

# Raiser's Edge SQL Server Codes

William Smith

# Raiser's Edge SQL Server Codes

This book on Raiser's Edge SQL Server queries explores common SQL Server codes that I have used in the industry over the last 2 years. The database is used for the purpose of customer relationship management (CRM) and fundraising within the marketing industry. Copyright © by William Smith, 2015-04-27.

## Contents

### [SQL SERVER OLAP QUERIES – RAISER'S EDGE](#)

[All Categories LIST](#)

[Categories & Descs & Shrt Descs LIST](#)

[All Codetables LIST](#)

[CATEGORIES COUNTS](#)

[Code Tables relationships code](#)

[TableEntries Search](#)

[PARTICIPANTATTRIBUTES SEARCH](#)

[JOBATTRIBUTES SEARCH](#)

[PROPOSALATTRIBUTES SEARCH](#)

[PACKAGEATTRIBUTES SEARCH](#)

[ACTIONATTRIBUTES SEARCH](#)

[CONSTITADDRESSATTRIBUTES SEARCH](#)

[GiftAttributes SEARCH](#)

[ConstituentAttributes SEARCH](#)

[APPEALATTRIBUTES SEARCH](#)

[CONSTITUENT\\_BANKATTRIBUTES SEARCH](#)

[EVENTATTRIBUTES SEARCH](#)

[FUNDATTRIBUTES SEARCH](#)

[RELATIONSHIPATTRIBUTES SEARCH](#)

[JOBATTRIBUTES SEARCH](#)

[PHONE TYPES TABLE ENTRIES E.G. BUSINESS PHONE](#)

[COUNTRIES TABLE ENTRIES E.G. UNITED STATES](#)

[Tableentries Sample Data \(Group Concat via AttributeTypes\)](#)

[NON ALPHANUM CHARS SEARCH](#)

[SPACED OR NULL CHARS SEARCH](#)

[ACTION STATUS AND TYPE TO TABLEENTRIES JOIN](#)

[COURSES PARTICIPANT ATTENDED COUNT](#)

[COURSES PARTICIPANT IS INSTRUCTOR COUNT](#)

[COURSES PARTICIPANT PARTICIPATION COUNT](#)

[COURSES PARTICIPANT STATUS COUNT](#)

[COURSES PARTICIPANT RESPONSE COUNT](#)

[COURSES EVENT CONTACT NAMES](#)

[COURSES PARTICIPANT EVENT CONTACT NAMES](#)

[/\\* Industries AKA Activity drop down list values \\*/](#)

[/\\* Profession drop down list values \\*/](#)

[/\\* Industries AKA Activity entered values that have been inputted or imported into RE\\*/](#)

[/\\* Profession entered values that have been inputted or imported into RE \\*/](#)

[SQL CODE FOR DISPLAYING ALL BASE TABLES IN SSMS.](#)

[SQL RECORDS AND INDUSTRY CROSS REFERENCE WITH TABLEENTRIES AND USERS](#)

[/\\* SQL PHONES, CONSTITUENT AND ADDRESS EXTRACT \\*/](#)

[SQL NON-ALPHABETIC NAMES](#)

[Phone Types Counts](#)

[DUPLICATE CONSTITUENTATTRIBUTES](#)

[DUPLICATES GIFTATTRIBUTES](#)

[DUPLICATE CONSTITADDRESSATTRIBUTES](#)

[SUM DUPLICATE CONSTITADDRESSATTRIBUTES](#)

[SUM DUPLICATE CONSTITUENTATTRIBUTES](#)

[SQL PUNCTUATION COUNTS IN LAST NAME](#)

[SQL LAST NAMES WITH NUMBERS](#)

[SQL FIRST NAMES WITH NUMBERS](#)

[SQL PRIMARY SALUTATION WITH NUMBERS](#)

[SQL INDUSTRY LIST](#)

[SQL ACTION NOTE TYPES](#)

[SQL CODE ACTION TYPES](#)

[SQL SERVER OLTP DATA WAREHOUSE QUERIES – RAISER'S EDGE](#)

[RAISER'S EDGE Data Warehouse SQL Stored Procedures Definitions](#)

# **SQL SERVER OLAP QUERIES – RAISER’S EDGE**

## **All Categories LIST**

Select \*

from AttributeTypes;

## Categories & Descs & Shrt Descs LIST

Select A.ATTRIBUTEID, A.[DESCRIPTION] as [Category], A.TYPEOFDATA,  
T.LONGDESCRIPTION as [Description], T.SHORTDESCRIPTION,  
A.CODETABLESID, A.ACTIVE as AttrActive, T.ACTIVE as TblEntryActive, C.NAME  
as [Codetable Names], T.TABLEENTRIESID

from AttributeTypes as A, TABLEENTRIES as T, CODETABLES as C

where A.CODETABLESID=T.CODETABLESID

and C.CODETABLESID=T.CODETABLESID

order by A.[DESCRIPTION], A.ATTRIBUTEID, T.[LONGDESCRIPTION],T.  
[SHORTDESCRIPTION];

## **All Codetables LIST**

Select \*

From Codetables;

## CATEGORIES COUNTS

/\*\*\*/ATTRIBUTE TYPE

COUNTS\*\*\*/

— CONSTITUENTATTRIBUTES  
SELECT A.ATTRIBUTETYPESID,  
A.[DESCRIPTION] AS  
“CATEGORY”,  
COUNT(AC.PARENTID) AS  
ATTRIBUTESCOUNT, A.ACTIVE,  
‘CONSTITUENTATTRIBUTES’  
AS ATTRIBUTETYPE  
FROM AttributeTypes as A WITH  
(NOLOCK)  
LEFT JOIN  
CONSTITUENTATTRIBUTES as  
AC WITH (NOLOCK)  
ON  
AC.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID  
LEFT JOIN RECORDS AS R  
WITH (NOLOCK)  
ON R.ID=AC.PARENTID  
where A.RECORDTYPE = 1  
GROUP  
BY A.ATTRIBUTETYPESID,  
A.[DESCRIPTION], A.ACTIVE  
UNION ALL  
— GIFTATTRIBUTES  
SELECT A.ATTRIBUTETYPESID,  
A.[DESCRIPTION] AS  
“CATEGORY”,  
COUNT(AC.PARENTID) AS



```
ATTRIBUTESCOUNT, A.ACTIVE,  
'GIFTATTRIBUTES' AS  
ATTRIBUTETYPE  
FROM AttributeTypes as A WITH  
(NOLOCK)  
LEFT JOIN GIFTATTRIBUTES as  
AC WITH (NOLOCK)  
ON  
AC.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID  
LEFT JOIN GIFT G WITH  
(NOLOCK)  
ON G.ID = AC.PARENTID  
WHERE A.RECORDTYPE = 2  
AND G.DTE > '01/01/1979' AND  
G.DTE < GETDATE()  
GROUP  
BY A.ATTRIBUTETYPESID,  
A.[DESCRIPTION], A.ACTIVE  
UNION  
SELECT A.ATTRIBUTETYPESID,  
A.[DESCRIPTION] AS  
"CATEGORY",  
COUNT(AC.PARENTID) AS  
ATTRIBUTESCOUNT, A.ACTIVE,  
'GIFTATTRIBUTES' AS  
ATTRIBUTETYPE  
FROM AttributeTypes as A WITH  
(NOLOCK)  
LEFT JOIN GIFTATTRIBUTES as  
AC WITH (NOLOCK)  
ON
```

```
AC.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID
LEFT JOIN GIFT G WITH
(NOLOCK)
ON G.ID = AC.PARENTID
where A.RECORDTYPE = 2
GROUP
BY A.ATTRIBUTETYPESID,
A.[DESCRIPTION], A.ACTIVE
HAVING COUNT(AC.PARENTID)
= 0
UNION ALL
— ACTIONATTRIBUTES
SELECT A.ATTRIBUTETYPESID,
A.[DESCRIPTION] AS
“CATEGORY”,
COUNT(AC.PARENTID) AS
ATTRIBUTESCOUNT, A.ACTIVE,
‘ACTIONATTRIBUTES’ AS
ATTRIBUTETYPE
FROM AttributeTypes as A WITH
(NOLOCK)
LEFT JOIN ACTIONATTRIBUTES
as AC WITH (NOLOCK)
ON
AC.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID
LEFT JOIN ACTIONS AS
ACT WITH (NOLOCK)
ON AC.PARENTID=ACT.ID
where A.RECORDTYPE = 3
GROUP
```

```

BY A.ATTRIBUTETYPESID,
A.[DESCRIPTION], A.ACTIVE
UNION ALL
— EDUCATIONATTRIBUTES
SELECT A.ATTRIBUTETYPESID,
A.[DESCRIPTION] AS
“CATEGORY”,
COUNT(AC.PARENTID) AS
ATTRIBUTESCOUNT, A.ACTIVE,
‘EDUCATIONATTRIBUTES’ AS
ATTRIBUTETYPE
FROM AttributeTypes as A WITH
(NOLOCK)
LEFT JOIN
EDUCATIONATTRIBUTES as AC
WITH (NOLOCK)
ON
AC.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID
LEFT JOIN EDUCATION AS
E WITH (NOLOCK)
ON AC.PARENTID=E.ID
where A.RECORDTYPE = 4
GROUP
BY A.ATTRIBUTETYPESID,
A.[DESCRIPTION], A.ACTIVE
UNION ALL
— EVENTATTRIBUTES
SELECT A.ATTRIBUTETYPESID,
A.[DESCRIPTION] AS
“CATEGORY”,
COUNT(AC.PARENTID) AS

