ANZ Databricks Lakehouse Bootcamp

SQL Analytics Student Workbook

Working with IoT Data

As part of this section of the workshop, we will be working with simulated IoT data derived from devices within three fictional power plants. We will be using two key tables in order to visualise data curated as part of the initial Data Engineering exercise completed earlier. Ultimately, we want to create a dashboard that can be used by our business to view the status of the devices in the three power plants that we are managing.

Your task will be to create a dashboard that provides an overview of the health of all three power plants using the ingested data.

Run Queries and Create Visualizations

Step 1: Access SQL Analytics

1. At the bottom of the sidebar within the Databricks web user interface, click the  app switcher icon and select 'SQL Analytics'.
2. Tap the 'Queries' icon in the sidebar.
3. Click the 'New Query' button.
4. Your SQL endpoint should already be selected. If it is not, please use the menu to select an appropriate SQL endpoint.

5. In the box below the endpoint, click the icon and select the database used as part of the earlier Data Engineering exercise (this database will be your username followed by **_sensor_iot_db**), alternatively you should be able to use **db_sensor_iot_db**:

New Query

student1_sql_endpoint

db_sensor_iot_db

Filter tables & columns...

- current_readings_labeled
- current_readings_unlabeled
- dim_plant
- readings_agg
- readings_stream_source
- sensor_readings_gold
- sensor_readings_historical_bronze
- sensor_readings_historical_interpolations
- sensor_readings_historical_silver
- sensor_readings_historical_silver_clone
- sensor_readings_historical_silver_with_constraints

Step 2: Query Batch IoT Data

Please note: In this step you will be creating and saving a query that will appear in the shared Databricks workspace. In order to avoid confusion, please **prepend** the saved name of any query with your initials and any 3 digit number. For example, Han Solo might name his saved query: **hs130 Device Status by Power Plant**. Later, you can then use the search functionality to identify your queries.

We can start to identify device status on a per power plant basis with a basic query on our Gold sensor readings table.

- Name this query 'Device Status by Power Plant'. *Don't forget to prepend that name with the identifier you chose using the pattern in the note above.*
- Write a query to show the operational status of devices grouped by power plant.
- Use the editor and the schema browser to try writing your own query to accomplish this, **or** copy and paste the following code into the editor (make sure to replace 'db_sensor_iot_db' with the relevant database name derived as part of Step 1).
- Execute the query.

```
SELECT
  plant_id,
  device_operational_status,
  count(*) as count
FROM
  db_sensor_iot_db.sensor_readings_gold
GROUP BY
  plant_id,
  device_operational_status
```

Your table should look like this:

plant_id	device_operational_status	count
2	RESETTING	284,233
1	FAILURE	280,525
2	DESCENDING	250,471
1	RESETTING	238,679
1	DESCENDING	242,437
1	RISING	342,751
2	IDLE	332,591
1	HIGH	265,190
1	IDLE	323,784
1	NOMINAL	288,300
3	HIGH	313,989

Step 3: Add Visualization

- Click the ‘Add Visualization’ button.
- Chart is the default visualization type. We are creating a chart, so do not change that selection.
- Change the ‘Visualization Name’ field to read ‘Device Status by Power Plant’.
- Ensure that the ‘Chart Type’ field reads ‘Bar’.
- In the ‘X Column’ field, choose ‘plant_id’.
- In the ‘Y Columns’ field, choose ‘count’.
- In the ‘Group by’ field, choose ‘device_operational_status’.
- Click the ‘Save’ button.

Step 4: Query Streaming IoT Data (optional)

Important: The 'readings_agg' table will only exist if you have completed the optional streaming exercise. If you have not completed the streaming exercise you can use db_sensor_iot_db.readings_agg or simply skip this query along with its associated visualization.

Please note: In this step you will be creating and saving a query that will appear in the shared Databricks workspace. In order to avoid confusion, please **prepend** the saved name of any query with your initials and any 3 digit number. For example, Han Solo might name his saved query: **hs130 Observations by Device Type**. Later, you can then use the search functionality to identify your queries.

Let's see if we can get an indication as to the number of observations made across the three power plants on a device type basis.

