**Guide to Managing Databases and Tables in Hive**

This guide reflects the step-by-step process of creating, managing, and querying databases and tables in Hive, including database creation, partitioning, table types, and data loading operations. It provides a clear and structured introduction to Hive database and table management tasks.

**Create Database Statement:**

hive> CREATE DATABASE IF NOT EXISTS Student;

**Verify the Databases List:**

hive> SHOW DATABASES;

default

Student

**To use The Database:**

hive> use Student;

**To See the Database Location:**

hive> DESCRIBE DATABASE EXTENDED student;

**Hive Table Types**

**Internal or Managed table:** You can drop the table with underlying data.

**External table:** You can drop an external table, only table metadata from Metastore will be removed but the underlying files will not be removed and still they can be accessed via HDFS commands, Spark or any other Hadoop compatible tools.

**Temporary table:** For temporary purpose.

**Transactional Table:** For transactional data purpose.

**Table Creation**

hive> CREATE External TABLE IF NOT EXISTS Student.Info(Id int,Name string,

Age int)

> PARTITIONED BY (Gender string)

> ROW FORMAT DELIMITED

> FIELDS TERMINATED BY ','

> STORED AS TEXTFILE

> LOCATION '/data/output/';

**To See the Table Structure:**

hive> DESCRIBE student.info;

**Insert Data (single record) into Table:**

hive> INSERT INTO student.info PARTITION(Gender=’M’) values(7,'Maruf',23);

hive> INSERT INTO student.info PARTITION(Gender=’F’) values(8,'Rina',50);

**For Bulk Data (multiple record) Load:**

hive> LOAD DATA INPATH '/samrat/data.txt' INTO TABLE Student.Info

PARTITION(Gender=’F’);

hive> LOAD DATA INPATH '/path/to/HDFS/dir/file.csv' OVERWRITE INTO TABLE Student.Info PARTITION (Gender='M');

hive> LOAD DATA INPATH '/path/to/HDFS/dir/file.csv' OVERWRITE INTO TABLE Student.Info PARTITION (Gender='F');

**To Retrieves the all data:**

hive> SELECT \* FROM Info;

**Conditional Data Retrieve:**

hive> SELECT \* FROM Info WHERE age=23;

**To See All Partitions:**

hive> USE student;

hive> show partitions Info;

**To Drop a Partition:**

hive>ALTER TABLE Student.Info DROP PARTITION (gender="F");

**To Repair a Partition:**

hive> Msck repair table Student.Info;

**Merging parttion**

To merge two partitions in Hive, you need to follow these steps:

1. \*\*Create a New Partition for Merging\*\*:

You will create a new partition that will hold the merged data from the two partitions.

2. \*\*Insert Data from Existing Partitions into the New Partition\*\*:

Using the `INSERT INTO` command, you can insert the data from the two existing partitions into the new merged partition.

3. \*\*Drop the Old Partitions\*\*:

Once the data has been merged into the new partition, you can drop the old partitions that you no longer need.

Here's how you can achieve this:

### Step 1: Create a New Partition

First, create a new partition that will store the merged data:

```sql

ALTER TABLE Student.Info ADD PARTITION (Gender='All');

```

### Step 2: Insert Data into the New Partition

Next, insert the data from both the `Gender='M'` and `Gender='F'` partitions into the new `Gender='All'` partition:

```sql

-- Insert data from 'M' partition

INSERT INTO TABLE Student.Info PARTITION (Gender='All')

SELECT Id, Name, Age FROM Student.Info WHERE Gender='M';

-- Insert data from 'F' partition

INSERT INTO TABLE Student.Info PARTITION (Gender='All')

SELECT Id, Name, Age FROM Student.Info WHERE Gender='F';

```

### Step 3: Drop the Old Partitions

After successfully merging the data, you can drop the original `M` and `F` partitions:

```sql

ALTER TABLE Student.Info DROP PARTITION (Gender='M');

ALTER TABLE Student.Info DROP PARTITION (Gender='F');

```

### Step 4: Verify the Merged Partition

You can now verify that the new partition has been created and the data has been merged successfully:

```sql

SELECT \* FROM Student.Info WHERE Gender='All';

```

### Step 5: Repair the Table (if necessary)

In case the Hive metastore is not aware of the changes made, you may need to repair the table:

```sql

MSCK REPAIR TABLE Student.Info;

```

This will ensure the metadata is updated to reflect the changes made to the partitions.