Hadoop Exam Preparation Guide

1. Local Mode, MapReduce and Pig Analysis

1.1 A. Install Hadoop in Local Mode

1. Download Hadoop from Apache website
2. Extract to directory (e.g., /usr/local/hadoop)
3. Set environment variables:

```
export HADOOP_HOME=/path/to/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
```

4. Edit hadoop-env.sh to set JAVA_HOME
5. Configure core-site.xml, hdfs-site.xml, mapred-site.xml, and yarn-site.xml
6. Initialize HDFS: hadoop namenode -format
7. Start Hadoop: start-all.sh
8. Verify with jps

1.2 C. Pig Analysis on Agriculture Dataset

```
-- Load agriculture dataset
agri_data = LOAD 'agriculture.csv' USING PigStorage(',') AS (
    year:int, state:chararray, crop:chararray,
    area:int, production:int, rainfall:int
);

-- Filter for specific state
karnataka_data = FILTER agri_data BY state == 'Karnataka';

-- Group by crop type and calculate average production
crop_stats = GROUP agri_data BY crop;
crop_avg_production = FOREACH crop_stats GENERATE
    group AS crop,
    AVG(agri_data.production) AS avg_production;

-- Sort by average production
sorted_crops = ORDER crop_avg_production BY avg_production DESC;

-- Find correlation between rainfall and production
rainfall_production = FOREACH agri_data GENERATE
    state, crop, rainfall, production;

-- Store results
STORE rainfall_production INTO 'rainfall_production_analysis';
```

2. Pseudo Mode, MapReduce and Hive Queries

2.1 A. Install Hadoop in Pseudo Mode

1. Configure files for pseudo-distributed mode:
    - `core-site.xml`: Set `fs.defaultFS` to `hdfs://localhost:9000`
    - `hdfs-site.xml`: Set `dfs.replication` to 1
    - `mapred-site.xml`: Set `mapreduce.framework.name` to `yarn`
    - `yarn-site.xml`: Configure `yarn.nodemanager.aux-services`
2. Format namenode: `hdfs namenode -format`
3. Start services: `start-dfs.sh` and `start-yarn.sh`
4. Verify with `jps`

2.2 C. Hive Queries on Employee Table

```sql
-- Create employee table
CREATE TABLE employees (
  id INT,
  name STRING,
  salary FLOAT,
  designation STRING
) ROW FORMAT DELIMITED FIELDS TERMINATED BY ’,’;

-- Load data
LOAD DATA LOCAL INPATH ’employees.csv’ INTO TABLE employees;

-- Count by designation
SELECT designation, COUNT(*) as employee_count
FROM employees
GROUP BY designation;

-- Average salary by designation
SELECT designation, AVG(salary) as avg_salary
FROM employees
GROUP BY designation
ORDER BY avg_salary DESC;

-- Find highest paid employee
SELECT name, salary, designation
FROM employees
ORDER BY salary DESC
LIMIT 1;

-- Filter for specific designation
SELECT * FROM employees
WHERE designation = ’Manager’;
```

3. Weather Mining and Sqoop

3.1 B. Implement Hadoop Commands

1. **cat**: Display file content

```
hadoop fs -cat /path/to/file
```

2. **copyToLocal**: Copy from HDFS to local

```
hadoop fs -copyToLocal /hdfs/source /local/destination
```

3. **mkdir**: Create directory in HDFS

```
hadoop fs -mkdir /hdfs/new/directory
```

4. **copyFromLocal**: Copy from local to HDFS

```
hadoop fs -copyFromLocal /local/source /hdfs/destination
```

3.2 C. Sqoop: MySQL to HDFS (Health Data)

1. Create MySQL table:

```
CREATE TABLE cancer_patients (
  patient_id INT PRIMARY KEY,
  age INT,
  gender VARCHAR(10),
  cancer_type VARCHAR(50),
  stage VARCHAR(10),
  treatment VARCHAR(100),
  survival_months INT
);
```

2. Import to HDFS:

```
sqoop import \
  --connect jdbc:mysql://localhost/database_name \
  --username username \
  --password password \
  --table cancer_patients \
  --target-dir /user/hadoop/cancer_data \
  --fields-terminated-by ’,’ \
  -m 1
```

4. Pig Latin and Word Count

4.1 A. Pig Latin Scripts for Student Data

```
-- Load student data
students = LOAD ’student_data.csv’ USING PigStorage(’,’) AS (
    id:int, name:chararray, age:int, grade:int,
    subject:chararray, score:int
);

-- Group students by subject and calculate average scores
subject_scores = GROUP students BY subject;
avg_scores = FOREACH subject_scores GENERATE
    group AS subject,
    AVG(students.score) AS average_score;

-- Find top performers
top_students = FILTER students BY score > 90;
top_by_subject = GROUP top_students BY subject;
top_count = FOREACH top_by_subject GENERATE
    group AS subject,
    COUNT(top_students) AS high_achievers;

-- Store results
STORE avg_scores INTO ’student_avg_scores’;
```

