

ANZ Databricks Lakehouse Bootcamp

Databricks SQL Student Workbook



Working with APJuice Data

As part of this section of the workshop, we will be working with simulated APJuice data derived from retail juice stores across multiple cities in ANZ. We will be using two key fact tables and 3 dimension tables in order to visualise data curated as part of the Data Modelling exercise completed earlier. Ultimately, we want to create a dashboard that can be used by our business to view the status of the APJuice business in ANZ.

Run Queries and Create Visualizations

















Step 1: Access Databricks SQL

1. Click the persona navigator on the top left and select 'SQL'.
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 schema browser panel select the database used as part of the earlier Data Modelling exercise (this database will be your username followed by **ap_juice_db**).

 hive_metastore >  deepak_sekar_ap_juice_db



Filter tables & columns...

-  apj_sale_items_fact
-  apj_sales_fact
-  bronze_customers
-  bronze_products
-  bronze_sales
-  bronze_store_locations
-  dim_customers
-  dim_products
-  dim_store_locations
-  silver_customers
-  silver_products
-  silver_sale_items
-  silver_sales
-  silver_store_locations
-  v_silver_sale_items
-  v_silver_sales

Step 2: Build your Orders Master queries

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 Orders Master**. Later, you can then use the search functionality to identify your queries.

We can start to do some analysis on orders by country and city.

- Name this query 'Orders Master'. *Don't forget to prepend that name with the identifier you chose using the pattern in the note above.*
- Write a query to join apj_sales_fact table with dim_store_locations 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 **'deepak_sekar_ap_juice_db'** with the relevant database name derived as part of Step 1).
- Execute the query.

```
select
  a.store_id,
  a.order_source,
  a.order_state,
  b.city,
  b.country_code,
  b.name as store_name,
  count(*) as cnt
from
  deepak_sekar_ap_juice_db.apj_sales_fact a
  join deepak_sekar_ap_juice_db.dim_store_locations b on
a.slocation_skey = b.slocation_skey
group by
  a.store_id,
  a.order_source,
  a.order_state,
  b.city,
  b.country_code,
  b.name
```


Your table should look like this:

store_id	order_source	order_state	city	country_code	store_name	cnt
MEL01	ONLINE	COMPLETED	Melbourne	AUS	Melbourne CBD	4509
AKL02	ONLINE	CANCELED	Auckland	NZL	Auckland CBD	186
SYD01	IN-STORE	CANCELED	Sydney	AUS	Sydney CBD	13838
AKL02	IN-STORE	CANCELED	Auckland	NZL	Auckland CBD	4353
WLG01	ONLINE	CANCELED	Wellington	NZL	Wellington CBD	337
SYD01	ONLINE	PENDING	Sydney	AUS	Sydney CBD	1584
AKL02	IN-STORE	PENDING	Auckland	NZL	Auckland CBD	4561
PER01	IN-STORE	COMPLETED	Perth	AUS	Perth CBD	60275
AKL01	ONLINE	COMPLETED	Auckland	NZL	Auckland Airport	1330
WLG01	ONLINE	PENDING	Wellington	NZL	Wellington CBD	467
AKL01	ONLINE	CANCELED	Auckland	NZL	Auckland Airport	239

1

2

3

>

Step 3: Add 7 Visualizations

- Click the 'Add Visualization' button for every time you want to create a new visualization and then create the following 7 visualizations.
- Make sure to go to the Data Labels tab in the visualization panel and turn on "Show Data Labels" for every visualization you create.
- Save every visualization with an appropriate name as indicated in the screenshots below.

Orders by Country

Visualization Type



Pie



General

Series

Colors

Data Labels

X Column

country_code



Y Columns

cnt x

Group By

Choose column...



Errors Column

Choose column...



