NAME:Aditya Somani CLASS:TE-A ROLL NO:T1851061

**ASSIGNMENT -4**

**Part A : Assignments based on the Hadoop**

**4. Write an application using HBase and HiveQL for flight information system which will include**

1. **Creating, Dropping, and altering Database tables**
2. **Creating an external Hive table to connect to the HBase for Customer Information Table**
3. **Load table with data, insert new values and field in the table, Join tables with Hive**
4. **Create index on Flight information Table**
5. **Find the average departure delay per day in 2008.**

**-------------------------------------------------------------------------------------------------------------------** HBase via Hive,

**DOWNLOAD & COPY:**

* Download hive on below path (nearly 93 MB): http://www.apache.org/dyn/closer.cgi/hive/
* Extract the .tar.gz file in Downloads/ and rename it to hive/ and move the folder to **/usr/lib/** path:

**sudo mv Downloads/hive /usr/lib**

**CHANGE THE OWNER:**

* Provide access to hive path by changing the owners and groups to hduser and hadoop respectively.

**sudo chown -R hduser:hadoop /usr/lib/hive**

**CONFIGURE ENVIRONMENT VARIABLES:**

* Configure environment variables in .bashrc file.

**su - hduser vim ~/.bashrc**

* Add following lines at the end of file

**export HIVE\_HOME=/usr/lib/hive/ export PATH=$PATH:$HIVE\_HOME/bin**

**export HADOOP\_USER\_CLASSPATH\_FIRST=true**

* Apply the changes: **source ~/.bashrc**

**MAKE DIRECTORIES:**

* Create temporary and folder for data warehouse of hive in HDFS as well as change the permissions.

**hadoop fs -mkdir /tmp**

**hadoop fs -mkdir -p /user/hive/warehouse hadoop fs -chmod g+w /tmp hadoop fs -chmod -R g+w /user/hive/warehouse**

**CONFIGURE HIVE:**

* To configure Hive with Hadoop, you need to edit the hive-env. sh file, which is placed in the

$HIVE\_HOME/conf directory. The following commands redirect to Hive config folder and copy the template file:

**cd $HIVE\_HOME/conf**

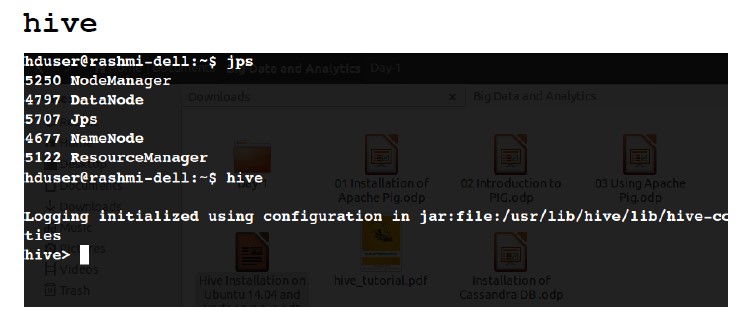
**cp hive-env.sh.template hive-env.sh**

* Edit the hive-env.sh file by appending the following line:

**export HADOOP\_HOME=/usr/local/hadoop**

**RUN HIVE:**

Make sure that Hadoop services are running.Then type hive



**HADOOP ECOSYSTEM:**

* The Hadoop ecosystem contains different subprojects (tools) such as Sqoop, Pig, and Hive that are used to help Hadoop modules.
* **Sqoop**: It is used to import and export data to and fro between HDFS and RDBMS.
* **Pig**: It is a procedural language platform used to develop a script for MapReduce operations. – **Hive**: It is a platform used to develop SQL type scripts to do MapReduce operations.

**DATABASE OPERATIONS;**

Hive is a database technology that can define databases and tables to analyze structured data. The theme for structured data analysis is to store the data in a tabular manner, and pass queries to analyze it. This chapter explains how to create Hive database. Hive contains a default database named **default**.

