**Saviynt Performance related Analytics**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | ANALYTICSNAME | CATEGORY | ANALYTICSQRY |
| 1 | UserImport\_SuccessRate | Admin | SELECT DISTINCT  JOBNAME AS JOB\_NAME, IFNULL(TRIGGERNAME, 'User Upload From UI') AS TRIGGER\_NAME, IFNULL(EXTERNALCONNECTION, '') AS JOB\_TRIGGER, CASE SAVRESPONSE WHEN 'Success' THEN 'Success' ELSE 'Failed' END AS STATUS\_OF\_JOB, JOBSTARTDATE AS STARTDATE\_OF\_JOB, JOBENDDATE AS ENDDATE\_OF\_JOB  FROM ecmimportjob  WHERE JOBNAME IN ('UserImportJob', 'UserImportFullJob', 'SchemaUserJob') AND JOBSTARTDATE >= (UTC\_DATE() - INTERVAL 30 DAY) |
| 2 | UserImport\_RuleRunEvaluationSuccessRate | Admin | SELECT DISTINCT  JOBNAME AS JOB\_NAME, SAVRESPONSE AS STATUS\_OF\_JOB, JOBSTARTDATE AS STARTDATE\_OF\_JOB, JOBENDDATE AS ENDDATE\_OF\_JOB  FROM ecmimportjob  WHERE JOBNAME LIKE 'RuleRunJob'  AND JOBSTARTDATE >= (UTC\_DATE() - INTERVAL 30 DAY) |
| 3 | SODEvaluationJob\_SuccessRate | SOD | SELECT DISTINCT JOBNAME AS JOB\_NAME, TRIGGERNAME AS TRIGGER\_NAME, EXTERNALCONNECTION AS CONNECTION\_NAME, SAVRESPONSE AS STATUS\_OF\_JOB, JOBSTARTDATE AS STARTDATE\_OF\_JOB, JOBENDDATE AS ENDDATE\_OF\_JOB FROM ecmimportjob WHERE JOBNAME LIKE 'RiskSODEvaluationJob' AND JOBSTARTDATE >= (UTC\_DATE() - INTERVAL 30 DAY) |
| 4 | Provisioning\_Top5ApplicationsByTasks | Admin | select a.TASKKEY, a.TASKTYPE, e.endpointname AS APPLICATION  from arstasks a join endpoints e on a.endpoint=e.endpointkey and e.displayname is not null  join (select count(\*) as totalCount, a.endpoint as EP from arstasks a group by a.endpoint order by totalCount desc limit 5) AS TEMP ON TEMP.EP=a.endpoint |
| 5 | Provisioning\_PendingTasks | Admin | SELECT 'Pending' AS STATUS\_OF\_TASK, CASE WHEN (UTC\_TIMESTAMP() - INTERVAL 12 HOUR) >= TASKDATE THEN 'More than 12 Hours' WHEN (UTC\_TIMESTAMP() - INTERVAL 8 HOUR) >= TASKDATE THEN 'More than 8 Hours' WHEN (UTC\_TIMESTAMP() - INTERVAL 4 HOUR) >= TASKDATE THEN 'More than 4 Hours' ELSE 'Less than 4 hours' END AS TASK\_PENDING\_FOR, endpoints.DISPLAYNAME AS APPLICATION  FROM arstasks  JOIN endpoints ON arstasks.endpoint=endpoints.endpointkey  WHERE arstasks.STATUS=1  AND arstasks.TASKTYPE IN (1,2,3,4,6, 8, 12, 14, 31) |
| 6 | Provisioning\_FailuresPerApplication | Admin | SELECT DISTINCT t.taskkey AS 'TASK ID', u.username AS USERNAME, u.DISPLAYNAME as 'USER NAME', u.email AS 'USER EMAIL', ep.displayname AS 'APPLICATION', t.accountname AS 'ACCOUNT NAME', enttype.entitlementname as 'ENTITLEMENT', ev.entitlement\_value AS 'ENTITLEMENT VALUE',ev.DISPLAYNAME AS 'ENTITEMENTVALUE DISPLAYNAME', t.source AS 'TASK SOURCE', CASE WHEN t.source = 'REQUEST' and t.REQUESTKEY is not null THEN (select substring\_index(JBPMPROCESSINSTANCEID,'.',-1) from ars\_requests where requestkey = t.REQUESTKEY) WHEN t.source in ('PROVRULE','ZERODAY') THEN (select NAME from hanarule where