Finally, we could make a profile of appropriate features for each user. Below the UserProfile Table Schema is presented in Figure1:



```
kexchange lusersprofile' at line 1
mysql> describe stackexchange.usersprofile;
+--------------------------+--------------+------+-----+---------+-------+
¦ Field                    ¦ Type         ¦ Null ¦ Key ¦ Default ¦ Extra ¦
+--------------------------+--------------+------+-----+---------+-------+
¦ id                       ¦ int(11)      ¦ NO   ¦ PRI ¦ NULL    ¦       ¦
¦ tag                      ¦ text         ¦ YES  ¦     ¦ NULL    ¦       ¦
¦ userAccountId            ¦ int(11)      ¦ YES  ¦     ¦ NULL    ¦       ¦
¦ UserReputation           ¦ int(11)      ¦ YES  ¦     ¦ NULL    ¦       ¦
¦ UserAge                  ¦ int(11)      ¦ YES  ¦     ¦ NULL    ¦       ¦
¦ UserVots                 ¦ int(11)      ¦ YES  ¦     ¦ NULL    ¦       ¦
¦ postRate                 ¦ int(11)      ¦ YES  ¦     ¦ NULL    ¦       ¦
¦ userAnswers              ¦ int(11)      ¦ YES  ¦     ¦ NULL    ¦       ¦
¦ userQuestions            ¦ int(11)      ¦ YES  ¦     ¦ NULL    ¦       ¦
¦ questionRate             ¦ int(11)      ¦ YES  ¦     ¦ NULL    ¦       ¦
¦ answerRate               ¦ int(11)      ¦ YES  ¦     ¦ NULL    ¦       ¦
¦ normalizedEntropyMeasure ¦ double       ¦ YES  ¦     ¦ NULL    ¦       ¦
¦ normalizedTopicEntropy   ¦ double       ¦ YES  ¦     ¦ NULL    ¦       ¦
¦ topicalReputation        ¦ double       ¦ YES  ¦     ¦ NULL    ¦       ¦
¦ UserCreationDate         ¦ date         ¦ YES  ¦     ¦ NULL    ¦       ¦
¦ UserLastAccessDate       ¦ date         ¦ YES  ¦     ¦ NULL    ¦       ¦
¦ UserDisplayName          ¦ varchar(40)  ¦ YES  ¦     ¦ NULL    ¦       ¦
¦ UserWebsiteUrl           ¦ varchar(200) ¦ YES  ¦     ¦ NULL    ¦       ¦
¦ UserLocation             ¦ varchar(255) ¦ YES  ¦     ¦ NULL    ¦       ¦
¦ UserEmailHash            ¦ varchar(32)  ¦ YES  ¦     ¦ NULL    ¦       ¦
+--------------------------+--------------+------+-----+---------+-------+
```

Figure 1 UsersProfile Table Schema

1-5- Tools Class

This class is supposed to connect to your Data Base, so before running the program you should change some variables based on your DBMS. It's worth noting that this class has Singleton design pattern so that's enough to change data base variables once here (in Tools class).

2- How to Run

RunJoinTable class

First run the **RunJoinTable class** and give (in order) the siteName, Users.xml address file and Posts.xml address file of that site to the program. If there're more than one site Log files, wait till the program imports first files into users and posts tables, then it will ask you whether you have others file to import or no; If yes, then you give the above inputs again and this process are repeated till you import all your desired files, then the program starts making a join table based on Users and Posts Table.

RunUserProfile

In the second phase, run the **RunUserProfile class** to make a user profile for each user. In this part, before running the program, Combine **all Tags.xml file of Sites** (sites that you have selected before in first phase) **into a single file** and then after running the program, give the address file of combined tag file to the program.

Inputs Format

As mentioned before, in this sample project we use three sites data dumps, so if you want to import these three site data into tables, enter

- stackoverflow name for **STACKOVERFLOW SITE**
- serverfault name for **SERVERFAULT SITE**
- datascinece name for **DATASCIEMNSE SITE**

If you need data of other's site,then you should add some codes into **JoinTable Class**.

We will finished this section with an example to make this manual clear; For instance if you have saved your users.xml file in the following address:

E:\Elmira\Expert Finding\Stackoverflow\Users.xml

just give

E:**\Elmira\Expert Finding\Stackoverflow**

address to the program.

1-3- Data Collection Size

USERS

This Data Collection consists of **48821** Users which **7799** are belonged to **STACKOVERFLOW SITE**, **27030** are belonged to **SERVERFAULT SITE** and **13992** are belonged to **DATASCIEMNSE SITE. 927** users have some posts in more than one site; It means, there are exactly **47892** distinct users.

POSTS

This Data Collection consists of **22975** posts which **10023** are belonged to **STACKOVERFLOW SITE**, **6174** are belonged to **SERVERFAULT SITE** and **6778** are belonged to **DATASCIEMNSE SITE.**

USERSPOSTSJOIN

After applying some filters (such as score/reputation condition) just **17762 valid posts of 3309 distinct users** are satisfied our conditions. In Table 1, sites details are presented:

Table 1

SiteName	#Valid Users	#Valid Posts	#Valid Users have Valid Posts
STACKOVERFLOW	**285**	**8572**	**6404**
SERVERFAULT	**5210**	**4100**	**5958**
DATASCIEMNSE	**19707**	**2312**	**5400**

UserSPROFILE

3309 users are saved into users profile table (As final Table) which will be used in future Data Analysis.