BMC Helix

Walk-through: Exploring Smart Reporting DB Repository

Posted by Asif Bhat in BMC Helix on Mar 11, 2019 8:23:00 PM

In this month's blog, I want to share insights on Smart Reporting Database Repository. Occasionally while working on the Smart Reporting DB, you may be stymied about the data here and its relation in context.

Let's discuss in greater detail about the Smart Reporting DB Repository, the data that gets stored and how it is linked. To make your job even easier I have listed some of the most useful SQL queries you will require while working on the DB. The Smart Reporting DB is a place where all the meta data is stored and includes:

- Report Metadata
- User accounts
- Dashboards
- Configuration options
- Events

Let's look at each of these comprehensively:

Stores information related to Users, Orgs and Groups.

- IpClass, Person, IpContact - Stores User information
- Organisation - Organisation details reside in this table.
- IpRltShp - user to org, org to primary org relationship.
- AccessGroup - Groups are stored in this db table.
- AccessGroupFlat - groupId to memberID relation.
- AccessGroupMembers - Members in group.

Audit data

- Event - This table stores all Smart Reporting usage data, such as; User logins, Running reports, Imports/exports. This data is all used for auditing.
- EventArchive - This table stores all the archived event data, so the data here is simply data that was moved from the 'Event' table after a specified period.

- ReportInstance - This table stores a record for each report, each time it is run or edited. If you run the same report multiple times, you will get multiple rows for that report in this table. This is used for things like; Getting ‘average report run time' and also used to 'Remember Filter Values' options, editing a report (the draft copy is a separate record so you can roll back to the original activated version), Results used for KPI reporting.
- ReportBroadcastResult - Information about broadcast run like recipient information and whether broadcast has been delivered or not.

Stores information about Roles and Functions.

- OrgRole - All available roles in the system
- AccessClassList - Roles to functions + Access Level (Create, Read, Update and Delete – CRUD)
- StaffMemberRole - User to Role information.

Information about DataSource and Views

- ReportViewSource - Stores DataSource related details.
- ReportViewSourceParameter - Data Source parameters e.g. Custom parameters
- ReportView - Information about views.
- ReportViewRltshp - Info about table joins.
- ReportFieldTemplate - Objects in the view.

Reports and Charts

- ReportHeader - Main table which holds report metadata.
- ReportField - linked to fieldtemplateid. Stores fields present in a report.
- ReportFilter - This is linked to fieldtemplate. Stores view level cached filters, grouped data filters, set analysis, variance etc.
- ReportFormat - Stores report, field, filter, chart formats.
- Chart, ChartField, ChartAnalyticField - core tables which hold chart information.
- ReportAlert - conditional format information.
- ReportBroadcast - Stores broadcast information.
- ReportBroadcastRecipient - Stores broadcast recipient list.
- ReportFieldConverter - analytic functions, view level converters, formatters.
- ReportFieldParameter - parameters for these objects.
- CachedFilter - links to ReportFilter, stores report definition.
- CachedFilterValue - filter values, links to parent filters.
- ReportGroup - Information about dashboard
- ReportGroupEntity - items on the dashboard.
- ReportGroupFormat - filters, analytic links.

Blobs & Clobs

- TextEntity, TextData - Stores discussion on Report, dashboard, storyboard etc.
- DocumentData - This table stores actual data of reports, like cached report result sets (this includes cached report, and snapshots created at the report level) , Freehand SQL (used for virtual tables, custom query filters, freehand SQL reports) , Cached filter results from older releases (we [illegible] results in different tables), CSV reports.

- ReportImageItem, ReportImageData - thumbnail images, backgrounds reportimagemapzone - raster map zones - colour to ref code.

System

- TaskSchedule - System tasks and background tasks.
- Configuration - System-wide settings per org.

Access Filter

- ReportSourceFilter – Stores access filter values.

SQL queries Walk through

1) List all active reports:

```
SELECT * FROM ReportHeader WHERE ReportStatusCode='OPEN'
```

ReportStatusCode can be DELETED, PENDING, ARCHIVED, OPEN and DRAFT'.

