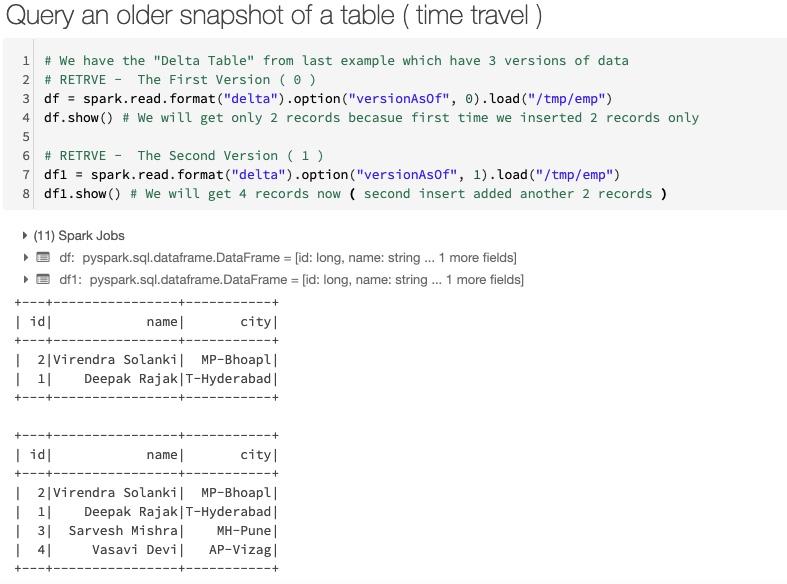
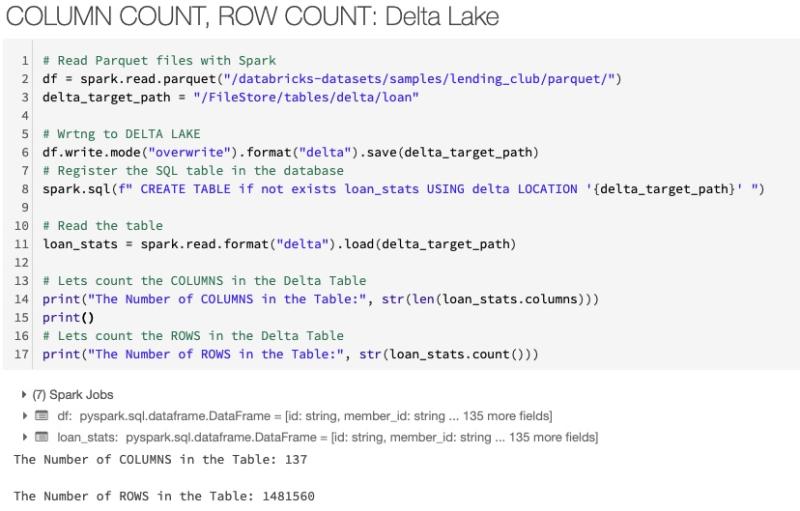
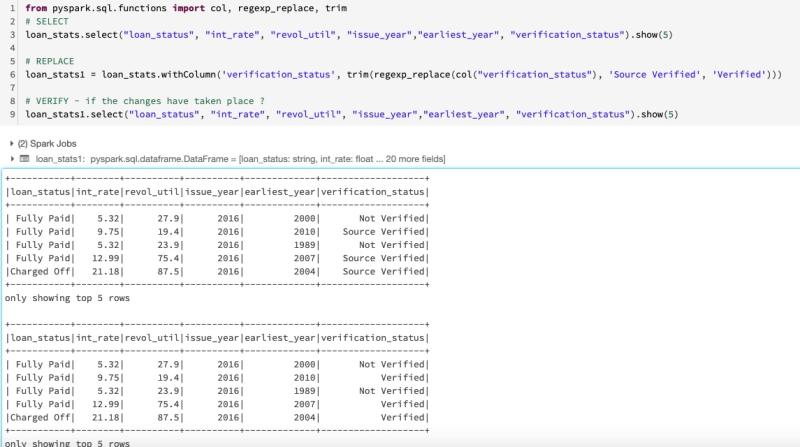
#Question151:  
Query an older snapshot of a Delta Table ( time travel ) in Spark ?  
  
By using the options:  
  
1. versionAsOf ( version starts from 0 )  
2. timestampAsOf



#Question152:  
COLUMN COUNT, ROW COUNT in Delta Lake ?  
  
By using  
  
1. df.columns - to get the columns  
2. Simple count() method



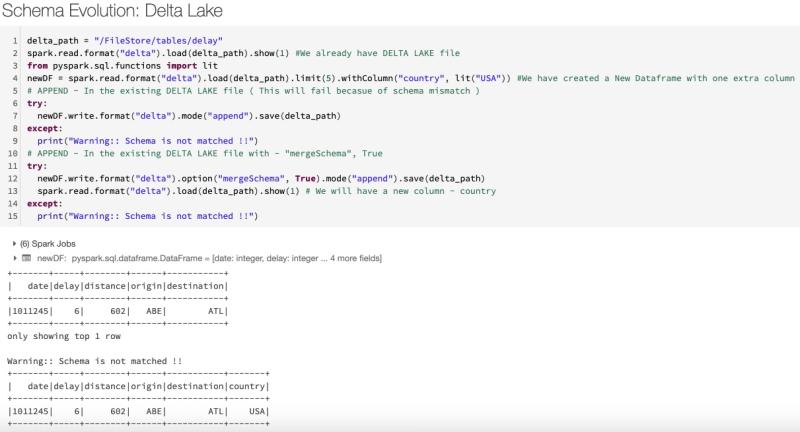
#Question153:  
How to REPLACE values of a Column in Spark Dataframe ?  
  
By using  
  
1. withColumn()  
2. regexp\_replace()



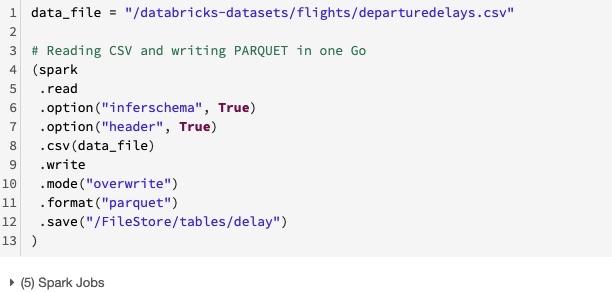
#Question154:  
SUBSTRACTING & ROUNDING of values of a Column in Spark Dataframe ?  
  
By using  
  
1. withColumn() method  
2. col() method and "-" operator  
3. Using round() function for rounding - In our Example, we have used upto 2 decimal points.



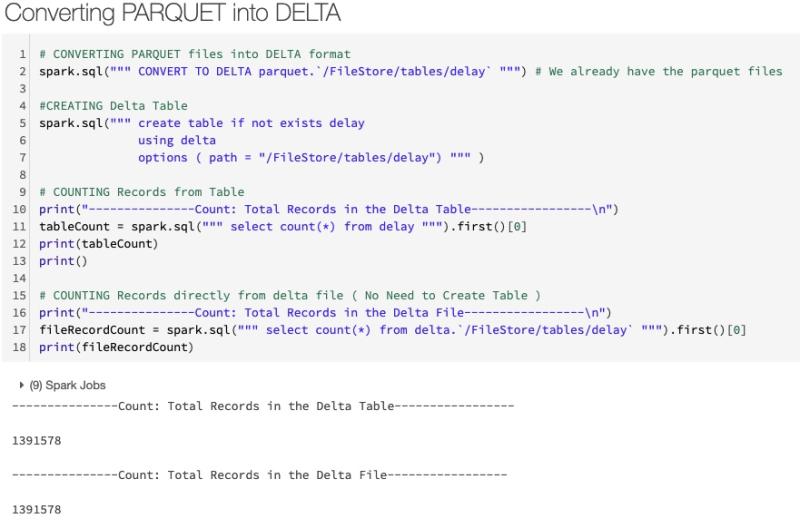
#Question155:  
What Is Schema Evolution in Delta Lake?  
  
Schema evolution is a feature that allows users to easily change a table’s current schema to accommodate data that is changing over time.  
  
Most commonly, it’s used when performing an append or overwrite operation, to automatically adapt the schema to include one or more new columns.  
  
How Does Schema Evolution Work?  
  
In the below example we can easily use schema evolution to add the new column that in the normal process will be rejected due to a schema mismatch.  
  
Schema evolution is activated by adding .option('mergeSchema', 'true') to your .write or .writeStream Spark command.



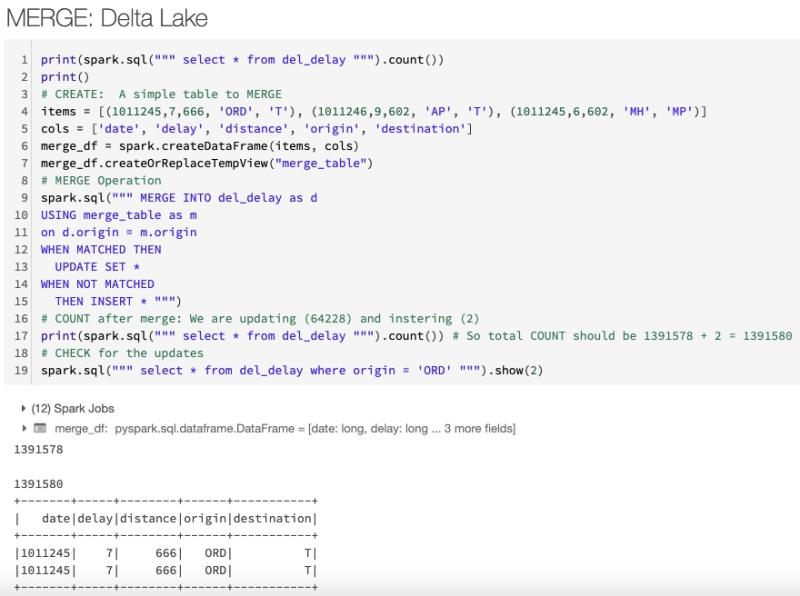
#Question156:  
Reading CSV and writing PARQUET in one Go in Spark ( without creating Dataframe variable )?  
  
We can chain the .read and .write method one after the another without creating the Dataframe variable.



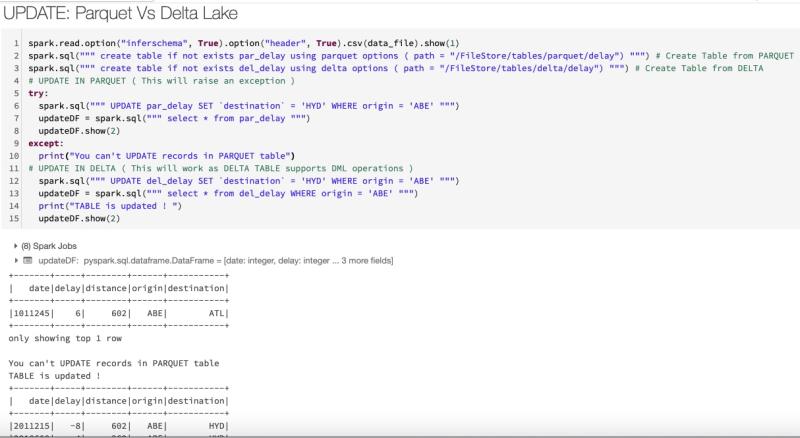
#Question157:  
Converting PARQUET into DELTA Lake format ?  
  
Using the command -  
  
CONVERT TO DELTA



#Question158:  
MERGE operation in DELTA Lake Table ?  
  
Using the command -  
  
MERGE Target\_Table into USING Delta\_Table



#Question159:  
UPDATE operation in DELTA Lake Table VS PARQUET file?  
  
We can directly UPDATE the DELTA Table because it supports the DML operations but we can't update tables backed by PARQUET.



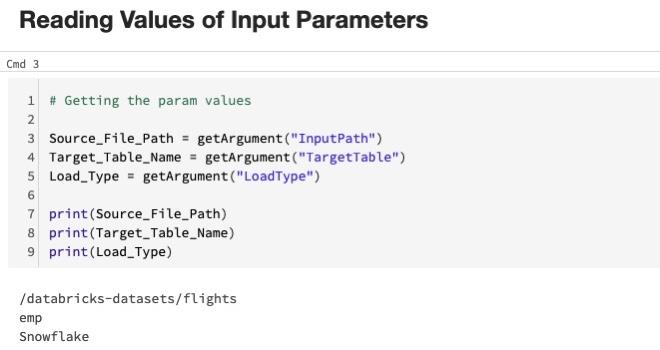
#Question160:  
DELETE operation in DELTA Lake Table VS PARQUET file?  
  
We can directly DELETE records from the DELTA Table because it supports the DML operations but we can't delete records from tables backed by PARQUET.



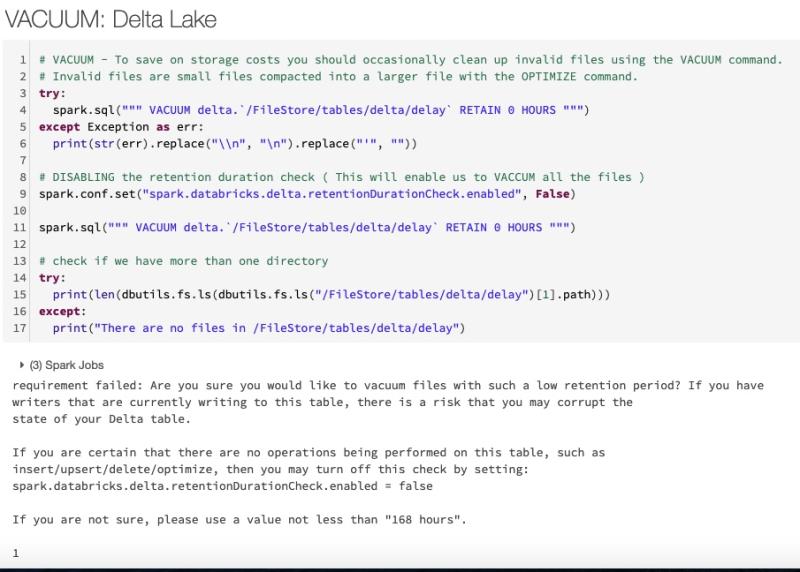
#Question161:  
How can we CREATE parameters which can be passed during runtime Or How can we run a Notebook with different sets of parameters ?  
  
By Using :  
  
dbutils.widgets utility.  
  
first argument : Parameter Name  
second argument: Default Value  
third argument: Parameter Caption



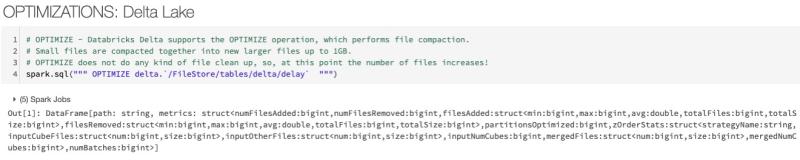
#Question162:  
How can we Read parameters which is passed during runtime in Notebook Or How can we run a Notebook with different sets of parameters ?  
  
By Using :  
  
getArgument() method.



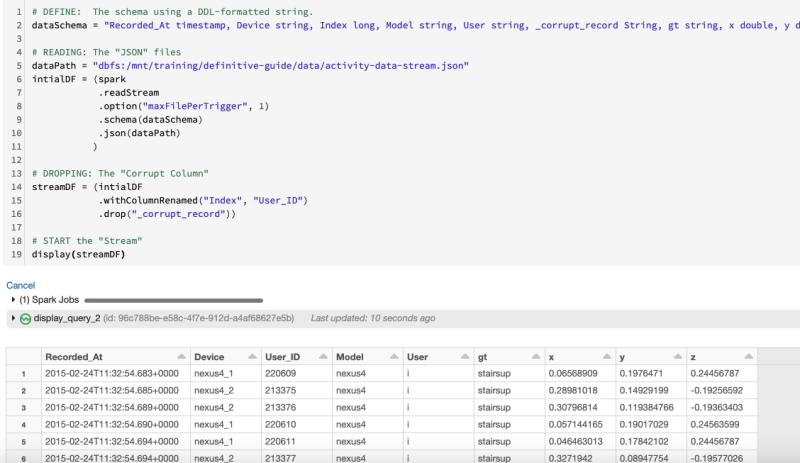
#Question163:  
VACCUM in DELTA LAKE ?  
  
You can remove files no longer referenced by a Delta table and are older than the retention threshold by running the vacuum command on the table. vacuum is not triggered automatically. The default retention threshold for the files is 7 days.  
  
Note:  
  
1. VACCUM deletes only data files, not log files. Log files are deleted automatically and asynchronously after checkpoint operations. The default retention period of log files is 30 days, configurable through the delta.logRetentionPeriod property which you set with the ALTER TABLE SET TBLPROPERTIES SQL method. See Table properties.  
  
2. The ability to time travel back to a version older than the retention period is lost after running vacuum.



#Question164:  
OPTIMIZE in DELTA LAKE ?  
  
Databricks DELTA LAKE supports the OPTIMIZE operation, which performs file compaction.  
  
1. Small files are compacted together into new larger files up to 1GB.  
2. OPTIMIZE does not do any kind of file clean up, so, at this point the number of files increases!



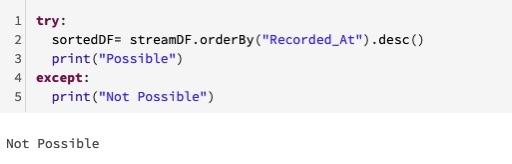
#Question165:  
READING streaming JSON data from the file ( S3/ HDFS / Blob ) location in Spark?  
  
Using the:  
  
spark.readStream method.  
  
The DataFrame API is unified and it works in the same manner for Stream the way it works for batch data.



#Question166:  
Find out if the Dataframe is "Streaming" in Spark?  
  
Using the:  
  
.isStreaming method



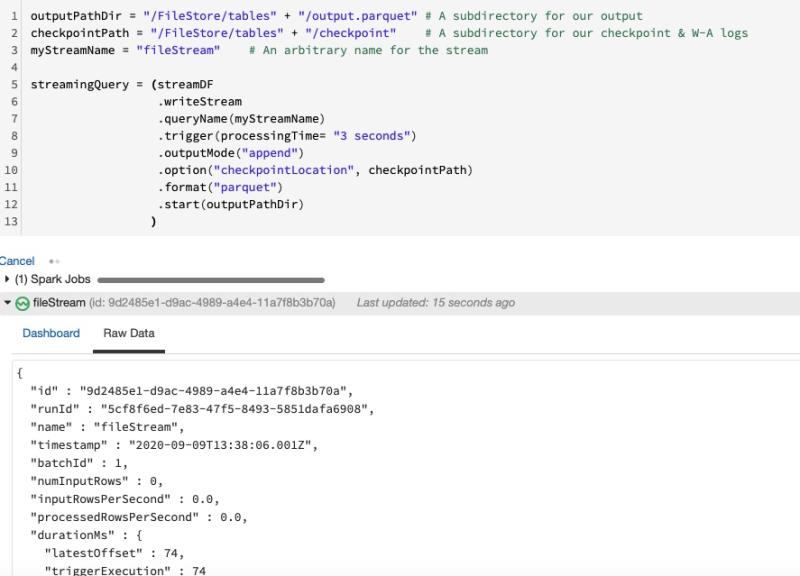
#Question167:  
What are the Operations Not possible in Structured Streaming in Spark?  
  
Operations like Sort, OrderBy etc are not possible in Streaming Data.



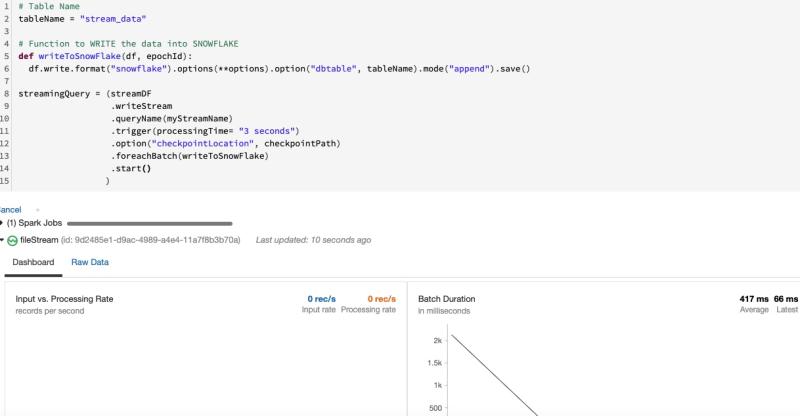
#Question168:  
How can we list all the "Active" streaming Query in Spark?  
  
By Using:  
  
spark.streams.active method.



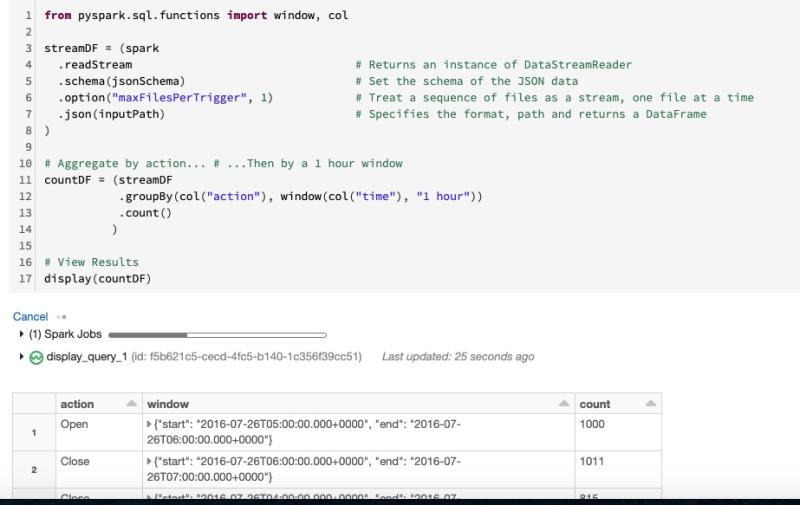
#Question169:  
How to WRITE streaming DATA in Spark in PARQUET format ?  
  
By Using:  
  
.writeStream method  
  
options:  
format = "parquet"  
checkpointLocation = A subdirectory for our checkpoint & W-A logs  
Output Directory = A subdirectory for our output



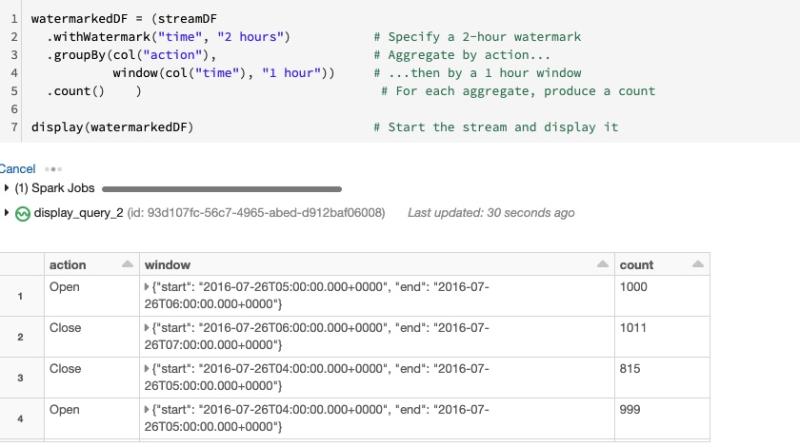
#Question170:  
How to WRITE streaming DATA to SNOWFLAKE in Spark ?  
  
By Using:  
.writeStream method  
  
A Custom function need to be written to write the data into Snowflake. Direct loading is not supported at this point.  
  
options:  
format = "snowflake"  
checkpointLocation = A subdirectory for our checkpoint & W-A logs



#Question171:  
How to do WINDOW operations in Spark on Streaming Data ?  
  
We have two kinds of Window Operations in Spark Structured Streaming.  
  
Sliding windows  
  
The windows overlap and a single event may be aggregated into multiple windows.  
  
Tumbling Windows  
  
The windows do not overlap and a single event will be aggregated into only one window.  
  
  
By Using:  
  
1. GroupBy() Method  
  
2. Window Function



#Question172:  
WATERMARKING operations in Spark on Streaming Data ?  
  
Watermarking  
  
it defines a cut-off. A point after which Structured Streaming will commit windowed data to sink, or throw it away if the sink is console or memory.  
  
In the example below:  
  
1. Data received 2 hour past the watermark will be dropped.  
2. Data received within 2 hours of the watermark will never be dropped.  
  
More specifically, any data less than 2 hours behind the latest data processed till then is guaranteed to be aggregated.  
  
The more delayed the data is, the less likely the engine is going to process it.



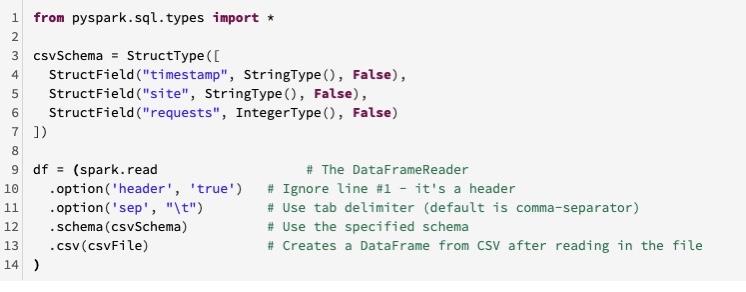
#Question173:  
How many SPARK Jobs will be generated from the below code & Why ?  
  
In the below example:  
  
1. We are reading the TSV file from spark.read.csv method  
2. Letting our method know - first row is header  
3. We are saying - file is "tab" delimited.



#Question174:  
How many SPARK Jobs will be generated from the below code & Why ?  
  
In the below example:  
  
1. We are reading the TSV file from spark.read.csv method  
2. Letting our method know - first row is header  
3. We are saying - file is "tab" delimited.  
4. Letting our method know - inferSchema is True



#Question175:  
How many SPARK Jobs will be generated from the below code & Why ?  
  
In the below example:  
  
1. We are reading the TSV file from spark.read.csv method  
2. Letting our method know - first row is header  
3. We are saying - file is "tab" delimited.  
4. Giving the User Defined Schema



#Question176:  
How many SPARK Jobs will be generated from the below code & Why ?  
  