HANARULEKEY = t.ASSIGNEDFROMRULE) WHEN t.source in ('CERTIFICATION') THEN (select cert\_name from certification where CERTKEY = t.sourceid) WHEN t.source in ('ANALYTICS\_V2') THEN (select ANALYTICSNAME from analyticsconfiges where ANALYTICSKEY = t.sourceid) WHEN t.source in ('ANALYTICS','ANALYTICS\_V1') THEN (select ANALYTICSNAME from analyticsconfig where ANALYTICSKEY = t.sourceid) ELSE t.sourceid END AS 'TASK SOURCE DETAILS', t.comments as 'TASK COMMENTS', case when t.tasktype = 1 then 'Add Access' when t.tasktype = 2 then 'Remove Access' when t.tasktype = 3 then 'New Account' when t.tasktype = 4 then 'Role' when t.tasktype = 5 then 'Change Password' when t.tasktype = 6 then 'Enable Account' when t.tasktype = 7 then 'Claim Account' when t.tasktype = 8 then 'Delete Account' when t.tasktype = 9 then 'Update User' when t.tasktype = 12 then 'Update Account' when t.tasktype = 13 then 'Proposed Entitlement Owner' when t.tasktype = 14 then 'Disable Account' when t.tasktype = 23 then 'Modify Privilege' when t.tasktype = 24 then 'Create Entitlement' when t.tasktype = 25 then 'Add Access Entitlement' when t.tasktype = 26 then 'Remove Access Entitlement' when t.tasktype = 27 then 'Update Entitlement' when t.tasktype = 28 then 'Delete Entitlement' when t.tasktype = 29 then 'Grant Firefighter ID' when t.tasktype = 30 then 'Revoke Firefighter ID' else t.tasktype end as TASKTYPE, CASE t.status WHEN 1 THEN 'NEW' WHEN 2 THEN 'INPROGRESS' WHEN 3 THEN 'COMPLETE' WHEN 4 THEN 'DISCONTINUE' WHEN 5 THEN 'PENDING\_CREATE' WHEN 6 THEN 'PENDING\_PROVISION' WHEN 7 THEN 'PROVISION\_FAILED' WHEN 8 THEN 'ERROR' WHEN 9 THEN 'NO ACTION NEEDED' ELSE t.status END AS 'TASK\_STATUS', t.taskdate AS 'TASK CREATED ON', t.updatedate AS 'TASK COMPLETED ON', t.PROVISIONINGCOMMENTS AS 'TASK PROVISIONING COMMENTS' FROM endpoints ep INNER JOIN arstasks t ON t.endpoint = ep.endpointkey INNER JOIN users u ON t.userkey = u.userkey LEFT OUTER JOIN entitlement\_values ev ON t.entitlement\_valuekey = ev.entitlement\_valuekey LEFT OUTER JOIN entitlement\_types enttype ON ev.entitlementtypekey = enttype.entitlementtypekey WHERE cast(t.taskdate as DATE) >= (UTC\_DATE() - INTERVAL 30 DAY) AND t.STATUS IN (4,8) |
| 7 | ProvisioningJob\_SuccessRate | Admin | SELECT DISTINCT JOBNAME AS JOB\_NAME, TRIGGERNAME AS TRIGGER\_NAME, EXTERNALCONNECTION AS CONNECTION\_NAME, SAVRESPONSE AS STATUS\_OF\_JOB, JOBSTARTDATE AS STARTDATE\_OF\_JOB, JOBENDDATE AS ENDDATE\_OF\_JOB FROM ecmimportjob WHERE JOBNAME LIKE 'WSRetryJob' AND JOBSTARTDATE >= (UTC\_DATE() - INTERVAL 30 DAY) |
| 8 | ApplicationImport\_OrphanVsCorrelated | Admin | select e.displayname as APPLICATION, a.NAME, a.ACCOUNTID, a.STATUS AS STATUS\_OF\_ACCOUNT, case when ua.USERKEY IS NOT NULL THEN 'Correlated' ELSE 'Orphan' END AS Type from accounts a join endpoints e ON a.endpointkey =e.endpointkey left join user\_accounts ua ON a.accountkey=ua.accountkey where a.status not in ('SUSPENDED FROM IMPORT SERVICE', 'InActive', 'Inactive', '2', 'Manually Suspended', '0') and e.displayname is not null and e.displayname !='' |
