[Download Statistics Daily Workings 4](#_Toc369767365)

[Monthly Ran Stored Procedures 4](#_Toc369767366)

[InsertMonthIntoProjectDownloadByDateRange 4](#_Toc369767367)

[Purpose of the Stored Procedure 4](#_Toc369767368)

[Variable 4](#_Toc369767369)

[Passed in 4](#_Toc369767370)

[Output 5](#_Toc369767371)

[Other Stored Procedures Called 5](#_Toc369767372)

[General outline and Algorithms and Formulas Used 5](#_Toc369767373)

[Daily Ran Stored Procedures 5](#_Toc369767374)

[UpdateProjectPopularity 5](#_Toc369767375)

[Purpose of the Stored Procedure 5](#_Toc369767376)

[Variable 5](#_Toc369767377)

[Passed in 5](#_Toc369767378)

[Output 5](#_Toc369767379)

[Other Stored Procedures Called 5](#_Toc369767380)

[General outline and Algorithms and Formulas Used 5](#_Toc369767381)

[Every 5 Minuet Ran Stored Procedures 6](#_Toc369767382)

[ProcessFileDownloadData 6](#_Toc369767383)

[Purpose of the Stored Procedure 6](#_Toc369767384)

[Variable 6](#_Toc369767385)

[Passed in 6](#_Toc369767386)

[Output 6](#_Toc369767387)

[Other Stored Procedures Called 6](#_Toc369767388)

[General outline and Algorithms and Formulas Used 6](#_Toc369767389)

[UpdateFileAndProjectCount 7](#_Toc369767390)

[Purpose of the Stored Procedure 7](#_Toc369767391)

[Variable 7](#_Toc369767392)

[Passed in 7](#_Toc369767393)

[Output 7](#_Toc369767394)

[Other Stored Procedures Called 7](#_Toc369767395)

[General outline and Algorithms and Formulas Used 7](#_Toc369767396)

[UpdateProjectDownloadsLastThirtyDays 7](#_Toc369767397)

[Purpose of the Stored Procedure 7](#_Toc369767398)

[Variable 7](#_Toc369767399)

[Passed in 7](#_Toc369767400)

[Output 7](#_Toc369767401)

[Other Stored Procedures Called 7](#_Toc369767402)

[General outline and Algorithms and Formulas Used 7](#_Toc369767403)

[Other Stored Procedures 7](#_Toc369767404)

[GetProjectPopularitiesByDateModified 8](#_Toc369767405)

[Purpose of the Stored Procedure 8](#_Toc369767406)

[Variable 8](#_Toc369767407)

[Passed in 8](#_Toc369767408)

[Output 8](#_Toc369767409)

[Other Stored Procedures Called 8](#_Toc369767410)

[GetProjectCountsByDateModified 8](#_Toc369767411)

[Purpose of the Stored Procedure 8](#_Toc369767412)

[Variable 8](#_Toc369767413)

[Passed in 8](#_Toc369767414)

[Output 8](#_Toc369767415)

[Other Stored Procedures Called 8](#_Toc369767416)

[GetFileCountsByDateModified 8](#_Toc369767417)

[Purpose of the Stored Procedure 8](#_Toc369767418)

[Variable 8](#_Toc369767419)

[Passed in 8](#_Toc369767420)

[Output 8](#_Toc369767421)

[Other Stored Procedures Called 8](#_Toc369767422)

[Schema 9](#_Toc369767423)

[File Table 9](#_Toc369767424)

[Table Definition 9](#_Toc369767425)

[Columns 9](#_Toc369767426)

[Keys 9](#_Toc369767427)

[Constraints 9](#_Toc369767428)

[Triggers 9](#_Toc369767429)

[Indexes 9](#_Toc369767430)

[FileDownload Table 9](#_Toc369767431)

[Table Definition 9](#_Toc369767432)

[Columns 9](#_Toc369767433)

[Keys 10](#_Toc369767434)

[Constraints 10](#_Toc369767435)

[Triggers 10](#_Toc369767436)