**CREATE DATABASE:**

* Create Database is a statement used to create a database in Hive.
* A database in Hive is a namespace or a collection of tables. The syntax for this statement is as follows:

**CREATE DATABASE|SCHEMA [IF NOT EXISTS]**

**<database name>;**

Here, IF NOT EXISTS is an optional clause, which notifies the user that a database with the same name already exists. We can use SCHEMA in place of DATABASE in this command.

* The following query is executed to create a database named mydb:

**hive>CREATE DATABASE [IF NOT EXISTS] mydb;** or

**hive>CREATE SCHEMA mydb;**

* The following query is used to verify a databases list:

**hive>SHOW DATABASES;**

**default mydb**

**DROP DATABSE:**

* Drop Database is a statement that drops all the tables and deletes the database. – Its syntax is as follows:

**DROP DATABASE StatementDROP (DATABASE|SCHEMA) [IF EXISTS]**

**database\_name [RESTRICT|CASCADE];**

* The following queries are used to drop a database. Let us assume that the database name is mydb.

**hive>DROP DATABASE IF EXISTS mydb;**

**CREATE TABLE:**

* Create Table is a statement used to create a table in Hive. The syntax and example are as follows:
* Syntax:

**CREATE [TEMPORARY] [EXTERNAL] TABLE [IF**

**NOT EXISTS] [db\_name.] table\_name [(col\_name data\_type [COMMENT**

**col\_comment], ...)]**

**[COMMENT table\_comment]**

**[ROW FORMAT row\_format] [STORED AS file\_format]**

# **Example:::**Create Table : Example

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Field Name** | **Data type** |
| 1 | Eid | Int |
| 2 | Name | String |
| 3 | Salary | Float |
| 4 | Designation | String |

• The following query creates a table named employee using the above data.

**hive>CREATE TABLE IF NOT EXISTS**

**employee ( eid int, name String, > salary String, destination String)**

**> COMMENT ‘Employee details’ > ROW FORMAT DELIMITED > FIELDS TERMINATED BY ‘\t’**

**> LINES TERMINATED BY ‘\n’**

**> STORED AS TEXTFILE;**

**ALTER TABLE:**

ALTER TABLE name RENAME TO new\_name

ALTER TABLE name ADD COLUMNS (col\_spec[, col\_spec ...])

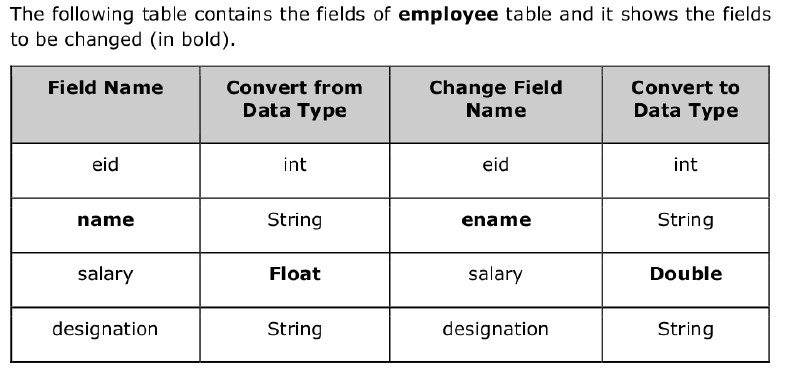
ALTER TABLE name DROP [COLUMN] column\_name ALTER TABLE name CHANGE column\_name new\_name new\_type

ALTER TABLE name REPLACE COLUMNS (col\_spec[, col\_spec ...])

