**Problem Overview** – Assume that the data/tables are big and stored in an RDBMS. Running queries in RDBMS for analysis is taking too much time to get the results.

Aim is to use the Big data tools and technologies for the analysis of the two given datasets based on covid-19 spread in India and the Testing associated with it and try to get some insights. The data is available between 1st-April to 10th June duration.

**Detailed Problem** –

A. Need to create a data flow pipeline, where data initially resides in RDBMS and needs to be brought to Hive so as to get a consolidated view of covid cases details as well as testing details, taken together in one table.

B. Optimizations during sqoop import/export, hive optimizations have to be considered.

C. The data field is not in proper format and has to be taken care

D. Try to infer from the final consolidated table in Hive like whether there is any discrepancy between Number of confirmed covid cases in the state Vs. Number of Positive Samples collected during testing. Which state shows least discrepancy.

E. Run some more interesting queries from your end in hive to get more insights on the consolidated data. Sample Query – For every state, find the total number of confirmed cases reported and also total number of positive samples tested, in the entire duration of 2 months, starting with the state with the highest cases.

**Dataset** – Two csv files are provided –

1.Statewise Testing Details.csv - [StatewiseTestingDetails-201004-194827.csv](https://nexturntech.sharepoint.com/:x:/s/DataAvengers/EXVblYLylptCqurqtbPMtNEBfOPLHERe_dk1SVXf5BrPbw?e=ajGeMg)

2.Covid19\_india.csv - [Covid19\_india-201004-194827.csv](https://nexturntech.sharepoint.com/:x:/s/DataAvengers/ERkfh1sYFMFMhRkQiMt42pcB1yVodZFiwXbyOgOayNvdbQ?e=8GapEi)

StatewiseTestingDetails.csv: This dataset contains statewise, daily count of Testing performed in India.   
Columns: Seq, Date, State, Total Samples, Negative, Positive

Covid19\_india.csv: This dataset contains statewise, daily count of covid cases in India.   
Columns: Sno, Date, State/Union Territory, Cured, Deaths, Confirmed

Points to Note/Few Assumptions

1. We do not have data for every state in these two files. Also, we do not have data for every date in these two files. Some States and dates might be missing in these datasets.
2. Joining of two datasets might not be of much significance in this case but this is to check the understanding of hive and whether in real time you can work on this type of data flow involving hive and sqoop.
3. We assume that more data is added to both the tables on a frequent basis. So we need an incremental sqoop job
4. More queries on state and dates can come in future so, bucketing and partitioning to be used on these columns.
5. We assume that StatewiseTestingDetails.csv is the smaller dataset among the two.

We will be using a pseudo distributed environment (Cloudera Quickstart VM) which comes with pre-installed HDFS, Hive, sqoop etc.

**Steps Overview**:

Step1. Oracle VM Virtual box Installation to run the VM -  
Download Virtual Box using the link - <https://www.virtualbox.org/wiki/Downloads> and install

Step2. Cloudera QuickStart VM Installation –   
Download the zipped file from this link - <https://drive.google.com/drive/folders/1fXViofZOqdvX2bKY3RVwyTpCbgvFS7iI?usp=sharing>

Extract the files, and open .vbox file with Oracle Virtual Box.

Graphical user interface, application

Description automatically generated

Assign exactly half of the resources to this VM (in the image below I have allocated 8GB RAM and 2 CPU cores as I have 4 CPU cores and 16 GB RAM in my system)

Graphical user interface, text, application, email

Description automatically generated

Step3. DataPreprocessing – Data is assumed to be in Mysql initially. To mimic that –

* Copy the given csv files from local to hdfs.
* Creation of Mysql Tables -   
  Hint: Sno and Seq can be used as Primary Keys. Date can be var char type.
* Sqoop export of data from HDFS to Mysql.   
  Hint: Use Staging table while exporting
* Delete the data from hdfs post export

Now data resides in Mysql.

Step4. Sqoop Import- Bring the data from Mysql tables to HDFS   
Hint: Create sqoop jobs for importing both tables

Step5- Create Hive External Tables on top of data in HDFS.

Step6- Create Optimized External tables in Hive:   
Hint: File format – use ORC file format.   
Use Dynamic Partitioning on State Column, Bucketing on Date Column

Note: There might be frequent queries on State and Date, so we choose these columns for Partitioning and Bucketing, both tables can be Partitioned on State and Bucketed on Date column.

Step7: Load data to the optimized hive tables from normal hive tables.

Hint: Use INSERT OVERWRITE clause. Date has also to be formatted to proper hive format which is yyyy- mm-dd.

Step8- Inner Join two tables in Hive and get a consolidated table.   
Hint: Perform Map-side join of the two tables.   
Join columns can be ‘date’ and ‘state’ for better optimization. Here it is assumed that the State\_Testing table is small enough to fit in memory.

Step 9: Analyse the consolidated data by writing queries  
For Example –

Q1. Ideally the number of samples tested positive and number of covid cases confirmed must be the same. See which state/ states have more consistent data collection like The number of positive samples ( table1 ) match mostly with number of confirmed cases (table2), for which state.

Q2. For every state, find the total number of confirmed cases reported and also total number of positive samples tested, in the entire duration of 2 months, starting with the state with the highest cases.

Q3. How many confirmed cases are coming on a daily basis for each state?

You can run additional queries as per your understanding to get more interesting insights from this consolidated data

Step 10. Dump the consolidated data in Hbase for quick search using Hive-Hbase integrated tables.