HBASE & PHOENIX INSTALLATION & WORKOUTS

1) **Go to the below path.**

```
cd /home/hduser/install/
```

2) Extract the tarball

HBase installation

```
tar xvzf hbase-0.98.4-hadoop2-bin.tar.gz
sudo mv hbase-0.98.4-hadoop2 /usr/local/hbase
sudo chown -R hduser:hadoop /usr/local/hbase
```

Zookeeper Installation

```
tar xvzf zookeeper-3.4.6.tar.gz
sudo mv zookeeper-3.4.6 /usr/local/zookeeper
sudo chown -R hduser:hadoop /usr/local/zookeeper
```

Phoenix Installation

```
cd /home/hduser/install
tar xvzf apache-phoenix-4.8.0-HBase-0.98-bin.tar.gz
sudo mv apache-phoenix-4.8.0-HBase-0.98-bin /usr/local/phoenix
sudo chown -R hduser:hadoop /usr/local/phoenix
```

3) Zookeeper config

```
cd /usr/local/zookeeper/conf
mv zoo_sample.cfg zoo.cfg
```

```
sed -i 's/dataDir=\/tmp\/zookeeper/dataDir=\/usr\/local\/zookeeper\/data/g' zoo.cfg
```

```
mkdir /usr/local/zookeeper/data
```

4) Start Zookeeper by running below command:

```
cd /usr/local/zookeeper/bin
./zkServer.sh start
```

5) Edit HBase environment script

```
cd /usr/local/hbase/conf
echo 'export HBASE_MANAGES_ZK=false' >> hbase-env.sh
echo 'export HBASE_CLASSPATH=/usr/local/hbase/lib' >> hbase-env.sh
```

```
cd /usr/local/phoenix/

cp phoenix-hive-4.8.0-HBase-0.98.jar /usr/local/hbase/lib/
cp phoenix-4.8.0-HBase-0.98-client.jar /usr/local/hbase/lib/
~~cp phoenix-4.6.0-HBase-0.98-client-minimal.jar /usr/local/hbase/lib/~~
~~cp phoenix-core-4.6.0-HBase-0.98.jar /usr/local/hbase/lib/~~
```

Hive-HBase Handler

Add the below line in hive-env.sh to locate hbase lib path as auxiliary hive jar path to use hbase jars

```
cd /usr/local/hive/conf/
mv hive-env.sh.template hive-env.sh
echo "export HIVE_AUX_JARS_PATH=/usr/local/hbase/lib" >> hive-env.sh
```

Copy the jars to the hive lib directory from hbase

```
cp /usr/local/hbase/lib/hbase-common-0.98.4-hadoop2.jar /usr/local/hive/lib/
cp /usr/local/hbase/lib/zookeeper-3.4.6.jar /usr/local/hive/lib/
cp /usr/local/hbase/lib/guava-12.0.1.jar /usr/local/hive/lib/
cp /usr/local/hbase/lib/hbase-protocol-0.98.4-hadoop2.jar /usr/local/hive/lib/
cp /usr/local/hbase/lib/hbase-server-0.98.4-hadoop2.jar /usr/local/hive/lib/
cp /home/hduser/install/antlr-runtime-3.5.2.jar /usr/local/hive/lib/
cp /home/hduser/install/phoenix-hive-4.8.0-HBase-0.98.jar /usr/local/hive/lib/
cp /home/hduser/install/twill-discovery-api-0.14.0.jar /usr/local/hive/lib/
cp /home/hduser/install/twill-zookeeper-0.14.0.jar /usr/local/hive/lib/
```

6) Edit the hbase-site.xml to set hbase distribution, hbase root data dir, zookeeper quorum and zk port.

```
vi /usr/local/hbase/conf/hbase-site.xml
<configuration>
<property>
<name>hbase.cluster.distributed</name>
<value>true</value>
</property>
<property>
<name>hbase.rootdir</name>
<value>hdfs://localhost:54310/user/hduser/hbase</value>
</property>
<property>
<name>hbase.zookeeper.quorum</name>
<value>localhost</value>
</property>
<property>
<name>hbase.zookeeper.property.clientPort</name>
<value>2181</value>
</property>
<property>
```

```
<name>hbase.zookeeper.property.dataDir</name>
<value>/usr/local/zookeeper/data</value>
</property>
<property>
<name>hbase.regionserver.wal.codec</name>
<value>org.apache.hadoop.hbase.regionserver.wal.IndexedWALEditCodec</value>
</property>
</configuration>
```

7) **Once completed the above steps start the hbase daemon**

```
start-hbase.sh
```

Type jps and see if zookeeper and hbase is running

8) **To get into the hbase interactive shell type the below command**

```
hbase shell
```

