Hadoop Optimization Techniques

-PARTITIONING

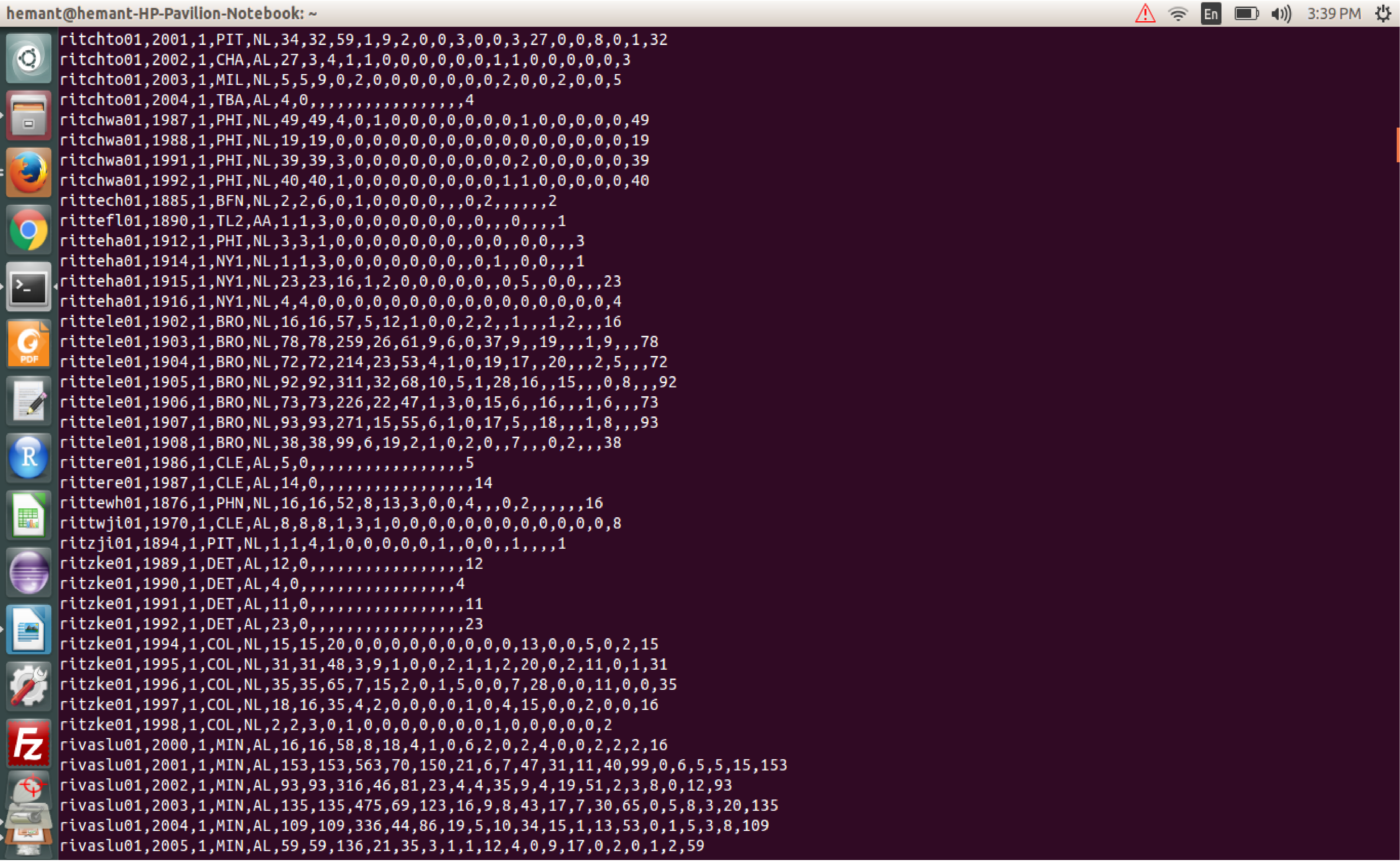
-BUCKETING

-VECTORIZATION

Partitioning

- Hive organizes tables into partitions. It is a way of dividing a table into related parts based on the values of partitioned columns such as date, city, and department. Using partition, it is easy to query a portion of the data.