```

```

ATTRIBUTESCOUNT, A.ACTIVE,
'EVENTATTRIBUTES' AS
ATTRIBUTETYPE
FROM AttributeTypes as A WITH
(NOLOCK)
LEFT JOIN EVENTATTRIBUTES
as AC WITH (NOLOCK)
ON
AC.ATTRIBUTETYPESID=A.ATT
RIBUTETYPESID
LEFT JOIN SPECIAL_EVENT AS
SE WITH (NOLOCK)
ON AC.PARENTID=SE.ID
where A.RECORDTYPE = 5
GROUP
BY A.ATTRIBUTETYPESID,
A.[DESCRIPTION], A.ACTIVE
UNION ALL
— PARTICIPANTATTRIBUTES
SELECT A.ATTRIBUTETYPESID,
A.[DESCRIPTION] AS
“CATEGORY”,
COUNT(AC.PARENTID) AS
ATTRIBUTESCOUNT, A.ACTIVE,
'PARTICIPANTATTRIBUTES' AS
ATTRIBUTETYPE
FROM AttributeTypes as A WITH
(NOLOCK)
LEFT JOIN
PARTICIPANTATTRIBUTES as
AC WITH (NOLOCK)
ON

```

```
AC.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID
LEFT JOIN PARTICIPANTS AS P WITH (NOLOCK)
ON AC.PARENTID=P.ID
where A.RECORDTYPE = 6
GROUP
BY A.ATTRIBUTETYPESID,
A.[DESCRIPTION], A.ACTIVE
UNION ALL
— CAMPAIGNATTRIBUTES
SELECT A.ATTRIBUTETYPESID,
A.[DESCRIPTION] AS
“CATEGORY”,
COUNT(AC.PARENTID) AS
ATTRIBUTESCOUNT, A.ACTIVE,
‘CAMPAIGNATTRIBUTES’ AS
ATTRIBUTETYPE
FROM AttributeTypes as A WITH
(NOLOCK)
LEFT JOIN
CAMPAIGNATTRIBUTES as AC
WITH (NOLOCK)
ON
AC.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID
LEFT JOIN CAMPAIGN AS
CAM WITH (NOLOCK)
ON AC.PARENTID=CAM.ID
where A.RECORDTYPE = 7
GROUP
BY A.ATTRIBUTETYPESID,
```

```

A.[DESCRIPTION], A.ACTIVE
UNION ALL
— APPEALATTRIBUTES
SELECT A.ATTRIBUTETYPESID,
A.[DESCRIPTION] AS
“CATEGORY”,
COUNT(AC.PARENTID) AS
ATTRIBUTESCOUNT, A.ACTIVE,
‘APPEALATTRIBUTES’ AS
ATTRIBUTETYPE
FROM AttributeTypes as A WITH
(NOLOCK)
LEFT JOIN APPEALATTRIBUTES
as AC WITH (NOLOCK)
ON
AC.ATTRIBUTETYPESID=A.ATT
RIBUTETYPESID
LEFT JOIN APPEAL AS
APP WITH (NOLOCK)
ON AC.PARENTID=APP.ID
where A.RECORDTYPE = 8
GROUP
BY A.ATTRIBUTETYPESID,
A.[DESCRIPTION], A.ACTIVE
UNION ALL
— FUNDATTRIBUTES
SELECT A.ATTRIBUTETYPESID,
A.[DESCRIPTION] AS
“CATEGORY”,
COUNT(AC.PARENTID) AS
ATTRIBUTESCOUNT, A.ACTIVE,
‘FUNDATTRIBUTES’ AS

```

```

ATTRIBUTETYPE
FROM AttributeTypes as A WITH
(NOLOCK)
LEFT JOIN FUNDATTRIBUTES
as AC WITH (NOLOCK)
ON
AC.ATTRIBUTETYPESID=A.ATT
RIBUTETYPESID
LEFT JOIN FUND AS F WITH
(NOLOCK)
ON AC.PARENTID=F.ID
where A.RECORDTYPE = 10
GROUP
BY A.ATTRIBUTETYPESID,
A.[DESCRIPTION], A.ACTIVE
UNION ALL
— PACKAGEATTRIBUTES
SELECT A.ATTRIBUTETYPESID,
A.[DESCRIPTION] AS
“CATEGORY”,
COUNT(AC.PARENTID) AS
ATTRIBUTESCOUNT, A.ACTIVE,
‘PACKAGEATTRIBUTES’ AS
ATTRIBUTETYPE
FROM AttributeTypes as A WITH
(NOLOCK)
LEFT JOIN
PACKAGEATTRIBUTES as AC
WITH (NOLOCK)
ON
AC.ATTRIBUTETYPESID=A.ATT
RIBUTETYPESID

```

```
LEFT JOIN PACKAGE AS
P WITH (NOLOCK)
ON AC.PARENTID=P.ID
where A.RECORDTYPE = 11
GROUP
BY A.ATTRIBUTETYPESID,
A.[DESCRIPTION], A.ACTIVE
UNION ALL
```

```
—
CONSTITADDRESSATTRIBUTES
SELECT A.ATTRIBUTETYPESID,
A.[DESCRIPTION] AS
“CATEGORY”,
COUNT(AC.PARENTID) AS
ATTRIBUTESCOUNT, A.ACTIVE,
‘CONSTITADDRESSATTRIBUTE
S’ AS ATTRIBUTETYPE
FROM AttributeTypes as A WITH
(NOLOCK)
LEFT JOIN
CONSTITADDRESSATTRIBUTES
as AC WITH (NOLOCK)
ON
AC.ATTRIBUTETYPESID=A.ATT
RIBUTETYPESID
LEFT JOIN ADDRESS AS
ADD WITH (NOLOCK)
ON AC.PARENTID=ADD.ID
where A.RECORDTYPE = 12
GROUP
BY A.ATTRIBUTETYPESID,
A.[DESCRIPTION], A.ACTIVE
```



```

UNION ALL
— JOBATTRIBUTES
SELECT A.ATTRIBUTETYPESID,
A.[DESCRIPTION] AS
“CATEGORY”,
COUNT(AC.PARENTID) AS
ATTRIBUTESCOUNT, A.ACTIVE,
‘JOBATTRIBUTES’ AS
ATTRIBUTETYPE
FROM AttributeTypes as A WITH
(NOLOCK)
LEFT JOIN JOBATTRIBUTES as
AC WITH (NOLOCK)
ON
AC.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID
LEFT JOIN JOB AS JOB WITH
(NOLOCK)
ON AC.PARENTID=JOB.ID
where A.RECORDTYPE = 13
GROUP
BY A.ATTRIBUTETYPESID,
A.[DESCRIPTION], A.ACTIVE
UNION ALL
— MEMBERATTRIBUTES
SELECT A.ATTRIBUTETYPESID,
A.[DESCRIPTION] AS
“CATEGORY”,
COUNT(AC.PARENTID) AS
ATTRIBUTESCOUNT, A.ACTIVE,
‘MEMBERATTRIBUTES’ AS
ATTRIBUTETYPE