################################ HBASE WORKOUTS ################################

Type 'list' to see if hbase is working properly

```
list
```

####### Creating a table "Patient" with the column Families (Personal and Medical) #######

```
create 'Patient','Personal','Medical'

create 'Patient1',{NAME => 'Personal', VERSIONS => 3}
```

####### Inserting a record into the table#######

```
put 'Patient','1','Personal:pname','Alan'
put 'Patient','2','Personal:pname','Bill'
put  'Patient','2','Personal:filenum','100'
put 'Patient','3','Personal:pname','Willy'
put 'Patient','4','Personal:pname','Dave'
put 'Patient','5','Personal:pname','Alex'
put 'Patient','2','Personal:age','24'
put 'Patient','105','Personal:pname','Alex'
put 'Patient','202','Personal:age','44'
put 'Patient','202','Personal:filenum','101'
put 'Patient','202','Personal:addr','3, first ave,NJ'
put 'Patient','1','Medical:history','Anemic'
put 'Patient','105','Medical:history','General check'
put 'Patient','102','Medical:history','Arthritis'
put 'Patient','102','Medical:oldhistory','Ostophenia'
```

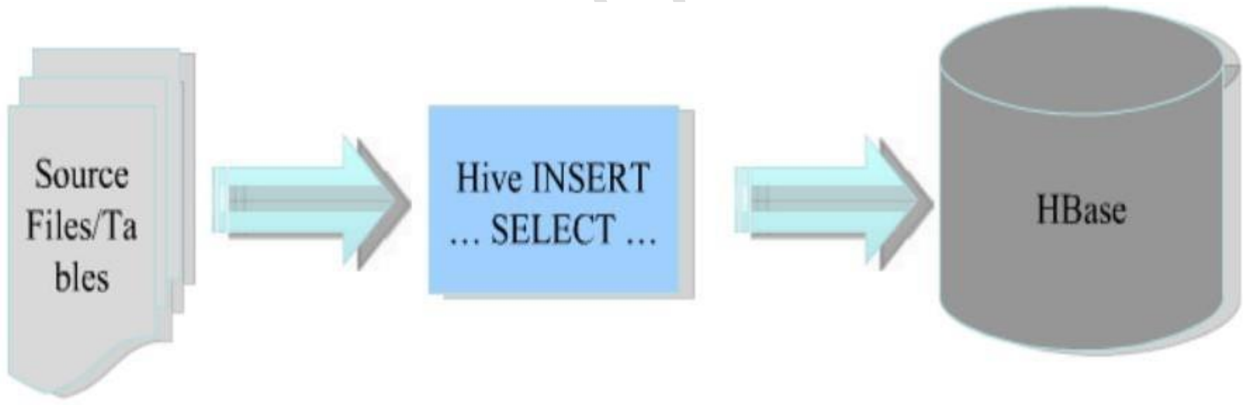
#####Check whether the below put works with the column family used as Medical1 instead of Medical###

```
put 'Patient','102','Medical1:oldhistory','Ostophenia'

####### Scan/select all the data from the table#######

scan 'Patient'

####### Scan/select with rowkey#######

get 'Patient','2'
```

Sqoop Import Data Into HBase

Sqoop supports additional import targets beyond HDFS and Hive. Sqoop can also import records into a table in HBase. By specifying --hbase-table, you instruct Sqoop to import to a table in HBase rather than a directory in HDFS. Sqoop will import data to the table specified as the argument to --hbase-table. Each row of the input table will be transformed into an HBase Put operation to a row of the output table. The key for each row is taken from a column of the input. By default Sqoop will use the split-by column as the row key column. If that is not specified, it will try to identify the primary key column, if any, of the source table. You can manually specify the row key column with --hbase-row-key. Each output column will be placed in the same column family, which must be specified with --column-family.

```
sqoop import --connect jdbc:mysql://localhost/custdb --username root --password root --table customer \
--hbase-table customer --hbase-create-table --hbase-bulkload --column-family custinfo \
--hbase-row-key custid -m 3 --split-by custid --delete-target-dir
```



```
hive --service metastore

hive

create database if not exists retail;
use retail;

create table customer_info(custno string, firstname string, lastname string, age int,profession
string)
row format delimited fields terminated by ',';

load data local inpath '/home/hduser/hive/data/custs' into table customer_info;

CREATE TABLE custinfo_hive (custno int, firstname string, lastname string,age int, profession string)
```

```
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
WITH SERDEPROPERTIES ("hbase.columns.mapping" =
":key,custbase:firstname,custbase:lastname,custbase:age,custaddon:profession")
TBLPROPERTIES ("hbase.table.name" = "custinfo_hbase");

insert into custinfo_hive select * from customer_info;
```

