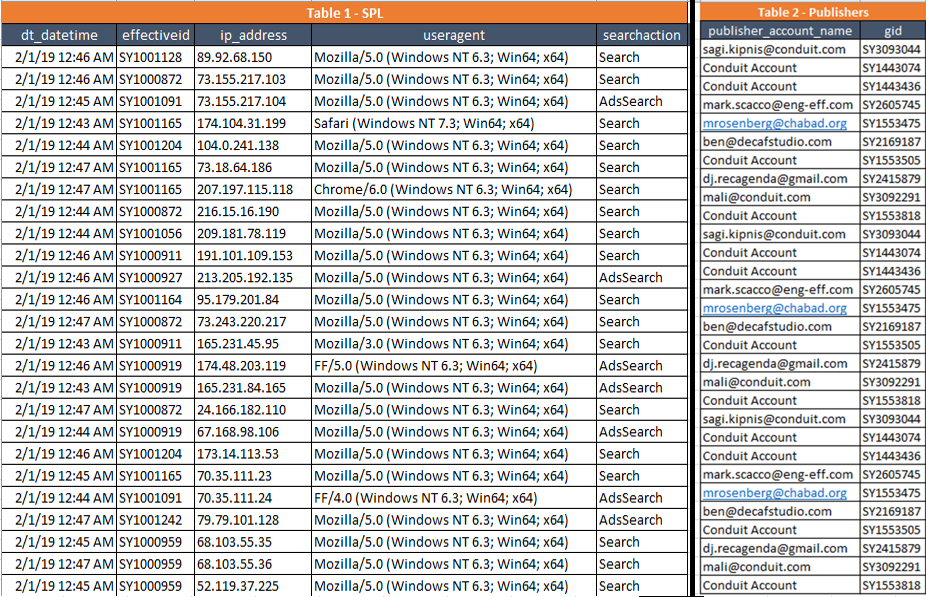
**SQL Exercise**

1. You want to load records from the staging table (SPL\_STG) to destination fact table (SPL).   
   The records in the stg table belong to different source files. You want to load records from source files you didn’t load before.  
   Write a query to check which source files from the stg table weren’t loaded to the fact table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SPL\_STG** | |  | **SPL** | |
| … | source\_file |  | … | source\_file |
| … | tb1 |  | … | ob1 |
| … | tb2 |  | … | ob2 |
| … | ob1 |  | … | tb1 |
| … | ob2 |  |  |  |

1. Display all the publishers (publishers might have more than one gid) where the searches came from ip’s belonging to user agents starting with Mozilla, mozilla, FF or ff.  
   Show only the publishers which had more than 50 searches.

\* effectiveid = gid  


1. Display the following metrics per day & hour:

* Searches (Search, AdsSearch)
* Clicks (AdsResultClick, SearchResultClick)
* Sponsored clicks (represented in link type under the values: SiteLink/Sponsored)
* Organic clicks (link type with no values)

Output should look like:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **dt** | **dt\_hour\_id** | **searches** | **clicks** | **sponsored\_clicks** | **organic\_clicks** |
| yyyymmdd | hh | int | int | int | int |

**SPL**

|  |  |  |
| --- | --- | --- |
| **dt\_datetime** | **searchaction** | **link\_type** |
| 2019-02-01 0:25 | Search |  |
| 2019-02-01 0:23 | AdsSearch |  |
| 2019-02-01 0:24 | AdsResultClick | Sponsored |
| 2019-02-01 0:24 | Search |  |
| 2019-02-01 0:25 | AdsResultClick | Sponsored |
| 2019-02-01 0:25 | AdsSearch |  |
| 2019-02-01 5:25 | Search |  |
| 2019-02-02 5:25 | AdsSearch |  |
| 2019-02-05 5:25 | SearchResultClick | Sponsored |
| 2019-02-01 0:23 | AdsSearch |  |
| 2019-02-01 0:24 | Search |  |
| 2019-02-01 0:24 | Search |  |
| 2019-02-01 3:25 | AdsResultClick | Sponsored |
| 2019-02-01 0:24 | Search |  |
| 2019-02-01 0:24 | AdsResultClick | Sponsored |
| 2019-02-01 0:24 | Search |  |
| 2019-02-01 0:24 | SearchResultClick | SiteLink |
| 2019-02-01 0:23 | Search |  |
| 2019-02-01 0:24 | Search |  |
| 2019-02-01 0:24 | SearchResultClick |  |
| 2019-02-01 0:22 | AdsResultClick | Sponsored |
|  |  |  |

1. Given a table:

|  |  |  |
| --- | --- | --- |
| **spl** | | |
| **Column** | **Type** | **Format** |
| dt | string | yyyymmdd |
| hour\_id | int | hh |
| gid | string |  |
| searches | int |  |

You need to calculate the total searches per date & hour, for each hour to show the accumulated searches until that hour and return the row (hour) with the 3rd most searches for that day.

