SCHEMA IMPORT TEMPLATE

Interface Control Document

Bridger DMA · bridger-dma.com

Version	1.0
Date	April 2026
Applies To	Bridger Schema Import (CSV format)
File Name	schema_import_template.csv

1. Purpose

This document describes the structure and column definitions of the Bridger schema import template — a CSV file used to load source or target database schemas into Bridger for mapping work. It is intended for database administrators and analysts responsible for extracting schema metadata and preparing it for import.

The template format is the canonical input format for Bridger's schema import function. All columns must be present in the CSV header, in the order defined in Section 2, even if some columns are empty for a given row.

2. File Format

Property	Value
Format	Comma-separated values (CSV)
Encoding	UTF-8
Header row	Required — first row must contain column names exactly as defined below
Row delimiter	CRLF or LF
Field delimiter	Comma (,)
Text qualifier	Double quote (") — required when field contains comma or newline
NULL values	Empty field (no quotes, no space)
Column order	Must match the order defined in Section 2.1

2.1 Row Types

The import file contains two types of rows, distinguished by the COLUMN_ID value:

Row Type	COLUMN_ID	Purpose
Column row	> 0	Describes a single column within a table. One row per column.
Table comment row	= 0	Carries the table-level description in the DESCRIPTION field. COLUMN_NAME, DATA_TYPE, DATA_LENGTH, DATA_PRECISION, DATA_SCALE, DATA_DEFAULT, and NULLABLE are left empty for these rows.

Table comment rows are optional. If a table has no comment defined in the data dictionary, omit the row rather than including a row with an empty DESCRIPTION.

2.2 Column Definitions

The following table defines each column in the import file. Columns must appear in this order in the CSV header.

#	Column Name	Type	Nullable	Empty OK	Description
1	COLUMN_ID	INTEGER	No	No	Sequence number indicating the ordinal position of the column within its table. For detail rows (actual columns), this is a positive integer reflecting the source system's column order. For table-level comment rows, this value is 0, which Bridger uses to distinguish table descriptions from column definitions.
2	DATABASE_NAME	VARCHAR	No	No	A logical name identifying the database or instance containing this schema. This is a free-text label — it does not need to match a connection string, SID, or service name exactly. Use a consistent value across all rows for the same source system. Example: CAPS_PROD, AUTOFLOW_UAT.
3	SCHEMA_NAME	VARCHAR	No	No	The schema or namespace from which this column was extracted. In Oracle this is the object owner; in SQL Server or PostgreSQL this is the schema name (e.g., dbo, public). Must be consistent across all rows for the same schema. Example: CAPS, dbo, public.
4	TABLE_NAME	VARCHAR	No	No	The name of the table or view containing this column. Used by Bridger as the primary grouping key when importing schema data. All rows sharing the same TABLE_NAME are grouped into a single table definition.
5	COLUMN_NAME	VARCHAR	No	Yes	The name of the column as defined in the source system. For table-level comment rows (COLUMN_ID = 0), this field is empty. All other rows must have a non-empty COLUMN_NAME. Bridger stores and displays column names exactly as provided — case is preserved.
6	DATA_TYPE	VARCHAR	No	Yes	The data type of the column as defined in the source system. Bridger stores and displays this value as-is without normalization, so use whatever type name the source system provides. Common values: VARCHAR2, NVARCHAR, VARCHAR, CHAR, NUMBER, INT, BIGINT, DECIMAL, FLOAT, DATE, TIMESTAMP, BOOLEAN, CLOB, TEXT. For table-level comment rows, this field is empty.
7	DATA_LENGTH	NUMBER	Yes	Yes	The declared maximum length of the column. Relevant primarily for character and binary types. May be NULL for types where length is not applicable, such as DATE, BOOLEAN, or unbounded LOB types. For table-level comment rows, this field is NULL.
8	DATA_PRECISION	NUMBER	Yes	Yes	For numeric types, the total number of significant digits. NULL for non-numeric types or when precision is unspecified by the schema. A numeric column declared without explicit precision is stored as NULL here. For table-level comment rows, this field is NULL.

#	Column Name	Type	Nullable	Empty OK	Description
9	DATA_SCALE	NUMBER	Yes	Yes	For numeric types, the number of digits to the right of the decimal point. NULL for non-numeric types. A scale of 0 indicates an integer-like numeric type. For table-level comment rows, this field is NULL.
10	DATA_DEFAULT	VARCHAR(4000)	Yes	Yes	The default value expression defined on the column, if any. NULL if no default is defined. Values are stored as-is from the source system's data dictionary, including surrounding quotes or function calls (e.g., CURRENT_TIMESTAMP, 'ACTIVE', 0). Note: some databases store default values in types that require special handling to extract — see Section 3.1 for the Oracle-specific workaround.
11	NULLABLE	CHAR(1)	No	Yes	Indicates whether the column permits NULL values. Valid values: Y (nullable) or N (NOT NULL). Populate from whatever equivalent the source system provides — most databases expose a nullable indicator in their system catalog or information schema. For table-level comment rows, this field is empty.
12	DESCRIPTION	VARCHAR(4000)	Yes	Yes	A human-readable description of the column or table, typically sourced from the data dictionary's comment or remarks facility. For column rows, this is the column-level comment if one exists. For table-level comment rows (COLUMN_ID = 0), this is the table-level comment. Leave empty if the source system has no comment defined. Bridger displays this value in the column and table properties panels.

