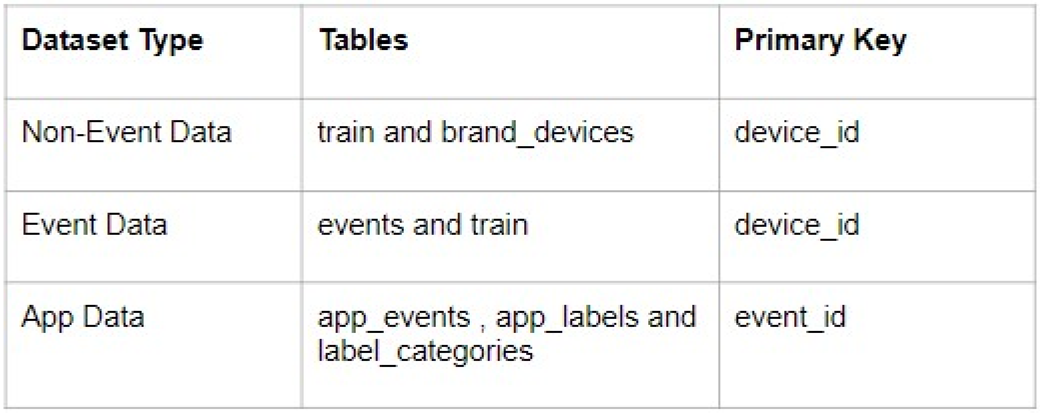
Prepare a script to create the data necessary for the next modeling exercises

Directive: You need to use the external Hive tables to categorize and create the data sets for your model-building activities. The tables, along with their primary keys for each table, are given below –



Dataset Type	Tables	Primary Key
Non-Event Data	train and brand_devices	device_id
Event Data	events and train	device_id
App Data	app_events , app_labels and label_categories	event_id

Note: Assume that the train data is your point of reference while creating event and non-event data by left join.

Hive external table creation and data loading for ‘non_event_data’ table:

As per the directive, we will first create the ‘non_event_data’ external table with all the columns from the ‘train’ and ‘brand_device’ tables along with specifying the PRIMARY KEY constraint on the device_id column –

```
0: jdbc:hive2://localhost:10000/default> create external table if not exists non_event_data (
. . . . . . . . . . . . . . . . . . . .> device_id string,
. . . . . . . . . . . . . . . . . . . .> gender string,
. . . . . . . . . . . . . . . . . . . .> age bigint,
. . . . . . . . . . . . . . . . . . . .> group_train string,
. . . . . . . . . . . . . . . . . . . .> phone_brand string,
. . . . . . . . . . . . . . . . . . . .> device_model string,
. . . . . . . . . . . . . . . . . . . .> PRIMARY KEY(device_id) DISABLE NOVALIDATE)
. . . . . . . . . . . . . . . . . . . .> row format delimited
. . . . . . . . . . . . . . . . . . . .> fields terminated by ','
. . . . . . . . . . . . . . . . . . . .> lines terminated by '\n'
. . . . . . . . . . . . . . . . . . . .> stored as textfile
. . . . . . . . . . . . . . . . . . . .> location '/user/hadoop/telco/non_event_data/';
No rows affected (2.796 seconds)
```

Next we will load the data into this table with the result set of a select query using a LEFT OUTER JOIN between the ‘train’ and ‘brand_device’ tables joined ON their common device_id column –

```
0: jdbc:hive2://localhost:10000/default> insert into non_event_data
. . . . . . . . . . . . . . . . . . . .> select t1.device_id, t1.gender, t1.age, t1.group_train,
. . . . . . . . . . . . . . . . . . . .> t2.phone_brand, t2.device_model
. . . . . . . . . . . . . . . . . . . .> from train t1 LEFT OUTER JOIN brand_device t2
. . . . . . . . . . . . . . . . . . . .> ON (t1.device_id = t2.device_id);
No rows affected (25.573 seconds)
```

