

FEEDS API Logging Data : Integration with Audit Log Data to create a View

Overview : This document outlines the SQL queries used to create views and a union for FEEDS API logging and audit data. These views consolidate data from multiple sources, enabling comprehensive analysis and reporting.

This setup is for the QA schema `ww_dv_apis_qa_vm`:

Source Table: The production table `ww_dv_platform_log.feeds_api_logging` is used as the primary source for logging data.

View in QA: A view `ww_dv_apis_qa_vm.audit_api_feeds_vw` is created in the QA schema, which is derived from the production table `us_dv_audit_log_prod.api_logs`.

1. Table: `ww_dv_platform_log.feeds_api_logging`

Description : This query retrieves all columns from the `feeds_api_logging` table in the `ww_dv_platform_log` schema. It serves as the base dataset for FEEDS API logging.

- Provides a complete dataset for NRTI API logging. Acts as a source for the union view.

`feeds_api_logging` table

```
sql("""
    SELECT
        request_id,
        correlation_id,
        consumer_id,
        endpoint,
        request_ts,
        response_ts,
        response_code,
        response_msg,
        response_size,
        num_of_records,
        request_payload_size,
        duration,
        backend_request_ts,
        backend_response_ts,
        api_version,
        feed,
        feed_type,
        method,
        created_ts,
        updated_ts,
        environment,
        request_date
    FROM
        ww_dv_platform_log.feeds_api_logging
    """).show(10, false)
```

2. View: `audit_api_logs_feeds_view`

Description : This view extracts and transforms data from the `api_logs` table in the `us_dv_audit_log_prod` schema. It includes specific transformations and mappings for FEEDS-related endpoints.

- Extracts and transforms audit log data specific to FEEDS services.
- Maps endpoint names to standardized values for consistency.
- Filters data for production environment and valid request IDs.
- Correlation IDs are aliased as request IDs that we are fetching from the headers columns of `audit_api_logs`, to identify api-proxy requests.

transformed_audit_logs_api view

```
sql"""
CREATE OR REPLACE VIEW ww_dv_apis_qa_vm.audit_api_feeds_vw AS
SELECT
    DISTINCT REGEXP_EXTRACT(REPLACE(headers, ' ', ''), '(?<="wm_qos.correlation_id":")[^"]*'') AS request_id,
    CAST(NULL AS VARCHAR(255)) AS correlation_id,
    consumer_id,
    endpoint_path AS endpoint,
    date_format(from_unixtime(request_ts / 1000), '%Y-%m-%dT%H:%i:%s.%f000+00:00') AS request_ts,
    date_format(from_unixtime(response_ts / 1000), '%Y-%m-%dT%H:%i:%s.%f000+00:00') AS response_ts,
    response_code,
    CAST(NULL AS VARCHAR(255)) AS response_msg,
    CAST(NULL AS VARCHAR(255)) AS response_size,
    CAST(NULL AS VARCHAR(255)) AS num_of_records,
    request_size_bytes AS request_payload_size,
    (response_ts - request_ts) AS duration,
    CAST(NULL AS VARCHAR(255)) AS backend_request_ts,
    CAST(NULL AS VARCHAR(255)) AS backend_response_ts,
    api_version,
    CASE
        WHEN endpoint_path LIKE '%/bulkfeeds/%/%' THEN
            regexp_extract(endpoint_path, '/bulkfeeds/([^/]+)(?:/.*)?$', 1)
        ELSE NULL
    END AS feed_type,
    CASE
        WHEN endpoint_path LIKE '%/bulkfeeds/%/%' THEN
            regexp_extract(endpoint_path, '/bulkfeeds/[^/]+/([^.]+)(?:/.*)$', 1)
        ELSE NULL
    END AS feed,
    method,
    date_format(from_unixtime(created_ts / 1000), '%Y-%m-%dT%H:%i:%s.%f000+00:00') AS created_ts,
    date_format(from_unixtime(created_ts / 1000), '%Y-%m-%dT%H:%i:%s.%f000+00:00') AS updated_ts,
    'PRODUCTION' AS environment,
    date AS request_date
FROM
    us_dv_audit_log_prod.api_logs
WHERE
    service_name LIKE '%BULK-FEEDS%'
    AND LOWER(REGEXP_EXTRACT(REPLACE(headers, ' ', ''), '(?<="wm_svc.env":")[^"]*'')) = 'prod'
    AND REGEXP_EXTRACT(REPLACE(headers, ' ', ''), '(?<="wm_qos.correlation_id":")[^"]*'') IS NOT NULL;
""").show(2, false)
```

3. View: feeds_api_logging_vw

Description : This view combines data from feeds_api_logging and audit_api_feeds_vw using a UNION ALL operation. It ensures a consolidated dataset with all columns aligned.