```

```

FROM AttributeTypes as A WITH
(NOLOCK)
LEFT JOIN
MEMBERATTRIBUTES as AC
WITH (NOLOCK)
ON
AC.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID
LEFT JOIN MEMBER AS
MEM WITH (NOLOCK)
ON AC.PARENTID=MEM.ID
where A.RECORDTYPE = 14
GROUP
BY A.ATTRIBUTETYPESID,
A.[DESCRIPTION], A.ACTIVE
UNION ALL
— IRA
RELATIONSHIPATTRIBUTES
SELECT A.ATTRIBUTETYPESID,
A.[DESCRIPTION] AS
“CATEGORY”,
COUNT(AC.PARENTID) AS
ATTRIBUTESCOUNT, A.ACTIVE,
‘IRA
RELATIONSHIPATTRIBUTES’ AS
ATTRIBUTETYPE
FROM AttributeTypes as A WITH
(NOLOCK)
LEFT JOIN
RELATIONSHIPATTRIBUTES as
AC WITH (NOLOCK)
ON

```

```

AC.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID
LEFT JOIN
CONSTIT_RELATIONSHIPS AS
REL WITH (NOLOCK)
ON AC.PARENTID=REL.ID
where A.RECORDTYPE = 15
GROUP
BY A.ATTRIBUTETYPESID,
A.[DESCRIPTION], A.ACTIVE
UNION ALL
— ORA
RELATIONSHIPATTRIBUTES
SELECT A.ATTRIBUTETYPESID,
A.[DESCRIPTION] AS
“CATEGORY”,
COUNT(AC.PARENTID) AS
ATTRIBUTESCOUNT, A.ACTIVE,
‘ORA
RELATIONSHIPATTRIBUTES’ AS
ATTRIBUTETYPE
FROM AttributeTypes as A WITH
(NOLOCK)
LEFT JOIN
RELATIONSHIPATTRIBUTES as
AC WITH (NOLOCK)
ON
AC.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID
LEFT JOIN
CONSTIT_RELATIONSHIPS AS
REL

```

```

ON AC.PARENTID=REL.ID
where A.RECORDTYPE = 16
GROUP
BY A.ATTRIBUTETYPESID,
A.[DESCRIPTION], A.ACTIVE
UNION ALL
—
CONSTITUENT_BANKATTRIBU
ES
SELECT A.ATTRIBUTETYPESID,
A.[DESCRIPTION] AS
“CATEGORY”,
COUNT(AC.PARENTID) AS
ATTRIBUTESCOUNT, A.ACTIVE,
‘CONSTITUENT_BANKATTRIBU
TES’ AS ATTRIBUTETYPE
FROM AttributeTypes as A WITH
(NOLOCK)
LEFT JOIN
CONSTITUENT_BANKATTRIBU
ES as AC WITH (NOLOCK)
ON
AC.ATTRIBUTETYPESID=A.ATT
RIBUTETYPESID
LEFT JOIN
CONSTITUENT_BANK AS
COB WITH (NOLOCK)
ON AC.PARENTID=COB.ID
where A.RECORDTYPE = 17
GROUP
BY A.ATTRIBUTETYPESID,
A.[DESCRIPTION], A.ACTIVE

```

```
UNION ALL
— PROPOSALATTRIBUTES
SELECT A.ATTRIBUTETYPESID,
A.[DESCRIPTION] AS
“CATEGORY”,
COUNT(AC.PARENTID) AS
ATTRIBUTESCOUNT, A.ACTIVE,
‘PROPOSALATTRIBUTES’ AS
ATTRIBUTETYPE
FROM AttributeTypes as A WITH
(NOLOCK)
LEFT JOIN
PROPOSALATTRIBUTES as AC
WITH (NOLOCK)
ON
AC.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID
LEFT JOIN PROPOSAL AS
PRO WITH (NOLOCK)
ON AC.PARENTID=PRO.ID
where A.RECORDTYPE = 18
GROUP
BY A.ATTRIBUTETYPESID,
A.[DESCRIPTION], A.ACTIVE;
```

## **Code Tables relationships code**

```
/* Relationships CODETABLES CODE */
```

```
SELECT *
```

```
FROM CODETABLES AS C
```

```
WHERE C.CODETABLESID=5008;
```

## **TableEntries Search**

```
/*TABLEENTRIES*/
```

```
Select A.[DESCRIPTION], T.LONGDESCRIPTION, *  
from AttributeTypes as A, TABLEENTRIES as T  
where A.CODETABLESID=T.[CODETABLESID]  
AND A.ATTRIBUTEID = 'ENTER ATTRIBUTEID'
```

## **PARTICIPANTATTRIBUTES SEARCH**

/\*\*PARTICIPANTATTRIBUTES\*\*/

```
select A.[DESCRIPTION], MIN(PA.BOOLEAN),MAX( PA.BOOLEAN), MIN(PA.  
[TEXT]), MAX(PA.[TEXT]) , MIN(PA.NUM), MAX(PA.NUM) ,  
MIN(PA.CURRENCY), MAX(PA.CURRENCY) , MIN(PA.[DATETIME]), MAX(PA.  
[DATETIME]), MIN(PA.COMMENTS),MAX(PA.COMMENTS)
```

```
from PARTICIPANTATTRIBUTES as PA, AttributeTypes as A
```

```
where PA.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID
```

```
AND A.ATTRIBUTETYPESID = 'ENTER ATTRIBUTETYPESID'
```

```
GROUP BY A.[DESCRIPTION];
```



## **JOBATTRIBUTES SEARCH**

**/\*JOBATTRIBUTES\*/**

**select A.[DESCRIPTION], MIN(JB.BOOLEAN),MAX( JB.BOOLEAN), MIN(JB.  
[TEXT]), MAX(JB.[TEXT]) , MIN(JB.NUM), MAX(JB.NUM) , MIN(JB.CURRENCY),  
MAX(JB.CURRENCY) , MIN(JB.[DATETIME]), MAX(JB.[DATETIME]),  
MIN(JB.COMMENTS),MAX(JB.COMMENTS)**

**from JOBATTRIBUTES as JB, AttributeTypes as A**

**where JB.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID**

**AND A.ATTRIBUTETYPESID = 'ENTER ATTRIBUTETYPESID'**

**GROUP BY A.[DESCRIPTION];**

## PROPOSALATTRIBUTES SEARCH

/\*\*PROPOSALATTRIBUTES\*\*/

```
select A.[DESCRIPTION], MIN(PR.BOOLEAN),MAX( PR.BOOLEAN), MIN(PR.  
[TEXT]), MAX(PR.[TEXT]) , MIN(PR.NUM), MAX(PR.NUM) ,  
MIN(PR.CURRENCY), MAX(PR.CURRENCY) , MIN(PR.[DATETIME]), MAX(PR.  
[DATETIME]), MIN(PR.COMMENTS),MAX(PR.COMMENTS)
```

```
from PROPOSALATTRIBUTES as PR, AttributeTypes as A
```

```
where PR.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID
```

```
AND A.ATTRIBUTETYPESID = 'ENTER ATTRIBUTETYPESID'
```

```
GROUP BY A.[DESCRIPTION];
```

