

NikSQL

User Manual

A fast, modern, skinnable SQL editor for MySQL, MariaDB, PostgreSQL and SQLite

Version 1.1.0

Contents

1. Introduction	4
1.1 About NikSQL	4
1.2 Why use NikSQL	4
1.3 What makes NikSQL shine	4
2. Installation and first launch	4
3. The interface at a glance	5
4. Connecting to a database	6
4.1 Creating a connection	6
4.2 Server types	6
4.3 Remote servers — SSH tunnels	7
4.4 SSL / TLS	7
4.5 Testing a connection	7
4.6 Managing saved connections	7
5. Browsing your database	7
5.1 The connection tree	7
5.2 The table list	7
5.3 Opening a table	7
6. Editing data	8
6.1 The data grid	8
6.2 Editing cells	8
6.3 Sorting	8
6.4 Adding and deleting rows	8
6.5 The right-click menu	8
6.6 Rename, truncate and drop	9
7. The SQL editor	9
7.1 Writing and running queries	9
7.2 Results	9
7.3 History	9
8. Import and export	10
8.1 Exporting SQL	10
8.2 NikPak compressed dumps	10
8.3 Importing dumps	10
9. Personalisation	11

9.1 Skins	11
9.2 Dark and bright	11
9.3 Languages	11
9.4 Zoom	11
10. Keyboard shortcuts	11
11. Licensing and registration	12
12. About, website and socials	12
13. Requirements and troubleshooting	12
13.1 System requirements	12
13.2 Common issues	13

1. Introduction

1.1 About NikSQL

NikSQL is a desktop SQL editor in the spirit of Navicat and phpMyAdmin: connect to a database, browse its structure in a tree, open tables in a sortable, editable grid, and run SQL queries — all through a fast, native interface. It is built in **Rust** with the **Tauri** framework, so the same lightweight, low-memory application runs on **Windows**, **macOS** and **Linux**.

NikSQL speaks four database engines out of the box: **MySQL**, **MariaDB**, **PostgreSQL** and **SQLite**. You can manage several servers side by side, each with its own connection settings, and switch between them instantly from the connection tree.

Like every tool in the Nikkoid family, NikSQL carries a distinctive design identity: a deep electric-blue theme, hand-crafted skins, a flag-based language selector and a sun/moon dark-bright toggle. The interface is designed to be both beautiful and fast, so you can stay in flow while you work.

1.2 Why use NikSQL

One tool, four engines. MySQL, MariaDB, PostgreSQL and SQLite share a single, consistent interface — no need to learn a different client for each server.

Native performance. The engine layer is written in Rust on top of `sqlx`, so browsing, editing and importing data feel instant even on large tables.

Remote-ready. SSH tunnelling, SSL/TLS modes and an initial-database field make connecting to hosted and cloud databases as simple as connecting to a local one.

Compressed dumps. Export and import SQL in the **NikPak** (`.nik`) format, in **Zip**, or as plain SQL — with a choice of compression and optional password protection.

A pleasure to look at. 55 skins, live skin preview, 20 interface languages and Chrome-style zoom let you make the editor truly yours.

1.3 What makes NikSQL shine

Four qualities set NikSQL apart from generic database tools.

The Nikkoid look. A shared visual language across the Nikkoid apps: painted gradient backgrounds, an electric-blue glow, and typography tuned for long sessions. The default “Nikkoid Blue” skin is unmistakable, and a “Nikkoid Bright” light mode is one click away.

Speed by design. Everything heavy — connecting, introspection, query execution, compression — happens in the Rust backend. The interface stays responsive because it never blocks on the database.

Respect for your workflow. Sortable and resizable columns, a right-click context menu everywhere, keyboard shortcuts for every frequent action (**F2** rename, **F5** run, **Ctrl+Enter** run, **Insert** add row), and a command palette (**Ctrl+Shift+P**) that lists every command.

