```
 (A) $ $HIVE_HOME/bin hive --service cli
(B) hive> set hive.cli.print.current.db=true;
(C) hive (default)> CREATE DATABASE ourfirstdatabase; OK
Time taken: 3.756 seconds
(D) hive (default)> USE ourfirstdatabase; OK
Time taken: 0.039 seconds
(E) hive (ourfirstdatabase)> CREATE TABLE our_first_table
(
> FirstName STRING,
> LastName STRING,
> EmployeeId INT);
OK
Time taken: 0.043 seconds
hive (ourfirstdatabase)> quit;
```

1)

hduser@ubuntu:~$ start-all.sh

hduser@ubuntu:~$ hive

hive> set hive.cli.print.current.db=true;

hive (default)> create database ourfirstdatabase;

OK

Time taken: 1.955 seconds

hive (default)> use ourfirstdatabase;

OK

Time taken: 0.048 seconds

hive (ourfirstdatabase)> create table our_first_table(firstname string,lastname string,employeeid int);

OK

Time taken: 0.873 seconds

hive> show tables;

OK

our_first_table

Time taken: 0.114 seconds, Fetched: 1 row(s)

//set db properties use alter command

hive> ALTER DATABASE ourfirstdatabase SET DBPROPERTIES

> ('creator'=Swapnali Ware',

> 'created_for'='Learning Hive DDL');

OK

Time taken: 0.315 seconds

hive> DESCRIBE DATABASE EXTENDED ourfirstdatabase;

OK

ourfirstdatabase hdfs://localhost:54310/user/hive/warehouse/ourfirstdatabase.db
hduser USER {created_for=Learning Hive DDL, creator=Swapnali Ware}

Time taken: 0.091 seconds, Fetched: 1 row(s)

hive> DROP DATABASE ourfirstdatabase CASCADE;

OK

Time taken: 1.691 seconds

2)

Table name Customer Information

Rows	**Column Families**				
	ContactInfo		CustomerName		
	EA	**SA**	**FN**	**MN**	**LN**
1	sbw.ware@sinhgad.edu	Lonavala	Swapnali	B	Ware
2	riya.singh@xyz.com	Pune	Riya	S	Singh

hbase(main):005:0> create 'CustomerInformation','ContactInfo','CustomerName'

0 row(s) in 2.5900 seconds

=> Hbase::Table - CustomerInformation

hbase(main):006:0> list

TABLE

CustomerInformation

1 row(s) in 0.1070 seconds

=> ["CustomerInformation"]

hbase(main):001:0> list

```
TABLE
CustomerInformation
1 row(s) in 0.5790 seconds
(main):002:0> put 'CustomerInformation',1,'ContactInfo:EA','sbw.sit@sinhgad.edu'
0 row(s) in 0.6280 seconds
hbase(main):003:0> put 'CustomerInformation',1,'ContactInfo:SA','Lanavala'
0 row(s) in 0.0200 seconds

hbase(main):004:0> put 'CustomerInformation',1,'CustomerName:FN','Swapnali'
0 row(s) in 0.0530 seconds
hbase(main):005:0> put 'CustomerInformation',1,'CustomerName:MN','B'
0 row(s) in 0.0290 seconds
hbase(main):006:0> put 'CustomerInformation',1,'CustomerName:LN','Ware'
0 row(s) in 0.0040 seconds
hbase(main):009:0> put 'CustomerInformation',2,'ContactInfo:EA','riya.singh@xyz.com'
0 row(s) in 0.0060 seconds
hbase(main):010:0> put 'CustomerInformation',2,'ContactInfo:SA','Pune'
0 row(s) in 0.0140 seconds
hbase(main):011:0> put 'CustomerInformation',2,'CustomerName:FN','Riya'
0 row(s) in 0.0090 seconds
hbase(main):012:0> put 'CustomerInformation',2,'CustomerName:MN','S'
0 row(s) in 0.0170 seconds
hbase(main):013:0> put 'CustomerInformation',2,'CustomerName:LN','Singh'
0 row(s) in 0.0160 seconds
hbase(main):014:0> scan 'CustomerInformation'
ROW                 COLUMN+CELL
 1                  column=ContactInfo:EA, timestamp=1512450536862, value=sbw.
```

```
                sit@sinhgad.edu
1               column=ContactInfo:SA, timestamp=1512450573649, value=Lana
                vala
1               column=CustomerName:FN, timestamp=1512450617894, value=Swa
                pnali
1               column=CustomerName:LN, timestamp=1512450643726, value=War
                e
1               column=CustomerName:MN, timestamp=1512450630986, value=B
2               column=ContactInfo:EA, timestamp=1512450750624, value=riya
                .singh@xyz.com
2               column=ContactInfo:SA, timestamp=1512450771492, value=Pune
2               column=CustomerName:FN, timestamp=1512450784697, value=Riy
                a
2               column=CustomerName:LN, timestamp=1512450840361, value=Sin
                gh
2               column=CustomerName:MN, timestamp=1512450819560, value=S
2 row(s) in 0.2040 seconds
```