[Indexes 10](#_Toc369767437)

[Project Table 10](#_Toc369767438)

[Table Definition 10](#_Toc369767439)

[Columns 10](#_Toc369767440)

[Keys 10](#_Toc369767441)

[Constraints 10](#_Toc369767442)

[Triggers 11](#_Toc369767443)

[Indexes 11](#_Toc369767444)

[ProjectDownloadsByDateRange 11](#_Toc369767445)

[Table Definition 11](#_Toc369767446)

[Columns 11](#_Toc369767447)

[Keys 11](#_Toc369767448)

[Constraints 11](#_Toc369767449)

[Triggers 11](#_Toc369767450)

[Indexes 12](#_Toc369767451)

[Record 12](#_Toc369767452)

[Table Definition 12](#_Toc369767453)

[Columns 12](#_Toc369767454)

[Keys 12](#_Toc369767455)

[Constraints 12](#_Toc369767456)

[Triggers 12](#_Toc369767457)

[Indexes 12](#_Toc369767458)

[Referrer 12](#_Toc369767459)

[Table Definition 12](#_Toc369767460)

[Columns 12](#_Toc369767461)

[Keys 12](#_Toc369767462)

[Constraints 12](#_Toc369767463)

[Triggers 12](#_Toc369767464)

[Indexes 13](#_Toc369767465)

[TableAggregateID 13](#_Toc369767466)

[Table Definition 13](#_Toc369767467)

[Columns 13](#_Toc369767468)

[Keys 13](#_Toc369767469)

[Constraints 13](#_Toc369767470)

[Triggers 13](#_Toc369767471)

[Indexes 13](#_Toc369767472)

[UserAgent 13](#_Toc369767473)

[Table Definition 13](#_Toc369767474)

[Columns 13](#_Toc369767475)

[Keys 13](#_Toc369767476)

[Constraints 13](#_Toc369767477)

[Triggers 13](#_Toc369767478)

[Indexes 14](#_Toc369767479)

# Download Statistics Daily Workings

## Monthly Ran Stored Procedures

### InsertMonthIntoProjectDownloadByDateRange

#### Purpose of the Stored Procedure

This procedure will count and store the raw total downloads and the unique total downloads for the previous month. It can do this any time in the current month that it is ran or can be modified to take in one or two parameters and calculate a different time frame (the table it stores the data in will need a change to its current unique key).

This is called from UpdateProjectPopularity on the first day of every month.

#### Variable

##### Passed in

None.

##### Output

None.

#### Other Stored Procedures Called

None.

#### General outline and Algorithms and Formulas Used

The first thing it does is gets the first day of the month that it is going to get totals for then it gets the last day of the month (or time frame) that it is going to get counts for. In the where clause it adds a day to the last day of the time frame and does a less than as to make sure it gets all downloads in that time frame.

## Daily Ran Stored Procedures

The one Stored Procedure is run once a day at about 01:17 UTC by a job in SQL server DLS\_UpdateProjectPopularity.

### UpdateProjectPopularity

#### Purpose of the Stored Procedure

This procedure updates each Projects Popularity by re-calculating it’s score, rank and how many unique downloads it has had in the past thirty days. If it is the first of the month it will call another Stored Procedure to calculate and store the previous months download counts in another table.

#### Variable

##### Passed in

None

##### Output

Just the regular return integer not overridden in the code.

#### Other Stored Procedures Called

InsertMonthIntoProjectDownloadByDateRange is called on the first of the month to calculate the previous months Project download totals and store them in the Table ProjectDownloadsByDateRange.

#### General outline and Algorithms and Formulas Used

First thing that is done is a count on the unique downloads of a Project for the past 30 days. That is stored (with and update) into the Project table. The work horse query is:

**SELECT F.ProjectID, COUNT(DISTINCT R.IpAddress) AS UniqueDownloadsLastThirtyDays**

**FROM [dbo].Record AS R**

**LEFT JOIN [dbo].[File] AS F ON R.FileID = F.ID**

**WHERE R.[Date] >= DATEADD(DAY, -31, @currDate) and R.[Date] < @currDate**

**GROUP BY F.ProjectID**

The second thing that is done is the calculation of the each project’s popularity score. This is based on how many times one of the Project’s Files has been downloaded by a unique IP Address (multiple downloads from the same IP Address only count as one download). There are seven time segments use with a decreasing impact on the score.