Portable by nature. A single binary per platform, with your connections, skins, language and licence kept in a small local store. Export a database, move it, and re-import it anywhere — compressed and, if you like, encrypted.

2. Installation and first launch

Download the installer for your platform from the official website, niksql.nikkoid.com. Windows users run the `.msi` or `.exe` installer; macOS users open the `.dmg` and drag NikSQL into Applications; Linux users can use the provided `.AppImage` or `.deb` package.

On first launch, NikSQL shows a short shareware reminder. The software is **shareware**: you may test it for free, without time limit, for personal use. Companies and organisations need one licence per seat. Registering – see Section 11 – removes the reminder and unlocks the registered title bar.

If the application ever closes unexpectedly, it writes a `niksql-panic.log` file next to the executable. Keep that file if you report a problem; it describes exactly what happened.

No database yet? Download the classic “employees” sample database for with every example in this manual.

3. The interface at a glance

The main window is organised as a row of controls on top, a three-panel work area in the middle, and a row of function keys at the bottom. Every panel can be resized by dragging the thin splitter between it and its neighbour.

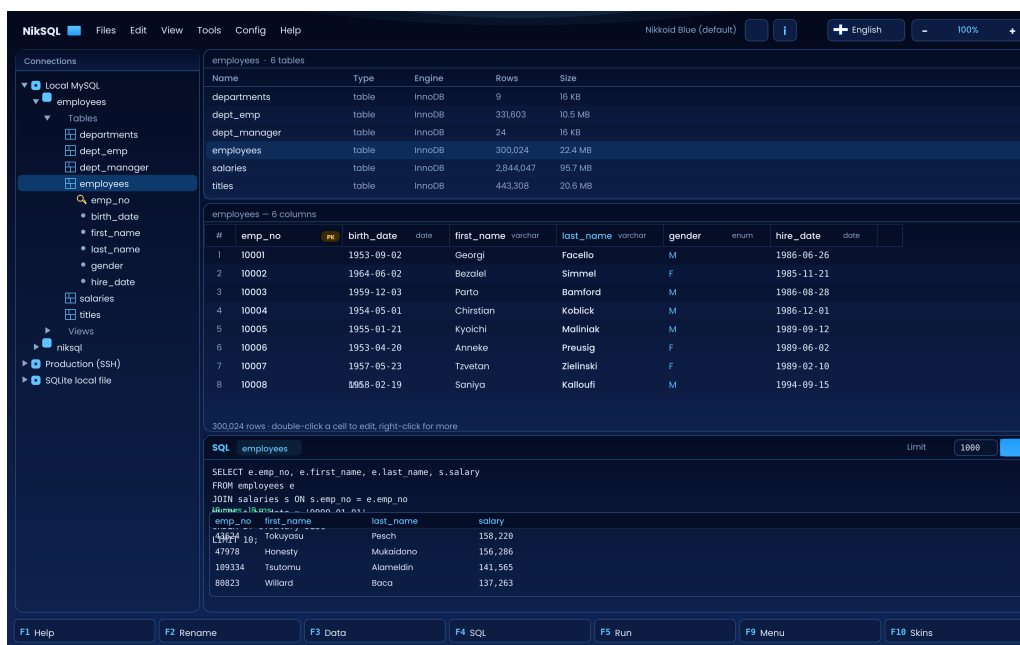


Figure 1 – The NikSQL main window: menu bar and zoom controls (top), connection tree (left), table list and data grid (centre), SQL editor (bottom), function-key bar (bottom edge).

Menu bar – Files, Edit, View, Tools, Config and Help. Press **F9** to open the menus from the keyboard.

Connection tree (left) – your saved connections; expand a connection to see its databases, tables, views and columns.

Table list – the tables of the currently selected database, sortable by name, type, engine, row count and size.

Data grid – the rows of the currently open table; editable, sortable, with a right-click menu.