4.2 C. Sqoop: HDFS to MySQL (Health Data)

1. Create MySQL table:

```
CREATE TABLE cancer_patients_export (
  patient_id INT PRIMARY KEY,
  age INT,
  gender VARCHAR(10),
  cancer_type VARCHAR(50),
  stage VARCHAR(10),
  treatment VARCHAR(100),
  survival_months INT
);
```

2. Export from HDFS:

```
sqoop export \
  --connect jdbc:mysql://localhost/database_name \
  --username username \
  --password password \
  --table cancer_patients_export \
  --export-dir /user/hadoop/cancer_data \
  --input-fields-terminated-by ’,’ \
  --input-lines-terminated-by ’\n’
```

5. Matrix Multiplication and Tableau

5.1 B. Execute Hadoop Commands

1. `hadoop fs -ls /`: List root directory
2. `hadoop fs -put localfile.txt /user/hadoop/`: Upload file
3. `hadoop fs -get /user/hadoop/file.txt ./`: Download file
4. `hadoop fs -rm /user/hadoop/oldfile.txt`: Remove file
5. `hadoop fs -chmod 755 /user/hadoop/script.sh`: Change permissions
6. `hadoop fs -count -q /user/hadoop/`: Show quota
7. `hadoop fs -du -s -h /user/hadoop/`: Show disk usage

5.2 C. Tableau with Bank.csv

1. **Import Data**:
   - Connect to Text file
   - Navigate to Bank.csv
   - Configure data types
2. **Save Workbook**:
   - File → Save or Ctrl+S
3. **Open Workbook**:
   - File → Open or Ctrl+O
4. **Share Workbook**:
   - Export as Packaged Workbook (.twbx)
   - Publish to Tableau Server/Online
   - Export as PDF/Image

6. Covid Dataset and Visualization

6.1 B. Analyze Covid-19 Dataset with HiveQL and Joins

```sql
-- Create tables
CREATE TABLE covid_cases (
  country STRING,
  date STRING,
  total_cases INT,
  new_cases INT,
  total_deaths INT,
  new_deaths INT
) ROW FORMAT DELIMITED FIELDS TERMINATED BY ’,’;

CREATE TABLE country_info (
  country STRING,
  population BIGINT,
  continent STRING
) ROW FORMAT DELIMITED FIELDS TERMINATED BY ’,’;

-- Basic join
SELECT c.country, c.date, c.total_cases, c.total_deaths,
       i.population, i.continent
FROM covid_cases c
JOIN country_info i ON c.country = i.country;

-- Calculate mortality rate
SELECT c.country,
       i.continent,
       MAX(c.total_cases) as max_cases,
       MAX(c.total_deaths) as max_deaths,
       (MAX(c.total_deaths) / MAX(c.total_cases)) * 100 as mortality_rate
FROM covid_cases c
JOIN country_info i ON c.country = i.country
GROUP BY c.country, i.continent, i.population
ORDER BY mortality_rate DESC;
```

6.2 C. Apply Charts on Agriculture Data

1. **Bar Charts**:
   - Drag ”Crop” to Columns and ”Production” to Rows
   - Sort by production values
   - Add data labels
2. **Legends**:
   - Add ”State” to Color on Marks card
   - Format legend by right-clicking → ”Edit Legend”
3. **Filters and Hierarchies**:
   - Create hierarchy with ”Year”, ”State”, ”Crop”
   - Add filters: Drag ”State” to Filters shelf
   - Create interactive filter
4. **Step Charts**:
   - Select ”Year” for Columns and ”Production” for Rows
   - Change Chart Type to ”Step Lines”
   - Add ”Crop” to Color
5. **Line Charts**:

- Place "Year" on Columns and "Rainfall" on Rows
- Add "State" to Color for multiple lines
- Use dual axis for "Production" and "Rainfall"

7. Matrix, Sqoop, and Maps

7.1 B. Transfer Agriculture Dataset from MySQL to HDFS using Sqoop

1. Create MySQL table:

```sql
CREATE TABLE agriculture (
  year INT,
  state VARCHAR(50),
  crop VARCHAR(50),
  area INT,
  production INT,
  rainfall INT
);
```

2. Import to HDFS:

```
sqoop import \
  --connect jdbc:mysql://localhost/database_name \
  --username username \
  --password password \
  --table agriculture \
  --target-dir /user/hadoop/agriculture_data \
  --fields-terminated-by ',' \
  -m 1
```

7.2 C. Apply Maps on Health Data with Locations

1. **Symbol Maps**:
   - Drag ”Longitude” to Columns and ”Latitude” to Rows
   - Switch to ”Map” mark type
   - Use ”Cancer Type” on Color for different symbols
   - Use ”Survival Months” for Size of markers

2. **Filled Maps**:
   - Drag ”State” or ”Region” to Detail
   - Change mark type to ”Map”
   - Place ”Cancer Count” on Color to create choropleth map

3. **Density Maps**:
   - Place geographic fields on Columns/Rows
   - Change mark type to ”Density”
   - Use ”Patient Count” for Color intensity

4. **Maps with Pie Charts**:
   - Create map with locations
   - Change mark type to ”Pie”
   - Add ”Cancer Type” to Color
   - Add ”Patient Count” to Size