| 9 | CCA\_CC\_User\_Manager\_Campaigns\_Created | Admin | SELECT cm.campaign\_name AS 'Campaign Name', u.SYSTEMUSERNAME AS 'Campaign Owner', DATE\_FORMAT(cm.start\_date,"%d-%b-%Y") AS 'Campaign Starting On', DATE\_FORMAT(cm.end\_date,"%d-%b-%Y") AS 'Campaign Ending On', CASE cm.campaign\_type WHEN '1' THEN 'ENTITLEMENTOWNER' WHEN '2' THEN 'USERMANAGER' WHEN '3' THEN 'SELF\_CERTIFICATION' WHEN '5' THEN 'ROLEOWNER' WHEN '6' THEN 'SERVICEACCOUNT' WHEN '7' THEN 'ORGANIZATION' WHEN '8' THEN 'APPLICATION\_OWNER' WHEN '9' THEN 'APPLICATIONOWNER\_METADATA' END AS campaign\_type FROM campaign cm, users u WHERE cm.campaign\_owner\_id = u.USERKEY AND cm.campaign\_type = 2 AND cm.start\_date > DATE\_SUB(CURDATE(), INTERVAL 30 DAY); |
| 10 | CCA\_CC\_User Manager Campaign Actions By Type | Admin | SELECT cm.CAMPAIGN\_NAME AS 'Campaign Name', CASE cm.STATUS WHEN 1 then 'New' WHEN 2 then 'Completed' WHEN 5 then 'In Progress' WHEN 7 then 'Expired' END AS 'Status of Campaign', sum(CASE WHEN cus.CERTIFIED=3 THEN 1 ELSE 0 END) AS 'Does Not Work For Me', sum(CASE WHEN cus.CERTIFIED=4 THEN 1 ELSE 0 END) AS 'Terminated', sum(CASE WHEN cus.AUDITTRAIL LIKE '%Consulted to%' THEN 1 ELSE 0 END) AS 'Consulted' FROM campaign cm INNER JOIN certification c ON cm.id = c.CAMPAIGNKEY INNER JOIN certification\_user\_status cus ON c.CERTKEY =cus.CERTKEY AND (cus.CERTIFIED in (3,4) or cus.AUDITTRAIL LIKE '%Consulted to%') WHERE cm.CAMPAIGN\_TYPE =2 and c.IS\_IGNORE\_STEP1DATA\_POPULATED = false and cm.STATUS in (1,2,5,7) AND cm.start\_date > DATE\_SUB(CURDATE(), INTERVAL 30 DAY) GROUP BY cm.CAMPAIGN\_NAME ORDER BY cm.CREATION\_DATE ; |
| 11 | Dasboard\_Managers\_Who\_Completed\_TransferCertififcation\_On\_Time | Admin | select cm.campaign\_name as 'Campaign Name', case cm.campaign\_type when '1' then 'ENTITLEMENTOWNER' when '2' then 'USERMANAGER' when '3' then 'SELF\_CERTIFICATION' when '5' then 'ROLEOWNER' when '6' then 'SERVICEACCOUNT' when '7' then 'ORGANIZATION' when '8' then 'APPLICATION\_OWNER' when '9' then 'APPLICATIONOWNER\_METADATA' end as 'Campaign Type', c.CERT\_NAME as 'Certification Name', u.username as 'Certifier', u.DEPARTMENTNAME as 'Certifier Department', case when c.status = 0 then 'New' when c.status = 1 then 'In Progress' when c.status = 2 then 'Completed' when c.status = 3 then 'Locked' when c.status = 4 then 'Expired' when c.status = 5 then 'Discontinued' when c.status = 6 then 'Preview' when c.status = 7 then 'Discontinued' when c.status = 8 then 'LockedandTaskCreated' when c.status = 9 then 'Cancelled' end as 'Certification Status', cm.start\_date as 'Campaign\_Start\_Date', cm.end\_date as 'Campaign\_End\_Date', c.STARTDATE as 'Certification\_Start\_Date', c.ENDDATE as 'Certification\_End\_Date' from certification c join campaign cm on c.CAMPAIGNKEY = cm.id join users u on c.certifier = u.USERKEY where c.UPDATEDATE > DATE\_SUB(CURDATE(), INTERVAL 30 DAY) and campaign\_type = 2 and c.status in (2,3,8) and c.UPDATEDATE < c.ENDDATE and c.CERT\_NAME like 'Transfer\_Certification\_%' order by c.CERTKEY desc; |
