# Big Bank, Big Problem

A leading banking and credit card services provider is trying to use Hadoop technologies to handle and analyze large amount of data.

Currently, the organization has data in RDBMS but wants to use hadoop ecosystem for storage, archival and analysis of large amount of data.

## Data Ingestion

Bring data from RDBMS to HDFS. This data import must be incremental and should happen every 2 minutes.

Following tables have to be imported:

|  |  |
| --- | --- |
| **Table** | **Incremental Column** |
| loan\_info | Loan\_id |
| credit\_card\_info | Cc\_number |
| shares\_info | Gmt\_timestamp |

All these data must be encrypted in HDFS.

HDFS data should be compressed to store less volume.

The sqoop password must also be encrypted.

## Table Details

### loan\_info

This table has following columns

|  |  |
| --- | --- |
| Loan\_id | Loan Identifier |
| User\_id | User Identifier |
| last\_payment\_date | Date on which last payment was made |
| payment\_installation | Amount payable in installation |
| Date\_payable | Date in month when payment is made by the user |

### credit\_card\_info

This table has following columns

|  |  |
| --- | --- |
| cc\_number | Credit Card number |
| user\_id | User Identifier |
| maximum\_credit | Maximum credit allowed by the user |
| outstanding\_balance | Current outstanding balance by the user |
| due\_date | Due date for the credit card payment |

### Shares\_info

This table has following information

|  |  |
| --- | --- |
| Share\_id | Share Identifier |
| Company\_name | Name of organisation |
| Gmt\_timestamp | Timestamp of the share price |
| Share\_price | Value of share |

## Analysis

1. Find out the list of users who have at least 2 loan instalments pending.

2. Find the list of users who have a healthy credit card but outstanding loan account.

Healthy credit card means no outstanding balance.

3. For every share and for every date, find the maximum profit one could have made on the share. Bear in mind that a share purchase must be before share sell and if share prices fall throughout the day, maximum possible profit may be negative.

## Archival

The organisation has lot of survey data scattered across different files in a directory in local file system. Provide a mechanism to effectively store the small files in Hadoop. It is expected to pack small files together before actually storing them in HDFS.

Survey files have below structure

|  |  |
| --- | --- |
| survey\_date | Date on which survey was conducted |
| survey\_question | Question asked in the survey |
| Rating | Rating received (1 - 5) |
| user\_id | User id who responded |
| survey\_id | Survey ID |

Following analysis is expected from survey files:

1. How many surveys got the average rating less than 3, provided at least 10 distinct users gave the rating?

2. Find the details of the survey which received the minimum rating. The condition is that the survey must have been rated by at least 20 users.

The organisation also has lots of emails stored in small files.

The metadata about the email is present in an XML file email\_metadata.xml

Read the XML file for email structure and pack all the email files in HDFS.

Following analysis is expected from the email data:

1. Which is the longest running email?

2. Find out the list of emails which were unanswered.

## Policy Setting

A policy file contains following fields in a CSV file:

|  |  |
| --- | --- |
| HDFS location | HDFS directory |
| HOT date | Date till which HDFS directory will be in HOT |
| WARM date | Date till which HDFS directory will be in WARM |
| COLD date | Date till which HDFS directory will be in COLD |

Run a job every day to make the policy over the HDFS locations to the new value if they are suitable for the same.