SMTDATA
IT Enterprise Insights

Using the ITBI Data Lake

itbi
SMTDATA

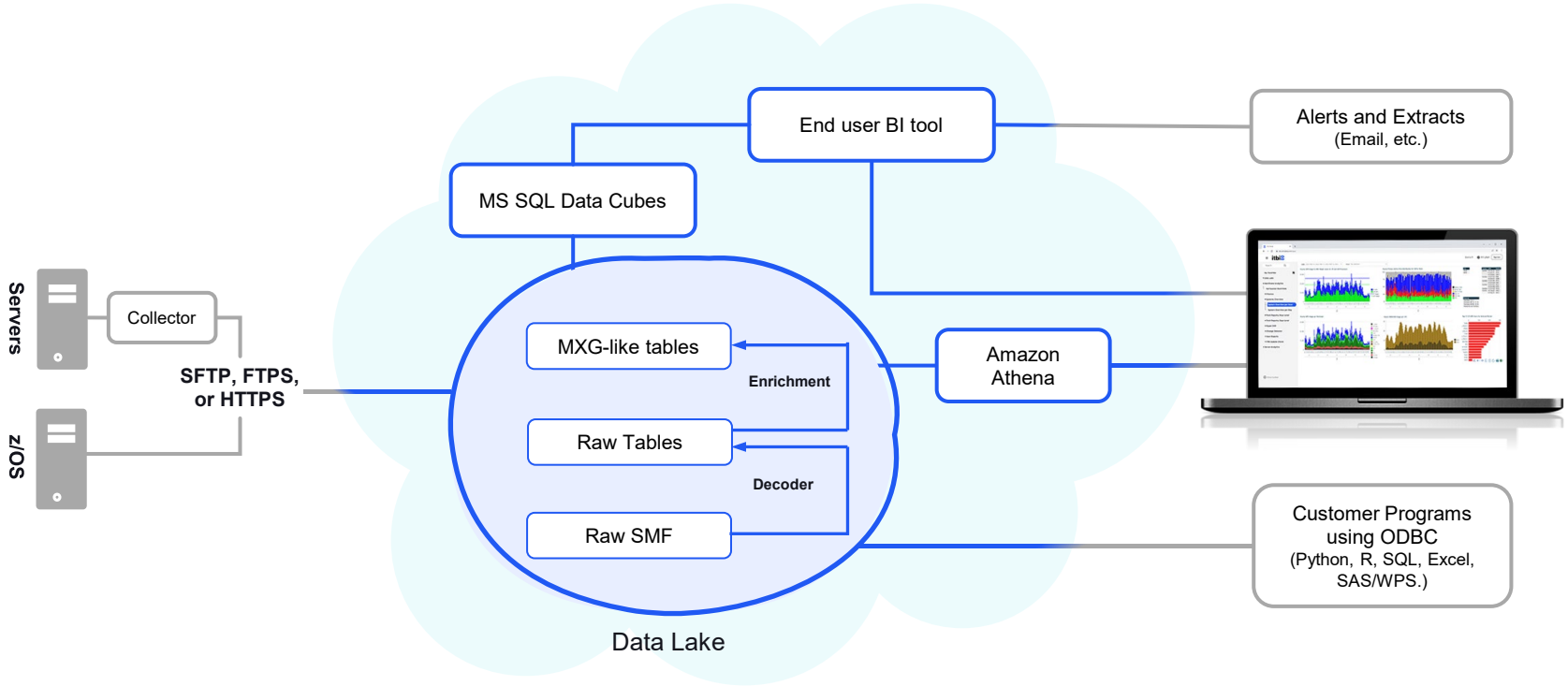
00 Agenda

SMTDATA
IT Enterprise Insights

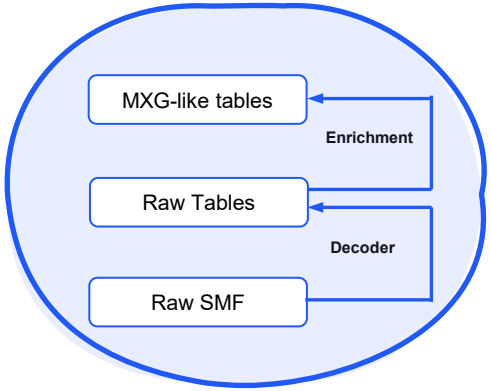
Topics

1. What is the Data Lake?
2. Use Cases
3. Accessing the Data Lake
4. Examples

01 What is the Data Lake?



SMTDATA
IT Enterprise Insights

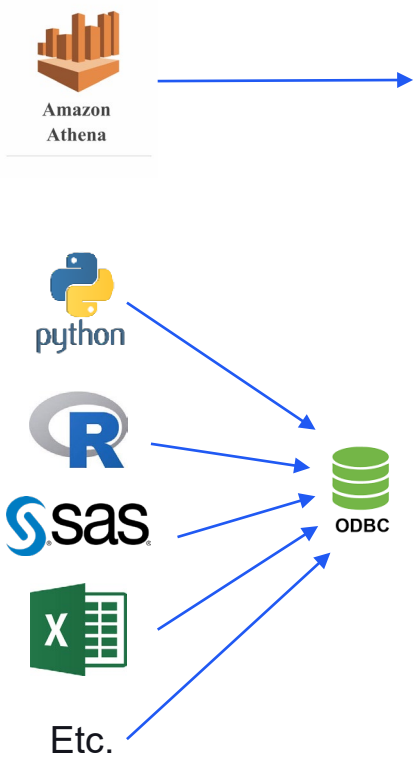


What is the Data Lake?

Data in the Data Lake:

- MXG-like Tables:
    - Only selected fields from the SMF types we support
    - Same naming and structure as MXG
- Raw SMF in Tables:
    - Essentially all fields from all SMF types we support
    - Mostly decoded
    - Same naming and structure as standard IBM SMF names
- Raw SMF files:
    - As sent from the customer

SMTDATA
IT Enterprise Insights



Ways of working with the Data Lake

- Direct access from the ITBI portal using Amazon Athena
  - Prototyping
  - Ad hoc queries
  - Hosted by SMT Data
- Programs using a remote ODBC connection
  - Programming
  - Re-use of existing SAS/MXG
  - Any language that supports remote ODBC
  - Hosted at by the customer

SMTDATA
IT Enterprise Insights

General Use Cases