| 12 | ApplicationImport\_SuccessRate | Admin | SELECT DISTINCT  JOBNAME AS JOB\_NAME, IFNULL(TRIGGERNAME, JOBNAME) AS TRIGGER\_NAME, EXTERNALCONNECTION AS CONNECTION\_NAME, CASE SAVRESPONSE WHEN 'Success' THEN 'Success' ELSE 'Failed' END AS STATUS\_OF\_JOB, JOBSTARTDATE AS STARTDATE\_OF\_JOB, JOBENDDATE AS ENDDATE\_OF\_JOB  FROM ecmimportjob  WHERE JOBNAME IN ('ApplicationDataImportJob', 'AccountsImportFullJob', 'SchemaAccountJob')  AND JOBSTARTDATE >= (UTC\_DATE() - INTERVAL 30 DAY) ORDER BY JOBSTARTDATE DESC |
| 13 | Detected - Aborted Jobs | Admin | select jobid, jobname,jobstartdate,jobenddate,'Aborted' as JobStatus from ecmimportjob ecm, endpoints ep,externalconnectiontype ext,externalconnection exc,securitysystems ss where jobenddate is null and ext.externalconnectiontypekey = exc.externalconnectiontype and exc.externalconnectionkey = ss.externalconnection and ss.systemkey = ep.securitysystemkey union select jobid, jobname,jobstartdate,jobenddate,'Aborted' as JobStatus from ecmimportjob where EXTERNALCONNECTION is null and jobenddate is null; |
| 14 | Detected - Ghost Job Triggers | Admin | select qt.trigger\_name,qt.job\_name, FROM\_UNIXTIME((qt.prev\_fire\_time / 1000),'%Y-%m-%d %h:%m:%s') AS 'Prev\_Fired\_Time', FROM\_UNIXTIME((qt.NEXT\_FIRE\_TIME / 1000),'%Y-%m-%d %h:%m:%s') AS 'Next Trigger Date',qt.trigger\_state,qt.trigger\_type, FROM\_UNIXTIME((qt.start\_time/ 1000),'%Y-%m-%d %h:%m:%s') AS 'Start\_time' from qrtz\_triggers qt where TRIGGER\_NAME like 'MT\_%'; |
| 15 | Detected - Jobs in Hung State | Admin | SELECT ENTRY\_ID, TRIGGER\_NAME, TRIGGER\_GROUP, IS\_VOLATILE, INSTANCE\_NAME, FIRED\_TIME, PRIORITY, STATE, JOB\_NAME, JOB\_GROUP, IS\_STATEFUL, REQUESTS\_RECOVERY FROM qrtz\_fired\_triggers WHERE Fired\_Time > DATE\_SUB(NOW(), INTERVAL 12 HOUR) |
| 16 | Detected - Long Running jobs | Admin | SELECT FROM\_UNIXTIME(fired\_time / 1000) AS start\_time, NOW() AS CurrentTime, TIMESTAMPDIFF(HOUR, FROM\_UNIXTIME(fired\_time / 1000), NOW()) AS RunningTime\_Hrs, trigger\_name, job\_name FROM qrtz\_fired\_triggers WHERE DATE\_ADD(FROM\_UNIXTIME(fired\_time / 1000), INTERVAL 4 HOUR) < NOW() |
| 17 | Detected - Newly Onboarded Users | Admin | SELECT USERNAME, FIRSTNAME, LASTNAME, EMAIL, OWNER as 'MANAGER', CREATEDATE, CASE STATUSKEY WHEN '0' THEN 'INACTIVE' WHEN '1' THEN 'ACTIVE' END AS USERSTATUS, U.DEPARTMENTNAME as 'DEPARTMENT',U.EMPLOYEETYPE as 'USER TYPE' FROM USERS U WHERE CREATEDATE > DATE\_SUB(CURDATE(), INTERVAL 30 DAY) AND U.STATUSKEY = 1; |

ss