First time segment is the past 30 days from the current day. The other six segments are the past full six months (this is where the Table ProjectDownloadsByDateRange comes in handy, this was being calculated each day for the past six months but now this data is stored in a table and has significantly improved its run time).

The algorithm is Where UDC is the Unique Download Count for that indexes time frame, Index (zero based) is the segment number. So Index 0 would be the past 30 days and the 1 through 6 index would be the past six months in descending order.

The Final thing it does is calculate the Popularity ranking and Game Popularity Ranking. These are just the Popularity score in a rank (ordered by score highest to lowest). For Project rank it is just each individual project and for the Game rank it is the total of the Game’s Projects scores in ranking order.

## Every 5 Minuet Ran Stored Procedures

Three Stored Procedures are ran about every 5 minutes by the SQL Server job DLS\_ProcessesDataAndUpdateCounts. The only order that is needed is the ProcessFileDownloadData procedure goes before the other two. The other two Stored Procedures do separate counts that are not needed by the other.

### ProcessFileDownloadData

#### Purpose of the Stored Procedure

This will process a batch of files from the FileDownload Table into the Record Table, Referrer Table and the UserAgent Table. See outline for what goes where.

#### Variable

##### Passed in

None.

##### Output

Just the regular return integer not overridden in the code.

#### Other Stored Procedures Called

None.

#### General outline and Algorithms and Formulas Used

No algorithms used just data moved from one Table to others.

The Referrer if not already in the Referrer table gets inserted into that Table, then the ID for that matching row is used in the insert for the Record Table.

The UserAgent if not already in the UserAgent Table get inserted into that table, then the ID for that match row is used in the insert for the record Table.

The ProjectFileID (FileID), DateDownloaded, IPAddress, ID of the matching row of the Referrer, and the ID of the matching row for the UserAgent gets inserted into the Record Table.

This Stored Procedure does take advantage of the TableAggregateID Table because in the process of moving data from this table to the Record Table more Rows are being inserted.

### UpdateFileAndProjectCount

#### Purpose of the Stored Procedure

This Stored Procedure updates the total download counts for Files and Projects.

#### Variable

##### Passed in

None.

##### Output

None.

#### Other Stored Procedures Called

None.

#### General outline and Algorithms and Formulas Used

The Table TableAggregateID is used to determine the last Record Row that was added to the total download counts of the Files. These are just the raw download counts not the unique counts. The Project download counts are gotten by summing the totals from the Files totals that are in that Project.

### UpdateProjectDownloadsLastThirtyDays

#### Purpose of the Stored Procedure

This is to update the Projects count for download in the past 30 days.

#### Variable

##### Passed in

None.

##### Output

None.

#### Other Stored Procedures Called

None.

#### General outline and Algorithms and Formulas Used

The counts are form the Record Table and are grouped by their Project. The date it goes back to is 30 day ago from the day it is ran but does not have a time stamp on it so it counts all of the 30th day not a portion of it.

## Other Stored Procedures

These are called by the Service to update its cache.

### GetProjectPopularitiesByDateModified

#### Purpose of the Stored Procedure

This gets the Popularity Score, rank and GameRank of the Projects if they have been modified since the date/time it was passed.

#### Variable

##### Passed in

A Datetime that will be used to filter the results by the DatePopularityModified column.

##### Output

The query results.

#### Other Stored Procedures Called

None.

### GetProjectCountsByDateModified

#### Purpose of the Stored Procedure

This gets the total download and the last 30 day download count for the Projects that the date modified is greatest than the date passed in.

#### Variable

##### Passed in

A Datetime that will be used to filter the results by the DatePopularityModified column.

##### Output

The query results.