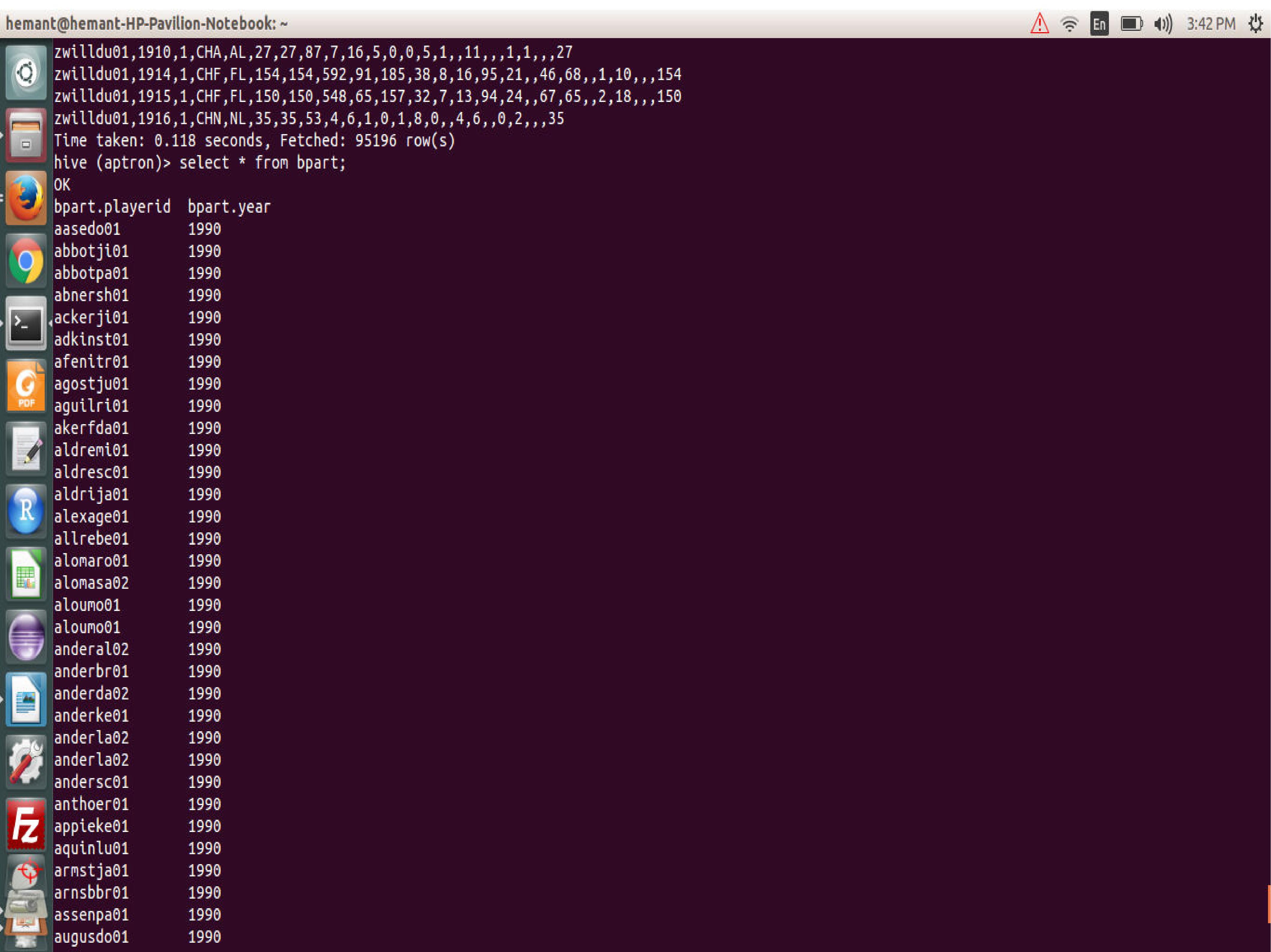
Batting table



```
ritchto01,2001,1,PIT,NL,34,32,59,1,9,2,0,0,3,0,0,3,27,0,0,8,0,1,32
ritchto01,2002,1,CHA,AL,27,3,4,1,1,0,0,0,0,0,0,1,1,0,0,0,0,0,3
ritchto01,2003,1,MIL,NL,5,5,9,0,2,0,0,0,0,0,0,0,2,0,0,2,0,0,5
ritchto01,2004,1,TBA,AL,4,0,,,,,,,,,,,,,,,,,4
ritchwa01,1987,1,PHI,NL,49,49,4,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,49
ritchwa01,1988,1,PHI,NL,19,19,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,19
ritchwa01,1991,1,PHI,NL,39,39,3,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,39
ritchwa01,1992,1,PHI,NL,40,40,1,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,40
rittech01,1885,1,BFN,NL,2,2,6,0,1,0,0,0,0,,,0,2,,,,,,2
rittefl01,1890,1,TL2,AA,1,1,3,0,0,0,0,0,0,0,,0,,,0,,,,1
ritteha01,1912,1,PHI,NL,3,3,1,0,0,0,0,0,0,0,,0,0,,0,0,,,3
ritteha01,1914,1,NY1,NL,1,1,3,0,0,0,0,0,0,0,,0,1,,0,0,,,1
ritteha01,1915,1,NY1,NL,23,23,16,1,2,0,0,0,0,0,,0,5,,0,0,,,23
ritteha01,1916,1,NY1,NL,4,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,4
rittele01,1902,1,BRO,NL,16,16,57,5,12,1,0,0,2,2,,1,,,1,2,,,16
rittele01,1903,1,BRO,NL,78,78,259,26,61,9,6,0,37,9,,19,,,1,9,,,78
rittele01,1904,1,BRO,NL,72,72,214,23,53,4,1,0,19,17,,20,,,2,5,,,72
rittele01,1905,1,BRO,NL,92,92,311,32,68,10,5,1,28,16,,15,,,0,8,,,92
rittele01,1906,1,BRO,NL,73,73,226,22,47,1,3,0,15,6,,16,,,1,6,,,73
rittele01,1907,1,BRO,NL,93,93,271,15,55,6,1,0,17,5,,18,,,1,8,,,93
rittele01,1908,1,BRO,NL,38,38,99,6,19,2,1,0,2,0,,7,,,0,2,,,38
rittere01,1986,1,CLE,AL,5,0,,,,,,,,,,,,,,,,,5
rittere01,1987,1,CLE,AL,14,0,,,,,,,,,,,,,,,,,14
rittewh01,1876,1,PHN,NL,16,16,52,8,13,3,0,0,4,,,0,2,,,,,,16
rittwji01,1970,1,CLE,AL,8,8,8,1,3,1,0,0,0,0,0,0,0,0,0,0,0,0,8
ritzji01,1894,1,PIT,NL,1,1,4,1,0,0,0,0,0,1,,0,0,,1,,,,1
ritzke01,1989,1,DET,AL,12,0,,,,,,,,,,,,,,,,,12
ritzke01,1990,1,DET,AL,4,0,,,,,,,,,,,,,,,,,4
ritzke01,1991,1,DET,AL,11,0,,,,,,,,,,,,,,,,,11
ritzke01,1992,1,DET,AL,23,0,,,,,,,,,,,,,,,,,23
ritzke01,1994,1,COL,NL,15,15,20,0,0,0,0,0,0,0,0,0,13,0,0,5,0,2,15
ritzke01,1995,1,COL,NL,31,31,48,3,9,1,0,0,2,1,1,2,20,0,2,11,0,1,31
ritzke01,1996,1,COL,NL,35,35,65,7,15,2,0,1,5,0,0,7,28,0,0,11,0,0,35
ritzke01,1997,1,COL,NL,18,16,35,4,2,0,0,0,0,1,0,4,15,0,0,2,0,0,16
ritzke01,1998,1,COL,NL,2,2,3,0,1,0,0,0,0,0,0,1,0,0,0,0,0,2
rivaslu01,2000,1,MIN,AL,16,16,58,8,18,4,1,0,6,2,0,2,4,0,0,2,2,2,16
rivaslu01,2001,1,MIN,AL,153,153,563,70,150,21,6,7,47,31,11,40,99,0,6,5,5,15,153
rivaslu01,2002,1,MIN,AL,93,93,316,46,81,23,4,4,35,9,4,19,51,2,3,8,0,12,93
rivaslu01,2003,1,MIN,AL,135,135,475,69,123,16,9,8,43,17,7,30,65,0,5,8,3,20,135
rivaslu01,2004,1,MIN,AL,109,109,336,44,86,19,5,10,34,15,1,13,53,0,1,5,3,8,109
rivaslu01,2005,1,MIN,AL,59,59,136,21,35,3,1,1,12,4,0,9,17,0,2,0,1,2,59
```