## PACKAGEATTRIBUTES SEARCH

/\*\*\*/PACKAGEATTRIBUTES\*\*\*/

```
select A.[DESCRIPTION], MIN(PK.BOOLEAN),MAX( PK.BOOLEAN), MIN(PK.  
[TEXT]), MAX(PK.[TEXT]) , MIN(PK.NUM), MAX(PK.NUM) ,  
MIN(PK.CURRENCY), MAX(PK.CURRENCY) , MIN(PK.[DATETIME]), MAX(PK.  
[DATETIME]), MIN(PK.COMMENTS),MAX(PK.COMMENTS)
```

```
from PACKAGEATTRIBUTES as PK, AttributeTypes as A
```

```
where PK.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID
```

```
AND A.ATTRIBUTETYPESID = 'ENTER ATTRIBUTETYPESID'
```

```
GROUP BY A.[DESCRIPTION];
```

## **ACTIONATTRIBUTES SEARCH**

**/\*\*ACTIONATTRIBUTES\*\*/**

```
select A.[DESCRIPTION], MIN(AA.BOOLEAN),MAX( AA.BOOLEAN), MIN(AA.  
[TEXT]), MAX(AA.[TEXT]) , MIN(AA.NUM), MAX(AA.NUM) ,  
MIN(AA.CURRENCY), MAX(AA.CURRENCY) , MIN(AA.[DATETIME]), MAX(AA.  
[DATETIME]), MIN(AA.COMMENTS),MAX(AA.COMMENTS)
```

```
from ACTIONATTRIBUTES as AA, AttributeTypes as A
```

```
where AA.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID
```

```
AND A.ATTRIBUTETYPESID = 'ENTER ATTRIBUTETYPESID'
```

```
GROUP BY A.[DESCRIPTION];
```

## CONSTITADDRESSATTRIBUTES SEARCH

/\*CONSTITADDRESSATTRIBUTES\*/

```
select A.[DESCRIPTION], MIN(CAD.BOOLEAN),MAX( CAD.BOOLEAN),
MIN(CAD.[TEXT]), MAX(CAD.[TEXT]) , MIN(CAD.NUM), MAX(CAD.NUM) ,
MIN(CAD.CURRENCY), MAX(CAD.CURRENCY) , MIN(CAD.[DATETIME]),
MAX(CAD.[DATETIME]), MIN(CAD.COMMENTS),MAX(CAD.COMMENTS)
from CONSTITADDRESSATTRIBUTES as CAD, AttributeTypes as A
where CAD.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID
AND A.ATTRIBUTETYPESID = 'ENTER ATTRIBUTETYPESID'
GROUP BY A.[DESCRIPTION];
```

## GiftAttributes SEARCH

/\*\*GiftAttributes V2\*\*/

```
select A.[DESCRIPTION], MIN(GA.BOOLEAN),MAX( GA.BOOLEAN), MIN(GA.  
[TEXT]), MAX(GA.[TEXT]) , MIN(GA.NUM), MAX(GA.NUM) ,  
MIN(GA.CURRENCY), MAX(GA.CURRENCY) , MIN(GA.[DATETIME]), MAX(GA.  
[DATETIME]), MIN(GA.COMMENTS),MAX(GA.COMMENTS)
```

```
from GIFTATTRIBUTES as GA, AttributeTypes as A
```

```
where GA.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID
```

```
AND A.ATTRIBUTETYPESID = 'ENTER ATTRIBUTETYPESID'
```

```
GROUP BY A.[DESCRIPTION];
```

## ConstituentAttributes SEARCH

/\*\*\*/ConstituentAttributes V2\*\*\*/

```
select A.[DESCRIPTION], MIN(CAT.BOOLEAN),MAX( CAT.BOOLEAN), MIN(CAT.  
[TEXT]), MAX(CAT.[TEXT]) , MIN(CAT.NUM), MAX(CAT.NUM) ,  
MIN(CAT.CURRENCY), MAX(CAT.CURRENCY) , MIN(CAT.[DATETIME]),  
MAX(CAT.[DATETIME]), MIN(CAT.COMMENTS),MAX(CAT.COMMENTS)  
from CONSTITUENTATTRIBUTES as CAT, AttributeTypes as A  
where CAT.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID  
AND A.ATTRIBUTETYPESID = 'ENTER ATTRIBUTETYPESID'  
GROUP BY A.[DESCRIPTION];
```

## APPEALATTRIBUTES SEARCH

/\*\*APPEALATTRIBUTES\*\*/

```
select A.[DESCRIPTION], MIN(AP.BOOLEAN),MAX( AP.BOOLEAN), MIN(AP.  
[TEXT]), MAX(AP.[TEXT]) , MIN(AP.NUM), MAX(AP.NUM) ,  
MIN(AP.CURRENCY), MAX(AP.CURRENCY) , MIN(AP.[DATETIME]), MAX(AP.  
[DATETIME]), MIN(AP.COMMENTS),MAX(AP.COMMENTS)
```

```
from APPEALATTRIBUTES as AP, AttributeTypes as A
```

```
where AP.ATTRIBUTEID=A.ATTRIBUTEID
```

```
AND A.ATTRIBUTEID = 'ENTER ATTRIBUTEID'
```

```
GROUP BY A.[DESCRIPTION];
```



## CONSTITUENT\_BANKATTRIBUTES SEARCH

/\*\*CONSTITUENT\_BANKATTRIBUTES\*\*/

```
select A.[DESCRIPTION], MIN(CAB.BOOLEAN),MAX( CAB.BOOLEAN),
MIN(CAB.[TEXT]), MAX(CAB.[TEXT]) , MIN(CAB.NUM), MAX(CAB.NUM) ,
MIN(CAB.CURRENCY), MAX(CAB.CURRENCY) , MIN(CAB.[DATETIME]),
MAX(CAB.[DATETIME]), MIN(CAB.COMMENTS),MAX(CAB.COMMENTS)
from CONSTITUENT_BANKATTRIBUTES as CAB, AttributeTypes as A
where CAB.ATTRIBUTEID=A.ATTRIBUTEID
AND A.ATTRIBUTEID = 'ENTER ATTRIBUTEID'
GROUP BY A.[DESCRIPTION];
```

## EVENTATTRIBUTES SEARCH

/\*\*EVENTATTRIBUTES\*\*/

```
select A.[DESCRIPTION], MIN(E.BOOLEAN),MAX( E.BOOLEAN), MIN(E.[TEXT]),  
MAX(E.[TEXT]) , MIN(E.NUM), MAX(E.NUM) , MIN(E.CURRENCY),  
MAX(E.CURRENCY) , MIN(E.[DATETIME]), MAX(E.[DATETIME]),  
MIN(E.COMMENTS),MAX(E.COMMENTS)
```

```
from EVENTATTRIBUTES as E, AttributeTypes as A
```

```
where E.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID
```

```
AND A.ATTRIBUTETYPESID = 'ENTER ATTRIBUTETYPESID'
```

```
GROUP BY A.[DESCRIPTION];
```

## FUNDATTRIBUTES SEARCH

/\*\*FUNDATTRIBUTES\*\*/

```
select A.[DESCRIPTION], MIN(F.BOOLEAN),MAX( F.BOOLEAN), MIN(F.[TEXT]),  
MAX(F.[TEXT]) , MIN(F.NUM), MAX(F.NUM) , MIN(F.CURRENCY),  
MAX(F.CURRENCY) , MIN(F.[DATETIME]), MAX(F.[DATETIME]),  
MIN(F.COMMENTS),MAX(F.COMMENTS)
```

```
from FUNDATTRIBUTES as F, AttributeTypes as A
```

```
where F.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID
```

```
AND A.ATTRIBUTETYPESID = 'ENTER ATTRIBUTETYPESID'
```

```
GROUP BY A.[DESCRIPTION];
```