- DELETED means you clicked the delete button from within the UI.
- PENDING means it has been created, but there is content approval on the content folder, and the report requires approval.
- ARCHIVED means that you have edited this report (each time you edit it creates a new record), and this specific row is no longer the active version of the report
- OPEN means the report is active
- DRAFT means that it has been edited but not published

2) List all active private reports:

```
SELECT * FROM ReportHeader WHERE ExternalAccessCode='PERSONAL' AND ReportStatusCode='OPEN'
```

3) List all active public reports:

```
SELECT * FROM ReportHeader WHERE ExternalAccessCode=' CORPORATE' AND
ReportStatusCode='OPEN'
```

4) List of all active Users:

```
SELECT p.firstname,
       p.lastname,
       c.emailaddress,
       i.emailleft,
       s.rolecode
FROM   person p
       INNER JOIN ipcontact c
ON p.ipperson = c.ipid
       INNER JOIN ipclass i
ON p.ipperson = i.ipid
       INNER JOIN staffmemberrole s
ON p.ipperson = s.ipemployee
WHERE  i.enddate > Getdate()
       AND s.enddate > Getdate();
```

Quick Links

5) List of Enabled functions for a role:

```
SELECT *
FROM   accessclasslist
WHERE  rolecode LIKE '%YFADMIN%'
       AND accesslevelcode IS NOT NULL
```

Results | Messages

	IpOrg	RoleCode	AccessLevelCode	ACLSeqNbr	FunctionName	FunctionTypeCode	SortOrder
11	1	YFADMIN	CRUD	8	BROADCAST	NULL	0
12	1	YFADMIN	CRUD	68	BROADCASTSUBSCRIBE	NULL	0
13	1	YFADMIN	CRUD	9	CLIENTORGS	NULL	0
14	1	YFADMIN	CRUD	39	CONTENTACCESS	CP	0
15	1	YFADMIN	CRUD	17	CREATESSTABLE	NULL	0
16	1	YFADMIN	CRUD	19	CUSTOMFILTER	NULL	0
17	1	YFADMIN	CRUD	50	DASHPUBLIC	NULL	0
18	1	YFADMIN	CRUD	54	DASHREFRESH	NULL	0
19	1	YFADMIN	CRUD	21	DATASOURCE	NULL	0
20	1	YFADMIN	CRUD	20	DATASOURCEADMIN	NULL	0
21	1	YFADMIN	CRUD	22	DATAVIEW	NULL	0
22	1	YFADMIN	CRUD	10	DISCUSSIONADMIN	NULL	0
23	1	YFADMIN	CRUD	23	DISCUSSIONGROUPS	NULL	0
24	1	YFADMIN	CRUD	24	DISTRIBUTEREPORT	NULL	0

Above query will list the enabled functions for the Admin Role.

6) User details with Role and Client Org Name

```
SELECT p.fullname as "User",
       ic.emailright as "Cleint Org",
       sm.rolecode as Role
FROM   ipclass ic
       INNER JOIN person p
ON ( ic.ipid = p.ipperson )
       INNER JOIN staffmemberrole sm
ON ( ic.ipid = sm.ipemployee )
WHERE  ic.emailright = 'calbro'
       AND ic.enddate = '99991231 00:00:00.000'
       AND sm.enddate = '99991231 00:00:00.000'
```

Quick Links

Results | Messages

	IpOrg	RoleCode	AccessLevelCode	ACLSeqNbr	FunctionName	FunctionTypeCode	SortOrder
11	1	YFADMIN	CRUD	8	BROADCAST	NULL	0
12	1	YFADMIN	CRUD	68	BROADCASTSUBSCRIBE	NULL	0
13	1	YFADMIN	CRUD	9	CLIENTORGS	NULL	0
14	1	YFADMIN	CRUD	39	CONTENTACCESS	CP	0
15	1	YFADMIN	CRUD	17	CREATESSTABLE	NULL	0
16	1	YFADMIN	CRUD	19	CUSTOMFILTER	NULL	0
17	1	YFADMIN	CRUD	50	DASHPUBLIC	NULL	0
18	1	YFADMIN	CRUD	54	DASHREFRESH	NULL	0
19	1	YFADMIN	CRUD	21	DATASOURCE	NULL	0
20	1	YFADMIN	CRUD	20	DATASOURCEADMIN	NULL	0
21	1	YFADMIN	CRUD	22	DATAVIEW	NULL	0
22	1	YFADMIN	CRUD	10	DISCUSSIONADMIN	NULL	0
23	1	YFADMIN	CRUD	23	DISCUSSIONGROUPS	NULL	0
24	1	YFADMIN	CRUD	24	DISTRIBUTEREPORT	NULL	0

