Databricks Workflows Exercise

In this exercise you will create your own Databricks Workflow to orchestrate the ETL pipeline.

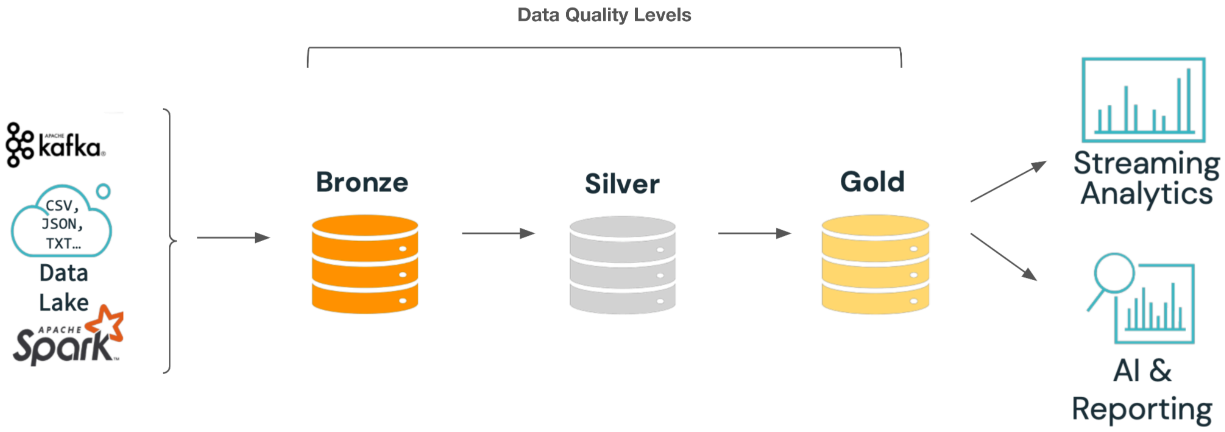
⚠️ **If you need any help with this exercise, please raise your hand and we will gladly assist you!** ⚠️

Step 1: Familiarize yourself with the ETL pipeline

The pipeline consists of three steps, represented by three Databricks notebooks.

1. prepare_bronze_table.py
2. prepare_silver_table.py
3. prepare_gold_table.py

These notebooks follow the medallion architecture where data flows through bronze, silver and gold tables. The first notebook loads the raw data (for the sake of this workshop, this is just a copy of another table). The second notebook cleans the data and the third notebook prepares the data for LLM processing.



Step 2: Get familiar with the Workflow UI

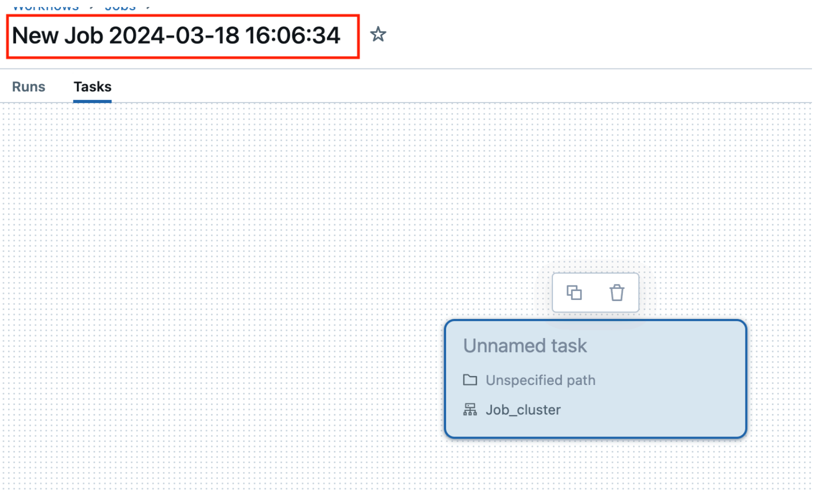
In the side panel of the workspace, navigate to **workflows**. This page has three sections:

1. **Jobs**: here, you will see all jobs that have been defined.
2. **Job runs**: here, you can see all previous runs of the jobs.
3. **Delta Live Tables**: here, you can see all DLT Pipelines that have been defined.

