

08 Databases

Architecture

- Wrap each query in a separate SubVI for reuse and documentation.
- Where useful, convert query results into typedef'ed clusters.
- Database connections are shared resources and should not be used in parallel.
- Use one DB connection instance per consumer to avoid blocking.

Design

Naming

- Use lowercase snake_case for table and field names.
- Use singular nouns for table names.
- Use descriptive names and avoid prefixes such as tbl_.
- Do not repeat the table name in field names.
- Avoid SQL and DBMS reserved words.

Keys and Relations

- Use id as the primary key.
- Name foreign keys <referenced_table>_id.
- Define foreign-key constraints wherever possible.
- Name many-to-many relationship tables after the related entities.

Standard Fields

Default fields where applicable:

Field	Description
id	Unique identifier / primary key
created_on	Creation date and time
created_by	Creator
changed_on	Last modification date and time
changed_by	Last modifier
status	Record state

Only add standard fields where they serve a purpose.

Data

- Use NOT NULL unless the absence of a value has a defined meaning.
- Use appropriate native datatypes instead of formatted strings.
- Include units in field names where otherwise ambiguous.
- Use descriptive positive names for boolean fields, e.g. `is_active` or `has_error`.
- Store timestamps in a consistent timezone; prefer UTC for distributed systems.
- Use database constraints such as PRIMARY KEY, FOREIGN KEY, NOT NULL, UNIQUE and CHECK to enforce data integrity.

Queries

- Always use parameterized queries for external values.
- Never construct SQL by concatenating external input.
- Explicitly select required fields instead of using SELECT *
- Add indexes according to actual query and relationship requirements.

Schema Changes

- Keep schema changes reproducible and version controlled.
- Define and track the database schema version where databases can evolve after deployment.
- Test migrations against representative existing databases before deployment.

The HSE Way of Working:

A set of guidelines that recommend programming style, better practices, and methods for all our LabVIEW projects. We ask all our peers to follow these guidelines to help improve the readability of our shared source code and make software maintenance easier.

[<< 07 Networking](#) [09 Cybersecurity >>](#)

From:

<https://dokuwiki.hampel-soft.com/> - **HAMPEL SOFTWARE ENGINEERING**

Permanent link:

<https://dokuwiki.hampel-soft.com/kb/bestpractices/codingconventions/database>

Last update: **2026/10/06 13:28**