## RELATIONSHIPATTRIBUTES SEARCH

/\*\*RELIATIONSHIPATTRIBUTES\*\*/

```
select A.[DESCRIPTION], MIN(R.BOOLEAN),MAX( R.BOOLEAN), MIN(R.  
[TEXT]), MAX(R.[TEXT]) , MIN(R.NUM), MAX(R.NUM) , MIN(R.CURRENCY),  
MAX(R.CURRENCY) , MIN(R.[DATETIME]), MAX(R.[DATETIME]),  
MIN(R.COMMENTS),MAX(R.COMMENTS)
```

```
from RELATIONSHIPATTRIBUTES as R, AttributeTypes as A
```

```
where R.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID
```

```
AND A.ATTRIBUTETYPESID = 'ENTER ATTRIBUTETYPESID'
```

```
GROUP BY A.[DESCRIPTION];
```

## **JOBATTRIBUTES SEARCH**

**/\*\*JOBATTRIBUTES\*\*/**

```
select A.[DESCRIPTION], R.LAST_NAME,R.FIRST_NAME,  
MIN(JB.BOOLEAN),MAX( JB.BOOLEAN), MIN(JB.[TEXT]), MAX(JB.[TEXT]) ,  
MIN(JB.NUM), MAX(JB.NUM) , MIN(JB.CURRENCY), MAX(JB.CURRENCY) ,  
MIN(JB.[DATETIME]), MAX(JB.[DATETIME]),  
MIN(JB.COMMENTS),MAX(JB.COMMENTS)  
  
from JOBATTRIBUTES as JB, AttributeTypes as A, RECORDS AS R  
  
where JB.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID  
  
AND JB.ConstitID=R.ID  
  
AND A.ATTRIBUTETYPESID = 'ENTER ATTRIBUTETYPESID'  
  
GROUP BY A.[DESCRIPTION],R.LAST_NAME,R.FIRST_NAME;
```

## **PHONE TYPES TABLE ENTRIES E.G. BUSINESS PHONE**

```
SELECT LONGDESCRIPTION,*  
FROM CODETABLES AS C INNER JOIN TABLEENTRIES AS T  
ON C.CODETABLESID=T.CODETABLESID  
WHERE C.[NAME] LIKE 'Phone Types';
```

## **COUNTRIES TABLE ENTRIES E.G. UNITED STATES**

```
SELECT LONGDESCRIPTION,*  
FROM CODETABLES AS C INNER JOIN TABLEENTRIES AS T  
ON C.CODETABLESID=T.CODETABLESID  
WHERE C.[NAME] LIKE 'COUNTRY';
```

## Tableentries Sample Data (Group Concat via AttributeTypes)

/\*TABLEENTRIES SAMPLE DATA\*/

Select DISTINCT T.[CODETABLESID], C.[NAME], AT.[DESCRIPTION], AT.  
[ATTRIBUTETYPESID],

Substring((Select ' , ' + T2.[LONGDESCRIPTION] FROM TABLEENTRIES as T2  
Where T.[CODETABLESID]=T2.[CODETABLESID] ORDER BY T2.  
[LONGDESCRIPTION] For XML Path('')),2,8000) As LONGDESCRIPTIONGROUP,

Substring((Select ' , ' + T2.[SHORTDESCRIPTION] FROM TABLEENTRIES as T2  
Where T.[CODETABLESID]=T2.[CODETABLESID] ORDER BY T2.  
[SHORTDESCRIPTION] For XML Path('')),2,8000) As SHORTDESCRIPTIONGROUP

from TABLEENTRIES as T

LEFT OUTER JOIN

AttributeTypes AS AT

ON T.[CODETABLESID]=AT.[CODETABLESID]

inner join CODETABLES AS C

ON T.[CODETABLESID]=C.[CODETABLESID]

—WHERE AT.TYPEOFDATA = 6

group by T.[CODETABLESID], C.[NAME], AT.[DESCRIPTION], AT.  
[ATTRIBUTETYPESID], T.[LONGDESCRIPTION] ,T.[SHORTDESCRIPTION]

ORDER BY T.[CODETABLESID], C.[NAME], AT.[DESCRIPTION], AT.  
[ATTRIBUTETYPESID], LONGDESCRIPTIONGROUP,  
SHORTDESCRIPTIONGROUP;



## **NON ALPHANUM CHARS SEARCH**

```
SELECT P.NUM, COUNT(P.NUM) AS NUMCOUNT /*R.ID AS  
RECORDSIDCOUNT*/
```

```
FROM RECORDS AS R INNER JOIN CONSTIT_ADDRESS AS CA
```

```
ON R.ID=CA.CONSTIT_ID
```

```
INNER JOIN CONSTIT_ADDRESS_PHONES AS CAP
```

```
ON CA.ID=CAP.CONSTITADDRESSID
```

```
INNER JOIN PHONES AS P
```

```
ON P.PHONESID=CAP.PHONESID
```

```
WHERE (P.NUM NOT LIKE '%[0-9a-z]%)
```

```
AND (R.IS_CONSTITUENT = -1)
```

```
GROUP BY P.NUM WITH ROLLUPGROUP BY P.NUM WITH ROLLUP
```

## **SPACED OR NULL CHARS SEARCH**

```
SELECT P.NUM, COUNT(P.NUM) AS NUMCOUNT, COUNT(R.ID) AS
RECORDSIDCOUNT
FROM RECORDS AS R INNER JOIN CONSTIT_ADDRESS AS CA
ON R.ID=CA.CONSTIT_ID
INNER JOIN CONSTIT_ADDRESS_PHONES AS CAP
ON CA.ID=CAP.CONSTITADDRESSID
INNER JOIN PHONES AS P
ON P.PHONESID=CAP.PHONESID
WHERE (P.NUM IS NULL OR P.NUM = '' OR P.NUM LIKE '%^%')
AND (R.IS_CONSTITUENT = -1)
GROUP BY P.NUM WITH ROLLUP;
```

## **ACTION STATUS AND TYPE TO TABLEENTRIES JOIN**

**/\* ACTION STATUS AND TYPE TO TABLEENTRIES JOIN \*/**

**SELECT TOP 10 T1.LONGDESCRIPTION [ACTION STATUS],  
T2.LONGDESCRIPTION [ACTION TYPE], \* FROM TABLEENTRIES T1,  
TABLEENTRIES T2, ACTIONS A WHERE T1.TABLEENTRIESID=A.STATUS AND  
T2.TABLEENTRIESID=A.TYPE;**

## **COURSES PARTICIPANT ATTENDED COUNT**

/\* PARTICIPANT ATTENDED COUNT \*/

SELECT COUNT(P.Attended) [ATTENDED COUNT],

CASE P.Attended

WHEN -1 THEN 'YES'

WHEN 0 THEN 'NO'

END AS [PARTICIPANT ATTENDED]

FROM PARTICIPANTS P, SPECIAL\_EVENT SE

WHERE P.EventID=SE.ID

AND SE.CATEGORY=3

GROUP BY P.Attended WITH ROLLUP;