- Ensure that you create a new query and name it 'Observations by Device Type'. *Don't forget to prepend that name with the identifier you chose using the pattern in the note above.*
- Write a query to show the total number of readings by device type (note that you will need to use the 'readings_agg' table).
- Use the editor and the schema browser to try writing your own query to accomplish this, **or** copy and paste the following code into the editor (make sure to replace 'db_sensor_iot_db' with the relevant database name derived as part of Step 1).
- Execute the query.

```
SELECT
  device_type AS `Device Type`,
  sum(count) AS `# of Observations`
FROM
  db_sensor_iot_db.readings_agg
GROUP BY
  1
```

Step 5: Add Visualization

- Click the ‘Add Visualization‘ button.
- Chart is the default visualization type. We are creating a chart, so do not change that selection.
- Change the ‘Visualization Name’ field to read ‘Observations by Device Type’.
- Ensure that the ‘Chart Type’ field reads ‘Pie’.
- In the ‘X Column’ field, choose ‘Device Type’.
- In the ‘Y Columns’ field, choose ‘# of Observations’.
- Click the ‘Save’ button.

Step 6: Visualise Sensor Readings Data

Please note: In this step you will be creating and saving a query that will appear in the shared Databricks workspace. In order to avoid confusion, please **prepend** the saved name of any query with your initials and any 3 digit number. For example, Han Solo might name his saved query: **hs130 Sensor Readings Data**. Later, you can then use the search functionality to identify your queries.

Let's create a query to see if we can leverage our Gold dataset in order to visualise sensor readings data.

- Ensure that you create a new query and name it 'Sensor Readings Data'. *Don't forget to prepend that name with the identifier you chose using the pattern in the note above.*
- Write a query to show all of the readings as well as the associated plant ID and device status across the entirety of our dataset.
- Use the editor and the schema browser to try writing your own query to accomplish this, **or** copy and paste the following code into the editor (make sure to replace 'db_sensor_iot_db' with the relevant database name derived as part of Step 1).
- Execute the query.

```
SELECT
  plant_id,
  device_operational_status,
  reading_1,
  reading_2,
  reading_3
FROM
  db_sensor_iot_db.sensor_readings_gold
```

Step 7: Add Visualization

- Click the 'Add Visualization' button.
- Chart is the default visualization type. We are creating a chart, so do not change that selection.
- Change the 'Visualization Name' field to read 'Sensor Readings by Status'.
- Ensure that the 'Chart Type' field reads 'Scatter'.
- In the 'X Column' field, choose 'device_operational_status'.

- In the ‘Y Columns’ field, choose ‘reading_1’, ‘reading_2’ and ‘reading_3’.
- In the ‘Group by’ field, choose ‘plant_id’.
- Click the ‘Save’ button.

Step 8: Add Another Visualisation

- Click the ‘Add Visualization’ button once more.
- Amend the ‘Visualization Type’ field to read ‘Word Cloud’.
- Change the ‘Visualization Name’ field to read ‘Sensor Readings Status’.
- In the ‘Words Column’ field, choose ‘device_operational_status’.
- Click the ‘Save’ button.

Create and Customize Your Dashboard

Now let's work on building our dashboard so that we can share our visualizations with the business.

Step 1: Create New Dashboard

- Click on the Dashboards tab, and then click the 'New Dashboard' button.
- Name your new dashboard 'Power Plant Status' (ensuring that you prepend the title with your unique identifier).

Step 2: Add Visualization

- Click on the 'Add Visualization' button.
- Type the identifier you selected into the search bar.
- Select 'Device Status by Power Plant'.
- Under Choose Visualization, select the chart titled 'Device Status by Power Plant'.
- Click on the 'Add to Dashboard' button.

Step 3: Add Second Visualization (if created)

- Click on the 'Add Visualization' button.
- Type the identifier you selected into the search bar.
- Select 'Observations by Device Type'.
- Under Choose Visualization, select the chart titled 'Observations by Device Type'.
- Click on the 'Add to Dashboard' button.

Step 4: Arrange and Resize

Once you've added the widgets above, you can use drag-and-drop functionality and resizing tools to arrange them. Drag, drop and resize the two visualizations so that they are placed next to one other.

Step 5: Add Remaining Visualizations

Follow the instructions to add and arrange the remaining visualizations and add the visualizations according to the table below:

Query	Visualization Type	Visualization Title
Sensor Readings Data	Chart	Sensor Readings by Status
Sensor Readings Data	Word Cloud	Sensor Readings Status

Parameterized Queries

Now, let's explore some more features in the query editor that will allow us to create dashboards that our users can interact with. This dataset represents IoT sensor data accumulated over time. So far, we have created queries and visualizations that capture metrics that include all of the data across all of our power plants. Now, we'll amend our existing queries and visualizations to allow the viewer to drill down to some relevant details.