Direction

Counterclockwise



Legend Placement

Automatic (Flexible)



Legend Items Order

Normal



Missing and NULL Values

Convert to 0 and display in chart



Order State by Country

Visualization Type

 Bar ▼

General X Axis Y Axis Series Colors Data Labels

☐ Horizontal Chart

X Column

country_code ▼

Y Columns

cnt ×

Group By

order_state ▼

Errors Column

Choose column... ▼

Legend Placement

Automatic (Flexible) ▼

Legend Items Order

Normal ▼

Stacking

▼

☒ Normalize values to percentage

Missing and NULL Values

Convert to 0 and display in chart ▼

Order Source by Country

Visualization Type

 Bar ▼

General Y Axis X Axis Series Colors Data Labels

☒ Horizontal Chart

Y Column

country_code ▼

X Columns

cnt ×

Group By

order_source ▼

Errors Column

Choose column... ▼

Legend Placement

Automatic (Flexible) ▼

Legend Items Order

Normal ▼

Stacking

▼

☒ Normalize values to percentage

Missing and NULL Values

Convert to 0 and display in chart ▼

Orders by City

Visualization Type

 Bar ▼

General X Axis Y Axis Series Colors Data Labels

☐ Horizontal Chart

X Column

city ▼

Y Columns

cnt ×

Group By

Choose column... ▼

Errors Column

Choose column... ▼

Legend Placement

Automatic (Hidden) ▼

Stacking

Disabled ▼

☐ Normalize values to percentage

Missing and NULL Values

Convert to 0 and display in chart ▼

Order State by City

Visualization Type

 Pie

▼

General Series Colors Data Labels

X Column

order_state

▼

Y Columns

cnt ×

Group By

city

▼

Errors Column

Choose column...

▼

Direction

Counterclockwise

▼

Legend Placement

Automatic (Flexible)

▼

Legend Items Order

Normal

▼

Missing and NULL Values

Convert to 0 and display in chart

▼

Order Source by City

Visualization Type

 Line

▼

General X Axis Y Axis Series Colors Data Labels

☐ Horizontal Chart

X Column

city

▼

Y Columns

cnt ×

Group By

order_source

▼

Errors Column

Choose column...

▼

Legend Placement

Automatic (Flexible)

▼

Legend Items Order

Normal

▼

☐ Normalize values to percentage

Missing and NULL Values

Convert to 0 and display in chart

▼

Order by Store Name

Visualization Type

 Line

▼

General X Axis Y Axis Series Colors Data Labels

☐ Horizontal Chart

X Column

store_name

▼

Y Columns

cnt ×

Group By

Choose column...

▼

Errors Column

Choose column...

▼

Legend Placement

Automatic (Flexible)

▼

Legend Items Order

Normal

▼

☐ Normalize values to percentage

Missing and NULL Values

Convert to 0 and display in chart

▼

Step 4: Build your Products Master queries

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 Products Master**. Later, you can then use the search functionality to identify your queries.

We can start to do some analysis on orders by country and city.

- Name this query 'Products Master'. *Don't forget to prepend that name with the identifier you chose using the pattern in the note above.*
- Write a query to join apj_sales_fact table with dim_store_locations 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 **'deepak_sekar_ap_juice_db'** with the relevant database name derived as part of Step 1).
- Execute the query.

```
select
  b.city,
  b.country_code,
  b.name as store_name,
  a.product_size,
  c.name as product_name,
  count(*) as cnt,
  sum(product_cost) as total_product_cost
from
  deepak_sekar_ap_juice_db.apj_sale_items_fact a
  join deepak_sekar_ap_juice_db.dim_store_locations b on
a.slocation_skey = b.slocation_skey
  join deepak_sekar_ap_juice_db.dim_products c on a.product_skey =
c.product_skey
where
  a.unique_customer_id is not null
  and b.current_record = 'Y'
  and c.current_record = 'Y'
group by
  b.city,
  b.country_code,
```

```
b.name,
a.product_size,
c.name --c.city, c.country_code, c.name as store_name, d.name as
product_name, sum(a.product_cost) as total_product_cost
```