SQL editor – a free-form editor with a Run button and a results area.

Function-key bar – **F1** Help, **F2** Rename, **F3** Data, **F4** SQL, **F5** Run, **F9** Menu, **F10** Skins.

Top-right controls – zoom out / reset / zoom in, language selector, about box, and the dark-bright toggle.

4. Connecting to a database

4.1 Creating a connection

Choose **Files New connection** (or press **Ctrl+N**). The connection dialog asks for the server details; only the name, server type, host, port and credentials are required for a local server.

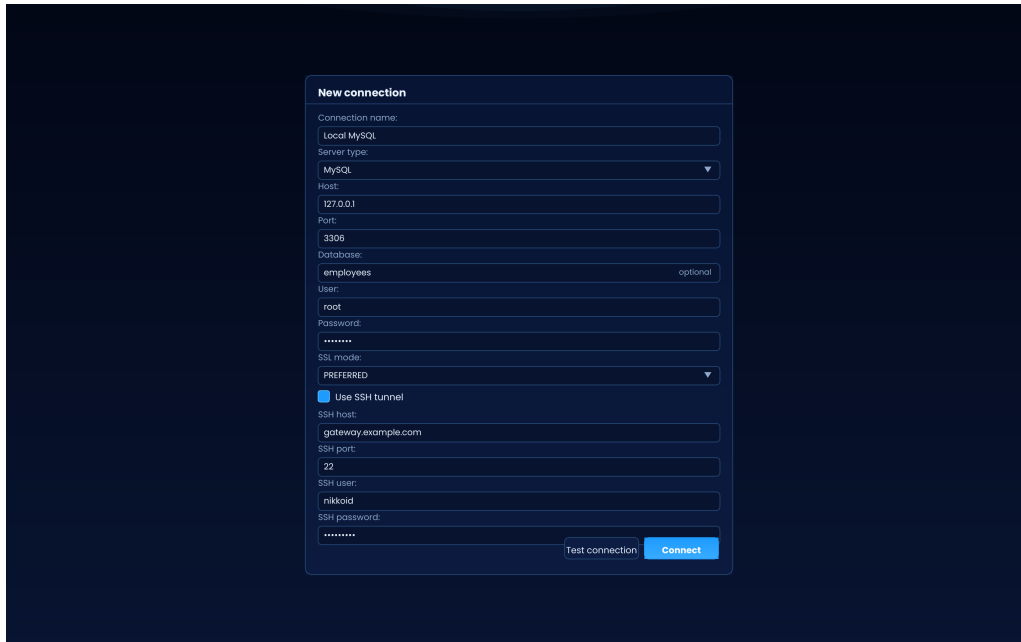


Figure 2 – The New connection dialog, with the SSH tunnel and SSL options expanded.

Field	Description
Connection name	A friendly label shown in the connection tree.
Server type	MySQL, MariaDB, PostgreSQL or SQLite.
Host	The server host name or IP address (e.g. <code>localhost</code> , <code>db.example.com</code>).
Port	The server port – 3306 for MySQL/MariaDB, 5432 for PostgreSQL.
Database	Optional initial database; you can also pick one in the tree after connecting.
User / Password	The database login credentials.
SSL mode	The TLS behaviour (Section 4.4).
Database file	SQLite only – the path to the <code>.db</code> / <code>.sqlite</code> file.
Use SSH tunnel	Route the connection through an SSH server (Section 4.3).

4.2 Server types

NikSQL adapts to the selected engine automatically. MySQL and MariaDB share a driver but are listed separately so the defaults are clear; PostgreSQL uses its own wire protocol; SQLite opens a local file and needs no host, port, user or password. For SQLite, choose the **Browse...** button to locate the file on disk.

Changing the server type resets the port to the engine default and swaps options that apply.

4.3 Remote servers — SSH tunnels

When your database is behind a firewall, or is only reachable from a specific machine, tick **Use SSH tunnel**. NikSQL then connects to the SSH server first and forwards the database traffic through an encrypted SSH channel — the same technique Navicat and other professional clients use.