In the below example:  
  
1. We are reading the JSON file from spark.read.json method  
2. We are saying - Inferschema is True



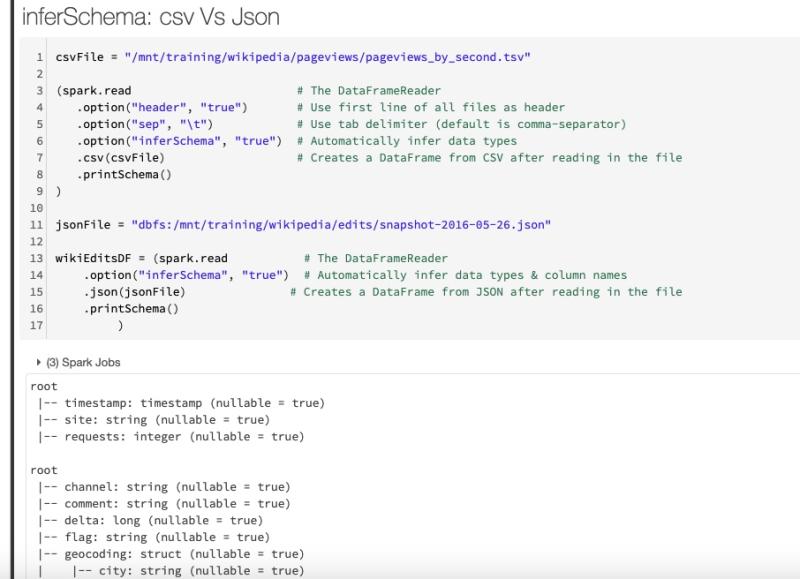
#Question177:  
Explain how does SPARK do the Memory Management ?

Here is the detailed article, I have written on Spark Memory Management.  
  
https://lnkd.in/gexWj-8

#Question178:  
How many SPARK Jobs will be generated from the below code & Why ?  
  
In the below example:  
  
1. We are reading the PARQUET file from spark.read.parquet method  
2. Note: PARQUET stores metadata information along with the actual data in the folder.

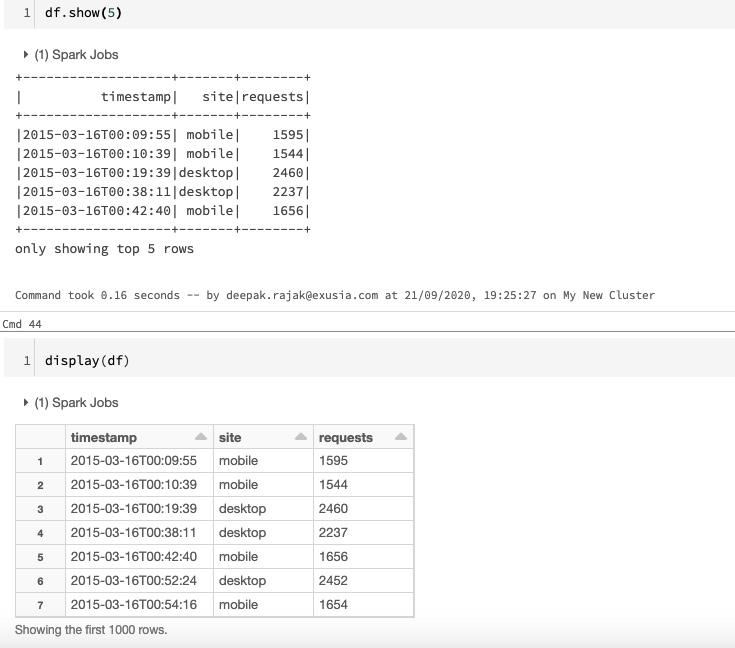


#Question179:  
inferSchema: CSV Vs JSON ?  
  
This is in continuation of the previous questions. Lots of you gave the correct answers but some of you got confused so let me clarify it.  
  
While there are similarities between reading in CSV & JSON there are some key differences:  
  
In case of CSV - 2 Jobs  
In case of JSON - 1 Job  
  
1. We only need one job even when inferring the schema in case of JSON.  
  
2. There is no header which is why there isn't a second job in case of JSON. The column names are extracted from the JSON object's attributes.  
  
3. Unlike CSV which reads in 100% of the data, the JSON reader only samples the data.

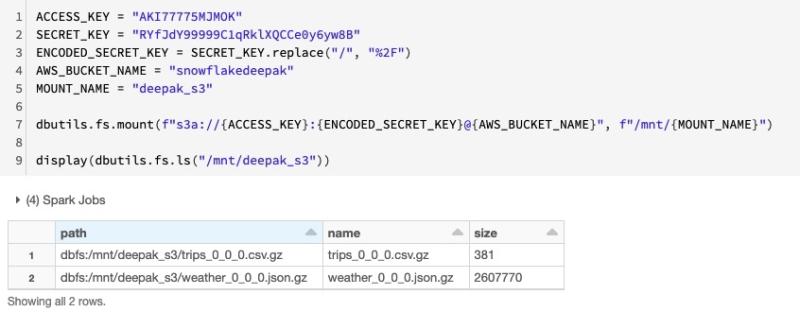


#Question180:  
Explain what is the Role of Catalyst and Tungsten in Spark ?  
  
Here is the detailed article, I have written on Catalyst and Tungsten Optimizer  
  
https://lnkd.in/g9DQSjy

#Question181:  
show(..) vs display(..) in Spark ( Databricks ) ?  
  
1. show(..) is part of core spark - display(..) is specific to our Databricks notebooks.  
  
2. show(..) has parameters for truncating both columns and rows - display(..) does not.  
  
3. show(..) is a function of the DataFrame/Dataset class - display(..) works with a number of different objects.  
  
4. display(..) is more powerful - with it, you can...  
  
a. Download the results as CSV  
b. Render line charts, bar chart & other graphs, maps and more.  
c. See up to 1000 records at a time.  
  
  
#Note  
  
Like DataFrame.show(..), display(..) is an action which triggers a job.

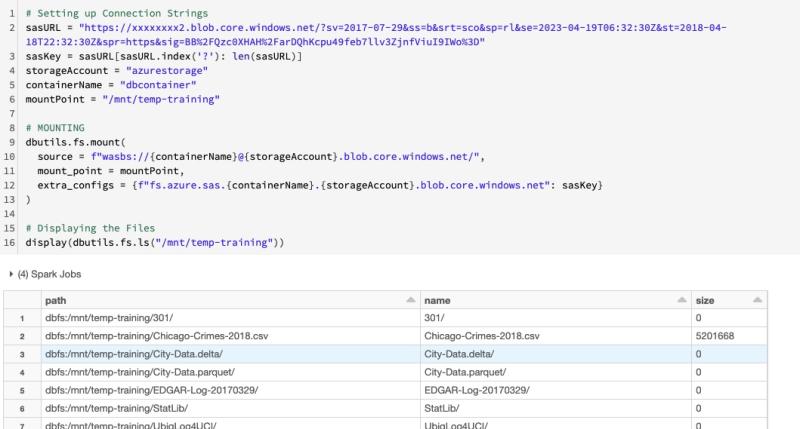


#Question182:  
How can we mount AWS S3 Bucket into Databricks File System ( DBFS )?  
  
We can mount in the following ways:  
  
1. We need the bucket name  
  
2. We need below keys for the AWS Bucket:  
  
AWS\_ACCESS\_KEY\_ID  
AWS\_SECRET\_ACCESS\_KEY  
  
3. We can use - dbutils.fs.mount command to mount the AWS bucket in DBFS  
  
4. We can list the files from the mount location.

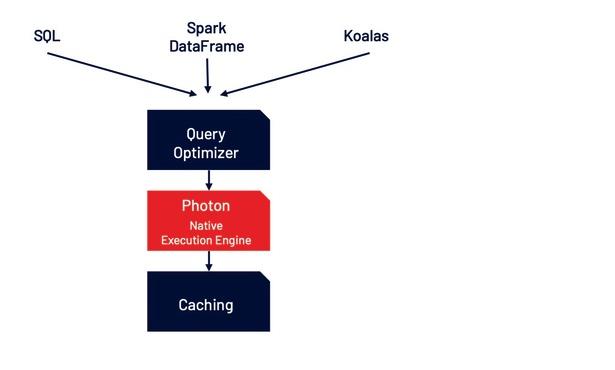


#Question183:  
Explain the Anatomy of Apache Spark's RDD ?  
  
I have written the detailed article on Spark's RDD - Resilient Distributed Datasets.  
  
Link is below.  
  
https://lnkd.in/gevnE3Z

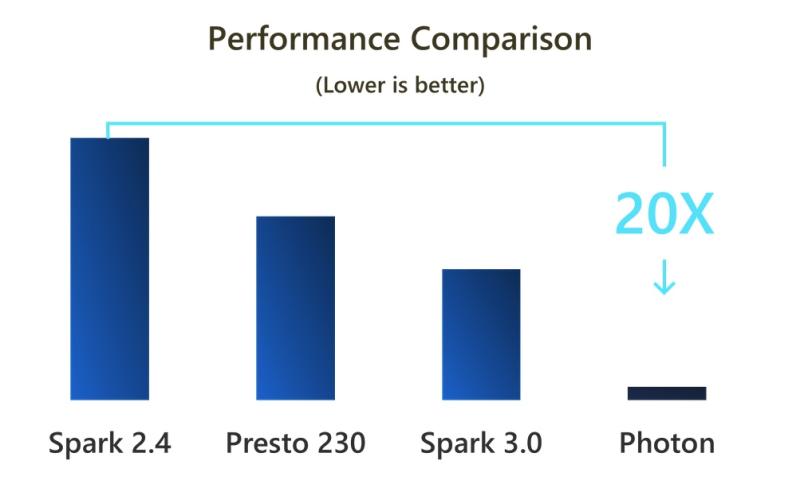
#Question184:  
How can we mount Azure BLOB Containers into Databricks File System ( DBFS )?  
  
We can mount in the following ways:  
  
1. We need the Container name  
  
2. We need Storage Account name  
  
3. We need SAS Url for the Storage Account  
  
4. We can use - dbutils.fs.mount command to mount the Azure Blob in DBFS  
  
4. We can list the files from the mount location.



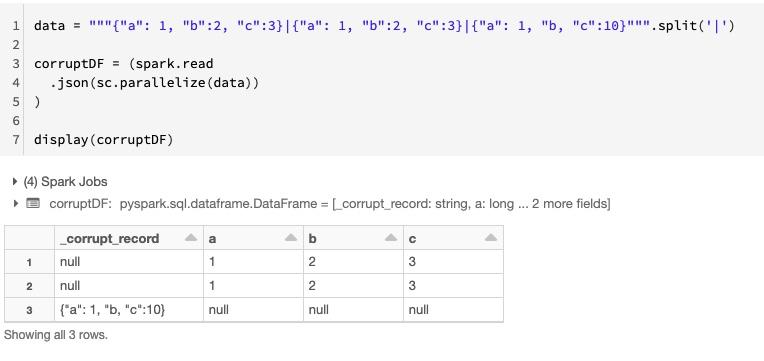
#Question185:  
What is a Photon powered Delta Engine in Azure Databricks ?  
  
Photon powered Delta Engine on Azure Databricks – fast, easy, and collaborative Analytics and AI service.  
  
Below are the Key features:  
  
1. Built from scratch in C++ and fully compatible with Spark APIs, Photon is a vectorized query engine that leverages modern CPU architecture along with Delta Lake to enhance Apache Spark 3.0’s performance by up to 20x.  
  
2. Photon is one of the three key components of Delta Engine in addition to an improved query optimizer and a caching layer.  
  
3. Together, these three components accelerate performance for big data use cases such as data engineering, data science, machine learning, and data analytics.



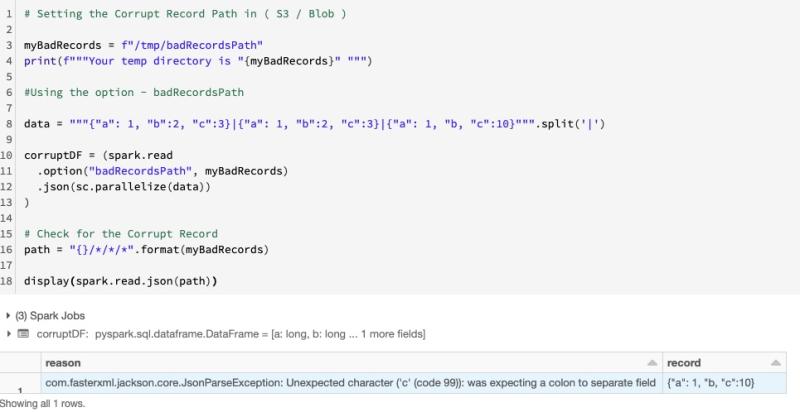
#Question186:  
How much Performance Improvement Photon Engine brings over Open Source Spark ?  
  
Delta Engine’s 3 components:  
1) Query optimizer  
2) Photon native execution engine  
3) Caching  
  
Performance Comparison between Databricks Vs Open Source Spark:  
  
1. Azure Databricks was already blazing fast compared to Apache Spark, and now, the Photon powered Delta Engine enables even faster performance for modern analytics and AI workloads on Azure.  
  
2. It is found the the Photon powered Delta Engine to be 20x faster than Spark 2.4.



#Question187:  
How to handle corrupt records in Spark ? What is the default behaviour of Spark Dataframe to handle corrupt records ?  
  
We can handle corrupt records by setting the .mode parameter. We have the following option for mode while reading the data.  
  
PERMISSIVE - Includes corrupt records in a "\_corrupt\_record" column (by default)  
  
DROPMALFORMED - Ignores all corrupted records  
  
FAILFAST - Throws an exception when it meets corrupted records  
  
#DefaultBehaviour:  
  
By default Spark includes corrupt records in a "\_corrupt\_record" column  
  
#Example: In the below example the 3rd record is the corrupt record which is captured in the "\_currupt\_record" column

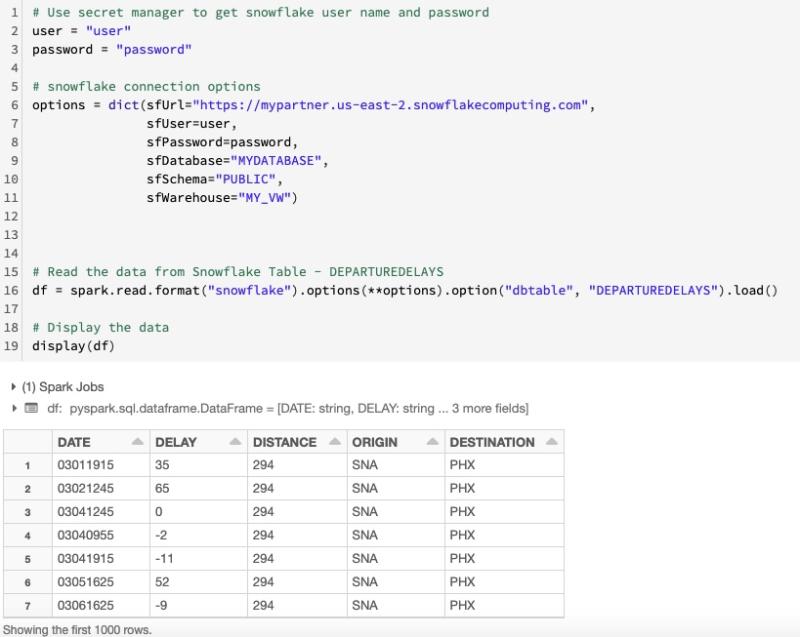


#Question188:  
How to store corrupt records in Spark for analysing later in the Cloud Data storage ( S3 / Blob ) ?  
  
We can store corrupt records by setting the .option with #badRecordsPath parameter.  
  
#Note - This is just available in Databricks only  
  
It persists the corrupt records for later analysis even after the cluster shuts down.

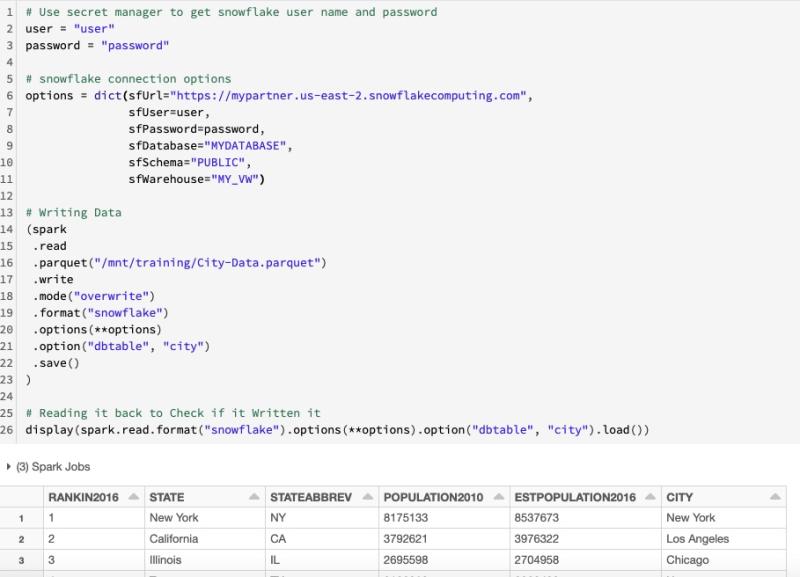


#Question189: What is Adaptive Query Execution & Dynamic Partition Pruning in Spark 3.0 ?  
  
I have written a detailed article on Spark 3.0 - Adaptive Query Execution & Dynamic Partition Pruning. Please go through it.  
  
Link is below:  
  
https://lnkd.in/edq5Dru

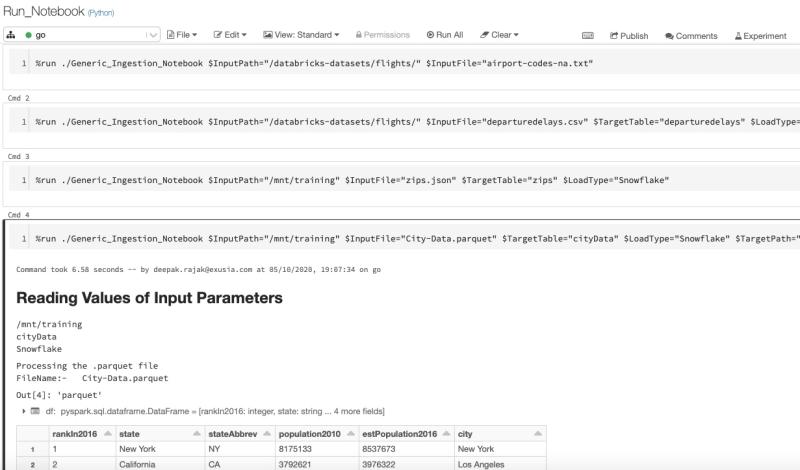
#Question190: How to Read from SNOWFLAKE tables via Spark ?  
  
SNOWFLAKE is a "SaaS" offerings on the Public Clouds & arguably the best Cloud Datawarehouse at this point.  
  
To read from Snowflake:  
  
1. Define the snowflake connection properties ( SNOWFLAKE URL, USER , PASSWORD , DATABASE, SCHEMA & VIRTUAL WAREHOUSE )  
  
2. Use format == "snowflake"  
  
3. provide the table name - in the case ( departuredelays )



#Question191: How to Write in SNOWFLAKE tables via Spark ?  
  
SNOWFLAKE is a "SaaS" offerings on the Public Clouds & arguably the best Cloud Datawarehouse at this point.  
  
To WRITE in Snowflake:  
  
1. Define the snowflake connection properties ( SNOWFLAKE URL, USER , PASSWORD , DATABASE, SCHEMA & VIRTUAL WAREHOUSE )  
  
2. Use format == "snowflake"  
  
3. provide the table name - in the case ( CITY )

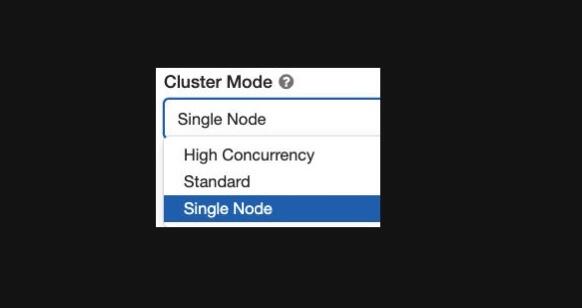


#Question192: How can We run the Spark Notebook from a different Notebook ?  
  
We can use - %run command to run a Notebook from another Notebook.  
  
We can pass different parameters into same Notebook in this way.  
  
#Note -  
In the below example, We are running a notebook - "Generic\_Ingestion\_Notebook" from another notebook - "Run\_Notebook"



#Question193: How to create a Generic Data Ingestion & Load Process in Apache Spark ?  
  
I have written a detailed step b step process to create a Generic Data Ingestion Process in Apache Spark on Databricks Notebooks.  
  
Here is the link. Please go through & let me know in case of any questions.  
  
https://lnkd.in/gq\_YmFi

#Question194: What are the Single Node clusters in Azure Databricks ?  
  
A Single Node cluster is a cluster consisting of a Spark driver and no Spark workers. Such clusters support Spark jobs and all Spark data sources, including Delta Lake.  
  
In contrast, Standard clusters require at least one Spark worker to run Spark jobs.  
  
Single Node clusters are helpful in the following situations:  
  
1. Running single node machine learning workloads that need Spark to load and save data  
2. Lightweight exploratory data analysis (EDA)  
  
Single Node cluster properties:  
  
A Single Node cluster has the following properties:  
  
1. Runs Spark locally with as many executor threads as logical cores on the cluster (the number of cores on driver - 1).  
2. Has 0 workers, with the driver node acting as both master and worker.  
3. The executor stderr, stdout, and log4j logs are in the driver log.  
4. Cannot be converted to a Standard cluster. Instead, create a new cluster with the mode set to Standard.



#Question195: How to Read from REDSHIFT tables via Spark ?  
  
Amazon Redshift is a fully managed, petabyte-scale data warehouse service in the cloud. You can start with just a few hundred gigabytes of data and scale to a petabyte or more.  
  
To read from REDSHIFT:  
  
1. Define the redshift connection properties ( URL, USER , PASSWORD )  
  
2. Use format == "com.databricks.spark.redshift"  
  
3. Provide the table name - in the case ( departuredelays )  
  
4. Provide the Temp S3 Directory - tempdir ( to hold the temp files )  
  
5. Set up the AWS S3 Keys ( Or IAM Role )

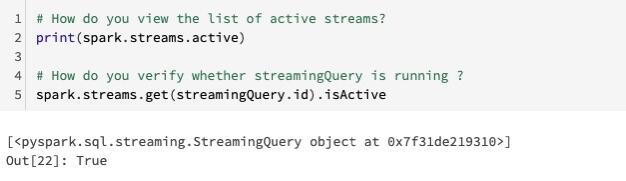


#Question196: How to WRITE into REDSHIFT tables via Spark ?  
  
Amazon Redshift is a fully managed, petabyte-scale data warehouse service in the cloud. You can start with just a few hundred gigabytes of data and scale to a petabyte or more.  
  
To write into REDSHIFT:  
  
1. Define the redshift connection properties ( URL, USER , PASSWORD )  
  
2. Use format == "com.databricks.spark.redshift"  
  
3. Provide the table name - in the case ( departuredelays )  
  
4. Provide the Temp S3 Directory - tempdir ( to hold the temp files )  
  
5. Set up the AWS S3 Keys ( Or IAM Role )  
  
6. Set tempformat to "CSV" ( default is AVRO ) - CSV is at least 50% faster than AVRO



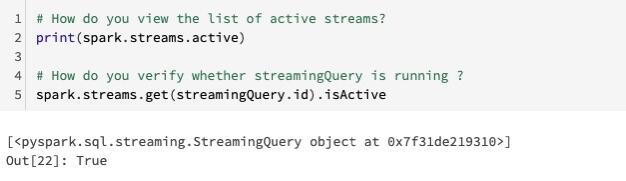
#Question197: What is the difference between SNOWFLAKE & DATABRICKS. How can we Ingesting, Parsing and Querying Semi Structured Data (JSON) into Snowflake Vs Databricks?  
  
Here is the detailed article I have written on - Ingesting, Parsing and Querying Semi Structured Data (JSON) into Snowflake Vs Databricks  
  
Below is the link. Please let me know in case of any questions.  
  
https://lnkd.in/gsuQCPa

#Question198: How do We view the list of active streams? How to verify whether a particular streaming query is running or not in Structured Streaming ?  
  
1. Invoke spark.streams.active to get the "ACTIVE" streams.  
  
2. Invoke spark.streams.get(streamingQuery.id).isActive to get whether the streaming is happening or not. It will give us BOOLEAN Output.

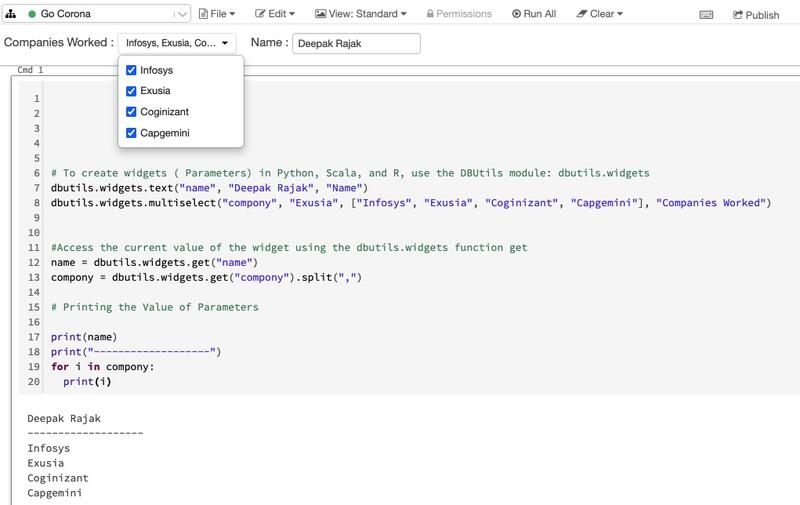


#Question197: What is the difference between SNOWFLAKE & DATABRICKS. How can we Ingesting, Parsing and Querying Semi Structured Data (JSON) into Snowflake Vs Databricks?  
  
