# Example 16 - Optimistic Locking (Row Version Control)

This example shows how to use optimistic locking strategy when updating or deleting table rows.

## How to Run this example

The **Example 16** is included in the download package. To run this example please refer to the section How to Run the Examples above.

## Case #1: Insert

This case inserts a row into a table with row version control. Note the value specified for the version control row (ROW\_VERSION in this case, as specified in the configuration file) is ignored and initialized at zero. The initial, maximum, and minimum can be configured (not this case).

The insert is always successful.

## Case #2: Successful update (no concurrency detected)

This case retrieves a table row and later tries to update it. Since there's no other process or thread that updates it in the mean time, the update succeeds.

## Case #3: Failed update (concurrency detected)

This case retrieves a table row and later tries to update it.

In between another update takes place on the same row (and the ROW\_VERSION value is incremented).

The update fails (since the ROW\_VERSION column does not match anymore) by throwing a StaleDataException.