Your table should look like this:

city	country_code	store_name	product_size	product_name	cnt	total_product_cost
Wellington	NZL	Wellington CBD	Large	Justified	1428	12852.00
Wellington	NZL	Wellington CBD	Medium	Fit Drink	1401	9807.00
Sydney	AUS	Sydney CBD	Small	Fruit Warehouse	3128	15640.00
Auckland	NZL	Auckland Airport	Small	Drink Your Vitamins	850	4250.00
Wellington	NZL	Wellington CBD	Medium	Packed Punch	1407	9849.00
Auckland	NZL	Auckland CBD	Large	Fit Fuel	1177	10593.00
Melbourne	AUS	Melbourne Airport	Small	Jumpstart	304	1520.00
Canberra	AUS	Canberra Airport	Small	Craze	548	2740.00
Perth	AUS	Perth CBD	Large	Nothing To Lose	1501	13509.00
Perth	AUS	Perth CBD	Small	Jungle	1613	8065.00
Canberra	AUS	Canberra Airport	Large	Get Clean	580	5220.00
Sydney	AUS	Sydney CBD	Small	Power Punch	3074	15370.00
Sydney	AUS	Sydney CBD	Large	Fit Fuel	3100	27900.00

1
2
3
4
5
...
40
>

Step 5: Add 6 Visualizations

- Click the 'Add Visualization' button for every time you want to create new visualization and then create the following 6 visualizations.
- Make sure to go to the Data Labels tab in the visualization panel and turn on "Show Data Labels" for every visualization you create.
- Save every visualization with an appropriate name as indicated in the screenshots below.

Product Size by Country

Visualization Type

 Bar

▼

General X Axis Y Axis Series Colors Data Labels

☐ Horizontal Chart

X Column

country_code

▼

Y Columns

cnt x

Group By

product_size

▼

Errors Column

Choose column...

▼

Legend Placement

Automatic (Flexible)

▼

Legend Items Order

Normal

▼

Stacking

▼

☐ Normalize values to percentage

Missing and NULL Values

Convert to 0 and display in chart

▼

cnt

Product Size by City

Visualization Type

 Bar

▼

General X Axis Y Axis Series Colors Data Labels

☐ Horizontal Chart

X Column

city

▼

Y Columns

cnt x

Group By

product_size

▼

Errors Column

Choose column...

▼

Legend Placement

Automatic (Flexible)

▼

Legend Items Order

Normal

▼

Stacking

▼

☐ Normalize values to percentage

Missing and NULL Values

Convert to 0 and display in chart

▼

cnt

Product Size by Cost by Country

Visualization Type

 Bar

▼

General X Axis Y Axis Series Colors Data Labels

☐ Horizontal Chart

X Column

country_code

▼

Y Columns

total_product_cost ×

Group By

product_size

▼

Errors Column

Choose column...

▼

Legend Placement

Automatic (Flexible)

▼

Legend Items Order

Normal

▼

Stacking

Normal

▼

☐ Normalize values to percentage

Missing and NULL Values

Convert to 0 and display in chart

▼

Product Size by Cost by City

Visualization Type

 Bar

▼

General X Axis Y Axis Series Colors Data Labels

☐ Horizontal Chart

X Column

city

▼

Y Columns

total_product_cost ×

Group By

product_size

▼

Errors Column

Choose column...

▼

Legend Placement

Automatic (Flexible)

▼

Legend Items Order

Normal

▼

Stacking

▼

☐ Normalize values to percentage


Missing and NULL Values

Convert to 0 and display in chart

▼

Product name by Country

Visualization Type

 Area

▼

General X Axis Y Axis Series Colors Data Labels

X Column

product_name

▼

Y Columns

cnt ×

Group By

country_code

▼

Errors Column

Choose column...

▼

Legend Placement

Automatic (Flexible)

▼

Legend Items Order

Normal

▼

Stacking

▼

☐ Normalize values to percentage


Missing and NULL Values

Convert to 0 and display in chart

▼

Product name by City

Visualization Type

 Line

▼

General X Axis Y Axis Series Colors Data Labels

☐ Horizontal Chart

X Column

product_name

▼

Y Columns

cnt ×

Group By

city

▼

Errors Column

Choose column...

▼

Legend Placement

Automatic (Flexible)

▼

Legend Items Order

Normal

▼

☐ Normalize values to percentage

Missing and NULL Values

Convert to 0 and display in chart

▼

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 Create Dashboard' button.
- Name your new dashboard " <Your Unique Identifier> APJuice Lakehouse' (ensuring that you prepend the title with your unique identifier).

