Task 3: Development on Azure Data Factory

1. **Define the steps that you will undertake and deploy on Azure API Management to connect to an on-premises get and post APIs.**
   - Go on Azure API Management of Azure Portal
     i. API Management > APIs > + Add API
   - Create a new API
     i. Select HTTP to create an HTTP API to connect to on-premises APIs
     ii. Fill in the fields as follows
       1. Name: OnPremisesAPI
       2. Web service URL (onPremises URL API)
       3. API URL suffix: /v1/data
   - Define GET and POST operations: Add Operation
     i. GET Operation :
       1. DisplayName : GetData
       2. URL: /data
       3. Method : Select GET
     ii. POST Operation :
       1. Display Name : PostData
       2. URL: /data
       3. Method : Select POST
   - Json code of opration is on the file **task3.json**

2. **Discuss how you will implement an ETL data pipeline on Azure Data Factory to lookup for:**
   - **a record from another tables of Dynamics 365 and update the source data prior to update to dynamics 365.**
     i. Access to Azure Data Factory and create a new pipeline
       1. In the Azure portal, access Azure Data Factory
       2. Go to Author > Pipelines > + New pipeline
       3. Give the pipeline a descriptive name: ETL_Pipeline_Dynamics365
     ii. Add a Copy Activity
       1. In the pipeline, find Copy Data in the activity panel and drag it into the canvas
       2. Rename this activity CopyFromDynamics
     iii. Configure Source (Source Dataset)
       1. Click on the CopyFromDynamics activity
       2. Under Source, click + New to create a Source Dataset
       3. Select Dynamics 365 as the connection type
       4. Configure the Dynamics 365 connection with the required credentials

a. Main Service (Customer ID), Secret, and organization URL
5. Select the source table in Dynamics 365

iv. Configure Destination (Sink Dataset)
1. In Sink, click + New to create a Sink Dataset
2. Choose the type of target database to which the extracted data will be sent
3. Configure the connection to the destination database and select the target table

v. Data transformation
1. If a transformation is required before insertion into Dynamics 365, add a transformation activity
2. Configure the required transformations

vi. JSON configuration for Copy Activity on the Json File **Task3-Development on Azure Data Factory.json**

➢ **an appropriate GUID from another Dynamics 365 and update the source data. Support same with some snapshots.**

i. Add a Lookup Activity
1. In the same pipeline, find the Lookup activity in the activity panel and add it before the copy activity
2. Rename this activity LookupGUID

ii. Configuring the Source for GUID Search
1. In Lookup Activity, configure the source by selecting a dataset that points to the reference table containing the GUIDs
2. Use a query to find the specific GUID based on a matching field
3. GUID query: SELECT GUID FROM ReferenceTable WHERE KeyField = 'value'

iii. Link Search Results with Copy Activity
1. In the copy activity (CopyFromDynamics), go to the activity parameters and select the LookupGUID activity output to transmit the GUID
2. Configure the mapping to use the retrieved GUID as a reference key in the Dynamics 365 table

iv. JSON for Activity Lookup on the Json File **Task3-Development on Azure Data Factory.json**

v. Set Copy Activity to Include GUID
1. In CopyFromDynamics, use the result of LookupGUID to map the GUID to the target field
2. Transformation mapping on the Json File **Task3-Development on Azure Data Factory.json**

3. Create and demonstrate an Azure Data pipeline to create & update existing contacts record from the core system to dynamics 365 based on below ERD. Discuss why you have been using the component.