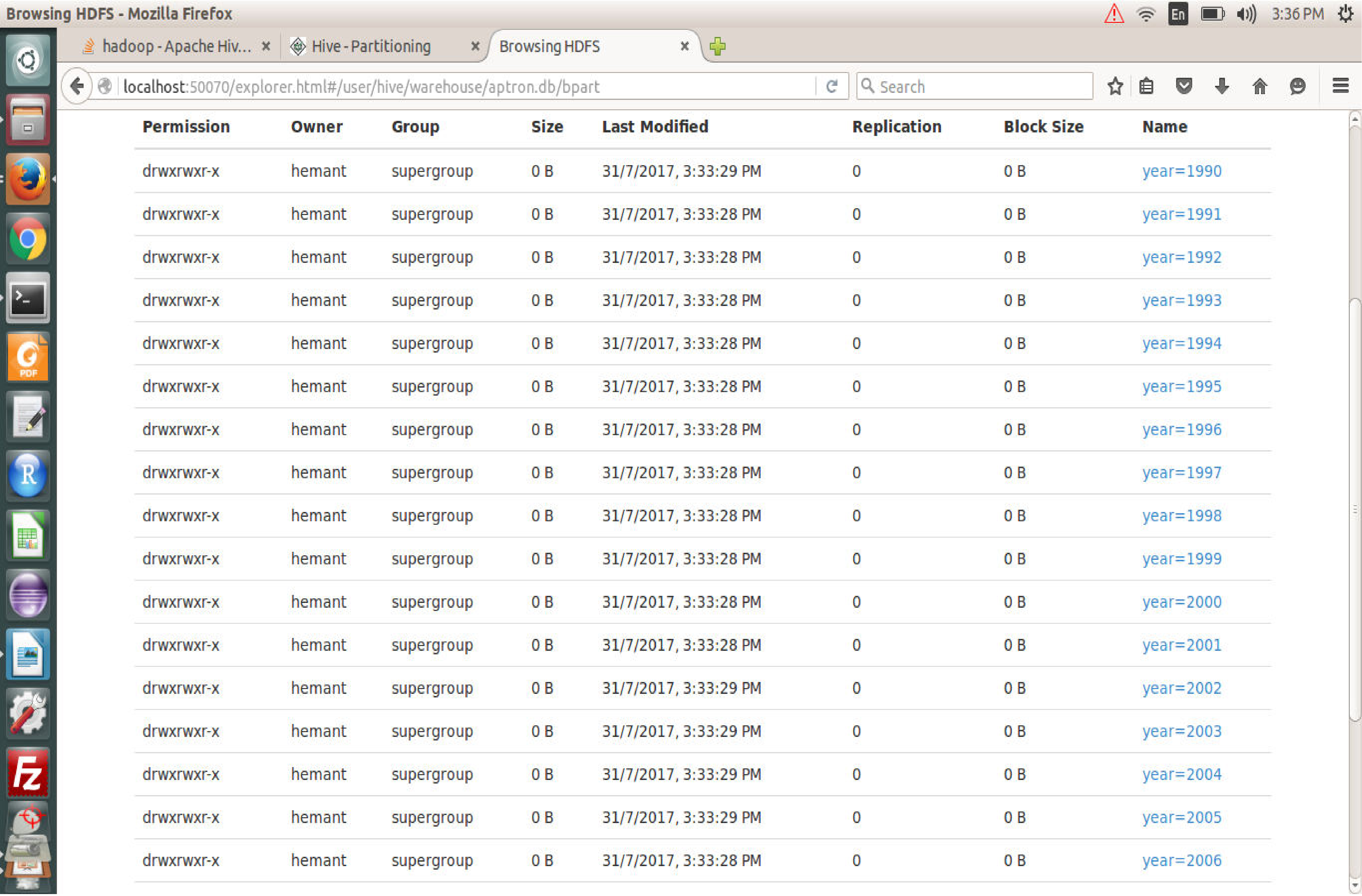
```
> create table bpart (playerid string) partitioned by (year string);
```

-
  ```
  > insert overwrite table bpart partition (year) select playerid,year from batting where year between '1990' and '2010';
  ```



```
zwilldu01,1910,1,CHA,AL,27,27,87,7,16,5,0,0,5,1,,11,,,1,1,,,27
zwilldu01,1914,1,CHF,FL,154,154,592,91,185,38,8,16,95,21,,46,68,,1,10,,,154
zwilldu01,1915,1,CHF,FL,150,150,548,65,157,32,7,13,94,24,,67,65,,2,18,,,150
zwilldu01,1916,1,CHN,NL,35,35,53,4,6,1,0,1,8,0,,4,6,,0,2,,,35
Time taken: 0.118 seconds, Fetched: 95196 row(s)
hive (aptron)> select * from bpart;
OK
bpart.playerid  bpart.year
aasedo01        1990
abbotji01       1990
abbotpa01       1990
abnersh01       1990
ackerji01       1990
adkinst01       1990
afenitr01       1990
agostju01       1990
aguilri01       1990
akerfda01       1990
aldremi01       1990
aldresc01       1990
aldrija01       1990
alexage01       1990
allrebe01       1990
alomaro01       1990
alomasa02       1990
aloumo01        1990
aloumo01        1990
anderal02       1990
anderbr01       1990
anderda02       1990
anderke01       1990
anderla02       1990
anderla02       1990
andersc01       1990
anthoer01       1990
appieke01       1990
aquinlu01       1990
armstja01       1990
arnsbbr01       1990
assenpa01       1990
augusdo01       1990
```