Goto Hive terminal

- In Step (A), you create an external table with a Key field to link up with the HBase row keys (1 and 2), and two **map data types (name and info)** to link up with the two column families (ContactInfo and CustomerName).
- Note the syntax for providing this linkage via the **WITH SERDEPROPERTIES** keywords. This SerDe con-figuration technique is quite common in Hive DDL.
- Note as well that the **TBLPROPERTIES** keyword is crucial for connecting the new **external hive_ hbase_table** with the actual CustomerInformation HBase table name.
- Step (B) shows how the key value pairs in HBase ({“FN”,”Swapnali”}, for example) are now available for querying with the help of the HiveQL. Note the syntax for accessing the Hive map data type in Step (C). You can select the value of the info map type using the notation info ["EA"] where "EA" is the key.

```
hive> CREATE EXTERNAL TABLE hive_hbase_table (
   > key INT,
```

```
    > name map<STRING,STRING>,
    > info map<STRING, STRING>)
    > STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
    > WITH SERDEPROPERTIES ("hbase.columns.mapping" =
    > ": key, CustomerName:, ContactInfo:")
    > TBLPROPERTIES ("hbase.table.name" = "CustomerInformation");
OK
Time taken: 6.248 seconds
hive>  SELECT * FROM hive_hbase_table;
OK
1       {"FN":"Swapnali","LN":"Ware","MN":"B"}
        {"EA":"sbw.sit@sinhgad.edu","SA":"Lanavala"}
2       {"FN":"Riya","LN":"Singh","MN":"S"}
        {"EA":"riya.singh@xyz.com","SA":"Pune"}
Time taken: 2.343 seconds, Fetched: 2 row(s)
hive> SELECT info["EA"] FROM hive_hbase_table WHERE
    > name["FN"] = "Swapnali" AND name["LN"] = "Ware";
OK
```

```
CREATE TABLE IF NOT EXISTS FlightInfo2007
    (
Year SMALLINT, Month TINYINT, DayofMonth TINYINT,
DayOfWeek TINYINT,
 DepTime SMALLINT, CRSDepTime SMALLINT, ArrTime SMALLINT,CRSArrTime SMALLINT,
    UniqueCarrier STRING, FlightNum STRING, TailNum STRING,
    ActualElapsedTime SMALLINT, CRSElapsedTime SMALLINT,
```

```
 AirTime SMALLINT, ArrDelay SMALLINT, DepDelay SMALLINT,
 Origin STRING, Dest STRING,Distance INT,
 TaxiIn SMALLINT, TaxiOut SMALLINT, Cancelled SMALLINT,
CancellationCode STRING, Diverted SMALLINT,
 CarrierDelay SMALLINT, WeatherDelay SMALLINT,
NASDelay SMALLINT, SecurityDelay SMALLINT,
LateAircraftDelay
SMALLINT)
COMMENT 'Flight InfoTable'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
TBLPROPERTIES ('creator'='PSB ', 'created_at'='Tues Dec 5 3:00:00 EDT 2017');
OK
> LINES TERMINATED BY '\n'

Time taken: 0.292 seconds
hive> load data local inpath '/home/student/Desktop/2007.csv' into table FlightInfo2007;
hive> CREATE TABLE IF NOT EXISTS FlightInfo2008 LIKE FlightInfo2007;
hive> load data local inpath '/home/hduser/Desktop/2008.csv' into table FlightInfo2008;