7) List of Reports with View Name

```
SELECT rh.reportname,
       rv.viewdescription AS "View Name"
FROM   reportheader rh
       INNER JOIN reportview rv
ON ( rh.viewid = rv.viewid )
```

8) View Object Usage:

```
SELECT rf.reportid,
       rh.reportname,
       rh.reportstatuscode
FROM   reportfield rf
       INNER JOIN reportfieldtemplate rft
ON ( rf.fieldtemplateid = rft.fieldtemplateid )
       INNER JOIN reportheader rh
ON ( rh.reportid = rf.reportid )
WHERE  rft.columnname LIKE 'Incident Number'
       AND rft.viewid = 108662
```

The above query will list all reports where **"Incident Number"** field is used.

9) List of active Views:

```
SELECT *
FROM   reportview
WHERE  viewname = 'New View'
       AND viewstatuscode = 'OPEN'
```

10) Fields deleted from the View:

```
SELECT *
FROM   reportfieldtemplate
WHERE  viewid IN (SELECT viewid
```

Quick Links

```
FROM reportview
WHERE viewstatuscode = 'OPEN'
                    AND viewname = 'New View')
      AND statuscode = 'DISCONNECTED';
```

11) Fields present in specific Report:

```
SELECT distinct rh.reportname,
      rft.shortdescription
FROM   reportfield rf
      INNER JOIN reportheader rh
ON rf.reportid = rh.reportid
      INNER JOIN reportfieldtemplate rft
ON rf.fieldtemplateid = rft.fieldtemplateid
WHERE  rh.reportname = 'Incident Details'
GROUP  BY rh.reportname,
rft.shortdescription
```

The above query will list all objects /columns present in the "Incident Details" Report

Results | Messages

	reportname	shortdescription
1	Incident Details	Assigned Support Company
2	Incident Details	Company
3	Incident Details	Department
4	Incident Details	Description
5	Incident Details	Impact
6	Incident Details	Incident Number Hyperlink
7	Incident Details	Organization
8	Incident Details	Owner Group
9	Incident Details	Priority
10	Incident Details	Status
11	Incident Details	Submit Date
12	Incident Details	Submit Date (ARJDBC)
13	Incident Details	Submitter
14	Incident Details	Urgency

12) Reports that need update on hyperlink object:

```
SELECT *
FROM   reportheader
WHERE  reportid IN (SELECT reportid
                    FROM   reportfieldparameter
                    WHERE  viewid = '84506'
                        AND entitytypecode = 'REPORTFIELD'
                        AND classname = 'com.hof.mi.custom.format.LinkFormatter'
                        AND parameterkey = 'URL')
```

13) List of private reports that belong to deleted Users:

Quick Links

```
SELECT *
FROM   reportheader R
       INNER JOIN ipclass I
ON ipid = ipowner
WHERE  I.enddate != '9999-12-31'
       AND R.reportstatuscode = 'OPEN'
       AND R.externalaccesscode = 'PERSONAL'
```

14) List of all active Dashboards:

```
SELECT *
FROM   reportgroup
WHERE  enddate = '9999-12-31'
       AND grouptypecode = 'ANALYTIC'
       AND statuscode = 'OPEN'
```

15) List of all PRIVATE Dashboards:

```
SELECT *
FROM   reportgroup
WHERE  grouptypecode = 'ANALYTIC'
       AND accesscode = 'ACCESSLEVEL'
```

16) List of all active Client Orgs:

```
SELECT     org.orgname,
org.iporg,
rvc.databaseurl,
rvc.username
FROM       organisation org
INNER JOIN iprltshp rlt
ON         (
                   rlt.ipchild = org.iporg
AND        rlt.endreasoncode IS NULL
AND        rlt.rltshptypecode = 'CUSTOMER')
INNER JOIN reportviewsource rvc
ON         (
                   rvc.iporg = org.iporg
AND        databasetypecode = 'BMCACTIONREQUEST')
```

Results | Messages

	orgname	iporg	username
1	calbro_Calbro Services	12402	Demoonbmc
2	calbro_default	12412	Demoonbmc
3	calbro_Invention, Inc.	12420	Demoonbmc
4	calbro_SAP	12415	Demoonbmc