Hdfs



Browsing HDFS - Mozilla Firefox
3:36 PM
hadoop - Apache Hiv...
Hive - Partitioning
Browsing HDFS
localhost:50070/explorer.html#/user/hive/warehouse/aptron.db/bpart
Search

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxrwxr-x	hemant	supergroup	0 B	31/7/2017, 3:33:29 PM	0	0 B	year=1990
drwxrwxr-x	hemant	supergroup	0 B	31/7/2017, 3:33:28 PM	0	0 B	year=1991
drwxrwxr-x	hemant	supergroup	0 B	31/7/2017, 3:33:28 PM	0	0 B	year=1992
drwxrwxr-x	hemant	supergroup	0 B	31/7/2017, 3:33:28 PM	0	0 B	year=1993
drwxrwxr-x	hemant	supergroup	0 B	31/7/2017, 3:33:28 PM	0	0 B	year=1994
drwxrwxr-x	hemant	supergroup	0 B	31/7/2017, 3:33:28 PM	0	0 B	year=1995
drwxrwxr-x	hemant	supergroup	0 B	31/7/2017, 3:33:28 PM	0	0 B	year=1996
drwxrwxr-x	hemant	supergroup	0 B	31/7/2017, 3:33:28 PM	0	0 B	year=1997
drwxrwxr-x	hemant	supergroup	0 B	31/7/2017, 3:33:28 PM	0	0 B	year=1998
drwxrwxr-x	hemant	supergroup	0 B	31/7/2017, 3:33:28 PM	0	0 B	year=1999
drwxrwxr-x	hemant	supergroup	0 B	31/7/2017, 3:33:28 PM	0	0 B	year=2000
drwxrwxr-x	hemant	supergroup	0 B	31/7/2017, 3:33:28 PM	0	0 B	year=2001
drwxrwxr-x	hemant	supergroup	0 B	31/7/2017, 3:33:29 PM	0	0 B	year=2002
drwxrwxr-x	hemant	supergroup	0 B	31/7/2017, 3:33:29 PM	0	0 B	year=2003
drwxrwxr-x	hemant	supergroup	0 B	31/7/2017, 3:33:29 PM	0	0 B	year=2004
drwxrwxr-x	hemant	supergroup	0 B	31/7/2017, 3:33:29 PM	0	0 B	year=2005
drwxrwxr-x	hemant	supergroup	0 B	31/7/2017, 3:33:28 PM	0	0 B	year=2006

Bucketing

Bucketing is a simple idea if you are already aware. You create multiple buckets. You read each record and place it into one of the buckets based on some logic mostly some kind of hashing algorithm. This allows you to organize your data by decomposing it into multiple parts

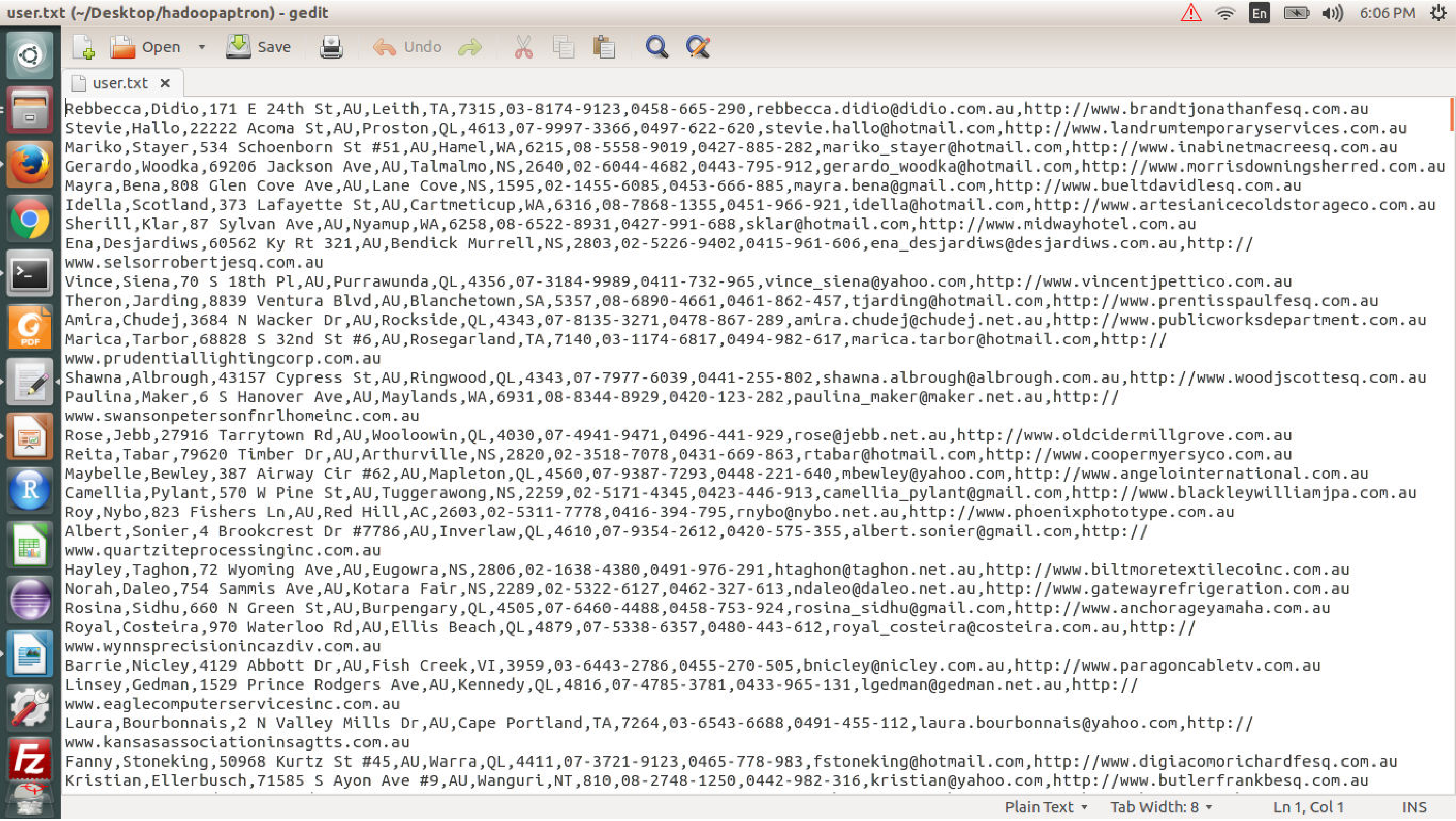
Usually Partitioning in Hive offers a way of segregating hive table data into multiple files/directories. But partitioning gives effective results when,

