Slip 1 – NoSQL (MongoDB CRUD)

Problem Statement:
Create a MongoDB collection named `Students`.
Perform the following using the Mongo shell or Python API:

1. Insert five student documents (fields: name, dept, marks).
2. Update marks of one student.
3. Delete one record.
4. Display all records.

Slip 2 – NoSQL (Querying Unstructured JSON Data)

Problem Statement:
Load a JSON file of product details into MongoDB.
Write queries to:

1. Display all products in the "Electronics" category.
2. Count the total number of items priced above ₹10,000.

Slip 3 – NoSQL (Aggregation Pipeline)

Problem Statement:
Use the MongoDB aggregation pipeline to group employees by department and calculate the **average salary per department**.
Display results in descending order of average salary.

Slip 4 – NoSQL (API Operations using PyMongo)

Problem Statement:
Using the PyMongo library in Python:

1. Connect to MongoDB.
2. Insert 3 employee documents.
3. Retrieve records where salary > 50,000.
4. Update one record and print all documents.

Slip 5 – Hive (Basic Querying)

Problem Statement:
Create a Hive table named `movies` using a CSV file (fields: title, type, release_year, country).
Write HiveQL queries to:

1. Display the number of movies per country.
2. Show top 5 recent release years

Slip 6 – Hive (Sorting and Aggregation)

Problem Statement:
Use a Hive dataset `sales_data` with fields (region, product, amount).
Write queries to:

1. Calculate total sales per region.
2. Sort results by total sales in descending order.

Slip 7 – Hive (Joins and Filtering)

Problem Statement:
Create two Hive tables:

- `customers (cust_id, name, city)`
- `orders (order_id, cust_id, amount)`
  Perform an inner join to find the **total order amount per customer**.

Slip 8 – Hive (User Defined Functions - UDFs)

Problem Statement:
Create a Hive UDF to convert movie titles to uppercase.
Apply the UDF on a table `movies` to display all titles in uppercase form.

Slip 9 – Pig (Basic Operations)

Problem Statement:
Write a Pig Latin script to:

1. Load a text file containing student data (name, marks).
2. Filter students who scored above 70.
3. Display their names and marks.

Slip 10 – Pig (Grouping and Aggregation)

Problem Statement:
Use Pig Latin to group sales data by product category and find the **average sales amount per category**.
Display the results.

Slip 11 – Pig (Join Operation)

Problem Statement:
Load two datasets in Pig:

- `employee_details (emp_id, name, dept_id)`
- `department (dept_id, dept_name)`
  Perform a join to display employee names along with their department names.

Slip 12 – Pig (Sorting and Filtering)

Problem Statement:
Write a Pig script to load movie dataset and:

1. Filter only “TV Shows”.
2. Sort them by release year in descending order.
3. Display the top 10 results.