# **Security Vulnerabilities Remediation**

## **1. Environment variable injection risk**

**Issue**: Environment variable placeholders might be vulnerable to injection.

"${ENV}" in 202206221918-mq-config-carriers-dml.sql (lines 7, 11)

"${ENV}" in 202208031010-predict-feed2.sql (line 14)

**Remediation**:

* Use prepared statements with variable binding rather than string concatenation

Implement a secure environment variable substitution mechanism:  
 sql  
*-- Instead of string template*

INSERT INTO public.mq\_config (..., s3\_dirty, ...)

VALUES (..., 'dsa-cdl-s3-borderforce-predict-dirty-${ENV}', ...);

*-- Use a parameter binding approach*

INSERT INTO public.mq\_config (..., s3\_dirty, ...)

* VALUES (..., ?, ...); *-- Bind parameter securely*
* Validate environment variable values against a specific pattern before use

## **2. Lack of statement parameterization**

**Issue**: SQL statements use direct value insertion rather than parameterization.

**Files affected**: All files with INSERT/UPDATE statements

**Remediation**:

Use prepared statements with parameter binding:  
 sql  
*-- Instead of direct insertion*

INSERT INTO public.cert\_config (id, description, ...)

VALUES (nextval('cert\_config\_id\_seq'), 'Test config', ...);

*-- Use parameterized queries (implementation depends on your database access library)*

INSERT INTO public.cert\_config (id, description, ...)

* VALUES (?, ?, ...); *-- Parameters bound securely*
* If using Liquibase, follow their parameter substitution patterns
* Consider using ORM frameworks that handle parameterization automatically

## **3. Insufficient data validation**

**Issue**: Tables lack constraints for data validation.

**Files affected**:

* 202206221737-cert-config-ddl.sql
* 202308241405-response-audit-ddl.sql

**Remediation**:

* Add appropriate CHECK constraints for data validation
* Specify length limits for VARCHAR columns
* Add NOT NULL constraints for required fields

Implement table and column constraints:  
 sql  
*-- Instead of unconstrained columns*

CREATE TABLE IF NOT EXISTS cert\_config (

id BIGINT NOT NULL PRIMARY KEY,

description VARCHAR,

filer\_carrier\_id VARCHAR NOT NULL,

...

);

*-- Add proper constraints*

CREATE TABLE IF NOT EXISTS cert\_config (

id BIGINT NOT NULL PRIMARY KEY,

description VARCHAR(255),

filer\_carrier\_id VARCHAR(50) NOT NULL CHECK (filer\_carrier\_id ~ '^[A-Za-z0-9]+$'),

...

* );
* Consider using domain types for specialized data formats

## **4. Exposed internal infrastructure details**

**Issue**: Queue names and directory structures are exposed in the scripts.

**Files affected**:

* 202206221918-mq-config-carriers-dml.sql
* 202207011045-update-data-feed-filename-pattern.sql
* 202305191505-data-feed-fix.sql

**Remediation**:

* Store infrastructure details in external configuration
* Use reference keys instead of actual infrastructure paths

Implement a configuration lookup pattern:  
 sql  
*-- Instead of hard-coded queue names*

INSERT INTO public.mq\_config (..., inbound\_mq\_queue, outbound\_mq\_queue, ...)

VALUES (..., 'PREDICT.IN.CARA', 'PREDICT.OUT.CARA', ...);

*-- Use configuration references*

INSERT INTO public.mq\_config (..., inbound\_mq\_queue, outbound\_mq\_queue, ...)

* VALUES (..., get\_config('QUEUE\_CARA\_IN'), get\_config('QUEUE\_CARA\_OUT'), ...);
* Consider encrypting sensitive infrastructure details

## **5. Limited column constraints**

Many columns in the created tables have no length restrictions or other constraints

**Issue**: VARCHAR columns lack length limits.

**Files affected**:

* 202206221737-cert-config-ddl.sql (lines 6-12)
* 202308241405-response-audit-ddl.sql (lines 9-18)

**Remediation**:

* Add appropriate length constraints to all VARCHAR columns

Implement data type constraints:  
 sql  
*-- Instead of unlimited VARCHAR*

CREATE TABLE IF NOT EXISTS cert\_config (

id BIGINT NOT NULL PRIMARY KEY,

description VARCHAR,

filer\_carrier\_id VARCHAR NOT NULL,

...

);

*-- Add length constraints*

CREATE TABLE IF NOT EXISTS cert\_config (

id BIGINT NOT NULL PRIMARY KEY,

description VARCHAR(255),

filer\_carrier\_id VARCHAR(50) NOT NULL,

...

* );
* Consider using specialized data types for specific formats (e.g., UUID)

## 