SSH host / port — the address of the SSH server (usually port 22).

SSH user / password — an account on the SSH server (not the database server).

The **Host** and **Port** fields in the general section should be the database address **as seen from the SSH server** — often `localhost` or the private network name.

The SSH server must allow TCP forwarding (`AllowTcpForwarding` yes) on most Linux servers.

4.4 SSL / TLS

The **SSL mode** field controls how the connection is encrypted and verified. For PostgreSQL the modes are `prefer`, `require`, `verify-ca`, `verify-full` and `disable`; for MySQL and MariaDB the uppercase equivalents `PREFERRED`, `REQUIRED`, `VERIFY_CA`, `VERIFY_IDENTITY` and `DISABLED` are used. `PREFERRED/prefer` encrypts when the server supports it and is a sensible default; `REQUIRED/require` insists on encryption; the `VERIFY_*` modes additionally check the server certificate.

4.5 Testing a connection

Before saving, click **Test connection**. NikSQL opens a real connection, reads the server version, and reports “Connection works” with the version number — or the exact reason it failed. A successful test does not save the connection; press **Connect** to store it and connect immediately.

4.6 Managing saved connections

Right-click a connection in the tree to **Connect**, **Rename**, **Edit** or **Delete** it. The same actions are available in the Files menu. Connections are remembered between sessions, so you only configure each server once. Use **Files Connect...** (`Ctrl+O`) to pick a saved connection from a searchable list.

5. Browsing your database

5.1 The connection tree

The tree on the left shows every saved connection. Expand a connection to list its databases, a database to list its **Tables** and **Views**, and a table or view to see its **columns** with their types and keys. A gold key marks primary-key columns. Double-clicking a node expands or collapses it; single-clicking selects it and opens the relevant view on the right.

5.2 The table list

Selecting a database fills the table list with every table and view, together with its engine, row count and size. Click any column header to sort by that column; click again to reverse the order. The column headers and the data-grid columns are both **resizable** — drag the right edge of a header.

5.3 Opening a table

Click a table (or right-click **Open**) to load its data into the grid. NikSQL reads the table’s columns and fetches the first 5,000 rows; a note under the grid tells you when the display is truncated.

6. Editing data

6.1 The data grid

The grid shows one column per table column, plus a row-number column and a delete column. Primary-key columns are bold and carry a gold **PK** badge; **NULL** values appear in dimmed italics; alternating rows are lightly shaded for readability.



#	emp_no	birth_date	first_name	last_name	gender	hire_date
1	10001	1953-09-02	Georgi	Facello	M	1986-06-26
2	10002	1964-06-02	Bezalel	Simmel	F	1985-11-21
3	10003	1959-12-03	Parto	Bamford	M	1986-08-28
4	10004	1954-05-01	Chirstian	Koblick	M	1986-12-01
5	10005	1955-01-21	Kyoichi	Maliniak	M	1989-09-12
6	10006	1953-04-20	Anneke	Preusig	F	1989-06-02
7	10007	1957-05-23	Tzvetan	Zielinski	F	1989-02-10
8	10008	1958-02-19	Saniya	Kozlov	F	1994-09-15

Figure 3 – The data grid, with a sorted column and the right-click menu open on a cell.

6.2 Editing cells

Double-click a cell (or right-click **Edit cell**, or press **F2**) to edit it inline. Type the new value and press **Enter**; **Escape** cancels, **Tab** commits and moves on. An empty value is stored as **NULL**. You can also right-click a cell and choose **Set to NULL** or **Copy cell value**.

A table needs a primary key before its rows can be edited or deleted. NikSQL asks for confirmation against the primary key so you never modify the wrong row by accident.

6.3 Sorting

Click a column header to sort the grid by that column; click again to reverse the order. Sorting is client-side and instant – it reorders the rows you have already loaded.