#### Other Stored Procedures Called

None.

### GetFileCountsByDateModified

#### Purpose of the Stored Procedure

This gets the total download count for the Files that the date modified is greatest than the date passed in.

#### Variable

##### Passed in

A Datetime that will be used to filter the results by the DatePopularityModified column.

##### Output

The query results.

#### Other Stored Procedures Called

None

## Schema

### File Table

#### Table Definition

Definition of File is a version of a project in JAR form. In other words when a User uploads the latest version of their project they do not upload all the separate file they upload it in a compressed form that we then call a File and the current release version of that Project File is typically the Default File.

This Table holds how many time each File has been downloaded.

#### Columns

1. ID -- Primary Key
2. ProjectID -- this is the ID of the Project from the Postgres database
3. TotalDownloads – This is how many times the File has been downloaded the Stored Procedure UpdateFileAndProjectCounts updates this number
4. HistoricalDownloads – This is how many times the File was downloaded in the pre-postgress database, static number will not change
5. DateModified -- this is the date/time the entry was changed/updated

#### Keys

The Normal PK.

#### Constraints

1. DF\_File\_DateModified – default value is GetUTCDate()
2. DF\_File\_HistoricDownloads – default value is 0
3. DF\_File\_TotalDownloads -- default value is 0

#### Triggers

None

#### Indexes

1. IX\_ID\_DateModified – Non-Unique,Non-Clustered on DateModified and ID

### FileDownload Table

#### Table Definition

This Table is the temp holding for the recording of the fact a File was downloaded later a Stored Procedure (ProcessFileDownloadData) runs and transitions the data to Tables Record, Referrer and UserAgent.

#### Columns

1. ID – Primary Key, Auto Incremented
2. ProjectFileID – This is the projects’s File ID
3. DateDownloaded – This is the date and time the File was downloaded
4. Referrer – If Known this is the page that sent them to the download
5. UserAgent – what program and OS downloaded the File
6. IPAddress – The IP Address of the connection that downloaded the File
7. Status – 1:New;2:Processing;3:Processed

#### Keys

The Normal PK.

#### Constraints

1. DF\_FileDownload\_Status – Default is 1

#### Triggers

None

#### Indexes

1. IX\_ProjectFileDownload\_Status -- Non-Unique,Non-Clustered On Status

### Project Table

#### Table Definition

This Table holds information on the individual Projects.

#### Columns

1. ID -- Primary Key
2. GameID – This is the PK of the Game the Project is for
3. PopularityScore – This is a score calculated daily with Stored Procedure UpdateProjectPopularity, it is based on how many downloads the Files in the Project have been download, see the Stored Procedures detail for exact algorithm
4. PopularityRank – This is a rank calculated daily with Stored Procedure UpdateProjectPopularity, it is and ordering by popularity score on each project
5. GamePopularityRank – This is a rank calculated daily with the Stored Procedure UpdateProjectPopularity, it is an ordering of the popularity score group by the all the project in a game.
6. DownloadsLastThirtyDays – This is how many times in the past 30 day the Project’s Files have been downloaded the Stored Procedure UpdateFileAndProjectCounts updates this number
7. UniqueDownloadsLastThirtyDays – This is how many times in the past 30 days a Project’s Files have been downloaded from an unique IP Address the Stored Procedure UpdateProjectPopularity updates this number
8. TotalDownloads – This is how many times the Project’s Files have been downloaded the Stored Procedure UpdateFileAndProjectCounts updates this number
9. DateModified – This is the date and time the row was last modified not including columns PopularityScore, PopularityRank, GamePopularityRank, UniqueDownloadsLastThirtyDays
10. DatePopularityModified – This is the last date and time that these columns were updated: PopularityScore, PopularityRank, GamePopularityRank, UniqueDownloadsLastThirtyDays
11. HistoricalDownloads -- This is how many times the Project’s Files were downloaded in the pre-postgress database, static number will not change

#### Keys

The Normal PK.