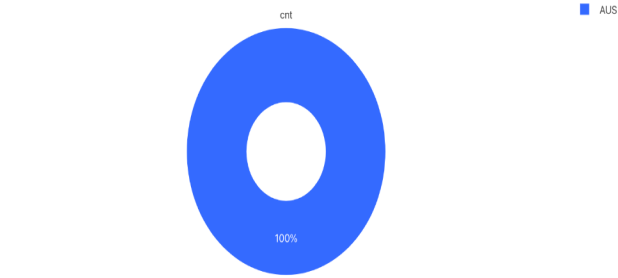
Step 2: Add Visualization

- Click on the 'Add Visualization' button.
- Type the identifier you selected into the search bar.
- Select all the visualizations you created (13 in total) and add to the dashboard one after another

Step 3: Arrange and Resize

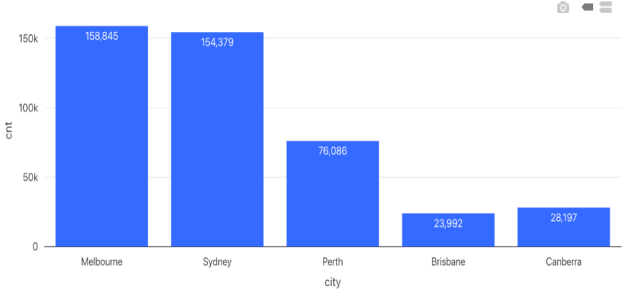
Once you've added the widgets above, you can use drag-and-drop functionality and resizing tools to arrange them. Drag, drop and resize the 13 visualizations so that they are placed next to one other in a 2 column style. Ideally, country-based visualizations on the left side and city-based visualizations on the right side. Please refer to the below screenshot as an example

Orders by Country - Orders Master



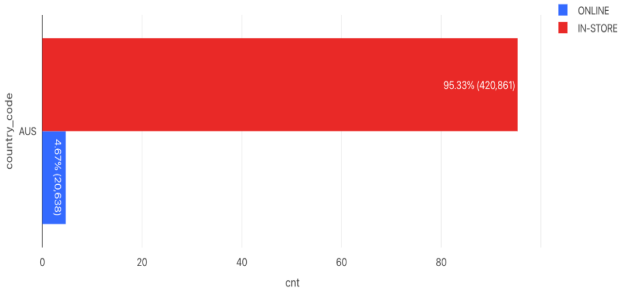
7 hours ago

Orders by City - Orders Master



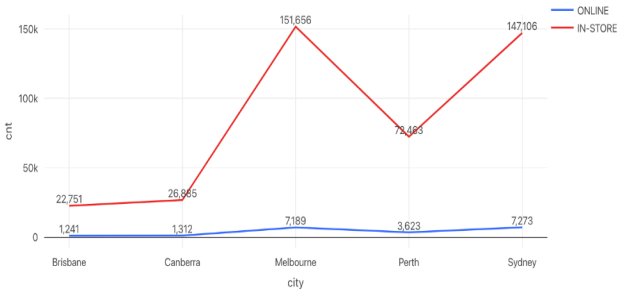
7 hours ago

Order Source by Country - Orders Master



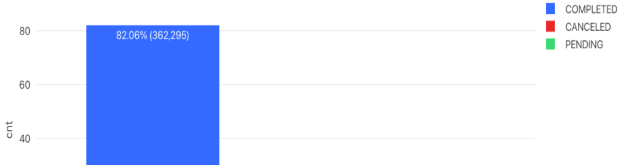
7 hours ago

Order Source by City - Orders Master

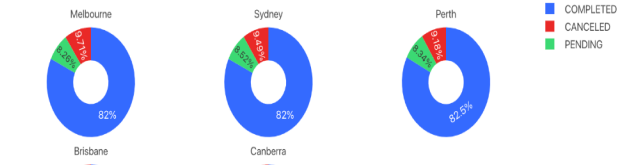


7 hours ago

Order State by Country - Orders Master



Order State by City - Orders Master



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 ANZ and NZL APJuice retail outlets. So far, we have created queries and visualizations that capture metrics that include all of the data across all of the stores. Now, we'll amend our existing queries and visualizations to allow the viewer to drill down based on country code.