Here is the detailed article I have written on - Ingesting, Parsing and Querying Semi Structured Data (JSON) into Snowflake Vs Databricks  
  
Below is the link. Please let me know in case of any questions.  
  
https://lnkd.in/gsuQCPa

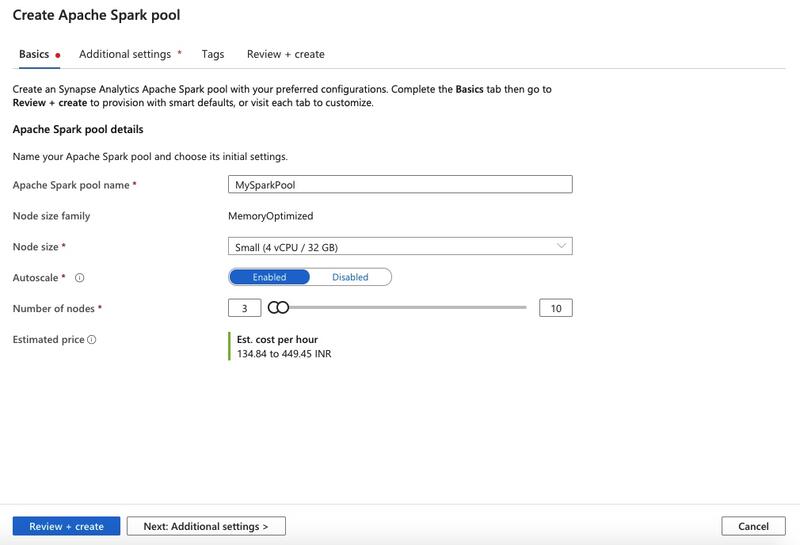
#Question198: How do We view the list of active streams? How to verify whether a particular streaming query is running or not in Structured Streaming ?  
  
1. Invoke spark.streams.active to get the "ACTIVE" streams.  
  
2. Invoke spark.streams.get(streamingQuery.id).isActive to get whether the streaming is happening or not. It will give us BOOLEAN Output.



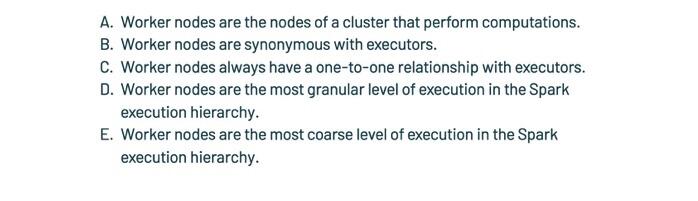
#Question199: How to create PARAMETERS in DATABRICKS & read the value from them ? How to remove all the parameters from the Notebook ?  
  
1. To create widgets ( Parameters) in Databricks, use the DBUtils module: dbutils.widgets  
  
2. Access the current value of the widget using the dbutils.widgets function get.  
  
3. Also we can remove all the Widgets ( Parameters ) by calling - dbutils.widgets.removeAll()



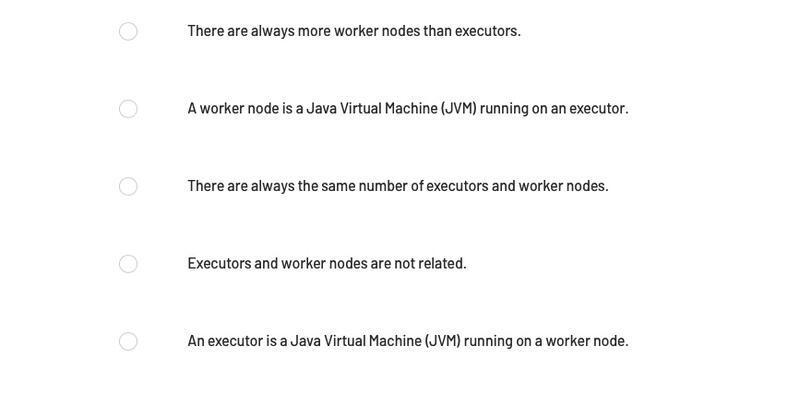
#Question200: What is Apache Spark pool in Azure Synapse Analytics ?  
  
Azure Synapse Analytics offers various analytics engines to help you ingest, transform, model, analyze, and distribute your data.  
  
An Apache Spark pool provides open-source big data compute capabilities. ( Comes with Delta Lake as well )  
  
After we created an Apache Spark pool in your Synapse workspace, data can be loaded, modeled, processed, and distributed for faster analytic insight.  
  
To Create a Spark Pool:  
  
1. Navigate to the Synapse workspace  
  
2. Create new Apache Spark pool ( Provide appropriate name )  
  
3. Set up Node Size - Small / Medium / Large ( Cost will change according to it )  
  
4. Autoscale can be enabled or disabled  
  
5. Number of Nodes ( Provide how many nodes you want )  
  
6. You can set additional settings like version , auto pause etc



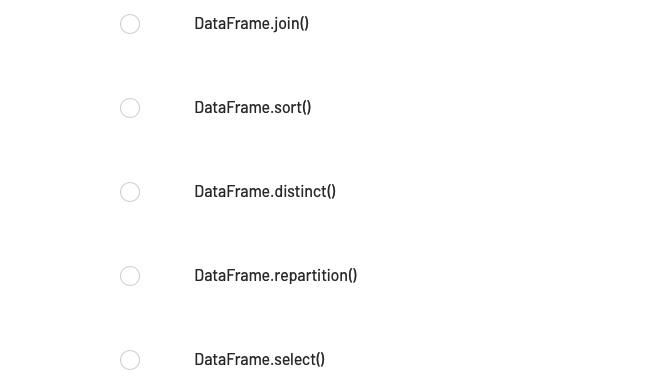
#Question201: What describes a WORKER NODE in Spark ?  
  
Hi Guys, Please comment your answer in the comment section. I will comment the CORRECT answer after 24 Hours.



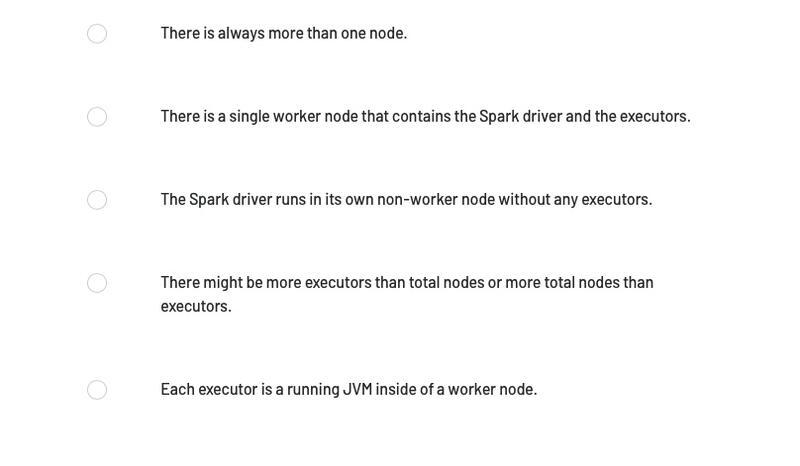
#Question202: Which of the following describes the relationship between WORKER nodes and EXECTORS?  
  
Hi Guys, Please comment your answer in the comment section. I will comment the CORRECT answer after 24 Hours.



#Question203: Which of the following DataFrame operations is always classified as a narrow transformation?  
  
Hi Guys, Please comment your answer in the comment section. I will comment the CORRECT answer after 24 Hours.



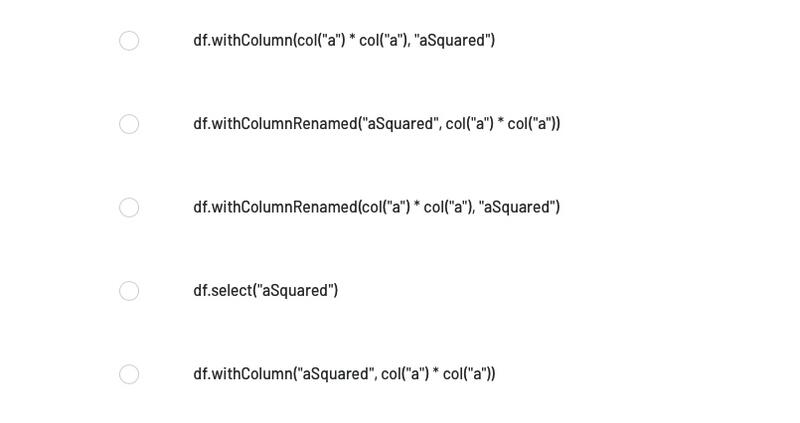
#Question204: If Spark is running in cluster mode, which of the following statements about nodes is incorrect ?  
  
Hi Guys, Please comment your answer in the comment section. I will comment the CORRECT answer after 24 Hours.



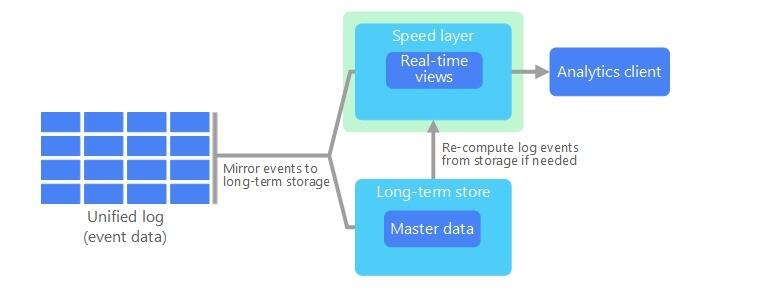
#Question205: Which of the following operations can be used to create a new DataFrame with a new column and all previously existing columns form an existing DataFrame ?  
  
Hi Guys, Please comment your answer in the comment section. I will comment the CORRECT answer after 24 Hours.



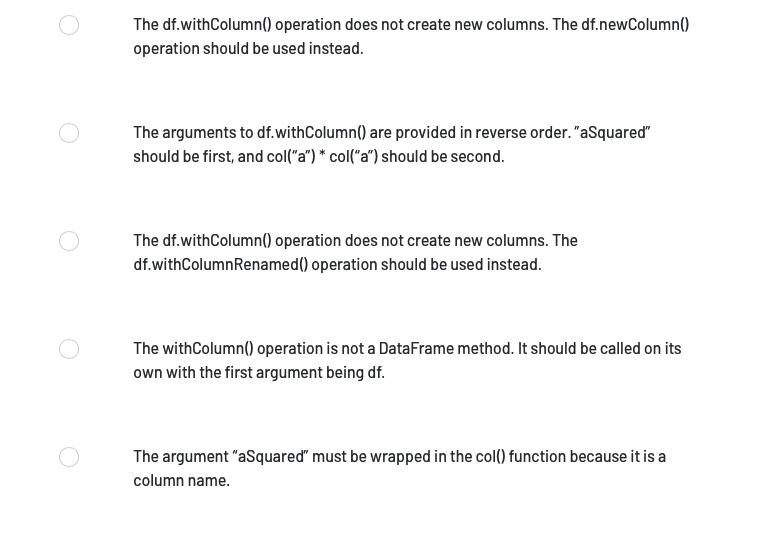
#Question206: Which of the following code blocks returns a DataFrame with a new column aSquared and all previously existing columns from DataFrame df ?  
  
Hi Guys, Please comment your answer in the comment section. I will comment the CORRECT answer after 24 Hours.



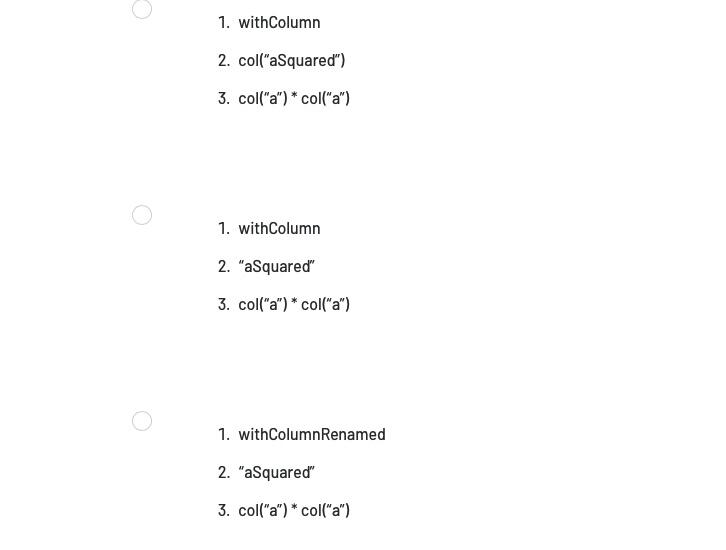
#Question207: What is a KAPPA Architecture & How it is different from the Lambda Architecture ?  
  
A drawback to the Lambda Architecture is its complexity. Processing logic appears in two different places — the cold and hot paths — using different frameworks. This leads to duplicate computation logic and the complexity of managing the architecture for both paths.  
  
The KAPPA Architecture was proposed by Jay Kreps as an alternative to the lambda architecture.  
  
It has the same basic goals as the lambda architecture, but with an important distinction: - All data flows through a single path, using a stream processing system.  
  
There are some similarities to the lambda architecture's batch layer, in that the event data is immutable and all of it is collected, instead of a subset.  
  
The data is ingested as a stream of events into a distributed and fault tolerant unified log.  
  
These events are ordered, and the current state of an event is changed only by a new event being appended.  
  
Similar to a lambda architecture's speed layer, all event processing is performed on the input stream and persisted as a real-time view.  
  
If you need to recompute the entire data set , you simply replay the stream.



#Question208: The code block shown below contains an error. The code block is intended to return a DataFrame with a new column aSquared and all previously existing columns from DataFrame df. Identify the error.  
  
Code block:  
  
df.withColumn(col(“a”) \* col(“a”), “aSquared”)  
  
Hi Guys, Please comment your answer in the comment section. I will comment the CORRECT answer after 24 Hours.



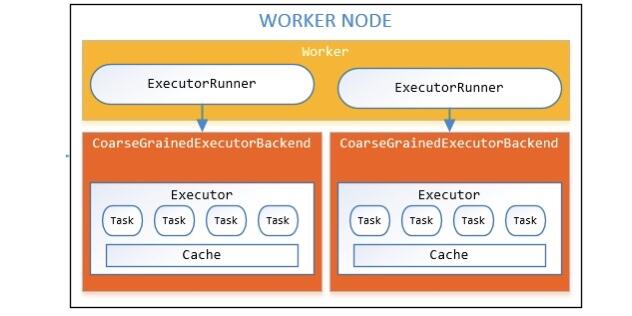
#Question209: The code block shown below should return a DataFrame with a new column aSquared and all previously existing columns from DataFrame df. Choose the response that correctly fills in the numbered blanks within the code block to complete this task.  
  
Code block:  
  
df.\_\_1\_\_(\_\_2\_\_, \_\_3\_\_)  
  
  
Hi Guys, Please comment your answer in the comment section. I will comment the CORRECT answer after 24 Hours.



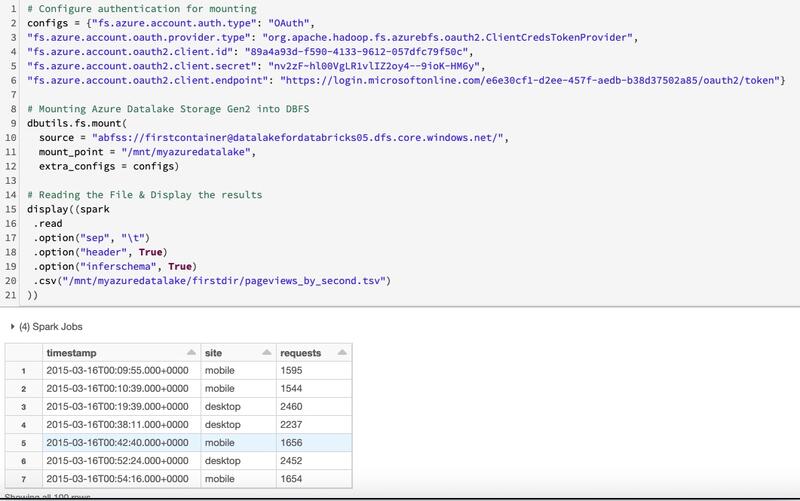
#Question210: In what order should the below lines of code be run in order to return a DataFrame with a new column aSquared and all previously existing columns from DataFrame df?  
  
  
1. df  
2. .withColumn(“aSquared”, “a” \* “a”)  
3 .withColumn(“aSquared”, col(“a”) \* col(“a”))  
4. DataFrame  
5. .withColumn(col(“aSquared”), col(“a”) \* col(“a”)  
  
  
Hi Guys, Please comment your answer in the comment section. I will comment the CORRECT answer after 24 Hours.



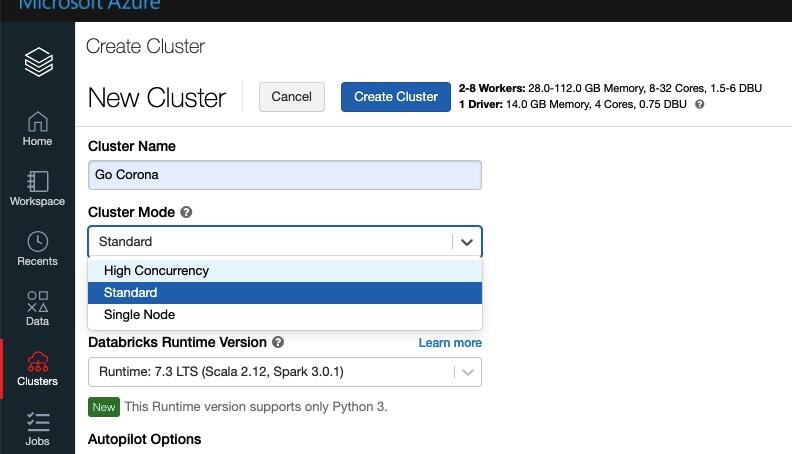
#Question211: Can a WORKER NODE hold more than one EXECUTOR process? or one worker node holds one executor?  
  
  
A worker node can hold multiple executors processes provided it has sufficient CPU, Memory and Storage.  
  
Workers hold many executors, for many applications. One application has executors on many workers.  
  
Note: Number of executors in a worker node at a given point of time entirely depends on workload on the cluster and capability of the node(Machine) to run multiple executors process.



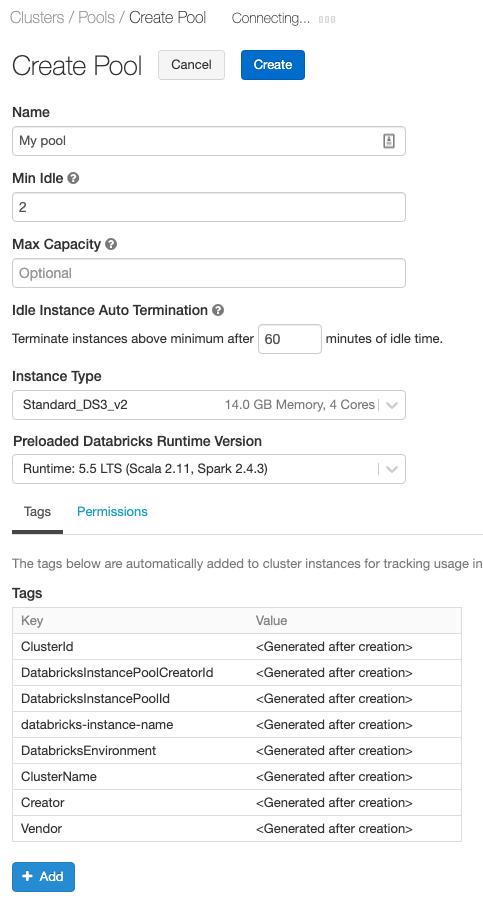
#Question212: How to MOUNT - Azure Datalake Storage Gen2 into Databricks File System ( DBFS )?  
  
  
#Step1: Create an Azure Data Lake Storage Gen2 account and initialize a filesystem  
  
#Step2: Create and grant permissions to service principal  
  
#1. Create an Azure AD application and service principal that can access resources. Note the following properties. We need to pass them into our mount command.  
  
application-id: An ID that uniquely identifies the application.  
  
directory-id: An ID that uniquely identifies the Azure AD instance.  
  
storage-account-name: The name of the storage account.  
  
service-credential: A string that the application uses to prove its identity.  
  
#2. Register the service principal, granting the correct role assignment, such as Storage Blob Data Contributor, on the Azure Data Lake Storage Gen2 account.  
  
#Step3: Mount Azure Data Lake Storage Gen2 filesystem like the below.



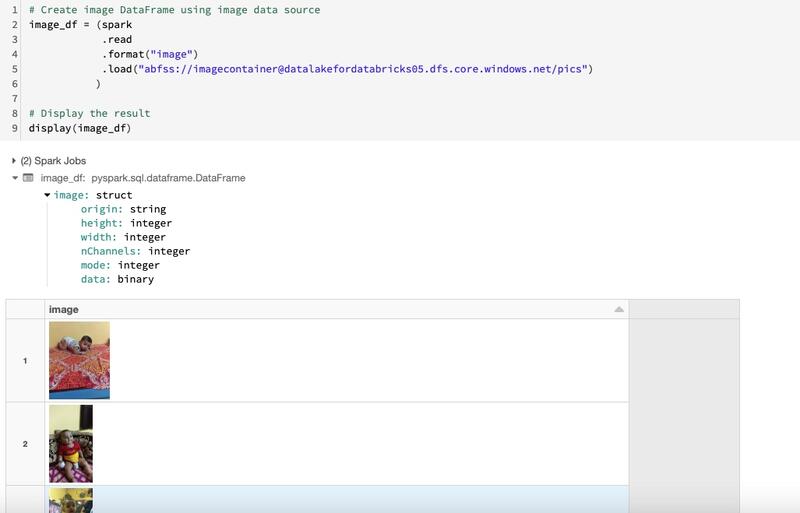
#Question213: How many different Cluster mode Azure Databricks offer? What is the difference between these modes ?  
  
  
Azure Databricks supports three cluster modes: Standard, High Concurrency, and Single Node.  
  
#Note: The default cluster mode is Standard.  
  
1. Standard clusters -  
  
Standard clusters are recommended for a single user. Standard clusters can run workloads developed in any language: Python, R, Scala, and SQL.  
  
2. High Concurrency clusters -  
  
A High Concurrency cluster is a managed cloud resource.  
  
The key benefits of High Concurrency clusters are that they provide Apache Spark-native fine-grained sharing for maximum resource utilization and minimum query latencies.  
  
High Concurrency clusters work only for SQL, Python, and R.  
  
The performance and security of High Concurrency clusters is provided by running user code in separate processes, which is not possible in Scala.  
  
3. Single Node clusters -  
  
A Single Node cluster has no workers and runs Spark jobs on the driver node. In contrast, Standard mode clusters require at least one Spark worker node.  
  
#Note: You cannot change the cluster mode after a cluster is created. If you want a different cluster mode, you must create a new cluster.



#Question214: What is the POOLs in Databricks ?  
  
Databricks pools reduce cluster start and auto-scaling times by maintaining a set of idle, ready-to-use instances.  
  
When a cluster attached to a pool needs an instance, it first attempts to allocate one of the pool’s idle instances.  
  
If the pool has no idle instances, the pool expands by allocating a new instance from the instance provider in order to accommodate the cluster’s request.  
  
When a cluster releases an instance, it returns to the pool and is free for another cluster to use.  
  
Only clusters attached to a pool can use that pool’s idle instances.  
  
Databricks does not charge DBUs while instances are idle in the pool  
  
Attach a cluster to a pool -  
  
To attach a cluster to a pool, select the pool in the Pool drop-down.



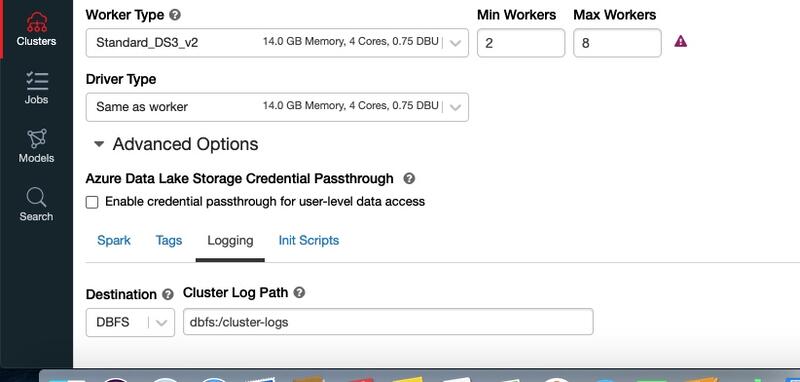
#Question215: How to read IMAGE data in Spark?  
  
Step1: Upload the images in the Cloud data source ( S3 / Blob )  
  
Step2: Use format = "image" & create Dataframe  
  
Step3: Display the results  
  
Generally this data will be fed to ML / DL pipelines for further analysis i.e suspicious behaviour in the CCTV footages, Traffic violations etc  
  
In the below example I have used my brother's son pics & uploaded them into Azure Data Lake Storage.



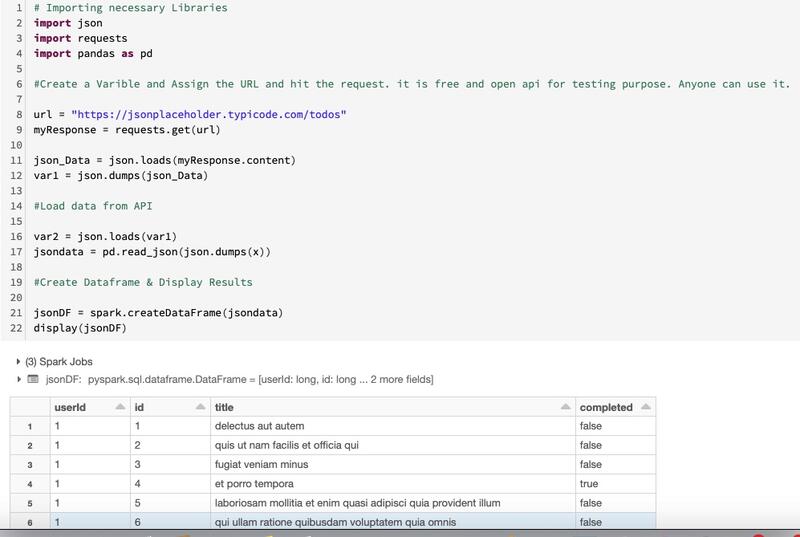
#Question216: What is DATABRICKS LIGHT ?  
  
Databricks Light is the Databricks packaging of the open source Apache Spark runtime.  
  
It provides a runtime option for jobs that don’t need the advanced performance, reliability, or autoscaling benefits provided by Databricks Runtime.  
  