- There are limited number of partitions

- Comparatively equal sized partitions

Bucketing concept is based on (hashing function on the bucketed column) mod (by total number of buckets).

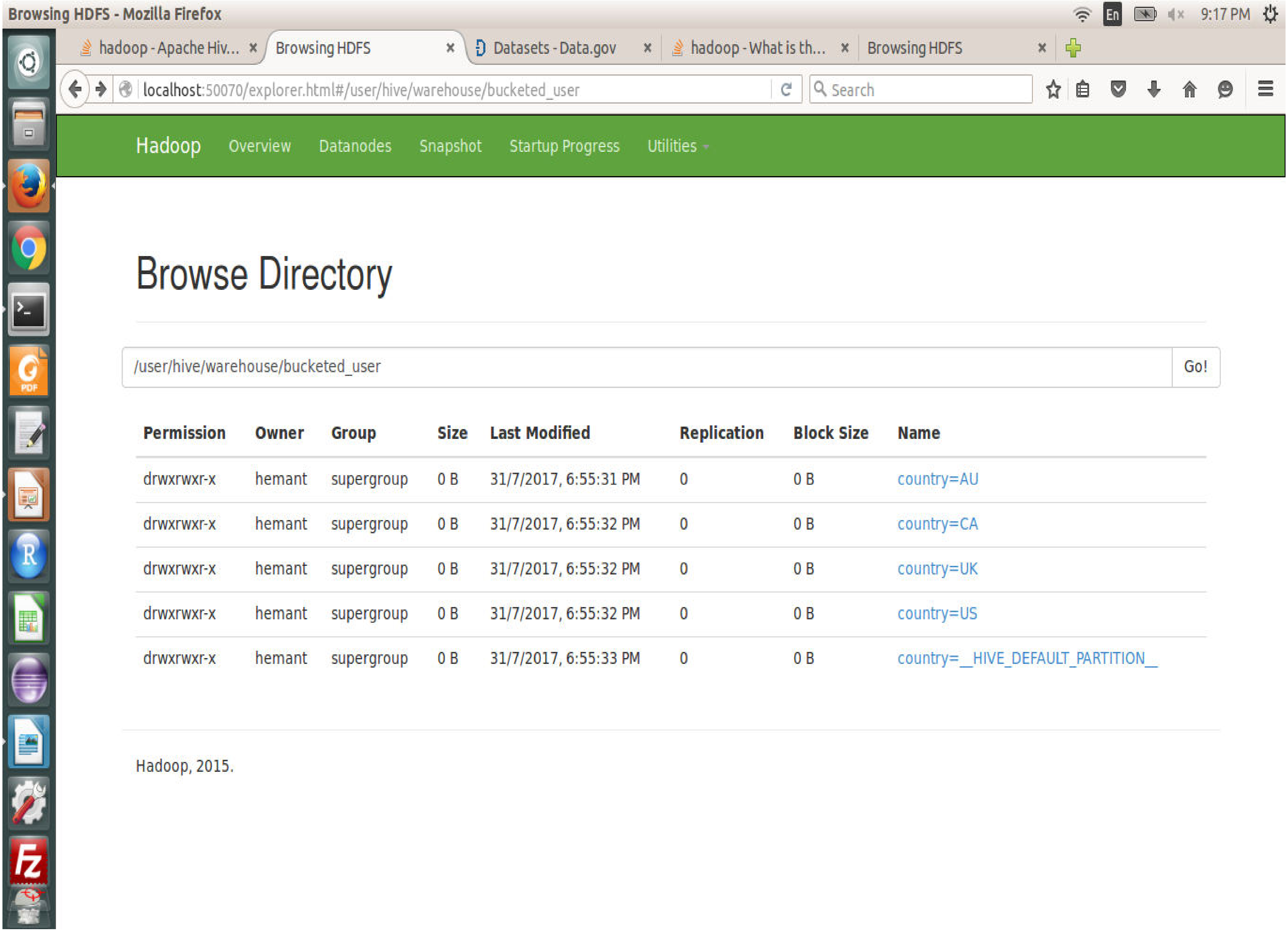
User .txt



```
Rebbecca,Didio,171 E 24th St,AU,Leith,TA,7315,03-8174-9123,0458-665-290,rebbecca.didio@didio.com.au,http://www.brandtjonathanfesq.com.au
Stevie,Hallo,22222 Acoma St,AU,Proston,QL,4613,07-9997-3366,0497-622-620,stevie.hallo@hotmail.com,http://www.landrumtemporaryservices.com.au
Mariko,Stayer,534 Schoenborn St #51,AU,Hamel,WA,6215,08-5558-9019,0427-885-282,mariko_stayer@hotmail.com,http://www.inabinetmacreesq.com.au
Gerardo,Woodka,69206 Jackson Ave,AU,Talmalmo,NS,2640,02-6044-4682,0443-795-912,gerardo_woodka@hotmail.com,http://www.morrisdowningsherred.com.au
Mayra,Bena,808 Glen Cove Ave,AU,Lane Cove,NS,1595,02-1455-6085,0453-666-885,mayra.bena@gmail.com,http://www.bueltdavidlesq.com.au
Idella,Scotland,373 Lafayette St,AU,Cartmeticup,WA,6316,08-7868-1355,0451-966-921,idella@hotmail.com,http://www.artesianicecoldstorageco.com.au
Sherill,Klar,87 Sylvan Ave,AU,Nyamup,WA,6258,08-6522-8931,0427-991-688,sklar@hotmail.com,http://www.midwayhotel.com.au
Ena,Desjardiws,60562 Ky Rt 321,AU,Bendick Murrell,NS,2803,02-5226-9402,0415-961-606,ena_desjardiws@desjardiws.com.au,http://
www.selsorrobertjesq.com.au
Vince,Siena,70 S 18th Pl,AU,Purrawunda,QL,4356,07-3184-9989,0411-732-965,vince_siena@yahoo.com,http://www.vincentjpettico.com.au
Theron,Jarding,8839 Ventura Blvd,AU,Blanchetown,SA,5357,08-6890-4661,0461-862-457,tjarding@hotmail.com,http://www.prentisspaulfesq.com.au
Amira,Chudej,3684 N Wacker Dr,AU,Rockside,QL,4343,07-8135-3271,0478-867-289,amira.chudej@chudej.net.au,http://www.publicworksdepartment.com.au
Marica,Tarbor,68828 S 32nd St #6,AU,Rosegarland,TA,7140,03-1174-6817,0494-982-617,marica.tarbor@hotmail.com,http://
www.prudentiallightingcorp.com.au
Shawna,Albrough,43157 Cypress St,AU,Ringwood,QL,4343,07-7977-6039,0441-255-802,shawna.albrough@albrough.com.au,http://www.woodjscottesq.com.au
Paulina,Maker,6 S Hanover Ave,AU,Maylands,WA,6931,08-8344-8929,0420-123-282,paulina_maker@maker.net.au,http://
www.swansonpetersonfnrlhomeinc.com.au
Rose,Jebb,27916 Tarrytown Rd,AU,Wooloowin,QL,4030,07-4941-9471,0496-441-929,rose@jebb.net.au,http://www.oldcidermillgrove.com.au
Reita,Tabar,79620 Timber Dr,AU,Arthurville,NS,2820,02-3518-7078,0431-669-863,rtabar@hotmail.com,http://www.coopermyersyco.com.au
Maybelle,Bewley,387 Airway Cir #62,AU,Mapleton,QL,4560,07-9387-7293,0448-221-640,mbewley@yahoo.com,http://www.angelointernational.com.au
Camellia,Pylant,570 W Pine St,AU,Tuggerawong,NS,2259,02-5171-4345,0423-446-913,camellia_pylant@gmail.com,http://www.blackleywilliamjpa.com.au
Roy,Nybo,823 Fishers Ln,AU,Red Hill,AC,2603,02-5311-7778,0416-394-795,rnybo@nybo.net.au,http://www.phoenixphototype.com.au
Albert,Sonier,4 Brookcrest Dr #7786,AU,Inverlaw,QL,4610,07-9354-2612,0420-575-355,albert.sonier@gmail.com,http://
www.quartziteprocessinginc.com.au