- Combines logging data from both feeds_api_logging and audit_api_feeds_vw. Provides a unified view for comprehensive analysis and reporting.

final view of nrti_api_logging_vw

```
sql"""
CREATE OR REPLACE VIEW ww_dv_apis_qa_vm.feeds_api_logging_vw AS
SELECT
    CAST(request_id AS VARCHAR(255)) AS request_id,
    CAST(correlation_id AS VARCHAR(255)) AS correlation_id,
    CAST(consumer_id AS VARCHAR(255)) AS consumer_id,
    CAST(endpoint AS VARCHAR(255)) AS endpoint,
    CAST(date_format(request_ts, '%Y-%m-%dT%H:%i:%s') || '.' ||
        substring(date_format(request_ts, '%f'), 1, 6) || '+00:00' AS VARCHAR(255)) AS request_ts,
    CAST(date_format(response_ts, '%Y-%m-%dT%H:%i:%s') || '.' ||
        substring(date_format(response_ts, '%f'), 1, 6) || '+00:00' AS VARCHAR(255)) AS response_ts,
    CAST(response_code AS INT) AS response_code,
    CAST(response_msg AS VARCHAR(255)) AS response_msg,
    CAST(response_size AS VARCHAR(50)) AS response_size,
    CAST(num_of_records AS VARCHAR(50)) AS num_of_records,
    CAST(request_payload_size AS VARCHAR(50)) AS request_payload_size,
    CAST(duration AS VARCHAR(50)) AS duration,
    CAST(date_format(backend_request_ts, '%Y-%m-%dT%H:%i:%s') || '.' ||
        substring(date_format(backend_request_ts, '%f'), 1, 6) || '+00:00' AS VARCHAR(255)) AS
backend_request_ts,
    CAST(date_format(backend_response_ts, '%Y-%m-%dT%H:%i:%s') || '.' ||
        substring(date_format(backend_response_ts, '%f'), 1, 6) || '+00:00' AS VARCHAR(255)) AS
backend_response_ts,
    CAST(api_version AS VARCHAR(255)) AS api_version,
    CAST(feed AS VARCHAR(255)) AS feed,
    CAST(feed_type AS VARCHAR(255)) AS feed_type,
    CAST(method AS VARCHAR(255)) AS method,
    CAST(date_format(created_ts, '%Y-%m-%dT%H:%i:%s') || '.' ||
        substring(date_format(created_ts, '%f'), 1, 6) || '+00:00' AS VARCHAR(255)) AS created_ts,
    CAST(date_format(updated_ts, '%Y-%m-%dT%H:%i:%s') || '.' ||
        substring(date_format(updated_ts, '%f'), 1, 6) || '+00:00' AS VARCHAR(255)) AS updated_ts,
    CAST(environment AS VARCHAR(255)) AS environment,
    request_date
FROM
    ww_dv_platform_log.feeds_api_logging
UNION ALL
SELECT
    CAST(request_id AS VARCHAR(255)) AS request_id,
    CAST(correlation_id AS VARCHAR(255)) AS correlation_id,
    CAST(consumer_id AS VARCHAR(255)) AS consumer_id,
    CAST(endpoint AS VARCHAR(255)) AS endpoint,
    CAST(request_ts AS VARCHAR(255)) AS request_ts,
    CAST(response_ts AS VARCHAR(255)) AS response_ts,
    CAST(response_code AS INT) AS response_code,
    CAST(response_msg AS VARCHAR(255)) AS response_msg,
    CAST(response_size AS VARCHAR(50)) AS response_size,
    CAST(num_of_records AS VARCHAR(50)) AS num_of_records,
    CAST(request_payload_size AS VARCHAR(50)) AS request_payload_size,
    CAST(duration AS VARCHAR(50)) AS duration,
    CAST(backend_request_ts AS VARCHAR(255)) AS backend_request_ts,
    CAST(backend_response_ts AS VARCHAR(255)) AS backend_response_ts,
    CAST(api_version AS VARCHAR(255)) AS api_version,
    CAST(feed AS VARCHAR(255)) AS feed,
    CAST(feed_type AS VARCHAR(255)) AS feed_type,
    CAST(method AS VARCHAR(255)) AS method,
    CAST(created_ts AS VARCHAR(255)) AS created_ts,
    CAST(updated_ts AS VARCHAR(255)) AS updated_ts,
    CAST(environment AS VARCHAR(255)) AS environment,
    request_date
FROM
    ww_dv_apis_qa_vm.audit_api_feeds_vw
""").show(2, false)
```