In particular, Databricks Light does not support:  
  
1. Delta Lake  
2. Autopilot features such as autoscaling  
3. Highly concurrent, all-purpose clusters  
4. Notebooks, dashboards, and collaboration features  
5. Connectors to various data sources and BI tools  
  
Databricks Light is a runtime environment for jobs.  
  
When you run jobs on Databricks Light clusters, they are subject to lower Jobs Light Compute pricing.  
  
You can select Databricks Light only when you create or schedule a JAR, Python, or spark-submit job and attach a cluster to that job; you cannot use Databricks Light to run notebook jobs or interactive workloads.  
  
Databricks Light can be used in the same workspace with clusters running on other Databricks runtimes and pricing tiers. You don’t need to request a separate workspace to get started.  
  
  
#Note: When you create a job cluster, select a Databricks Light version from the Databricks Runtime Version drop-down.



#Question217: How to configure the Databricks Cluster Log for storing them in S3 / Azure Blob etc ?  
  
When We create a cluster, We can specify a location to deliver Spark driver, worker, and event logs.  
  
Logs are delivered every five minutes to our chosen destination. When a cluster is terminated, Databricks guarantees to deliver all logs generated up until the cluster was terminated.  
  
The destination of the logs depends on the cluster ID. If the specified destination is dbfs:/cluster-log-delivery, cluster logs for 0630-191345-leap375 are delivered to dbfs:/cluster-log-delivery/0630-191345-leap375.  
  
To configure the log delivery location:  
  
1. On the cluster configuration page, click the Advanced Options toggle.  
  
2. At the bottom of the page, click the Logging tab.  
  
3. Select a destination type.  
  
4. Enter the cluster log path.



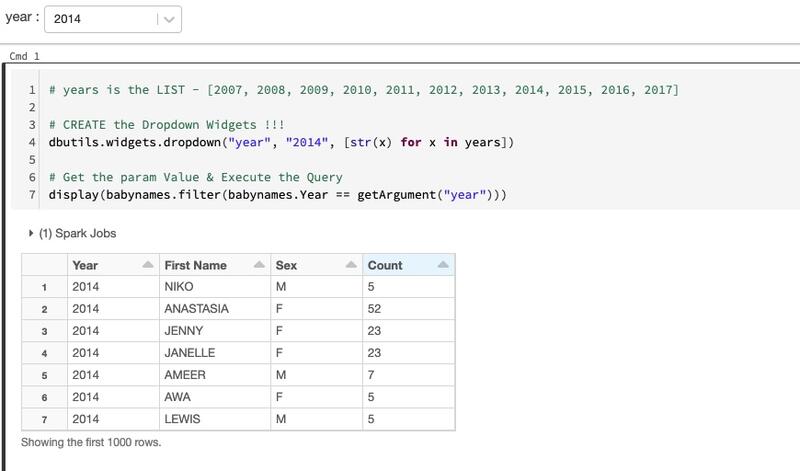
#Question218: How to call web API in Spark ( Databricks )?  
  
Step1: Import the necessary libraries  
  
Step2: Create variable and assign the url and hit the request.  
  
Step3: Read the Json from Pandas  
  
Step4: Create the Dataframe & Display the results.



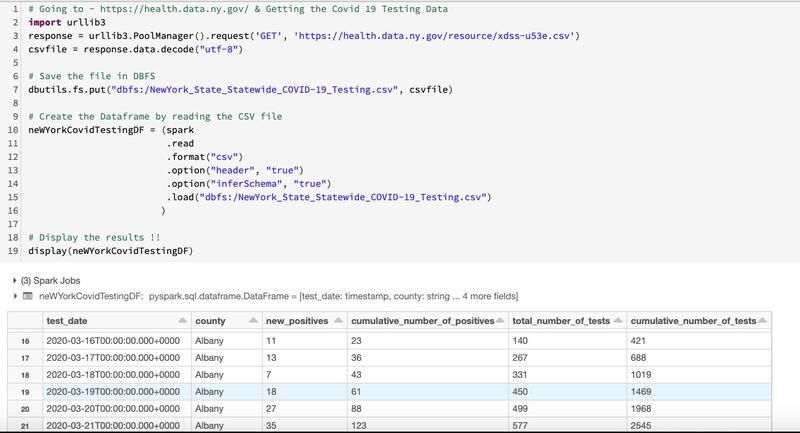
#Question219: What is SNOWFLAKE Cloud Datawarehouse ?  
  
Here is the detailed article, I have written on SNOWFLAKE. Please go through it & feel free to have any questions in the comment section.  
  
Below is the link of the article.  
  
https://lnkd.in/ggzvJBk

[](https://www.linkedin.com/pulse/5-reasons-why-snowflake-5-star-cloud-data-warehouse-deepak-rajak)

#Question220: How to CREATE Dropdown parameters in Databricks ?  
  
Step1: Use dbutils.widgets.dropdown to create the parameter.  
  
Step2: Provide a LIST for dropdown values.  
  
Step3: Use getArgument to read the parameter value.  
  
Step4: If we need to remove the widget then we can use - dbutils.widgets.remove



#Question221: How to Scrape a website & download the file in Spark ( Pyspark ) ?  
  
Step1: Going to - "[https://health.data.ny.gov](https://health.data.ny.gov/)" & Getting the Covid 19 Testing Data. ( In CSV format )  
  
Step2: Save the file in DBFS .  
  
Step3: Create the Dataframe by reading the CSV file.  
  
Step4: Display the results.



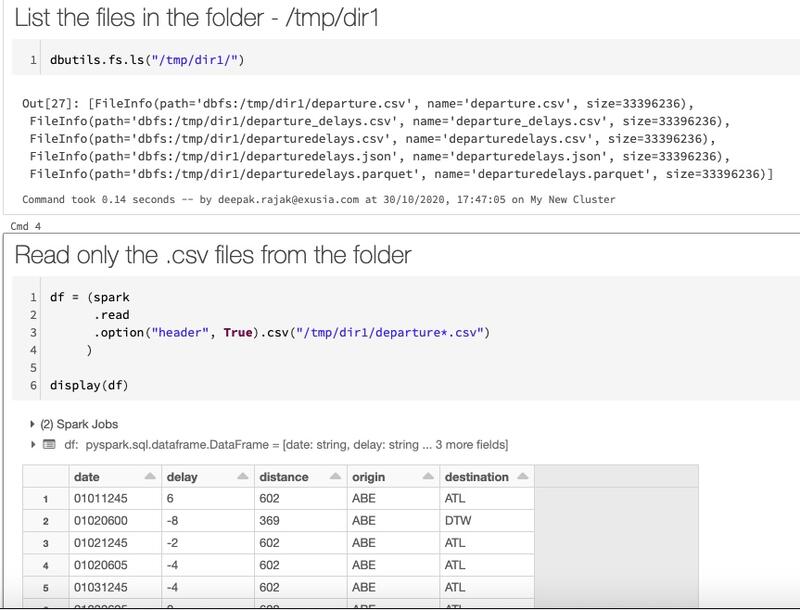
#Question222: What is the Cluster node initialization scripts ( init ) in Spark / Databricks ?  
  
An init script is a shell script that runs during startup of each cluster node before the Apache Spark driver or worker JVM starts.  
Some examples of tasks performed by init scripts include:  
  
1. Install packages and libraries not included in Databricks Runtime.  
2. Modify the JVM system classpath in special cases.  
3. Set system properties and environment variables used by the JVM.  
4. Modify Spark configuration parameters.  
  
Init script types:  
  
Databricks supports two kinds of init scripts: cluster-scoped and global.  
  
1. Cluster-scoped: run on every cluster configured with the script. This is the recommended way to run an init script.  
2. Global (Public Preview): run on every cluster in the workspace. They can help you to enforce consistent cluster configurations across your workspace.  
  
We can configure a Cluster Scopred init scripts like the below.  
Advanced Options -> Init Scripts -> Destination -> Specify Path -> ADD



#Question223: What is Task preemption in Spark / Databricks ? & How to set up different options. ?  
  
The Apache Spark scheduler in Databricks automatically preempts tasks to enforce fair sharing. This guarantees interactive response times on clusters with many concurrently running jobs.  
  
#Note -  
When tasks are preempted by the scheduler, their kill reason will be set to preempted by scheduler. This reason is visible in the Spark UI and can be used to debug preemption behavior.  
  
Preemption options:  
  
By default, preemption is conservative: jobs can be starved of resources for up to 30 seconds before the scheduler intervenes.  
  
Spark configuration properties at cluster launch time:  
  
1. Whether preemption should be enabled.  
  
spark.databricks.preemption.enabled true  
  
2. The fair share fraction to guarantee per job.The default setting is 0.5, which means at worst a jobs will get half of its fair share.  
  
spark.databricks.preemption.threshold 0.5  
  
3. How long a job must remain starved before preemption kicks in. Recommended values are from 1-100 seconds.  
  
spark.databricks.preemption.timeout 30s  
  
4. How often the scheduler will check for task preemption.  
  
spark.databricks.preemption.interval 5s



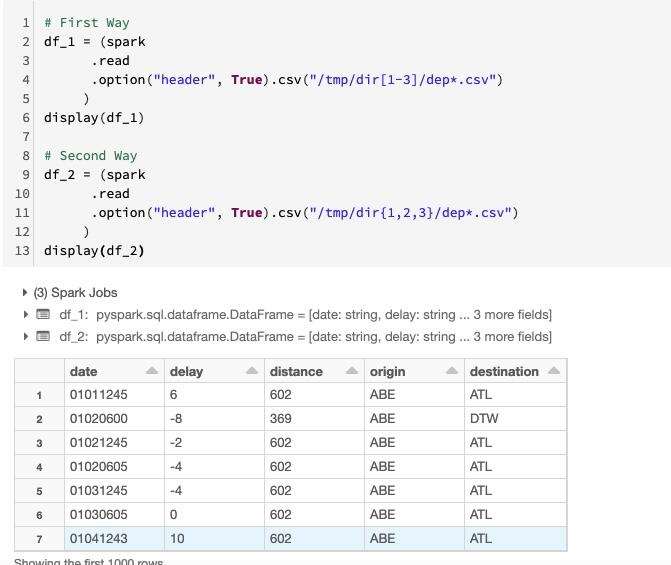
#Question224: How to read all the files from a folder & create the Dataframe in Spark ?  
  
Step1: We have 5 files in the folder - /tmp/dir1  
  
Step2: We want to read only CSV files - in this case 3 csv files at one go. ( leaving the json & parquet )  
  
Step3: We will use the WILD card (\*) & create the Dataframe  
  
Step4: Display the results.



#Question225: How to read all the files from multiple folders & create the Dataframe in Spark ?  
  
Step1: We have 1 file in each folder - /tmp/dir1, /tmp/dir2, /tmp/dir3  
  
Step2: We want to read all the CSV files - in this case 3 csv files at one go from different folders  
  
Step3: We will pass all the paths as list in our .csv / .load method & create the Dataframe  
  
Step4: Display the results.



#Question226: How to read all the files from multiple folders & create the Dataframe in Spark ?  
  
Step1: We have 1 file in each folder - /tmp/dir1, /tmp/dir2, /tmp/dir3  
  
Step2: We want to read all the CSV files - in this case 3 csv files at one go from different folders  
  
Step3: We will can use 2 other way ( what we have seen in Q225 ) by utilising the braces and brackets in the below manner and pass that to our .csv / .load method & create the Dataframe  
  
Step4: Display the results.



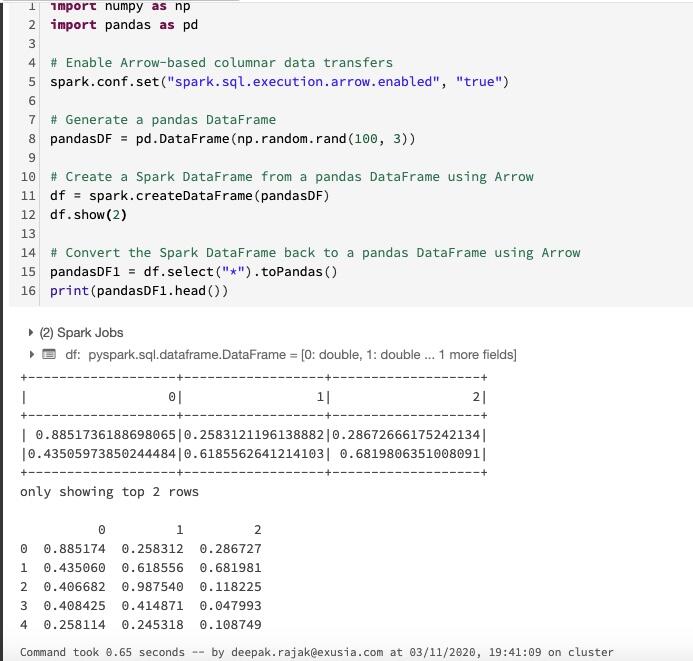
#Question227: How to read all the files recursively inside a folder & create the Dataframe in Spark ?  
  
Step1: We have 1 file in each level. inside /tmp we have one ".csv" file, inside /tmp/Level1 - we have another ".csv" file, inside /tmp/Level1/Level2 - we have another ".csv" file  
  
Step2: We want to read all the CSV files - in this case 3 csv files at one go from different folders  
  
Step3: We will use the option - recursiveFileLookup & create the Dataframe  
  
Step4: Display the results.  
  
Note: This option is available from Spark 3.0



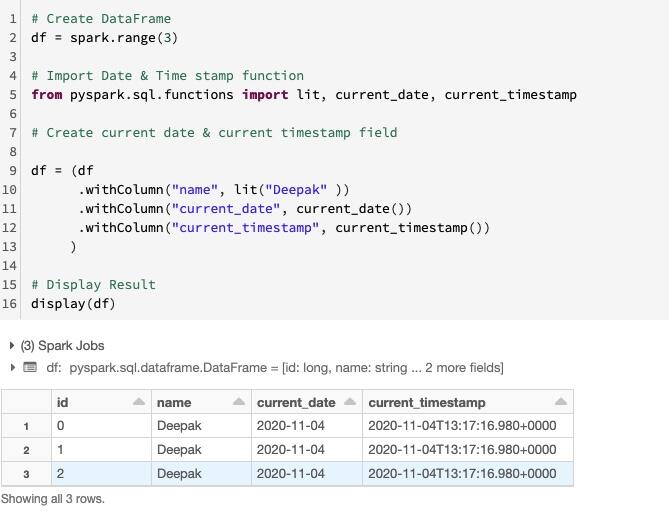
#Question228: How to read all the files from selected folders , subfolders & create the Dataframe in Spark ?  
  
Step1: We have 3 folders - SFFolder / NYFolder / MAFolder each folder contains 3 more folders - data2017 / data2018 / data2019 . Now each of these folders contains some .csv files  
  
Step2: We want to read all the CSV files - one go from two different folders SFFolder ( data2017 , data2018 ) & MAFolder ( data2017 , data2018 )  
  
Step3: We will use the - [] bracets to prune the folders & create the Dataframe  
  
Step4: Display the results.



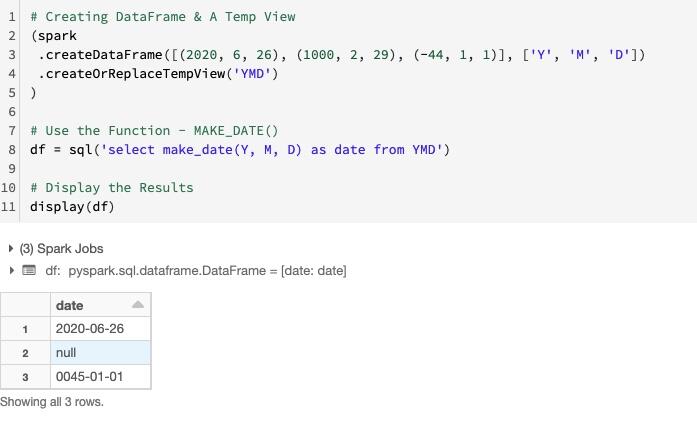
#Question229: How to Optimize conversion between PySpark and pandas DataFrames by using Apache Arrow ?  
  
Apache Arrow is an in-memory columnar data format used in Apache Spark to efficiently transfer data between JVM and Python processes.  
  
This is beneficial to Python developers that work with pandas and NumPy data.  
  
However, its usage is not automatic and requires some minor changes to configuration or code to take full advantage and ensure compatibility.  
  
Step1: Arrow is available as an optimization when converting a PySpark DataFrame to a pandas DataFrame with toPandas() and when creating a PySpark DataFrame from a pandas DataFrame with createDataFrame(pandas\_df).  
  
Step2: To use Arrow for these methods, set the Spark configuration spark.sql.execution.arrow.enabled to true. This configuration is disabled by default.  
  
Step3: optimizations enabled by spark.sql.execution.arrow.enabled could fall back to a non-Arrow implementation if an error occurs before the computation within Spark. You can control this behavior using the Spark configuration spark.sql.execution.arrow.fallback.enabled.



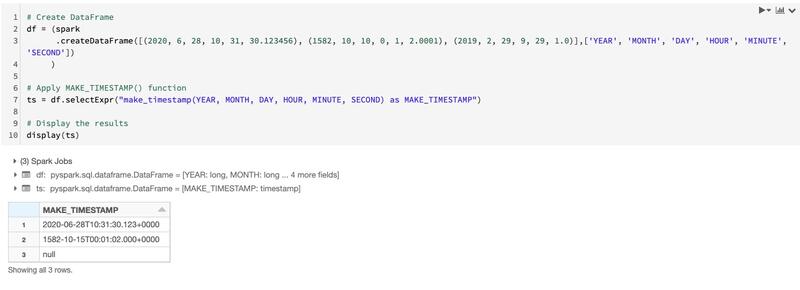
#Question230: How to populate CURRENT DATE & CURRENT TIMESTAMP in Spark (PySpark )?  
  
By Using the method CURRENT\_DATE() & CURRENT\_TIMESTAMP() respectively.



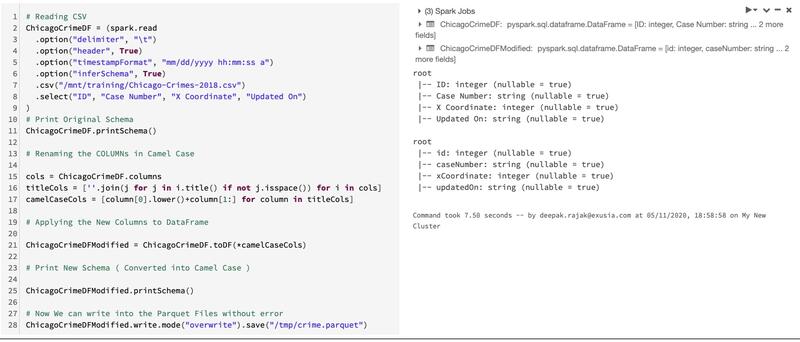
#Question231: How to Construct dates by MAKE\_DATE() function in Spark (PySpark )?  
  
Step1: The function MAKE\_DATE introduced in Spark 3.0 takes three parameters: YEAR, MONTH of the year, and DAY in the month and makes a DATE value.  
  
Step2: All input parameters are implicitly converted to the INT type whenever possible.  
  
Step3: The function checks that the resulting dates are valid dates in the Proleptic Gregorian calendar, otherwise it returns NULL.



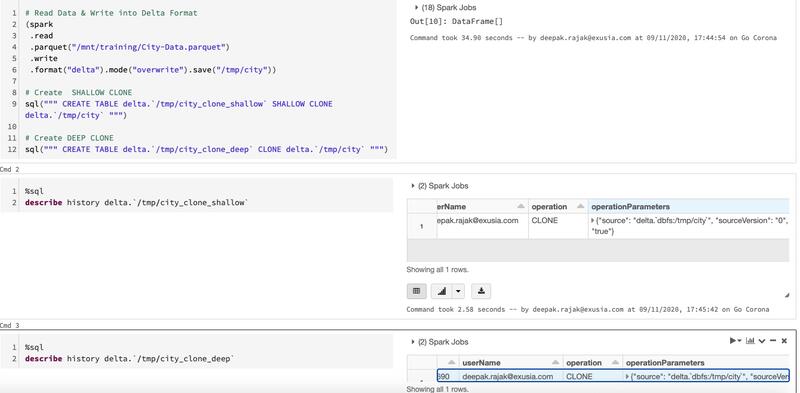
#Question232: How to Construct timestamp by MAKE\_TIMESTAMP() function in Spark (PySpark )?  
  
Step1: We can make timestamp values via the MAKE\_TIMESTAMP functions. Like MAKE\_DATE, it performs the same validation for date fields, and additionally accepts time fields HOUR (0-23), MINUTE (0-59) and SECOND (0-60).  
  
Step2: If we don't provide any time zone, Spark takes a time zone from the SQL configuration spark.sql.session.timeZone and applies it to function invocations. We can also pick a different time zone by passing it as the last parameter of MAKE\_TIMESTAMP. .  
  
Step3: In the below example, Spark cannot create the last timestamp because this date is not valid: 2019 is not a leap year.



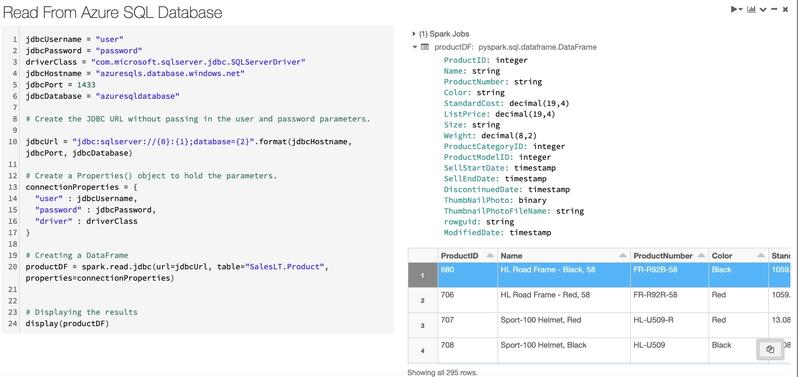
#Question233: How to Handle Column name with spaces in Spark ( Pyspark )?  
  
Step1: If we have columns names like - "Case Number", "X Coordinate", "Updated On". We will get error while writing this dataframe in the file.  
  
Step2: We need to convert these columns in suitable format so that we can write into the file / table. Basically we need to remove the spaces. Either we can add "\_" or convert these columns into camelCase which is recommended way in PySpark  
  
Step3: We will use the simple python string join and for loop to get this done.  
  
Note: This is one of the way, We can achieve this functionality in a number of different ways. This problem can be tackled in different ways and this is one of the way



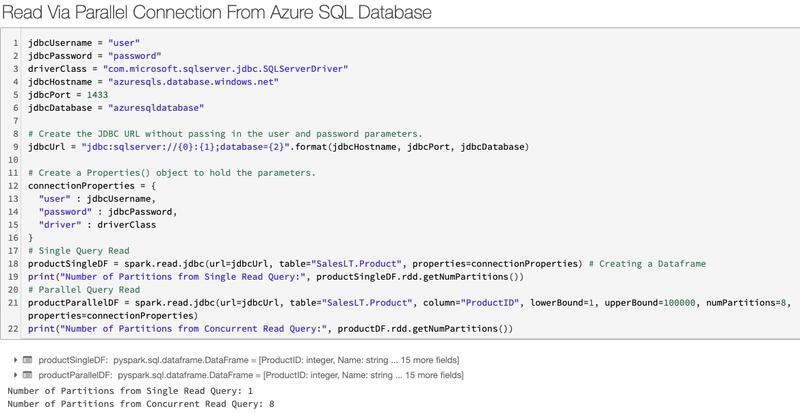
#Question234: What are CLONES in Databricks Delta Lake Table ?  
  
Clones are replicas of a source table at a given point in time.  
  
They have the same metadata as the source table: same schema, constraints, column descriptions, statistics, and partitioning.  
  
However, they behave as a separate table with a separate lineage or history.  
  
Any changes made to clones only affect the clone and not the source. Any changes that happen to the source during or after the cloning process also do not get reflected in the clone due to Snapshot Isolation.  
  
In Databricks Delta Lake we have two types of clones: shallow or deep.  
  
Shallow Clones:  
  
A shallow (also known as Zero-Copy) clone only duplicates the metadata of the table being cloned; the data files of the table itself are not copied. This type of cloning does not create another physical copy of the data resulting in minimal storage costs.  
  
Deep Clones:  
  
Shallow clones are great for short-lived use cases, but some scenarios require a separate and independent copy of the table’s data. A deep clone makes a full copy of the metadata and data files of the table being cloned.  
  
We can identify the deep / swallow clone by the flag - "isShallow"



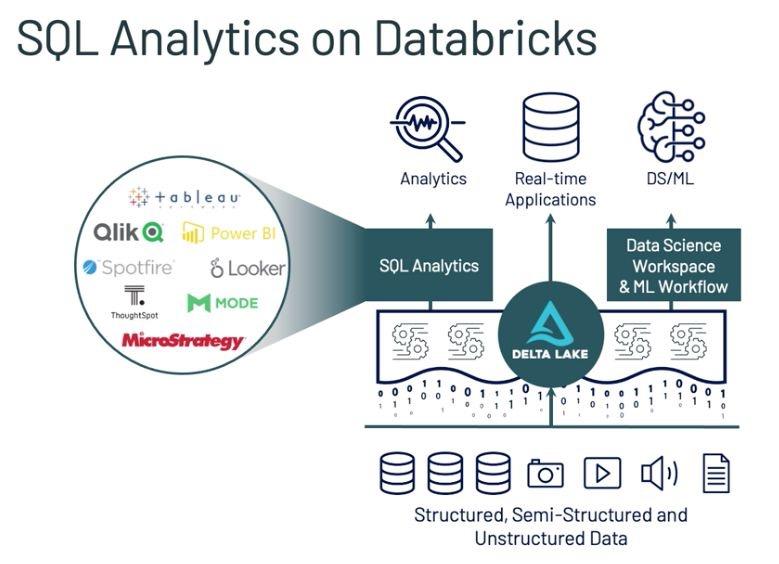
#Question235: How to Read from Azure SQL Database in Spark ( PySpark ) ?  
  