Step 3: Build your own workflow

1. Click on **Create job** on the top right, so start setting up the new job.

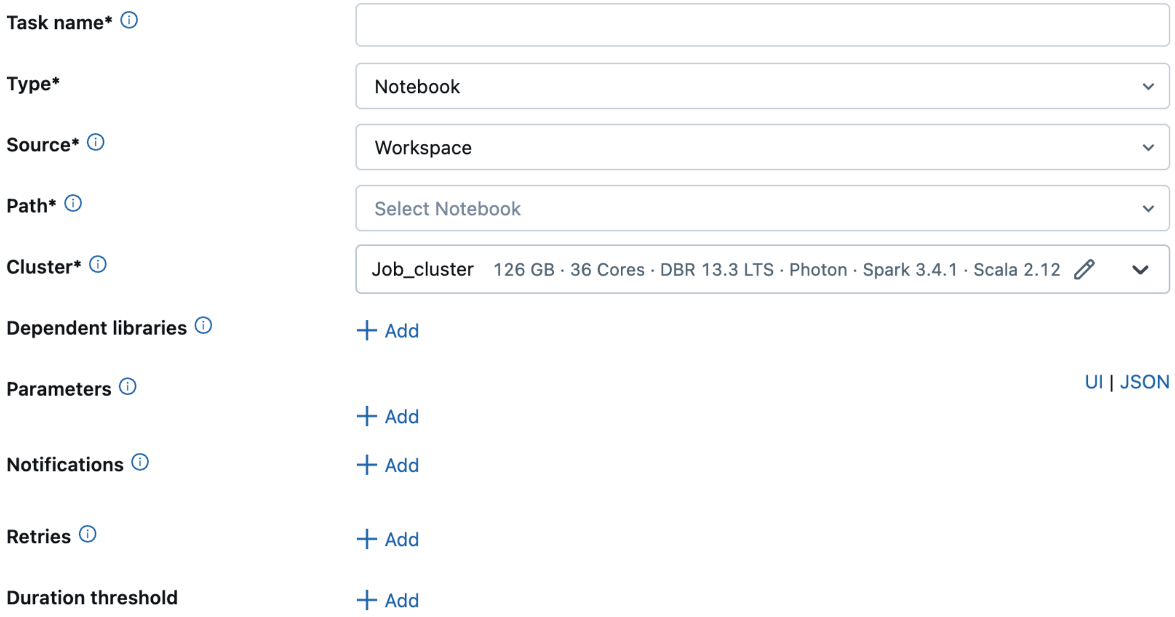
2. Give the job a name by clicking on the current name:



Name the workflow **<your-username>_etl_job**. For example: **john_doe_etl_job**

Step 4: Define the first task

Creating a job already created a first task with a configuration block:



Task name* ⓘ	
Type*	Notebook
Source* ⓘ	Workspace
Path* ⓘ	Select Notebook
Cluster* ⓘ	Job_cluster 126 GB · 36 Cores · DBR 13.3 LTS · Photon · Spark 3.4.1 · Scala 2.12
Dependent libraries ⓘ	+ Add
Parameters ⓘ	UI | JSON + Add
Notifications ⓘ	+ Add
Retries ⓘ	+ Add
Duration threshold	+ Add

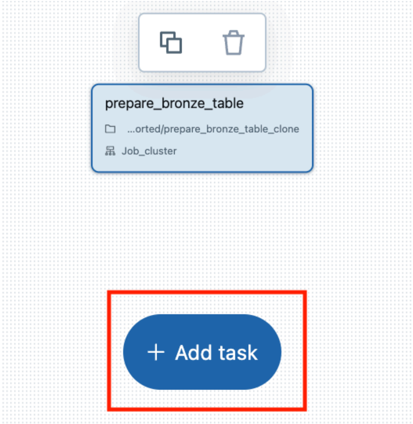
Now, we will configure this task:

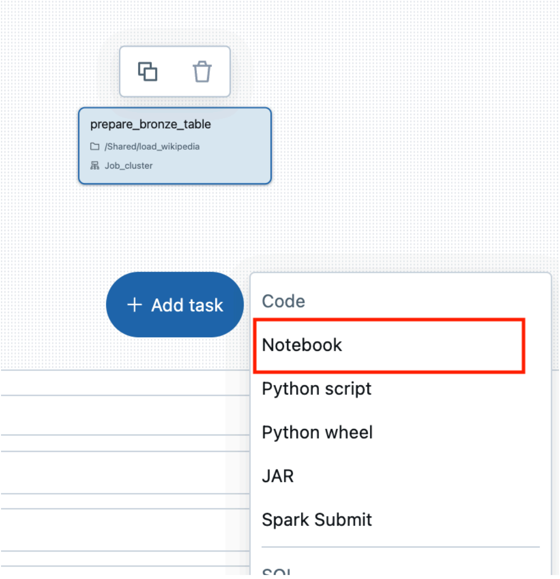
1. First, set the Task name to **prepare_bronze_table**
2. Type: **Notebook**
3. Source: **Workspace**
4. Path: **<your-home-folder>/notebooks/Workflows Exercise/prepare_bronze_table.py**
5. Cluster: **<the all purpose cluster you created in the previous exercise>**

Complete the task definition by clicking on **Create task** in the bottom right corner.