**ALTER TABLE – RENAME TO:**

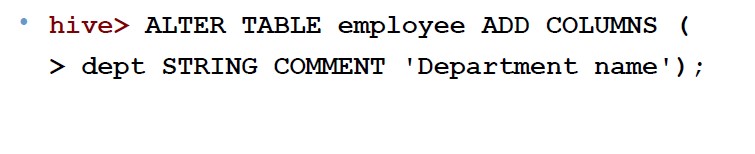
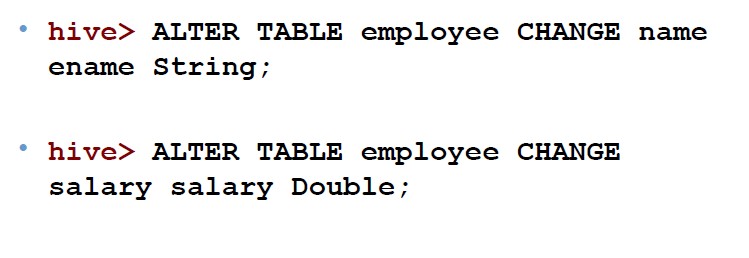
ALTER TABLE employee RENAME TO emp;

**CHANGE STATEMENT:**

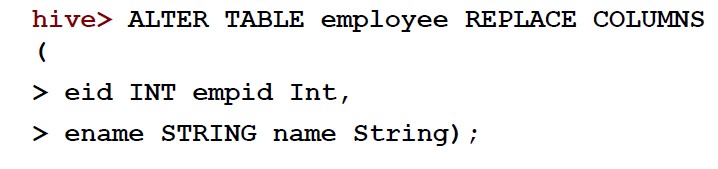


**CHANGE STATEMENT EXAMPLE:**

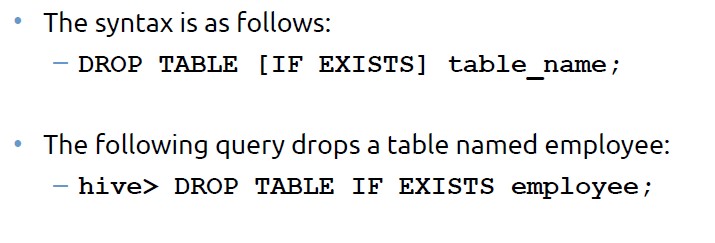
**ADD COLUMN STATEMENT:**



**REPLACE STATEMENT:**



**DROP TABLE STATEMENT:**



**INDEX:**

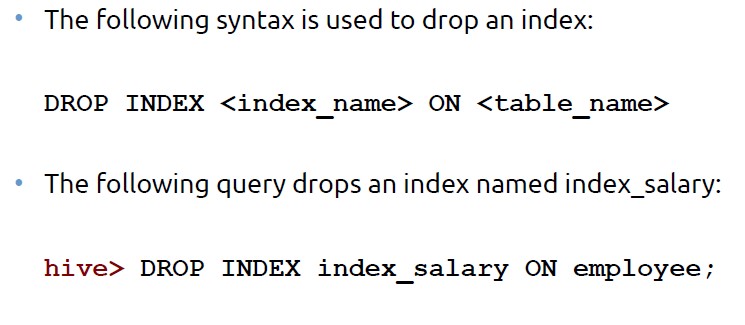
* An Index is nothing but a pointer on a particular columnof a table.
* Creating an index means creating a pointer on a particular column of a table.
* hive> CREATE INDEX index\_yoj ON TABLE file(yoj)

> AS 'org.apache.hadoop.hive.ql.index.compact.CompactIndexHandler'

WITH DEFERRED REBUILD;

**DROP INDEX:**

**Select…Order By:**



* The ORDER BY clause is used to retrieve the details based on one column and sort the result set by ascending or descending order.
* Syntax:

SELECT [ALL | DISTINCT] select\_expr, select\_expr, ...