Step1: Setup the JDBC credentials.  
  
jdbcUsername = "user"  
jdbcPassword = "password"  
driverClass = "com.microsoft.sqlserver.jdbc.SQLServerDriver"  
jdbcHostname = " https://azuresqls.database.windows.net"  
jdbcPort = 1433  
jdbcDatabase = "azuresqldatabase"  
  
Step2: Setup the JDBC URL & connection properties dictionary .  
  
jdbcUrl = "jdbc:sqlserver://{0}:{1};database={2}".format(jdbcHostname, jdbcPort, jdbcDatabase)  
  
Step3: Create the DataFrame from the command - spark.read.jdbc  
  
#Note : Default Behaviour: Single Read Query from JDBC

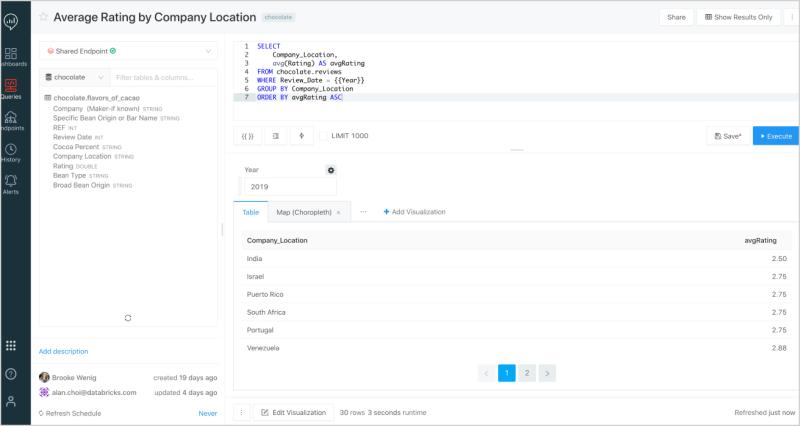


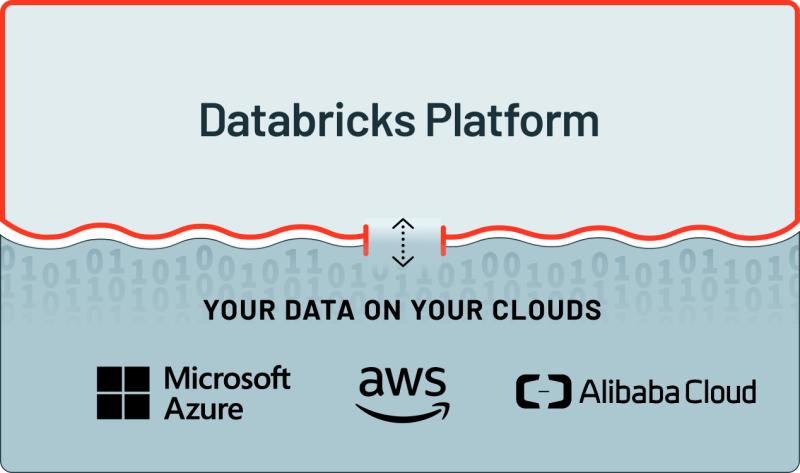
#Question236: How to Read Concurrently ( In Parallel ) from Azure SQL Database in Spark ( PySpark ) ?  
  
Step1: Setup the JDBC credentials.  
  
jdbcUsername = "user"  
jdbcPassword = "password"  
driverClass = "com.microsoft.sqlserver.jdbc.SQLServerDriver"  
jdbcHostname = " https://azuresqls.database.windows.net"  
jdbcPort = 1433  
jdbcDatabase = "azuresqldatabase"  
  
Step2: Setup the JDBC URL & connection properties dictionary .  
  
jdbcUrl = "jdbc:sqlserver://{0}:{1};database={2}".format(jdbcHostname, jdbcPort, jdbcDatabase)  
  
Step3: Create the DataFrame from the command - spark.read.jdbc  
  
Step4: Setup parameters for concurrent read  
  
column="ProductID" ( Column for partitioning )  
lowerBound=1 ( Min of column )  
upperBound=100000 ( Max of Column )  
numPartitions=8 ( Number of connections you wish to open )



#Question237: What is SQL ANALYTICS in Databricks ?  
  
SQL Analytics allows customers to operate a multi-cloud lakehouse architecture that provides data warehousing performance at data lake economics for up to 9x better price/performance for SQL workloads than traditional cloud data warehouses.  
  
Benefits:  
  
1. SQL-native Interface  
  
2. Easily Create Visualizations and Share Dashboards  
  
3. Up to 9x Better Price/Performance for BI  
  
4. Reliability and Governance for Data Lakes  
  
5. Multi-cloud Support



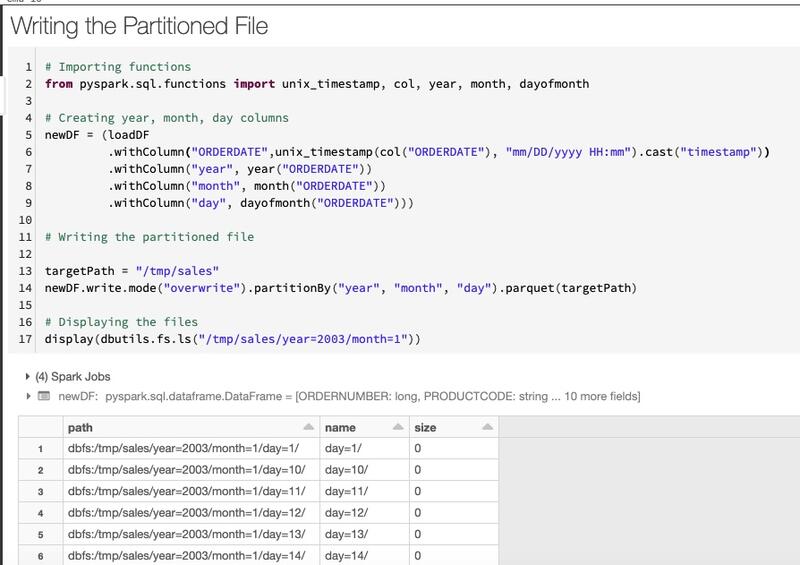




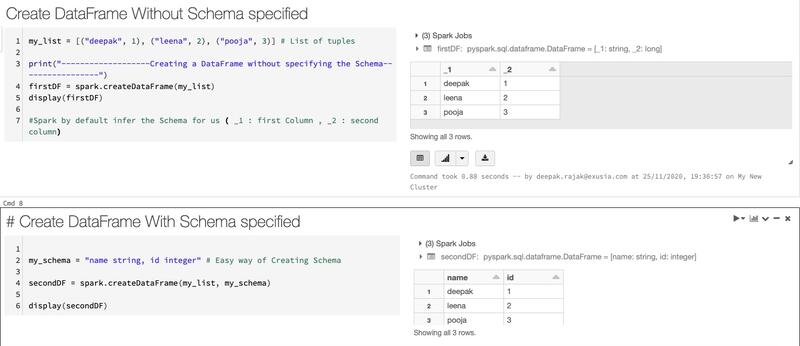
#Question238: What is Exclusion Mechanism is Spark? How Databricks has improved it ?  
  
The exclusion mechanism was introduced for task scheduling in Apache Spark 2.2.0 (as “blacklisting”). The motivation for having exclusion is to enhance fault tolerance in Spark, especially against the following problematic scenario:  
  
1. In a cluster with hundreds or thousands of nodes, there is a decent probability that executor failures happen on one of the nodes during a long-running Spark application and this can lead to task failure.  
  
2. When a task failure happens, there is a high probability that the scheduler will reschedule the task to the same node and same executor because of locality considerations. Now, the task will fail again.  
  
3. After failing .task.maxFailures number of times on the same task, the Spark job would be aborted.  
  
New features introduced in Databricks Runtime 7.3  
  
#Feature1: by adding a configuration called spark.databricks.blacklist.decommissionNode.enabled. If spark.databricks.blacklist.decommissionNode.enabled is set to true, when a node is excluded on the application level, it will be decommissioned, and a new node would be launched to keep the cluster to its desired size.  
  
#Feature2: By tuning spark.blacklist.application.blacklistedNodeThreshold



#Question239: How to WRITE Parquet files partitioned by YEAR, MONTH & DAY ?  
  
Step1: Covert the String date/timestamp into date or timestamp format.  
  
Step2: Create 3 Columns by using the - .withColumn method.  
  
Step3: Apply YEAR , MONTH, DAYOFMONTH function  
  
Step4: Write the file by using partitionBy method. Pass the comma separated columns as parameters.



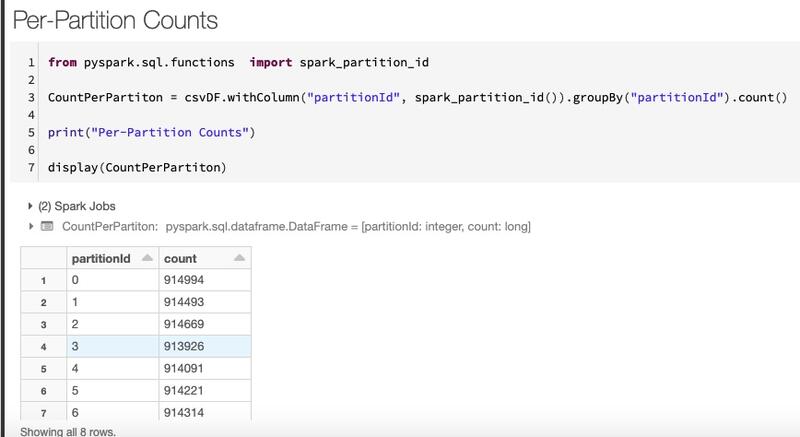
#Question240: How to Create DataFrame With & Without Schema specified in Spark ?  
  
Without Schema: Use the .createDataFrame method and pass the list. Spark will create the default schema  
  
With Schema: Use the .createDataFrame method and pass the list as the first parameter & schema as the second parameter.



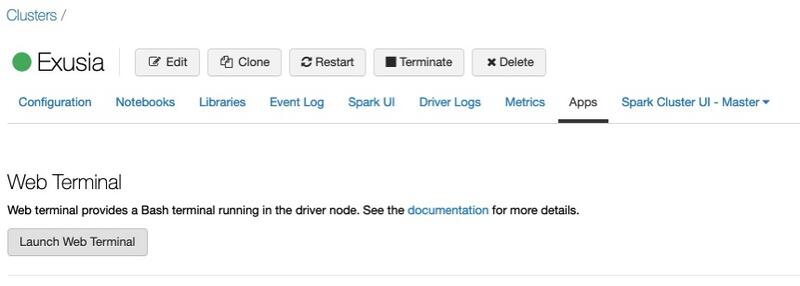
#Question241: What is the difference between spark.sql.shuffle.partitions and spark.default.parallelism?  
  
spark.sql.shuffle.partitions configures the number of partitions that are used when shuffling data for joins or aggregations. default value is 200.  
  
spark.default.parallelism is the default number of partitions in RDDs returned by transformations like join, reduceByKey, and parallelize when not set explicitly by the user.

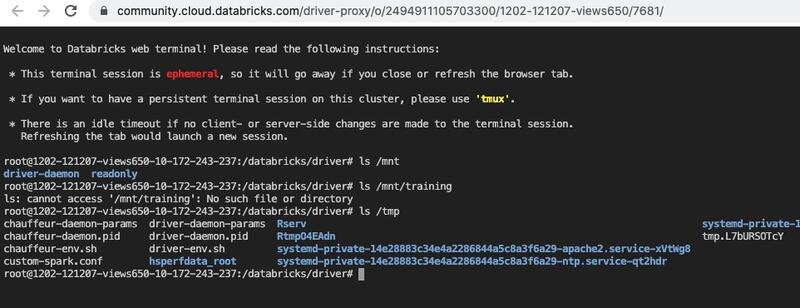


#Question242: How to get the COUNT of number of records in each Partition of a Dataframe in Spark?  
  
Step1: Import spark\_partition\_id  
  
Step2: Create a column name via .withColumn() method and assign spark\_partition\_id() as value  
  
Step3: Apply groupBy() and count() method to get the number of records in each partition.



#Question243: How to Launch the WEB TERMINAL in Databricks ?  
  
Databricks web terminal provides a convenient and highly interactive way for you to run shell commands and use editors, such as Vim or Emacs, on the Spark driver node.  
  
Web terminal is disabled by default for all workspace users.  
  
Requirements:  
  
1. Databricks Runtime 7.0 and above.  
2. Can Attach To permission on a cluster.  
3. Your Databricks workspace must have web terminal enabled.  
  
How to Launch -  
In a cluster detail page, click the Apps tab and then click Launch Web Terminal.





#Question244: How many Spark Jobs will be generated with the below code & Why ?  
  
Please answer in the comment section. I will posting the correct answer with the explanation after 24 hours.



Ans: 2 (Why ? Spark needs to touch the data physically 2 times. First to get the number of columns & second to scan the full csv file to get the data type. Both the jobs could be avoided by passing the User Defined Schema.)

#Question245: How many Spark Jobs will be generated with the below code & Why ?  
  
Please answer in the comment section. I will posting the correct answer with the explanation after 24 hours.

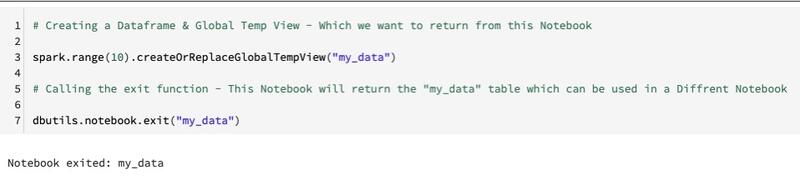


Ans: 0 (Why ? In this case Spark doesn’t need to touch the data physically in the storage layer Becasue it has all the relevant information to create the dataframe.)

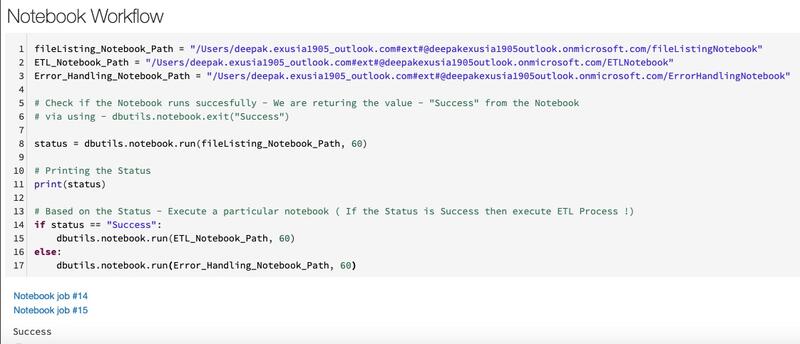
#Question246: How to Repartition by a Column name ? What is the default number of partitions will it create ?  
  
Step1: Use the method .repartition and pass the column name as parameter  
  
Step2: When partitioning by a column, Spark will create a minimum of 200 partitions by default.  
  
Step3: This example will have three partitions with data and 197 empty partitions.



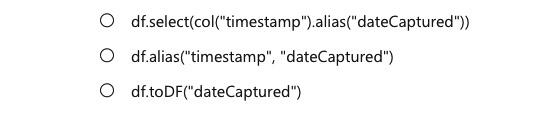
#Question247: How can we return values from a notebook in Databricks ?  
  
Step1: Create a table or a file which you want as return value.  
  
Step2: Use dbutils.notebook.exit() method to return the value. pass the table name or file name in the parameter.  
  
Step3: Now this notebook you can call from different Notebook & make use of these return values to easily build complex workflows and pipelines with dependencies.  
  
In the below example, We are returning table as return value. in the same way, We can return files or any other value.



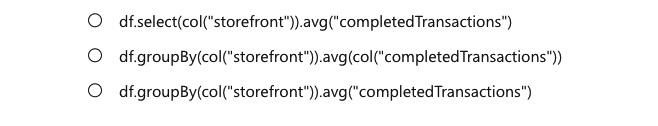
#Question248: How to create if/then/else workflows based on return values in Databricks ? How to manage Notebook Workflows.  
  
Step1: Create different Notebooks which you want to be in the End to End Data Pipeline i.e. file listing Notebook, ETL process & Error Handling  
  
Step2: Use dbutils.notebook.exit() method to return the value from the fileListingNotebook & pass the value in the parameter i.e. Success  
  
Step3: Now this notebook you can call from different Notebook & make use of these return values to easily build complex workflows and pipelines with dependencies.  
  
Step4: We can use a simple if / else logic to build dependencies. In the below example, We are running the ETL Notebook or Error Handling Notebook based on the Success status.



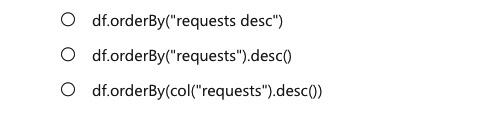
#Question249: Which method for renaming a DataFrame's column is incorrect ?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



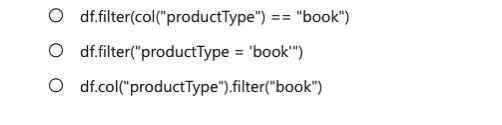
#Question250: We need to find the average of sales transactions by storefront. Which of the following aggregates would you use? ?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



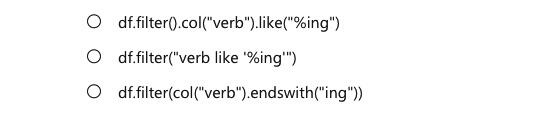
#Question251: Which command orders by a column in descending order?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



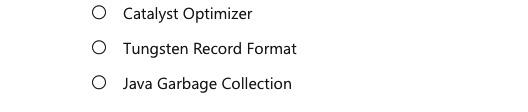
#Question252: Which command specifies a column value in a DataFrame's filter? Specifically, filter by a productType column where the value is equal to book?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



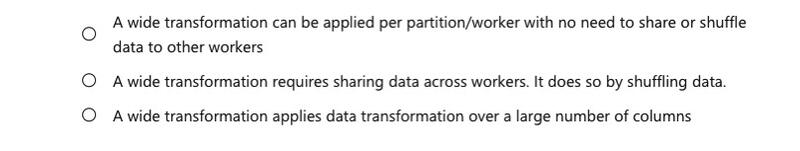
#Question253: When using the Column Class, which command filters based on the end of a column value? For example, a column named verb and filtered by words ending with "ing". ?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



#Question254: Which feature of Spark determines how your code is executed?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



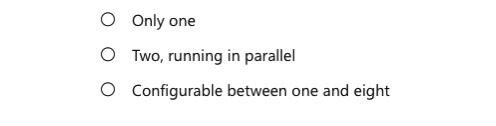
#Question255: Which of the following statements describes a wide transformations?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



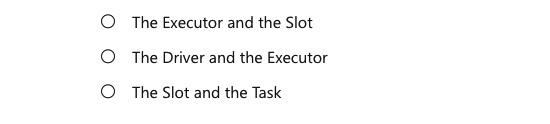
#Question256: If you create a DataFrame that will read some data from Azure Blob Storage, and then you create another DataFrame by filtering the initial DataFrame. What feature of Spark causes these transformation to be analyzed ?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



#Question257: How many drivers does a Cluster have ?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



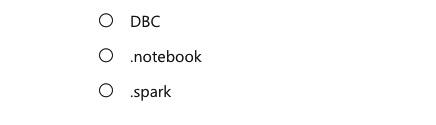
#Question258: Spark is a distributed computing environment. Therefore, work is parallelized across executors. At which two levels does this parallelization occur ?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



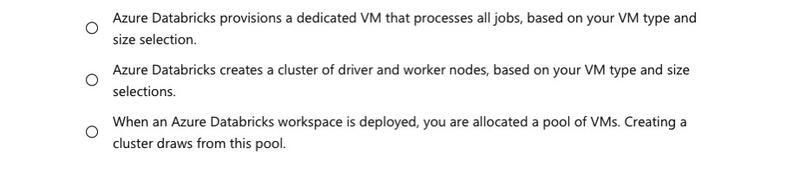
#Question259: What type of process are the driver and the executors?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



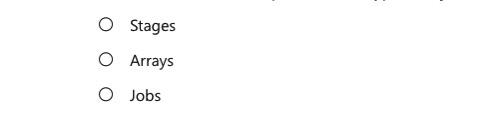
#Question260: Which notebook format is used in Databricks ?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



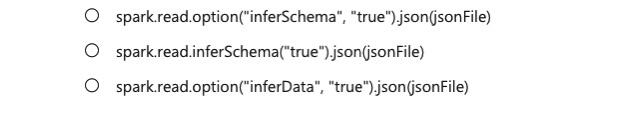
#Question261: When creating a new cluster in the Azure Databricks workspace, what happens behind the scenes ?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



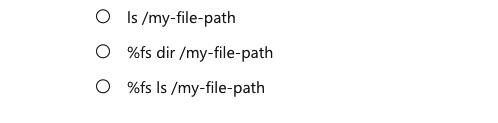
#Question262: To parallelize work, the unit of distribution is a Spark Cluster. Every Cluster has a Driver and one or more executors. Work submitted to the Cluster is split into what type of object ?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



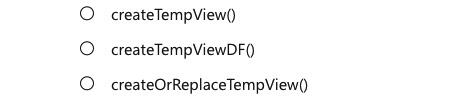
#Question263: How do you infer the data types and column names when you read a JSON file?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



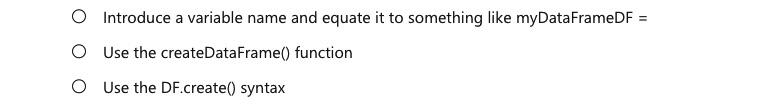
#Question264: How do you list files in DBFS within a notebook?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



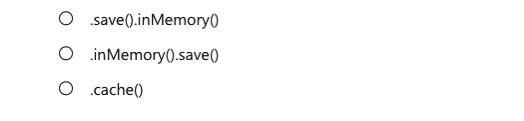
#Question265: Which DataFrame method do you use to create a temporary view?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



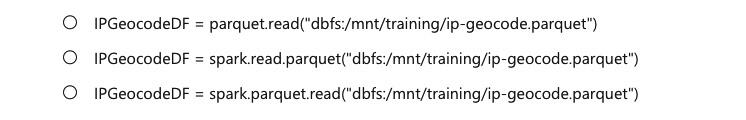
#Question266: How do you create a DataFrame object?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



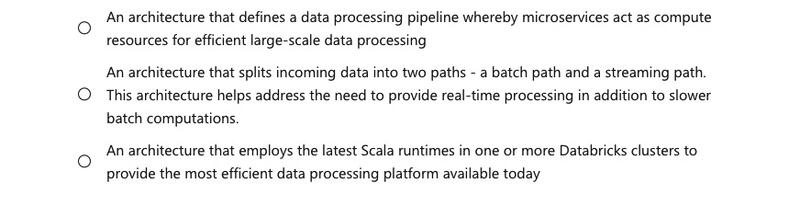
#Question267: How do you cache data into the memory of the local executor for instant access?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



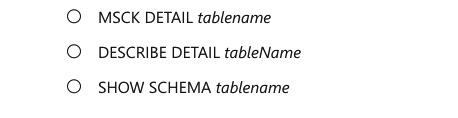
#Question268: What is the Python syntax for defining a DataFrame in Spark from an existing Parquet file in DBFS?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



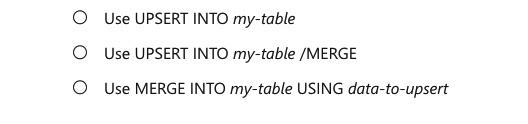
#Question269: What is a lambda architecture and what does it try to solve?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



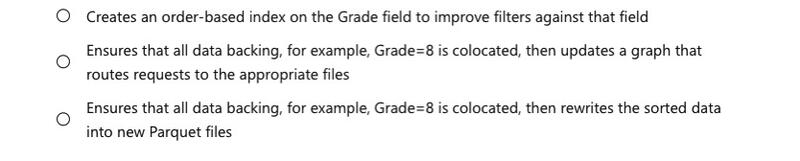
#Question270: What is the Databricks Delta command to display metadata?  
  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



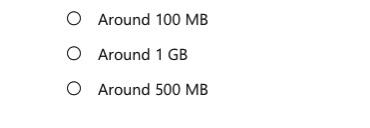
#Question271: How do you perform UPSERT in a Delta dataset?  
  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



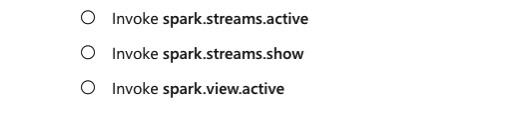
#Question272: What optimization does the following command perform: OPTIMIZE Students ZORDER BY Grade?  
  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



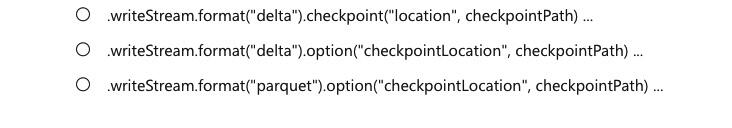
#Question273: What size does OPTIMIZE compact small files to?  
  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



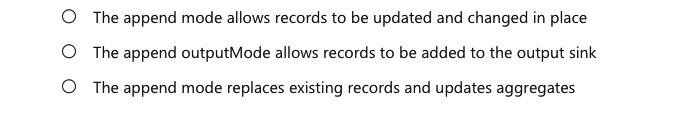
#Question274: What command should be issued to view the list of active streams?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



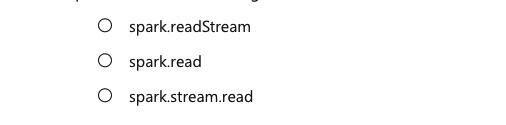
#Question275: What is required to specify the location of a checkpoint directory when defining a Delta Lake streaming query?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



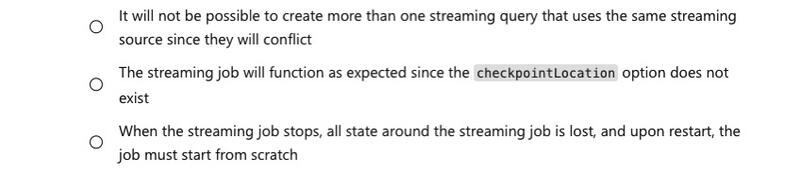
#Question276: When doing a write stream command, what does the outputMode("append") option do?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



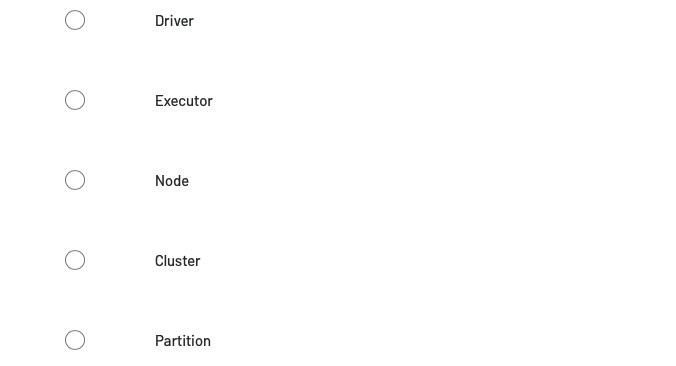
#Question277: In Spark Structured Streaming, what method should be used to read streaming data into a DataFrame?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



