Data Science & Engi... ▾

⊕ New

▮ Workspace

Repos

🕐 Recents

⛁ Data

Compute

Workflows

🏪 Marketplace NEW

Cmd 1

```sql
USE f1_processed;
```

OK

Command took 0.41 seconds -- by avinashchinta@my.unt.edu at 7/13/2023, 8:00:16 PM on dbkcourse_cluster

Cmd 2

```sql
CREATE TABLE f1_presentation.calculated_race_results_sql
USING parquet
AS
SELECT races.race_year,
        constructors.name AS team_name,
       drivers.name AS driver_name,
       results.position,
       results.points,
       11 - results.position As calculated_points
FROM results
JOIN f1_processed.drivers ON (results.driver_id = drivers.driver_id)
JOIN f1_processed.constructors ON (results.constructor_id = constructors.constructor_id)
JOIN f1_processed.races ON (results.race_id = races.race_id)
WHERE results.position <= 10
```

▶ (8) Spark Jobs

OK

Command took 21.96 seconds -- by avinashchinta@my.unt.edu at 7/13/2023, 8:08:18 PM on dbkcourse_cluster

▶ Run all   ● dbkcourse_cluster ⌄   🗓 Schedule   Share

Data Science & Engi... ▼

⊕ New

▊ Workspace

⬚ Repos

🕐 Recents

⬚ Data

⬚ Compute

⬚ Workflows

▸ (8) Spark Jobs

OK

Command took 21.96 seconds -- by avinashchinta@my.unt.edu at 7/13/2023, 8:08:18 PM on dbkcourse_cluster

Cmd 3

SQL ▶⌄ ▥ ⌄ ━ ✕

```sql
SELECT * FROM f1_presentation.calculated_race_results_sql;
```

▸ (2) Spark Jobs

Table ⌄   +

	race_year	team_name	driver_name	position	points	calculated_points
1	1987	Ligier	René Arnoux	10	0	1
2	1987	Brabham	Riccardo Patrese	9	0	2
3	1987	Zakspeed	Christian Danner	8	0	3
4	1987	McLaren	Stefan Johansson	7	0	4
5	1987	Arrows	Eddie Cheever	6	1	5
6	1987	Williams	Nigel Mansell	5	2	6
7	1987	Ferrari	Gerhard Berger	4	3	7

⬇ ⌄   10,000 rows | Truncated data | 0.56 seconds runtime   Refreshed 5 minutes ago

Command took 0.56 seconds -- by avinashchinta@my.unt.edu at 7/13/2023, 8:08:59 PM on dbkcourse_cluster

Shift+Enter to run
Shift+Ctrl+Enter to run selected text

🛒 Marketplace NEW

⬚ Partner Connect

1/4 Tasks Completed

New

Workspace

Repos

Recents

Data

Compute

Workflows

Cmd 1

```sql
SELECT driver_name,
       COUNT(1) AS total_races,
       SUM(calculated_points) AS total_points,
       AVG(calculated_points) AS avg_points
  FROM f1_presentation.calculated_race_results_sql
GROUP BY driver_name
HAVING COUNT(1) >= 50
ORDER BY avg_points DESC
```

SQL ▶▾ ◫ ∨ — ✕

▸ (2) Spark Jobs

Table ∨   +

	driver_name	total_races	total_points	avg_points
1	Alain Prost	138	1156	8.376811594202898
2	Ayrton Senna	104	869	8.35576923076923
3	Jackie Stewart	61	507	8.311475409836065
4	Lewis Hamilton	232	1903	8.202586206896552
5	Michael Schumacher	226	1796	7.946902654867257
6	Nigel Mansell	90	713	7.9222222222222225
7	Niki Lauda	81	639	7.888888888888889

↓ 64 rows | 0.97 seconds runtime   Refreshed 2 hours ago

Marketplace   NEW

Partner Connect

1.find_dominant_drivers SQL ⌄

File Edit View Run Help Last edit was 2 hours ago Provide feedback

▶ Run all ● dbkcourse_cluster ⌄ 📅 Schedule Share ⌃

New

Workspace

Repos

Recents

Data

Compute

Workflows

Cmd 2

```sql
SELECT driver_name,
       COUNT(1) AS total_races,
       SUM(calculated_points) AS total_points,
       AVG(calculated_points) AS avg_points
  FROM f1_presentation.calculated_race_results_sql
 WHERE race_year BETWEEN 2011 AND 2020
GROUP BY driver_name
HAVING COUNT(1) >= 50
ORDER BY avg_points DESC
```

▶ (2) Spark Jobs

Table ⌄ +

	driver_name	total_races	total_points	avg_points
1	Lewis Hamilton	177	1478	8.350282485875706
2	Sebastian Vettel	164	1282	7.817073170731708
3	Nico Rosberg	93	673	7.236559139784946
4	Max Verstappen	88	605	6.875
5	Valtteri Bottas	117	793	6.777777777777778
6	Fernando Alonso	95	584	6.147368421052631
7	Kimi Räikkönen	121	721	5.958677685950414

⤓ 14 rows | 0.51 seconds runtime   Refreshed 2 hours ago

Command took 0.51 seconds -- by avinashchinta@my.unt.edu at 7/13/2023, 8:43:18 PM on dbkcourse_cluster

Cmd 3

Marketplace NEW

Partner Connect

▶ Run all   ● dbkcourse_cluster ∨   📅 Schedule   **Share**

Cmd 1

```sql
SELECT team_name,
        COUNT(1) AS total_races,
        SUM(calculated_points) AS total_points,
        AVG(calculated_points) AS avg_points
   FROM f1_presentation.calculated_race_results_sql
GROUP BY team_name
HAVING COUNT(1) >= 100
ORDER BY avg_points DESC
```

▶ (2) Spark Jobs

Table ∨   +

	team_name	total_races	total_points	avg_points
1	Mercedes	385	2931	7.612987012987013
2	Ferrari	1478	10500	7.104194857916103
3	McLaren	1032	6865	6.6521317829457365
4	Red Bull	433	2851	6.584295612009238
5	Williams	745	4759	6.387919463087249
6	Benetton	296	1806	6.101351351351352
7	Team Lotus	352	2100	5.965909090909091

⬇ 23 rows | 7.58 seconds runtime   Refreshed 1 minute ago

Command took 7.58 seconds -- by avinashchinta@my.unt.edu at 7/13/2023, 10:42:02 PM on dbkcourse_cluster