We will check the final count of the inserted data and also verify the data for the first 5 rows –

```
0: jdbc:hive2://localhost:10000/default> select count(*) from non_event_data;

+--------+
|  _c0   |
+--------+
| 74645  |
+--------+
1 row selected (6.635 seconds)


0: jdbc:hive2://localhost:10000/default> select * from non_event_data limit 5;

+-----------------------+---------+------+--------------+--------------+-----------------+
|       device_id       | gender  | age  | group_train  | phone_brand  |  device_model   |
+-----------------------+---------+------+--------------+--------------+-----------------+
| -1819925713085810000  | F       | 23   | F0-24        | OPPO         | N1 Mini         |
| 3670076507269740000   | M       | 33   | M32+         | Meizu        | menote1 2       |
| 5333872006968810000   | M       | 34   | M32+         | Xiaomi       | xnote           |
| 4216041491117040000   | M       | 60   | M32+         | lshi         | ihv1            |
| -3441149835823130000  | M       | 30   | M25-32       | Huawei       | è�£è€€ç•…çŽ©5X  |
+-----------------------+---------+------+--------------+--------------+-----------------+
5 rows selected (0.373 seconds)
```

Hive external table creation and data loading for ‘event_data’ table:

As per the directive, we will create the ‘event_data’ external table with all the columns from the ‘train’ and ‘events’ tables along with specifying the PRIMARY KEY constraint on the device_id column –

```
0: jdbc:hive2://localhost:10000/default> create external table if not exists event_data (
. . . . . . . . . . . . . . . . . . . .> device_id string,
. . . . . . . . . . . . . . . . . . . .> gender string,
. . . . . . . . . . . . . . . . . . . .> age bigint,
. . . . . . . . . . . . . . . . . . . .> group_train string,
. . . . . . . . . . . . . . . . . . . .> event_id bigint,
. . . . . . . . . . . . . . . . . . . .> event_time string,
. . . . . . . . . . . . . . . . . . . .> longitude string,
. . . . . . . . . . . . . . . . . . . .> latitude string,
. . . . . . . . . . . . . . . . . . . .> PRIMARY KEY(device_id) DISABLE NOVALIDATE)
. . . . . . . . . . . . . . . . . . . .> row format delimited
. . . . . . . . . . . . . . . . . . . .> fields terminated by ','
. . . . . . . . . . . . . . . . . . . .> lines terminated by '\n'
. . . . . . . . . . . . . . . . . . . .> stored as textfile
. . . . . . . . . . . . . . . . . . . .> location '/user/hadoop/telco/event_data/';
No rows affected (0.097 seconds)
```

Next we will load the data into this table with the result set of a select query using a LEFT OUTER JOIN between the ‘train’ and ‘events’ tables joined ON their common device_id column –

```
0: jdbc:hive2://localhost:10000/default> insert into event_data
. . . . . . . . . . . . . . . . . . . .> select t1.device_id, t1.gender, t1.age, t1.group_train,
. . . . . . . . . . . . . . . . . . . .> t2.event_id, t2.event_time, t2.longitude, t2.latitude
. . . . . . . . . . . . . . . . . . . .> from train t1 LEFT OUTER JOIN events t2
. . . . . . . . . . . . . . . . . . . .> ON (t1.device_id = t2.device_id);
No rows affected (41.572 seconds)
```