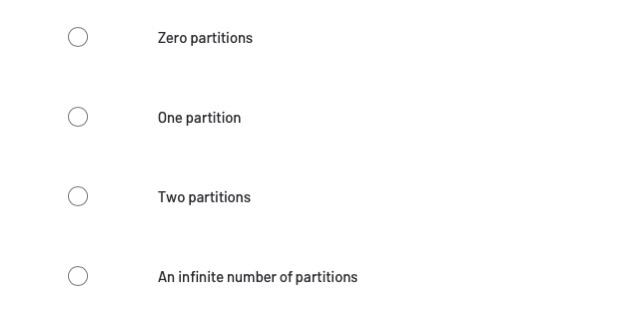
#Question278: What happens if the command option("checkpointLocation", pointer-to-checkpoint directory) is not specified?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



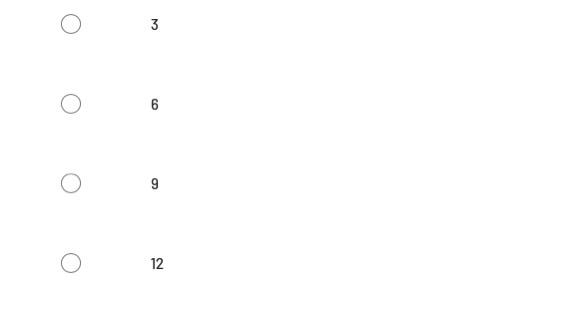
#Question279: Which part of the Spark architecture is responsible for deciding which task processes which piece of data?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



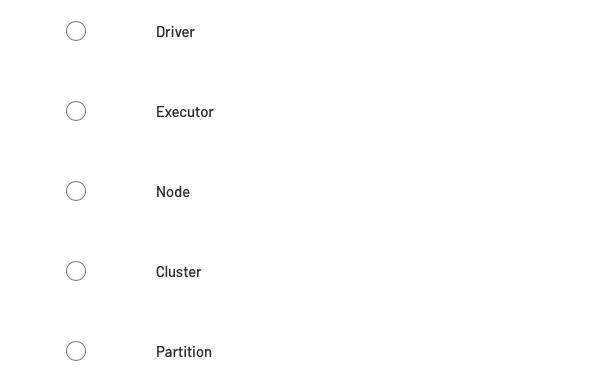
#Question280: Against how many partitions can a single task execute its instructions?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



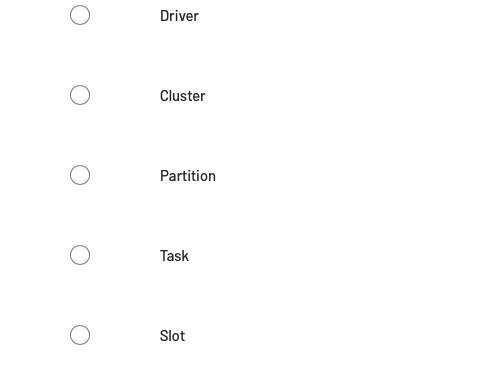
#Question281: If you have 3 executors and each executor has 3 slots, what is the maximum number of tasks that can be executed at any one time?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



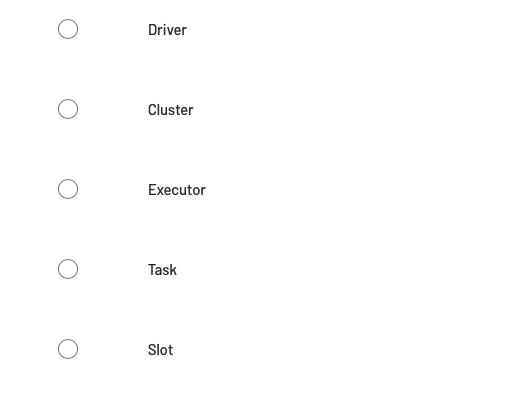
#Question282: Which part of the Spark architecture is responsible for deciding how to subdivide the larger dataset into at 128 MB chunks?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



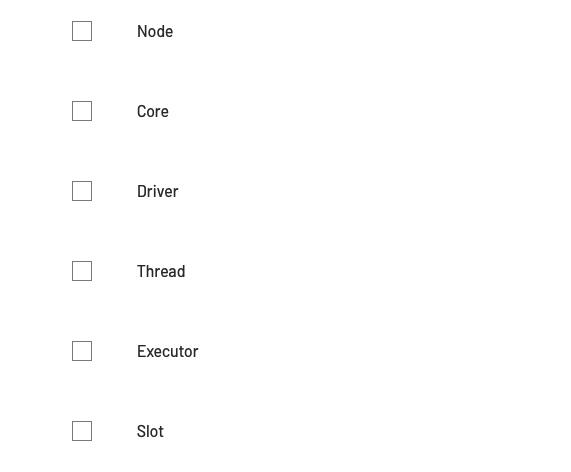
#Question283: What term identifies the smallest unit of work in a Spark application?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



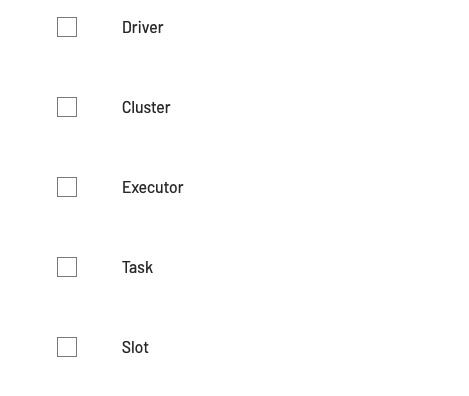
#Question284: Which term identifies the environment in which a task is executed?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



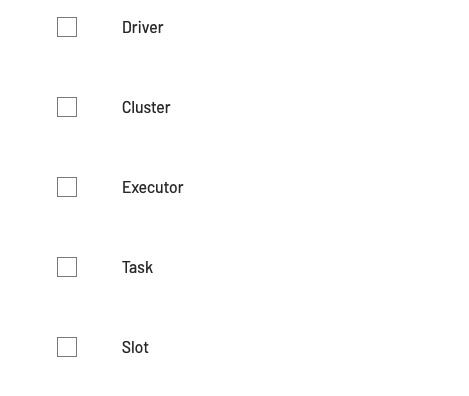
#Question285: While technically different, which three terms are generally considered to be synonymous?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



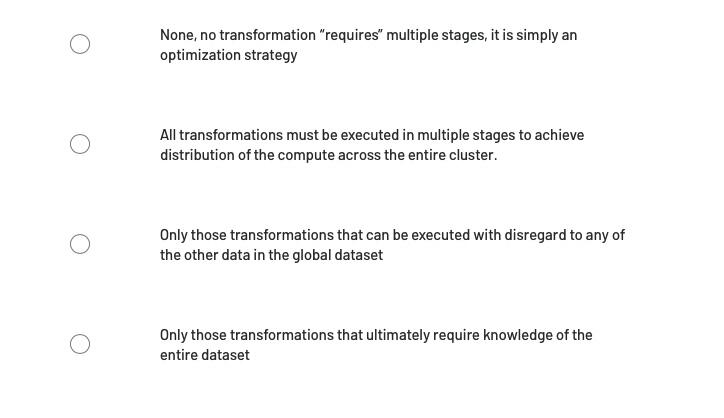
#Question286: Which two parts of the Spark architecture are run inside of a Java Virtual Machine (JVM)?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



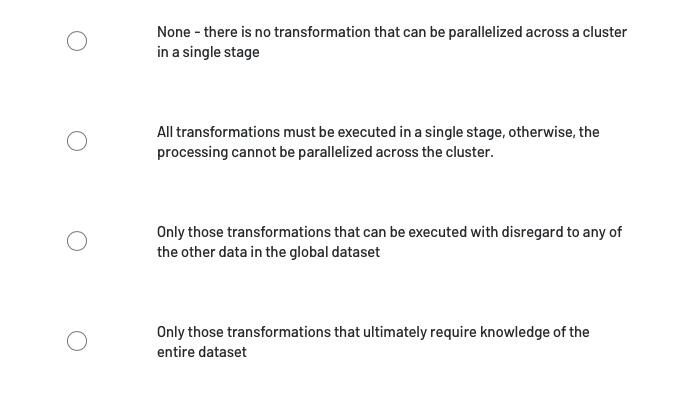
#Question287: Which two parts of the Spark architecture are run inside of a Java Virtual Machine (JVM)?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



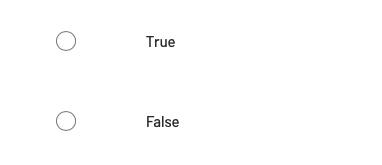
#Question288: Which transformations must Apache Spark execute with multiple stages?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



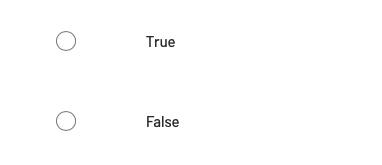
#Question289: Which transformations must Apache Spark execute in only one stage?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



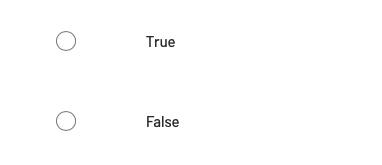
#Question290: Transformations that can be executed in a single stage are generally referred to as “wide” transformations ?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



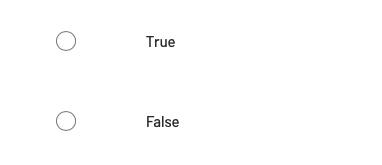
#Question291: Transformations that must be executed across multiple stages are generally referred to as “narrow” transformations ?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



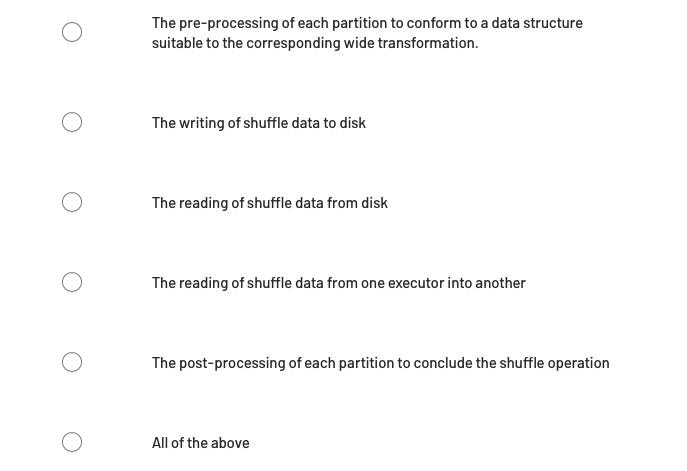
#Question292: In a job with multiple stages, subsequent stages can be started if the majority of the tasks in the previous stage have completed ?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



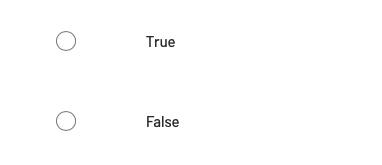
#Question294: In a job with multiple stages, the execution of the secondary stages can be delayed by a single, slow-running task in the previous stage ?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



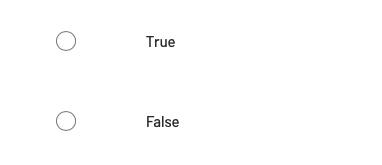
#Question293: Which key factors relate to the performance hit associated with wide transformations ?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



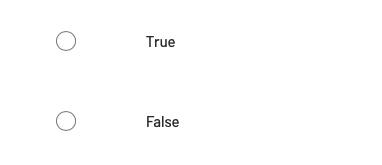
#Question295: Given the correct strategy, the shuffle operation for any wide transformations can be avoided ?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



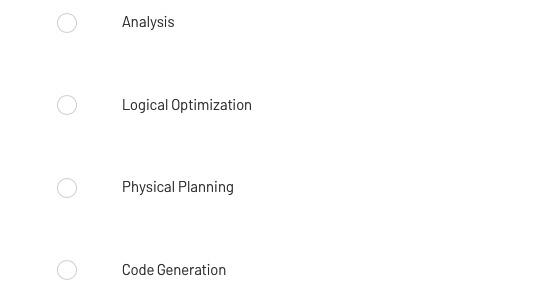
#Question296: Because of the performance cost associated with wide transformations, namely the cost of their respective shuffle operations, wide transformations should be avoided altogether. ?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



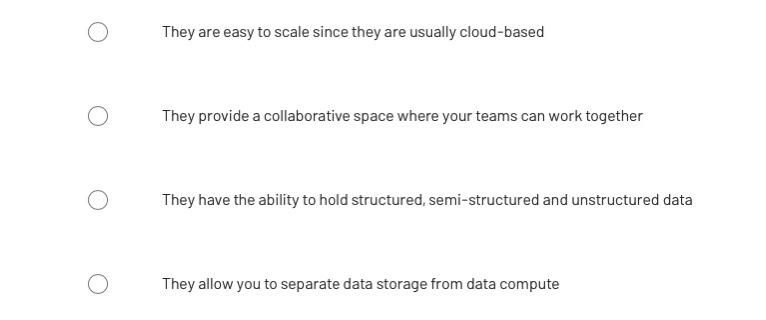
#Question297: One can assist the Catalyst Optimizer’s optimization processes by executing multiple wide transformations back-to-back.?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



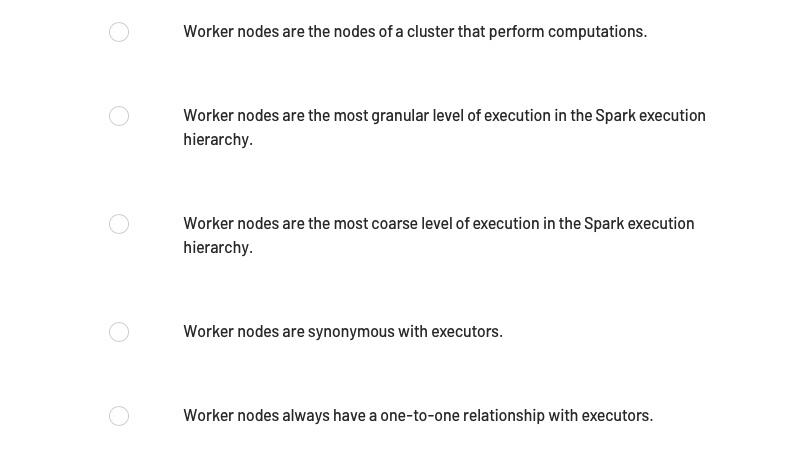
#Question298: At which stage do the first set of optimizations take place.?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



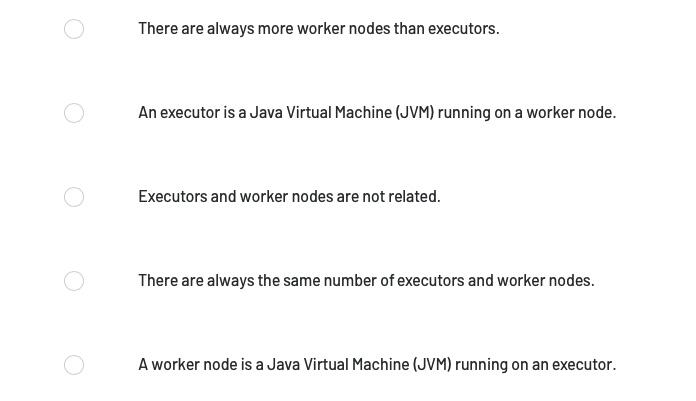
#Question299: Which of the following statements describes how unified cloud data platforms differ from data lakes?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



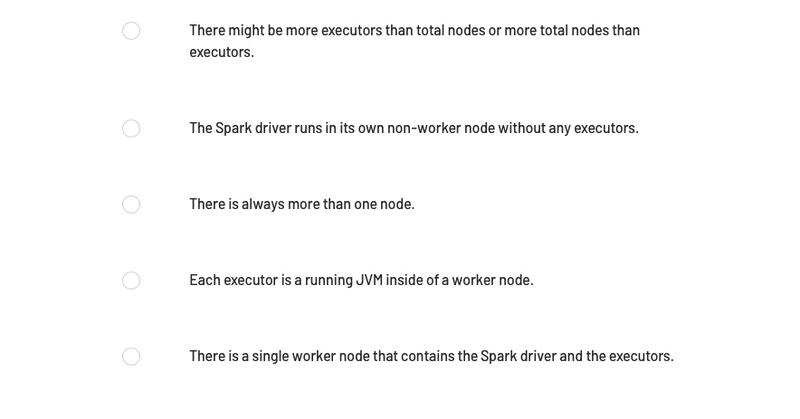
#Question300: Which of the following describes a worker node?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



#Question301: Which of the following describes the relationship between worker nodes and executors?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



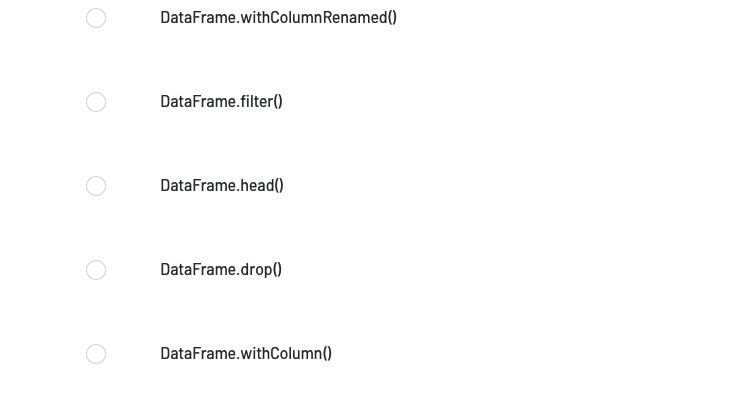
#Question302: If Spark is running in cluster mode, which of the following statements about nodes is incorrect?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



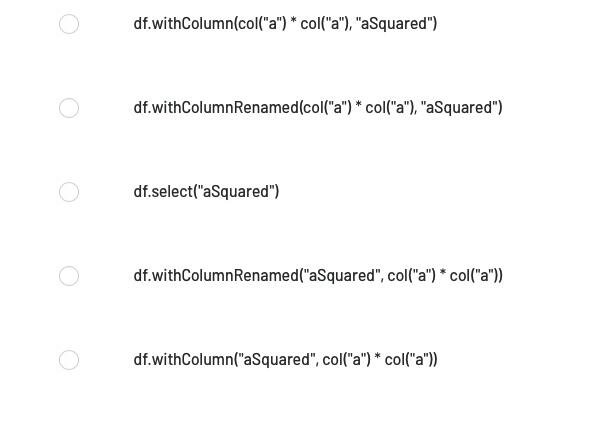
#Question303: Which of the following DataFrame operations is always classified as a narrow transformation?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



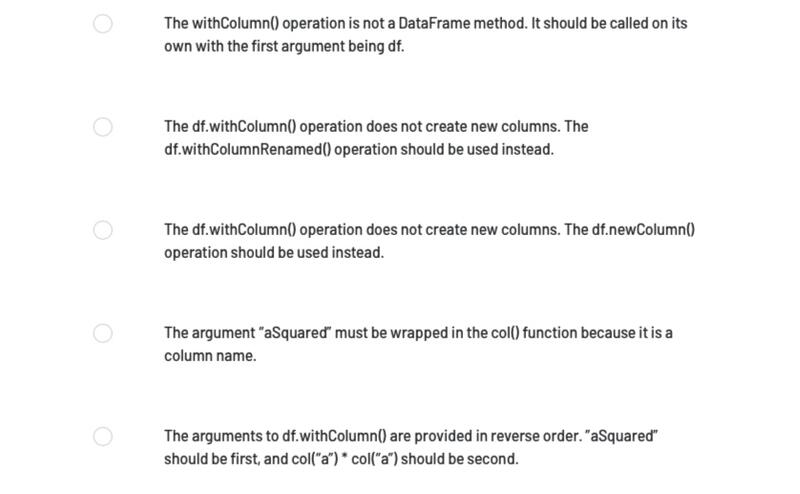
#Question304: Which of the following operations can be used to create a new DataFrame with a new column and all previously existing columns form an existing DataFrame?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



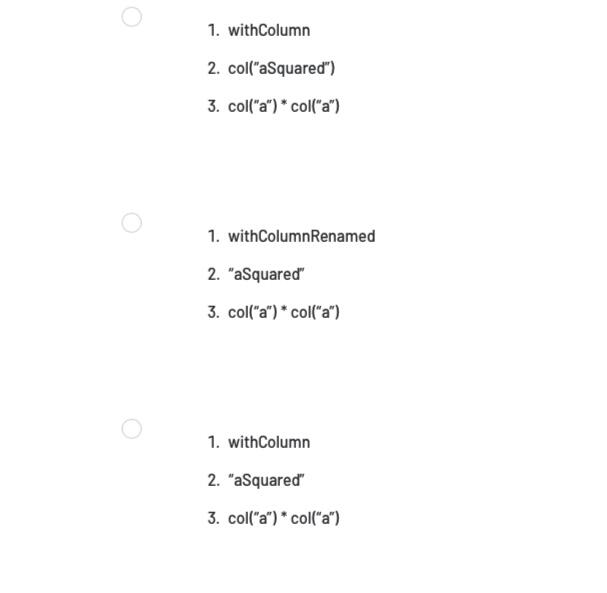
#Question306: Which of the following code blocks returns a DataFrame with a new column "aSquared" and all previously existing columns from DataFrame df?  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



#Question307: The code block shown below contains an error. The code block is intended to return a DataFrame with a new column aSquared and all previously existing columns from DataFrame df. Identify the error.  
  
Code block:  
  
df.withColumn(col(“a”) \* col(“a”), “aSquared”)  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



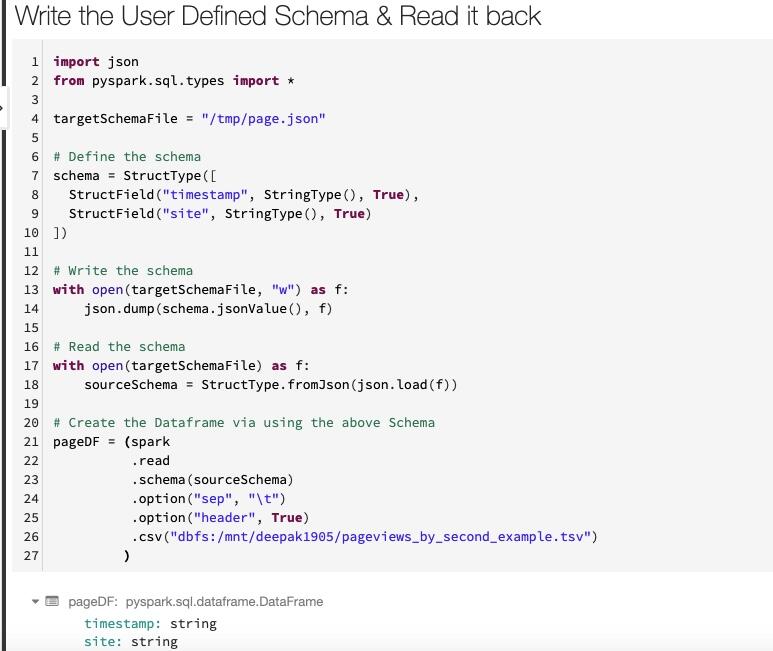
#Question308: The code block shown below should return a DataFrame with a new column aSquared and all previously existing columns from DataFrame df. Choose the response that correctly fills in the numbered blanks within the code block to complete this task.  
  
Code block:  
  
df.\_\_1\_\_(\_\_2\_\_, \_\_3\_\_)  
  
Please answer in the comment section. I will be posting the correct answer with the explanation after 24 hours.



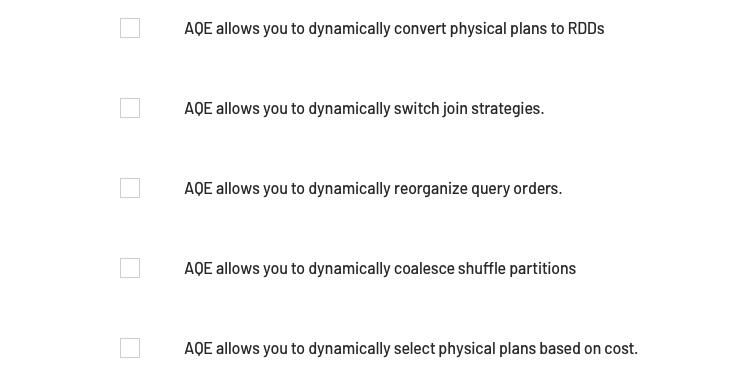
#Question310: In what order should the below lines of code be run in order to return a DataFrame with a new column aSquared and all previously existing columns from DataFrame df?  
  
  
1 df  
2 .withColumn(“aSquared”, “a” \* “a”)  
3 .withColumn(“aSquared”, col(“a”) \* col(“a”))  
4 DataFrame  
5 .withColumn(col(“aSquared”), col(“a”) \* col(“a”)



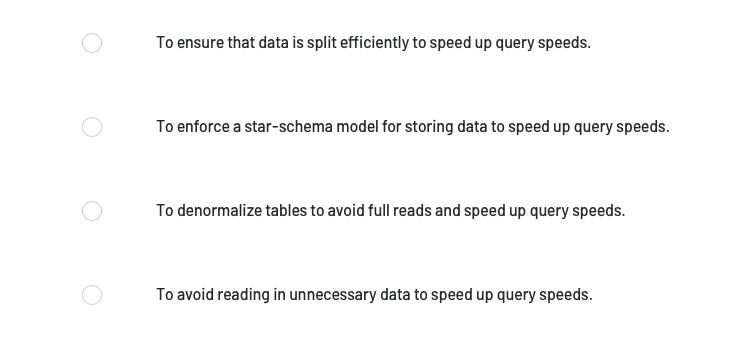
#Question311: How to Save User Defined Schema in Spark in a file & use it while creating a Dataframe ?  
  
Step1: Create a User Defined Schema using StructType() & StructField()  
  
Step2: Write the Schema in Json file using json.dump & json.value() function  
  
Step3: Read the Schema back using the fromJson() & json.load() function  
  
Step4: Create the Dataframe using .schema() method.  
  