Cmd 2

▶ Run all   ● dbkcourse_cluster ∨   📅 Schedule   Share

Cmd 2

```sql
SELECT team_name,
       COUNT(1) AS total_races,
       SUM(calculated_points) AS total_points,
       AVG(calculated_points) AS avg_points
  FROM f1_presentation.calculated_race_results_sql
 WHERE race_year BETWEEN 2011 AND 2020
GROUP BY team_name
HAVING COUNT(1) >= 100
ORDER BY avg_points DESC
```

▶ (2) Spark Jobs

Table ∨   +

	team_name	total_races	total_points	avg_points
1	Mercedes	331	2580	7.794561933534744
2	Red Bull	313	2201	7.031948881789138
3	Ferrari	314	2095	6.671974522292993
4	McLaren	204	984	4.823529411764706
5	Williams	125	531	4.248
6	Force India	189	671	3.5502645502645502
7	Toro Rosso	119	316	2.6554621848739495

⬇ 7 rows | 1.11 seconds runtime   Refreshed 2 minutes ago

Command took 1.11 seconds -- by avinashchinta@my.unt.edu at 7/13/2023, 10:42:02 PM on dbkcourse_cluster

Cmd 3

New
Workspace
Repos
Recents
Data
Compute
Workflows

```python
%python
html = """<h1 style="color:Black;text-align:center;font-family:Ariel">Report on Dominant Formula 1 Drivers </h1>"""
displayHTML(html)
```

Report on Dominant Formula 1 Drivers

Command took 0.17 seconds -- by avinashchinta@my.unt.edu at 7/13/2023, 10:53:32 PM on dbkcourse_cluster

Cmd 2

```sql
CREATE OR REPLACE TEMP VIEW v_dominant_drivers
AS
SELECT driver_name,
       COUNT(1) AS total_races,
       SUM(calculated_points) AS total_points,
       AVG(calculated_points) AS avg_points,
       RANK() OVER(ORDER BY AVG(calculated_points) DESC) driver_rank
  FROM f1_presentation.calculated_race_results_sql
GROUP BY driver_name
HAVING COUNT(1) >= 50
ORDER BY avg_points DESC
```

OK

Command took 0.55 seconds -- by avinashchinta@my.unt.edu at 7/13/2023, 10:53:32 PM on dbkcourse_cluster

Cmd 3

SQL ▶▾ ⬚ ∨ − ✕

```sql
SELECT * FROM v_dominant_drivers;
```

▶ (3) Spark Jobs

Table ▾ +

	driver_name	total_races	total_points	avg_points	driver_rank
1	Alain Prost	138	1156	8.376811594202898	1
2	Ayrton Senna	104	869	8.35576923076923	2
3	Jackie Stewart	61	507	8.311475409836065	3
4	Lewis Hamilton	232	1903	8.202586206896552	4
5	Michael Schumacher	226	1796	7.946902654867257	5
6	Nigel Mansell	90	713	7.9222222222222225	6
7	Niki Lauda	81	639	7.888888888888889	7

Marketplace
Partner Connect
1/4 Tasks Completed
Enable new UI
Menu options

3.viz_dominant_drivers SQL ▾

▶ Run all   ● dbkcourse_cluster ▾   🗓 Schedule   Share

- New
- **Workspace**
- Repos
- Recents
- Data
- Compute
- Workflows

```sql
SELECT race_year,
       driver_name,
       COUNT(1) AS total_races,
       SUM(calculated_points) AS total_points,
       AVG(calculated_points) AS avg_points
  FROM f1_presentation.calculated_race_results_sql
 WHERE driver_name IN (SELECT driver_name FROM v_dominant_drivers WHERE driver_rank <= 10)
GROUP BY race_year, driver_name
ORDER BY race_year, avg_points DESC
```

▸ (5) Spark Jobs

Visualization ▾   +

▸ (5) Spark Jobs

driver_name

- ━ Jackie Stewart
- ━ Niki Lauda
- ━ Alain Prost
- ━ Nigel Mansell
- ━ Ayrton Senna
- ━ Michael Schumacher
- ━ Damon Hill
- ━ Juan Pablo Montoya
- ━ Lewis Hamilton
- ━ Sebastian Vettel

- Marketplace   NEW
- Partner Connect
- 1/4 Tasks Completed
- Enable new UI   NEW
- Menu options

File Edit View Run Help   Last edit was 3 minutes ago   Provide feedback

▶ Run all   ● dbkcourse_cluster ∨   📅 Schedule   **Share**

```sql
SELECT race_year,
        driver_name,
        COUNT(1) AS total_races,
        SUM(calculated_points) AS total_points,
        AVG(calculated_points) AS avg_points
  FROM f1_presentation.calculated_race_results_sql
 WHERE driver_name IN (SELECT driver_name FROM v_dominant_drivers WHERE driver_rank <= 10)
GROUP BY race_year, driver_name
ORDER BY race_year, avg_points DESC
```

▶ (5) Spark Jobs

Visualization ∨   +

▶ (5) Spark Jobs

New

Workspace

Repos

Recents

Data

Compute

Workflows

▶ Run all   ● dbkcourse_cluster ▾   📅 Schedule   Share

Cmd 6

```sql
SELECT race_year,
       driver_name,
       COUNT(1) AS total_races,
       SUM(calculated_points) AS total_points,
       AVG(calculated_points) AS avg_points
  FROM f1_presentation.calculated_race_results_sql
 WHERE driver_name IN (SELECT driver_name FROM v_dominant_drivers WHERE driver_rank <= 10)
 GROUP BY race_year, driver_name
 ORDER BY race_year, avg_points DESC
```

▸ (5) Spark Jobs

Visualization ▾   +

▸ (5) Spark Jobs

driver_name
■ Jackie Stewart   ■ Niki Lauda
■ Alain Prost   ■ Nigel Mansell
■ Ayrton Senna   ■ Michael Schum...
■ Damon Hill   ■ Juan Pablo Mo...
■ Lewis Hamilton   ■ Sebastian Vettel

⬇ 118 rows | 1.65 seconds runtime   Refreshed 7 minutes ago

Command took 1.65 seconds -- by avinashchinta@my.unt.edu at 7/13/2023, 10:53:32 PM on dbkcourse_cluster