6.4 Adding and deleting rows

Press **Insert**, or right-click a row and choose **Add row**, to insert a new empty row into the table. To delete a row, click the **delete** icon at its end, or right-click the row and choose **Delete row**; NikSQL asks for confirmation on delete, so use the action deliberately.

6.5 The right-click menu

Right-clicking a cell, a row, a table in the list, or a node in the tree opens a context menu with the actions that make sense there – edit, set to NULL, copy, add or delete a row on cells; open, rename, truncate or drop on tables; connect, rename, edit or delete on connections. Every action is also reachable from the keyboard via the menus.

6.6 Rename, truncate and drop

Select a table or view and press **F2** (or right-click **Rename**) to give it a new name. **Truncate** empties a table of all its rows; **Drop** removes the table or view permanently after a confirmation. These actions build the correct SQL for the active engine — for example, SQLite truncates with **DELETE FROM** because it has no **TRUNCATE** statement.

7. The SQL editor

The editor at the bottom of the window is a complete SQL console. Type any number of statements, press **Run** (**F5** or **Ctrl+Enter**), and read the result in the area below.



Figure 4 — The SQL editor showing a **JOIN** query and its results in a grid.

7.1 Writing and running queries

Statements are split safely on **;**, with quotes, backticks and comments respected, so you can paste a whole dump and run it at once. **SELECT**, **SHOW**, **DESCRIBE**, **EXPLAIN**, **PRAGMA** and **WITH** queries return a result set; everything else (**INSERT**, **UPDATE**, **CREATE**, **DROP**...) reports the number of affected rows.

7.2 Results

A query that returns columns shows them in a grid, with the elapsed time and row count in a status line. Errors are shown inline in red, in a monospaced font so the server's message stays readable. The **Limit** field caps the number of rows fetched — raise it for larger result sets.

7.3 History

The editor keeps a history of the last 60 statements. With the caret on the first line, press **Up** / **Down** to step through what you have run, and press **Ctrl+L** to clear the editor.

8. Import and export

8.1 Exporting SQL

Choose **Files Export SQL...** (**Ctrl+E**) with a database selected. You can include the **structure** (**CREATE TABLE**), the **data** (**INSERT** statements), and an optional **DROP TABLE before CREATE**. Leave the table list empty to export the whole database, or list specific tables separated by commas.

8.2 NikPak compressed dumps

After the SQL is generated, NikSQL asks how you would like to save it. **NikPak (.nik)** is the default: a single-file format shared with the NikPak archiver, using Zstandard compression.