- Reporting on recent data – within minutes of data being received by SMT Data
- Reporting on fields that are not supported in the cubes
- Reporting on details that are aggregated away in the cubes
- Reporting on ‘event’ based data
- Complex logic or calculations that is not easily implemented in the BI tool
- Graphics or formatting not supported in the BI tool
- Integration with customer or third-party systems

SMTDATA
IT Enterprise Insights

smf_smf03001	Partitioned
recordid	string
recordhash	bigint
smf30flg	int
smf30tme	int
smf30dte	timestamp
smf30sid	string
smf30wid	string
smf30typ	int
smf30rs1	string
smf30rvn	string
smf30pnm	string
smf30osl	string
smf30syn	string

Use Cases – Raw Tables

- Reporting on fields that are not supported in the cubes
- Reporting on details that are aggregated away in the cubes
- Reporting on ‘event’ based data
- Requires a good understanding of SMF

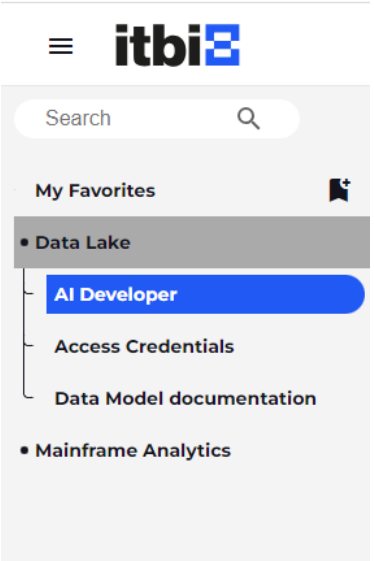
SMTDATA
IT Enterprise Insights

v_smtmxg_type30_1	
batch	int
system	string
smftime	timestamp
jinttime	timestamp
origist	timestamp
readtime	timestamp
smf30iss	timestamp
smf30iet	timestamp
account1	string
account2	string
account3	string
account4	string
account5	string

