

Session 9: Advance Hive

Assignment 1

Assignment 1 – You must perform the given tasks.

HIVE Queries create a Database and a table providing the Input Dataset.

CREATE DATABASE olympic;

USE olympic;

CREATE TABLE olympic_data

(Athlete STRING, Age INT, Country STRING, Year INT, Closing_Date STRING, Sport STRING, Gold_Medals INT, Silver_Medals INT, Bronze_Medals INT, Total_Medals INT)

ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';

LOAD DATA LOCAL INPATH '/home/acadgild/hadoop/olympic_data.json'

INTO TABLE olympic_data;

Task1

Write a Hive program to find the number of medals won by each country in swimming.

HIVE Query

SELECT country, SUM(total_medals) as Total_Medals FROM olympic_data WHERE sport<=>'Swimming' GROUP BY country;

```
hive (olympic)>  
> select country, SUM(total_medals) as Total_Medals FROM olympic_data where sport<=>'Swimming' GROUP BY country;
```

Expected Output

country	total_medals
Argentina	1
Australia	163
Austria	3
Belarus	2
Brazil	8
Canada	5
China	35
Costa Rica	2
Croatia	1
Denmark	1
France	39
Germany	32
Great Britain	11
Hungary	9
Italy	16
Japan	43
Lithuania	1
Netherlands	46
Norway	2
Poland	3
Romania	6
Russia	20
Serbia	1
Slovakia	2
Slovenia	1
South Africa	11
South Korea	4
Spain	3
Sweden	9
Trinidad and Tobago	1
Tunisia	3
Ukraine	7
United States	267
Zimbabwe	7

TASK2

Write a Hive program to find the number of medals that India won year wise.

HIVE Query

SELECT year,SUM(total_medals) FROM olympic_dataWhere country<=>'India' GROUP BY year;

```
hive (olympic)>
>
> Select year,SUM(total_medals) FROM olympic_data where country<=>'India' GROUP BY year;
```

Expected Output

```
OK
year      c1
2000      1
2004      1
2008      3
2012      6
```

TASK3

Write a Hive Program to find the total number of medals each country won.

HIVE Query

SELECT country,SUM(total_medals) as Total_Medals FROM olympic_data GROUP BY country;

```
hive (olympic)> SELECT country,SUM(total_medals) as Total_Medals FROM olympic_data GROUP BY country;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider
```

Expected Output

OK			Iran	24	
country total_medals			Ireland	9	
Afghanistan	2		Israel	4	
Algeria	8		Italy	331	
Argentina	141		Jamaica	80	
Armenia	10		Japan	282	
Australia	609		Kazakhstan	42	
Austria	91		Kenya	39	
Azerbaijan	25		Kuwait	2	
Bahamas	24		Kyrgyzstan	3	
Bahrain	1		Latvia	17	
Barbados	1		Lithuania	30	
Belarus	97		Macedonia	1	
Belgium	18		Malaysia	3	
Botswana	1		Mauritius	1	
Brazil	221		Mexico	38	
Bulgaria	41		Moldova	5	
Cameroon	20		Mongolia	10	
Canada	370		Montenegro	14	
Chile	22		Morocco	11	
China	530		Mozambique	1	
Chinese Taipei	20		Netherlands	318	
Colombia	13		New Zealand	52	
Costa Rica	2		Nigeria	39	
Croatia	81		North Korea	21	
Cuba	188		Norway	192	
Cyprus	1		Panama	1	
Czech Republic	81		Paraguay	17	
Denmark	89		Poland	80	
Dominican Republic	5		Portugal	9	
Ecuador	1		Puerto Rico	2	
Egypt	8		Qatar	3	
Eritrea	1		Romania	123	
Estonia	18		Russia	768	
Ethiopia	29		Saudi Arabia	6	
Finland	118		Serbia	31	
France	318		Serbia and Montenegro	38	
Gabon	1		Singapore	7	
Georgia	23		Slovakia	35	
Germany	629		Slovenia	25	
Great Britain	322		South Africa	25	
Greece	59		South Korea	308	
Grenada	1		Spain	205	
Guatemala	1		Sri Lanka	1	
Hong Kong	3		Sudan	1	
Hungary	145		Sweden	181	
Iceland	15		Switzerland	93	
India	11		Syria	1	
			Tajikistan	3	
			Thailand	18	

```

Togo      1
Trinidad and Tobago      19
Tunisia  4
Turkey   28
Uganda   1
Ukraine  143
United Arab Emirates      1
United States  1312
Uruguay  1
Uzbekistan      19
Venezuela      4
Vietnam  2
Zimbabwe      7
Time taken: 46.332 seconds, Fe
hive (olympic)>

```

Task4

Write a Hive program to find the number of gold medals each country won.