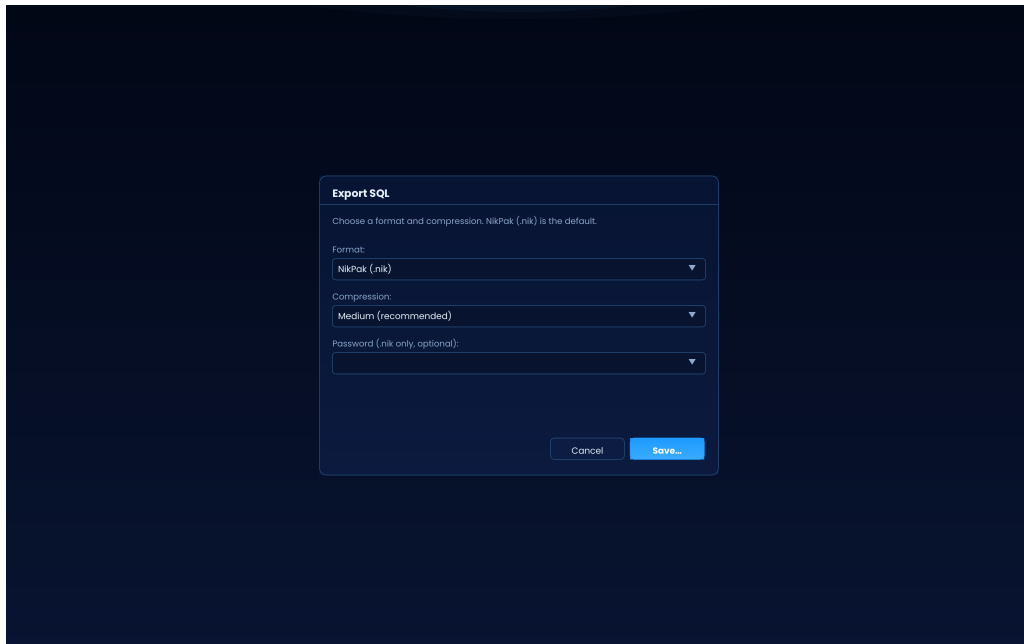


Figure 5 — Choosing a format and compression for an export.

Format — NikPak (**.nik**), Zip (**.zip**) or plain SQL (**.sql**).

Compression — Medium (recommended), Low (fastest) or Maximum (smallest).

Password — optional, for **.nik** files only. A password-protected dump is encrypted with Argon2id key derivation and XChaCha20-Poly1305, the same scheme NikPak uses, so a protected dump opens in NikPak and vice-versa.

A **.nik** dump written by NikSQL opens in the NikPak application. Note that older archives use a more advanced codec that NikSQL does not yet read; for more compatibility, use NikPak, or create NikPak archives at its fast/normal levels.

8.3 Importing dumps

Choose **Files Import SQL...** (**Ctrl+I**). You can pick a **.nik**, **.zip** or **.sql** file — NikSQL detects the format automatically and decompresses it for you — or paste SQL directly. If you select an encrypted **.nik** file, NikSQL asks for its password. Imported statements are split on **;** and executed in order; if one fails, the import stops and reports the statement number.

9. Personalisation

9.1 Skins

NikSQL ships with **55 skins**, from the signature Nikkoid Blue to bright, high-contrast and monochrome themes. Press **F10** (or **Config Choose skin**) to open the picker; moving up and down the list previews each skin live, and **Enter** applies the one you like.

9.2 Dark and bright

The sun/moon button in the top-right corner toggles instantly between **Nikkoid Blue** (dark) and **Nikkoid Bright** (light). The shortcut is **Ctrl+Shift+B**.

9.3 Languages

The flag button opens the language selector — **20 languages** with their native names and flags. Choose a language and NikSQL restarts in it. The shortcut is **Ctrl+Shift+L**.

9.4 Zoom

The **- / 100% / +** controls in the top-right zoom the whole interface in Chrome's steps, exactly like a browser. The shortcuts are **Ctrl+-**, **Ctrl+0** and **Ctrl++**; holding **Ctrl** and scrolling the mouse wheel also zooms. The level is remembered between sessions.

10. Keyboard shortcuts

Shortcut	Action
F1	Help and shortcuts
F2	Rename the selected connection, table or view
F3	Focus the data grid
F4	Focus the SQL editor
F5 / Ctrl+Enter	Run SQL
F9	Open the menus
F10 / Alt+S	Choose skin
Ctrl+N	New connection
Ctrl+0	Connect to a saved connection
Ctrl+W	Disconnect
Ctrl+I	Import SQL
Ctrl+E	Export SQL
Ctrl+R	Refresh the current view
Insert	Add a row
Ctrl+Shift+P	Command palette
Ctrl+Shift+B	Dark / bright toggle
Ctrl+Shift+L	Language
Ctrl+- / Ctrl+0 / Ctrl++	Zoom out / reset / in

11. Licensing and registration

NikSQL is shareware. Personal use is free without a time limit; companies and organisations need **one licence per seat**. Registration unlocks the clean title bar and removes the start-up reminder.