Use Cases – MXG-like Tables

- Reuse of existing SAS programs that work on MXG tables, but note, only selected fields are supported
- Taking advantage of existing skills with SAS/MXG
- Requires a good understanding of MXG

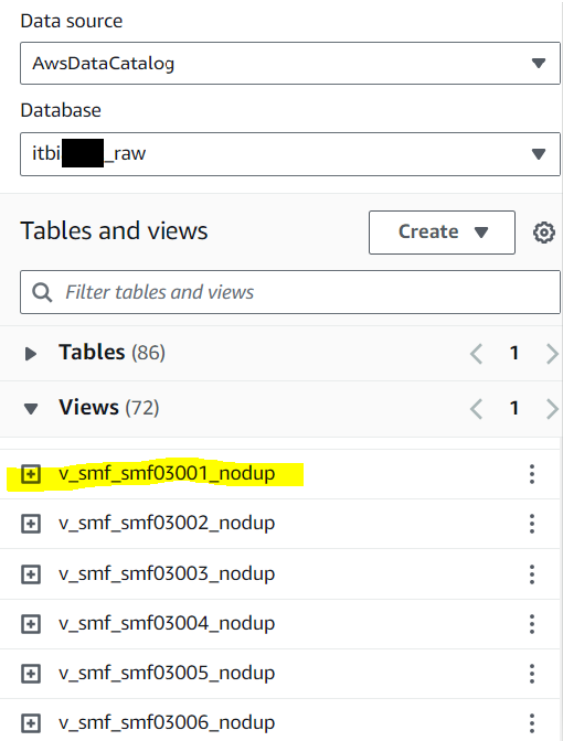
SMTDATA
IT Enterprise Insights



Via SQL from Athena

- Log on to the Portal
- Choose Data Lake
- Choose AI Developer

SMTDATA
IT Enterprise Insights

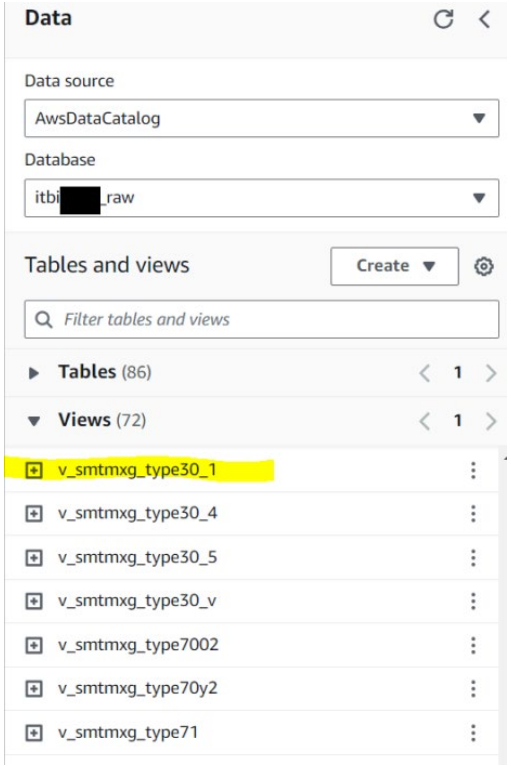


Accessing the Raw Tables

- Choose the Raw Database
- Choose a ‘nodup’ view based on an SMF tables
    - View names start with v_smf_*
    - The main views are named v_smf_smfyyyzz_nodup
    - Where yyy is the SMF number
    - And zz is the subtype
    - So v_smf_smf03001_nodup is SMF30 subtype 1

(Note the difference between the ‘nodup’ view and the underlying raw table is that the ‘nodup’ view removes any duplicated records)

SMTDATA
IT Enterprise Insights



Accessing the MXG-like Tables

- Choose the Raw Database
- Choose the MXG-like views
    - View names start with v_smtmxg_*
    - In general, the views follow the mxg naming conventions

SMTDATA
IT Enterprise Insights

Query 1 | Query 2 | Query 3

```
1 select smf70tme, smf70sid, smf70lac
2 from
3 v_smf_smf07001_nodup
4 where batch = 20240402
5 order by smf70tme, smf70sid
6 limit 100
```