## **COURSES PARTICIPANT IS INSTRUCTOR COUNT**

**/\* PARTICIPANT IS INSTRUCTOR COUNT \*/**

**SELECT COUNT(P.IsInstructor) [IS AN INSTRUCTOR COUNT],**

**CASE P.IsInstructor**

**WHEN -1 THEN 'YES'**

**WHEN 0 THEN 'NO'**

**END AS [PARTICIPANT IS AN INSTRUCTOR]**

**FROM PARTICIPANTS P, SPECIAL\_EVENT SE**

**WHERE P.EventID=SE.ID**

**AND SE.CATEGORY=3**

**GROUP BY P.IsInstructor WITH ROLLUP;**

## **COURSES PARTICIPANT PARTICIPATION COUNT**

**/\* PARTICIPANT PARTICIPATION COUNT \*/**

**SELECT COUNT(P.Participation) [PARTICIPANT PARTICIPATION COUNT],  
T.LONGDESCRIPTION PARTICIPATION**

**FROM PARTICIPANTS P, SPECIAL\_EVENT SE, TABLEENTRIES T**

**WHERE P.EventID=SE.ID**

**AND T.TABLEENTRIESID=P.Participation**

**AND SE.CATEGORY=3**

**GROUP BY T.LONGDESCRIPTION WITH ROLLUP;**

## **COURSES PARTICIPANT STATUS COUNT**

**/\* PARTICIPANT STATUS COUNT \*/**

```
SELECT COUNT(P.[Status]) [PARTICIPANT STATUS COUNT],  
T.LONGDESCRIPTION [PARTICIPANT STATUS]  
FROM PARTICIPANTS P, SPECIAL_EVENT SE, TABLEENTRIES T  
WHERE P.EventID=SE.ID  
AND T.TABLEENTRIESID=P.[Status]  
AND SE.CATEGORY=3  
GROUP BY T.LONGDESCRIPTION WITH ROLLUP;
```

## **COURSES PARTICIPANT RESPONSE COUNT**

**/\* PARTICIPANT RESPONSE COUNT \*/**

**SELECT COUNT(P.[Response]) [PARTICIPANT RESPONSE COUNT],  
T.LONGDESCRIPTION [PARTICIPANT RESPONSE]**

**FROM PARTICIPANTS P, SPECIAL\_EVENT SE, TABLEENTRIES T**

**WHERE P.EventID=SE.ID**

**AND T.TABLEENTRIESID=P.[Response]**

**AND SE.CATEGORY=3**

**GROUP BY T.LONGDESCRIPTION WITH ROLLUP;**



## **COURSES EVENT CONTACT NAMES**

**/\* EVENT CONTACT NAMES \*/**

```
SELECT COUNT(SE.LocationContact) [EVENT CONTACT NAMES], SE.NAME  
FROM SPECIAL_EVENT SE  
WHERE SE.CATEGORY=3  
AND SE.LocationContact IS NOT NULL  
GROUP BY SE.NAME WITH ROLLUP;
```

## **COURSES PARTICIPANT EVENT CONTACT NAMES**

**/\* TRAINING PARTICIPANT EVENT CONTACT NAMES \*/**

**SELECT COUNT(P.CONTACT) AS [PARTICIPANT CONTACT COUNT],**

**CASE SE.CATEGORY**

**WHEN 3 THEN 'TRAINING'**

**ELSE '?'**

**END AS**

**[EVENT CATEGORY]**

**FROM SPECIAL\_EVENT AS SE INNER JOIN PARTICIPANTS AS P ON SE.ID =  
P.EVENTID**

**WHERE P.Contact IS NOT NULL**

**AND SE.CATEGORY =3**

**—AND P.Contact LIKE '%ENTER NAME%'**

**GROUP BY SE.CATEGORY /\*P.CONTACT\*/;**

**/\* Industries AKA Activity drop down list values \*/**

/\* Industries AKA Activity list values \*/

Select DISTINCT T.LONGDESCRIPTION as [INDUSTRIES AKA ACTIVITY],  
T.ACTIVE, T.CODETABLESID, T.SHORTDESCRIPTION, T.TABLEENTRIESID

from TABLEENTRIES as T

WHERE T.CODETABLESID = [ENTER INDUSTRY CODE];

**/\* Profession drop down list values \*/**

/\* Profession list values \*/

```
Select DISTINCT T.LONGDESCRIPTION as [PROFESSION], T.ACTIVE,  
T.CODETABLESID, T.SHORTDESCRIPTION, T.TABLEENTRIESID  
from TABLEENTRIES as T  
WHERE T.CODETABLESID = [ENTER PROFESSIONS CODE TABLES ID];
```

**/\* Industries AKA Activity entered values that have been inputted or imported into RE\*/**

**/\* Industries AKA Activity entered values \*/**

```
SELECT DISTINCT T1.LONGDESCRIPTION AS INDUSTRY, T1.ACTIVE,  
T1.CODETABLESID,T1.SHORTDESCRIPTION,T1.TABLEENTRIESID  
FROM CONSTIT_RELATIONSHIPS AS CR1, TABLEENTRIES AS T1  
WHERE CR1.INDUSTRY=T1.TABLEENTRIESID;
```

**/\* Profession entered values that have been inputted or imported into RE  
\*/**

/\* Profession entered values \*/

```
SELECT DISTINCT T2.LONGDESCRIPTION AS PROFESSION, T2.ACTIVE,  
T2.CODETABLESID,T2.SHORTDESCRIPTION,T2.TABLEENTRIESID  
FROM CONSTIT_RELATIONSHIPS AS CR2, TABLEENTRIES AS T2  
WHERE CR2.PROFESSION=T2.TABLEENTRIESID;
```

## **SQL CODE FOR DISPLAYING ALL BASE TABLES IN SSMS.**

```
select *  
from information_schema.tables  
where table_type like 'BASE TABLE';
```

## SQL RECORDS AND INDUSTRY CROSS REFERENCE WITH TABLEENTRIES AND USERS

```
/* RECORDS INDUSTRY CROSS REFERENCE WITH TABLEENTRIES AND  
USERS */
```

```
SELECT RECORDS.ORG_NAME [R.ORG_NAME], RECORDS.CONSTITUENT_ID,  
RECORDS.ADDED_BY [R.ADDED_BY], U1.[NAME] [U1.NAME],  
RECORDS.LAST_CHANGED_BY [R.LAST_CHANGED_BY], U2.[NAME]  
[U2.NAME], T1.LONGDESCRIPTION [T.INDUSTRY], T1.TABLEENTRIESID  
[T.TABLEENTRIESID], T1.ACTIVE [T.ACTIVE], T1.ADDED_BY_ID  
[T.ADDED_BY_ID], U3.[NAME] [U3.NAME], T1.ADDEDBYID [T.ADDEDBYID],  
U4.[NAME] [U4.NAME], T1.LAST_CHANGED_BY_ID  
[T.LAST_CHANGED_BY_ID], U5.[NAME] [U5.NAME], T1.LASTCHANGEDBYID  
[T.LASTCHANGEDBYID], U6.[NAME] [U6.NAME]  
  
FROM RECORDS INNER JOIN TABLEENTRIES T1  
  
ON RECORDS.INDUSTRY=T1.TABLEENTRIESID  
  
LEFT JOIN USERS U1 ON U1.[USER_ID]=RECORDS.ADDED_BY  
LEFT JOIN USERS U2 ON U2.[USER_ID]=RECORDS.LAST_CHANGED_BY  
LEFT JOIN  USERS U3 ON U3.[USER_ID]=T1.ADDED_BY_ID  
LEFT JOIN  USERS U4 ON U4.[USER_ID]=T1.ADDEDBYID  
LEFT JOIN  USERS U5 ON U5.[USER_ID]=T1.LAST_CHANGED_BY_ID  
LEFT JOIN  USERS U6 ON U6.[USER_ID]=T1.LASTCHANGEDBYID  
  
WHERE RECORDS.IS_CONSTITUENT=-1  
  
GROUP BY RECORDS.ORG_NAME, RECORDS.CONSTITUENT_ID,  
RECORDS.ADDED_BY, U1.[NAME], RECORDS.LAST_CHANGED_BY, U2.  
[NAME], T1.LONGDESCRIPTION, T1.TABLEENTRIESID, T1.ACTIVE,  
T1.ADDED_BY_ID, U3.[NAME], T1.ADDEDBYID, U4.[NAME],  
T1.LAST_CHANGED_BY_ID, U5.[NAME], T1.LASTCHANGEDBYID, U6.[NAME];
```