#Note: We can generalise this process by creating the parameters to enforce the schema validation process.



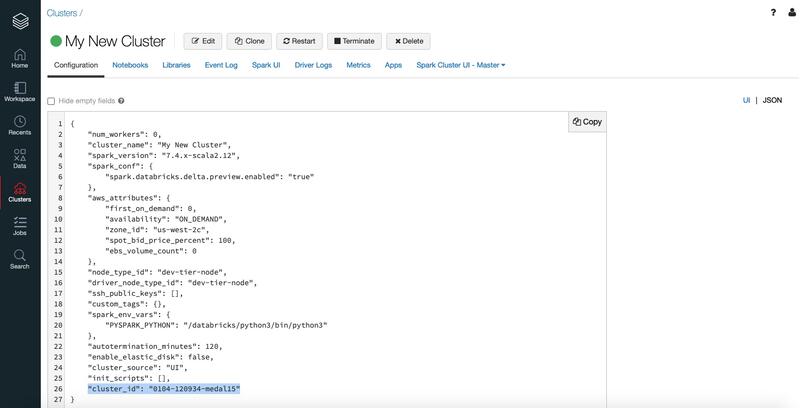
#Question312: Which of the following describe optimizations enabled by adaptive query execution (AQE)? Choose two.  
  
Please write your answer in the comment section.



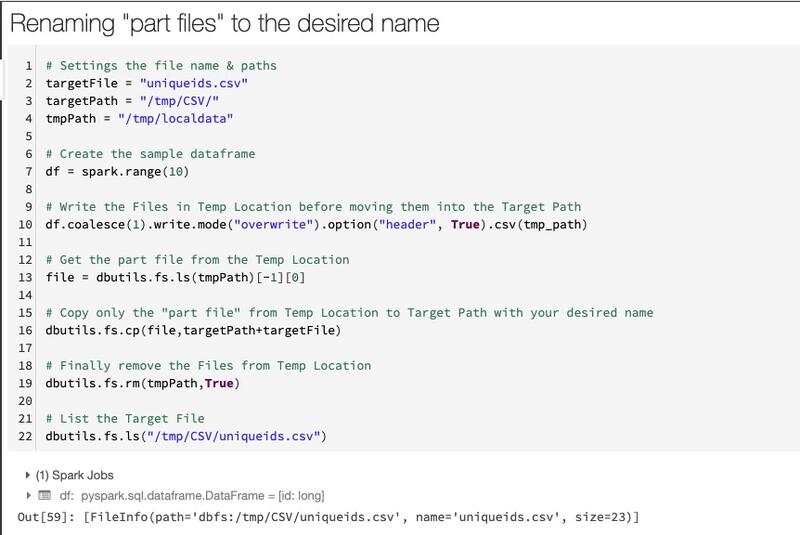
#Question313: What is the ultimate goal behind dynamic pruning partitioning?  
  
Please write your answer in the comment section.



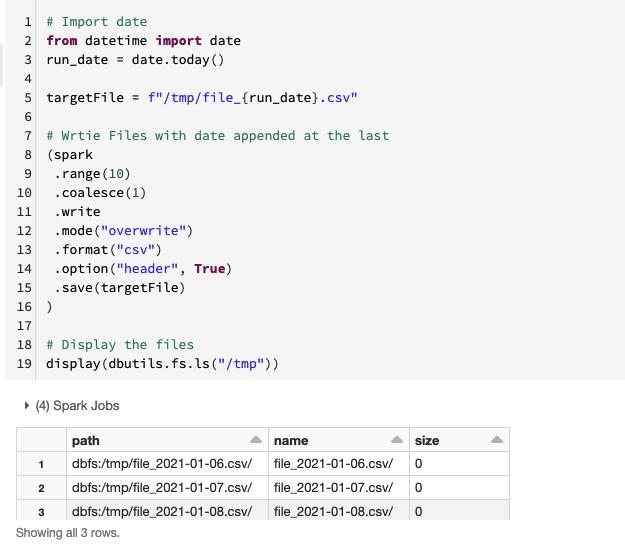
#Question314: How to find the Cluster id in Databricks ?  
  
Step1: Navigate to the cluster Tab  
  
Step2: Click on the Cluster Name  
  
Step3: We have 2 options at right ( default UI is selected ) - UI / JSON. Click on JSON  
  
Step4: Right at the bottom, we will see cluster\_id Key - Value pair.



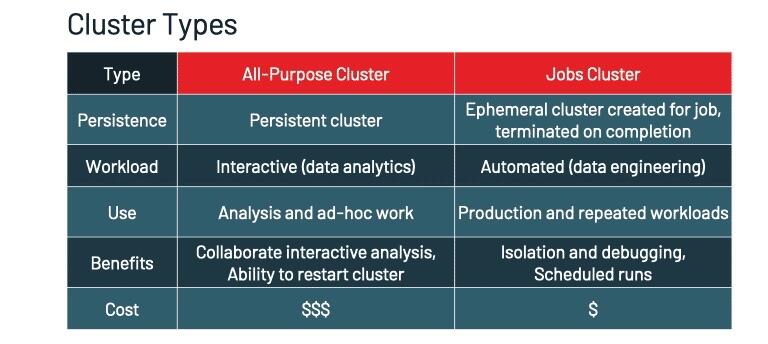
#Question315: How can we rename "Part files" to the desired name in Databricks ?  
  
Step1: Set the file name & paths  
  
Step2: Write the Files in Temp Location before moving them into the Target Path with desired name.  
  
Step3: Get the part file from the Temp Location  
  
Step4: Copy only the "part file" from Temp Location to Target Path with your desired name.  
  
Step5: Finally remove the Files from Temp Location. ( We don't need them now )  
  
Step6: List the Target File. ( Should have your desired file name )



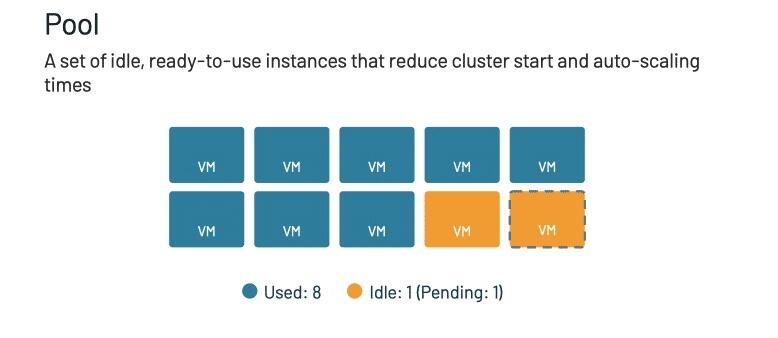
#Question316: How can we append current date while writing the files to create separate files for daily run in Pyspark?  
  
Step1: Import the date module  
  
Step2: Get the current date with date.today(). You can very well format it by your choice. You can also include the timestamp instead of date.  
  
Step3: Wrtie Files with date appended at the last  
  
Step4: Display the files  
  
Note: In the below example I have increased the date with relativedelta method and executed the code 3 times to show how the file names will look like when the code will run daily.



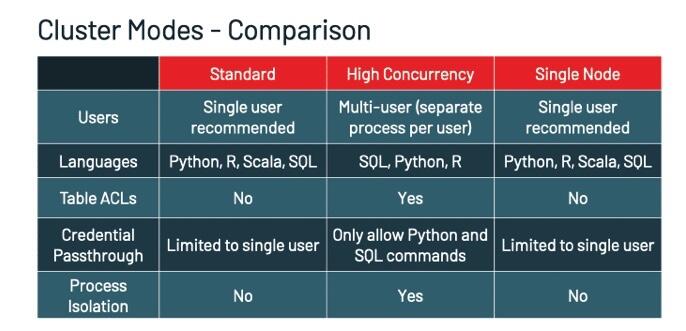
#Question317: How many Cluster Types we have in Databricks ?  
  
Two Types:  
  
1. All-Purpose Clusters  
2. Job Cluster  
  
When you create a cluster using the Clusters UI, CLI, or API, you are creating an all-purpose cluster, which can be used to run workloads interactively with notebooks.  
  
When you create a job, you have the option to use an existing all-purpose cluster, or create a new jobs cluster. Jobs clusters are ephemeral; they are created for the job and terminate upon completion, unlike all-purpose clusters, which are persistent and can be manually restarted.



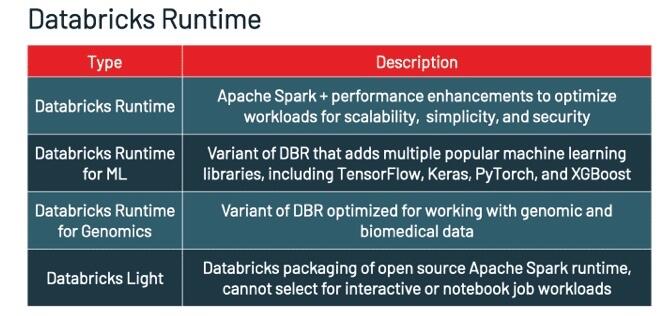
#Question318: What is a POOL in Databricks ?  
  
We can optionally configure clusters to pull virtual machines from a pool.  
  
A pool is a set of idle, ready-to-use instances that allow you to reduce cluster start and auto-scaling times.  
  
When a cluster attached to a pool needs an instance, it first attempts to allocate one of the pool’s idle instances.  
  
If the pool has no idle instances, the pool expands by allocating a new instance from the instance provider in order to accommodate the cluster’s request.  
  
When a cluster releases an instance, it returns to the pool and is free for another cluster to use. Only clusters attached to a pool can use that pool’s idle instances.  
  
We can manage pools using the Pools UI, CLI, and API.  
  
Attached a cluster to a pool saves instance acquisition time, which can be significant amount of time with many libraries and dependencies. Without pools, short jobs can waste more time starting than running the actual workload.



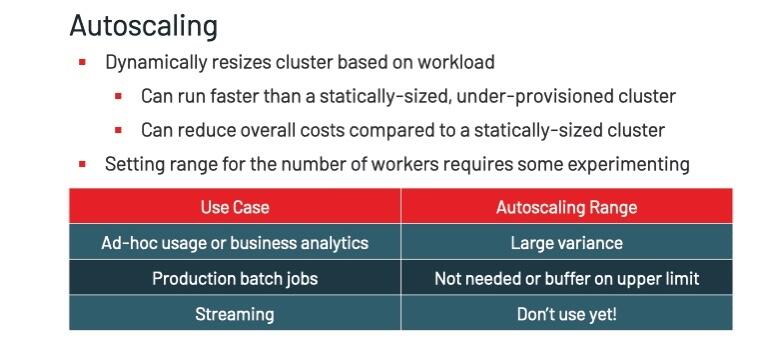
#Question319: What are different Cluster Modes in Databricks ?  
  
Three Modes:  
  
1. Standard  
2. High Concurrency  
3. Single Mode  
  
First is the standard cluster, which is recommended for a single user.  
  
Second is the high concurrency cluster, which is a managed cloud resource that can be shared among multiple users with fair scheduling and isolated notebook environments.  
  
A single node cluster has no workers and runs Spark jobs on the driver node. In contrast, standard mode clusters require at least one Spark worker node in addition to the driver node to execute Spark jobs.



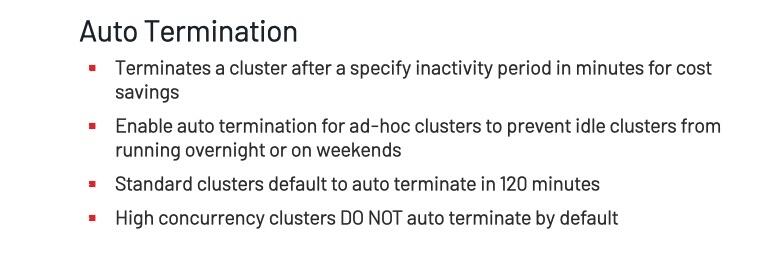
#Question320: What are different Databricks Runtime available ?  
  
Databricks offers several types of runtimes and several versions of those runtime types in the Databricks Runtime Version drop-down when you create or edit a cluster.  
  
1. Databricks Runtime includes Apache Spark but also adds a number of components and updates that substantially improve the usability, performance, and security of big data analytics.  
  
2. Databricks Runtime ML is a variant of Databricks Runtime that adds multiple popular machine learning libraries, including TensorFlow, Keras, PyTorch, and XGBoost.  
  
3. Databricks Runtime for Genomics is a variant of Databricks Runtime optimized for working with genomic and biomedical data.  
  
4. Databricks Light provides a runtime option for jobs that don’t need the advanced performance, reliability, or autoscaling benefits provided by Databricks Runtime.



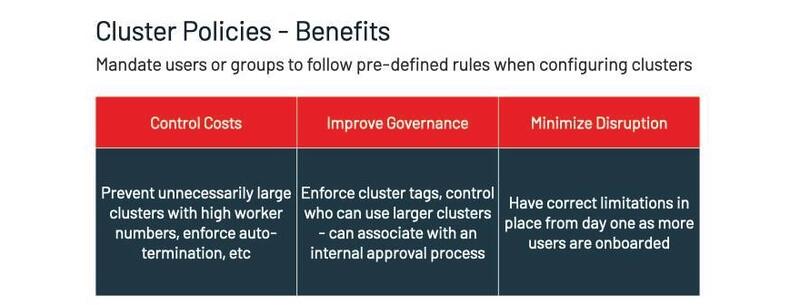
#Question321: What is Autoscaling in Databricks Cluster ?  
  
Autoscaling automatically adds and removes worker nodes in response to changing workloads to optimize resource usage.  
  
With autoscaling enabled, Databricks automatically chooses the appropriate number of workers required to run your Spark job.  
  
Autoscaling makes it easier to achieve high cluster utilization as you do not need to worry about the exact provisioning of cluster to match workloads.  
  
This can offer two advantages:  
  
(1) The workloads can run faster compared to running a constant sized under-provisioned cluster.  
(2) You can reduce overall costs compared to a statically sized cluster.  
  
When you enable autoscaling, you can specify the minimum and maximum number of nodes in the cluster.



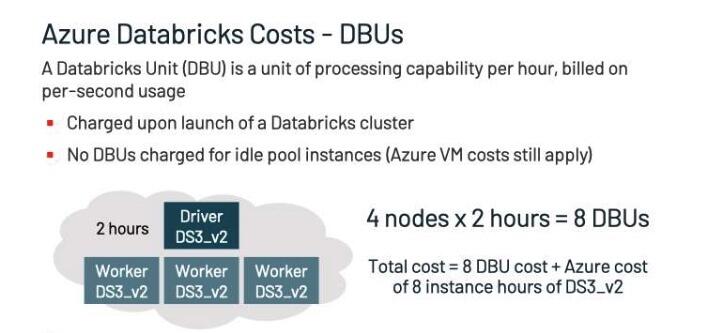
#Question322: What is Auto-termination in Databricks Cluster ?  
  
During cluster creation, you can specify an inactivity period in minutes after which you want the cluster to terminate.  
  
If the difference between the current time and the last command run on the cluster is more than the inactivity period specified, Databricks automatically terminates that cluster. It’s recommended to enable auto termination for ad-hoc clusters to prevent idle clusters from running overnight or on weekends.  
  
The default value of the auto terminate setting depends on whether you choose to create a standard or high concurrency cluster.  
  
Standard clusters are configured to terminate automatically after 120 minutes.  
  
High concurrency clusters are configured to not terminate automatically.



#Question323: What is Cluster Policies in Databricks Cluster ?  
  
As an admin, you can configure cluster policies to limit what attributes or attribute values are available for cluster creation.  
  
Users can then select a cluster policy from the policy drop-down on the cluster configuration page. You can configure ACLs that limit cluster policies to specific users and groups.  
  
Cluster policies allow you to:  
  
(1) Limit users to create clusters with prescribed settings.  
(2) Simplify the user interface by fixing and hiding some values to enable more users to create their clusters.  
(3) Control cost by limiting per cluster maximum cost (for example, by setting limits on attributes whose values contribute to hourly price).

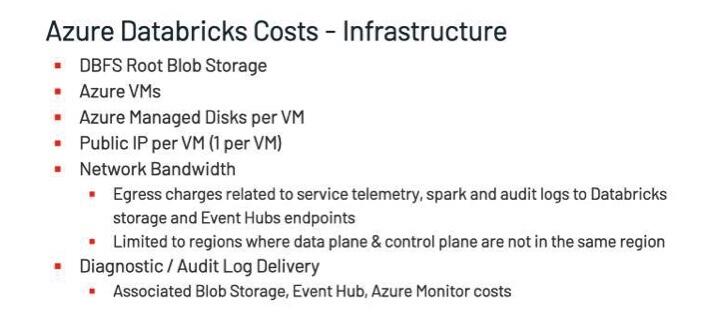


#Question324: How do Azure Databricks Charge to its customers ( DBU Costs ) ?  
  
Azure Databricks charges for Databricks usage by Databricks Unit (DBU), a unit of processing capability per hour.  
  
DBUs are charged upon launch of a Databricks cluster up to the second (there can be fractional DBU costs).  
  
No DBUs charged for idle pool instances, although Azure costs for the compute instances still apply.  
  
For example, if you create a cluster with one driver node and 3 worker nodes of type DS3\_v2 and run the cluster for 2 hours, you compute the DBUs as follows:  
  
Total instance hours = total number of nodes \* number of hours = (1+3) \* 2 = 8 DBUs.  
  
The total cost includes the cost of 8 DBUs, in addition to the Azure cost for 8 instances hours of the DS3\_v2 instances.

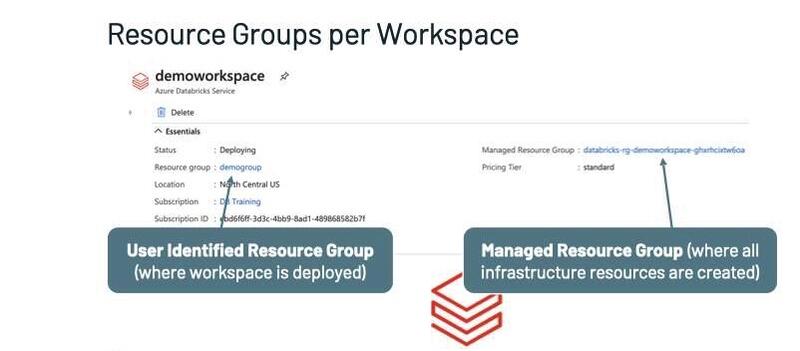


#Question325: When we get "429 Too Many Requests" response status code in Databricks ?  
  
Databricks set the following limits to ensure it can always guarantee jobs availability given the current capacity and architecture:  
  
1000 concurrent runs per workspace  
  
5000 runs / hour / workspace  
  
When these limits are exceeded users will receive 429 response codes and need to retry their request

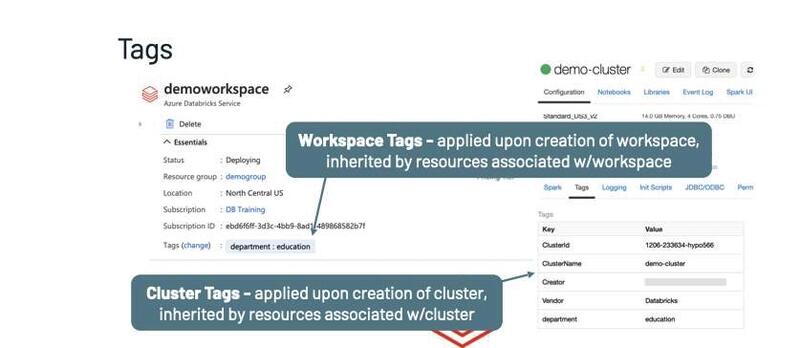
#Question326: What is the Infrastructure cost associated with Azure Databricks apart from the DBU cost ?  
  
In addition to DBU cost, there are also costs from Azure Databricks infrastructure.  
  
1. Blob storage costs for the DBFS root managed storage created for each workspace  
  
2. Azure VM costs for the instances used in clusters to run workloads  
  
3. Azure managed disk costs for the disks attached to each worker node, including a 30GB instance root disk attached to each VM and a 150GB data disk attached to the Databricks Runtime container in each VM  
  
4. Public IP Address costs as cluster VMs each have a dynamic public IP address unless the customer is using No Public IP feature  
  
5. Network Bandwidth costs from egress charges related to service telemetry, spark and audit logs to Databricks storage and Event hubs endpoints (if control plane and not in the same region)  
  
6. Optional Storage, Event Hub & Log Analytics costs associated with the delivery of diagnostic/audit logs



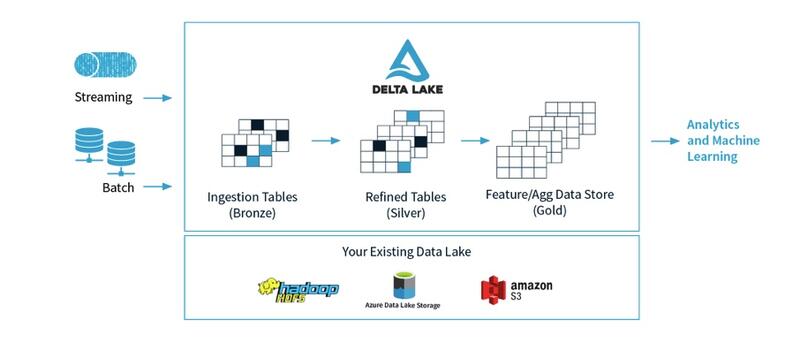
#Question327: How many Resource Groups are associated with Azure Databricks ?  
  
There are 2 Resource Groups associated with each Azure Databricks Workspace the user-identified resource group and the auto generated, managed resource group.  
  
The Azure Databricks workspace is deployed in the user identified resource group.  
  
All infrastructure resources are created in the managed resource group. The naming convention for this resource group is: “databricks-rg-”+ user specified resource group + “ws” + random characters



#Question328: What is Tags in Azure Databricks ?  
  
To monitor cost and accurately attribute Azure Databricks usage to your organization’s business units and teams (for chargebacks, for example).  
  
We can tag workspaces (resource groups), clusters, and pools. These tags propagate to detailed cost analysis reports that you can access in the Azure portal.  
  
These tags are applied upon creation of the workspace, cluster, or pool, and are inherited by associated infrastructure resources.



#Question329: What is Delta Lake ?  
  
Delta Lake is an open-source storage layer that brings ACID  
transactions to Apache Spark™ and big data workloads.  
  
Key Features:  
  
1. ACID Transactions  
2. Time Travel (data versioning)  
3. Open Format  
4. Schema Enforcement  
5. Schema Evolution  
6. Updates and Deletes  
7. 100% Compatible with Apache Spark API  
8. Audit History



#Question330: How to Create Database in PySpark ?  
  
Step1: Use spark.sql() or sql() method to execute SQL command in PySpark.  
  
Step2: Use - "CREATE DATABASE IF NOT EXISTS" command  
  
Step3: If we want to use that database, then issue the "USE" command  
  
Step4: For listing the available databases in the Hive Catalog use - spark.catalog.listDatabases() method. All databases should get listed.



#Question331: How to Create Delta Tables in PySpark ?  
  
Delta Lake supports creating tables directly based on the path using DataFrameWriter (Scala or Java/Python).  
  
Delta Lake also supports creating tables in the metastore using standard DDL CREATE TABLE. When you create a table in the metastore using Delta Lake, it stores the location of the table data in the metastore.  
  
Step1: Create table in the metastore by using - saveAsTable() method  
  
Step2: Create table by path using Save() method  
  
Step3: For listing the available tables in the Hive Catalog use - spark.catalog.listTables() method. All tables inside that databases should get listed.



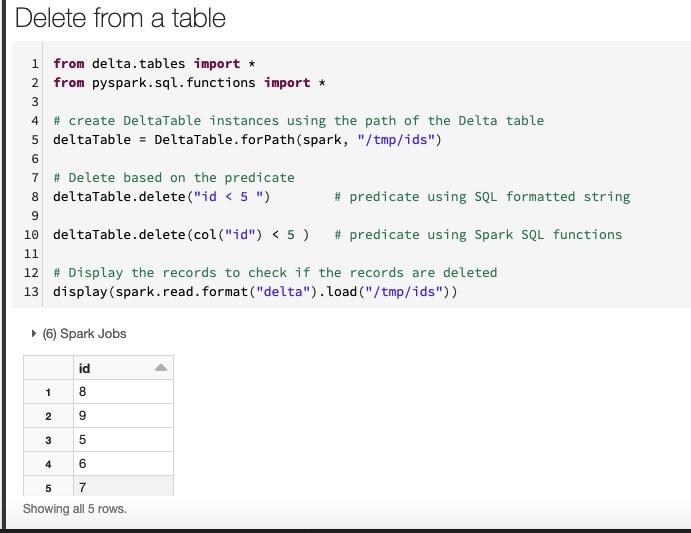
#Question332: How to Control Data Location while Creating Delta Tables in PySpark ?  
  
To control the location of the Delta table files, you can optionally specify the LOCATION as a path on DBFS.  
  
Tables created with a specified LOCATION are considered unmanaged by the metastore. Unlike a managed table, where no path is specified, an unmanaged table’s files are not deleted when you DROP the table.  
  
When you run CREATE TABLE with a LOCATION that already contains data stored using Delta Lake, Delta Lake does the following:  
  
#Scenario1:  
If you specify only the table name and location. ( Like the below code snippet ) the table in the Hive metastore automatically inherits the schema, partitioning, and table properties of the existing data. This functionality can be used to “import” data into the metastore.  
  
#Scenario2:  
If you specify any configuration (schema, partitioning, or table properties), Delta Lake verifies that the specification exactly matches the configuration of the existing data.  
  
#Important-  
If the specified configuration does not exactly match the configuration of the data, Delta Lake throws an exception that describes the discrepancy.



