Data_Sources_and_Data_flow

Last modified 4 months ago by sharab.

What is the wiki about?

This wiki provides info about the technical data flow of HeadCT from share point to Redshift to Quick Sight.

Data source

Data source 1: Share point forms:

Share point form consists of data of input request from finance and business user for various requests. Details about these forms and business logics can be found in wiki https://w.amazon.com/bin/view/Advertising_Finance/Central_FP&A/HeadCT/HeadCT_Guide/

https://share.amazon.com/sites/adv_fin/_layouts/15/start.aspx#/Lists/Headcount_Transfers/AllItems.aspx
https://share.amazon.com/sites/adv_fin/_layouts/15/start.aspx#/Lists/Long_Leaves/AllItems.aspx
https://share.amazon.com/sites/adv_fin/_layouts/15/start.aspx#/Lists/Bulk_HC_Updates/AllItems.aspx
https://share.amazon.com/sites/adv_fin/_layouts/15/start.aspx#/Lists/Pull_Forward/AllItems.aspx

Loading this data to redshift is currently 2 sept process using the Excel based tool in the drive. https://drive.corp.amazon.com/personal/sharab/HeadCT/Macro%20Redshhift%20uploader?filter=none

Step 1: Click on refresh all button in data tab

Step 2: Click on ‘Click here to upload button’ to load the data into redshift.

Data source 2: Redshift HR BI data:

HR BI data contain of active Butts in seats, pending stats and req data, along with historical data. The data from HR BI is shared from recruiting team analyst davlee@: tharasan@ using data net jobs and they are loaded into a simple table and the snapshot table as per below table on daily cadence.

Schema	Table	Source	Transform Job	PK	New Load Job	Snapshot

headct	ads_bis	nextgen(HRBI)	7100297	employee_id	577844	588706
headct	ads_tranfers	nextgen(HRBI)	7257995	empl_id	572900	
headct	ads_pending_starts	nextgen(HRBI)	7120196	candidate_icims_id	577847	588707
headct	ads_all_reqs_phoenix	Phoenix	7353216	icims_job_id	588521	588708
headct	ww_ads_bis_historical	nextgen(HRBI)	7366320	none	607201	

Data source 3: PID Mapping:

PID Mapping is the main list total PID planned and is also consist information about position budget, funding Kotas directs, cost center and hierarchy data for NYO. It is also contains crucial mapping data where ever the system couldn't map the details. I.e. if a PID data is not present in justification data, then it is needed to manually map them using PID mapping tables.

Steps to get the latest data and load it back to redshift.

Step 1: Run a simple query **select * from headct.head_count_tracking** in workbench.

Sept 2: Copy it to excel and make the necessary modification.

Note: ensure position type is set accordingly, i.e. if an employee id is mapped then BIS, if REQ is mapped then REQ else NYO.

Once all the updates are made, save it as a csv with date format 'YYYY/MM/DD' in the column and file name as 'Op2_main_PIDS_MM_DD' then follow the below septs

1) Go to https://access.amazon.com/

2) Click on Conduit AWS Accounts

3) Click on DA-Finance-BI

4) Click on 'Access AWS Console' next to role 'S3ReadWrite'

5) Search & Click on S3

6) Click on da-finance-bi-s3-bucket >> adhoc >>headct folder.

7) Find the latest ‘Op2_main_PIDS’ file using the last modified date.

8) Download the data and do the bulk update and push forward the request**. (This part is curial)**

9) Ensure no pids are tagged to multiple employee id using the dashboard. I.e. PID could already be used the justification

11) Upload the data back into S3 folder

12) Once uploaded, open job: https://datanet.amazon.com/dw-platform/servlet/dwp/template/EtlViewExtractJobs.vm/job_profile_id/7548718

13) Click on edit profile

14) Change the file name with the latest name updated (Op2_main_PIDS_MM_DD) and click save

15) Run the job by choosing the yesterday as run_date and wait for the job to be successful.

Main HeadCT Head Count Tracking high level logic and jobs to run:

Schema	Table	Transform Job	Load Job	Description
headct	head_count_tracking_v1	7343520	587125	Main job where all the transformation happens where primary table is PID_mapping
headct	head_count_tracking_snapshot		587100	A table to store the daily snapshot of the main table
headct	head_count_tracking_v1_phoenix	7415673	639386	Similar table to main table, however with PID_mapping is not primary table. This table included details even if there is no PID mapping, i.e. backfill, leaves and no maps. Also used for audit table
headct	change_log	7374753	607318	This table store difference data between each snapshot.
headct	pid_missing_audit	7539253		Audit table which highlights all the missing pids or dups to team to fix the issues.
headct	head_count_tracking_total	7557163		Final table combination of Main table along with

				backfills, on_leave and only BIS without PIDs
headct	Gross_net_heads	7415736		
Headct	Attrition			
Headct	Month_end_tables	7670058		

High level logic:

The main table goal is tag a PID for each BIS (Butts in Seat), PS (Pending start), Reqs an NYO (Not yet Open) and stack them together from multiple tables and remove any duplicates Reqs. I.e. there could a BIS entry, but at the same time its PS or Req could be still open. Such entries need to be eliminated.

Key logic used.

1. PID is extracted from Reqs justification. If PID info is not present, then tagged using manual PID mapping table.
2. The employee is tied as backfill if there is open backfill Req and used the same PID or using manual PID mapping.
3. Dups are removed using the prioritization order as follow
    1.
        i. 'Butts in Seat'
        ii. 'Pending_starts'
        iii. 'Req Filled'
        iv. 'Accepted Offer'
        v. 'Req Approved'
        vi. 'Req Opened'
        vii. 'Req Eliminated'
        viii. 'Req Pooling'
        ix. 'NYO'
4. Data from SharePoint is joining the final table and loading data into relevant columns.

Audit checks:

In an ideal world we need to have one PID mapped to one active BIS and no duplicates. I.e. same PID shared with multiple BIS or 1 BIS having multiple PIDS tagged. To avoid such cases we proactively created a table: pid_missing_audit to highlight such issues.