Cmd 7

Marketplace

Partner Connect

1/4 Tasks Completed

Enable new UI

```python
%python
html = """<h1 style="color:Black;text-align:center;font-family:Ariel">Report on Dominant Formula 1 Teams </h1>"""
displayHTML(html)
```

Report on Dominant Formula 1 Teams

Command took 0.07 seconds -- by avinashchinta@my.unt.edu at 7/13/2023, 11:03:18 PM on dbkcourse_cluster

Cmd 2

```sql
CREATE OR REPLACE TEMP VIEW v_dominant_teams
AS
SELECT team_name,
       COUNT(1) AS total_races,
       SUM(calculated_points) AS total_points,
       AVG(calculated_points) AS avg_points,
       RANK() OVER(ORDER BY AVG(calculated_points) DESC) team_rank
  FROM f1_presentation.calculated_race_results_sql
GROUP BY team_name
HAVING COUNT(1) >= 100
ORDER BY avg_points DESC
```

OK

Command took 0.38 seconds -- by avinashchinta@my.unt.edu at 7/13/2023, 11:03:18 PM on dbkcourse_cluster

Cmd 3

```sql
SELECT * FROM v_dominant_teams;
```

▶ (3) Spark Jobs

Table ⌄   +

	team_name	total_races	total_points	avg_points	team_rank
1	Mercedes	385	2931	7.612987012987013	1
2	Ferrari	1478	10500	7.104194857916103	2
3	McLaren	1032	6865	6.6521317829457365	3
4	Red Bull	433	2851	6.584295612009238	4

4.viz_dominant_teams SQL ∨

File Edit View Run Help   Last edit was 4 minutes ago   Provide feedback

▶ Run all   ● dbkcourse_cluster ∨   🗓 Schedule   **Share**

New

Workspace

Repos

Recents

Data

Compute

Workflows

Cmd 4

```sql
SELECT race_year,
       team_name,
       COUNT(1) AS total_races,
       SUM(calculated_points) AS total_points,
       AVG(calculated_points) AS avg_points
  FROM f1_presentation.calculated_race_results_sql
 WHERE team_name IN (SELECT team_name FROM v_dominant_teams WHERE team_rank <= 5)
 GROUP BY race_year, team_name
 ORDER BY race_year, avg_points DESC
```

▶ (5) Spark Jobs

Visualization ∨   +

▶ (5) Spark Jobs

Marketplace   NEW

Partner Connect

1/4 Tasks Completed

Enable new UI   NEW

Menu options

⊕ New

▦ Workspace

▣ Repos

🕐 Recents

⊟ Data

🏠 Compute

☰ Workflows

Command took 1.37 seconds -- by avinashchinta@my.unt.edu at 7/13/2023, 11:03:18 PM on dbkcourse_cluster

Cmd 5

SQL ▶▾ 📊 ▾ ─ ✕

```sql
SELECT race_year,
       team_name,
       COUNT(1) AS total_races,
       SUM(calculated_points) AS total_points,
       AVG(calculated_points) AS avg_points
  FROM f1_presentation.calculated_race_results_sql
 WHERE team_name IN (SELECT team_name FROM v_dominant_teams WHERE team_rank <= 5)
 GROUP BY race_year, team_name
 ORDER BY race_year, avg_points DESC
```

▸ (5) Spark Jobs

Visualization ▾   +

▸ (5) Spark Jobs

⤓ 194 rows | 1.10 seconds runtime   Refreshed 4 minutes ago

Command took 1.10 seconds -- by avinashchinta@my.unt.edu at 7/13/2023, 11:03:18 PM on dbkcourse_cluster

🏪 Marketplace   NEW

🔗 Partner Connect

1/4 Tasks Completed

▶ Run all   ● dbkcourse_cluster ˅   📅 Schedule   Share   ˄

⊕ New

▦ Workspace

🗃 Repos

🕐 Recents

⛁ Data

🖧 Compute

☰ Workflows

🏪 Marketplace   NEW

🕹 Partner Connect

1/4 Tasks Completed

👁 Enable new UI   NEW

Report on Dominant Formula 1 Drivers

F1 Dominant Drivers✎

View of notebook:
3.viz_dominant_drivers

▶ Present Dashboard

Layout option:

| Stack | Float |

Dashboard width:

1024px ˅

🖊 Remove all graphs

🗑 Delete this dashboard

F1 Dominant Drivers

Report on Dominant Formula 1 Drivers

4.viz_dominant_teams SQL ⌄
File Edit View Run Help   Provide feedback

▶ Run all   ● dbkcourse_cluster ⌄   📅 Schedule   **Share**   ⌃

⊕ New

▦ **Workspace**

⌷ Repos

🕐 Recents

△ Data

🖧 Compute

≔ Workflows

🏪 Marketplace NEW

⚏ Partner Connect

1/4 Tasks Completed

Report on Dominant Formula 1 Teams

F1 Dominant Teams ✎

View of notebook:
4.viz_dominant_teams

▶ Present Dashboard

Layout option:
Stack   Float

Dashboard width:
1024px ⌄

✐ Remove all graphs

🗑 Delete this dashboard

← → C 🔒 adb-4375995990155053.13.azuredatabricks.net/?o=4375995990155053#notebook/2064557753314925/dashboard/2064557753314932/present   ⊖ ⮐ ☆ ▢ 🌑 :

Microsoft Azure   databricks   Q Search data, notebooks, recents, and more...   CTRL + P   dbkcourse_ws 🎁 ⑦ avinashchinta@my.unt.edu ⌄

F1 Dominant Teams

Report on Dominant Formula 1 Teams

▶ Run all   ● dbkcourse_cluster ∨   🗓 Schedule   **Share**

Cmd 1

Drop all the tables

Cmd 2

```sql
DROP DATABASE IF EXISTS f1_processed CASCADE;
```

OK

Command took 0.28 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 12:37:59 PM on dbkcourse_cluster

Cmd 3

```sql
CREATE DATABASE IF NOT EXISTS f1_processed
LOCATION "abfss://processed@erformula1dl.dfs.core.windows.net/";
```

OK

Command took 0.47 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 12:37:59 PM on dbkcourse_cluster

Cmd 4

```sql
DROP DATABASE IF EXISTS f1_presentation CASCADE;
```

OK

Command took 1.50 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 12:37:59 PM on dbkcourse_cluster

