Homework4

Step1: enter into hive, create a new database called homework4, grant permission to mysql homework4 database.

```
[root@sandbox ~]# hive
hive> create database homework4;
OK
Time taken: 3.397 seconds
hive> show databases;
OK
default
foodmart
homework4
lab_demo
sqoop_examples
xademo
Time taken: 0.016 seconds, Fetched: 6 row(s)
mysql> grant all privileges on homework4.* to ''@'localhost';
Query OK, 0 rows affected (0.22 sec)
```

Step2: use sqoop to export both tables into hive, creating 2 jobs and execute jobs.

```
[root@sandbox ~]# sqoop job --create importAccount -- import --connect
jdbc:mysql://localhost/homework4 --driver com.mysql.jdbc.Driver --table accounts --hive-import
--hive-table homework4.accounts -m 1

[root@sandbox ~]# sqoop job --create importContacts -- import --connect
jdbc:mysql://localhost/homework4 --driver com.mysql.jdbc.Driver --table contacts --hive-import
--hive-table homework4.contacts -m 1

[root@sandbox ~]# sqoop job --exec importAccount
[root@sandbox ~]# sqoop job --exec importContacts
```

Step3: Write java code for masking email address and phone number

```
package UDF;

import org.apache.hadoop.hive.ql.exec.UDF;
import org.apache.hadoop.io.Text;

public class Mask extends UDF {
   public Text evaluate(Text column) {
      if (column.toString().contains("@")) {
```

```
            String trailing = column.toString().split("@")[1];
            return new Text("XXX@" + trailing);
        } else {
            String ending = column.toString().split("-")[2];
            return new Text("XXX-XXX-" + ending);
        }
    }
}
```

```
hive> add jar UDF.jar;
hive> create temporary function masking as 'UDF.Mask';
```

Step4: Transform 2 tables by applying masking method on phone and email columns of two tables

```
hive> create table encrypted_accounts(
    > id INT,
    > name STRING,
    > phone_encrypted STRING)
    > row format delimited
    > fields terminated by '\t'
    > lines terminated by '\n';

hive> insert overwrite table encrypted_account
    > select id, name, masking(phone) as phone_encrypted from accounts;

hive> create table encrypted_contacts(
    >  id INT, account_id INT, first_name STRING, last_name STRING, phone_encrypted
STRING, email_encrypted STRING)
    > row format delimited
    > fields terminated by '\t'
    > lines terminated by '\n';

hive> insert overwrite table encrypted_contacts
    > select id, account_id, first_name, last_name, masking(phone) as phone_encrypted,
masking(email) as email_encrypted from contacts;
```

Step5: export 2 newly created tables into mysql homework4 database using sqoop

```
mysql> create table encrypted_accounts(
    -> id INTEGER NOT NULL,
    -> name VARCHAR(50),
    -> encrypted_phone CHAR(12),
```

```
   -> PRIMARY KEY(id)
   -> );
```

```
mysql> create table encrypted_contacts ( id INTEGER NOT NULL, account_id INTEGER,
first_name VARCHAR(50), last_name VARCHAR(50), encrypted_phone CHAR(12),
encrypted_email VARCHAR(50), PRIMARY KEY(id));
```

```
[root@sandbox ~]# sqoop export --connect jdbc:mysql://localhost/homework4 --driver
com.mysql.jdbc.Driver --table encrypted_accounts --export-dir
/apps/hive/warehouse/homework4.db/encrypted_accounts --input-fields-terminated-by '\t' -m 1;
```

```
[root@sandbox ~]# sqoop export --connect jdbc:mysql://localhost/homework4 --driver
com.mysql.jdbc.Driver --table encrypted_contacts --export-dir
/apps/hive/warehouse/homework4.db/encrypted_contacts --input-fields-terminated-by '\t' -m 1;
```

SAMPLE OUTPUT

```
mysql> use homework4;
mysql> show tables;
+--------------------+
| Tables_in_homework4 |
+--------------------+
| accounts           |
| contacts           |
| encrypted_accounts  |
| encrypted_contacts  |
+--------------------+
```

```
mysql> select * from encrypted_accounts;
+----+-----------------------+-----------------+
| id | name                  | encrypted_phone |
+----+-----------------------+-----------------+
|  1 | Small Company LLC     | XXX-XXX-2222    |
|  2 | Medium Company LLC    | XXX-XXX-2222    |
|  3 | Large Company LLC     | XXX-XXX-2222    |
|  4 | Very Large Company LLC | XXX-XXX-2222    |
+----+-----------------------+-----------------+
```

```
mysql> select * from encrypted_contacts;
+----+------------+------------+-----------+-----------------+-----------------+
| id | account_id | first_name | last_name | encrypted_phone | encrypted_email |
+----+------------+------------+-----------+-----------------+-----------------+
| 11 |          1 | John       | Smith     | XXX-XXX-2222    | XXX@gmail.com   |
| 12 |          2 | Bob        | Smith     | XXX-XXX-3334    | XXX@gmail.com   |
| 13 |          3 | Mark       | Taylor    | XXX-XXX-3335    | XXX@gmail.com   |
| 14 |          4 | Pat        | Taylor    | XXX-XXX-3336    | XXX@gmail.com   |
+----+------------+------------+-----------+-----------------+-----------------+
```