Step 4: Define the second task

You can add another task by clicking on the **Add task** button in the workflow lineage view. In the popup, select **Notebook.**





Configure the task with the following settings:

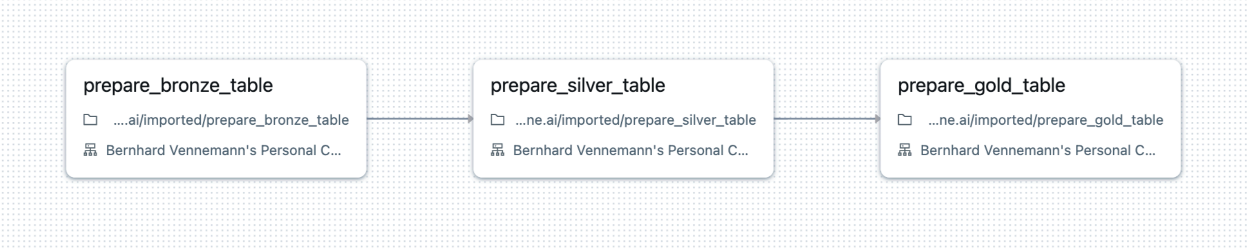
1. Task name: **prepare_silver_table**
2. Type: **Notebook**
3. Source: **Workspace**
4. Path: **<your-home-folder>/notebooks/Workflows Exercise/prepare_silver_table.py**
5. Cluster: **<the all purpose cluster you created in the previous exercise>**
6. Depends on: **prepare_bronze_table**
7. Run if dependencies: **All succeeded**

Notice that this task is now dependent on the **prepare_bronze_table** task and only runs if the previous task was executed successfully.

Step 4: Define the third task

Repeat the previous step to create a task **prepare_gold_table** based on the notebook **prepare_gold_table.py**

After completing this step, your workflow should look like this:



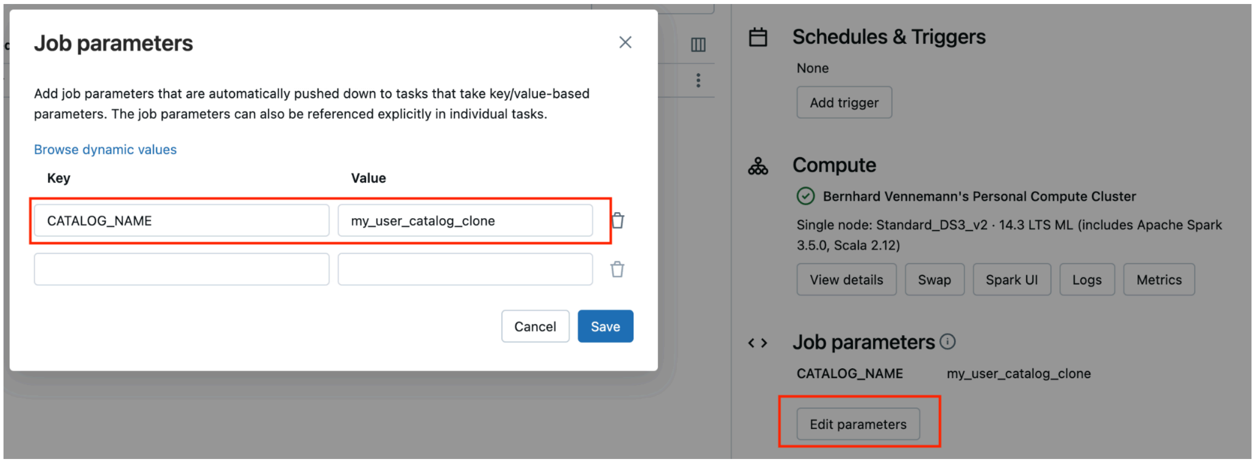
prepare_bronze_table
....ai/imported/prepare_bronze_table
Bernhard Vennemann's Personal C...

prepare_silver_table
...ne.ai/imported/prepare_silver_table
Bernhard Vennemann's Personal C...

prepare_gold_table
...ne.ai/imported/prepare_gold_table
Bernhard Vennemann's Personal C...

Step 5: Set the Job parameter

The notebooks that make up the three tasks expect a parameter **CATALOG_NAME** that defines which catalog should host the newly created bronze, silver and gold tables. Parameters can be defined on the task level and the job level. Since all tasks will share this parameter, we will define the **CATALOG_NAME** parameter on the job level.
You can do so my clicking on the button **Edit parameters** in the right hand panel in the **Job parameters** section:



Job parameters

Add job parameters that are automatically pushed down to tasks that take key/value-based parameters. The job parameters can also be referenced explicitly in individual tasks.

Browse dynamic values

Key	Value
CATALOG_NAME	my_user_catalog_clone

Schedules & Triggers
None

Compute
Bernhard Vennemann's Personal Compute Cluster
Single node: Standard_DS3_v2 · 14.3 LTS ML (includes Apache Spark 3.5.0, Scala 2.12)

Job parameters
CATALOG_NAME my_user_catalog_clone

1. Define a new parameter with the Key **CATALOG_NAME**
2. For the Value, provide the **name of the catalog that you created** in the previous exercise.
3. Confirm with **Save**

Step 6: Run the workflow

Finally, we are ready to run the workflow.

1. Run the workflow by clicking on **Run now** in the top right corner of the screen.
2. Navigate to **Runs** (under the workflow name) and click on the current run (link in the column **Start time**)
3. You will see the three tasks and the status of the run (starting with the first task and working its way through the pipeline until the gold table has been computed). This should take a couple of minutes.
4. You can click on the individual tasks to see how the notebooks are being executed.