Hayley,Taghon,72 Wyoming Ave,AU,Eugowra,NS,2806,02-1638-4380,0491-976-291,htaghon@taghon.net.au,http://www.biltmoretextilecoinc.com.au
Norah,Daleo,754 Sammis Ave,AU,Kotara Fair,NS,2289,02-5322-6127,0462-327-613,ndaleo@daleo.net.au,http://www.gatewayrefrigeration.com.au
Rosina,Sidhu,660 N Green St,AU,Burpengary,QL,4505,07-6460-4488,0458-753-924,rosina_sidhu@gmail.com,http://www.anchorageyamaha.com.au
Royal,Costeira,970 Waterloo Rd,AU,Ellis Beach,QL,4879,07-5338-6357,0480-443-612,royal_costeira@costeira.com.au,http://
www.wynnsprecisionincazdiv.com.au
Barrie,Nicley,4129 Abbott Dr,AU,Fish Creek,VI,3959,03-6443-2786,0455-270-505,bnicley@nicley.com.au,http://www.paragoncabletv.com.au
Linsey,Gedman,1529 Prince Rodgers Ave,AU,Kennedy,QL,4816,07-4785-3781,0433-965-131,lgedman@gedman.net.au,http://
www.eaglecomputerservicesinc.com.au
Laura,Bourbonnais,2 N Valley Mills Dr,AU,Cape Portland,TA,7264,03-6543-6688,0491-455-112,laura.bourbonnais@yahoo.com,http://
www.kansasassociationinsagtts.com.au
Fanny,Stoneking,50968 Kurtz St #45,AU,Warra,QL,4411,07-3721-9123,0465-778-983,fstoneking@hotmail.com,http://www.digiacomorichardfesq.com.au
Kristian,Ellerbusch,71585 S Ayon Ave #9,AU,Wanguri,NT,810,08-2748-1250,0442-982-316,kristian@yahoo.com,http://www.butlerfrankbesq.com.au
```

```
--
set hive.exec.dynamic.partition=true;
set hive.exec.dynamic.partition.mode=nonstrict;
set hive.exec.max.dynamic.partitions.pernode=1000;


DROP TABLE IF EXISTS bucketed_user;

CREATE TEMPORARY TABLE temp_user(
       firstname VARCHAR(64),
        lastname  VARCHAR(64),
        address   STRING,
        country   VARCHAR(64),
        city      VARCHAR(64),
        state     VARCHAR(64),
        post      STRING,
        phone1    VARCHAR(64),
        phone2    STRING,
        email     STRING,
        web       STRING
        )
        ROW FORMAT DELIMITED
            FIELDS TERMINATED BY ','
            LINES TERMINATED BY '\n'
       STORED AS TEXTFILE;

LOAD DATA LOCAL INPATH '/home/hemant/Desktop/hadoopaptron/user.txt' INTO TABLE temp_user;
```

```

CREATE TABLE bucketed_user(
        firstname VARCHAR(64),
        lastname  VARCHAR(64),
        address   STRING,
        city       VARCHAR(64),
       state     VARCHAR(64),
        post      STRING,
        phone1    VARCHAR(64),
        phone2    STRING,
        email     STRING,
        web       STRING
        )
       COMMENT 'A bucketed sorted user table'
        PARTITIONED BY (country VARCHAR(64))
       CLUSTERED BY (state) SORTED BY (city) INTO 32 BUCKETS
        STORED AS SEQUENCEFILE;


INSERT OVERWRITE TABLE bucketed_user PARTITION (country)
       SELECT  firstname ,
                    lastname  ,
                    address   ,
               city      ,
               state     ,
                    post      ,
                    phone1    ,
                    phone2    ,
                    email     ,
                    web       ,
                    country
            FROM temp_user;
```