#### Constraints

1. DF\_Project\_DateModified -- default value is GetUTCDate()
2. DF\_Project\_DownloadsLastThirtyDays – default value is 0
3. DF\_Project\_GamePopularityRank – default value is 0
4. DF\_Project\_HistoricalDownloads – default value is 0
5. DF\_Project\_PopularityRank – default value is 0
6. DF\_Project\_PopularityScore – default value is 0
7. DF\_Project\_TotalDownloads – default value is 0
8. DF\_Project\_TotalDownloads – default value is 0

#### Triggers

None.

#### Indexes

1. IX\_ID\_DataModified -- Non-Unique,Non-Clustered on DateModified and ID

### ProjectDownloadsByDateRange

#### Table Definition

This table holds the monthly download totals for each project for a specified time frame. The total downloads and Unique total downloads for the specified time frame. Right now the time frame is an individual month but the table can be used for any time frame (a change to the UK will need to be made to include the ToDate if it will be used for other time frames, also anything that currently queries on this should start including the Todate in the where clause). The Stored Procedure InsertMonthIntoProjectDownloadByDateRange inserts the data into this Table. There should never be any duplicate date in this Table.

#### Columns

1. ID – Primary Key, Auto Incremented
2. ProjectID – This is the ID of the Project from the Postgres database
3. FromDate – This is the start date of the time frame, start is morning 00:00:00
4. ToDate – This is the end date of the time frame, end includes all of the day 23:59:59:999
5. TotalDownloads – This is the total amount of time the Project’s Files were download during the specified time frame
6. UniqueTotalDownloads – This is the total amount of time the Project’s Files were download from an unique IP Address during the specified time frame

#### Keys

The Normal PK.

#### Constraints

None.

#### Triggers

None.

#### Indexes

1. IX\_ProjectDownloadsByDateRange\_FromDate\_ProjectID -- Unique,Non-Clustered on FromDate and ProjectID (good for now with table just holding month time frames but if it starts storing other time frames then this should include the ToDate)

### Record

#### Table Definition

This Tables holds information on each time a File is downloaded.

#### Columns

1. ID – Primary Key, Auto Incremented
2. FileID – This is the ID of the File from the Postgres database
3. ReferrerID – This is the PK of the Referrer Table
4. UserAgentID – This is the PK of the UserAgentTable
5. Date – This is the date and time that the given File was downloaded
6. IpAddress – This is the address that the File was downloaded from

#### Keys

The normal PK.

#### Constraints

None.

#### Triggers

None.

#### Indexes

1. IDX\_FileID\_Date\_IpAddress -- Non-Unique,Non-Clustered on FileId, Date, IpAddress

### Referrer

#### Table Definition

This Table holds information about where user came form to get the download.

#### Columns

1. ID – Primary Key, Auto Incremented
2. Referrer – The web URL of where the user came form to get the download

#### Keys

The normal PK.

#### Constraints

None.

#### Triggers

None.

#### Indexes

Normal one.

### TableAggregateID

#### Table Definition

This Table holds information of what was the last row ID that was processed by a given Stored Procedure. It is where to store the last ID that was processed by a given procedure.

#### Columns

1. ID – Primary Key, Auto Incremented
2. OnTableName – This is the name of the Table of who’s row you are tracking the last row that was processed
3. TrackedRowID – This is the last Id that was processed by the given Stored Procedure for the given Table
4. ForSproc – This is the Stored Procedure name that need to know what row ID it last processed.

#### Keys

The normal ID.

#### Constraints

1. DF\_TableAggregateID\_DLLE\_ID – Default is 0.

#### Triggers

None

#### Indexes

1. NonClusteredIndex-20130612-084147 -- Non-Unique,Non-Clustered on all columns

### UserAgent

#### Table Definition

This table holds information on the program and OS that the person was using when they downloaded the file.

#### Columns

1. ID – Primary Key, Auto Incremented
2. UserAgent – the platform the user was on when they downloaded the File.

#### Keys

The normal one.

#### Constraints

None.

#### Triggers

None.

#### Indexes

Normal One only.