SQL Ln 1, Col 36

Run again | Explain | Cancel

Selecting data from the SMF 70.1 Raw table

This example shows:

A select statement that chooses the following columns from the SMF70 subtype 1 table:

- SMF70tme (100^{ths} of a second since midnight)
- SMF70sid (LPAR system name)
- SMF70lac (4HRA MSU)

For the date 2024-04-02 (batch number is the date)

Sorted by time and system

Limited to 100 rows

SMTDATA
IT Enterprise Insights

Working with dates and times in the raw tables

```sql
select smf70dte, smf70tme
, DATE_ADD('MILLISECOND', (SMF70TME * 10), SMF70DTE)
as SMFDateTime
from v_smf_smf07001_nodup
where batch = 20240402
limit 100
```

SQL Ln 4, Col 6

Run again | Explain | Cancel | Clear | Create

Query results | Query stats

Completed Time in queue: 57 ms Run time: 1.696 sec Data s

Results (100+) Copy Dov

Search rows

#	smf70dte	smf70tme	SMFDateTime
1	2024-04-02 00:00:00.000	8520004	2024-04-02 23:40:00.040
2	2024-04-02 00:00:00.000	4860008	2024-04-02 13:30:00.080

The SMF dates in the raw tables have generally been converted to a datetime datatype (rather than the original packed decimal format).

The timestamps however are often still in unusual units such as hundredths of a second since midnight.

The query on the left shows how to merge those two fields into a single datetime column

SMTDATA
IT Enterprise Insights

Creating a view in the datalake

```
CREATE OR REPLACE VIEW "v_demo_smf70" AS
(select
    DATE_ADD('MILLISECOND', (SMF70TME * 10), SMF70DTE) SMFTIME,
    DATE_ADD('MILLISECOND', (SMF70IST * 1000), SMF70DAT) STARTIME,
    SMF70XNM SYSPLEX,
    SMF70SID System,
    smf70lac R4HAMSU
from itbi[redacted]_raw.v_smf_smf07001_nodup)
```

It can be useful to create you own views on top of the existing ones in the datalake, for example if you want to:

- Rename the fields to something more useful
- Do changes the formatting or calculations in the fields
- Join multiple views

Views are created in the ‘ITBIxxx_user’ database and are shared across the organization.

The example on the left shows the creation of a view on top of the SMF70 raw data.

04 Example – CPU per JOB from SMF30 via the MXG view

SMTDATA
IT Enterprise Insights

```
Select
   Year(DATE_ADD('Minute', 2, (SMF30ISS at time zone 'Europe/Copenhagen'))) as Year,-- In local time
   Month (DATE_ADD('Minute', 2, (SMF30ISS at time zone 'Europe/Copenhagen’))) as Month,
   Day (DATE_ADD('Minute', 2, (SMF30ISS at time zone 'Europe/Copenhagen'))) as Day,
   Hour (DATE_ADD('Minute', 2, (SMF30ISS at time zone 'Europe/Copenhagen'))) as Hour,
   I.system,
   I.job,
   I.racfuser as UserID,
   sum(CPISRBTM + CPITCBTM + CPUHPTTM + CPUIIPTM + CPURCTTM + CPUSRBTM + CPUTCBTM)
AS CPUTM
FROM "v_smtmxg_type30_v" I
where
   Year(DATE_ADD('Minute', 2, (SMF30ISS at time zone 'Europe/Copenhagen'))) = 2023
   And Month(DATE_ADD('Minute', 2, (SMF30ISS at time zone 'Europe/Copenhagen'))) = 12
   And Day(DATE_ADD('Minute', 2, (SMF30ISS at time zone 'Europe/Copenhagen'))) = 10
   And Batch between 20231209 and 20231211 --  (use +/- on day)
   Group by 1,2,3,4,5,6,7
   Limit 100
```

