**Mock Test 1 - Big Data - PPT**

1. Write a PySpark code to read a CSV file named "employees.csv" containing the following columns: "employee\_id", "name", "age", "department". Display the top 10 records from the DataFrame.

**------------**

**Solution:**

**------------**

from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("Read CSV").getOrCreate()

df = spark.read.csv("path/to/csv/employees.csv", header=True, inferSchema=True, sep=';')

df.limit(10).show

3. Write a PySpark code to read a JSON file named "students.json" containing student records with the following schema: "name" (string), "age" (integer), "grade" (string). Filter the DataFrame to include only students whose age is greater than 18.

**------------**

**Solution:**

**------------**

schema = StructType([

StructField("name",StringType(),True),

StructField("age",IntegerType(),True),

StructField("grade",StringType(),True)

])

df\_with\_schema = spark.read.schema(schema) \.json("resources/students.json")

df\_with\_schema.printSchema()

df\_with\_schema.show()

6.  Retrieve all the customers from the "Customers" table whose age is greater than 25 and have made at least one purchase.

**-----------**

**Solution:**

**-----------**

SELECT \* FROM customers WHERE age > 25 GROUP BY customer\_id HAVING MIN(purchase) > 1;

7. Find the total number of orders placed by each customer and display the results in descending order of the number of orders.

**-----------**

**Solution:**

**-----------**

SELECT COUNT(DISTINCT(customer\_num)) FROM customers ORDER BY customer\_num 'DESC';

8. Retrieve the names of all products that are currently out of stock from the "Products" table.

**-----------**

**Solution:**

**-----------**

SELECT product\_name, units\_order , units\_stock FROM products WHERE (((Discontinued)=False) AND ((units\_stock)< units\_order));

9. Calculate the average price of all products in each category and display the results along with the category name.

**-----------**

**Solution:**

**-----------**

SELECT AVG(pro\_price), pro\_category FROM products GROUP BY pro\_category;

10. Retrieve the top 5 customers who have spent the highest total amount on purchases.

**-----------**

**Solution:**

**-----------**

SELECT customer\_id, SUM(purchase\_amt) AS HighTotalAmt FROM customers LIMIT 5;