For example after you calculate the accumulated searches you will get:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **dt** | **hour\_id** | **searches** | **acc\_searches** |  |
| 20200609 | 0 | 192688 | 192688 |  |
| 20200609 | 1 | 217735 | 410423 |  |
| 20200609 | 2 | 232725 | 643148 |  |
| 20200609 | 3 | 237026 | 880174 |  |
| 20200609 | 4 | 241570 | 1121744 |  |
| 20200609 | 5 | 265270 | 1387014 |  |
| 20200609 | 6 | 331929 | 1718943 |  |
| 20200609 | 7 | 347889 | 2066832 |  |
| 20200609 | 8 | 358254 | 2425086 | 2 |
| 20200609 | 9 | 369415 | 2794501 | 1 |
| **20200609** | **10** | **356227** | **3150728** | 3 |
| 20200609 | 11 | 342007 | 3492735 |  |
| 20200609 | 12 | 324909 | 3817644 |  |
| 20200609 | 13 | 299267 | 4116911 |  |
| 20200609 | 14 | 300405 | 4417316 |  |
| 20200609 | 15 | 247215 | 4664531 |  |
| 20200609 | 16 | 211621 | 4876152 |  |
| 20200609 | 17 | 223756 | 5099908 |  |
| 20200609 | 18 | 184896 | 5284804 |  |
| 20200609 | 19 | 166919 | 5451723 |  |
| 20200609 | 20 | 153827 | 5605550 |  |
| 20200609 | 21 | 143958 | 5749508 |  |
| 20200609 | 22 | 147960 | 5897468 |  |
| 20200609 | 23 | 161442 | 6058910 |  |

Your final output would be:

|  |  |  |  |
| --- | --- | --- | --- |
| **dt** | **hour\_id** | **searches** | **acc\_searches** |
| 20200609 | 10 | 356227 | 3150728 |

1. Write a query to aggregate the internal, external and payout measures with internal measures of searchprovider with type 2, external measures with report name (file\_description) “Bing Feed AdUnit” and payout measures with search\_provider\_id type 2.  
     
   You should display all the data from a table even if it is not in other tables.  
   You should get the following aggregate output structure with the data types:

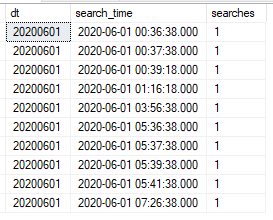
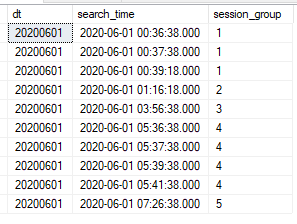
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| dt | hour\_id | device | country\_code | int\_searches | int\_clicks | ext\_searches | ext\_clicks | payout | provider |
| yyyymmdd | hh | string | string | int | int | int | int | double | ‘bing’ (string) |

Sample of raw data tables:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| internal\_data |  |  |  |  |  |  |
| dt | hour\_id | device | country\_code | searchprovider | searches | clicks |
| 20200809 | 1 | Desktop | CH | 2 | 3000 | 500 |
| 20200809 | 3 | Desktop | CL | 6 | 1008 | 20 |
| 20200809 | 8 | Mobile | TR | 4 | 133 | 1 |
| 20200809 | 8 | Desktop | LK | 4 | 46 | 1 |
| 20200809 | 9 | Mobile | GB | 2 | 11966 | 308 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| external\_data |  |  |  |  |  |  |
| dt | hour\_id | device | country\_code | file\_description | searches | clicks |
| 20200809 | 0 | Desktop | DE | Bing Feed AdUnit | 8393 | 1141 |
| 20200809 | 1 | Desktop | AU | Bing Feed Privado AdUnit | 19672 | 1076 |
| 20200809 | 1 | Mobile | DE | Bing Feed Privado AdUnit | 6225 | 876 |
| 20200809 | 3 | Mobile | GB | Bing Feed AdUnit | 6144 | 817 |
| 20200809 | 5 | Mobile | AU | Bing Feed AdUnit | 12620 | 736 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| payout |  |  |  |  |  |
| date | hour\_id | device | country\_code | search\_provider\_id | payout |
| 2020-08-09 | 0 | Mobile | FR | 4 | 7.16 |
| 2020-08-09 | 0 | Mobile | GB | 2 | 18.21 |
| 2020-08-09 | 1 | Mobile | NL | 2 | 0.15 |
| 2020-08-09 | 0 | Desktop | DE | 2 | 206.67 |
| 2020-08-09 | 1 | Desktop | CA | 4 | 5.13 |
| 2020-08-09 | 2 | Desktop | TH | 2 | 0 |

1. Given a search\_session table, we want to group searches to sessions. Assume that session is active if the time difference between searches is less than 5 minutes. The output should show for each search, the session he is related to.  
   For example for the input:  
     
   Output:  
   

Please find the following sample data for your assistance.  
  
CREATE TABLE search\_session(dt varchar(8),search\_time DATETIME, searches int)

INSERT INTO search\_session(dt,search\_time,searches)

SELECT 20200601,CAST('2020-06-01 00:36:38' AS datetime),1

UNION ALL

SELECT 20200601,CAST('2020-06-01 00:37:38' AS datetime),1

UNION ALL

SELECT 20200601,CAST('2020-06-01 00:39:18' AS datetime),1

UNION ALL

SELECT 20200601,CAST('2020-06-01 01:16:18' AS datetime),1

UNION ALL

SELECT 20200601,CAST('2020-06-01 03:56:38' AS datetime),1

UNION ALL

SELECT 20200601,CAST('2020-06-01 05:36:38' AS datetime),1

UNION ALL

SELECT 20200601,CAST('2020-06-01 05:37:38' AS datetime),1

UNION ALL

SELECT 20200601,CAST('2020-06-01 05:39:38' AS datetime),1

UNION ALL

SELECT 20200601,CAST('2020-06-01 05:41:38' AS datetime),1

UNION ALL

SELECT 20200601,CAST('2020-06-01 07:26:38' AS datetime),1

Good Luck!