VIEW DATA - feeds_api_logging_vw

feeds_api_logging_vw view

```
sql("""SELECT * from ww_dv_apis_qa_vm.feeds_api_logging_vw""").show(20, false)
```

KEY NOTES :

1. Feeds API Logging Query

- Direct query to access raw feed API call logs from `ww_dv_platform_log.feeds_api_logging`
- Displays complete logging schema with all available fields
- Used for initial data exploration and validation

2. audit_api_feeds_vw View

- Transforms general audit logs into feeds API logging format
- *Key transformations:*
 - Maps `source_request_id` to `request_id`
 - Extracts correlation_id from JSON headers using `regex_extract`
 - Converts Unix timestamps to ISO format with timezone
 - Extracts feed_type and feed using regex patterns from endpoint_path
 - Filters only production BULK-FEEDS service logs
 - Standardizes format to match the feeds API logging structure

3. feeds_api_logging_vw View

- Consolidated view combining data from both sources using UNION ALL
- Provides a unified interface for querying all bulk feed operations
- Maintains consistent column structure across different data sources

SANDBOX VIEW(QA - ENVIRONMENT)

1. View: audit_api_logs_feeds_sandbox_view

transformed_audit_logs_api view

```
sql"""
CREATE OR REPLACE VIEW ww_dv_apis_qa_vm.audit_api_feeds_vw AS
    SELECT
        source_request_id AS request_id,
        REGEXP_EXTRACT(REPLACE(headers, ' ', ''), '(?<="wm_qos.correlation_id":")[^"]*'') AS correlation_id,
        consumer_id,
        endpoint_path AS endpoint,
        date_format(from_unixtime(request_ts / 1000), '%Y-%m-%dT%H:%i:%s.%f000+00:00') AS request_ts,
        date_format(from_unixtime(response_ts / 1000), '%Y-%m-%dT%H:%i:%s.%f000+00:00') AS response_ts,
        response_code,
        CAST(NULL AS VARCHAR(255)) AS response_msg,
        CAST(NULL AS VARCHAR(255)) AS response_size,
        CAST(NULL AS VARCHAR(255)) AS num_of_records,
        request_size_bytes AS request_payload_size,
        (response_ts - request_ts) AS duration,
        CAST(NULL AS VARCHAR(255)) AS backend_request_ts,
        CAST(NULL AS VARCHAR(255)) AS backend_response_ts,
        api_version,
        CASE
            WHEN endpoint_path LIKE '%/bulkfeeds/%/%' THEN
                regexp_extract(endpoint_path, '/bulkfeeds/([^/]+)(?:/.*)?$', 1)
            ELSE NULL
        END AS feed_type,
        CASE
            WHEN endpoint_path LIKE '%/bulkfeeds/%/%' THEN
                regexp_extract(endpoint_path, '/bulkfeeds/[^/]+/([^.]+)(?:/|$)', 1)
            ELSE NULL
        END AS feed,
        method,
        date_format(from_unixtime(created_ts / 1000), '%Y-%m-%dT%H:%i:%s.%f000+00:00') AS created_ts,
        date_format(from_unixtime(created_ts / 1000), '%Y-%m-%dT%H:%i:%s.%f000+00:00') AS updated_ts,
        'SANDBOX' AS environment,
        date AS request_date
    FROM
        us_dv_audit_log_dev.api_logs
    WHERE
        service_name LIKE '%BULK-FEEDS%'
        AND LOWER(REGEXP_EXTRACT(REPLACE(headers, ' ', ''), '(?<="wm_svc.name":")[^"]*'')) LIKE '%sandbox%'
        AND REGEXP_EXTRACT(REPLACE(headers, ' ', ''), '(?<="wm_qos.correlation_id":")[^"]*'') IS NOT NULL;
    """).show(2, false)
```