Composite Key insertion:

```
 CREATE TABLE custinfo_hive_complex ( fullname string,custno int,compositename struct
 <firstname:string,lastname:string>,age int, profession string)
 STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
 WITH SERDEPROPERTIES ("hbase.columns.mapping" =
 ":key,custbase:custno,custbase:compositename,custbase:age,custaddon:profession")
 TBLPROPERTIES ("hbase.table.name" = "custinfo_hbase_composite");

 insert into custinfo_hive_complex select concat(firstname,' ',lastname) as
 fullname,custno,NAMED_STRUCT('firstname',firstname,'lastname',lastname) as compositename,
   age,profession from customer_info;
```

####### Retrieve more versions#######

```
 alter 'Patient',{NAME=>'Medical',VERSIONS=>1}
 put 'Patient','1','Medical:history','Hyper Tension'
 put 'Patient','1','Medical:history','General check'

 scan 'Patient',{VERSIONS => 3}
 alter 'Patient',{NAME=>'Medical',VERSIONS=>3}
 put 'Patient','1','Medical:history','Dental'
 scan 'Patient',{VERSIONS => 3}
```

#######List only the latest version#######

```
 get 'Patient','1'
 put 'Patient','1','Medical:history','Flu'
 scan 'Patient',{VERSIONS => 6}
```

####### delete a specific column for A rowkey #######

```
 delete 'Patient','2','Personal:pname'
```

####### delete entire rowkey details#######

```
 deleteall 'Patient','1'
```

####### Describe the table#######

```
 describe 'Patient'
```

####### Table columns #######

```
echo "scan 'Patient'" | hbase shell | awk -F'=' '{print $2}' | awk -F ':' '{print $2}'|awk -F ',' '{print $1}'
```

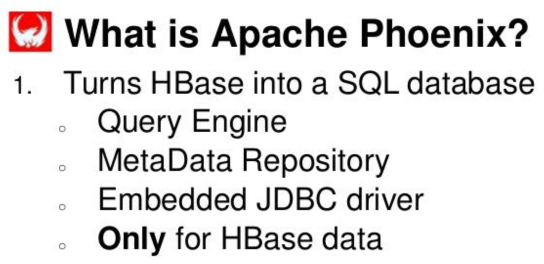
####### Add column family #######

```
alter 'Patient',{NAME=>'Drugs'}
```

####### drop the column family from the table#######

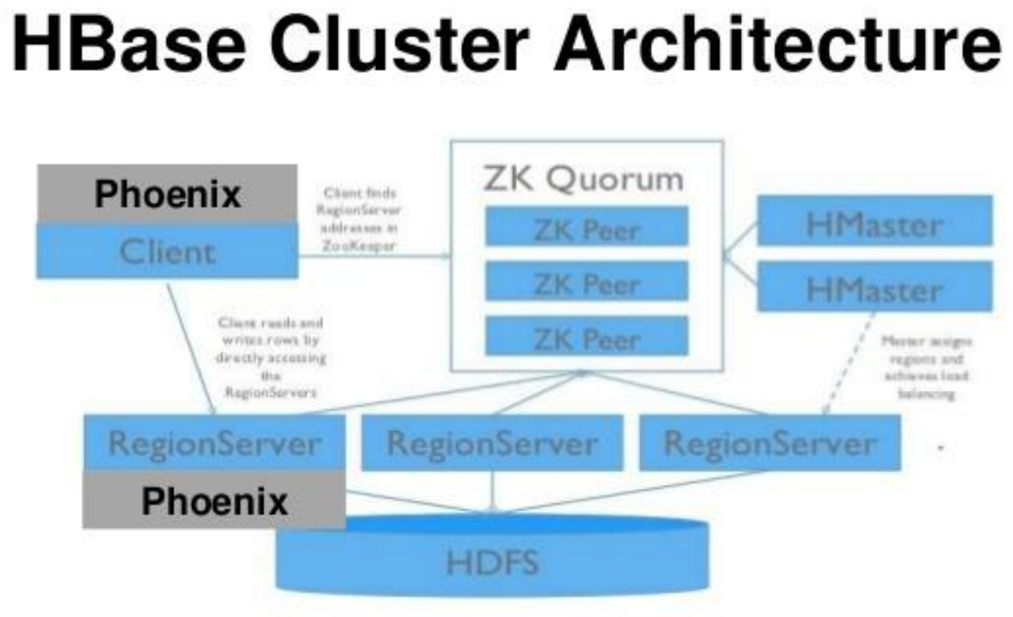
```
alter 'Patient',{NAME=>'Medical',METHOD=>'delete'}

scan 'Patient', {TIMERANGE => [1303668804, 1303668904]}

alter 'Patient', CONFIGURATION => {NAME => 'Personal', BLOCKCACHE => 'false'}
```

Apache Phoenix Workouts:



What is Apache Phoenix?

