C# Microsoft SQL Architect Task:

1. Initially I have created a DataGenerator rest API in Azure DevOps. The code in below URL <https://dev.azure.com/abusayem/_git/EID-DataGenerator/branches>. I have already invited [nirav.patel@empowerid.com](mailto:nirav.patel@empowerid.com) and [Patrick.Parker@empowerid.com](mailto:Patrick.Parker@empowerid.com)
2. This DataGenerator rest api, I have been published in Azure App Service in below URL through Azure DevOps pipelines. <https://eiddatagenerator.azurewebsites.net/swagger/index.html>

Why I Choose this?

* To get control over data generator, some third-party tools can generate the data but we cannot control them.
* We can change or modification whenever we need as per our need in the data generations.

**The main application:**

The architecture I follow is clean architecture (Onion Architecture) with repository patterns included multiple database context. I choose repository patterns to BULK entry for CDC process. It’s have great performance for bulk entry and easy to handle. It’s also have globally handled the unhandled exceptions and showing in the console as log.

**Azure Data Factory Configuration**

1. Firstly, I have created a database as per the requirements. Script added in the repository under DatabaseScript folder.
2. Configured the Azure Data Factory to extract data from a source system which is our DataGenerator Rest API application.
3. In the steps of ADF configuration, I have set the datasets and pipelines. There is one pipeline which I call from the application and this pipeline executes the other pipelines one by one.
4. In the process of pipeline running, ADF transforms the data and loads it into the Azure SQL database which is configured in the ADF datasets. There are not many challenges in the above ADF configuration, we can achieve those steps to write code in the application, but configuration is much easier to handle to transform the data.
5. In the application we can initiate and monitor the Pipeline which is Option 1 in our application.

Azure Cognitive Search (Azure AI Search)

1. For the AI search, I have created Azure AI service where I have also created Index and Indexer.
2. In Index I have mapped the columns which are retrievable, Filterable, Sortable, FaceTable and Searchable. Before that I have created View in the database to pull the all necessary information about Product that mentioned in the requirements.
3. Now we can invoke the AI search through our application which is option 2 in our application.
4. We can search by product name, category, price, description, and date added specific columns or in any columns.

CDC – Change data capture

1. I have named the Primary database as EmpowerIDProductOrder and Secondary Database names is Secondary\_EmpowerIDProductOrder.
2. Now our target is to any change in the Primary database (Insert, Update, Delete), we’ll change in the Secondary Database as well. This is the option 3 in our application.
3. Initially I must enable the CDC in the Primary Database and get the data from changes data using system function and procedure. To achieve this, I have created a stored procedure for the tables and execute from the application. Through repository patterns, get the changed data and make the changes in the secondary database as a BULK entries.

The Challenges in CDC,

I have faced several challenges to implement CDC in the application.

* Normally we used single dbContext in the repository patterns. In CDC we need multiple dbContext and without restart the application we need to change the dbContext Primary to Secondary or vice versa. Though I have implemented, successfully and I have learns something here 😊 .
* In the implementation process, I have separated the dbContext and write a method to change the dbContext. When I needed to change the dbContext, I can called this method through repository instance or unitOfWork instance. It immediately changes the dbContext.

Trade Offs.

1. In CDC, we have to rely on the Microsoft Azure Database changes, I have noticed, sometimes the changes are not correct or I can say the changes operation values not correctly in some operation. As a result sometimes, we face errors when we make the changes in the destination database. We may need to investigate this issue to resolve.
2. I didn’t use the Key Vaults to store the applications settings. I have used **appsettings.json** in the application. Since I have to share the code in the public github, I have encrypted the values and in the application, it’ll decrypt when read.The encryption and decryption handled in the appsettings.json attribute named IsEncrypted. If the value is true, the application will decrypt the provided values. If false, the application just reads the values as it is.