Cmd 5

```sql
CREATE DATABASE IF NOT EXISTS f1_presentation
LOCATION "abfss://presentation@erformula1dl.dfs.core.windows.net/";
```

OK

Command took 0.22 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 12:37:59 PM on dbkcourse_cluster

Cmd 6

presentation ⋯
Container

Search

- ▣ Overview
- ✐ Diagnose and solve problems
- ৪ Access Control (IAM)

Settings

- ⇆ Shared access tokens
- ৪ Manage ACL
- 🔑 Access policy
- ☰ Properties
- ⓘ Metadata

⏶ Upload   ＋ Add Directory   ↻ Refresh | ↱ Rename   🗑 Delete   ⇄ Change tier   🖉 Acquire lease   🖉 Break lease   🗩 Give feedback

Authentication method: Access key (Switch to Azure AD User Account)
Location: presentation

Search blobs by prefix (case-sensitive)   ⬤ Show deleted objects

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state

No blobs found.

processed ⋯

Container

✕

🔍 Search .... «

⬛ **Overview**

🔗 Diagnose and solve problems

🔑 Access Control (IAM)

Settings

🔗 Shared access tokens

🔑 Manage ACL

🔑 Access policy

📊 Properties

ℹ️ Metadata

⬆ Upload   ➕ Add Directory   ↻ Refresh | ⤴ Rename   🗑 Delete   ⇄ Change tier   🔗 Acquire lease   🔗 Break lease   🔗 Give feedback

Authentication method: Access key (Switch to Azure AD User Account)
Location: processed

| Search blobs by prefix (case-sensitive) | ⬤ Show deleted objects |

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state

No results

raw
Container

Overview

Diagnose and solve problems

Access Control (IAM)

Settings

Shared access tokens

Manage ACL

Access policy

Properties

Metadata

↑ Upload + Add Directory ↻ Refresh | ↰ Rename 🗑 Delete ⇄ Change tier 🔑 Acquire lease 🔑 Break lease 🗨 Give feedback

Delete blobs and directories

If you delete the selected directories, all blobs and directories in those selected directories will also be deleted.

Are you sure you want to delete the selected blobs and directories?

[Yes] [No]

☑ 📁 lap_times				-		···
☑ 📁 qualifying				-		···
☑ 📄 circuits.csv	7/11/2023, 11:15:55 PM	Hot (Inferred)	Block blob	9.81 KiB	Available	···
☑ 📄 constructors.json	7/11/2023, 11:15:55 PM	Hot (Inferred)	Block blob	29.7 KiB	Available	···
☑ 📄 drivers.json	7/11/2023, 11:15:55 PM	Hot (Inferred)	Block blob	176.57 KiB	Available	···
☑ 📄 pit_stops.json	7/11/2023, 11:15:57 PM	Hot (Inferred)	Block blob	1.31 MiB	Available	···
☑ 📄 races.csv	7/11/2023, 11:15:55 PM	Hot (Inferred)	Block blob	114.11 KiB	Available	···
☑ 📄 results.json	7/11/2023, 11:16:01 PM	Hot (Inferred)	Block blob	6.83 MiB	Available	···

raw ···
Container

- Overview
- Diagnose and solve problems
- Access Control (IAM)

Settings

- Shared access tokens
- Manage ACL
- Access policy
- Properties
- Metadata

↑ Upload + Add Directory ↻ Refresh | ↱ Rename 🗑 Delete ⇄ Change tier 🔑 Acquire lease 🔑 Break lease 🔗 Give feedback

Authentication method: Access key (Switch to Azure AD User Account)
Location: raw

| | Search blobs by prefix (case-sensitive) | ⬤ Show deleted objects |

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state	
☐ 📁 2021-03-21						-	···
☐ 📁 2021-03-28						-	···
☐ 📁 2021-04-18						-	···

▶ Run all   • • Starting ⌄   📅 Schedule   Share   ⌃

p_data_source
```
testing
```

p_file_date
```
2021-03-21
```

Cmd 1

Markdown 📊 ⌄ − ✕

Ingest circuits.csv file

Cmd 2

```python
dbutils.widgets.text("p_data_source", "")
v_data_source = dbutils.widgets.get("p_data_source")
```

Cmd 3

```python
dbutils.widgets.text("p_file_date", "2021-03-21")
v_file_date = dbutils.widgets.get("p_file_date")
```

Cmd 4

```python
%run "../includes/configuration"
```

Cmd 5

```python
%run "../includes/common_functions"
```

Cmd 6

Step 1 - Read the CSV file using the spark dataframe reader

Cmd 7

```python
from pyspark.sql.types import StructType, StructField, IntegerType, StringType, DoubleType
```

▶ Run all   •• Starting ∨   🗓 Schedule   **Share**

p_data_source

testing

p_file_date

2021-03-21

Cmd 9

```python
circuits_df = spark.read \
.option("header", True) \
.schema(circuits_schema) \
.csv(f"{raw_folder_path}/{v_file_date}/circuits.csv")
```

Cmd 10

Step 2 - Select only the required columns

Cmd 11

```python
from pyspark.sql.functions import col
```

Cmd 12

```python
circuits_selected_df = circuits_df.select(col("circuitId"), col("circuitRef"), col("name"), col("location"), col("country"), col("lat"), col("lng"), col("alt"))
```

Cmd 13

Step 3 - Rename the columns as required

Cmd 14

```python
from pyspark.sql.functions import lit
```

▶ Run all   •••• Starting ∨   📅 Schedule   **Share**

p_data_source

```
testing
```

p_file_date

```
2021-03-21
```

```python
from pyspark.sql.functions import lit
```

Cmd 15

```python
circuits_renamed_df = circuits_selected_df.withColumnRenamed("circuitId", "circuit_id") \
    .withColumnRenamed("circuitRef", "circuit_ref") \
    .withColumnRenamed("lat", "latitude") \
    .withColumnRenamed("lng", "longitude") \
    .withColumnRenamed("alt", "altitude") \
    .withColumn("data_source", lit(v_data_source)) \
    .withColumn("file_date", lit(v_file_date))
```

Cmd 16

Step 4 - Add ingestion date to the dataframe

Cmd 17

```python
circuits_final_df = add_ingestion_date(circuits_renamed_df)
```