1. Turns HBase into a SQL database
    - Query Engine
    - MetaData Repository
    - Embedded JDBC driver
    - **Only** for HBase data


HBase Cluster Architecture
Phoenix
Client
Client finds RegionServer addresses in ZooKeeper
ZK Quorum
ZK Peer
ZK Peer
ZK Peer
HMaster
HMaster
Client reads and writes rows by directly accessing the RegionServers
Master assigns regions and achieves load balancing
RegionServer
RegionServer
RegionServer
Phoenix
HDFS

Phoenix Workouts:

sqlline.py localhost
!set maxwidth 800
!set maxcolumnwidth 20

Create table if not exists customer (custid integer not null primary key, firstname varchar,lastname varchar,age integer,profession varchar);

UPSERT INTO customer VALUES (4000001,'Kristina','Chung',74,'Pilot');
UPSERT INTO customer VALUES (4000002,'Paige','Chen',42,'Teacher');
UPSERT INTO customer VALUES (4000003,'Sherri','Melton',43,'Firefighter');
UPSERT INTO customer VALUES (4000004,'Gretchen','Hill',63,'Computer hardware engineer');
UPSERT INTO customer VALUES (4000005,'Karen','Puckett',39,'Lawyer');
UPSERT INTO customer VALUES (4000006,'Patrick','Song',60,'Veterinarian');
UPSERT INTO customer VALUES (4000007,'Elsie','Hamilton',47,'Pilot');
UPSERT INTO customer VALUES (4000008,'Hazel','Bender',26,'Carpenter');
UPSERT INTO customer VALUES (4000009,'Malcolm','Wagner',41,'Artist');
UPSERT INTO customer VALUES (4000010,'Dolores','McLaughlin',65,'Writer');
UPSERT INTO customer VALUES (4000011,'Francis','McNamara',49,'Therapist');
UPSERT INTO customer VALUES (4000012,'Sandy','Raynor',52,'Writer');
UPSERT INTO customer VALUES (4000013,'Marion','Moon',72,'Carpenter');
UPSERT INTO customer VALUES (4000014,'Beth','Woodard',45,'');
UPSERT INTO customer VALUES (4000015,'Julia','Desai',63,'Musician');
UPSERT INTO customer VALUES (4000016,'Jerome','Wallace',67,'Pharmacist');
UPSERT INTO customer VALUES (4000017,'Neal','Lawrence',39,'Computer support specialist');
UPSERT INTO customer VALUES (4000018,'Jean','Griffin',66,'Childcare worker');
UPSERT INTO customer VALUES (4000019,'Kristine','Dougherty',42,'Financial analyst');

```
UPSERT INTO customer VALUES (4000020,'Crystal','Powers',28,'Engineering technician');
UPSERT INTO customer VALUES (4000021,'Alex','May',42,'Environmental scientist');
UPSERT INTO customer VALUES (4000022,'Eric','Steele',27,'Doctor');
UPSERT INTO customer VALUES (4000023,'Wesley','Teague',64,'Carpenter');
UPSERT INTO customer VALUES (4000024,'Franklin','Vick',45,'Dancer');
UPSERT INTO customer VALUES (4000025,'Claire','Gallagher',40,'Musician');
UPSERT INTO customer VALUES (4000026,'Marian','Solomon',50,'Lawyer');
UPSERT INTO customer VALUES (4000027,'Marcia','Walsh',59,'Accountant');
UPSERT INTO customer VALUES (4000028,'Dwight','Monroe',24,'Economist');
UPSERT INTO customer VALUES (4000029,'Wayne','Connolly',58,'Real estate agent');
UPSERT INTO customer VALUES (4000030,'Stephanie','Hawkins',38,'Human resources assistant');
UPSERT INTO customer VALUES (4000031,'Neal','Middleton',25,'Civil engineer');
UPSERT INTO customer VALUES (4000032,'Gretchen','Goldstein',35,'Engineering technician');
UPSERT INTO customer VALUES (4000033,'Tim','Watts',27,'Lawyer');
UPSERT INTO customer VALUES (4000034,'Jerome','Johnston',73,'Childcare worker');
UPSERT INTO customer VALUES (4000035,'Shelley','Weeks',33,'Reporter');
UPSERT INTO customer VALUES (4000036,'Priscilla','Wilkerson',27,'Agricultural and food scientist');
UPSERT INTO customer VALUES (4000037,'Elsie','Barton',65,'Childcare worker');
UPSERT INTO customer VALUES (4000038,'Beth','Walton',44,'Firefighter');
UPSERT INTO customer VALUES (4000039,'Erica','Hall',63,'Police officer');
UPSERT INTO customer VALUES (4000040,'Douglas','Ross',67,'Secretary');
UPSERT INTO customer VALUES (4000041,'Donald','Chung',44,'Computer hardware engineer');