17) Report and System tasks scheduled in minutes:

```
SELECT *
FROM   taskschedule
```

Quick Links

```
WHERE  frequencytypecode = 'MINUTES'
       AND scheduleison = 1
```

Results | Messages

	IpOrg	ScheduleSubjectCode	ScheduleUnitCode	FrequencyTypeCode	FrequencyUnit
1	1	MIREPORT	REPORTID	MINUTES	30
2	1	SYSTEMTASK	PersistentWebserviceSessionMonitorTask	MINUTES	5

18) Entries created by Report Refresh in DocumentItem, DocumentRevision and DocumentData table:

```
SELECT *
FROM   documentitem
WHERE  documenttypecode = 'REPORTXML'


SELECT *
FROM   documentrevision
WHERE  documentid IN (SELECT documentid
                      FROM   documentitem
                      WHERE  documenttypecode = 'REPORTXML')

SELECT *
FROM   documentdata
WHERE  revisionid IN (SELECT revisionid
                      FROM   documentrevision
                      WHERE  documentid IN (SELECT documentid
                                            FROM documentitem
                                            WHERE
                             documenttypecode = 'REPORTXML'))
```

19) List of Broadcasts with Report Info:

```
SELECT RB.broadcastid,
       RB.reportid,
         RH.reportname,
         RB.SubjectLine,
       RB.broadcasttypecode,
       RB.broadcastcode,
       RB.startdate,
       RB.enddate,
       TS.frequencytypecode,
       TS.frequencyunit,
       TS.lastruntime,
       TS.lastrunstatus
FROM   reportbroadcast RB,
reportheader RH,
taskschedule TS
WHERE  RB.reportid = RH.reportid
       AND TS.scheduleunitid = RB.broadcastid
```

Quick Links

Results Messages

	broadcastid	reportid	reportname	SubjectLine	broadcasttypecode	broadcastcode	startdate	enddate	frequencytypecode	frequencyunit
1	145939	145917	CO-DISPLAY_Parent	Test Broadcast	EMAIL	XLS	2018-09-12	9999-12-31	MONTHLY	1
2	148140	109971	Problem - Resolution Rate Vs Workaround Rate by...	TEST	EMAIL	XLS	2018-09-28	9999-12-31	WEEKLY	6

20) Entry page for existing users:

```
SELECT person.firstname as "First Name",
person.lastname as "Last Name",
ipclass.emailleft as "Login ID",
personconfiguration.configcode,
personconfiguration.configdata
FROM   personconfiguration,
ipclass,
person
WHERE  configcode = 'ENTRYPAGE'
       AND person.ipperson = ipclass.ipid
       AND personconfiguration.ipperson = ipclass.ipid
```

Results Messages

	First Name	Last Name	Login ID	configcode	configdata
1	Asif	Bhat	abhat	ENTRYPAGE	DASHBOARD
2	siadmin	siadmin	siadmin	ENTRYPAGE	DASHBOARD
3	tmart	user	tmart_user	ENTRYPAGE	DASHBOARD
4	App	Admin	appadmin	ENTRYPAGE	DASHBOARD
5	Chat	User	chatuser	ENTRYPAGE	BROWSE
6	Test1	user1	testuser	ENTRYPAGE	DASHBOARD

21) Number of reports that are currently cached and the category they belong to:

```
SELECT a.categorycode,
       a.subcategorycode,
       Count (DISTINCT a.reportid) as "Report Count"
FROM   reportheader a,
reportinstance b,
documentitem c
WHERE  a.reportid = b.reportid
       AND a.reportid = c.subjectid
GROUP  BY a.categorycode,
a.subcategorycode
```

22) List of cached reports in the document data table and count of reports copies per category:

```
SELECT a.reportid,
       a.reportname,
       Count (b.reportinstanceid) AS ReportInstance#,
       a.categorycode,
       a.subcategorycode,
       C.subjectid
```

Quick Links

```
FROM   reportheader a,
reportinstance b,
documentitem c
WHERE  a.reportid = b.reportid
      AND a.reportid = c.subjectid
GROUP  BY a.categorycode,
a.subcategorycode,
a.reportname,
a.reportid,
c. subjectid
```

