Contents

Introduction
Upgrading
File Format
Data types
Identity columns and sequences
INFORMATION_SCHEMA
General

Introduction

Between version 1.4.200 and version 2.0.202 there have been considerable changes, such that a simple update is not possible.

It would have been nice to write some kind of migration tool, or auto-detect the file and upgrade. Unfortunately, this is purely a volunteer-run project, so this is just the way it has to be. There exists a migration tool H2MigrationTool available in GitHub, but it hasn't been tested by our team. Use at your own risk.

Upgrading

The official way to upgrade is to export it into SQL script with the SCRIPT command **USING YOUR CURRENT VERSION OF H2.**

Then create a fresh database **USING THE NEW VERSION OF H2**, then perform a RUNSCRIPT to load your data. You may need to specify FROM_1X flag, see documentation of this command for details.

MVStore file format

The MVStore file format we use (i.e. the default) is still mostly the same, but some subtle changes have been made to the undo logs, for the purposes of improving crash safety and also read/write performance.

Data types

The maximum length of CHARACTER and CHARACTER VARYING data types is 1_000_000_000 characters. For larger values use CHARACTER LARGE OBJECT.

BINARY and BINARY VARYING are now different data types. BINARY means fixed-length data type and its default length is 1. The maximum length of binary strings is 1_000_000_000 bytes. For larger values use BINARY LARGE OBJECT

NUMERIC / DECIMAL / DEC without parameters now have scale 0. For a variable-scale data type see DECFLOAT. Negative scale isn't allowed for these data types any more. The maximum precision is now 100,000.

ENUM values now have 1-based ordinal numbers.

Arrays are now typed. Arrays with mixed types of elements aren't supported. In some cases they can be replaced with a new ROW data type.

All non-standard data types, with exception for TINYINT, JAVA_OBJECT, ENUM, GEOMETRY, JSON, and UUID are deprecated.

Identity columns and sequences

Various legacy vendor-specific declarations and expressions are deprecated and may not work at all depending on compatibility mode.

Identity columns should be normally declared with GENERATED BY DEFAULT AS IDENTITY or GENERATED ALWAYS AS IDENTITY clauses, options may also be specified. GENERATED ALWAYS AS IDENTITY columns cannot be assigned to a user-provided value unless OVERRIDING SYSTEM VALUE is specified.

NULL cannot be specified as a value for IDENTITY column to force identity generation (with exception for some compatibility modes). Use DEFAULT or simply exclude this column from insert column list.

IDENTITY() and SCOPE_IDENTITY() aren't available in Regular mode. If you need to get a generated value, you need to use data change delta tables or Statement.getGeneratedKeys().

Undocumented Oracle-style .NEXTVAL and .CURRVAL expressions are restricted to Oracle compatibility mode. Other functions are deprecated for Regular mode. Use sequence value expression instead.

INFORMATION_SCHEMA

INFORMATION_SCHEMA in H2 is now compliant with the SQL Standard and other database systems, but it isn't compliant with previous versions of H2. You may need to update your queries.

General

There are a lot more SQL keywords now. Many SQL statements feature far better support of SQL-Standard behaviour. There is a NON_KEYWORDS setting that can be used as a temporary workaround if your application uses them as unquoted identifiers.

Numeric and boolean values aren't comparable. It means you need to use TRUE, FALSE, or UNKNOWN (NULL) as boolean literals. 1 and 0 don't work any more (with exception for some compatibility modes).

Some other non-standard SQL syntax has been restricted to related compatibility modes. Since H2 2.0.204 there is a LEGACY compatibility mode that provides some limited compatibility with previous versions.

Various deprecated grammar elements are marked in red in documentation. Please, avoid their usage.

Migrating an old database to the new version works most of the times. However, there are a couple of important changes in the new version to keep in mind:

- Oracle-style units were never supported officially without being in Oracle compatibility mode, although some worked before. For example, the length of the VARCHAR datatype cannot be more specified using CHAR but CHARACTERS or OCTETS. CHAR and BYTE need to be used in Oracle compatibility mode.
- IDENTITY syntax changed when type is specified: if the type for IDENTITY is specified, then the clause needs to be expanded as INTEGER GENERATED ALWAYS AS IDENTITY. Using just INTEGER IDENTITY is no more working.
- LOG connection setting removed: PageStore was removed from H2 so the "LOG=0" setting at the end of the URL (like "jdbc:h2:file:/tmp/test;LOG=0") is no longer available.