hive> CREATE TABLE IF NOT EXISTS myFlightInfo (
Year SMALLINT, DontQueryMonth TINYINT, DayofMonth
TINYINT, DayOfWeek TINYINT, DepTime SMALLINT, ArrTime SMALLINT,
UniqueCarrier STRING, FlightNum STRING,
AirTime SMALLINT, ArrDelay SMALLINT, DepDelay SMALLINT,
Origin STRING, Dest STRING, Cancelled SMALLINT,
CancellationCode STRING)
COMMENT 'Flight InfoTable'
```

```
   PARTITIONED BY(Month TINYINT)
   ROW FORMAT DELIMITED
   FIELDS TERMINATED BY ','
   LINES TERMINATED BY '\n'
   STORED AS RCFILE TBLPROPERTIES ('creator'='’swapnali ware’,
   'created_at'='Mon sep 2 14:24:19 EDT 2017');
OK
Time taken: 1.697 seconds
hive> INSERT OVERWRITE TABLE myflightinfo PARTITION (Month=1)
   > SELECT Year, Month, DayofMonth, DayOfWeek, DepTime,
   > ArrTime, UniqueCarrier,FlightNum, AirTime, ArrDelay, DepDelay, Origin,
   > Dest, Cancelled,CancellationCode FROM FlightInfo2008 WHERE Month=1;
hive> FROM FlightInfo2008 INSERT INTO TABLE myflightinfo
   > PARTITION (Month=2) SELECT Year, Month, DayofMonth, DayOfWeek, DepTime,
   > ArrTime, UniqueCarrier, FlightNum,
   > AirTime, ArrDelay, DepDelay, Origin, Dest, Cancelled,
   > CancellationCode WHERE Month=2
   > INSERT INTO TABLE myflightinfo
   > PARTITION (Month=12)
   > SELECT Year, Month, DayofMonth, DayOfWeek, DepTime,
   > ArrTime, UniqueCarrier, FlightNum,
   > AirTime, ArrDelay, DepDelay, Origin, Dest, Cancelled,
   > CancellationCode WHERE Month=12;

hive> SHOW PARTITIONS myflightinfo;
OK
month=1
month=12
month=2
```

```
Time taken: 0.344 seconds, Fetched: 3 row(s)
hive> CREATE TABLE myflightinfo2007 AS
    > SELECT Year, Month, DepTime, ArrTime, FlightNum,
    > Origin, Dest FROM FlightInfo2007
    > WHERE (Month = 7 AND DayofMonth = 3) AND
    > (Origin='JFK' AND Dest='ORD');
hive>SELECT * FROM myFlightInfo2007;
OK
2007    7       700     834     5447    JFK     ORD
2007    7       1633    1812    5469    JFK     ORD
2007    7       1905    2100    5492    JFK     ORD
2007    7       1453    1624    4133    JFK     ORD
2007    7       1810    1956    4392    JFK     ORD
2007    7       643     759     903     JFK     ORD
2007    7       939     1108    907     JFK     ORD
2007    7       1313    1436    915     JFK     ORD
2007    7       1617    1755    917     JFK     ORD
2007    7       2002    2139    919     JFK     ORD
Time taken: 1.219 seconds, Fetched: 10 row(s)
hive> CREATE TABLE myFlightInfo2008 AS
    > SELECT Year, Month, DepTime, ArrTime, FlightNum,
    > Origin, Dest FROM FlightInfo2008
    > WHERE (Month = 7 AND DayofMonth = 3) AND
    > (Origin='JFK' AND Dest='ORD');
hive> SELECT * FROM myFlightInfo2008;
OK
2008    7       930     1103    5199    JFK     ORD
2008    7       705     849     5687    JFK     ORD
2008    7       1645    1914    5469    JFK     ORD
```

