A. Create an external table in Hive from a csv file:

1. Copy the given csv file from local file system to hadoop file system:
   - Syntax: hdfs dfs -put <local-file-path> <directory-path-in-hadoop>
   - Eg: hdfs dfs -put ~/MTCS_202/PRACTICALS/A4_Hive/StatewiseTestingDetails.csv /user/cloudera/state

(This puts the statewise data into the state directory (which you should have already created)).

2. Create the external table:
   - Syntax: CREATE EXTERNAL TABLE IF NOT EXISTS <table-name> (<table-schema>) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE LOCATION '<directory-path-in-hadoop>'
   - Eg: CREATE EXTERNAL TABLE IF NOT EXISTS covid_states (Sno int, Date string, State_UnionTerritory string, TotalSamples int, Negative int, Positive int) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE LOCATION '/user/cloudera/state'

To see the table in hive, type show tables;
covid_states must be listed. SQL queries can now be run on this table.

B. Load the result of a query into a csv file in local filesystem:

- Syntax: INSERT OVERWRITE LOCAL DIRECTORY '<local-directory-path>' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' <query>;
- Eg: INSERT OVERWRITE LOCAL DIRECTORY 'MTCS_202/PRACTICALS/A4_Hive/Output/' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT State_UnionTerritory, count(Positive) FROM covid_states GROUP BY State_UnionTerritory;
- The query counts the number of positive cases for each state.
- The result is stored in the specified local directory as a part file. To store it in a csv file, run (for this example):
  cat Output/* > StatewisePositive.csv
  (Here, this command is run in the directory A4_Hive. That's why the path starts directly from Output; the csv file is stored in A4_Hive.)

C. Find the discrepancies between the positive cases reported in StatewiseTestingDetails.csv and the confirmed cases reported in Covid19_India.csv:

1. Obtain state wise confirmed cases:

```
SELECT State_UnionTerritory, sum(Confirmed) FROM covid19 GROUP BY State_UnionTerritory;
```

(covid19 is an external table in hive created from Covid19_India.csv)

2. Obtain state wise positive cases:

```
SELECT State_UnionTerritory, sum(Positive) FROM covid_states GROUP BY State_UnionTerritory;
```

3. The results of the above two queries are stored in csv files and then external tables created for them as confirmed (schema: State_UnionTerritory string, Confrmed int) and positive (schema: State_UnionTerritory string, Positives int), respectively.

4. Find the desired discrepancy by joining confirmed and positive on State_UnionTerritory:

```
SELECT c.State_UnionTerritory, abs(c.Confirmed-p.Positives) AS Discrepancy FROM confirmed c JOIN positive p ON (c.State_UnionTerritory = p.State_UnionTerritory);
```

D. Dynamic partition on State_UnionTerritory and cluster on Date using Covid19_India.csv:

1. Create the partitoned table:

```
CREATE TABLE covid_partSt_clustDt (Sno int, Date string, Cured int, Deaths int, Confirmed int) PARTITIONED BY (State_UnionTerritory string) CLUSTERED BY (Date) INTO 30 BUCKETS STORED AS SEQUENCEFILE;
```

2. We now fill data in this parititoned table:

Before filling the data, first execute each of these commands individually:

```
set hive.exec.dynamic.partition=true;
set hive.exec.dynamic.partition.mode=nonstrict;
set hive.exec.max.dynamic.partitions.pernode=1000;
set hive.enforce.bucketing = true;
```

```
INSERT OVERWRITE TABLE covid_partSt_clustDt PARTITION (State_UnionTerritory)
SELECT Sno, Date, Cured, Deaths, Confirmed, State_UnionTerritory FROM covid19;
```

3. The partitioned table is stored as part files in hdfs:

```
hdfs dfs -ls /user/hive/warehouse
```

4. Corresponding to each State/Union Territory there's a partition. To see all the data in the partition, for example, State_UnionTerritory=Tripura:

```
SELECT * from covid_partSt_clustDt WHERE State_UnionTerritory='Tripura';
```

5. To get data only from the 2nd bucket from all partitions:

```
SELECT * from covid_partSt_clustDt TABLESAMPLE(BUCKET 2 OUT OF 30 ON Date);
```

NOTE: The outputs of all these queries are submitted in appropriately named files.