Suppose there are 2 tables (Tb_Batch and Tb_Batch_Item) in a database that hold following records:

Tb_Batch:

Batch_Id	Requestor	Request_Date_Time	Request_Status	Reqeust_For_System
518	bmore	10/27/2013 12:34:23 PM	Queue	ClinOp

Tb_Batch_Item:

Batch_Id	Item	Name	Email	Init_Password	Role	Reason_For_Access
518	1	Susan Smith	ssmith@comany1.com	susan12%#?	SuperUser	Because I am "cool", I can do whatever I want.
518	2	Alex O'Connor	alexoconnor@univ1.edu	itsuniv1	ReadOnly	I need to access report for budget < 1M $
518	3	John J. Peterson	john.p@comany2.com	J.Pe1234!	Auditor	Access to 1) all reports; 2) server system logs for "Audit" and [app]_Access_Log
518	4	Chen, Mei 陈梅	chehmei12@123.com	<:-)>{;=0}	ReadOnly	我负责中国分公司财务

Coding assignment:

1. SQL script (ANSI syntax)
   1.1 Write table creation script to create the 2 tables: Tb_Batch and Tb_Batch_Item
   1.2 Create insert statements to insert data shown above into the tables.

2. Choose one programming language from Java, python, Golang, or C#.NET
   2.1 Create a function to validate password complexity. Make sure the passwords meet following requirements:
      a. Must be at least 8 characters long
      b. Passwords must not contain the user's entire name, or token value. Both checks are not case sensitive:
         The user's name is parsed for delimiters: commas, periods, dashes or hyphens, underscores, spaces, pound signs, and tabs. If any of these delimiters are found, the name is split and all parsed sections (tokens) are confirmed not to be included in the password. Tokens that are less than three characters in length are ignored, and substrings of the tokens are not checked. For example, the name "Erin M. Hagens" is split into three tokens: "Erin," "M," and "Hagens." Because the second token is only one character long, it is ignored. Therefore, this user could not have a password that included either "erin" or "hagens" as a substring anywhere in the password.
      c. Cannot contain the local part and domain part of email address
      d. Must contain characters from 3 of the following 5 character sets:
         - Upper Case Characters: A-Z
         - Lower Case Characters: a-z
         - Numbers: 0-9
         - Punctuation Characters: !'?"-:,;()[]{}
         - Symbols: ~@#$%^&*+=|<>/\

   2.2 Create a function to generate random password that satisfies the password complexity rules specified in section 2.2

   2.3 Write a RESTful API, or a console application to validate each Init_Password provided in table Tb_Batch_Item using function created in 1); if it does not meet password complexity rules, generate a password using function created in 2).

3. Create one csv (comma-separated values) file, and one json file to hold following data:

Name	Email	Init_Password	Role	Reason_For_Access
Susan Smith	ssmith@comany1.com	susan12%#?	SuperUser	Because I am "cool", I can do whatever I want.
Alex O'Connor	alexoconnor@univ1.edu	itsuniv1	ReadOnly	I need to access report for budget < 1M $
John J. Peterson	john.p@comany2.com	J.Pe1234!	Auditor	Access to 1) all reports; 2) server system logs for "Audit" and [app]_Access_Log
Chen, Mei 陈梅	chehmei12@123.com	<:-)>{;=0}	ReadOnly	我负责中国分公司财务

Email back following within the time specified:

- **SQL script file(s) (from section 1)**
- **Source code file(s) (from section 2.2 and 2.3**
- **CLI commands or script to invoke the API or console application, with outputs demonstrating your test execution results.**
- **CSV and JSON files (from section 3)**