Lab 5 - Pipelines

Requirements

In order to start the lab, it is essential that Lab4 has been completed.

Objective

It's time now that we start moving data from point A to point B. We do this by creating a pipeline with a copy activity. The activity ensures that a Source and Sink (Destination / Target) can be connected to each other and that a literal pump & dump can take place. Follow the instructions step by step.

Assignment 1 - Database pipelines

1. Next to **Pipeline**, you currently see a 0. When you hover over the **Pipeline** box with your mouse, an option with **3 dots** (Pipeline Actions) appears on the right side. Click on **Pipeline Actions** and then click on **New Pipeline**.

2. Give the Pipeline a clear name. The recommended format is to start with `PL_`, the type of activity, (schema), the table/file name, source, target, and end with `_environment`. If you have a pipeline that orchestrates multiple pipelines, you can keep the format more general.

   - Practical example: `PL_copy_visits_clubmanager_to_datalake_prd`
   - Training example: `PL_copy_address_training`

3. On the left side we see a list of **Activities** categories. Click on **Move & transform**. 2 options will appear: **Copy data** and **Data flow**. We will ignore the Data flow for today. Click and drag the **Copy data** to the canvas in the middle of the screen.

4. Give the Activity a clear name.

5. Click on the **Source** tab. You will be asked to specify a **Source dataset**. Click on it and choose the Dataset for **Address** from the **sqldb-source**.

6. Click on the **Sink** tab. You will be asked to specify a **Sink dataset**. Click on it and choose the Dataset for **Address** from the **sqldb-target**.

7. Various options will appear, including the option for a **Pre-copy script**. Here you can execute SQL code before the Copy activity starts moving data. Since we want to be able to run the pipeline multiple times without getting duplicate data, you can fill in / paste the following:

   ```
   Truncate table [Stg].[Address]
   ```

8. Click on the **Mapping** tab. You will see a button with **Import schemas**, click on it.

   > Remember how Datasets can list the columns and datatypes? By running this process, matching columns are linked to each other. This ensures that the data ends up in the right

> column. This is useful for a table where there is a 1 to 1 mapping. If you are doing multiple tables at once, there are other options.

9. Repeat steps 1 to 8, but now also for **ProductCategoryDiscount** and **SalesPersonal**. For this you can use the following **Pre-copy scripts**:

    - `Truncate table [Stg].[ProductCategoryDiscount]`
    - `Truncate table [Stg].[SalesPersonal]`

10. When all 3 pipelines have been created, create a new pipeline named: `PL_copy_Master_Training`.

11. Under the **Activities** tab there is an option called **General**, which contains an **Execute Pipeline** activity. Drag 3 onto the canvas.

12. Rename each pipeline one by one to the 3 pipelines you have created before for **Address**, **SalesPersonal**, and **ProductCategoryDiscount**. If the name is too long for what is allowed, make it shorter for now.

13. Go to the **Settings** tab for each

pipeline and choose the corresponding pipeline. Do this for all 3 pipelines.

14. At this moment, all 3 pipelines would run in parallel, which should be easy as there is no dependence on each other. Despite this, we are going to make them sequential. Click on one of the 3 pipelines. On the right side of the pipeline block you will see **four squares with symbols**. Click on it and a list with the following options will appear:

    On Skip = When the pipeline is skipped, move on to the next one.

    On Success = When the pipeline has run successfully, move on to the next one.

    On Failure = When the pipeline fails, move on to the next one.

    On Completion = When the pipeline is finished, regardless of success or failure, move on to the next one.

    Click and drag the **green block** to one of the other pipelines, and then do it again for another pipeline. You should now have connected all 3 pipelines to each other with 2 **green arrows**.

15. Click on the **Blue button** with the text **Publish all** and then on the **Publish** button. By publishing, the other changes become **Live**, and can be used.

16. Hooray! Your first pipelines are ready. Now we want to run the pipeline, which can be done in different ways:

    - In the pipeline screen, you see a `Play button` with the text **Debug**. This allows you to run the pipeline as you have created it now. Even if you haven't saved or published yet, the pipeline will be executed as you see it on your screen now.
    - Next to **Debug** we see a **Lightning bolt** with the text **Add trigger**. If you click on this, you get the option for **Trigger now**, with this you run the pipeline as it is published. Click **Trigger now** and an option would appear to fill in parameter value, since there are none we can click **OK**.

17. Wait until you get the message at the top right of the screen saying that the pipeline has run successfully. Run the pipeline again via the **Debug button**. You will see that the information about the pipeline running appears at the bottom of the screen.

Assignment 2 - Monitoring

In the "monitoring" section, you can not only view how previous pipelines have run, but also send notifications when certain conditions are met.

1. Click on the meter icon (**Monitor**) on the left side. You will now immediately go to **Pipeline runs**, and will see 2 options in the form of **Triggered** and **Debug** in the horizontal navigation bar. In both tabs, the `PL_copy_Master` pipeline should be present, as well as the corresponding underlying pipelines.

