Hive HQL Hands-on: Logs & Errors Analysis

Scenario

You have a log file containing application events. Each line contains a timestamp, log level (INFO, WARN, ERROR), service name, and message. Your task is to load, parse, and analyze these logs using Hive.

Sample Log Format (log_data.txt)

```
2025-05-01 10:00:00,INFO,AuthService,User login successful
2025-05-01 10:05:15,ERROR,PaymentService,Payment failed for user 101
2025-05-01 10:10:05,WARN,InventoryService,Low stock alert
2025-05-01 10:12:35,INFO,PaymentService,Payment processed for user 102
```

Step-by-Step HQL Tasks

1. Create External Table for Log Data

```
CREATE EXTERNAL TABLE IF NOT EXISTS logs (
  log_date STRING,
  log_level STRING,
  service_name STRING,
  message STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION '/user/hive/logs/';
```

2. Load Data into Table

```
hdfs dfs -mkdir -p /user/hive/logs
hdfs dfs -put log_data.txt /user/hive/logs/
```

3. Basic Log Queries

- a) View All Logs

```
SELECT * FROM logs;
```

- b) Count Total Logs

```
SELECT COUNT(*) FROM logs;
```

- c) Count Logs by Log Level

```
SELECT log_level, COUNT(*) AS total FROM logs GROUP BY log_level;
```

- d) Count ERROR logs per service

```
SELECT service_name, COUNT(*) AS error_count FROM logs WHERE log_level = 'ERROR' GROUP BY service_name;
```

---

4. Filter Logs by Date

```
SELECT * FROM logs WHERE log_date LIKE '2025-05-01%';
```

---

5. Create Partitioned Table by Date

```
CREATE EXTERNAL TABLE logs_partitioned (
  log_time STRING,
  log_level STRING,
  service_name STRING,
  message STRING
)
PARTITIONED BY (log_date STRING)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```

Bonus: Create a View for ERROR Logs

```
CREATE VIEW error_logs AS SELECT * FROM logs WHERE log_level = 'ERROR';
```
