DDBMS RU LAB Question Solve

Israk Ahmed
ID: 2037820103
CSE 2019-20

Question 1

(a) Create a database named **BankDB**. Under BankDB, create the following three tables with partitions for **Account** and **Transaction** tables partitioning on `branch_id`:

1. **Customer Table** (not partitioned)
   - customer_id (Primary Key, BIGINT)
   - name (VARCHAR(100), NOT NULL)
   - email (VARCHAR(100), UNIQUE)
2. **Account Table** (partitioned by branch_id)
   - account_id (BIGINT, NOT NULL)
   - customer_id (BIGINT, NOT NULL, FK → Customer)
   - branch_id (INT, NOT NULL)
   - balance (DECIMAL(12,2), default 0.00)
   - Primary Key (account_id, branch_id)
3. **Transaction Table** (partitioned by branch_id)
   - transaction_id (BIGINT, Primary Key)
   - account_id (BIGINT, NOT NULL, FK → Account)
   - branch_id (INT, NOT NULL)
   - amount (DECIMAL(12,2), NOT NULL)

(b) Insert `Customer.csv`, `Account.csv`, and `Transaction.csv` data into respective tables. If necessary you should use tmp table and delete the temp table after data insertion.

- Customer: 200 records → Customer.csv
- Account: 200 records → Account.csv
- Transaction: 500 records → Transaction.csv

(c) Write Hive commands to **add** a partition (branch_id=6) to the **Account and Transaction** tables.

(d) Write Hive commands to **drop** the partition (branch_id=3) from the **Account and Transaction** tables.

(e) Show the number of transactions in each partition.

Solution

Step 1: Start Hadoop
[Run from: Windows Command Prompt In Docker]

```
# Run Hadoop Container
docker run-p 9870:9870-p 8088:8088-it--name=testHadoop macio232/hadoop-pseudo-
    distributed-mode

# Restart Old Hadoop Container
docker container start-i testHadoop
```

Step 2: Upload CSV Files to Docker Container
[Run from: Windows Command Prompt In Different Terminal In Docker]

```
# Copy CSV files from local Windows system to Docker container's /tmp
    directory
# Replace the paths below with your actual file locations

docker cp "F:\Study\Academic\Customer.csv" testHadoop:/tmp/

docker cp "F:\Study\Academic\Account.csv" testHadoop:/tmp/

docker cp "F:\Study\Academic\Transaction.csv" testHadoop:/tmp/

# Verify files are copied successfully (optional)
docker exec testHadoop ls -l /tmp/
```

Step 3: Start Hive Shell
[Run from: Hadoop Environment In Docker]

```
# Launch Hive interactive shell
hive
```

Part (a) & (b): Database and Table Creation with Data Loading

[Run from: Hive Shell - All commands below]

```
-- Create the BankDB database
CREATE DATABASE BankDB;

-- Switch to use the BankDB database
USE BankDB;

-- Create Customer table (not partitioned)
-- Stores customer information with id, name, and email
CREATE TABLE Customer (
    customer_id BIGINT,
    name STRING,
    email STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```

```
-- Create Account table (partitioned by branch_id)
-- Stores account details for each customer at specific branches
CREATE TABLE Account (
    account_id BIGINT,
    customer_id BIGINT,
    balance DECIMAL(12,2)
)
PARTITIONED BY (branch_id INT)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;

-- Create Transaction table (partitioned by branch_id)
-- Stores all transactions made on accounts at specific branches
CREATE TABLE Transaction (
    transaction_id BIGINT,
    account_id BIGINT,
    amount DECIMAL(12,2)
)
PARTITIONED BY (branch_id INT)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;

-- Create temporary table for Customer data loading
-- Used as staging area to load CSV data before moving to main table
CREATE TABLE Customer_tmp (
    customer_id BIGINT,
    name STRING,
    email STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;

-- Create temporary table for Account data loading
-- Includes branch_id as regular column (not partition) for loading
CREATE TABLE Account_tmp (
    account_id BIGINT,
    customer_id BIGINT,
    balance DECIMAL(12,2),
    branch_id INT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;

-- Create temporary table for Transaction data loading
-- Includes branch_id as regular column for loading
CREATE TABLE Transaction_tmp (
    transaction_id BIGINT,
    account_id BIGINT,
    amount DECIMAL(12,2),
    branch_id INT
)
ROW FORMAT DELIMITED
```