2008 7 1345 1514 4392 JFK ORD

2008 7 1718 1907 1217 JFK ORD

2008 7 757 929 1323 JFK ORD

2008 7 928 1057 907 JFK ORD

2008 7 1358 1532 915 JFK ORD

2008 7 1646 1846 917 JFK ORD

2008 7 2129 2341 919 JFK ORD

Time taken: 0.424 seconds, Fetched: 10 row(s)

JOIN

Hive>SELECT m8.Year, m8.Month, m8.FlightNum, m8.Origin, m8.Dest, m7.Year, m7.Month, m7.FlightNum, m7.Origin, m7.Dest FROM myFlightinfo2008 m8 JOIN myFlightinfo2007 m7 ON m8.FlightNum=m7.FlightNum;

2008 7 5469 JFK ORD 2007 7 5469 JFK ORD

2008 7 4392 JFK ORD 2007 7 4392 JFK ORD

2008 7 907 JFK ORD 2007 7 907 JFK ORD

2008 7 915 JFK ORD 2007 7 915 JFK ORD

2008 7 917 JFK ORD 2007 7 917 JFK ORD

2008 7 919 JFK ORD 2007 7 919 JFK ORD

hive> SELECT m8.FlightNum,m8.Origin,m8.Dest,m7.FlightNum,m7.Origin,m7.Dest FROM myFlightinfo2008 m8 FULL OUTER JOIN myFlightinfo2007 m7 ON m8.FlightNum=m7.FlightNum;

1217 JFK ORD NULL NULL NULL

1323 JFK ORD NULL NULL NULL

NULL NULL NULL 4133 JFK ORD

4392 JFK ORD 4392 JFK ORD

5199 JFK ORD NULL NULL NULL

NULL NULL NULL 5447 JFK ORD

5469 JFK ORD 5469 JFK ORD

NULL NULL NULL 5492 JFK ORD

5687 JFK ORD NULL NULL NULL

NULL NULL NULL 903 JFK ORD

907 JFK ORD 907 JFK ORD

915 JFK ORD 915 JFK ORD

917 JFK ORD 917 JFK ORD

919 JFK ORD 919 JFK ORD

Time taken: 10.33 seconds, Fetched: 14 row(s)

hive>SELECT m8.Year,m8.Month,m8.FlightNum,m8.Origin,m8.Dest,m7.Year,m7.Month,m7.FlightNum,m7.Origin,m7.Dest FROM myFlightinfo2008 m8 LEFT OUTER JOIN myFlightinfo2007 m7 ON m8.FlightNum=m7.FlightNum;

2008 7 5199 JFK ORD NULL NULL NULL NULL NULL

2008 7 5687 JFK ORD NULL NULL NULL NULL NULL

2008 7 5469 JFK ORD 2007 7 5469 JFK ORD

2008 7 4392 JFK ORD 2007 7 4392 JFK ORD

2008 7 1217 JFK ORD NULL NULL NULL NULL NULL

2008 7 1323 JFK ORD NULL NULL NULL NULL NULL

2008 7 907 JFK ORD 2007 7 907 JFK ORD

2008 7 915 JFK ORD 2007 7 915 JFK ORD

2008 7 917 JFK ORD 2007 7 917 JFK ORD

2008 7 919 JFK ORD 2007 7 919 JFK ORD

hive> CREATE INDEX f08_index ON TABLE flightinfo2008 (Origin) AS

> 'COMPACT' WITH DEFERRED REBUILD;

OK

Time taken: 1.124 seconds

hive> ALTER INDEX f08_index ON flightinfo2008 REBUILD;

hive>SHOW INDEXES ON FlightInfo2008;

OK

f08_index flightinfo2008 origin default__flightinfo2008_f08_index__ compact

```
Time taken: 2.549 seconds, Fetched: 1 row(s)
hive> SELECT Origin, COUNT(1) FROM
    > flightinfo2008 WHERE Origin = 'SYR' GROUP BY Origin;
hive> DESCRIBE default__flightinfo2008_f08_index__;
OK
origin              string
_bucketname                 string
_offsets            array<bigint>
Time taken: 0.927 seconds, Fetched: 3 row(s)
hive> SELECT Origin, SIZE(`_offsets`)
    > FROM default__flightinfo2008_f08_index__ WHERE origin='SYR';
OK
SYR     12032
Time taken: 0.705 seconds, Fetched: 1 row(s)
hive> CREATE VIEW avgdepdelay AS
    > SELECT DayOfWeek, AVG(DepDelay) FROM
    > FlightInfo2008 GROUP BY DayOfWeek;
hive> SELECT * FROM avgdepdelay;
3           8.289761053658728
6           8.645680904903614
1           10.269990244459473
4           9.772897177836702
7           11.568973392595312
2           8.97689712068735
5           12.158036387869656
```

Day 5 under the results in Step (B) — had the highest number of delays.

Step (A): We want to point out that Hive's Data Definition Language (DDL) also includes the CREATE VIEW statement, which can be quite useful. In Hive, views allow a query to be saved but data is not stored as with the Create Table as Select (CTAS) statement.

When a view is referenced in HiveQL, Hive executes the query and then uses the results which could be part of a larger query. This can be very useful to simplify complex queries and break them down into logical components.

Additionally, the GROUP BY clause, which gathers all the days per week and allows the AVG aggregate function to provide a consolidated answer per day.

After we answered our question above about average flight delays per day,