23) Count of different types of Dashboard Events:

```
SELECT DISTINCT
"Events"."EventDate",
"Events"."EventCode",
  COUNT(DISTINCT("Events"."EventId")) as "Event Count"
FROM (
  select * from Event
UNION ALL
Select * from EventArchive

) AS "Events"
WHERE (
"Events"."EventCode" IN (N'DASHRUN', N'EXPORTDASHBOARD', N'DASHEDIT', N'DASHCREATE',
N'IMPORTDASHBOARD')
)
GROUP BY
"Events"."EventDate",
"Events"."EventCode"
```

Results | Messages

	EventDate	EventCode	Event Count
1	2018-05-21	DASHCREATE	1
2	2018-05-21	DASHEDIT	2
3	2018-09-18	DASHEDIT	1
4	2018-09-28	DASHEDIT	3
5	2018-04-02	DASHRUN	8
6	2018-04-23	DASHRUN	4
7	2018-05-07	DASHRUN	8
8	2018-06-21	DASHRUN	25
9	2018-07-04	DASHRUN	136
10	2018-07-20	DASHRUN	23
11	2018-07-25	DASHRUN	12

24) Count of Users belonging to a Specific Group:

```
SELECT DISTINCT
"User Group Details"."ShortDescription" as "Group Name",
  COUNT(DISTINCT("User"."IpPerson")) as "Users"
```

Quick Links

```
FROM "Person" AS "User"
LEFT OUTER JOIN "AccessGroupFlat" AS "User Group"
ON (
"User"."IpPerson" = "User Group"."IpPerson"
)
LEFT OUTER JOIN "AccessGroup" AS "User Group Details"
ON (
"User Group"."AccessGroupId" = "User Group Details"."AccessGroupId"
)
WHERE (
"User Group Details"."ShortDescription" IS NOT NULL
)
GROUP BY  "User Group Details"."ShortDescription"
```

Results | Messages

	Group Name	Users
1	CMDB User	18
2	General Users	2
3	Support Staff	5

25) Count of Users belonging to a specific Role:

```
SELECT DISTINCT
  "User Role Details"."ShortDescription" as "Role Name",
  COUNT(DISTINCT("User"."IpPerson")) as "Users"
FROM "Person" AS "User"
INNER JOIN "StaffMemberRole" AS "User Role"
ON (
  "User"."IpPerson" = "User Role"."IpEmployee"
)
INNER JOIN "OrgReferenceCodeDesc" AS "User Role Details"
ON (
  "User Role"."RoleCode" = "User Role Details"."RefCode"
)
GROUP BY
  "User Role Details"."ShortDescription"
```

Results Messages

	Role Name	Users
1	ASIF TEST	1
2	Consumer & Collaborator	2
3	Dashboard Consumer	37
4	Public Content Writer & Collaborator	1
5	Public Content Writer & Collaborator - Advanced	1
6	System Administrator	13

26) Avg Report Run Time:

```
SELECT DISTINCT
"Report"."ReportName",
  AVG("Report Instance"."ReportDuration") as "Average Runtime",
  AVG("Report Instance"."RowsProcessed") as "Average Rows",
"Report"."HitCount"
FROM "ReportHeader" AS "Report"
LEFT OUTER JOIN "ReportInstance" AS "Report Instance"
ON (
"Report"."ReportId" = "Report Instance"."ReportId"
)
WHERE (
"Report"."RoleCode" != N'SUBQUERY'
)
AND (
"Report"."ReportStatusCode" = N'OPEN'
  AND "Report"."RoleCode" != N'SUBQUERY'
)
GROUP BY
"Report"."ReportName",
"Report"."HitCount",
"Report"."PublishUUID"
HAVING AVG("Report Instance"."ReportDuration") IS NOT NULL
```

> **Note - ReportDuration** Column present in **ReportInstance** table represents the time report query takes to return from the database. This is NOT the total time the report takes to fully render in the browser. Rendering in the browser will take longer for reports with large amounts of post processing elements like cross-tabs, analytical functions, etc. This value is measured in whole seconds. The datatype is an INT, and therefore can't show less than 1 second with precision. This is why a large number of values in this column may be 0.