FROM table\_reference

[WHERE where\_condition]

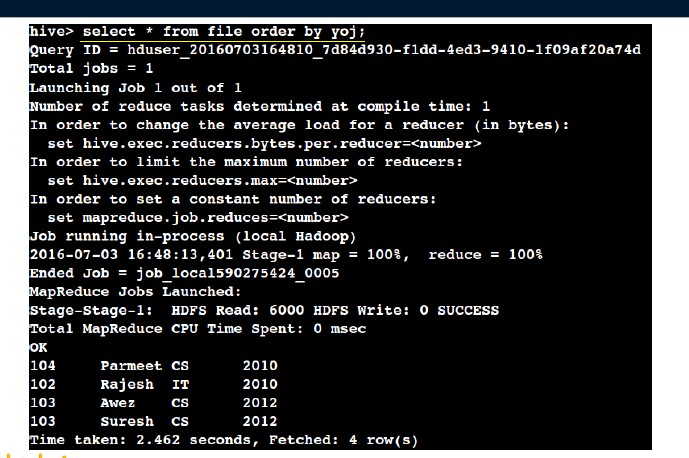
[GROUP BY col\_list]

[HAVING having\_condition]

[ORDER BY col\_list]]

[LIMIT number];

**Example:**



**Select…Group By:**

* The GROUP BY clause is used to group all the records in a result set using a particular collection column. It is used to query a group of records.
* Syntax:

SELECT [ALL | DISTINCT] select\_expr, select\_expr, ...

FROM table\_reference

[WHERE where\_condition]

[GROUP BY col\_list]

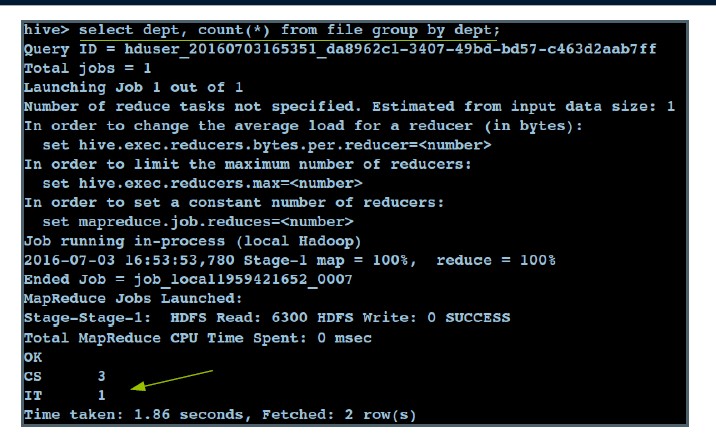
[HAVING having\_condition]

[ORDER BY col\_list]]

[LIMIT number];

Exam

ple:

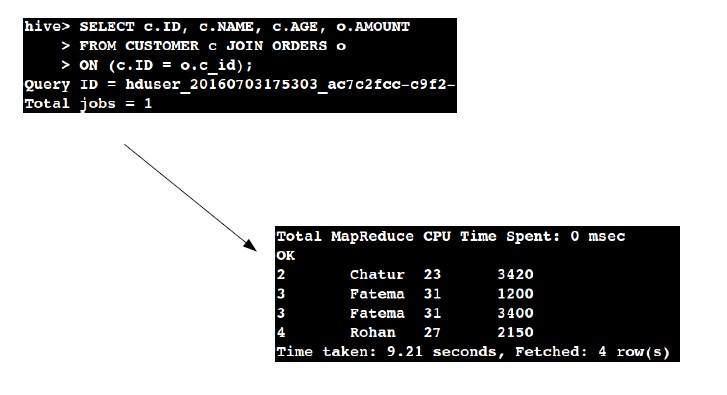
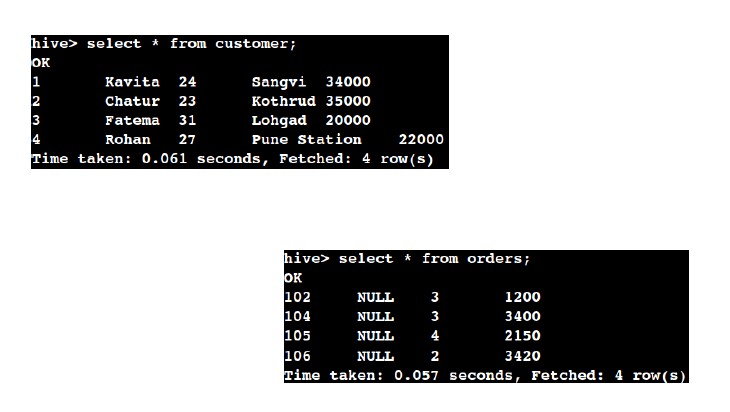


