

NRTI 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 NRTI 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.nrti_api_logging` is used as the primary source for logging data.

View in QA: A view `ww_dv_apis_qa_vm.nrti_logging_api_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.nrti_api_logging`

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

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

`nrti_api_logging` table

```
SELECT
    request_id,
    correlation_id,
    consumer_id,
    gtin,
    store,
    endpoint,
    endpoint_name,
    request_ts,
    response_ts,
    response_code,
    response_msg,
    response_size,
    request_payload_size,
    duration,
    backend_request_ts,
    backend_response_ts,
    api_version,
    method,
    created_ts,
    updated_ts,
    environment,
    request_date
FROM ww_dv_platform_log.nrti_api_logging;
```

2. View: `audit_api_logs_vw`

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 NRTI-related endpoints.

- Extracts and transforms audit log data specific to NRTI 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_logs_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,
    CAST(NULL AS VARCHAR(255)) AS gtin,
    CAST(NULL AS VARCHAR(255)) AS store,
    CASE
        WHEN endpoint_name = 'transactionHistory' THEN 'nrti_transactionHistory'
        WHEN endpoint_name = 'inventoryActions' THEN
            CASE REGEXP_EXTRACT(REPLACE(headers, ' ', ''), '(?<="wm_svc.name":")[^"]*')
                WHEN 'channelperformance-iac' THEN 'iac_inventoryActions'
                ELSE 'nrti_inventoryActions'
            END
        ELSE endpoint_name
    END AS endpoint_name,
    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 BIGINT) as response_size,
    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,
    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 '%NRT%'
    AND LOWER(REGEXP_EXTRACT(REPLACE(headers, ' ', ''), '(?<="wm_svc.env":")[^"]*')) = 'prod'
    AND REGEXP_EXTRACT(REPLACE(headers, ' ', ''), '(?<="wm_qos.correlation_id":")[^"]*') IS NOT NULL;
""").show()
```

3. View: nrti_logging_api_vw

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

- Combines logging data from both nrti_api_logging and audit_api_logs_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.nrti_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(gtin AS VARCHAR(255)) AS gtin,
    CAST(store AS VARCHAR(255)) AS store,
    CAST(endpoint AS VARCHAR(255)) AS endpoint,
    CAST(endpoint_name AS VARCHAR(255)) AS endpoint_name,
    CAST(date_format(CAST(request_ts AS TIMESTAMP), '%Y-%m-%dT%H:%i:%s') || '.' ||
        substring(date_format(CAST(request_ts AS TIMESTAMP), '%f'), 1, 6) || '+00:00' AS VARCHAR(255)) AS
request_ts,
    CAST(date_format(CAST(response_ts AS TIMESTAMP), '%Y-%m-%dT%H:%i:%s') || '.' ||
        substring(date_format(CAST(response_ts AS TIMESTAMP), '%f'), 1, 6) || '+00:00' AS VARCHAR(255)) AS
response_ts,
    response_code,
    CAST(response_msg AS VARCHAR(255)) AS response_msg,
    response_size,
    request_payload_size,
    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(method AS VARCHAR(255)) AS method,
    CAST(date_format(CAST(created_ts AS TIMESTAMP), '%Y-%m-%dT%H:%i:%s') || '.' ||
        substring(date_format(CAST(created_ts AS TIMESTAMP), '%f'), 1, 6) || '+00:00' AS VARCHAR(255)) AS
created_ts,
    CAST(date_format(CAST(updated_ts AS TIMESTAMP), '%Y-%m-%dT%H:%i:%s') || '.' ||
        substring(date_format(CAST(updated_ts AS TIMESTAMP), '%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.nrti_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(gtin AS VARCHAR(255)) AS gtin,
    CAST(store AS VARCHAR(255)) AS store,
    CAST(endpoint AS VARCHAR(255)) AS endpoint,
    CAST(endpoint_name AS VARCHAR(255)) AS endpoint_name,
    CAST(request_ts AS VARCHAR(255)) AS request_ts,
    CAST(response_ts AS VARCHAR(255)) AS response_ts,
    response_code,
    CAST(response_msg AS VARCHAR(255)) AS response_msg,
    response_size,
    request_payload_size,
    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(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_logs_vw
""").show(10, false)
```

VIEW DATA - [nrti_api_logging_vw](#)

nrti_api_logging_vw view

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

KEY NOTES

View Architecture

- **Intermediate View**: `ww_dv_apis_qa_vm.audit_api_logs_vw`
- Transforms audit logs data specifically for NRT APIs
- Maps to consistent column format matching the primary source
- **Final Union View**: `ww_dv_apis_qa_vm.nrti_logging_api_vw`
- Combines data from both sources with identical schema structure