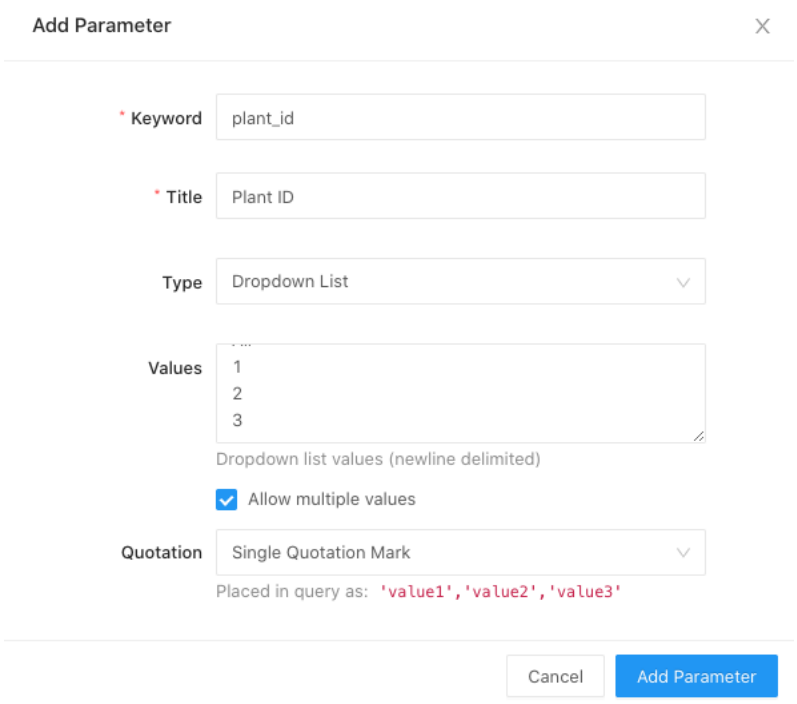
For this, we will write a parameterized query. Parameterized queries allow the viewer to substitute values into a query at runtime without having to edit the query source.

The next few steps will walk through an example of how you might use parameterized queries in practice.

Step 1: Add a Parameter

Let's update an existing query to incorporate a query parameter that we can then leverage as part of the dashboard that we have created.

- Navigate to the 'Queries' tab and type the identifier you selected into the search bar using the 'Search Queries' text box on the right-hand side of the user interface.
- Click on the query titled 'Device Status by Power Plant'.
- Tap the 'Add New Parameter' button ('{{ }}').
- Enter 'plant_id' in the 'Keyword field.
- Update the 'Title' field to read 'Plant ID'.
- Amend the 'Type' field to read 'Dropdown List'.
- In the 'Values' field, enter 'All' and press the 'Enter' key.
- Type '1' in the subsequent line, followed by '2' and, finally, '3'.
- Tick the 'Allow multiple values' checkbox.
- Update the 'Quotation' field to read 'Single Quotation Marks':



- Click the 'Add Parameter' button.
- You should now see the query parameter visible in the UI and represented in the query editor text box.

- Update your query to incorporate the 'plant_id' query parameter that has been created (using a WHERE clause) **or** copy and paste the following code into the editor (make sure to replace 'db_sensor_iot_db' with the relevant database name derived as part of the first section).
- Select 'All' from the 'Plant ID' field and tap the 'Apply Changes' button.
- Click 'Save' to save the changes to the query.

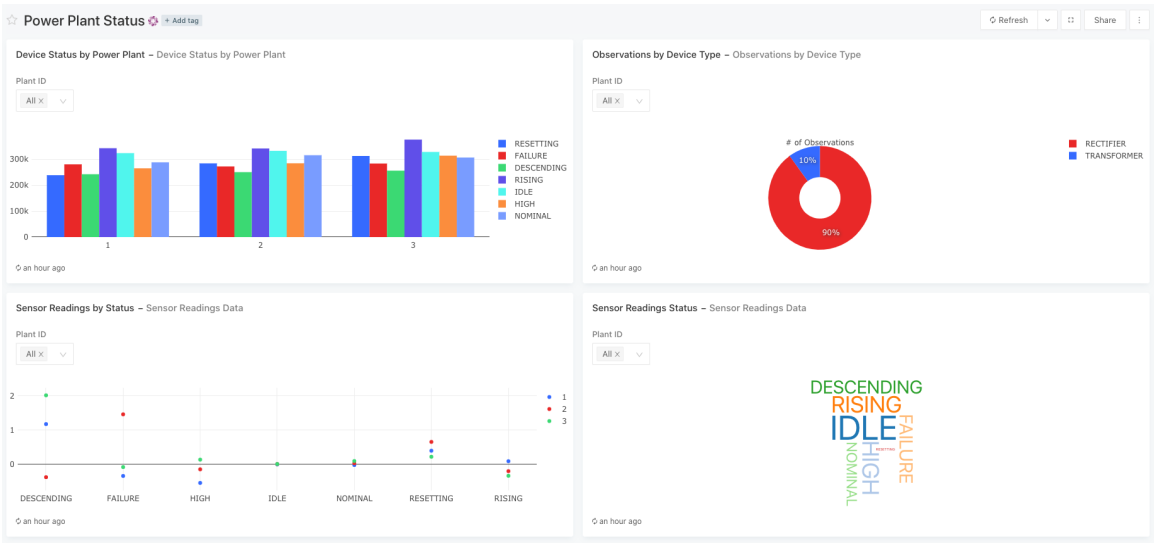
```
SELECT
  plant_id,
  device_operational_status,
  count(*) as count
FROM
  db_sensor_iot_db.sensor_readings_gold
WHERE
  (
    'All' IN ({{ plant_id }})
    OR plant_id IN ({{ plant_id }})
  )
GROUP BY
  plant_id,
  device_operational_status
```

