Batch (1)

**Building Database Tables using queries through Java code connection.**

**Expected:**

* Student understand SQL
* Students knows how to connect to SQL Server using Java code
* Student knows how to send SQL commands to SQL server through Java
* Student knows how to save/insert data to SQL datatable using java code
* Student knows how to delete/update data through SQL Code.
* Student knows how to query data in SQL table and perform any required joins.
* Student knows how to capture the returned data/records.
* Student knows how to process the returned data/records and then further process them.
* Student knows how to display results on table on console.
* Student knows how to work with Git and GitHub commands and repositories.

**List of Tasks:**

* Build a database in SQL Server – Name: “SchoolMgt”
* Create new repository on your GitHub. Please add Danyal and Afrah to your repository. Name it: “SchoolSystem-SQL”
* Create anew project on your IDE/Eclipse: “SchoolSystem-SQL”
* Connect your project to your SQL Server database “SchoolMgt”.
* Using Java code: Create a method to create a new table in the database with the following structure:   
   Table name: Students  
   with these fields:   
   id: int   
   fname: varchar(8)  
   lname: varchar(8)  
   birthdate: date  
  this method should return Boolean – true if it was successful and false if it failed to create this table.
* Using Java code: Create a method to create a new table in the database with the following structure:   
   Table name: Subjects  
   with these fields:   
   id: int   
   title: varchar(8)  
   desc: varchar(250)  
   pricePerStudent: decimal  
  this method should return Boolean – true if it was successful and false if it failed to create this table.
* Create a function/method which selects top 10 students and then print them on the console.
* Create a function/method which selects top 15 subjects and then print them on the console.
* Create a function/method named: “addFakeStudents” which takes number as parameter and then create fake students data in “Students” table.   
  Example: if it given 100 as argument, it should insert 100 students fake data in this table.   
  How to generate fake data?   
   Id: iterator number + random number  
   fname: yourName concatenated with number  
   lname: yourLastName concatenated with a number  
   birthdate: any random date

This function/method should return (long) which is “How long it took to process everything.

* Create a function/method named: “addFakeStudents” which takes number as parameter and then create fake students data in “Students” table.   
  Example: if it given 100 as argument, it should insert 100 students fake data in this table.   
  How to generate fake data?

Id: iterator number + random number  
 title: “fakeproduct” concatenated with number  
 Desc: “very long desc for a product” concatenated with a number  
 pricePerStudent: 150  
 This function/method should return (long) which is “How long it took to process everything.

* Create a console program with the following menu:   
   1-List Top 100 Students   
   2- List Top 50 Subjects  
   3- Add a new Student  
   4- Add a new Subject  
   5- Load 1000000 student in Students Table  
   6- Load 100000 subjects in Subjects Table  
   7- Exit App

All functions should be performed using the previously created tasks.

How to:

* **Generate Random Number:**

**Using Math.random**

For generating random numbers within a range using Math.random(), follow the steps below:

* Declare the minimum value of the range
* Declare the maximum value of the range
* Use the formula Math.floor(Math.random()\*(max-min+1)+min) to generate values with the min and the max value inclusive.

**Note:** This method can only be used if you need an integer or float random value.

{

    public static void main( String args[] ) {

      int min = 50;

      int max = 100;

      //Generate random int value from 50 to 100

      System.out.println("Random value in int from "+min+" to "+max+ ":");

      int random\_int = (int)Math.floor(Math.random()\*(max-min+1)+min);

      System.out.println(random\_int);

    }

}

* **How to count time elapsed for a process:?**

**long** start = System.currentTimeMillis();

// ... **adding fake data**

finish = System.currentTimeMillis(); **long** timeElapsed = finish - start;