04 Example – CPU, Transactions, Response time from SMF 110 via the MXG view

SMTDATA
IT Enterprise Insights

```
With CICS as (
        SELECT
                Year(stop at time zone 'Europe/Copenhagen') as Year,
                Month(stop at time zone 'Europe/Copenhagen') as Month,
                Day(stop at time zone 'Europe/Copenhagen') as Day,
                Hour(stop at time zone 'Europe/Copenhagen') as Hour,
                tranname as Transaction_Name,
                system as Sysid,
                iresptm as Respons_Time,
                CASE
                        WHEN RTYPE = 'T' THEN 1
                END AS Transaction_Count,
                CPUTM - TASZIPTM AS CPU_Time,
                batch
        FROM “v_smtmxg_cicstran")

Select year, month,       day, hour, Transaction_Name, Sysid,
        sum(Transaction_Count) as Transaction_Count,
        avg(Respons_Time) as Response_time,
        Sum(CPU_Time) as CPU_Time
From CICS
Where batch = 20240410 -- for performance
   and year = 2024 and month = 4 and day = 10 and hour = 10
group by 1,2,3,4,5,6
order by 7 desc
Limit 100
```

04 Example – CPU, Transactions, Response time from IMS 56FA via the MXG view

SMTDATA
IT Enterprise Insights

```
Select
        Year(TPENDUCH at time zone 'Europe/Copenhagen') as Year, -- in local time
        Month(TPENDUCH at time zone 'Europe/Copenhagen') as Month,
        Day(TPENDUCH at time zone 'Europe/Copenhagen') as Day,
        Hour(TPENDUCH at time zone 'Europe/Copenhagen') as Hour,
        CASE WHEN ProgType = 'B' THEN RegionId ELSE Transact END AS Trans_Name,
        regionid,
        sum(nmsgproc) as Transaction_Count, -- nmsgrpoc = 1 for transactions
        sum(imscputm) AS CPUTM,
        avg(respnstm) As Response_Time
FROM "v_smtmxg_ims56fa"
where
        Batch between 20231130 and 20231202 -- use +/- one day
        And Year(TPENDUCH at time zone 'Europe/Copenhagen') = 2023
        And Month(TPENDUCH at time zone 'Europe/Copenhagen') = 12
        And Day(TPENDUCH at time zone 'Europe/Copenhagen') = 1
group by 1,2,3,4,5,6
order by 7 desc
Limit 100
```

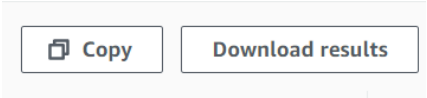
04 Example – CPU time from DDF using the MXG view

SMTDATA
IT Enterprise Insights

```
Select
        Year(QWHSSTCK at time zone 'Europe/Copenhagen') as Year, -- In local time
        Month(QWHSSTCK at time zone 'Europe/Copenhagen') as Month,
        Day(QWHSSTCK at time zone 'Europe/Copenhagen') as Day,
        Hour(QWHSSTCK at time zone 'Europe/Copenhagen') as Hour,
        qwhsssid as System,
        QWHCAID as user_ID,
        QWHSSSID as DB2_Subsystem,
        QWHCEUTX as End_User_Transaction,
        sum(DB2tcbtm) as TCB_time
FROM "v_smtmxg_db2acct"
where Year(QWHSSTCK at time zone 'Europe/Copenhagen') = 2023
        And Month(QWHSSTCK at time zone 'Europe/Copenhagen') = 12
        And Day(QWHSSTCK at time zone 'Europe/Copenhagen') = 1
        And Batch between 20231130 and 20231202 --  (use +/- on day)
        And qwhcatyp = 8 -- 'DDF'
group by 1,2,3,4,5,6,7,8
order by 9 desc
Limit 100
```

04 Tips, tricks and documentation

SMTDATA
IT Enterprise Insights



- You can cut and paste the results or download them as csv using Athena.
- Always include a where clause specifying the date (or range of dates) of interest using ‘batch’. E.g. where batch > 20221001 will give you all days after October 1, 2022. If you don’t specify a batch, then Athena will scan all of the data in the data lake.
- When developing a query or just exploring the data it is a good idea to limit the number of rows returned using the ‘limit’ statement. Then you can remove the limit when you are ready to ask for a full set of data.
- We are developing additional documentation. Have a look here to see the version under development: https://dev-api.smtdata.com/data-model-documentation/swagger/index.html
- You can also find lots of documentation of SMF on the IBM website: https://www.ibm.com/docs/en/zos/2.4.0?topic=smf-records