**B1 - BDE Crossover Capstone Project Submission**

**Project : Student Competency Analysis**

**By Megha Rajan**

**GitHub Link -** [**https://github.com/Megha-Rajan/CapstoneProject**](https://github.com/Megha-Rajan/CapstoneProject)

**Introduction**

The competence of the students is the key to a college's success. To ensure competency level of student performance in various courses he participated and trained. The college placement officer is interested in getting insights about each student and recommending core companies based on his core strengths.

**Getting Started with the Project**

Step 1: Loading CSV files to HDFS

Graphical user interface, text, application

Description automatically generated

Step 2 : Set the Hive Metastore Warehouse Directory location

set hive.metastore.warehouse.dir=/user/ana002672/warehouse;

Step 3 : Creating new database

Database Name : capstoneproject

Graphical user interface, text, application, email

Description automatically generated

**Task 1: Defining Tables**

1.a TABLE 1

TABLE NAME : studentcoursecompletionstatus

DESCRIPTION : Data distributed randomly into 5 buckets based on marks column

**Step 1 :** Creating Bucketed Table

create table studentcoursecompletionstatus

(studentsid string, courseid string, examdate date, attendedstatus string, marks int, result string)

clustered by (marks) into 5 buckets

row format delimited fields terminated by ',';

**Step 2 :** Loading data from hdfs file to the table

load data inpath 'StudentCourseCompletionStatus.csv' into table studentcoursecompletionstatus;

Graphical user interface, text

Description automatically generated

**Step 3:** Verifying the data inside the table

Select \* from studentcoursecompletionstatus limit 10;

Graphical user interface, text, application

Description automatically generated

**Step 4:** Verifying the bucketed output file in the warehouse directory

hadoop fs -ls warehouse/capstoneproject.db/studentcoursecompletionstatus/

Graphical user interface, text, application

Description automatically generated

**Step 5:** Verifying the content inside the bucketed folder

hadoop fs -cat warehouse/capstoneproject.db/studentcoursecompletionstatus/000000\_0 | head

Graphical user interface, text, application, email

Description automatically generated

1.b TABLE 2

DESCRIPTION: Splitting the CourseDetails table into partitions using dynamic partitioning based on the CourseType column

STEPS FOLLOWED:

**Step 1**: We will enable the dynamic partition using the following commands

set hive.exec.dynamic.partition=true;

set hive.exec.dynamic.partition.mode=nonstrict;

**Step 2:** Create a table without partition to store the data.

create table studentcoursedetails\_no\_partition

(coursetype string, courseid string, title string, competency string, complexity string)

row format delimited fields terminated by ',';

**Step 3:** Loading data from hdfs file to the table

load data inpath 'StudentCourseDetails.csv' into table studentcoursedetails;

Graphical user interface, text, application, email

Description automatically generated

**Step 4 :** create a partitioned table

create table studentcoursedetails

(courseid string, title string, competency string, complexity string)

partitioned by (coursetype string)

row format delimited fields terminated by ',';

**Step 5 :** Now we will load the data from non-partitioned table to the partitioned table using the following command.

insert overwrite table studentcoursedetails

partition (coursetype)

select courseid, title, competency, complexity, coursetype from studentcoursedetails\_no\_partition

Graphical user interface, text, email

Description automatically generated

**Step 6:** verifying the data inside the table

Text

Description automatically generated

**Step 7:** Verifying the partitioned output folders inside the warehouse directory

hadoop fs -ls warehouse/capstoneproject.db/studentcoursedetails

Graphical user interface, text, email

Description automatically generated

**Step 8:** Verifying the content inside the partitioned file

hadoop fs -cat warehouse/capstoneproject.db/studentcoursedetails/coursetype=Cloud/\*

Graphical user interface, text

Description automatically generated

**Task 2: Data Analysis using Hive**

2.1 List the count of students qualified for various courses

select courseid, count(studentsid) as qualified\_count from studentcoursecompletionstatus where result="Qualified" group by courseid;

Graphical user interface, text, application

Description automatically generated

2.2 List the courses available

Select distinct courseid, title from studentcoursedetails;

Graphical user interface, text

Description automatically generated

2.3 Extract the StudentID, CourseTitle, Result from both table and save the following result set to HDFS output directory

insert overwrite directory "queryresult"

select s1.studentsid, s2.title, s1.result from studentcoursecompletionstatus s1 join studentcoursedetails s2 on s1.courseid=s2.courseid;

Text, email

Description automatically generated

Verifying the result set in the HDFS output directory

hadoop fs -cat queryresult/\* | head -n 25

Graphical user interface, text

Description automatically generated

**Task 3: PySpark**

Create PySpark dataframes and read data from the two Hive tables.

Find the following using the PySpark dataframes -

1. Find the total number of students per result category
2. Find the total number of students absent
3. Find the maximum, minimum and average marks scored by students

Solution : <https://github.com/Megha-Rajan/CapstoneProject/blob/main/Capstone.ipynb>

**Task 4: Version control**

GitHub Link : <https://github.com/Megha-Rajan/CapstoneProject>

4.1 Create a Github account and create a new repository.

Graphical user interface, text, application, email

Description automatically generated

4.2 Create a new branch in the repository

Commands Used:

git clone <https://github.com/Megha-Rajan/CapstoneProject.git>

git checkout -b branch1

Graphical user interface, text, application, email

Description automatically generated

4.3 Upload Pyspark Notebook to the new branch

Commands Used :

git add Capstone.ipynb

git commit -m "pyspark file added"

Graphical user interface, text, application, email

Description automatically generated

Push your changes back to the main branch

Commands Used :

git checkout main

git merge branch1

Graphical user interface, text, application, email

Description automatically generated

Git log

Graphical user interface, text, application, email

Description automatically generated

Push your changes to the online repository:

git push --set-upstream https://github.com/Megha-Rajan/CapstoneProject.git branch1

Graphical user interface, text, application, email

Description automatically generated

On Main branch

Graphical user interface, text, application

Description automatically generated

After Merge and Pull request

Graphical user interface, text, application, email

Description automatically generated