UPSERT INTO customer VALUES (4000042,'Katherine','Bender',31,'Physicist');
UPSERT INTO customer VALUES (4000043,'Paul','Woods',47,'Doctor');
UPSERT INTO customer VALUES (4000044,'Patricia','Mangum',54,'Civil engineer');
UPSERT INTO customer VALUES (4000045,'Lois','Joseph',70,'Musician');
UPSERT INTO customer VALUES (4000046,'Louis','Rosenthal',53,'');
UPSERT INTO customer VALUES (4000047,'Christina','Bowden',52,'Computer software engineer');
UPSERT INTO customer VALUES (4000048,'Darlene','Barton',21,'Doctor');
UPSERT INTO customer VALUES (4000049,'Harvey','Underwood',56,'Engineering technician');
UPSERT INTO customer VALUES (4000050,'William','Jones',22,'Photographer');
UPSERT INTO customer VALUES (4000051,'Frederick','Baker',39,'Writer');
UPSERT INTO customer VALUES (4000052,'Shirley','Merritt',53,'Reporter');
UPSERT INTO customer VALUES (4000053,'Jason','Cross',37,'Civil engineer');
UPSERT INTO customer VALUES (4000054,'Judith','Cooper',40,'Economist');
UPSERT INTO customer VALUES (4000055,'Gretchen','Holmes',45,'Childcare worker');
UPSERT INTO customer VALUES (4000056,'Don','Sharpe',44,'Social worker');
UPSERT INTO customer VALUES (4000057,'Glenda','Morgan',37,'Real estate agent');
UPSERT INTO customer VALUES (4000058,'Scott','Hoyle',66,'Doctor');
UPSERT INTO customer VALUES (4000059,'Pat','Allen',27,'Secretary');
UPSERT INTO customer VALUES (4000060,'Michelle','Rich',65,'Artist');
UPSERT INTO customer VALUES (4000061,'Jessica','Rich',52,'Actor');
UPSERT INTO customer VALUES (4000062,'Evan','Grant',55,'Agricultural and food scientist');
UPSERT INTO customer VALUES (4000063,'Melinda','Proctor',69,'Teacher');
UPSERT INTO customer VALUES (4000064,'Calvin','Diaz',67,'Athlete');
UPSERT INTO customer VALUES (4000065,'Eugene','Graham',47,'Police officer');
UPSERT INTO customer VALUES (4000066,'Vickie','Watkins',53,'Computer support specialist');
UPSERT INTO customer VALUES (4000067,'Luis','Hinton',24,'Childcare worker');
UPSERT INTO customer VALUES (4000068,'Allan','Marsh',44,'Athlete');
UPSERT INTO customer VALUES (4000069,'Melanie','Hewitt',62,'Real estate agent');
UPSERT INTO customer VALUES (4000070,'Marianne','Branch',39,'Judge');
UPSERT INTO customer VALUES (4000071,'Natalie','Walton',54,'Recreation and fitness worker');
UPSERT INTO customer VALUES (4000073,'Arlene','Case',27,'Musician');
UPSERT INTO customer VALUES (4000074,'Kyle','Watts',44,'Engineering technician');
UPSERT INTO customer VALUES (4000075,'Calvin','Christensen',31,'Architect');
UPSERT INTO customer VALUES (4000076,'Gary','Parks',52,'Pharmacist');
UPSERT INTO customer VALUES (4000077,'Samantha','Hardin',26,'Doctor');
UPSERT INTO customer VALUES (4000078,'Sara','Lucas',36,'Loan officer');
UPSERT INTO customer VALUES (4000079,'Stacy','Eason',61,'Musician');
UPSERT INTO customer VALUES (4000080,'Gladys','Davidson',44,'Recreation and fitness worker');
UPSERT INTO customer VALUES (4000081,'Mike','Whitehead',24,'Politician');
UPSERT INTO customer VALUES (4000082,'Lynne','Rose',30,'Loan officer');
UPSERT INTO customer VALUES (4000083,'Faye','Sparks',73,'Civil engineer');
UPSERT INTO customer VALUES (4000084,'Diana','Moore',66,'Computer support specialist');
UPSERT INTO customer VALUES (4000085,'Leon','Pearson',75,'Physicist');
UPSERT INTO customer VALUES (4000086,'Ethel','Rodgers',48,'Librarian');
UPSERT INTO customer VALUES (4000087,'Steve','Graves',69,'Nurse');
UPSERT INTO customer VALUES (4000088,'Alison','Scarborough',25,'Designer');
UPSERT INTO customer VALUES (4000089,'Sherri','Sutton',45,'Social worker');
UPSERT INTO customer VALUES (4000090,'Patsy','Sinclair',61,'Police officer');
UPSERT INTO customer VALUES (4000091,'Kelly','Bowman',54,'Childcare worker');
UPSERT INTO customer VALUES (4000092,'Stacy','Olsen',39,'Veterinarian');
UPSERT INTO customer VALUES (4000093,'Curtis','Love',29,'Secretary');