Cmd 18

Step 5 - Write data to datalake as parquet

Cmd 19

```python
circuits_final_df.write.mode("overwrite").format("parquet").saveAsTable("f1_processed.circuits")
```

Cmd 20

```python
display(spark.read.parquet(f"{processed_folder_path}/circuits"))
```

1.ingest_circuits_file Python ▼

File Edit View Run Help   Last edit was 2 days ago   Provide feedback

▶ Run all   ● dbkcourse_cluster ▼   📅 Schedule   Share ^

p_data_source
```
testing
```

p_file_date
```
2021-03-21
```

```python
circuits_final_df.write.mode("overwrite").format("parquet").saveAsTable("f1_processed.circuits")
```

▶ (1) Spark Jobs

Command took 7.00 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 1:35:25 PM on dbkcourse_cluster

Cmd 20

```python
display(spark.read.parquet(f"{processed_folder_path}/circuits"))
```

▶ (2) Spark Jobs

Table ▼ +

	circuit_id	circuit_ref	name	location	country	latitude	longitude	altitude	data_source	file_date	ingestion_date
1	1	albert_park	Albert Park Grand Prix Circuit	Melbourne	Australia	-37.8497	144.968	10	testing	2021-03-21	2023-07-14T18:35:50.119+0000
2	2	sepang	Sepang International Circuit	Kuala Lumpur	Malaysia	2.76083	101.738	18	testing	2021-03-21	2023-07-14T18:35:50.119+0000
3	3	bahrain	Bahrain International Circuit	Sakhir	Bahrain	26.0325	50.5106	7	testing	2021-03-21	2023-07-14T18:35:50.119+0000
4	4	catalunya	Circuit de Barcelona-Catalunya	Montmeló	Spain	41.57	2.26111	109	testing	2021-03-21	2023-07-14T18:35:50.119+0000
5	5	istanbul	Istanbul Park	Istanbul	Turkey	40.9517	29.405	130	testing	2021-03-21	2023-07-14T18:35:50.119+0000
6	6	monaco	Circuit de Monaco	Monte-Carlo	Monaco	43.7347	7.42056	7	testing	2021-03-21	2023-07-14T18:35:50.119+0000
7	7	villeneuve	Circuit Gilles Villeneuve	Montreal	Canada	45.5	-73.5228	13	testing	2021-03-21	2023-07-14T18:35:50.119+0000

⬇ 77 rows | 4.98 seconds runtime   Refreshed now

Command took 4.98 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 1:35:25 PM on dbkcourse_cluster

Cmd 21

```sql
%sql
SELECT * FROM f1_processed.circuits;
```

▶ (1) Spark Jobs

▶ ▤ _sqldf: pyspark.sql.dataframe.DataFrame = [circuit_id: integer, circuit_ref: string ... 9 more fields]

Table ▼ +

	circuit_id	circuit_ref	name	location	country	latitude	longitude	altitude	data_source	file_date	ingestion_date
1	1	albert_park	Albert Park Grand Prix Circuit	Melbourne	Australia	-37.8497	144.968	10	testing	2021-03-21	2023-07-14T18:35:50.119+0000

2.ingest_races_file Python ∨
File Edit View Run Help   Last edit was now   Provide feedback

▶ Run all ● dbkcourse_cluster ∨ 🗓 Schedule **Share** ^

⊕ New
▣ **Workspace**
📁 Repos
🕐 Recents
⧉ Data
⚌ Compute
☰ Workflows

p_data_source   p_file_date
[   ]   [2021-03-21   ]

🖉 ⚙ 📌

Cmd 16

Write the output to processed container in parquet format

Cmd 17

```
races_selected_df.write.mode("overwrite").partitionBy('race_year').format("parquet").saveAsTable("f1_processed.races")
```

▸ (3) Spark Jobs

Command took 20.84 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 1:44:05 PM on dbkcourse_cluster

Cmd 18

⎯⎯⎯ SQL ▶▾ 📊 ∨ ─ ✕

```
%sql
SELECT * FROM f1_processed.races;
```

▸ (4) Spark Jobs

▸ ▦ _sqldf: pyspark.sql.dataframe.DataFrame = [race_id: integer, round: integer ... 6 more fields]

Table ∨   +

	race_id	round	circuit_id	name	ingestion_date	race_timestamp	file_date	race_year
1	1053	2	21	Emilia Romagna Grand Prix	2023-07-14T18:44:10.500+0000	2021-04-18T13:00:00.000+0000	2021-03-21	2021
2	1052	1	3	Bahrain Grand Prix	2023-07-14T18:44:10.500+0000	2021-03-28T15:00:00.000+0000	2021-03-21	2021
3	1051	21	1	Australian Grand Prix	2023-07-14T18:44:10.500+0000	2021-11-21T06:00:00.000+0000	2021-03-21	2021
4	1054	3	20	TBC	2023-07-14T18:44:10.500+0000	null	2021-03-21	2021
5	1055	4	4	Spanish Grand Prix	2023-07-14T18:44:10.500+0000	2021-05-09T13:00:00.000+0000	2021-03-21	2021
6	1056	5	6	Monaco Grand Prix	2023-07-14T18:44:10.500+0000	2021-05-23T13:00:00.000+0000	2021-03-21	2021
7	1057	6	73	Azerbaijan Grand Prix	2023-07-14T18:44:10.500+0000	2021-06-06T12:00:00.000+0000	2021-03-21	2021

⬇ 1,058 rows | 2.87 seconds runtime   Refreshed now

ⓘ SQL cell result stored as PySpark data frame _sqldf . Learn more

Command took 2.87 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 1:44:05 PM on dbkcourse_cluster

Cmd 19

```
dbutils.notebook.exit("Success")
```

🏪 Marketplace `NEW`
🔗 Partner Connect
1/4 Tasks Completed
👁 Enable new UI `NEW`
▣ Menu options

⊕ New

▮ Workspace

🗂 Repos

🕓 Recents

🔺 Data

🖧 Compute

≔ Workflows

p_data_source

p_file_date

2021-03-21

Cmd 16

Step 4 Write output to parquet file

Cmd 17

```python
constructor_final_df.write.mode("overwrite").format("parquet").saveAsTable("f1_processed.constructors")
```

▸ (1) Spark Jobs

Command took 1.50 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 1:41:49 PM on dbkcourse_cluster