2. View: feeds_api_logging_sandbox_vw

final view of nrti_api_logging_vw

```
sql"""
CREATE OR REPLACE VIEW ww_dv_apis_qa_vm.feeds_api_logging_sandbox_vw AS SELECT
    CAST(request_id AS VARCHAR(255)) AS request_id,
    CAST(correlation_id AS VARCHAR(255)) AS correlation_id,
    CAST(consumer_id AS VARCHAR(255)) AS consumer_id,
    CAST(endpoint AS VARCHAR(255)) AS endpoint,
    CAST(date_format(request_ts, '%Y-%m-%dT%H:%i:%s') || '.' ||
        substring(date_format(request_ts, '%f'), 1, 6) || '+00:00' AS VARCHAR(255)) AS request_ts,
    CAST(date_format(response_ts, '%Y-%m-%dT%H:%i:%s') || '.' ||
        substring(date_format(response_ts, '%f'), 1, 6) || '+00:00' AS VARCHAR(255)) AS response_ts,
    CAST(response_code AS INT) AS response_code,
    CAST(response_msg AS VARCHAR(255)) AS response_msg,
    CAST(response_size AS VARCHAR(50)) AS response_size,
    CAST(num_of_records AS VARCHAR(50)) AS num_of_records,
    CAST(request_payload_size AS VARCHAR(50)) AS request_payload_size,
    CAST(duration AS VARCHAR(50)) AS duration,
    CAST(date_format(backend_request_ts, '%Y-%m-%dT%H:%i:%s') || '.' ||
        substring(date_format(backend_request_ts, '%f'), 1, 6) || '+00:00' AS VARCHAR(255)) AS
backend_request_ts,
    CAST(date_format(backend_response_ts, '%Y-%m-%dT%H:%i:%s') || '.' ||
        substring(date_format(backend_response_ts, '%f'), 1, 6) || '+00:00' AS VARCHAR(255)) AS
backend_response_ts,
    CAST(api_version AS VARCHAR(255)) AS api_version,
    CAST(feed AS VARCHAR(255)) AS feed,
    CAST(feed_type AS VARCHAR(255)) AS feed_type,
    CAST(method AS VARCHAR(255)) AS method,
    CAST(date_format(created_ts, '%Y-%m-%dT%H:%i:%s') || '.' ||
        substring(date_format(created_ts, '%f'), 1, 6) || '+00:00' AS VARCHAR(255)) AS created_ts,
    CAST(date_format(updated_ts, '%Y-%m-%dT%H:%i:%s') || '.' ||
        substring(date_format(updated_ts, '%f'), 1, 6) || '+00:00' AS VARCHAR(255)) AS updated_ts,
    CAST(environment AS VARCHAR(255)) AS environment,
    request_date
FROM
    ww_dv_platform_log.feeds_api_logging
Where environment = 'SANDBOX'
UNION ALL
SELECT
    CAST(request_id AS VARCHAR(255)) AS request_id,
    CAST(correlation_id AS VARCHAR(255)) AS correlation_id,
    CAST(consumer_id AS VARCHAR(255)) AS consumer_id,
    CAST(endpoint AS VARCHAR(255)) AS endpoint,
    CAST(request_ts AS VARCHAR(255)) AS request_ts,
    CAST(response_ts AS VARCHAR(255)) AS response_ts,
    CAST(response_code AS INT) AS response_code,
    CAST(response_msg AS VARCHAR(255)) AS response_msg,
    CAST(response_size AS VARCHAR(50)) AS response_size,
    CAST(num_of_records AS VARCHAR(50)) AS num_of_records,
    CAST(request_payload_size AS VARCHAR(50)) AS request_payload_size,
    CAST(duration AS VARCHAR(50)) AS duration,
    CAST(backend_request_ts AS VARCHAR(255)) AS backend_request_ts,
    CAST(backend_response_ts AS VARCHAR(255)) AS backend_response_ts,
    CAST(api_version AS VARCHAR(255)) AS api_version,
    CAST(feed AS VARCHAR(255)) AS feed,
    CAST(feed_type AS VARCHAR(255)) AS feed_type,
    CAST(method AS VARCHAR(255)) AS method,
    CAST(created_ts AS VARCHAR(255)) AS created_ts,
    CAST(updated_ts AS VARCHAR(255)) AS updated_ts,
    CAST(environment AS VARCHAR(255)) AS environment,
    request_date
FROM
    ww_dv_apis_qa_vm.audit_api_feeds_sandbox_vw
""").show(2, false)
```