Step 2: Update Other Queries to Include the Parameter

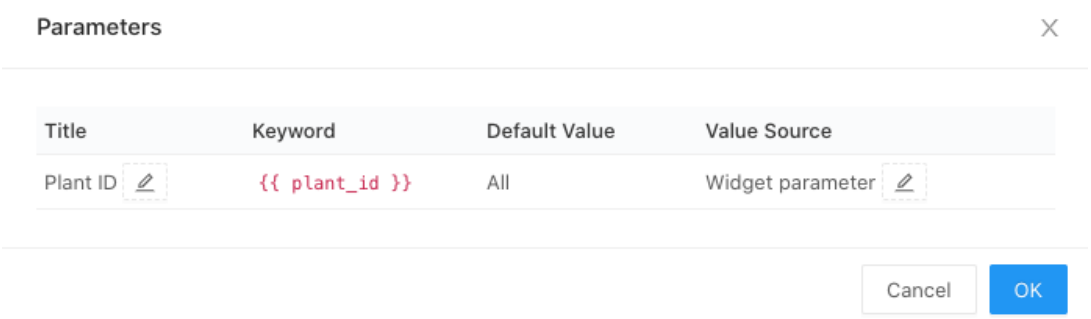
- Navigate back to the 'Queries' tab and type the identifier you selected into the search bar using the 'Search Queries' text box on the right-hand side of the user interface.
- Click on the query titled 'Observations by Device Type' (if created).
- Following the steps incorporated as part of Step 1, add the 'Plant ID' query parameter to the query (complete with Dropdown List values) and add the WHERE clause. Save your updated query.
- Once saved, navigate back to the 'Queries' tab and type the identifier you selected into the search bar using the 'Search Queries' text box on the right-hand side of the user interface.
- Click on the query titled 'Sensor Readings Data'.
- Following the steps incorporated as part of Step 1, add the 'Plant ID' query parameter to the query (complete with Dropdown List values) and add the WHERE clause. Save your updated query.

Step 3: Create Dashboard Filters

- Click on the Dashboards tab, and type the identifier you selected into the search bar using the 'Search Queries' text box on the right-hand side of the user interface.
- Click on the 'Power Plant Status' dashboard and note that you should now see the query parameters that you have created within each dashboard widget:



- Tap the ellipsis button on the right-hand side of the user interface and click the 'Edit' menu option.
- Select the 'Use Dashboard Level Filters' check-box.
- Hover over the 'Device Status by Power Plant' widget within your dashboard, click the ellipsis button that appears in the upper right corner of the widget and click the 'Edit Parameters' menu option:



- Tap the edit icon next to the text that reads ‘Widget parameter’ and, from the dialogue box that is displayed, select the ‘New dashboard parameter’ option and click the ‘OK’ button.
- Tap the ‘OK’ button once more and you should now see the ‘Plant ID’ filter represented on the dashboard itself (and no longer visible within the dashboard widget).
- Hover over the ‘Observations by Device Type’ widget (if created) within your dashboard, click the ellipsis button that appears in the upper right corner of the widget and click the ‘Edit Parameters’ menu option.
- Tap the edit icon next to the text that reads ‘Widget parameter’ and, from the dialogue box that is displayed, select the ‘Existing dashboard parameter’ option, ensure that ‘plant_id’ is populated in the ‘Key’ field and click the ‘OK’ button.
- Repeat the two steps above for the remaining two dashboard widgets.
- Once done, tap the ‘Done Editing’ button.
- Using the ‘Plant ID’ dashboard filter, experiment with contextualising the visualisations based on the appropriate plant that has been selected from the drop-down menu.