UPSERT INTO customer VALUES (4000094,'Dana','McLean',25,'Artist');
UPSERT INTO customer VALUES (4000095,'Jennifer','Christian',32,'Human resources assistant');
UPSERT INTO customer VALUES (4000096,'Brett','Lamb',71,'Engineering technician');
UPSERT INTO customer VALUES (4000097,'Brandon','James',27,'Musician');
UPSERT INTO customer VALUES (4000098,'Keith','Chandler',73,'Coach');
UPSERT INTO customer VALUES (4000099,'Joann','Stout',54,'Real estate agent');
UPSERT INTO customer VALUES (4000100,'Ronnie','Cowan',41,'Photographer');
UPSERT INTO customer VALUES (4000101,'Scott','Golden',51,'Teacher');
UPSERT INTO customer VALUES (4000102,'Gene','Bowling',35,'Recreation and fitness worker');
UPSERT INTO customer VALUES (4000103,'Louise','Beasley',70,'Loan officer');
UPSERT INTO customer VALUES (4000104,'Geoffrey','Clapp',34,'Photographer');
UPSERT INTO customer VALUES (4000105,'Patricia','Abrams',31,'Veterinarian');
UPSERT INTO customer VALUES (4000106,'Jennifer','Tilley',32,'Agricultural and food scientist');
UPSERT INTO customer VALUES (4000107,'Mary','Morse',70,'Automotive mechanic');
UPSERT INTO customer VALUES (4000108,'Shawn','Boykin',26,'Photographer');
UPSERT INTO customer VALUES (4000109,'Vincent','Sumner',73,'Lawyer');
UPSERT INTO customer VALUES (4000110,'Kurt','Cassidy',73,'Dancer');
UPSERT INTO customer VALUES (4000111,'Danny','Davidson',45,'Agricultural and food scientist');
UPSERT INTO customer VALUES (4000112,'Charlene','Heath',51,'Electrician');
UPSERT INTO customer VALUES (4000113,'Alice','Blanchard',61,'Economist');
UPSERT INTO customer VALUES (4000114,'Joan','McAllister',37,'Engineering technician');
UPSERT INTO customer VALUES (4000115,'Betty','McKenzie',42,'Computer support specialist');
UPSERT INTO customer VALUES (4000116,'Danny','Byrne',67,'Dancer');
UPSERT INTO customer VALUES (4000117,'Peggy','Schroeder',37,'Loan officer');
UPSERT INTO customer VALUES (4000118,'Leslie','Griffin',52,'Photographer');
UPSERT INTO customer VALUES (4000119,'Marshall','Gross',45,'Actor');
UPSERT INTO customer VALUES (4000120,'Sara','Perkins',64,'Actor');
UPSERT INTO customer VALUES (4000121,'Martha','Robertson',38,'Agricultural and food scientist');
UPSERT INTO customer VALUES (4000122,'Jack','Palmer',59,'Human resources assistant');
UPSERT INTO customer VALUES (4000123,'Gayle','Brady',63,'Firefighter');
UPSERT INTO customer VALUES (4000124,'Benjamin','Rowe',35,'Childcare worker');
UPSERT INTO customer VALUES (4000125,'Roberta','Zhang',58,'Statistician');
UPSERT INTO customer VALUES (4000126,'Patricia','Hodge',46,'Artist');
UPSERT INTO customer VALUES (4000127,'Clifford','Li',62,'Photographer');
UPSERT INTO customer VALUES (4000128,'Joanne','Bowling',72,'Musician');
UPSERT INTO customer VALUES (4000129,'Martin','Justice',72,'Electrician');
UPSERT INTO customer VALUES (4000130,'Toni','Glass',69,'Lawyer');
UPSERT INTO customer VALUES (4000131,'Beth','Willis',33,'Carpenter');
UPSERT INTO customer VALUES (4000132,'Jessica','Hester',68,'Civil engineer');
UPSERT INTO customer VALUES (4000133,'Samantha','Floyd',34,'Childcare worker');
UPSERT INTO customer VALUES (4000134,'Jimmy','Graves',25,'Nurse');
UPSERT INTO customer VALUES (4000135,'Vincent','Fischer',34,'Statistician');
UPSERT INTO customer VALUES (4000136,'Dianne','Norman',54,'Veterinarian');
UPSERT INTO customer VALUES (4000137,'Rhonda','Chan',70,'Pharmacist');
UPSERT INTO customer VALUES (4000138,'Tamara','Hunt',51,'Psychologist');
UPSERT INTO customer VALUES (4000139,'Mary','Byrd',28,'Environmental scientist');
UPSERT INTO customer VALUES (4000140,'Sidney','Lane',63,'Statistician');
UPSERT INTO customer VALUES (4000141,'Jeff','Kaplan',23,'Chemist');
UPSERT INTO customer VALUES (4000142,'Sandra','Heller',37,'Photographer');