Choose **Help Register NikSQL**, enter your name and e-mail exactly as they appear on your licence, and paste the key (**NKM1 - ...**). Keys are signed with Nikkoid's private key and verified on your machine, so a valid key works offline. **Help Buy a licence** opens the purchase page.

12. About, website and socials

The **About NikSQL** box (**Help ▶ About**) shows the version, the copyright, a link to the website niksql.nikkoid.com, and links to the Nikkoid communities on **Telegram**, **X** and **Discord**.

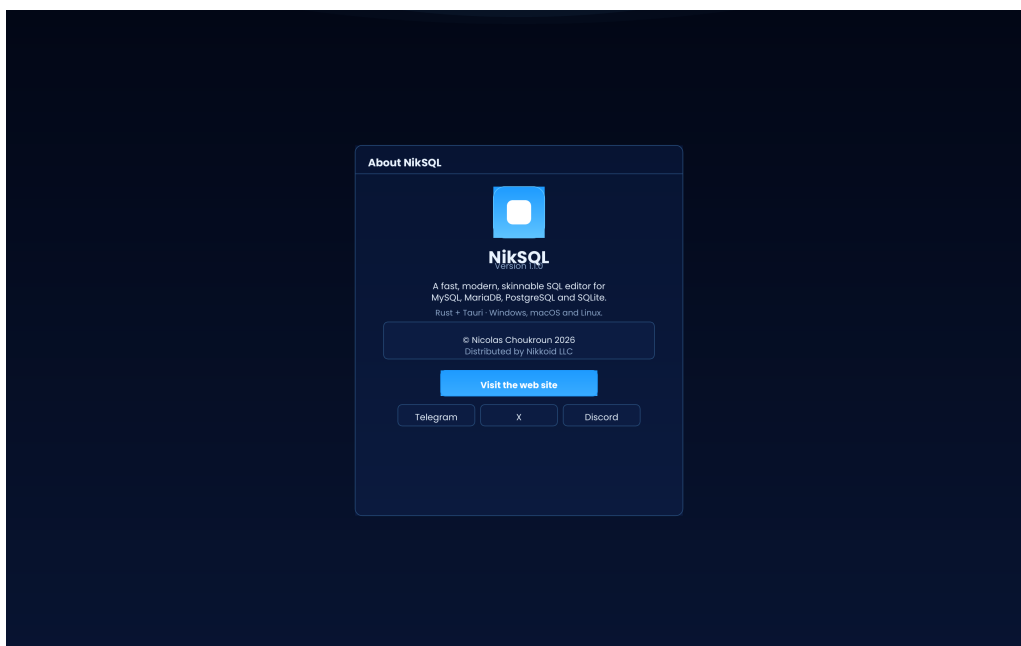


Figure 6 — The About box, with the website link and the Telegram / X / Discord buttons.

13. Requirements and troubleshooting

13.1 System requirements

Windows 10 or later (x64), **macOS** 11 or later (Intel / Apple Silicon), **Linux** with a modern desktop (glibc 2.31+).

Around 60 MB of disk space and 150 MB of memory for a typical session.

Network access to any database server (or an SSH server for tunnelled connections).

13.2 Common issues

“Connection refused” — check the host, port and that the database server is running; for remote servers, confirm the firewall allows the port.

SSH authentication failed — the SSH user/password must be a valid account on the SSH server, and TCP forwarding must be enabled.

Certificate errors in `VERIFY_*` mode — the server certificate is not signed by a trusted CA; switch to `REQUIRED/require` or install the CA.

“This table has no primary key” — add a primary key before editing rows in the grid.

A `.nik` file won’t import — NikPak’s maximum-compression archives use a codec NikSQL does not yet read; re-export from NikSQL or create the NikPak archive at a lower level.

The app closed unexpectedly — inspect `niksql-panic.log` next to the executable and include it in any bug report.

For help, announcements and community support, visit niksql.nikkoid.com or the social channels listed in the About box.