HIVE Query

SELECT country,SUM(gold_medals) as GOLD_Medals FROM olympic_data GROUP BY country;

```

hive (olympic)>
>
> SELECT country,SUM(gold_medals) as GOLD_Medals FROM olympic_data GROUP BY country;

```

Expect Output

country	gold_medals		
Afghanistan	0	Italy	86
Algeria	2	Jamaica	24
Argentina	49	Japan	57
Armenia	0	Kazakhstan	13
Australia	163	Kenya	11
Austria	36	Kuwait	0
Azerbaijan	6	Kyrgyzstan	0
Bahamas	11	Latvia	3
Bahrain	0	Lithuania	5
Barbados	0	Macedonia	0
Belarus	17	Malaysia	0
Belgium	2	Mauritius	0
Botswana	0	Mexico	19
Brazil	46	Moldova	0
Bulgaria	8	Mongolia	2
Cameroon	20	Montenegro	0
Canada	168	Morocco	2
Chile	3	Mozambique	1
China	234	Netherlands	101
Chinese Taipei	2	New Zealand	18
Colombia	2	Nigeria	6
Costa Rica	0	North Korea	6
Croatia	35	Norway	97
Cuba	57	Panama	1
Cyprus	0	Paraguay	0
Czech Republic	14	Poland	20
Denmark	46	Portugal	1
Dominican Republic	3	Puerto Rico	0
Ecuador	0	Qatar	0
Egypt	1	Romania	57
Eritrea	0	Russia	234
Estonia	6	Saudi Arabia	0
Ethiopia	13	Serbia	1
Finland	11	Serbia and Montenegro	11
France	108	Singapore	0
Gabon	0	Slovakia	10
Georgia	6	Slovenia	5
Germany	223	South Africa	10
Great Britain	124	South Korea	110
Greece	12	Spain	19
Grenada	1	Sri Lanka	0
Guatemala	0	Sudan	0
Hong Kong	0	Sweden	57
Hungary	77	Switzerland	21
Iceland	0	Syria	0
India	1	Tajikistan	0
Indonesia	5	Thailand	6
Iran	10	Togo	0
Ireland	1	Trinidad and Tobago	1
		Tunisia	2

Task 2

Write a hive UDF that implements functionality of string concat_ws(string SEP, array<string>). This UDF will accept two arguments, one string and one array of string. It will return a single string where all the elements of the array are separated by the SEP.

We have fortune 20 companies list and its company website URL, but the 'www' and the remaining domain are separated. In our output we try to achieve the output as below,

1 walmart www.walmart.com

Dataset

The below data contains, the column name as,

Rank, company_name, website, protocol.

1	Walmart	www	walmart.com
2	Exxon Mobil	www	exxonmobil.com
3	Apple	www	apple.com
4	Berkshire Hathaway	www	berkshirehathaway.com
5	McKesson	www	mckesson.com
6	UnitedHealth Group	www	unitedhealthgroup.com
7	CVS Health	www	cvshealth.com
8	General Motors	www	gm.com
9	Ford Motor	www	ford.com
10	AT&T	www	att.com
11	General Electric	www	ge.com
12	AmerisourceBergen	www	amerisourcebergen.com
13	Verizon	www	verizon.com
14	Chevron	www	chevron.com
15	Costco	www	costco.com
16	Fannie Mae	www	fanniemae.com
17	Kroger	www	thekrogerco.com
18	Amazon.com	www	amazon.com
19	Walgreens Boots Alli	www	walgreensbootsalliance.com
20	HP	www	hp.com

Prerequisites

Create Database and Table

Create Database FORTUNE20

HIVE QL

CREATE DATABASE FORTUNE20

Use FORTUNE20;

```

hive (Default)>
>
> CREATE DATABASE FORTUNE20;
OK
Time taken: 0.36 seconds
hive (Default)>
>
> SHOW Databases;
OK
database_name
abu
amit
custom
default
emp_details
fortune20
nyse
olympic
petrol
Time taken: 0.122 seconds, Fetched: 9 row(s)
hive (Default)>
>
> Use FORTUNE20;
OK
Time taken: 0.03 seconds
hive (FORTUNE20)>

```

Create Table Fortune_company

[HIVE QL](#)

CREATE TABLE fortune_company(rank int, company_namestring,website string, protocol string)

ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';

LOAD DATA LOCAL INPATH '/home/acadgild/hadoop/fortune20.txt'

INTO TABLE fortune20.fortune_company;


```

hive (FORTUNE20)>
>
> CREATE TABLE Fortune_Company
> (
> rank int,
> company_name string,
> website string,
> protocol string
> )
> ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';
OK
Time taken: 0.619 seconds
hive (FORTUNE20)> LOAD DATA LOCAL INPATH '/home/acadgild/hadoop/Fortune20.txt'
> INTO TABLE FORTUNE20.Fortune_Company;
Loading data to table fortune20.fortune_company
OK
Time taken: 0.704 seconds
hive (FORTUNE20)>
>
>
>
> SHOW Tables;
OK
tab_name
fortune_company
Time taken: 0.149 seconds, Fetched: 1 row(s)
hive (FORTUNE20)>