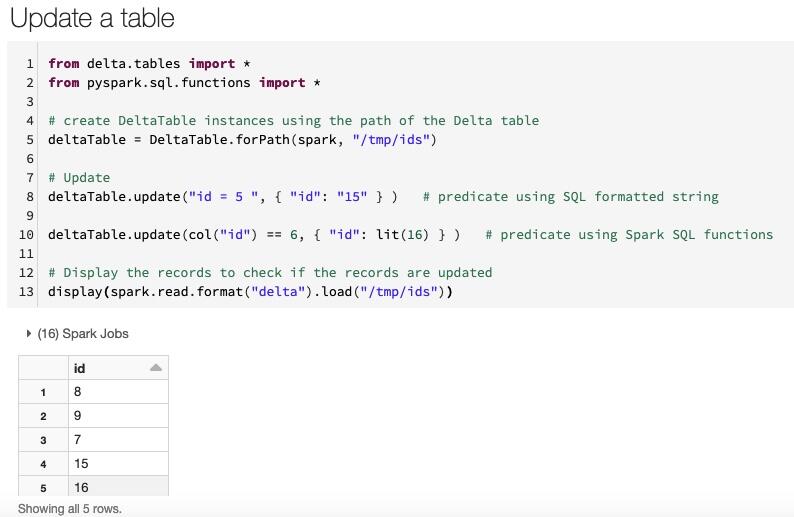
#Question333: How to Read Delta Tables in PySpark ?  
  
Step1: We can read the Delta Table by using the .table() method  
  
Step2: We can load a Delta table as a DataFrame by specifying a path  
  
#Note: The DataFrame returned automatically reads the most recent snapshot of the table for any query.  
  
We never need to run REFRESH TABLE. Delta Lake automatically uses partitioning and statistics to read the minimum amount of data when there are applicable predicates in the query.



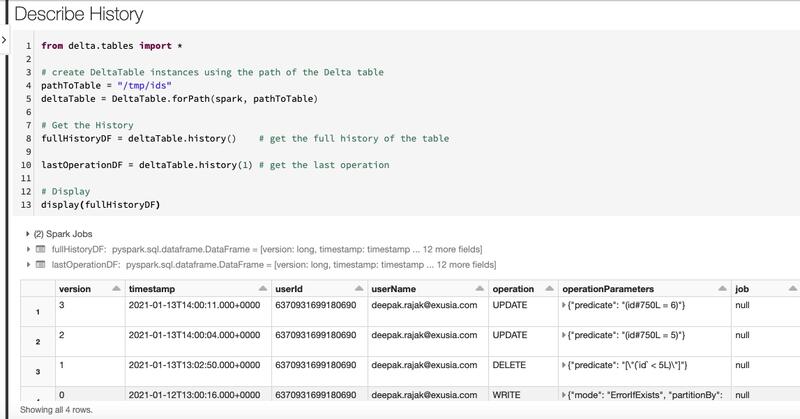
#Question334: How to Delete from Delta Tables in PySpark ?  
  
Step1: Use the method DeltaTable() to read the delta files from the path  
  
Step2: Predicate using SQL formatted string using .delete() method  
  
Step3: Predicate using Spark SQL functions using .delete() method



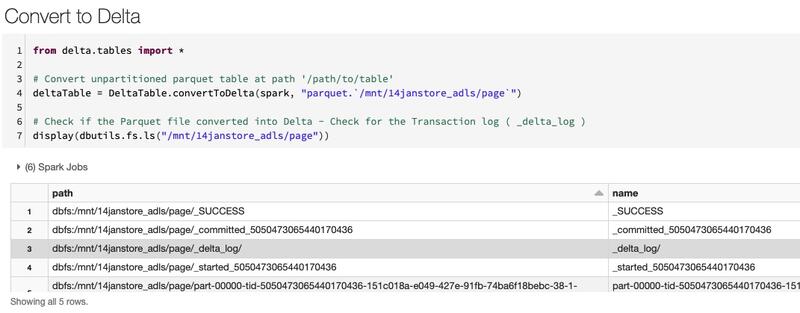
#Question335: How to Update records in Delta Tables in PySpark ?  
  
Step1: Use the method DeltaTable() to read the delta files from the path  
  
Step2: Predicate using SQL formatted string using .update() method  
  
Step3: Predicate using Spark SQL functions using .update() method



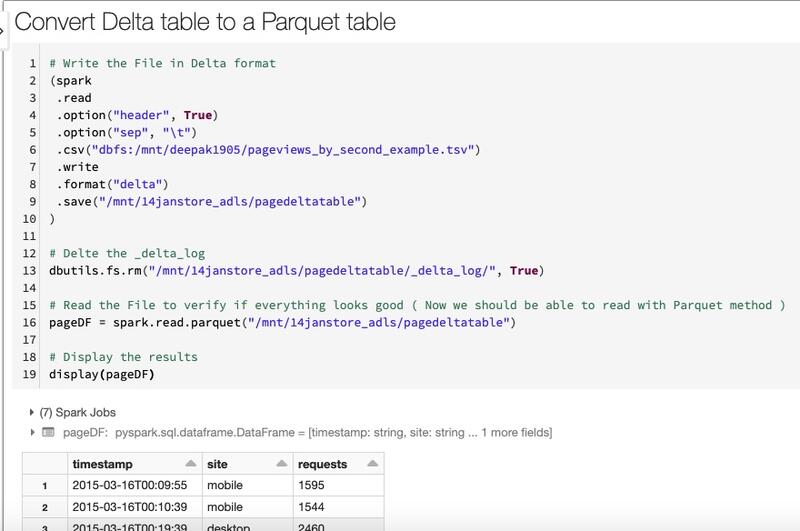
#Question336: How to Describe History in Delta Tables in PySpark ?  
  
Step1: Use the method DeltaTable() to read the delta files from the path  
  
Step2: get the full history of the table using .history() method  
  
Step3: get the last operation using .history(1) method



#Question337: How to Convert Parquet Files into in Delta Tables in PySpark ?  
  
Step1: Use the method convertToDelta()  
  
Step2: Provide the parquet file path as argument - parquet.'/path/to/table'  
  
Step3: Check for the Transaction Log to verify whether the conversion is successful



#Question338: How to Convert Delta Files/Tables back into Parquet Files in PySpark ?  
  
Step1: If you have performed Delta Lake operations that can change the data files (for example, delete or merge), run vacuum with retention of 0 hours to delete all data files that do not belong to the latest version of the table.  
  
  
Step2: Delete the \_delta\_log directory in the table directory.  
  
  
Step3: Check if the conversion is successful by reading the Parquet File



#Question339: How to Write data into existing Delta Table using Append Mode in PySpark ?  
  
Step1: Get the new data in the dataframe or in the table which you want to append.  
  
Step2: While writing use - mode("append")  
  
Step3: You can write into the path via the .save() method or write into the table via the .saveAsTable() method.  
  
Step4: Check for the History table - operationParameters column & operation column if the write is recorded.  
  
Step5: Verfiy the data via reading back the delta table by using the spark.read.format("delta").load("path/delta/table") command.

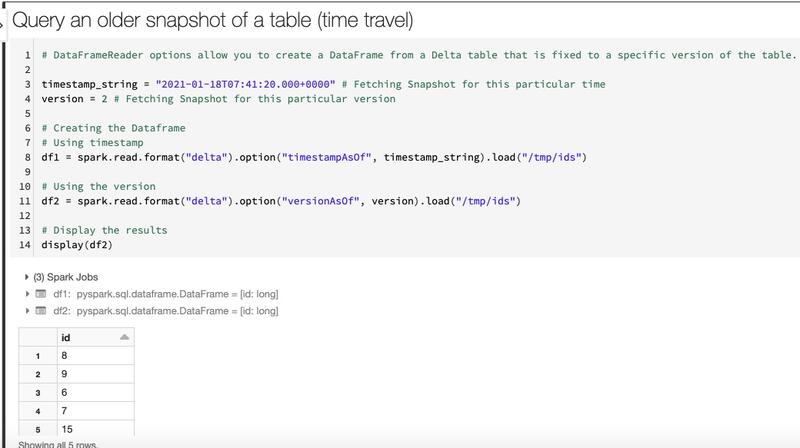


#Question340: How to Write data into existing Delta Table using Overwrite Mode in PySpark ?  
  
To atomically replace all of the data in a table, you can use overwrite mode  
  
Step1: Get the new data in the dataframe or in the table which you want to overwrite.  
  
Step2: While writing use - mode("overwrite")  
  
Step3: You can overwrite into the path via the .save() method or write into the table via the .saveAsTable() method.  
  
Step4: Check for the History table - operationParameters column & operation column if the write is recorded.  
  
Step5: Verfiy the data via reading back the delta table by using the spark.read.format("delta").load("path/delta/table") command.

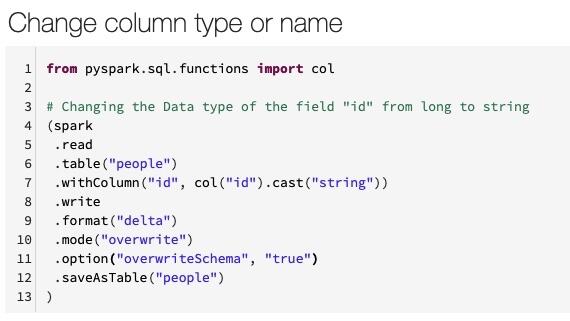


#Question341: How Delta Lake Validate Schema ?  
  
Delta Lake automatically validates that the schema of the DataFrame being written is compatible with the schema of the table.  
  
Delta Lake uses the following rules to determine whether a write from a DataFrame to a table is compatible  
  
Rule1: All DataFrame columns must exist in the target table. If there are columns in the DataFrame not present in the table, an exception is raised. Columns present in the table but not in the DataFrame are set to null.  
  
Rule2: DataFrame column data types must match the column data types in the target table. If they don’t match, an exception is raised.  
  
  
Rule3: DataFrame column names cannot differ only by case. This means that We cannot have columns such as “Foo” and “foo” defined in the same table.  
  
While we use Spark in case sensitive or insensitive (default) mode, Parquet is case sensitive when storing and returning column information.  
  
Delta Lake is case-preserving but insensitive when storing the schema and has this restriction to avoid potential mistakes, data corruption, or loss issues.

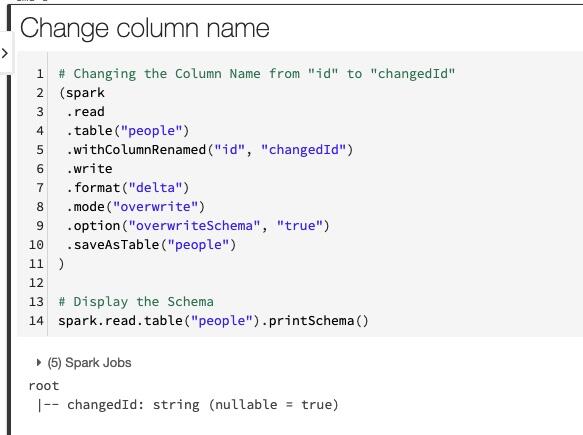
#Question342: How to Query an older snapshot of a table (time travel) in Delta Lake ?  
  
Delta Lake time travel allows you to query an older snapshot of a Delta table.  
  
We can query the older snapshot either by version or timestamp.  
  
DataFrameReader options allow you to create a DataFrame from a Delta table that is fixed to a specific version of the table.  
  
#Solution1: Using the version number by querying the history table associated with the delta table  
  
#Solution2: Using the timestamp\_string. For timestamp\_string, only date or timestamp strings are accepted. For example, "2020-01-01" and "2020-01-01T00:00:00.000Z".



#Question343: How to Change column type in Delta Lake ?  
  
Changing a column’s type or name or dropping a column requires rewriting the table.  
  
To do this, We need use the overwriteSchema option.  
  
Step1: Read the Table via the .table() method  
  
Step2: Change the data type of a column using the .withColumn() method and casting it to appropriate data type.  
  
Step3: Use the overwriteSchema option to True  
  
Step4: Overwrite the table



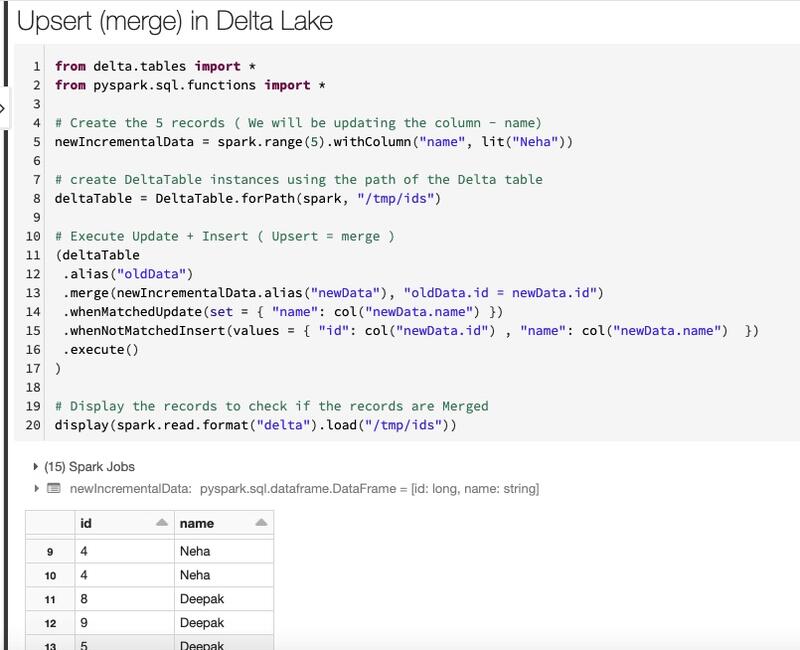
#Question344: How to Change column Name in Delta Lake ?  
  
Changing a column’s type or name or dropping a column requires rewriting the table.  
  
To do this, We need use the overwriteSchema option.  
  
Step1: Read the Table via the .table() method  
  
Step2: Change the Name of a column using the .withColumnRenamed() method  
  
Step3: Use the overwriteSchema option to True  
  
Step4: Overwrite the table  
  
Step5: Display the Schema using printSchema()



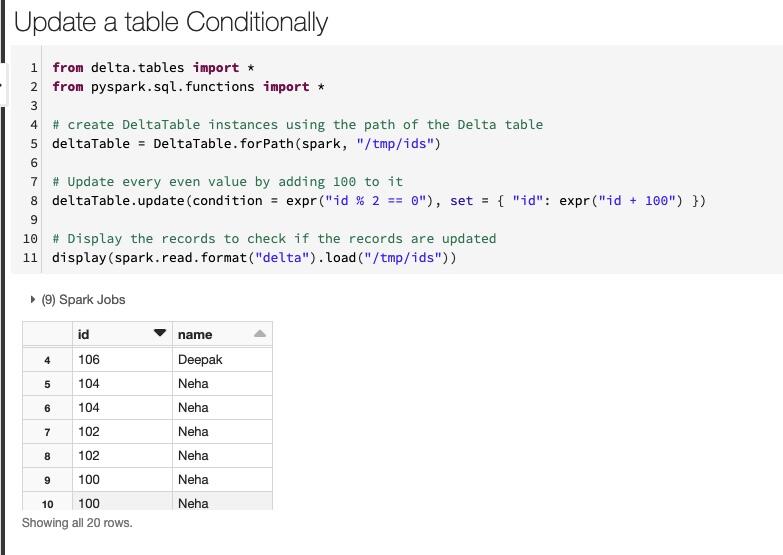
#Question345: How Automatic schema update happens in Delta Lake ?  
  
Delta Lake can automatically update the schema of a table as part of a DML transaction (either appending or overwriting), and make the schema compatible with the data being written.  
  
Add columns:  
  
Columns that are present in the DataFrame but missing from the table are automatically added as part of a write transaction when:  
  
Option1: write or writeStream have .option("mergeSchema", "true")  
  
Option2: spark.databricks.deltaschema.autoMerge.enabled is true  
  
Note: When both options are specified, the option from the DataFrameWriter takes precedence. The added columns are appended to the end of the struct they are present in. Case is preserved when appending a new column.  
  
Step1: Create the Dataframe with additional column. In the below example - "name"  
  
Step2: Append the DF with mergeSchema to True option  
  
Step3: Read back to check if everything is working fine. For existing records the name will be NULL. For new records the value will be populated.



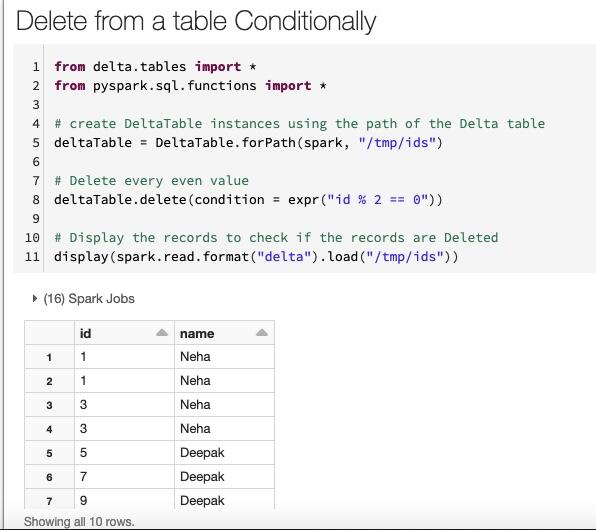
#Question346: How to perform UPSERT ( merge ) operations in Delta Lake in PySpark ?  
  
by using the .merge() method  
  
Step1: Create the Dataframe with new incremental data which we want to merge with the existing delta table. It can have both old records for updates & new records for inserts.  
  
Step2: Use the merge() method for the upset operation. use the key which is present in both the dataframe. we can have different key columns as well & multiple keys as well.  
  
Step3: whenMatchedUpdate for updating the records.  
  
Step4: whenNotMatchedInsert for inserting new records.  
  
Step5: Check if the records are merged properly by reading the table back



#Question347: How can we perform conditional updates ( update all even id values ) in Delta Lake in PySpark ?  
  
by using the .update() method  
  
Step1: Use the .update method.  
  
Step2: The method takes condition as argument. In the below example, we have provided - expr("id % 2 == 0") condition to only update the even id values.  
  
Step3: Also provide the value which we want to replace. In the below example we are adding 100 to every id value once we meet the above condition.  
  
Step5: Check if the records are udpdate properly by reading the table back



#Question348: How can we perform conditional Deletes ( Delete all even id values ) in Delta Lake in PySpark ?  
  
by using the .delete() method  
  
Step1: Use the .delete() method.  
  
Step2: The method takes condition as argument. In the below example, we have provided - expr("id % 2 == 0") condition to only delete the even id values.  
  
Step3: Check if the records are Deleted properly by reading the table back



#Question349: Does Delta Lake support multi-table transactions?  
  
Delta Lake does not support multi-table transactions and foreign keys. Delta Lake supports transactions at the table level.

#Question350: What are the best practices to choose the right partition column for Delta Lake Tables ?  
  
We can partition a Delta table by a column. The most commonly used partition column is date. Follow these two rules of thumb for deciding on what column to partition by:  
  
Strategy1: If the cardinality of a column will be very high, do not use that column for partitioning. For example, if you partition by a column userId and if there can be 1M distinct user IDs, then that is a bad partitioning strategy.  
  
Strategy2: Amount of data in each partition: You can partition by a column if you expect data in that partition to be at least 1 GB.

#Question351: What is Compaction in Delta Lake ?  
  
If We continuously write data to a Delta table, it will over time accumulate a large number of files, especially if we add data in small batches. This can have an adverse effect on the efficiency of table reads, and it can also affect the performance of our file system.  
  
Ideally, a large number of small files should be rewritten into a smaller number of larger files on a regular basis. This is known as compaction.  
  
Step1: We can compact a table by repartitioning it to smaller number of files. Using the repartition method  
  
Step2: In addition, we can specify the option dataChange to be false indicates that the operation does not change the data, only rearranges the data layout. This would ensure that other concurrent operations are minimally affected due to this compaction operation.  
  
#Note: This operation does not remove the old files. To remove them, run the VACUUM command.  
  
   



#Question352: Delta Lake supports multi-cluster writes ?  
  
Yes, Delta Lake supports Multi-Cluster writes.  
  
It means that Delta Lake does locking to make sure that queries writing to a table from multiple clusters at the same time won’t corrupt the table.  
  
However, it does not mean that if there is a write conflict (for example, update and delete the same thing) that they will both succeed. Instead, one of writes will fail atomically and the error will tell us to retry the operation.

#Question353: Can we modify a Delta table from different workspaces?  
  
Yes, We can.  
  
We can concurrently modify the same Delta table from different workspaces. Moreover, if one process is writing from a workspace, readers in other workspaces will see a consistent view.

#Question354: How to Create the Widget ( Parameter ) which takes Text values in Databricks ?  
  
  
The Widgets ( parameters ) will appear at the top of the notebook.  
  
Step1: Create the text widget via dbutils.widget.text()  
  
Step2: In the 3 arguments below - first is the name of the parameter, second one is the default value & third is the label name. ( second & third argument is optional )  
  
Step3: Read the widget via dbutils.widget.get()  
  
Step4: Print the values to check if everything looks good.  
  
Step5: Now we can use these parameter values in subsequent cells for performing different tasks.  
  
   



#Question355: What are clones in Delta Lake ? Shallow Clones  
Vs Deep Clones.  
  
Clones are replicas of a source table at a given point in time. They have the same metadata as the source table: same schema, constraints, column descriptions, statistics, and partitioning.  
  
However, they behave as a separate table with a separate lineage or history. Any changes made to clones only affect the clone and not the source.  
  
Any changes that happen to the source during or after the cloning process also do not get reflected in the clone due to Snapshot Isolation.  
  
In Databricks Delta Lake we have two types of clones: shallow or deep.  
  
Shallow Clones:  
  
A shallow (also known as Zero-Copy) clone only duplicates the metadata of the table being cloned; the data files of the table itself are not copied. This type of cloning does not create another physical copy of the data resulting in minimal storage costs. Shallow clones are inexpensive and can be extremely fast to create.  
  
Deep Clones:  
  
A deep clone makes a full copy of the metadata and data files of the table being cloned.  
  
   



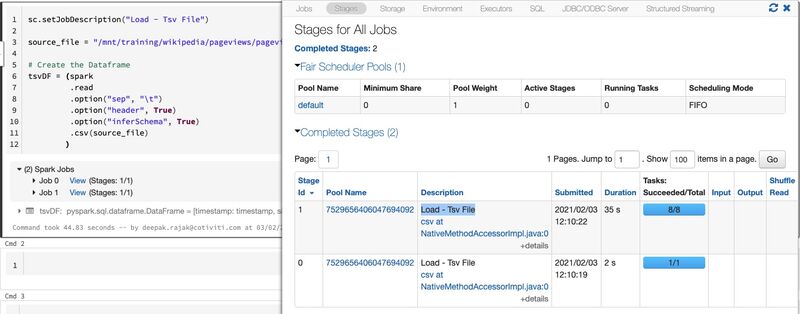
#Question356: How can We control the History of Delta Lake Table ? Or How we can control the Time Travel duration in Delta lake ?  
  
We can easily control the history of your Delta Lake table retention by the following properties:  
  
#delta.logRetentionDuration: Controls how long the history for a table (i.e. transaction log history) is kept. By default, 30 days of history is kept but you may want to alter this value based on your requirements (e.g. GDPR historical context)  
  
#delta.deletedFileRetentionDuration: Controls how long ago a file must have been deleted before being a candidate for VACUUM. By default, data files older than 7 days are deleted.  
  
In the below example we are increasing the transaction log retention to 60 days & deleted file retention to 15.  
  
   



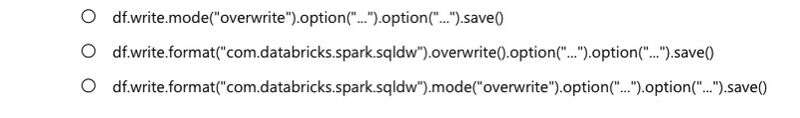
#Question357: What's the performance impact on live queries, when VACUUM is in progress?  
  
There should be minimal to no impact to live queries as vacuum is typically running on data that is on a different set of files than your queries.  
  
Where there is a potential impact is if you’re doing a time travel query on the same data that you’re about to vacuum (e.g. running vacuum of default of 7 days while attempting to query data that is older than 7 days).

#Question358: When & When not to Cache and Persist?  
  
When to Cache and Persist:  
  
Common use cases for caching are scenarios where you will want to access a large data set repeatedly for queries or transformations. Some examples include:  
  
1. DataFrames commonly used during iterative machine learning training.  
  
2. DataFrames accessed commonly for doing frequent transformations during ETL or building data pipelines.  
  
When Not to Cache and Persist:  
  
Not all use cases dictate the need to cache. Some scenarios that may not warrant caching your DataFrames include:  
  
1. DataFrames that are too big to fit in memory.  
  
2. An inexpensive transformation on a DataFrame not requiring frequent use, regardless of size  
  
As a general rule we should use memory caching judiciously, as it can incur resource costs in serializing and deserializing, depending on the StorageLevel used.

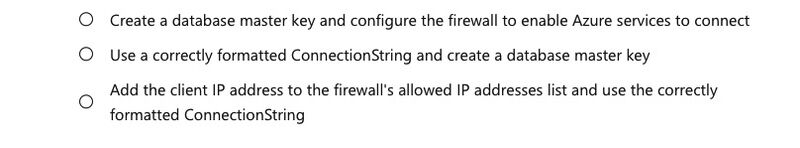
#Question359: How can be set the description for Spark Jobs & Stages to identify which job correspond to which command ?  
  
by using - sc.setJobDescription  
  
Step1: Pass the custom msg in the command - sc.setJobDescription  
  
Step2: Check in Spark UI. Each stage corresponds to that job can be recognised by the user defined message.  
  
Step3: In the below example, I have passed the msg - "Load - Tsv File"  
  
Step4: We can see that in Spark UI.  
  
   



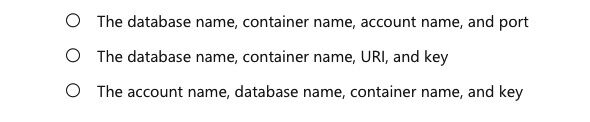
#Question360: Which is the correct syntax for overwriting data in Azure Synapse Analytics from a Databricks notebook?  
  
Please write your answer in the comment section.



#Question361: What are the two prerequisites for connecting Azure Databricks with Azure Synapse Analytics that apply to the Azure Synapse Analytics instance?  
  
Please write your answer in the comment section.



#Question362: When connecting to Azure Cosmos DB from a Databricks notebook, what information is required?  
  
Please write your answer in the comment section.



#Question363: How many Spark Jobs will be created for - csvdf & for - parquetdf? & Why ?  
  
Code Snippet is below.  
  
Please write your answer in the comment section.



#Question364: Which DataFrame #csvdf or #parquetdf will be created faster assuming we have the same number of records in csv file & in the parquet file & #Why ?  
  
Code Snippet is below.  
  
Please write your answer in the comment section.

