



# Project 1 - Apache Hive

Trevor Buck



## Question 1

Which English wikipedia article got the most traffic on October 20?

- Wget '\_\_\_' from wikipedia.
  - All 24 files related to october 20, 2020.
- CREATE TABLE oct20 (Nothing fancy here, I just included the same columns)
- LOAD INPATH LOCAL \_\_\_ INTO TABLE oct20



# Process

Select Statement:

- SELECT
  - Title,
  - SUM(num\_views) AS total\_views
- FROM oct20
- WHERE domain LIKE '%en%'
- GROUP BY title
- ORDER BY total\_views DESC
- LIMIT 10;

Results: Main Page was the top result

```
Total MapReduce CPU Time Spent: 5 minutes 14 seconds 280 msec
OK
+-----+-----+
|      title      | total_views |
+-----+-----+
| Main_Page       | 5993224    |
| Special:Search  | 1567856    |
| -               | 556824     |
| Jeffrey_Toobin  | 321459     |
| C._Rajagopalachari | 211147    |
| The_Haunting_of_Bly_Manor | 185139    |
| Robert_Redford  | 178779     |
| Jeff_Bridges    | 159163     |
| Bible           | 151535     |
| Chicago_Seven   | 149966     |
+-----+-----+
10 rows selected (110.509 seconds)
0: jdbc:hive2://>
```

## Table Creation:



## Question 2

What English wikipedia article has the largest fraction of its readers follow an internal link to another wikipedia article?

-- Key Assumptions --

- October 20 represents a typical day
- Any analysis run on October 20 can be multiplied by 30 to represent that of a 30-day month

-- Effects on Data

- Results can have > 100% link rate

- Wget  
<https://dumps.wikimedia.org/other/clickstream/2020-09/clickstream-enwiki-2020-09.tsv.gz>
- CREATE TABLE prev\_clicks AS
  - SELECT prev,
  - SUM(CASE WHEN type = 'link' THEN num ELSE 0 END) AS link\_occurences
  - FROM sept\_click
  - GROUP BY prev;
- CREATE TABLE one\_day\_views AS
  - SELECT title, SUM(views) AS num\_views
  - FROM october20
  - WHERE domain LIKE '%en%'
  - GROUP BY title;

# Merge Tables and get Results

- CREATE TABLE merged AS
  - SELECT one\_day\_views.title,
    - One\_day\_views.num\_views,
    - Prev\_clicks.link\_occurrences,
    - (prev\_clicks.link\_occurrences/(one\_day\_views.num\_views \* 30)) AS percentage
  - FROM one\_day\_views JOIN prev\_clicks
  - ON (one\_day\_views.title = prev\_clicks.prev)
  - ORDER BY percentage DESC

SELECT \* FROM merged WHERE num\_views > 10000 LIMIT 10;

```
0: jdbc:hive2://> SELECT * FROM merged WHERE num_views> 10000 LIMIT 10;
OK
```

merged.title	merged.num_views	merged.link_occurrences	merged.percentage
Ruth_Bader_Ginsburg	19911	2489227	4.167255955669395
Cobra_Kai	19018	2241751	3.929174115749991
Enola_Holmes_(film)	17708	1356311	2.553104058429335
September_11_attacks	11359	850181	2.4948821785955335
Supreme_Court_of_the_United_States	14517	1002716	2.3023948933434366
Mulan_(2020_film)	25958	1749519	2.246602203559596
The_Devil_All_the_Time_(film)	16335	1071565	2.1866442199775533
Dennis_Nilsen	10210	660393	2.1560333006856025
Ratched_(TV_series)	26384	1668477	2.1079404184354154
Nurse_Ratched	10192	546567	1.7075686813186813

```
10 rows selected (0.204 seconds)
0: jdbc:hive2://>
```

Side Note: This analysis also contains data as to which articles were more popular in september than October



## Question 3

What series of wikipedia articles, starting with 'Hotel\_California' keeps the largest fraction of its readers clicking on internal sites

```
SELECT * FROM sept_click  
WHERE type = 'link' AND prev LIKE 'Eagles_(box_set)'  
SORT BY num DESC  
LIMIT 1;
```

- Used an existing Table
- Ran the Above Command Multiple times
- At each iteration, I recorded the highest link, as well as the link's new title
- At the next iteration, I swapped the title

# Results

This is what the 10th Link looked like

```
Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 1.89 sec HDFS Read: 10019 HDFS Write: 147
Total MapReduce CPU Time Spent: 47 seconds 430 msec
OK
+-----+-----+-----+-----+
| sept_click.prev | sept_click.curr | sept_click.type | sept_click.num |
+-----+-----+-----+-----+
| Eagles_(box_set) | Long_Road_Out_of_Eden | link | 670 |
+-----+-----+-----+-----+
1 row selected (36.611 seconds)
0: jdbc:hive2://>
```

