

SMS Messaging (SMS)

SMS Message Stats

Download SQL file: [SMSMessageStatsSnow.sql](#)

The example below queries the SMSMessageFact table for the number of unique SMS users

SMS Unique User Count
<pre>-- SMS Message Fact table -- How many users are using SMS? SELECT COUNT(distinct T.userId) FROM (SELECT smf_sendingUserId AS userId FROM SMSMessageFact WHERE smf_messageDirection = 'OUTBOUND' AND smf_sendingUserId is not NULL UNION SELECT smf_receivingUserId AS userId FROM SMSMessageFact WHERE smf_messageDirection = 'INBOUND' AND smf_receivingUserId is not NULL) AS T</pre>

The example below queries the SMSMessageFact table for a list of unique SMS users

SMS Unique User List
<pre>-- List of users that are using SMS SELECT distinct T.userId FROM (SELECT smf_sendingUserId AS userId FROM SMSMessageFact WHERE smf_messageDirection = 'OUTBOUND' AND smf_sendingUserId is not NULL UNION SELECT smf_receivingUserId AS userId FROM SMSMessageFact WHERE smf_messageDirection = 'INBOUND' AND smf_receivingUserId is not NULL) AS T ORDER BY userId</pre>

The example below queries the SMSMessageFact table for SMS detailed usage

SMS Detailed Usage

```
-- Usage report
-- date sent, direction, SMS number, SMS UserId, other party
SELECT
    TO_VARCHAR(smf_originationTimestamp::timestamp, 'yyyy-mm-dd hh:mi:ss') AS "Date Sent", smf_messageDirection
AS "Direction",
    snd_number AS "SMS Number",
    CASE WHEN smf_messageDirection = 'INBOUND' THEN smf_receivingUserId ELSE smf_sendingUserId END AS "SMS
UserId",
    CASE WHEN smf_messageDirection = 'INBOUND' THEN smf_fromContactNumber ELSE smf_toContactNumber END AS
"Other Party"
FROM
    SMSMessageFact LEFT OUTER JOIN SMSNumberDimension ON CASE WHEN smf_messageDirection = 'INBOUND' THEN
smf_toNumberFk ELSE smf_fromNumberFk END = snd_id
WHERE
    smf_originationDateLocal >= 20210101 AND smf_originationDateLocal < 20220101
ORDER BY smf_originationTimestamp;
```

SMS Contact Stats

Download SQL file: [SMSContactsStatsSnow.sql](#)

The example below queries the SMSMessageFact for a count of unique SMS Contacts

SMS Unique Contacts

```
-- How many unique contacts (external phone numbers) have been texted?
SELECT COUNT(DISTINCT contact) AS "Contact Count"
FROM (SELECT
    smf_toNumber AS contact
FROM
    SMSMessageFact
WHERE
    smf_messageDirection = 'OUTBOUND' AND smf_sendingUserId is not NULL
UNION
SELECT
    smf_fromNumber AS contact
FROM
    SMSMessageFact
WHERE
    smf_messageDirection = 'INBOUND' AND smf_receivingUserId is not NULL) AS T;
```

SMS Unique Contact List

```
-- List of unique contacts that have been texted
SELECT DISTINCT contact AS Contacts
FROM (SELECT
    smf_toNumber AS contact
FROM
    SMSMessageFact
WHERE
    smf_messageDirection = 'OUTBOUND' AND smf_sendingUserId is not NULL
UNION
SELECT
    smf_fromNumber AS contact
FROM
    SMSMessageFact
WHERE
    smf_messageDirection = 'INBOUND' AND smf_receivingUserId is not NULL) AS T
ORDER BY contact;
```

SMS Stop Request Stats

Download SQL file: [SMSStopRequestStatsSnow.sql](#)

The example below queries the SMSStopRequestFact table for a count of SMS Stop Requests by SMS number

SMS Stop Requests by SMS Number

```
-- Stop request counts by SMS number
SELECT
    ssrf_smsNumber AS "SMS Number", COUNT(*) "Total Request Count",
    SUM( CASE WHEN ssrf_restartTimestamp IS NULL OR ssrf_restartTimestamp > GETDATE() OR
ssrf_restartTimestamp >= ssrf_stopTimestamp THEN 1 ELSE 0 END ) AS "Active Request Count",
    SUM( CASE WHEN ssrf_restartTimestamp IS NULL OR ssrf_restartTimestamp > GETDATE() OR
ssrf_restartTimestamp >= ssrf_stopTimestamp THEN 0 ELSE 1 END ) AS "Inactive Request Count"
FROM
    SMSStopRequestFact
WHERE
    ssrf_stopDateLocal >= 20210101 AND ssrf_stopDateLocal < 20220101
GROUP BY
    ssrf_smsNumber
ORDER BY ssrf_smsNumber;
```

The example below queries the SMSStopRequestFact table for detailed SMS Stop Requests

SMS Stop Request Detail

```
-- Usage report
-- SMS number, SMS UserId, requesting number, stop request date, start request date, is an active request
SELECT
    ssrf_smsNumber AS "SMS Number",
    ssrf_userId AS "SMS UserId",
    ssrf_fromNumber AS "Requesting Number",
    TO_VARCHAR(ssrf_stopTimestamp::timestamp, 'yyyy-mm-dd hh:mi:ss') AS "Stop Request Date",
    CASE WHEN ssrf_restartTimestamp IS NOT NULL THEN TO_VARCHAR(ssrf_restartTimestamp::timestamp, 'yyyy-mm-
dd hh:mi:ss') ELSE 'N/A' END AS "Start Request Date",
    CASE WHEN (ssrf_restartTimestamp IS NULL) OR ssrf_restartTimestamp > GETDATE() OR ssrf_restartTimestamp
>= ssrf_stopTimestamp THEN 'true' ELSE 'false' END AS "Active Request"
FROM
    SMSStopRequestFact
WHERE
    ssrf_stopDateLocal >= 20210101 AND ssrf_stopDateLocal < 20220101
ORDER BY ssrf_stopDateLocal;
```