UPSERT INTO customer VALUES (4000143,'Katie','May',65,'Recreation and fitness worker');
UPSERT INTO customer VALUES (4000144,'Raymond','Jennings',74,'Coach');
UPSERT INTO customer VALUES (4000145,'Roger','Hanna',40,'Musician');
UPSERT INTO customer VALUES (4000146,'Natalie','Locklear',31,'Politician');
UPSERT INTO customer VALUES (4000147,'Kathy','Holloway',59,'Pharmacist');
UPSERT INTO customer VALUES (4000148,'Troy','Jones',22,'Secretary');
UPSERT INTO customer VALUES (4000149,'Neal','Glover',40,'Real estate agent');
UPSERT INTO customer VALUES (4000150,'Martin','Vick',23,'Physicist');
UPSERT INTO customer VALUES (4000152,'Vincent','Goldman',68,'Electrician');
UPSERT INTO customer VALUES (4000153,'Beth','McKenna',64,'Veterinarian');
UPSERT INTO customer VALUES (4000154,'Milton','Starr',51,'Carpenter');
UPSERT INTO customer VALUES (4000155,'Tamara','Stone',29,'Firefighter');
UPSERT INTO customer VALUES (4000156,'Mitchell','McClure',73,'Loan officer');
UPSERT INTO customer VALUES (4000157,'Franklin','Watson',33,'Coach');
UPSERT INTO customer VALUES (4000158,'Leroy','Monroe',35,'Computer support specialist');
UPSERT INTO customer VALUES (4000159,'Glen','Abbott',66,'Loan officer');
UPSERT INTO customer VALUES (4000160,'Judith','Singer',62,'Actor');
UPSERT INTO customer VALUES (4000161,'Alice','Hall',67,'Recreation and fitness worker');
UPSERT INTO customer VALUES (4000162,'Bruce','Farrell',31,'Librarian');
UPSERT INTO customer VALUES (4000163,'Kathleen','Lucas',45,'Chemist');
UPSERT INTO customer VALUES (4000164,'Amy','Norman',47,'Automotive mechanic');
UPSERT INTO customer VALUES (4000165,'Ronnie','Atkins',39,'Dancer');
UPSERT INTO customer VALUES (4000166,'Martha','Monroe',70,'Judge');
UPSERT INTO customer VALUES (4000167,'Lynn','Robertson',30,'Lawyer');
UPSERT INTO customer VALUES (4000168,'Jose','Sykes',69,'Writer');
UPSERT INTO customer VALUES (4000169,'Robert','Reid',72,'Carpenter');
UPSERT INTO customer VALUES (4000170,'Pauline','Chandler',36,'Economist');
UPSERT INTO customer VALUES (4000171,'Stephen','Finch',35,'Coach');
UPSERT INTO customer VALUES (4000172,'Peggy','Hobbs',21,'Musician');
UPSERT INTO customer VALUES (4000173,'Donna','Adkins',51,'Electrical engineer');
UPSERT INTO customer VALUES (4000174,'Doris','Kinney',73,'Athlete');
UPSERT INTO customer VALUES (4000175,'Ben','Whitaker',39,'Computer support specialist');
UPSERT INTO customer VALUES (4000176,'Kristin','Alexander',35,'Coach');
UPSERT INTO customer VALUES (4000177,'Ryan','Conner',39,'Electrical engineer');
UPSERT INTO customer VALUES (4000178,'Tracey','Waters',42,'Computer hardware engineer');
UPSERT INTO customer VALUES (4000179,'Mark','Becker',36,'Computer support specialist');
UPSERT INTO customer VALUES (4000180,'Louis','Rollins',73,'Economist');
UPSERT INTO customer VALUES (4000181,'Janet','Love',62,'Politician');
UPSERT INTO customer VALUES (4000182,'Leo','Adkins',68,'Economist');
UPSERT INTO customer VALUES (4000183,'Constance','Black',48,'Firefighter');
UPSERT INTO customer VALUES (4000184,'Sarah','Fox',40,'Psychologist');
UPSERT INTO customer VALUES (4000185,'Gladys','Hatcher',54,'Musician');
UPSERT INTO customer VALUES (4000186,'Hazel','Wu',38,'Therapist');
UPSERT INTO customer VALUES (4000187,'Hazel','Lloyd',59,'Politician');
UPSERT INTO customer VALUES (4000188,'Jerome','Joyce',26,'Artist');
UPSERT INTO customer VALUES (4000189,'Vincent','Welch',61,'Psychologist');
```

```
SELECT * FROM customer;

select avg(age),count(1),upper(profession)
from customer
group by profession
having count(1)>2
order  by avg(age) desc;

Create table if not exists customer_aggr (profession varchar(100) not null primary key, avg_age float,cnt
integer);
```

```
UPSERT INTO customer_aggr select upper(profession),avg(age),count(1)
from customer
group by profession
having count(1)>2
order  by avg(age) desc;
```