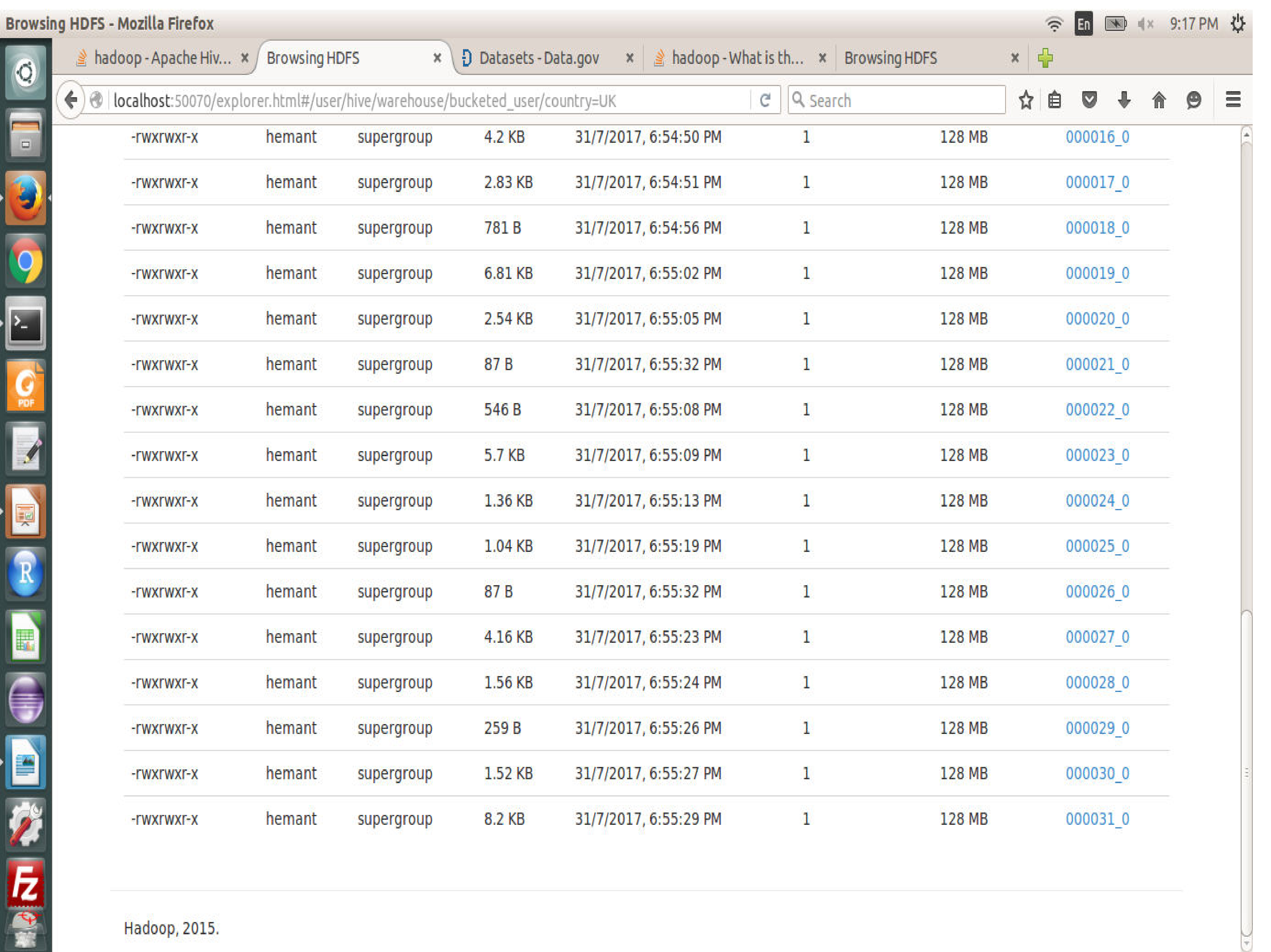


Browse Directory

/user/hive/warehouse/bucketed_user Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxrwxr-x	hemant	supergroup	0 B	31/7/2017, 6:55:31 PM	0	0 B	country=AU
drwxrwxr-x	hemant	supergroup	0 B	31/7/2017, 6:55:32 PM	0	0 B	country=CA
drwxrwxr-x	hemant	supergroup	0 B	31/7/2017, 6:55:32 PM	0	0 B	country=UK
drwxrwxr-x	hemant	supergroup	0 B	31/7/2017, 6:55:32 PM	0	0 B	country=US
drwxrwxr-x	hemant	supergroup	0 B	31/7/2017, 6:55:33 PM	0	0 B	country=__HIVE_DEFAULT_PARTITION__

Hadoop, 2015.