**JOINS:**

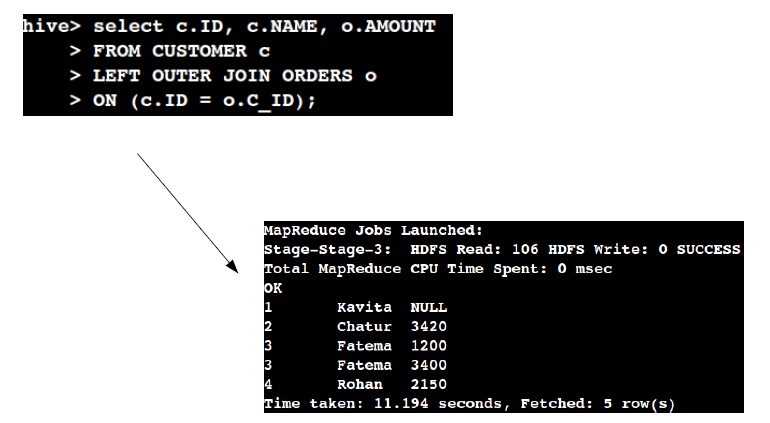
* JOINS is a clause that is used for combining specific fields from two tables by using values common to each one.
* It is used to combine records from two or more tables in the database.
* It is more or less similar to SQL JOINS.

Example:;

**Left Outer Join:**

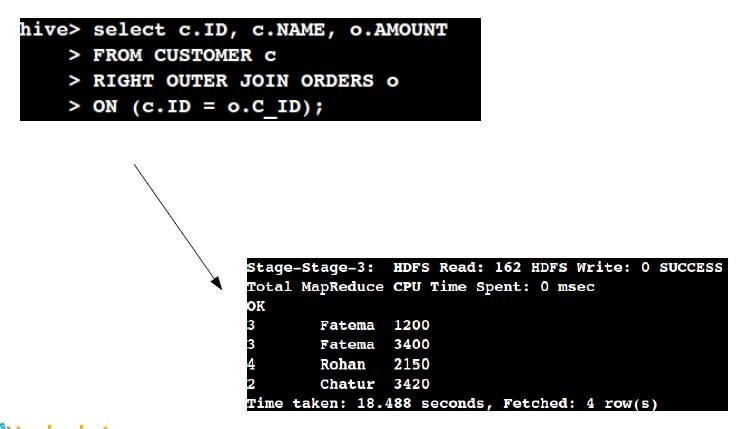


* The HiveQL LEFT OUTER JOIN returns all the rows from the left table, even if there are no matches in the right table.
* This means, if the ON clause matches 0 (zero) records in the right table, the JOIN still returns a row in the result, but with NULL in each column from the right table.
* A LEFT JOIN returns all the values from the left table, plus the matched values from the right table, or NULL in case of no matching JOIN predicate.



**Right Outer Join:**

* The HiveQL RIGHT OUTER JOIN returns all the rows from the right table, even if there are no matches in the left table.
* If the ON clause matches 0 (zero) records in the left table, the JOIN still returns a row in the result, but with NULL in each column from the left table.
* A RIGHT JOIN returns all the values from the right table, plus the matched values from the left table, or NULL in case of no matching join predicate.



**Conclusion:** Thus we have learnt how to execute queries with hive and various operations related to database (like Creating Table/Database, Dropping Table ,Altering Table, Joins in Table, Index in Table)with HiveQL & HBase.