```
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;

-- Load Customer data from HDFS location into temporary table
LOAD DATA LOCAL INPATH '/tmp/Customer.csv' INTO TABLE Customer_tmp;

-- Copy all data from temporary table to main Customer table
INSERT INTO TABLE Customer
SELECT * FROM Customer_tmp;

-- Verify Customer data loaded successfully
SELECT * FROM Customer;

-- Load Account data from HDFS location into temporary table
LOAD DATA LOCAL INPATH '/tmp/Account.csv' INTO TABLE Account_tmp;

-- Enable dynamic partitioning for efficient data insertion
SET hive.exec.dynamic.partition = true;
SET hive.exec.dynamic.partition.mode = nonstrict;

-- Insert data into partitioned Account table
-- Hive automatically creates partitions based on branch_id values
INSERT INTO TABLE Account PARTITION (branch_id)
SELECT account_id, customer_id, balance, branch_id FROM Account_tmp;

-- Verify Account data loaded successfully across partitions
SELECT * FROM Account;

-- Load Transaction data from HDFS location into temporary table
LOAD DATA LOCAL INPATH '/tmp/Transaction.csv' INTO TABLE Transaction_tmp;

-- Insert data into partitioned Transaction table
-- Dynamic partitioning creates separate folders for each branch_id
INSERT INTO TABLE Transaction PARTITION (branch_id)
SELECT transaction_id, account_id, amount, branch_id FROM Transaction_tmp;

-- Verify Transaction data loaded successfully across partitions
SELECT * FROM Transaction;

-- Clean up: Drop temporary tables after data migration
DROP TABLE Customer_tmp;
DROP TABLE Account_tmp;
DROP TABLE Transaction_tmp;
```

Part (c): Add Partition (branch_id=6)

[Run from: Hive Shell]

```
-- Add new empty partition for branch_id=6 to Account table
-- Creates storage location for future branch 6 account data
ALTER TABLE Account ADD IF NOT EXISTS PARTITION (branch_id=6);

-- Add new empty partition for branch_id=6 to Transaction table
-- Creates storage location for future branch 6 transaction data
ALTER TABLE Transaction ADD IF NOT EXISTS PARTITION (branch_id=6);
```

Part (d): Drop Partition (branch_id=3)

[Run from: Hive Shell]

```
-- Remove partition for branch_id=3 from Account table
-- Deletes all account data for branch 3 permanently
ALTER TABLE Account DROP IF EXISTS PARTITION (branch_id=3);

-- Remove partition for branch_id=3 from Transaction table
-- Deletes all transaction data for branch 3 permanently
ALTER TABLE Transaction DROP IF EXISTS PARTITION (branch_id=3);
```

Part (e): Show Number of Transactions in Each Partition

[Run from: Hive Shell]

```
-- Method 1: Count transactions by branch using GROUP BY
-- Returns branch_id and count of transactions for each partition
SELECT
    branch_id,
    COUNT(*) as transaction_count
FROM Transaction
GROUP BY branch_id
ORDER BY branch_id;

-- Method 2: Show all partition names in Transaction table
-- Displays list of existing partitions (e.g., branch_id=1, branch_id=2, ...)
SHOW PARTITIONS Transaction;

-- Method 3: Detailed statistics for each partition
-- Provides comprehensive analysis of transactions per branch
SELECT
    branch_id,
    COUNT(*) as num_transactions,
    SUM(amount) as total_amount,
    AVG(amount) as avg_amount,
    MIN(amount) as min_amount,
    MAX(amount) as max_amount
FROM Transaction
GROUP BY branch_id
ORDER BY branch_id;

-- Bonus: Describe table structure
-- Shows column names, data types, and partitioning information
DESC Transaction;


-- Insert New Values
INSERT INTO TABLE Account PARTITION (branch_id=6)
VALUES
  (201, 105, 5000.00),
  (202, 108, 7500.00);

-- Drop Database
DROP DATABASE bankdb CASCADE;
```

Exiting Hive and Stopping Services

[Run from: Hive Shell]

```
-- Exit Hive shell and return to command prompt
exit;
```

[Run from: Windows Command Prompt In Different Terminal In Docker]

```
# Stop Hadoop container when finished
docker stop testHadoop
```