Quick Links

Results Messages

	ReportName	Average Runtime	Average Rows	HitCount
26	Incident Analysis by Support Hierarchy	0	4	23
27	Incident By Service Type	0	2	23
28	Incident by Status	0	0	2
29	Incident Crosstab	0	3	23

27) Available Fonts in Smart Reporting:

```
SELECT *
FROM   configuration
WHERE configcode = 'ADDITIONALFONTS'
```

Results Messages

	IpOrg	ConfigTypeCode	ConfigCode	ConfigData
1	1	SYSTEM	ADDITIONALFONTS	SourceSansPro|sourceSansPro|Roboto|Roboto|Arial|Arial|Helvetica|Helvetica|Geneva|Genevalsans-serf|sans-serf
2	12001	SYSTEM	ADDITIONALFONTS	SourceSansPro|sourceSansPro|Roboto|Roboto|Arial|Arial|Helvetica|Helvetica|Geneva|Genevalsans-serf|sans-serf
3	12011	SYSTEM	ADDITIONALFONTS	SourceSansPro|sourceSansPro|Roboto|Roboto|Arial|Arial|Helvetica|Helvetica|Geneva|Genevalsans-serf|sans-serf
4	12021	SYSTEM	ADDITIONALFONTS	SourceSansPro|sourceSansPro|Roboto|Roboto|Arial|Arial|Helvetica|Helvetica|Geneva|Genevalsans-serf|sans-serf
5	12151	SYSTEM	ADDITIONALFONTS	SourceSansPro|sourceSansPro|Roboto|Roboto|Arial|Arial|Helvetica|Helvetica|Geneva|Genevalsans-serf|sans-serf
6	12402	SYSTEM	ADDITIONALFONTS	SourceSansPro|sourceSansPro|Roboto|Roboto|Arial|Arial|Helvetica|Helvetica|Geneva|Genevalsans-serf|sans-serf
7	12412	SYSTEM	ADDITIONALFONTS	SourceSansPro|sourceSansPro|Roboto|Roboto|Arial|Arial|Helvetica|Helvetica|Geneva|Genevalsans-serf|sans-serf
8	12415	SYSTEM	ADDITIONALFONTS	SourceSansPro|sourceSansPro|Roboto|Roboto|Arial|Arial|Helvetica|Helvetica|Geneva|Genevalsans-serf|sans-serf
9	12420	SYSTEM	ADDITIONALFONTS	SourceSansPro|sourceSansPro|Roboto|Roboto|Arial|Arial|Helvetica|Helvetica|Geneva|Genevalsans-serf|sans-serf

28) List of Views in a specific Org:

```
SELECT *
FROM   reportview v,
reportviewsource s
WHERE  v.sourceid = s.sourceid
       AND v.viewstatuscode NOT IN ( 'DELETED', 'REPLACED', 'ROLLEDBACK' )
       AND s.accesscode != 'DELETED'
       AND ( s.iporg = (SELECT iporg
                        FROM   organisation
                        WHERE  orgname = 'calbro_Calbro Services') )
       AND viewname = 'New View'
```

The above query will list all active Views present in **'calbro_Calbro Services'** Client Org.

29) Private Reports accessible by a Specific User:

```
SELECT entityid,
entitytypecode,
accesslevelcode,
accesssubjectcode,
      accessgroupid
FROM   accesslevel
WHERE  entitytypecode = 'PRIVATEREPORT'
       AND ( ( accesssubjectcode = 'PERSON'
AND accessgroupid = 12251 )
OR ( accesssubjectcode = 'GROUP'
```

Quick Links

```
AND accessgroupid IN (SELECT accessgroupid
                      FROM accessgroupflat
                      WHERE  ipperson = (SELECT ipid
FROM ipclass
                                         WHERE
emailleft = 'sradmin'))
) )
```

The above query will list all accessible private reports for **"sradmin"** User.

30) Query to verify the EXTERNALURLPARAMS (Share Link Parameters):

```
select * from configuration        where configcode = 'EXTERNALURLPARAMS'
```

31) List of all PRIVATE Content folders:

```
SELECT *       FROM   contentmanagement       WHERE  loginaccesscode = 'ACCESSLEVEL'
```

32) List of all Content folders where Expert Approval is set:

```
SELECT *
FROM   contentmanagement
WHERE  approvalflag = 1
```