Create Views:

```
create view customer_aggr_view AS
SELECT * FROM customer_aggr
WHERE profession not in ('POLICE OFFICER','LAWYER','DOCTOR');
```

Index Creation:

```
CREATE INDEX IDX_CUSTOMER ON CUSTOMER (CUSTID);
CREATE INDEX IDX_CUSTOMER_COMPOSITE ON CUSTOMER (AGE,PROFESSION);
CREATE INDEX IDX_CUSTOMER_COVERED ON CUSTOMER (AGE) INCLUDE (FIRSTNAME);

DROP INDEX IDX_CUSTOMER_COVERED ON CUSTOMER;

CREATE INDEX IDX_CUSTOMER_COVERED ON CUSTOMER (AGE) INCLUDE (FIRSTNAME)
SALT_BUCKETS=3;
```

CBO :

```
UPDATE STATISTICS CUSTOMER COLUMNS;

UPDATE STATISTICS CUSTOMER INDEX;
```

Joins:

```
select a.*,b.*
from CUSTOMER as a inner join customer_aggr as b on
UPPER(a.PROFESSION)=b.PROFESSION;

select a.*,b.*
from CUSTOMER as a left outer join customer_aggr as b on
UPPER(a.PROFESSION)=b.PROFESSION;

select a.*,b.*
from CUSTOMER as a right outer join customer_aggr as b on
UPPER(a.PROFESSION)=b.PROFESSION;

select a.*,b.*
from CUSTOMER as a full outer join customer_aggr as b on
UPPER(a.PROFESSION)=b.PROFESSION;
```

Non interactive Query execution: Table creation and Bulk loading:

```
psql.py localhost /home/hduser/phoenixdata/WEB_STAT.sql /home/hduser/phoenixdata/WEB_STAT.csv
```

Perform Webdata analytics:

```
psql.py localhost /home/hduser/phoenixdata/WEB_STAT_QUERIES.sql
```

Hive Phoenix Handler:

```
create table retail.customer_hive(custno string, firstname string, lastname string, age
int,profession string)
row format delimited fields terminated by ',';

load data local inpath '/home/hduser/hive/data/custs' into table retail.customer_hive;

create table retail.customer_hive_phoenix (custno int,fname varchar(100),lname varchar(100),age
int,profession varchar(100))
STORED BY 'org.apache.phoenix.hive.PhoenixStorageHandler'
TBLPROPERTIES (
  "phoenix.table.name" = "cust_hive_phoenix",
  "phoenix.zookeeper.quorum" = "localhost",
  "phoenix.zookeeper.znode.parent" = "/hbase",
  "phoenix.zookeeper.client.port" = "2181",
  "phoenix.rowkeys" = "custno",
  "phoenix.column.mapping" = "rowkey:custno");

insert into table retail.customer_hive_phoenix select * from retail.customer_hive;
```

HBase Filters for Additional Practices:

####### Example of keyonlyfilter
####### This filter does not take any arguments. It returns only the key component of each key-value.#######

```
scan 'Patient',{ FILTER => "KeyOnlyFilter()"}
```

####### FirstKeyOnlyFilter#######
####### This filter does not take any arguments. It returns only the first key-value from each row.####### scan

'Patient',{ FILTER => "FirstKeyOnlyFilter()"}

####### prefixfilter: #######
####### This filter takes one argument a prefix of a row key. It returns only those key-values present in a
row that starts with the specified row prefix#######

```
scan 'Patient', {FILTER => "(PrefixFilter ('2'))"}
```

####### ColumnPrefixFilter - This filter takes one argument a column prefix. It returns only those keyvalues present in a column that starts with the specified column prefix. The column prefix must be of the form qualifier#######

```
scan 'Patient', {FILTER => "(PrefixFilter ('2')) AND ColumnPrefixFilter('a')"}
```

####### MultipleColumnPrefixFilter - This filter takes a list of column prefixes. It returns key-values that are present in a column that starts with any of the specified column prefixes. Each of the column prefixes must be of the form qualifier#######

```
scan 'Patient',{FILTER => "MultipleColumnPrefixFilter('p','a')"}
```

####### InclusiveStopFilter - This filter takes one argument a row key on which to stop scanning. It returns all key-values present in rows up to and including the specified row. #######

```
scan 'Patient',{FILTER => "InclusiveStopFilter('3')"}
```

####### Selecting columns and introducing limit #######

```
scan 'Patient', { COLUMNS => 'Personal:pname', LIMIT => 2}
```

####### Disable table#######

```
disable 'Patient'
```

####### Enable table#######

```
enable 'Patient'
```

####### drop the table. Table should be disabled to drop. #######

```
drop 'Patient'
```