Cmd 18

```sql
%sql
SELECT * FROM f1_processed.constructors;
```

▸ (1) Spark Jobs

▸ ▦ _sqldf: pyspark.sql.dataframe.DataFrame = [constructor_id: integer, constructor_ref: string ... 5 more fields]

Table ⌄   +

	constructor_id	constructor_ref	name	nationality	data_source	file_date	ingestion_date
1	1	mclaren	McLaren	British		2021-03-21	2023-07-14T18:41:53.939+0000
2	2	bmw_sauber	BMW Sauber	German		2021-03-21	2023-07-14T18:41:53.939+0000
3	3	williams	Williams	British		2021-03-21	2023-07-14T18:41:53.939+0000
4	4	renault	Renault	French		2021-03-21	2023-07-14T18:41:53.939+0000
5	5	toro_rosso	Toro Rosso	Italian		2021-03-21	2023-07-14T18:41:53.939+0000
6	6	ferrari	Ferrari	Italian		2021-03-21	2023-07-14T18:41:53.939+0000
7	7	toyota	Toyota	Japanese		2021-03-21	2023-07-14T18:41:53.939+0000

⬇ 211 rows | 0.49 seconds runtime   Refreshed now

ℹ SQL cell result stored as PySpark data frame _sqldf . Learn more

Command took 0.49 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 1:41:49 PM on dbkcourse_cluster

Cmd 19

```python
dbutils.notebook.exit("Success")
```

🛍 Marketplace NEW

⚘ Partner Connect

1/4 Tasks Completed

◉ Enable new UI NEW

▯ Menu options

New

Workspace

Repos

Recents

Data

Compute

Workflows

4.ingest_drivers_file Python ∨
File Edit View Run Help Last edit was 2 days ago Provide feedback

▶ Run all ● dbkcourse_cluster ∨ 📅 Schedule **Share** ^

p_data_source

p_file_date

2021-03-21

Cmd 18

```
drivers_final_df.write.mode("overwrite").format("parquet").saveAsTable("f1_processed.drivers")
```

▸ (1) Spark Jobs

Command took 1.60 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 1:42:51 PM on dbkcourse_cluster

Cmd 19

```
%sql
SELECT * FROM f1_processed.drivers
```

▸ (1) Spark Jobs

▸ ▦ _sqldf: pyspark.sql.dataframe.DataFrame = [driver_id: integer, driver_ref: string ... 8 more fields]

Table ∨ +

	driver_id	driver_ref	number	code	name	dob	nationality	ingestion_date	data_source	file_date
1	1	hamilton	44	HAM	Lewis Hamilton	1985-01-07	British	2023-07-14T18:42:56.007+0000		2021-03-21
2	2	heidfeld	null	HEI	Nick Heidfeld	1977-05-10	German	2023-07-14T18:42:56.007+0000		2021-03-21
3	3	rosberg	6	ROS	Nico Rosberg	1985-06-27	German	2023-07-14T18:42:56.007+0000		2021-03-21
4	4	alonso	14	ALO	Fernando Alonso	1981-07-29	Spanish	2023-07-14T18:42:56.007+0000		2021-03-21
5	5	kovalainen	null	KOV	Heikki Kovalainen	1981-10-19	Finnish	2023-07-14T18:42:56.007+0000		2021-03-21
6	6	nakajima	null	NAK	Kazuki Nakajima	1985-01-11	Japanese	2023-07-14T18:42:56.007+0000		2021-03-21
7	7	bourdais	null	BOU	Sébastien Bourdais	1979-02-28	French	2023-07-14T18:42:56.007+0000		2021-03-21

⬇ 853 rows | 0.47 seconds runtime

Refreshed now

ⓘ SQL cell result stored as PySpark data frame _sqldf . Learn more

Command took 0.47 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 1:42:51 PM on dbkcourse_cluster

Cmd 20

```
dbutils.notebook.exit("Success")
```

Notebook exited: Success

Marketplace NEW

Partner Connect

1/4 Tasks Completed

Enable new UI NEW

Menu options

processed ···
Container

×

↑ Upload   + Add Directory   ↻ Refresh | ↶ Rename   🗑 Delete   ⇄ Change tier   🔑 Acquire lease   🔑 Break lease   📋 Give feedback

Authentication method: Access key (Switch to Azure AD User Account)
Location: processed

Search blobs by prefix (case-sensitive)   ⬤─   Show deleted objects

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state	
📁 circuits						-	···
📁 constructors						-	···
📁 drivers						-	···
📁 races						-	···

p_data_source
```
Ergast
```

p_file_date
```
2021-03-28
```

Step 4 - Write to output to processed container in parquet format

Cmd 18

Method 1

Cmd 19

```python
# for race_id_list in results_final_df.select("race_id").distinct().collect():
#   if (spark._jsparkSession.catalog().tableExists("f1_processed.results")):
#     spark.sql(f"ALTER TABLE f1_processed.results DROP IF EXISTS PARTITION (race_id = {race_id_list.race_id})")
```

Cmd 20

```python
# results_final_df.write.mode("append").partitionBy('race_id').format("parquet").saveAsTable("f1_processed.results")
```

Cmd 21

Method 2

Cmd 22

```python
overwrite_partition(results_final_df, 'f1_processed', 'results', 'race_id')
```

Cmd 23

```python
dbutils.notebook.exit("Success")
```

Cmd 24

```sql
%sql
SELECT race_id, COUNT(1)
FROM f1_processed.results
GROUP BY race_id
ORDER BY race_id DESC;
```

common_functions Python ∨

File Edit View Run Help   Last edit was 4 minutes ago   Provide feedback

▶ Run all   ● Connect ∨   📅 Schedule   Share

- ⊕ New
- ▮ Workspace
- Repos
- 🕐 Recents
- 🔗 Data
- 🏠 Compute
- ☱ Workflows

Cmd 1

```python
from pyspark.sql.functions import current_timestamp
def add_ingestion_date(input_df):
    output_df = input_df.withColumn("ingestion_date", current_timestamp())
    return output_df
```

Cmd 2

Python ▶▾ ∨ − ✕

```python
# function for rearranging the partition column to the end of the table
def re_arrange_partition_column(input_df, partition_column):
    column_list = []
    for column_name in input_df.schema.names:
        if column_name != partition_column:
            column_list.append(column_name)
    column_list.append(partition_column)
    output_df = input_df.select(column_list)
    return output_df
```