<u>Page Title</u>	<u>Links To Next Page</u>
Hotel_California	->2222->
Hotel_California_(Eagles_album)	->2127->
The_Long_Run_(album)	->1322->
Eagles_Live	->1136->
Eagles_Greatest_Hits,_Vol._2 -	>996->
The_Very_Best_of_the_Eagles	->892->
Hell_Freezes_Over	->735->
Selected_Works:_1972-1999	->705->
The_Very_Best_Of_(Eagles_album)	->646->
Eagles_(box_set)	->670->
Long_Road_Out_of_Eden	



## Question 4

Find an example of a wikipedia article that is relatively more popular in the UK. Find the same for the US and Australia.

--Key Assumptions--

- October 20th represents a typical day
- Each country is most active for six hours surrounding lunch/mid-afternoon
- Data will then be summed by country from 10:00 AM to 4:00 PM

### Hours Used:

America (5 Hours behind)	->	15:00 - 20:00 (inclusive)
UK is on time	->	10:00 - 15:00 (inclusive)
Australia (11 Hours ahead)	->	23:00 - 04:00 (inclusive)

### Loading Data:

```
LOAD DATA LOCAL INPATH  
'/home/trevorbuck/project2/data/pageviews-20201020-??0000'  
INTO TABLE one_day_??;
```

### Table Creation:

```
CREATE TABLE one_day_usa(  
  domain STRING,  
  title STRING,  
  views INT,  
  response INT  
)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY ' ';
```

### Side-Note:

I copied the same  
table format for both  
UK and AUS



# Merge Table and Results

```
CREATE TABLE usa_uk AS
SELECT one_day_usa.domain, one_day_usa.title, one_day_usa.views
AS views_usa, one_day_uk.views AS views_uk
FROM one_day_usa JOIN one_day_uk
ON (one_day_usa.title = one_day_uk.title AND one_day_usa.domain
= one_day_uk.domain);
```

```
SELECT title, SUM(views_usa), SUM(views_uk), SUM(views_uk - views_usa) AS diff
FROM usa_uk
WHERE domain LIKE '%en%'
GROUP BY title
ORDER BY diff DESC
LIMIT 20;
```

Note: This same process was used in creating the table for usa\_aus.

```
hive>
Stage-Stage-1: Map: 14 Reduce: 15 Cumulative CPU: 237.5 sec HDFS Read: 367795608 HDFS Write: 116616670 SUCCESS
Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 7.0 sec HDFS Read: 116028930 HDFS Write: 1013 SUCCESS
Total MapReduce CPU Time Spent: 4 minutes 4 seconds 500 msec
OK

```

title	views_usa	views_uk	diff
FS_Networks	89244	293490	204246
Murder_of_Robert_McCartney	11952	194220	182268
Big_Muskie	55452	145482	90030
List_of_countries_by_GDP_(nominal)	38484	120534	82050
Jeff_Bridges	186606	256608	70002
Firass_Dirani	2718	68742	66024
Frankenstein_Castle	11274	71124	59850
Centenarian	55862	112518	57456
American_comic_book	4098	43878	39780
Robert_Redford	266478	301968	35490
Conor_Coady	30846	64986	34140
Rampage_(2018_film)	8718	40794	32076
Postcolonialism	2052	32598	30546
Daniel_Sans_(cricketer)	76524	102792	26268
Ken_McElroy	3996	26742	22746
Beau_Bridges	49748	70206	20458
SB19	25182	46968	21786
Aaron_Swartz	12870	34548	21678
Lloyd_Bridges	44634	65994	21360
Max_Kilnan	19656	40626	20970

```
20 rows selected (73.654 seconds)
0 jdbc:hive2://>
```

```
hive>
Stage-Stage-1: Map: 10 Reduce: 11 Cumulative CPU: 195.55 sec HDFS Read: 2578150372 HDFS Write: 100570247 SUCCESS
Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 6.62 sec HDFS Read: 100581460 HDFS Write: 1064 SUCCESS
Total MapReduce CPU Time Spent: 3 minutes 22 seconds 170 msec
OK