Browsing HDFS - Mozilla Firefox
9:17 PM
hadoop - Apache Hiv...
Browsing HDFS
Datasets - Data.gov
hadoop - What is th...
Browsing HDFS
localhost:50070/explorer.html#/user/hive/warehouse/bucketed_user/country=UK
Search

-rwxrwxr-x	hemant	supergroup	4.2 KB	31/7/2017, 6:54:50 PM	1	128 MB	000016_0
-rwxrwxr-x	hemant	supergroup	2.83 KB	31/7/2017, 6:54:51 PM	1	128 MB	000017_0
-rwxrwxr-x	hemant	supergroup	781 B	31/7/2017, 6:54:56 PM	1	128 MB	000018_0
-rwxrwxr-x	hemant	supergroup	6.81 KB	31/7/2017, 6:55:02 PM	1	128 MB	000019_0
-rwxrwxr-x	hemant	supergroup	2.54 KB	31/7/2017, 6:55:05 PM	1	128 MB	000020_0
-rwxrwxr-x	hemant	supergroup	87 B	31/7/2017, 6:55:32 PM	1	128 MB	000021_0
-rwxrwxr-x	hemant	supergroup	546 B	31/7/2017, 6:55:08 PM	1	128 MB	000022_0
-rwxrwxr-x	hemant	supergroup	5.7 KB	31/7/2017, 6:55:09 PM	1	128 MB	000023_0
-rwxrwxr-x	hemant	supergroup	1.36 KB	31/7/2017, 6:55:13 PM	1	128 MB	000024_0
-rwxrwxr-x	hemant	supergroup	1.04 KB	31/7/2017, 6:55:19 PM	1	128 MB	000025_0
-rwxrwxr-x	hemant	supergroup	87 B	31/7/2017, 6:55:32 PM	1	128 MB	000026_0
-rwxrwxr-x	hemant	supergroup	4.16 KB	31/7/2017, 6:55:23 PM	1	128 MB	000027_0
-rwxrwxr-x	hemant	supergroup	1.56 KB	31/7/2017, 6:55:24 PM	1	128 MB	000028_0
-rwxrwxr-x	hemant	supergroup	259 B	31/7/2017, 6:55:26 PM	1	128 MB	000029_0
-rwxrwxr-x	hemant	supergroup	1.52 KB	31/7/2017, 6:55:27 PM	1	128 MB	000030_0
-rwxrwxr-x	hemant	supergroup	8.2 KB	31/7/2017, 6:55:29 PM	1	128 MB	000031_0

Hadoop, 2015.

Vecotorization

- Vectorized query execution is a Hive feature that greatly reduces the CPU usage for typical query operations like scans, filters, aggregates, and joins. A standard query execution system processes one row at a time. This involves long code paths and significant metadata interpretation in the inner loop of execution. Vectorized query execution streamlines operations by processing a block of 1024 rows at a time. Within the block, each column is stored as a vector (an array of a primitive data type). Simple operations like arithmetic and comparisons are done by quickly iterating through the vectors in a tight loop, with no or very few function calls or conditional branches inside the loop. These loops compile in a streamlined way that uses relatively few instructions and finishes each instruction in fewer clock cycles, on average, by effectively using the processor pipeline and cache memory.

Vectorization

- Using Vectorized Query Execution
- To use vectorized query execution, you must store your data in ORC format, and set the following variable as shown in Hive SQL (see Configuring Hive):
- set hive.vectorized.execution.enabled = true;
- Vectorized execution is off by default, so your queries only utilize it if this variable is turned on. To disable vectorized execution and go back to standard execution, do the following:
- set hive.vectorized.execution.enabled = false;

Vectorized execution is a way of running multiple steps of the engine’s query processing in large chunks instead of doing it serially, one step at a time. It takes advantage of the availability of larger per-node memory and modern storage formats to perform the process more efficiently. A simple demonstration of it would be to imagine two arrays in which you want to sum up the numbers at identical indices: if you only have enough memory to hold two numbers at a time, you would have to

Grab the numbers at index 0 from each array, add them, and write them to disk;

Increment the index, and repeat the operation until you reach the end of the arrays.

But with more memory, you would be able to

Retrieve a large block of numbers from each array, add up the pairs in efficient, tight, in-memory loops, and write the block to disk;

Retrieve the next block, and repeat until you reach the end of the arrays.

It’s rather obvious that approach #2 would be much faster… but it also requires sufficient memory to hold the block, and a format that stores the data to be processed in a way that supports that type of operation (currently only the ORC format is supported - although I understand Parquet and Avro support is in development.)

Unless you’re rewriting Hive itself, there’s very little you need to do other than enable it to get its benefits - it’s core feature, not something you do as an end user. If you’re familiar with MapReduce programming, it’s somewhat similar to using a combiner on mapper output: if the operation you’re performing is both associative and commutative - i.e., distributive - then enabling vectorization for that query is likely to produce a benefit. If it is not - e.g., an averaging operation, or one using unsupported operators - then vectorization will be automatically disabled by Hive.

All of that being said: on some queries, the acceleration with vectorization enabled can be 3x or 4x of the original. As is usually the case, profiling and examining/optimizing the execution plan is the key

References

- Quora
- Google
- Stackover flow
- Hadooptutorial.info