```

Viewing the data in the table fortune_company,

SELECT * FROM fortune_company;

```

hive (fortune20)>
>
> SELECT * FROM fortune_company;
OK
fortune_company.rank    fortune_company.company_name    fortune_company.website    fortune_company.protocol
1      Walmart    www    walmart.com
2      Exxon Mobil    www    exxonmobil.com
3      Apple    www    apple.com
4      Berkshire Hathaway    www    berkshirehathaway.com
5      McKesson    www    mckesson.com
6      UnitedHealth Group    www    unitedhealthgroup.com
7      CVS Health    www    cvshealth.com
8      General Motors    www    gm.com
9      Ford Motor    www    ford.com
10     AT&T    www    att.com
11     General Electric    www    ge.com
12     AmerisourceBergen    www    amerisourcebergen.com
13     Verizon    www    verizon.com
14     Chevron    www    chevron.com
15     Costco    www    costco.com
16     Fannie Mae    www    fanniemae.com
17     Kroger    www    thekrogerco.com
18     Amazon.com    www    amazon.com
19     Walgreens Boots Alli    www    walgreensbootsalliance.com
20     HP    www    hp.com
Time taken: 0.195 seconds, Fetched: 20 row(s)
hive (fortune20)>

```

HIVE UDF Java code

```
import java.util.ArrayList;
import org.apache.hadoop.hive.ql.exec.UDF;
```

```
public class CONCAT_WS extend UDF {
public String evaluate(String sep, ArrayList<String> arr)
{
StringBuffer strBuffer;

if (arr==null)
{
return null;
}
strBuffer = new StringBuffer();
strBuffer.append(arr.get(0));
for(int i =1; i <arr.size(); i ++){
strBuffer.append(sep);
strBuffer.append(arr.get(i));
}
return strBuffer.toString();
}
```

} After that we are adding JAR created from the JAVA class which is defining the UDF using below syntax-

HIVE UDF CONCAT_WS function

add jar /home/acadgild/hadoop/concatws.jar;

```
hive (fortune20)>
>
> add jar /home/acadgild/hadoop/concatws.jar;
Added [/home/acadgild/hadoop/concatws.jar] to class path
Added resources: [/home/acadgild/hadoop/concatws.jar]
hive (fortune20)>
```

After that we are creating a temporary function "CONCAT_WS" using below syntax-

CREATE TEMPORARY FUNCTION CONCAT_WS AS 'concatws.concatws';

```
hive (fortune20)>
>
> CREATE TEMPORARY FUNCTION CONCAT_WS AS 'concatws.concatws';
OK
Time taken: 0.023 seconds
hive (fortune20)>
```

After that we run below query to take one column (company_name) input as String and another array(website,',',protocol) as Array of Strings and concatenate them,

HIVE QL

SELECT rank, company_name, CONCAT_WS(website, '.', protocol) from fortune_company; SELECT rank, company_name, CONCAT_WS(website, '.', protocol) from fortune_company;

```
hive (fortune20)> SELECT rank, company_name, CONCAT_WS(website, '.', protocol) from fortune_company;
OK
rank      company_name      c2
```

Required Output

```
hive (fortune20)> SELECT rank, company_name, CONCAT_WS(website, '.', protocol) from fortune_company;
OK
rank      company_name      c2
1         Walmart          www.walmart.com
2         Exxon Mobil      www.exxonmobil.com
3         Apple            www.apple.com
4         Berkshire Hathaway  www.berkshirehathaway.com
5         McKesson          www.mckesson.com
6         UnitedHealth Group  www.unitedhealthgroup.com
7         CVS Health        www.cvshealth.com
8         General Motors     www.gm.com
9         Ford Motor        www.ford.com
10        AT&T             www.att.com
11        General Electric   www.ge.com
12        AmerisourceBergen  www.amerisourcebergen.com
13        Verizon            www.verizon.com
14        Chevron            www.chevron.com
15        Costco             www.costco.com
16        Fannie Mae         www.fanniemae.com
17        Kroger             www.thekrogerco.com
18        Amazon.com         www.amazon.com
19        Walgreens Boots Alli  www.walgreensbootsalliance.com
20        HP                 www.hp.com
Time taken: 0.211 seconds, Fetched: 20 row(s)
hive (fortune20)>
```

Task 3

Link: <https://acadgild.com/blog/transactions-in-hive/>

Refer the above given link for transactions in Hive and implement the operations given in the blog using your own sample data set and send us the screenshot.

6. Expected Output Solution report with commands, explanation to commands and screenshot for output.

Report shall be in PDF format. Submitted in GitHub

7. Approximate Time to Complete Task

6 hours.

Big Data Hadoop and Spark Development

8.