Key Transformations

- **Regex Extraction**: Correlation ID extracted from headers using `get_json_object()`
- **Timestamp Formatting**: Unix timestamps converted to standardized format
- **Endpoint Normalization**:
- `transactionHistory` `nrti_transactionHistory`
- `inventoryActions` `nrti_inventoryActions` or `iac_inventoryActions` based on service name
- **Duration Calculation**: Computed as difference between response and request timestamps

Filters Applied

- Only includes records where:
- Service name contains "NRT"
- Environment is "prod"
- Correlation ID is not null

Schema Standardization

- All columns from primary source are preserved in the unified view
- NULL values used as placeholders for fields not present in secondary source

SANDBOX VIEW(QA - ENVIRONMENT)

1. View: audit_api_logs_sandbox_vw

transformed_audit_logs_api view

```
sql"""
CREATE OR REPLACE VIEW ww_dv_apis_qa_vm.audit_api_logs_sandbox_vw AS
SELECT
    REGEXP_EXTRACT(REPLACE(headers, ' ', ''), '(?<="wm_qos.correlation_id":")[^"]*'') AS request_id,
    CAST(NULL AS VARCHAR(255)) AS correlation_id,
    consumer_id,
    CAST(NULL AS VARCHAR(255)) AS gtin,
    CAST(NULL AS VARCHAR(255)) AS store,
    CASE
        WHEN endpoint_name = 'transactionHistory' THEN 'nrti_transactionHistory'
        WHEN endpoint_name = 'inventoryActions' THEN
            CASE REGEXP_EXTRACT(REPLACE(headers, ' ', ''), '(?<="wm_svc.name":")[^"]*'')
                WHEN 'channelperformance-iac' THEN 'iac_inventoryActions'
                ELSE 'nrti_inventoryActions'
            END
        ELSE endpoint_name
    END AS endpoint_name,
    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 BIGINT) as response_size,
    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,
    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 '%NRT%'
    AND LOWER(REGEXP_EXTRACT(REPLACE(headers, ' ', ''), '(?<="wm_svc.name":")[^"]*'')) LIKE '%sandbox%'
    AND REGEXP_EXTRACT(REPLACE(headers, ' ', ''), '(?<="wm_qos.correlation_id":")[^"]*'') IS NOT NULL;
""").show()
```

3. View: nrti_logging_api_sandbox_vw

final view of nrti_api_logging_vw

```
sql"""
CREATE OR REPLACE VIEW ww_dv_apis_qa_vm.nrti_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(gtin AS VARCHAR(255)) AS gtin,
    CAST(store AS VARCHAR(255)) AS store,
    CAST(endpoint AS VARCHAR(255)) AS endpoint,
    CAST(endpoint_name AS VARCHAR(255)) AS endpoint_name,
    CAST(date_format(CAST(request_ts AS TIMESTAMP), '%Y-%m-%dT%H:%i:%s') || '.' ||
        substring(date_format(CAST(request_ts AS TIMESTAMP), '%f'), 1, 6) || '+00:00' AS VARCHAR(255)) AS
request_ts,
    CAST(date_format(CAST(response_ts AS TIMESTAMP), '%Y-%m-%dT%H:%i:%s') || '.' ||
        substring(date_format(CAST(response_ts AS TIMESTAMP), '%f'), 1, 6) || '+00:00' AS VARCHAR(255)) AS
response_ts,
    response_code,
    CAST(response_msg AS VARCHAR(255)) AS response_msg,
    response_size,
    request_payload_size,
    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(method AS VARCHAR(255)) AS method,
    CAST(date_format(CAST(created_ts AS TIMESTAMP), '%Y-%m-%dT%H:%i:%s') || '.' ||
        substring(date_format(CAST(created_ts AS TIMESTAMP), '%f'), 1, 6) || '+00:00' AS VARCHAR(255)) AS
created_ts,
    CAST(date_format(CAST(updated_ts AS TIMESTAMP), '%Y-%m-%dT%H:%i:%s') || '.' ||
        substring(date_format(CAST(updated_ts AS TIMESTAMP), '%f'), 1, 6) || '+00:00' AS VARCHAR(255)) AS
updated_ts,
    CAST(environment AS VARCHAR(255)) AS environmexnt,
    request_date
FROM
    ww_dv_platform_log.nrti_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(gtin AS VARCHAR(255)) AS gtin,
    CAST(store AS VARCHAR(255)) AS store,
    CAST(endpoint AS VARCHAR(255)) AS endpoint,
    CAST(endpoint_name AS VARCHAR(255)) AS endpoint_name,
    CAST(request_ts AS VARCHAR(255)) AS request_ts,
    CAST(response_ts AS VARCHAR(255)) AS response_ts,
    response_code,
    CAST(response_msg AS VARCHAR(255)) AS response_msg,
    response_size,
    request_payload_size,
    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(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_logs_sandbox_vw
""").show(10, false)
```