#	Input	Output
1	**Data Description** ➢ All the customer records are stored in the HDFS directory */user/spark/dataset/retail_db/customers-tab-delimited* ➢ Data is in Text format ➢ Data is Tab delimited **Schema** customer_id int customer_fname string customer_lname string customer_email string customer_password string customer_street string customer_city string customer_state string customer_zipcode string	**Output Requirement** ➢ Output all the customers who live in California ➢ Use text format for the output files ➢ Place the result data in */user/spark/dataset/result/scenario1/solution* ➢ Result should only contain records that have state value as "CA" ➢ Output should only contain customer's full name Example: Robert Hudson
2	**Data Description** ➢ All the order records are stored in the HDFS directory */user/spark/dataset/retail_db/orders_parquet* ➢ Data is in Parquet format **Output Requirement** ➢ Output all the completed orders (where order_status value = "COMPLETE") ➢ Use JSON format for the output files ➢ Place the result data in HDFS directory */user/spark/dataset/result/scenario2/solution* ➢ order_date should be in format yyyy-MM-dd ➢ Compress the output using gzip compression ➢ Output should only contain order_id, order_date,status	
3	**Data Description** ➢ All the customer records are stored at */user/spark/dataset/retail_db/customers-tab-delimited* ➢ Data is in text format ➢ Data is tab delimited **Schema** customer_id int customer_fname string customer_lname string customer_email string customer_password string customer_street string customer_city string customer_state string customer_zipcode String	**Output Requirement** ➢ Output all the customers who live in "Caguas" city ➢ Place the result data in HDFS directory /user/spark/dataset/result/scenario3/solution ➢ Result should only contain records that have customer_city value as "Caguas" ➢ Compress the output using snappy compression ➢ Save the output using orc format

4	**Data Description** ➢ All the categories records are stored at */user/spark/dataset/retail_db/categories* ➢ Data is in text format ➢ Data is comma separated **Schema** category_id int category_department_id int category_name string	**Output Requirement** ➢ Convert data into tab delimited file ➢ Use text format for the output files ➢ Place the result data in HDFS directory /user/spark/dataset/result/scenario4/solution ➢ Compress the output using lz4 compression
5	**Data Description** ➢ All the product records are stored at */user/spark/dataset/retail_db/products_avro* ➢ Data is in avro format ➢ Data is compressed with snappy compression **Output Requirement** ➢ Output should only contain the products with price greator than 1000.0 ➢ Use parquet format for the output files ➢ Place the result data in HDFS directory */user/spark/dataset/result/scenario5/solution* ➢ Compress the output using snappy compression	
6	**Data Description** ➢ Get data from metastore table named "orders" ➢ Table is present in the database "default" **Output Requirement** ➢ Fetch orders from Jan-2013 to Dec-2013 ➢ Use parquet format for the output files ➢ Place the result data in HDFS directory */user/spark/dataset/result/scenario6/solution* ➢ Compress the output using Gzip compression	

7	**Data Description** ➢ All the category records are stored at */user/spark/dataset/retail_db/categories* ➢ Data is in text format ➢ Data is comma separated **Schema** category_id int category_department_id int category_name string **Output Requirement** ➢ Save all categories in metastore table categories_replica in default database ➢ Use no compression	
8	**Data Description** ➢ All the category records are stored at */user/spark/dataset/retail_db/categories* ➢ Data is in text format comma separated **Schema** category_id int category_department_id int category_name string **Output Requirement** ➢ create a metastore table named 'categories_parquet' ➢Table should only contain category_id, category_name ➢ Save all categories in metastore table categories_parquet ➢ Use parquet format for the output files	
9	**Data Description** ➢ All the product records are stored at */user/spark/dataset/retail_db/products_avro* ➢ Data is in avro format ➢ Data is compressed with snappy compression **NOTE:** ➢Use below command to start spark shell on Cloudera VM spark-shell --packages org.apache.spark:spark-avro_2.11:2.4.4 In CCA 175 ➢ spark shell is already enabled with avro packages. **Output Requirement** ➢ Output should contain columns product_id, product_price ➢ Save output as a JSON file ➢ Place the result data in HDFS directory */user/spark/dataset/result/scenario9/solution* ➢ Use no compression	

10	**Data Description** ➢ All the category records are stored at */user/spark/dataset/retail_db/categories* ➢ Data is in text format comma separated **Schema** category_id int category_department_id int category_name string **Output Requirement** ➢ Create a metastore table named 'categories_partitioned' ➢ Save all categories in metastore table categories_partitioned	