We will check the final count of the inserted data and also verify the data for the first 5 rows –

```
0: jdbc:hive2://localhost:10000/default> select count(*) from event_data;
```

```
+----------+
|   _c0    |
+----------+
| 1266933  |
+----------+
1 row selected (11.087 seconds)


0: jdbc:hive2://localhost:10000/default> select * from event_data limit 5;

+----------------------+---------+------+--------------+-----------+-----------------------+------------+-----------+
|      device_id       | gender  | age  | group_train  | event_id  |      event_time       | longitude  | latitude  |
+----------------------+---------+------+--------------+-----------+-----------------------+------------+-----------+
| -1000369272589010000 | F       | 26   | F25-32       | NULL      | NULL                  | NULL       | NULL      |
| -1000572055892390000 | F       | 27   | F25-32       | NULL      | NULL                  | NULL       | NULL      |
| -1000643208750510000 | M       | 29   | M25-32       | NULL      | NULL                  | NULL       | NULL      |
| -1001337759327040000 | M       | 30   | M25-32       | 2774404   | 2016-05-07 09:14:24.0 | 119.61     | 29.7      |
| -1001337759327040000 | M       | 30   | M25-32       | 3065018   | 2016-05-04 10:26:14.0 | 120.29     | 30.42     |
+----------------------+---------+------+--------------+-----------+-----------------------+------------+-----------+
5 rows selected (0.156 seconds)
```

Hive external table creation and data loading for ‘app_data’ table:

As per the directive, we will create the ‘app_data’ external table with all the columns from the ‘app_events’, ‘app_labels’ and ‘label_categories’ tables along with specifying the PRIMARY KEY constraint on the event_id column –

```
0: jdbc:hive2://localhost:10000/default> create external table if not exists app_data (
. . . . . . . . . . . . . . . . . . . .> event_id string,
. . . . . . . . . . . . . . . . . . . .> app_id string,
. . . . . . . . . . . . . . . . . . . .> is_installed bigint,
. . . . . . . . . . . . . . . . . . . .> is_active bigint,
. . . . . . . . . . . . . . . . . . . .> label_id bigint,
. . . . . . . . . . . . . . . . . . . .> category string,
. . . . . . . . . . . . . . . . . . . .> PRIMARY KEY(event_id) DISABLE NOVALIDATE)
. . . . . . . . . . . . . . . . . . . .> row format delimited
. . . . . . . . . . . . . . . . . . . .> fields terminated by ','
. . . . . . . . . . . . . . . . . . . .> lines terminated by '\n'
. . . . . . . . . . . . . . . . . . . .> stored as textfile
. . . . . . . . . . . . . . . . . . . .> location '/user/hadoop/telco/app_data/';
No rows affected (0.085 seconds)
```

Next we will load the data into this table with the result set of a select query using a LEFT OUTER JOIN between the ‘app_events’ and ‘app_labels’ tables joined ON their common app_id column, along with a LEFT OUTER JOIN between the ‘app_labels’ and ‘label_categories’ tables joined ON their common label_id column –

```
0: jdbc:hive2://localhost:10000/default> insert into app_data
. . . . . . . . . . . . . . . . . . . .> select t1.event_id, t1.app_id, t1.is_installed, t1.is_active,
. . . . . . . . . . . . . . . . . . . .> t2.label_id, t3.category from app_events t1
. . . . . . . . . . . . . . . . . . . .> LEFT OUTER JOIN app_labels t2 ON (t1.app_id = t2.app_id)
. . . . . . . . . . . . . . . . . . . .> LEFT OUTER JOIN label_categories t3 ON (t2.label_id = t3.label_id);
No rows affected (1228.74 seconds)
```

We will check the final count of the inserted data and also verify the data for the first 5 rows –

```
0: jdbc:hive2://localhost:10000/default> select count(*) from app_data;

+------------+
|    _c0     |
+------------+
| 209355710  |
+------------+
1 row selected (206.104 seconds)


0: jdbc:hive2://localhost:10000/default> select * from app_data limit 5;

+-----------+-----------------------+--------------+------------+-----------+-------------------------+
| event_id  |        app_id         | is_installed | is_active  | label_id  |        category         |
+-----------+-----------------------+--------------+------------+-----------+-------------------------+
| 3231904   | -1000044012126765960  | 1            | 0          | 810       | Casual puzzle categories |
| 3231904   | -1000044012126765960  | 1            | 0          | 405       | Custom label            |
| 3231904   | -1000044012126765960  | 1            | 0          | 795       | game                    |
| 3069897   | -1000044012126765960  | 1            | 0          | 810       | Casual puzzle categories |
| 3069897   | -1000044012126765960  | 1            | 0          | 405       | Custom label            |
+-----------+-----------------------+--------------+------------+-----------+-------------------------+
5 rows selected (0.224 seconds)
```