Cmd 3

```python
#function for overqriting the partition using insertInto
#First we rearrange the partition column to the end of df and then we set the partitionoverWriteMode to dynamic so that each time the insertInto gets executed it only writes new data instead of overwriting entire data
#The if stmt states that if the table alrdy exists then it will only add new data using insertInto or else if table doesnt exist it will overwrite the data only for the first time using saveAsTable stmt
def overwrite_partition(input_df, db_name, table_name, partition_column):
    output_df = re_arrange_partition_column(input_df, partition_column)
    spark.conf.set("spark.sql.sources.partitionOverwriteMode","dynamic")
    if (spark._jsparkSession.catalog().tableExists(f"{db_name}.{table_name}")):
        output_df.write.mode("overwrite").insertInto(f"{db_name}.{table_name}")
    else:
        output_df.write.mode("overwrite").partitionBy(partition_column).format("parquet").saveAsTable(f"{db_name}.{table_name}")
```

Cmd 4

```python
def df_column_to_list(input_df, column_name):
    df_row_list = input_df.select(column_name) \
                        .distinct() \
                        .collect()

    column_value_list = [row[column_name] for row in df_row_list]
    return column_value_list
```

- 🏪 Marketplace   NEW
- Partner Connect
- 1/4 Tasks Completed
- 👁 Enable new UI   NEW
- ☐ Menu options

5.ingest_results_file   Python ∨

File   Edit   View   Run   Help   Last edit was 2 days ago   Provide feedback

▶ Run all   ● dbkcourse_cluster ∨   🗓 Schedule   Share   ^

⊕ New

▮ Workspace

🔖 Repos

🕐 Recents

🔺 Data

🕸 Compute

☰ Workflows

p_data_source
```
Ergast
```

p_file_date
```
2021-03-21
```

🗒 Method 2

Cmd 22

```
overwrite_partition(results_final_df, 'f1_processed', 'results', 'race_id')
```

▸ (3) Spark Jobs

Command took 3.10 minutes -- by avinashchinta@my.unt.edu at 7/14/2023, 2:47:49 PM on dbkcourse_cluster

Cmd 23

```
dbutils.notebook.exit("Success")
```

Notebook exited: Success

Command took 0.70 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 2:47:49 PM on dbkcourse_cluster

Cmd 24

```sql
%sql
SELECT race_id, COUNT(1)
FROM f1_processed.results
GROUP BY race_id
ORDER BY race_id DESC;
```

▸ (3) Spark Jobs

▸ ▦ _sqldf: pyspark.sql.dataframe.DataFrame = [race_id: integer, count(1): long]

Table ∨   +

	race_id	count(1)
1	1047	20
2	1046	20
3	1045	20
4	1044	20
5	1043	20
6	1042	20
7	1041	20

⬇ 1,035 rows | 22.71 seconds runtime   Refreshed now

ⓘ SQL cell result stored as PySpark data frame _sqldf . Learn more

Command took 22.71 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 2:52:07 PM on dbkcourse_cluster

🛍 Marketplace   NEW

🔗 Partner Connect

1/4 Tasks Completed

◎ Enable new UI   NEW

▯ Menu options

⊕ New

▮ Workspace

⌖ Repos

⏱ Recents

⛁ Data

⌗ Compute

≡ Workflows

5.ingest_results_file Python ∨

File Edit View Run Help   Last edit was 2 days ago   Provide feedback

▶ Run all   ● dbkcourse_cluster ∨   🗓 Schedule   **Share**   ⌃

p_data_source
```
Ergast
```

p_file_date
```
2021-03-28
```

Method 2

Cmd 22

```
overwrite_partition(results_final_df, 'f1_processed', 'results', 'race_id')
```

▸ (1) Spark Jobs

Command took 2.30 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 2:53:50 PM on dbkcourse_cluster

Cmd 23

```
dbutils.notebook.exit("Success")
```

Notebook exited: Success

Command took 0.09 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 2:53:50 PM on dbkcourse_cluster

Cmd 24

SQL ▶▾ ⅄ ∨ − ✕

```sql
%sql
SELECT race_id, COUNT(1)
FROM f1_processed.results
GROUP BY race_id
ORDER BY race_id DESC;
```

▸ (3) Spark Jobs

▸ ▦ _sqldf: pyspark.sql.dataframe.DataFrame = [race_id: integer, count(1): long]

Table ∨   +

	race_id ▲	count(1) ▲	
1	1052	20	
2	1047	20	
3	1046	20	
4	1045	20	
5	1044	20	
6	1043	20	
7	1042	20	

↓ 1,036 rows | 15.74 seconds runtime   Refreshed now

ⓘ SQL cell result stored as PySpark data frame _sqldf . Learn more

Command took 15.74 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 2:54:02 PM on dbkcourse_cluster

🛒 Marketplace NEW

🔗 Partner Connect

1/4 Tasks Completed

⊙ Enable new UI NEW

▯ Menu options

New

Workspace

Repos

Recents

Data

Compute

Workflows

p_data_source
```
Ergast
```

p_file_date
```
2021-04-18
```

🖉 ⚙ 📌

Method 2

Cmd 22

```python
overwrite_partition(results_final_df, 'f1_processed', 'results', 'race_id')
```

▸ (1) Spark Jobs

Command took 1.80 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 2:55:04 PM on dbkcourse_cluster

Cmd 23

```python
dbutils.notebook.exit("Success")
```

Notebook exited: Success

Command took 0.08 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 2:55:04 PM on dbkcourse_cluster

Cmd 24

SQL ▶▾ ᴸᵃˡ ∨ ─ ✕

```sql
%sql
SELECT race_id, COUNT(1)
FROM f1_processed.results
GROUP BY race_id
ORDER BY race_id DESC;
```

▸ (3) Spark Jobs

▸ ▦ _sqldf: pyspark.sql.dataframe.DataFrame = [race_id: integer, count(1): long]

Table ∨   +

	race_id ▲	count(1) ▲
1	1053	20
2	1052	20
3	1047	20
4	1046	20
5	1045	20
6	1044	20
7	1043	20