- **Create a Contact Creation/Update Pipeline**
    i. Access Azure Data Factory and create a new pipeline
        1. In the Azure portal, access Azure Data Factory
        2. Go to Author > Pipelines > + New pipeline
        3. Give the pipeline name: Contact_Update_Pipeline
    ii. Add a Copy Activity
        1. In the pipeline, find Copy Data in the activity panel and drag it into the canvas
        2. Rename this activity 'CopyContactsToDynamics'
    iii. Source Dataset configuration
        1. Click on the CopyContactsToDynamics activity
        2. Under Source, click + New to create a Source Dataset
        3. Select the connection type corresponding to your database system
        4. Configure the connection with the credentials of the source system and select the contact table
    iv. Set Destination
        1. In Sink, click + New to create a Sink Dataset
        2. Select Dynamics 365 as the connection type for sending data to Dynamics 365
        3. Configure the Dynamics 365 connection with the required credentials:
            a. Main Service (Customer ID), Secret, and organization URL
        4. Choose the target entity for contacts in Dynamics 365
- **Use Data Flow to align fields**
    i. Add a Data Flow activity
        1. In the pipeline, search for Data Flow in the activities panel and add it before the Copy activity
        2. Rename this activity TransformContactData
    ii. Create a Data Flow Mapping
        1. Click on the TransformContactData activity, then on Open Data Flow to open the Data Flow editor
        2. In the editor, add a Source to load contact data from the source system
    iii. Configure Field Mapping
        1. In Data Flow, add a Select or Derived Column transformation to map source system fields to Dynamics 365 fields.
        2. Map source system columns (CustomerNumber, FirstName, LastName, Country, Manager, Phone, Owner) to Dynamics 365 fields (CustomerNumber, FirstName, LastName, Country, Manager, Phone, Owner)
        3. JSON configuration for Data Flow Mapping on the Json File **Task3-Development on Azure Data Factory.json**
    iv. Link Transformed Fields to Copy Activity
- **Configure Update Method to ensure data integrity**
    i. Configure Write Mode in Sink (Dynamics 365)

1. In the CopyContactsToDynamics activity, under Sink, configure the write mode to allow existing records to be updated
2. Choose Upsert if you want to insert new records and update existing ones
3. Configure the key field for record identification (e.g. CustomerNumber as the unique key for contacts)

ii. JSON configuration for Copy Activity with Upsert

iii. Add a Validation to check data integrity

1. After the copy activity, add a Validation activity to verify that the data has been inserted/updated correctly in Dynamics 365
2. Set up a query to count the records inserted/updated and compare them with the records in the source system to ensure integrity

4. Create an Azure Data Pipeline to migrate Leads & tasks from on-premises Dynamics 365. One key requirement is to retain the creationOn, creationBy and modifiedBy values and also maintain the relationship with existing contacts lookup and Team sharing on the records.

- Create a Migration Pipeline in Azure Data Factory
- Configuring Sources and Destinations

i. Add a Copy Activity for Leads

1. Drag a Copy Data activity into the pipeline and Name it Copy_Leads

ii. Configure the Source for Leads

1. In the Copy_Leads activity, under the Source tab, click + New to create a new dataset
2. Select the appropriate source type SQL Serverm,CSV, ...
3. Configure the connection with the required credentials
4. Select the table or source file containing the Leads data

iii. Configure Target for Leads

1. Under the Sink tab of the Copy_Leads activity, click + New to create a destination dataset
2. Choose Dynamics 365 as the connection type
3. Configure the connection with Dynamics 365 credentials
4. Select the target entity, lead

iv. Repeat Steps for Tasks

1. Add another Copy Data activity named Copy_Tasks
2. Configure sources and destinations in the same way as for Leads, by selecting the task entity in Dynamics 365

- Preserving Champs ‘createdon’, ‘createdby’ and ‘modifiedby’

i. Add Custom Fields in Dynamics 365

1. In Dynamics 365, go to Settings > Customize > Customize system
2. Select the lead entity and add custom fields, for example, **historical_createdby** and **historical_modifiedby**

3. Repeat the operation for the task entity

ii. Map Fields in the Copy Activity
    1. In the Copy_Leads activity, under the Mapping tab, map the source fields to Dynamics 365 fields:
        a. createdon → overriddencreatedon
        b. createdby → historical_createdby
        c. modifiedby → historical_modifiedby
    2. Repeat the mapping for the Copy_Tasks activity

- Maintain relationships with Contact and Team entities via Lookup

i. Add Lookup Activities for Contacts and Teams
    1. Before copying activities, add two **Lookup** activities named **Lookup_Contacts** and **Lookup_Teams**
    2. Configure them to retrieve GUIDs for contact and team entities from Dynamics 365

ii. Configure Lookup Activity Sources
    1. For **Lookup_Contacts**, configure the source to query the **contact** entity and retrieve the fields **contactid** and **fullname**
    2. For **Lookup_Teams**, configure the source to query the **team** entity and retrieve the **teamid** and **name** fields

iii. Using Lookup Results in Copy Activities
    1. map relationship fields
    @activity('Lookup_Contacts').output.value[?(@.fullname == source.contact_name)].contactid
    2. Repeat the process for relationships with team entities in the Copy_Tasks activity