Shape of the final datasets:

Dataset	Rows or Observations count	Columns or Features count
non_event_data	**74645**	**6**
event_data	**1266933**	**8**
app_data	**209355710**	**6**

Exporting the datasets to S3:

We will first verify the dataset details in Hadoop –

```
[hadoop@ip-172-31-81-78 ~]$ hadoop fs -ls /user/hadoop/telco/* | grep "_data"

-rwxr-xr-x   1 hadoop hadoop 2157190341 2021-09-29 09:33 /user/hadoop/telco/app_data/000000_0
-rwxr-xr-x   1 hadoop hadoop 1738942712 2021-09-29 09:31 /user/hadoop/telco/app_data/000001_0
-rwxr-xr-x   1 hadoop hadoop 1904284276 2021-09-29 09:31 /user/hadoop/telco/app_data/000002_0
-rwxr-xr-x   1 hadoop hadoop 1976829842 2021-09-29 09:39 /user/hadoop/telco/app_data/000003_0
-rwxr-xr-x   1 hadoop hadoop 2622831455 2021-09-29 09:40 /user/hadoop/telco/app_data/000004_0

-rwxr-xr-x   1 hadoop hadoop   88454657 2021-09-29 09:17 /user/hadoop/telco/event_data/000000_0

-rwxr-xr-x   1 hadoop hadoop    3539045 2021-09-29 09:16 /user/hadoop/telco/non_event_data/000000_0
```

We can see that the 'non_event_data' and 'event_data' datasets are available in single partitions, while the 'app_data' dataset is split across 5 partitions. Accordingly we will proceed with exporting the datasets from Hadoop to a new **public S3 location s3://upgrad-capstone-mlc/mid-submission/** –

```
hadoop distcp /user/hadoop/telco/non_event_data/000000_0 s3://upgrad-capstone-mlc/mid-submission/non_event_data.csv

        Map-Reduce Framework
                ...
                GC time elapsed (ms)=197
                CPU time spent (ms)=8460
                Physical memory (bytes) snapshot=403492864
                Virtual memory (bytes) snapshot=3292930048
                Total committed heap usage (bytes)=378011648
        DistCp Counters
                Bytes Copied=3539045
                Bytes Expected=3539045
                Files Copied=1
```

```
hadoop distcp /user/hadoop/telco/event_data/000000_0 s3://upgrad-capstone-mlc/mid-submission/event_data.csv

        Map-Reduce Framework
                ...
                GC time elapsed (ms)=204
                CPU time spent (ms)=10840
                Physical memory (bytes) snapshot=391241728
                Virtual memory (bytes) snapshot=3287490560
                Total committed heap usage (bytes)=305135616
        DistCp Counters
                Bytes Copied=88454657
                Bytes Expected=88454657
                Files Copied=1
```

```
hadoop distcp /user/hadoop/telco/app_data/* s3://upgrad-capstone-mlc/mid-submission/app_data/

        Map-Reduce Framework
                ...
                GC time elapsed (ms)=3202
                CPU time spent (ms)=231220
                Physical memory (bytes) snapshot=2432987136
                Virtual memory (bytes) snapshot=16498540544
                Total committed heap usage (bytes)=2024275968
        DistCp Counters
                Bytes Copied=10400078626
                Bytes Expected=10400078626
                Files Copied=5
```

Verifying the datasets in S3:

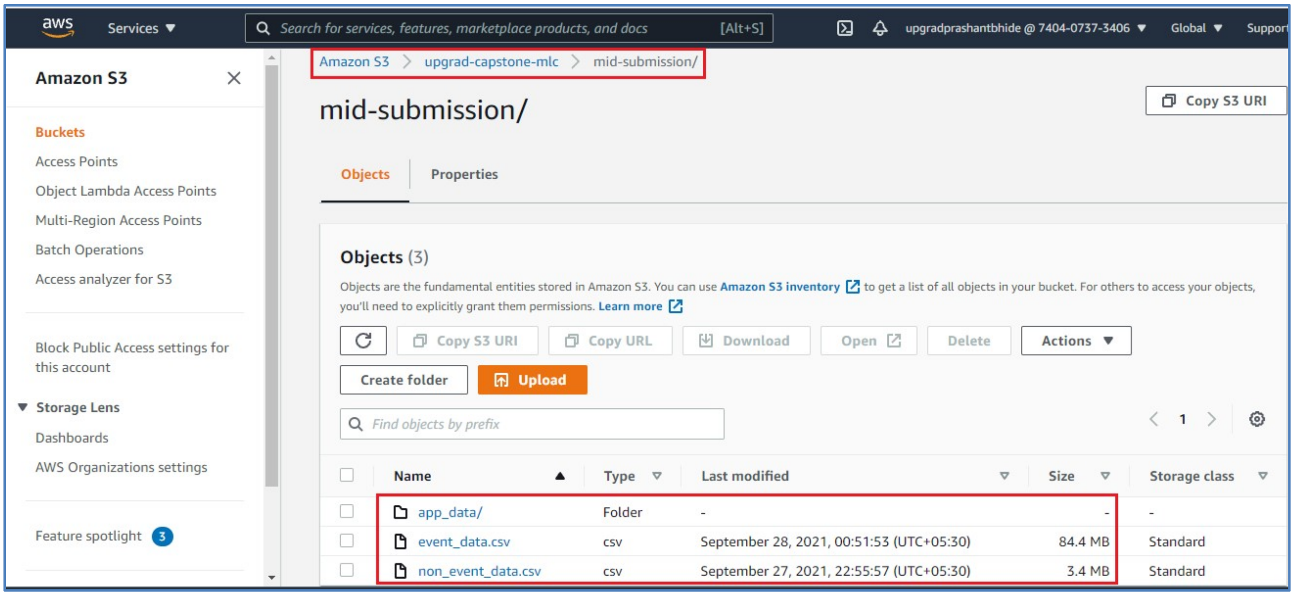
We will finally verify the exported dataset details in S3 –

```
hadoop fs -ls s3://upgrad-capstone-mlc/mid-submission/*

-rw-rw-rw-   1 hadoop hadoop 2157190341 2021-09-27 19:29 s3://upgrad-capstone-mlc/mid-submission/app_data/000000_0
-rw-rw-rw-   1 hadoop hadoop 1738942712 2021-09-27 19:27 s3://upgrad-capstone-mlc/mid-submission/app_data/000001_0
-rw-rw-rw-   1 hadoop hadoop 1904284276 2021-09-27 19:27 s3://upgrad-capstone-mlc/mid-submission/app_data/000002_0
-rw-rw-rw-   1 hadoop hadoop 1976829842 2021-09-27 19:27 s3://upgrad-capstone-mlc/mid-submission/app_data/000003_0
-rw-rw-rw-   1 hadoop hadoop 2622831455 2021-09-27 19:29 s3://upgrad-capstone-mlc/mid-submission/app_data/000004_0

-rw-rw-rw-   1 hadoop hadoop   88454657 2021-09-27 19:21 s3://upgrad-capstone-mlc/mid-submission/event_data.csv

-rw-rw-rw-   1 hadoop hadoop    3539045 2021-09-27 17:25 s3://upgrad-capstone-mlc/mid-submission/non_event_data.csv
```



Amazon S3 > upgrad-capstone-mlc > mid-submission/

mid-submission/

Objects (3)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	app_data/	Folder	-	-	-
□	event_data.csv	csv	September 28, 2021, 00:51:53 (UTC+05:30)	84.4 MB	Standard
□	non_event_data.csv	csv	September 27, 2021, 22:55:57 (UTC+05:30)	3.4 MB	Standard