Project 1

Jordan Juel

Question 1

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

Prerequisites: Hourly data for October 20th loaded into tables.

Assumptions: Data analyzed involves only desktop data ‘en’ tag. Mobile views are not included. Page title 'Main_Page' and 'Special:Search' can be ignored

Question 1

Data from all hourly tables are UNION ALL'd into one table

Next a query is ran on that table to sum the views for the day

```
SELECT s.page_title, MAX(s.view_sum)
FROM(
SELECT page_title, sum(count_views)
view_sum
FROM ENWIKI
WHERE page_title != 'Main_Page' AND
page_title != 'Special:Search'
GROUP BY page_title)s
GROUP BY s.page_title
ORDER BY MAX(s.view_sum) DESC
LIMIT 1
```

Output where c1 is the sum of views

```
+--------------+---------+
| s.page_title |   _c1   |
+--------------+---------+
| Bible        | 148726  |
+--------------+---------+
```

Question 2

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

Prerequisites: Clickstream data loaded into a table named oct_clickstream, View count data from Q1.

Assumptions: View data from october 20th is an accurate representation of normal viewing patterns. Page title 'Main_Page' and 'Special:Search' can be ignored

Question 2

Only select data from rows with type ‘link’

Join view count table and divide count of links by total views to get a weight value

The greater the weight value the more likely a user will click on a link to another page.

```
SELECT * FROM
CLICKRATIO
ORDER BY click_ratio
DESC
LIMIT 5;
```

Output

```
+-----------------------------------------------+-------------------+
|                    prev                       |    click_ratio    |
+-----------------------------------------------+-------------------+
| Islam_in_the_Arctic                           | 8984.0            |
| P._Sai_Kumar                                  | 7937.0            |
| 2020_French_Open_–_Women's_Singles_Qualifying | 5794.5            |
| Dune_(2020_film)                              | 5316.190265486725 |
| Tennis_on_ESPN                                | 4946.0            |
+-----------------------------------------------+-------------------+
```

Question 3

What series of wikipedia articles, starting with Hotel California, keeps the largest fraction of its readers clicking on internal links?

Prerequisites: Clickstream data loaded into a table named oct_clickstream

Assumptions: Click ratio data from Q2 is valid.

Question 3

Join the clickstream and ratio data to generate a table of weights moving from one page to another

```
CREATE TABLE PATHWEIGHT
AS SELECT oct_clickstream.prev,
oct_clickstream.curr, clickratio.click_ratio
FROM oct_clickstream
INNER JOIN clickratio ON
oct_clickstream.curr = clickratio.prev
WHERE oct_clickstream.type = 'link'
```

Question 3

recursive queries are not possible in hive

Adds up the click ratio values to find the largest

```
SELECT path, (t1.newratio + t0.click_ratio) newratio
FROM PATHWEIGHT t0
INNER JOIN
(SELECT CONCAT (t3.path, "/", t2.curr) path, (t2.curr) currentpage, (t3.newratio + t2.clic
newratio FROM PATHWEIGHT t2
INNER JOIN (SELECT CONCAT(t5.prev,"/", t4.prev, "/", t4.curr) path,
(t4.curr) currentpage, (t4.click_ratio) newratio
FROM PATHWEIGHT t5 INNER JOIN PATHWEIGHT t4
ON t5.curr = t4.prev
WHERE t5.prev = "Hotel_California")t3
ON t3.currentpage = t2.prev)t1
ON t1.currentpage = t0.prev
ORDER BY newratio DESC
LIMIT 1;
```

```
+-----------------------------------------------+-------------------+
|                     path                      |     newratio      |
+-----------------------------------------------+-------------------+
| Hotel_California/Taxi_Driver/Academy_Award_for_Best_Actor/Timothée_Chalamet | 5461.007468817275  |
+-----------------------------------------------+-------------------+
```

Question 4

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

Prerequisites: Hourly data from October 20th is in tables based on hour

Assumptions: Wikipedia records are on PST, Waking hours are US 6am-10pm, UK 2pm- 6am, Australia 1am - 5pm PST.

Question 4

Use the hour data and time frames for waking hours to build tables for each region

Sum up view counts for each region table

If the difference is positive the article is more popular in the US, negative UK

```
Select t1.page_title, (t1.view_sum - t2.view_sum)
difference
from USWIKICOUNT t1
INNER JOIN UKWIKICOUNT t2
ON t1.page_title = t2.page_title
ORDER BY ABS(difference) DESC
LIMIT 5;
```

US vs UK

t1.page_title	difference
Wikipedia:Hauptseite	39704
Murder_of_Robert_McCartney	36131
Moise_Kean	32003
Sisters_at_Heart	-30661
Wikipédia:Accueil_principal	24741

Question 4

US vs AU

t1.page_title	difference
Moise_Kean	35847
Richard_Mansfield	28085
Sisters_at_Heart	-27360
Sophia_Thomalla	22774
Dancing_with_the_Stars_(American_season_29)	-21740

UK vs AU

t1.page_title	difference
Wikipedia:Hauptseite	-40282
Murder_of_Robert_McCartney	-33633
Richard_Mansfield	28788
Wikipédia:Accueil_principal	-26918
Jeffrey_Toobin	25264

Question 5

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

Prerequisites: View count from Q1, revision history loaded into a table named revisionhistory

Assumptions: Any page reverted was vandalized, October 20th data is an accurate representation of normal daily views

Question 5

Trim data from revisionhistory to get needed fields

Join trimmedrevisionhistory on itself to connect reverter id with reverted edit and calculate uptime of revision in days

Join on view data to multiply uptime by viewers per day for the page for each reverted edit

Average the values for estimated viewers

Question 5

```
CREATE TABLE REVISIONUPTIME
AS SELECT t1.id, t1.page_title, t1.reverted,
t1.reverter_id, t1.revisiondate editdate,
t2.revisiondate revertdate,
(((unix_timestamp(t2.revisiondate) -
unix_timestamp(t1.revisiondate))/60)/1440)
uptimeDays, t3.view_sum
FROM TRIMMEDREVISIONHISTORY t1
INNER JOIN TRIMMEDREVISIONHISTORY t2
ON t2.id = t1.reverter_id
INNER JOIN ENWIKICOUNT t3
ON t1.page_title = t3.page_title
```

```
SELECT AVG(t1.estimatedviews)
FROM
(
SELECT id, (uptimedays*view_sum)
estimatedviews FROM REVISIONUPTIME
)t1;
```

```
+-------------------+
|       _c0         |
+-------------------+
| 914.4948418680623  |
+-------------------+
```

Question 6

Which page has the most reverted edits in october 2020?

Prerequisites: Same as Q5

Assumptions: Any page reverted was vandalized, Q5 trimmedrevisionhistory is accurate

Question 6

Use the Trimmedrevisionhistory data from Q5 to find the page with the most unique revision ids that were reverted

```
SELECT page_title, COUNT(id)
edit_count
FROM TRIMMEDREVISIONHISTORY
WHERE reverter_id IS NOT NULL
GROUP BY page_title
ORDER BY edit_count DESC
LIMIT 4;
```

```
+-----------------------------------------+------------+
|                page_title               | edit_count |
+-----------------------------------------+------------+
| Administrator_intervention_against_vandalism | 8061       |
| Teahouse                                | 4158       |
| Usernames_for_administrator_attention   | 3396       |
| Administrators'_noticeboard/Incidents   | 3322       |
+-----------------------------------------+------------+
```

https://github.com/jordan-juel/Project1.git