3. Sample Extraction Query

Bridger accepts schema import files from any source system that can produce the 12-column CSV format described in Section 2. The extraction query below is an Oracle example. Similar queries can be constructed for other platforms using their respective system catalogs:

Platform	Catalog / View
Oracle	ALL_TAB_COLUMNS, ALL_TAB_COMMENTS, DBA_COL_COMMENTS
SQL Server	INFORMATION_SCHEMA.COLUMNS, sys.extended_properties
PostgreSQL	information_schema.columns, pg_catalog.pg_description
MySQL	INFORMATION_SCHEMA.COLUMNS (COLUMN_COMMENT inline)

Regardless of the source platform, the goal is the same: produce one row per column with the required fields populated, plus an optional COLUMN_ID = 0 row per table carrying the table-level description.

3.1 The Oracle DATA_DEFAULT LONG Workaround

Oracle stores the DATA_DEFAULT column in ALL_TAB_COLUMNS as a LONG data type. LONG columns cannot be used in subqueries, UNION operations, or many string functions. The query below works around this limitation by using DBMS_XMLGEN.GETXML to execute a per-row dynamic query and EXTRACTVALUE(XMLTYPE()) to retrieve the result

as a VARCHAR2. This approach is safe for defaults up to 4000 characters. Defaults longer than 4000 characters (extremely rare in practice) will be truncated.

3.2 Table Comment Rows

The UNION ALL block at the end of the query generates one row per table with COLUMN_ID = 0. This row carries the table-level comment from ALL_TAB_COMMENTS into the DESCRIPTION field. Bridger uses the COLUMN_ID = 0 sentinel to distinguish table descriptions from column definitions during import. The ORDER BY clause sorts by table name (column 4) then by COLUMN_ID (column 1), ensuring table comment rows appear before the column rows for each table.

3.3 Query

Replace DBNAME with your database identifier, SCHEMA with the schema owner, and the table name lists with the tables to extract. Both the main SELECT and the UNION ALL table comment block must reference the same table list.

```
SELECT
t.column_id,
'DBNAME' AS DATABASE_NAME,
'SCHEMA' AS SCHEMA_NAME,
t.table_name,
t.column_name,
t.data_type,
t.data_length,
t.data_precision,
t.data_scale,
/* Workaround for LONG type: extract via XMLTYPE */
(
SELECT EXTRACTVALUE(
XMLTYPE(
DBMS_XMLGEN.GETXML(
'SELECT data_default FROM all_tab_columns ' ||
'WHERE owner = ''' || t.owner ||
''' AND table_name = ''' || t.table_name ||
''' AND column_name = ''' || t.column_name || ''''
)
),
'/ROWSET/ROW/DATA_DEFAULT'
)
FROM dual
) AS data_default,
t.nullable,
c.comments AS DESCRIPTION
FROM all_tab_columns t
LEFT JOIN dba_col_comments c
ON t.owner = c.owner
AND t.table_name = c.table_name
AND t.column_name = c.column_name
WHERE t.owner = 'SCHEMA'
AND t.table_name IN (
```

```
'TABLE1'
,'TABLE2'
)
UNION ALL
SELECT
0 AS column_id,
'DBNAME' AS DATABASE_NAME,
'SCHEMA' AS SCHEMA_NAME,
d.table_name,
'' AS column_name,
'' AS data_type,
CAST(NULL AS INTEGER) AS data_length,
CAST(NULL AS INTEGER) AS data_precision,
CAST(NULL AS INTEGER) AS data_scale,
CAST(NULL AS VARCHAR2(4000)) AS data_default,
'' AS nullable,
d.comments AS DESCRIPTION
FROM all_tab_comments d
WHERE d.owner = 'SCHEMA'
AND d.table_name IN (
'TABLE1'
,'TABLE2'
)
ORDER BY 4, 1;
```

3.4 Notes on Privileges

DBA_COL_COMMENTS vs ALL_COL_COMMENTS

The query uses DBA_COL_COMMENTS for column descriptions, which requires DBA or SELECT_CATALOG_ROLE privileges. If those are unavailable, substitute ALL_COL_COMMENTS — it covers tables accessible to the current user.

Schema scope

The WHERE owner = 'SCHEMA' clause scopes the extract to a single schema owner. To extract from multiple schemas, add additional owner values to an IN clause or run the query once per schema and concatenate the results.

Table list maintenance

The table name IN lists in both the main SELECT and the UNION ALL must be kept in sync. Omitting a table from one but not the other will produce mismatched results — column rows with no table comment row, or vice versa.

Exporting to CSV

When exporting from SQL*Plus or SQLcl, set MARKUP CSV ON or use a client tool that supports CSV export with proper quoting. Ensure the header row is included and that NULL values export as empty fields rather than the literal string NULL.

4. Blank Template Header

A blank CSV template ready for manual population or tool export. Copy the header row below as the first line of any new import file.

```
COLUMN_ID,DATABASE_NAME,SCHEMA_NAME,TABLE_NAME,COLUMN_NAME,DATA_TYPE,DATA_LENGTH,DATA_PRECISION,
DATA_SCALE,DATA_DEFAULT,NULLABLE,DESCRIPTION
```