**/\* SQL PHONES, CONSTITUENT AND ADDRESS EXTRACT \*/**

/\* SQL PHONES, CONSTITUENT AND ADDRESS EXTRACT Answerphone\*/

```
SELECT R.CONSTITUENT_ID, R.LAST_NAME, T.LONGDESCRIPTION [PHONE
TYPE], P.NUM [PHONE NUMBER], [CA].IMPORT_ID [ADDRESS IMPORT ID],
[ADD].POST_CODE, T2.LONGDESCRIPTION [COUNTRY], CA.DATE_ADDED
[ADDRESS DATE ADDED], CR.IMPORT_ID [RELATED CONSTITUENT IMPORT
ID], R2.CONSTITUENT_ID [RELATED CONSTITUENT ID], R2.[LAST_NAME]
[RELATED SURNAME]
FROM RECORDS AS R
INNER JOIN CONSTIT_ADDRESS AS CA
ON R.ID=CA.CONSTIT_ID
INNER JOIN [ADDRESS] [ADD]
ON [ADD].ID=CA.ADDRESS_ID
INNER JOIN CONSTIT_ADDRESS_PHONES AS CAP
ON CA.ID=CAP.CONSTITADDRESSID
INNER JOIN PHONES AS P
ON P.PHONESID=CAP.PHONESID
RIGHT JOIN TABLEENTRIES AS T
ON P.PHONETYPEID=T.TABLEENTRIESID
RIGHT JOIN TABLEENTRIES T2
ON [ADD].COUNTRY=T2.TABLEENTRIESID
LEFT JOIN CONSTIT_RELATIONSHIPS CR
ON CR.CONSTIT_ID=R.ID
LEFT JOIN RECORDS R2
ON R2.ID=CR.RELATION_ID
WHERE T.CODETABLESID = [ENTER PHONES ID]
AND R.IS_CONSTITUENT = -1
AND T.LONGDESCRIPTION = [ENTER PHONES LONGDESCRIPTION VALUE]
GROUP BY R.CONSTITUENT_ID, R.LAST_NAME, T.LONGDESCRIPTION, P.NUM,
[CA].IMPORT_ID, [ADD].POST_CODE, T2.LONGDESCRIPTION,
CA.DATE_ADDED, CR.IMPORT_ID, R2.CONSTITUENT_ID, R2.[LAST_NAME];
```

## SQL NON-ALPHABETIC NAMES

—SQL NON-ALPHABETIC NAMES

```
SELECT R.CONSTITUENT_ID, R.FIRST_NAME, R.LAST_NAME
FROM RECORDS R
WHERE R.IS_CONSTITUENT = -1
AND R.KEY_INDICATOR = 'i'
AND R.FIRST_NAME NOT LIKE '%[a-z]%'
OR (LEN(R.FIRST_NAME)>15 AND R.FIRST_NAME NOT LIKE '%-%' AND
R.FIRST_NAME LIKE '% %')
or R.FIRST_NAME like '%~%'
or R.FIRST_NAME like '%`%'
or R.FIRST_NAME like '%!%'
or R.FIRST_NAME like '%“%’
or R.FIRST_NAME like '%£%'
or R.FIRST_NAME like '%$%'
or R.FIRST_NAME like '%”%’
or R.FIRST_NAME like '%^%'
or R.FIRST_NAME like '%&%'
or R.FIRST_NAME like '%*%'
or R.FIRST_NAME like '%(%’
or R.FIRST_NAME like '%)%’
or R.FIRST_NAME like '%”_%’
or R.FIRST_NAME like '%-‘
or R.FIRST_NAME like ‘-%’
or R.FIRST_NAME like ‘%+%’
or R.FIRST_NAME like ‘%=%'
or R.FIRST_NAME like ‘%{%'
or R.FIRST_NAME like ‘%[%’
or R.FIRST_NAME like ‘%}%’
or R.FIRST_NAME like ‘%]%'
or R.FIRST_NAME like ‘%:%’
or R.FIRST_NAME like ‘%;%’
```

or R.FIRST\_NAME like '@%'  
or R.FIRST\_NAME like '%''  
or R.FIRST\_NAME like '''%'  
or R.FIRST\_NAME like '~%'  
or R.FIRST\_NAME like '#%'  
or R.FIRST\_NAME like '%|%'  
or R.FIRST\_NAME like '%\%'  
or R.FIRST\_NAME like '%<%'  
or R.FIRST\_NAME like '%,%'  
or R.FIRST\_NAME like '%>%'  
—or R.FIRST\_NAME like '%.'  
—or R.FIRST\_NAME like '%.'  
or R.FIRST\_NAME like '%.%'  
or R.FIRST\_NAME like '%?%'  
or R.FIRST\_NAME like '%/%'  
or R.FIRST\_NAME like '%1%'  
or R.FIRST\_NAME like '%2%'  
or R.FIRST\_NAME like '%3%'  
or R.FIRST\_NAME like '%4%'  
or R.FIRST\_NAME like '%5%'  
or R.FIRST\_NAME like '%6%'  
or R.FIRST\_NAME like '%7%'  
or R.FIRST\_NAME like '%8%'  
or R.FIRST\_NAME like '%9%'  
or R.FIRST\_NAME like '%0%';

## Phone Types Counts

```
SELECT T.LONGDESCRIPTION, COUNT(P.NUM) AS COUNT_PHONE_ENTRIES,  
COUNT(*) COUNT_ALL_ROWS  
FROM  
DBO.RECORDS AS R  
INNER JOIN DBO.QCONSTIT_ADDRESS AS QCA  
ON R.ID = QCA.CONSTIT_ID  
INNER JOIN DBO.CONSTIT_ADDRESS_PHONES AS CAP  
ON QCA.ID = CAP.CONSTITADDRESSID  
INNER JOIN DBO.PHONES AS P ON CAP.PHONESID = P.PHONESID  
INNER JOIN TABLEENTRIES AS T ON T.TABLEENTRIESID=P.PHONETYPEID  
WHERE  
(  
(P.PHONETYPEID IS NOT NULL)  
AND (R.IS_CONSTITUENT=-1)  
)  
GROUP BY T.LONGDESCRIPTION  
WITH ROLLUP;
```

## **DUPLICATE CONSTITUENTATTRIBUTES**

**/\* DUPLICATE CONSTITUENTATTRIBUTES \*/**

```
SELECT R.CONSTITUENT_ID, COUNT(*) AS COUNTS
FROM AttributeTypes as A WITH (NOLOCK)
INNER JOIN CONSTITUENTATTRIBUTES as AC WITH (NOLOCK)
ON AC.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID
INNER JOIN RECORDS AS R WITH (NOLOCK)
ON R.ID=AC.PARENTID
where A.RECORDTYPE = 1
AND R.IS_CONSTITUENT=-1
AND AC.ATTRIBUTETYPESID= [ENTER ID]
GROUP BY R.CONSTITUENT_ID
HAVING COUNT(AC.ATTRIBUTETYPESID)>1;
```

## **DUPLICATES GIFTATTRIBUTES**

/\* DUPLICATES GIFTATTRIBUTES \*/

```
SELECT R.CONSTITUENT_ID, COUNT(AC.PARENTID) AS ATTRIBUTESCOUNT
FROM AttributeTypes as A WITH (NOLOCK)
INNER JOIN GIFTATTRIBUTES as AC WITH (NOLOCK)
ON AC.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID
INNER JOIN GIFT G WITH (NOLOCK)
ON G.ID = AC.PARENTID
INNER JOIN CONSTIT_GIFTS AS CG
ON CG.GIFT_ID=G.ID
INNER JOIN RECORDS AS R WITH (NOLOCK)
ON CG.CONSTIT_ID=R.ID
WHERE A.RECORDTYPE = 2
AND R.IS_CONSTITUENT=-1
AND AC.ATTRIBUTETYPESID= [ENTER ID]
GROUP BY R.CONSTITUENT_ID
HAVING COUNT(AC.PARENTID)>1;
```