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 to Orders Master Query

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 '<Your unique identifier> Orders Master'.
- Tap the 'Add New Parameter' button('{{ }}').
- Enter 'country_code' in the 'Keyword' field.
- Update the 'Title' field to read 'Country_Code'.
- Amend the 'Type' field to read 'Dropdown List'.
- In the 'Values' field, enter 'All' and press the 'Enter' key.
- Type 'AUS' in the subsequent line, followed by 'NZL' in the third line
- Tick the 'Allow multiple values' checkbox.
- Update the 'Quotation' field to read 'Single Quotation Marks':

Country_Code

X

* Title

Country_Code

Type

Dropdown List

▼

Values

All
AUS
NZL

Dropdown list values (newline delimited)

☒ Allow multiple values

Quotation

Single Quotation Mark

▼

Placed in query as: 'value1','value2','value3'

Cancel

OK

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

```
select
  a.store_id,
  a.order_source,
  a.order_state,
  b.city,
  b.country_code,
  b.name as store_name,
  count(*) as cnt
from
  deepak_sekar_ap_juice_db.apj_sales_fact a
  join deepak_sekar_ap_juice_db.dim_store_locations b on
a.slocation_skey = b.slocation_skey
WHERE
  (
    'All' IN ({{ Country_Code }})
    OR country_code IN ({{ Country_Code }})
  )
group by
  a.store_id,
  a.order_source,
  a.order_state,
  b.city,
  b.country_code,
  b.name
```

Step 2: Update Products Master Query to Include the Parameter

- 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 '<Your unique identifier> Products Master'.
- Tap the 'Add New Parameter' button('{{ }}').
- Enter 'country_code' in the 'Keyword' field.
- Update the 'Title' field to read 'Country_Code'.

- Amend the 'Type' field to read 'Dropdown List'.
- In the 'Values' field, enter 'All' and press the 'Enter' key.
- Type 'AUS' in the subsequent line, followed by 'NZL' in the third line
- Tick the 'Allow multiple values' checkbox.
- Update the 'Quotation' field to read 'Single Quotation Marks':

Country_Code ✕

* Title

Country_Code

Type

Dropdown List ▼

Values

All
AUS
NZL

Dropdown list values (newline delimited)

☒ Allow multiple values

Quotation

Single Quotation Mark ▼

Placed in query as: 'value1', 'value2', 'value3'

Cancel

OK

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

```
select
  b.city,
```

```

    b.country_code,
    b.name as store_name,
    a.product_size,
    c.name as product_name,
    count(*) as cnt,
    sum(product_cost) as total_product_cost
from
    deepak_sekar_ap_juice_db.apj_sale_items_fact a
    join deepak_sekar_ap_juice_db.dim_store_locations b on
a.slocation_skey = b.slocation_skey
    join deepak_sekar_ap_juice_db.dim_products c on a.product_skey =
c.product_skey
WHERE
    (
        'All' IN ({{ Country_Code }})
        OR country_code IN ({{ Country_Code }})
    )and
    a.unique_customer_id is not null
    and b.current_record = 'Y'
    and c.current_record = 'Y'
group by
    b.city,
    b.country_code,
    b.name,
    a.product_size,
    c.name --c.city, c.country_code, c.name as store_name, d.name as
product_name, sum(a.product_cost) as total_product_cost

```

Step 3: Create Dashboard Filters

- Click on the Dashboards tab, and type the identifier you selected into the search bar using the 'Search' text box on the right-hand side of the user interface.
- Click on the '<Your Unique Identifier> APJuice Lakehouse' 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.

- Click the ellipsis button that appears in the upper right corner of any of the widgets and click the 'Edit' from the menu option.



Edit Visualization
×

Title

Orders by Country - Orders Master

Description

Parameters

Title	Keyword	Default Value	Value Source
Country_Code 	{{ Country_Code ...	AUS	Widget parameter 

☐ Dynamically resize panel height

Cancel

OK

- 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 'Country_Code' filter represented on the dashboard itself (and no longer visible within the dashboard widget).
- Repeat this for all the other widgets except this time select '**Existing dashboard parameter**' option, ensure that 'Country_Code' is populated in the 'Key' field and click the 'OK' button.
- Once done, tap the 'Done Editing' button.
- Using the 'Country_Code' dashboard filter, experiment with contextualising the visualisations based on the appropriate country that has been selected from the drop-down menu.