```

title	views_usa	views_aus	diff
Wyller_Murray	50286	500030	489744
Sisters_at_Heart	22152	458382	436230
Dancing_with_the_Stars_(American_season_29)	65382	406524	341142
Andy_Dalton	22446	255470	233028
Jeffrey_Tobin	540922	727166	192234
Budda_Baker	22656	170022	147366
Larry_Fitzgerald	9870	151110	141240
Wliff_Kingsbury	10206	104472	132266
Christhal_Stauss	23028	155992	132964
Chicago_Seven	203442	333042	129600
Abbie_Hoffman	128628	257538	128910
Jeff_Bridges	186606	307506	126900
Killing_in_the_Name	3066	116772	113706
Tom_Hayden	99486	209220	109734
Jennifer_Garner	23191	130735	107544
Patrick_Rehomes	20192	131408	107256
The_Haunting_of_Bly_Manor	276510	383262	106752
Robert_Redford	266478	372540	106062
Jeannie_Rai	14142	119880	105738
Skai_Jackson	14476	117528	101052

```
20 rows selected (66.028 seconds)
0 jdbc:hive2://>
```



## Question 5

Analyze how many users will see the average vandalized wikipedia article page before the offending edit is reversed.

- Big Long Create Table Statement that did nothing but create the table with all 70 empty fields
- Load Data
  - `LOAD DATA LOCAL INPATH`
  - `'/home/trevorbuck/project2/data/revisions/2020-10.enwiki.2020-10.tsv.bz2'`
  - `INTO TABLE revisions;`
- `CREATE TABLE revision_plus_views AS`
  - `SELECT`
    - `revisions.page_title AS title,`
    - `revisions.revision_seconds_to_identity_revert AS seconds_to_revert,`
    - `october20.views AS views`
  - `FROM revisions JOIN october20`
  - `ON (revisions.page_title = october20.title);`



# Results

Select Statement:

- `SELECT AVG(seconds_to_revert) AS seconds_average,`  
`AVG(views) AS views_average_per_day`
- `FROM revision_plus_views`
- `WHERE seconds_to_revert > 0;`

--Final Math

--  $\text{seconds\_average} * \text{views\_average\_per\_day} * (1 \text{ day} / 86400 \text{ second}) = x \text{ views before edit}$

--  $65287.89730011033 * 28.415237131249395 / 86400 =$   
 $21.47188754147329 \text{ views before edit}$

Answer: 21.47 views before edit

```
Total MapReduce CPU Time Spent: 2 minutes 5 seconds 950 msec  
OK
```

```
+-----+-----+  
| seconds_average | views_average_per_day |  
+-----+-----+  
| 65287.89730011033 | 28.415237131249395 |  
+-----+-----+  
1 row selected (35.172 seconds)  
0: jdbc:hive2://> |
```



## Question 6

Run an analysis you find interesting on the wikipedia sets we're using.

Which wikipedia articles were more popular throughout the months of the year?

Note: This process was repeated for each month of the year.

For January  
CREATE TABLE jan (  
prev STRING,  
curr STRING,  
type STRING,  
num INT  
)

ROW FORMAT DELIMITED  
FIELDS TERMINATED BY '\t';

LOAD DATA LOCAL INPATH  
'/home/trevor buck/project2/data/months/click  
stream-dewiki-2020-02.tsv.gz'  
INTO TABLE jan;

## Results (Feb)

- SELECT
  - feb.curr AS title,
  - SUM(feb.num - sept\_click.num) AS diff
- FROM feb JOIN sept\_click
- ON (feb.curr = sept\_click.curr)
- GROUP BY feb.curr
- ORDER BY diff DESC
- LIMIT 10;

Once again, repeated for each month

```
Total MapReduce CPU Time Spent: 4 minutes 12 seconds 150 msec
OK
+-----+-----+
|      title      |      diff      |
+-----+-----+
| Pornhub         | 194344226      |
| Kirk_Douglas    | 110968832      |
| Super_Bowl      | 85809219       |
| New_York_City   | 58526567       |
| Billie_Eilish   | 53046582       |
| Influenza       | 50939874       |
| Kobe_Bryant     | 50596787       |
| Bernie_Sanders  | 44691591       |
| Jennifer_Lopez  | 40316686       |
| Super_Bowl_LIV  | 38982930       |
+-----+-----+
10 rows selected (99.043 seconds)
0: jdbc:hive2://>
```



## Results By Month

January - Joaquin Pheonix, Patrick Mahomes

February - Super Bowl, Kobe Bryant, Influenza

March - Coronavirus, Spanish Flu, Kenny Rogers

April - Kim Jong Un, Joe Exotic, Covid19  
pandemic

May - Michael Jordan, Elon Musk, Covid19  
pandemic

June - Sushant Singh Rajput, George Floyd, Jeffrey  
Epstein

July - Grant Imahara, Alexander Hamilton,  
Lin-Manuel Miranda

August - Kamala Harris, Deaths in 2020, Beirut

September - Ruth Bader Ginsburg, Chadwick  
Boseman, Enola Holmes



**Thanks!**