## **DUPLICATE CONSTITADDRESSATTRIBUTES**

```
/* DUPLICATE CONSTITADDRESSATTRIBUTES */  
SELECT R.CONSTITUENT_ID, COUNT(*) AS COUNTS  
FROM AttributeTypes as A WITH (NOLOCK)  
INNER JOIN CONSTITADDRESSATTRIBUTES as AC WITH (NOLOCK)  
ON AC.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID  
INNER JOIN RECORDS AS R WITH (NOLOCK)  
ON R.ID=AC.ConstitID  
where A.RECORDTYPE = 12  
AND R.IS_CONSTITUENT=-1  
AND AC.ATTRIBUTETYPESID= [ENTER ID]  
GROUP BY R.CONSTITUENT_ID  
HAVING COUNT(AC.ATTRIBUTETYPESID)>1;
```

## **SUM DUPLICATE CONSTITADDRESSATTRIBUTES**

/\* DUPLICATE CONSTITADDRESSATTRIBUTES SUM\*/

SELECT SUM(COUNTS)

FROM

(SELECT COUNT(\*) AS [COUNTS]

FROM AttributeTypes as A WITH (NOLOCK)

INNER JOIN CONSTITADDRESSATTRIBUTES as AC WITH (NOLOCK)

ON AC.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID

INNER JOIN RECORDS AS R WITH (NOLOCK)

ON R.ID=AC.ConstitID

where A.RECORDTYPE = 12

AND R.IS\_CONSTITUENT=-1

AND AC.ATTRIBUTETYPESID= [ENTER ID]

GROUP BY R.CONSTITUENT\_ID

HAVING COUNT(AC.ATTRIBUTETYPESID)>1) AS W;



## **SUM DUPLICATE CONSTITUENTATTRIBUTES**

**/\* SUM DUPLICATE CONSTITUENTATTRIBUTES \*/**

**SELECT SUM([COUNTS]) AS [COUNTS]**

**FROM**

**(SELECT COUNT(\*) AS [COUNTS]**

**FROM AttributeTypes as A WITH (NOLOCK)**

**INNER JOIN CONSTITUENTATTRIBUTES as AC WITH (NOLOCK)**

**ON AC.ATTRIBUTETYPESID=A.ATTRIBUTETYPESID**

**INNER JOIN RECORDS AS R WITH (NOLOCK)**

**ON R.ID=AC.PARENTID**

**where A.RECORDTYPE = 1**

**AND R.IS\_CONSTITUENT=-1**

**AND AC.ATTRIBUTETYPESID= [ENTER ID]**

**GROUP BY R.CONSTITUENT\_ID**

**HAVING COUNT(AC.ATTRIBUTETYPESID)>1) AS W;**

## **SQL PUNCTUATION COUNTS IN LAST NAME**

**/\* SQL PUNCTUATION COUNTS IN LAST NAME \*/**

**SELECT R.LAST\_NAME**

**FROM RECORDS AS R WITH (NOLOCK)**

**WHERE**

**R.IS\_CONSTITUENT=-1**

**AND R.INACTIVE=0**

**AND R.LAST\_NAME NOT LIKE '%[a-Z]%'**

**AND R.LAST\_NAME NOT LIKE '%[0-9]%' ;**

## **SQL LAST NAMES WITH NUMBERS**

```
/* SQL LAST NAMES WITH NUMBERS */  
SELECT COUNT(R.LAST_NAME)  
FROM RECORDS AS R WITH (NOLOCK)  
WHERE R.IS_CONSTITUENT=-1  
AND R.INACTIVE=0  
AND R.LAST_NAME LIKE '%[0-9]%'  
AND R.LAST_NAME NOT LIKE '%@%';
```

## **SQL FIRST NAMES WITH NUMBERS**

```
/* SQL FIRST NAMES WITH NUMBERS */  
SELECT COUNT(R.FIRST_NAME)  
FROM RECORDS AS R WITH (NOLOCK)  
WHERE R.IS_CONSTITUENT=-1  
AND R.INACTIVE=0  
AND R.FIRST_NAME LIKE '%[0-9]%'  
AND R.FIRST_NAME NOT LIKE '%@%';
```

## **SQL PRIMARY\_SALUTATION WITH NUMBERS**

```
/* SQL PRIMARY_SALUTATION WITH NUMBERS */
```

```
SELECT COUNT(R.PRIMARY_SALUTATION) AS  
[PRIMARY_SALUTATION_COUNT]
```

```
FROM RECORDS AS R WITH (NOLOCK)
```

```
WHERE R.IS_CONSTITUENT=-1
```

```
AND R.INACTIVE=0
```

```
AND R.PRIMARY_SALUTATION LIKE '%[0-9]%'
```

```
AND R.PRIMARY_SALUTATION NOT LIKE '%@%';
```

## SQL INDUSTRY LIST

/\* SQL INDUSTRY LIST \*/

```
SELECT A.ATTRIBUTEID, A.[DESCRIPTION] AS "CATEGORY",
A.ACTIVE, 'CONSTITUENTATTRIBUTES' AS [ATTRIBUTE],
CASE A.TYPEOFDATA
WHEN 6 THEN 'TABLEENTRIES'
ELSE 'CONSTITUENTATTRIBUTES'
END AS "TABLENAME",
CASE A.[REQUIRED]
WHEN 0 THEN 'FALSE'
WHEN -1 THEN 'TRUE'
ELSE 'ERROR'
END AS "REQUIRED",
CASE A.[MUSTBEUNIQUE]
WHEN 0 THEN 'FALSE'
WHEN -1 THEN 'TRUE'
ELSE 'ERROR'
END AS "MUSTBEUNIQUE",
T.LONGDESCRIPTION,
T.TABLEENTRIESID
FROM AttributeTypes as A WITH (NOLOCK)
INNER JOIN TABLEENTRIES T
ON A.CODETABLESID=T.CODETABLESID
where A.RECORDTYPE = 1
AND A.ATTRIBUTEID= [ENTER ID]
GROUP BY A.ATTRIBUTEID, A.[DESCRIPTION], A.ACTIVE,
A.TYPEOFDATA, A.[REQUIRED],A.[MUSTBEUNIQUE], T.LONGDESCRIPTION,
T.TABLEENTRIESID;
```

## **SQL ACTION NOTE TYPES**

```
SELECT DISTINCT ActionNotepad.NoteTypeId,  
TABLEENTRIES.LONGDESCRIPTION AS [ACTION NOTE TYPE]  
FROM ACTIONNOTEPAD, TABLEENTRIES  
WHERE  
ACTIONNOTEPAD.NoteTypeId=TABLEENTRIES.TABLEENTRIESID;
```

## **SQL CODE ACTION TYPES**

```
SELECT AT.[DESCRIPTION] AS [CATEGORY], T.LONGDESCRIPTION
AS [ACTIONTYPE], MAX(YEAR(A.DateChanged)) AS [DateChanged],
COUNT(T.TABLEENTRIESID) AS [COUNTACTIONTYPES]
FROM TABLEENTRIES AS T WITH (NOLOCK)
INNER JOIN ACTIONS AS A WITH (NOLOCK)
ON A.[TYPE]=T.TABLEENTRIESID
INNER JOIN ACTIONATTRIBUTES AS AC WITH (NOLOCK)
ON AC.PARENTID=A.ID
INNER JOIN AttributeTypes AS AT WITH (NOLOCK)
ON AT.ATTRIBUTEID=AC.ATTRIBUTEID
WHERE T.CODETABLESID= [ENTER ID]
GROUP BY AT.[DESCRIPTION], T.LONGDESCRIPTION;
```



# **SQL SERVER OLTP DATA WAREHOUSE QUERIES – RAISER’S EDGE**

## **RAISER'S EDGE Data Warehouse SQL Stored Procedures Definitions**

```
select P.object_id, so.name, m.[definition], so.type_desc
from sys.procedures p
inner join sys.all_sql_modules m
on m.object_id = p.object_id
inner join sys.objects so
on so.object_id=p.object_id;
```