⤓ 1,037 rows | 14.22 seconds runtime   Refreshed 4 minutes ago

ⓘ SQL cell result stored as PySpark data frame _sqldf . Learn more

Command took 14.22 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 2:55:15 PM on dbkcourse_cluster

Marketplace ⁿᵉʷ

Partner Connect

Tasks Completed

Enable new UI ⁿᵉʷ

Menu options

0.ingest_all_files   Python ∨
File   Edit   View   Run   Help   Last edit was 21 minutes ago   Provide feedback

▶ Run all   ● dbkcourse_cluster ∨   📅 Schedule   Share

Cmd 1

```python
v_result = dbutils.notebook.run("1.ingest_circuits_file", 0, {"p_data_source": "Ergast API", "p_file_date": "2021-03-21"})
```

Notebook job #417531727997483

Command took 31.19 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 3:10:16 PM on dbkcourse_cluster

Cmd 2

```python
v_result
```

Out[2]: 'Success'

Command took 0.13 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 3:10:16 PM on dbkcourse_cluster

Cmd 3

```python
v_result = dbutils.notebook.run("2.ingest_races_file", 0, {"p_data_source": "Ergast API", "p_file_date": "2021-03-21"})
```

Notebook job #311893066073158

Command took 41.14 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 3:10:16 PM on dbkcourse_cluster

Cmd 4

```python
v_result
```

Out[4]: 'Success'

Command took 0.09 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 3:10:16 PM on dbkcourse_cluster

Cmd 5

```python
v_result = dbutils.notebook.run("3.ingest_constructors_file", 0, {"p_data_source": "Ergast API", "p_file_date": "2021-03-21"})
```

Notebook job #696376804587306

Command took 21.82 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 3:10:16 PM on dbkcourse_cluster

Cmd 6

```python
v_result
```

Out[6]: 'Success'

Command took 0.10 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 3:10:16 PM on dbkcourse_cluster

Cmd 7

```python
v_result = dbutils.notebook.run("4.ingest_drivers_file", 0, {"p_data_source": "Ergast API", "p_file_date": "2021-03-21"})
```

Notebook job #181624116980309

Command took 20.82 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 3:10:16 PM on dbkcourse_cluster

Data Science & Engi... ▼

⊕ New

🔲 Workspace

📙 Repos

🕐 Recents

🔼 Data

🏠 Compute

☲ Workflows

▶ Run all   ● dbkcourse_cluster ∨   📅 Schedule   Share   ⌃

Find 2021-03-21   0 of 0 Replace 2021-03-28   Replace   Replace All   ‹ Prev   Next ›   ✕

Cmd 1
```
    v_result = dbutils.notebook.run("1.ingest_circuits_file", 0, {"p_data_source": "Ergast API", "p_file_date": "2021-03-28"})
```
Notebook job #417531727997483

Command took 31.19 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 3:10:16 PM on dbkcourse_cluster

Cmd 2
```
    v_result
```
Out[2]: 'Success'

Command took 0.13 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 3:10:16 PM on dbkcourse_cluster

Cmd 3
```
    v_result = dbutils.notebook.run("2.ingest_races_file", 0, {"p_data_source": "Ergast API", "p_file_date": "2021-03-28"})
```
Notebook job #311893066073158

Command took 41.14 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 3:10:16 PM on dbkcourse_cluster

Cmd 4
```
    v_result
```
Out[4]: 'Success'

Command took 0.09 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 3:10:16 PM on dbkcourse_cluster

Cmd 5
```
    v_result = dbutils.notebook.run("3.ingest_constructors_file", 0, {"p_data_source": "Ergast API", "p_file_date": "2021-03-28"})
```
Notebook job #696376804587306

Command took 21.02 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 3:10:16 PM on dbkcourse_cluster

Cmd 6
```
    v_result
```
Out[6]: 'Success'

Command took 0.10 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 3:10:16 PM on dbkcourse_cluster

Cmd 7
```
    v_result = dbutils.notebook.run("4.ingest_drivers_file", 0, {"p_data_source": "Ergast API", "p_file_date": "2021-03-28"})
```
Notebook job #181624116089300

📦 Marketplace NEW

🔗 Partner Connect

1/4 Tasks Completed

👁 Enable new UI NEW

🔲 Menu options

0.ingest_all_files Python ⌄
File Edit View Run Help   Provide feedback

▶ Run all   ● dbkcourse_cluster ⌄   🗓 Schedule   **Share**   ⌃

⊕ New

🔲 **Workspace**

⌈🗟 Repos

🕓 Recents

🔠 Data

🏠 Compute

≣ Workflows

Find `2021-03-28`   0 of 0   Replace `2021-04-18`   Replace   Replace All   ‹ Prev   Next ›   ✕

Cmd 1

```
    v_result = dbutils.notebook.run("1.ingest_circuits_file", 0, {"p_data_source": "Ergast API", "p_file_date": "2021-04-18"})
```

Notebook job #225713118702292

Command took 20.85 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 3:31:40 PM on dbkcourse_cluster

Cmd 2

```
    v_result
```

Out[21]: 'Success'

Command took 0.09 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 3:31:40 PM on dbkcourse_cluster

Cmd 3

```
    v_result = dbutils.notebook.run("2.ingest_races_file", 0, {"p_data_source": "Ergast API", "p_file_date": "2021-04-18"})
```

Notebook job #1109865327277878

Command took 48.90 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 3:31:40 PM on dbkcourse_cluster

Cmd 4

```
    v_result
```

Out[23]: 'Success'

Command took 0.09 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 3:31:40 PM on dbkcourse_cluster

Cmd 5

```
    v_result = dbutils.notebook.run("3.ingest_constructors_file", 0, {"p_data_source": "Ergast API", "p_file_date": "2021-04-18"})
```

Notebook job #214090257372898

Command took 20.92 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 3:31:40 PM on dbkcourse_cluster

Cmd 6

```
    v_result
```

Out[25]: 'Success'

Command took 0.10 seconds -- by avinashchinta@my.unt.edu at 7/14/2023, 3:31:40 PM on dbkcourse_cluster

Cmd 7

```
    v_result = dbutils.notebook.run("4.ingest_drivers_file", 0, {"p_data_source": "Ergast API", "p_file_date": "2021-04-18"})
```

Notebook job #37696032194273

🏪 Marketplace NEW

⧉ Partner Connect

1/4 Tasks Completed

👁 Enable new UI NEW

▯ Menu options