2. Click on one of the underlying pipelines, in one of the two tabs. Just like when running the Debug variant, you see a line of information about the run pipeline. Hover your mouse over the name of the activity at the bottom of the screen, now 2 options appear: **Input**, **Output**, and **Details**.

3. Click on **Input**, now you will see a piece of JSON code from which you can read which column from the source went to which column in the sink. In this, you can also see information if you fetch specific data using a query, parameters, variables, and more. Close the **Input Tab** by clicking on the **Cross**.

4. Click on **Output**, here too you will see a piece of JSON code. The **Output** contains information about the run, such as: How long did it take, how many rows were read and how many were transferred, and more. Close the **Output Tab** by clicking on the **Cross**.

5. Click on **Details**, you see a visual representation of the **Output**.
   Close the **Details** by clicking on the **Cross**.

6. On the left side, we see **Notifications** with the option **Alerts & metrics**. Click on this.

7. In the horizontal navigation bar, we see the option **New alert rule**. Click on this.

8. We're going to create an Alert Rule that sends a notification when the pipeline encounters an error. Give the **Alert rule name** a clear name that covers the situation (e.g., `Alert on error`).

9. Under **Severity**, multiple options are possible:

   Sev 0 = Critical

   Sev 1 = Error

   Sev 2 = Warning

   Sev 3 = Informational

   Sev 4 = Verbose

   For our purpose, choose **Sev0**.

10. Click on **Target criteria** on the **Add criteria**. A long list of options will appear for different types of metrics that can be reported on. Choose the **Succeeded pipeline runs metrics** and click on **Continue**.

11. Click on the **Values** option at **Name** and choose the `PL_copy_Master` pipeline.

12. The other settings can remain as they are. Then click on **Add criteria**.

13. Click on **Configure Email/SMS/Push/Voice notification** on **Configure notification**.

14. A new **Action group** will need to be created. This is a group in which people can be placed to be notified about the rule you have created. Fill in a clear name at **Action group name** and provide a recognizable abbreviation of the group name at **Short name**.

15. Click on **Notifications** on **Add notification** and give the **Action name** a clear name. Then choose the **Email** option at **Select which notifications you'd like to receive** and enter an email address to which you currently have access. Other options may also be chosen so that you can try them out. When you have added everything you want, click on **Add notification**.

16. Then click on **Add action group

**. If this goes wrong, let the trainer know.

17. Click on **Create alert rule**.

18. Go back to **Pipeline runs** and the **Triggered** tab, hover your mouse over the name of the `PL_copy_Master`. A `Play button with arrows` (rerun) will appear, click on it. Wait until the pipeline is ready again, after just over a minute you should receive an email and/or other notifications.

Assignment 3 - Parameters and Variables

With the help of parameters, you can make your pipeline more dynamic. For example, by only fetching data that has changed after a certain date/time.

1. Click on the **Pencil** (Author) on the left and then go back to the pipeline for **Address**.

2. In the bar at the bottom, you see the **Parameters** tab, click on this if you are not already on it.

3. Click on **New**, a new parameter is created. Enter the following for **Name**: **ModifiedDate**. The **Type** can remain set to **String**.

4. Click on the block for the **Copy data**. Then click on the **Source** tab and select the **Query** option under **Use query**.

5. A Query field will now appear, click on this. The option **Add dynamic content** appears below the field, click on this.

6. Type or paste the following query into the field:

```
SELECT * FROM [SalesLT].[Address] WHERE ModifiedDate >=
'@{formatDateTime(pipeline().parameters.ModifiedDate,'yyyy-MM-dd')}'
```

7. When you now click on **Preview data**, you are asked to enter a value. Enter **1900-01-01** here to test. Then click on **OK**.

8. Now go to the `PL_copy_Master` and click on the "execute pipeline activity" for **Address**. In the **Settings** tab, you will see that a **Value** for the **ModifiedDate** parameter is requested. We are not going to fill this manually, but with the help of a variable.

9. Click on the canvas and then on the **Variables** tab. Create a new variable by clicking on **New**.

10. Name the variable `FilterDate`

11. From the list of **Activities**, click on the **General** option. Click and drag **Set variable** onto the canvas.

12. Connect the **green block** with the **Address** pipeline.

13. Click on the **Set variable** block and give it a clear name, for example "Set ModifiedDate".

14. Go to the **Settings** tab and choose **FilterDate**. It is now possible to enter a value. Enter the following here: **2007-01-01**.

15. Then click again on the **Address** pipeline and then on the **Settings** tab.

16. Click on the field under **Value** and then click on **Add dynamic content**.

17. In the new screen, go to the **Variables** tab. Click on the **FilterDate** variable and then on **OK**.

18. Click on the **Blue button** with the text **Publish all** and then on the **Publish** button.

19. Click on **Add trigger** then **Trigger now** and finally click on **OK**.

20. Click on **Monitor** (the meter) on the left. Go to **Pipeline runs** if it doesn't open immediately. You should now see new pipelines running and in the pipeline of **Address** you should now see an **[@]** under the **Parameters** column. Click on this, you should see the value that you put in your variable.

Table of Contents