33) Data Source Usage:

```
SELECT DISTINCT
"Data Source"."SourceName",
  COUNT(DISTINCT("Report"."PublishUUID")) as "Active Reports",
  COUNT(DISTINCT("Events"."EventId")) as "Report Hits"
FROM (
  select * from Event
UNION ALL
Select * from EventArchive

) AS "Events"
LEFT OUTER JOIN "ReportHeader" AS "Report"
ON (
"Events"."ContentId" = "Report"."ReportId"
)
AND (
"Report"."RoleCode" != N'SUBQUERY'
)
LEFT OUTER JOIN "ReportView" AS "View"
ON (
"Report"."ViewId" = "View"."ViewId"
)
LEFT OUTER JOIN "ReportViewSource" AS "Data Source"
ON (
"View"."SourceId" = "Data Source"."SourceId"
)
WHERE (
"Events"."EventCode" IN (N'DASHRUN', N'RPTRUN')
```

```
    AND "Report"."ReportStatusCode" = N'OPEN'
    AND "Data Source".SourceName in ('AR System')
  )
  GROUP BY
  "Data Source"."SourceName"
```

Results | Messages

	SourceName	Active Reports	Report Hits
1	AR System	44	10321

34) View Usage:

```
  SELECT
    "View"."ViewDescription",
    COUNT(DISTINCT("Report"."PublishUUID")) as "Active Reports",
    COUNT(DISTINCT("Events"."EventId")) as "Report Hits"
  FROM (
    select * from Event
  UNION ALL
  Select * from EventArchive

  ) AS "Events"
  LEFT OUTER JOIN "ReportHeader" AS "Report"
  ON (
    "Events"."ContentId" = "Report"."ReportId"
  )
  AND (
    "Report"."RoleCode" != N'SUBQUERY'
  )
  LEFT OUTER JOIN "ReportView" AS "View"
  ON (
    "Report"."ViewId" = "View"."ViewId"
  )
  WHERE (
    "View"."ViewStatusCode" NOT IN (N'DELETED', N'REPLACED')
    AND "Events"."EventCode" IN (N'DASHRUN', N'RPTRUN')
    AND "Report"."ReportStatusCode" = N'OPEN'
  )
  GROUP BY
    "View"."ViewDescription"
```

Results | Messages

	ViewDescription	Active Reports	Report Hits
1	App - Incident Management	1	6
2	Change Management	4	6
3	Configuration Management	3	243
4	DLP View	1	2
5	Incident Management	2	2
6	Incident Management-OOTB	28	9934
7	KPI	2	94
8	Permission	1	1
9	Report Field - TEST	1	5

Disclaimer - *It is not recommended to perform any kind of DML and DDL operations on Repository DB, It may break referential integrity of content Metadata.*

I hope you found this post helpful in understanding Smart Reporting DB Repository. Also take an opportunity to visit our Smart Reporting User Group The specified item was not found.

You can find additional blogs like this at The specified item was not found.

1096 Views Categories: Administration, Tips & Tricks, Troubleshooting
Tags : repository, blog, smartreporting, bmchelix

Average User Rating
(9 ratings)

Your Rating:

1 HELPFUL **MOST LIKED**

8 Comments

Mohammad Rehman Mar 11, 2019 11:31 PM

Thanks Asif Bhat for must needed knowledge sharing.

3 of 3 people found this helpful

Actions ☆Helpful | Like (1)

Swapnil Fusate Mar 12, 2019 1:03 AM

Great Stuff. Thanks a lot Asif Bhat
Special thanks for the SQL queries.

Quick Links

Actions ☆Helpful | Liked (1)

Sharad Joshi Mar 12, 2019 1:12 AM

Good Stuff Asif..

Actions ☆Helpful | Liked (1)

Minil Nair Mar 12, 2019 10:55 PM

Great stuff Asif Bhat (y) . This is really helpful!!

Thank you for sharing .

Actions ☆Helpful | Liked (2)

Stefan Hall Mar 20, 2019 8:53 PM

Asif Bhat I'd like it better if we put that knowledge under The specified item was not found.. Helix is one of these marketing terms and meanwhile everything is Helix. I don't think I'll find the article here again.

Besides, the article's great

Actions ☆Helpful | Liked (2)

Jim Wheeler May 13, 2019 6:10 PM

thank you for sharing!

Actions ☆Helpful | Liked (2)

Chakradharan Tirumala Jul 16, 2019 7:59 PM

This is awesome! Thank you for sharing!

Actions ☆Helpful | Liked (3)

David K Hill Aug 30, 2019 3:22 AM

Demystification attributed!!!
Thanks

Actions ☆H

Quick Links