Nhance Unity Database Tool - User Manual

Welcome to the Nhance Unity Database Tool, your all-in-one solution for working with SQLite and MySQL databases directly inside the Unity Editor (possibly even more in future). This tool provides powerful, user-friendly features for managing databases, editing tables, running queries, and inspecting schemas, supercharged with AI capabilities for SQL generation.

Installation

There is **PackageManifestPatcher.cs** script is a Unity Editor script designed to automate the installation of necessary Unity Package Manager (UPM) packages and the import of specific files and embedded packages when a Unity project is opened. It operates in Editor Mode, ensuring your project has all its required dependencies.

How PackageManifestPatcher Works

The script runs every time the editor updates (EditorApplication.update) and executes a sequence of steps:

1. Checking Installed UPM Packages
2. Installing UPM Packages
3. Importing Embedded Package and Moving File

Newtonsoft.Json and Editor Coroutines

com.unity.nuget.newtonsoft-json@3.0.2 (Newtonsoft.Json)

- **What is it?** Newtonsoft.Json (also known as Json.NET) is a very popular, high-performance JSON library for .NET. It's widely used for serializing and deserializing C# objects to and from JSON.
- **Why is it needed?** UnityDatabaseTool handles data in JSON format (e.g., for configuration, saving/loading data, network communication), Newtonsoft.Json is an excellent choice for these operations. While Unity has a built-in JSON parser (JsonUtility), Newtonsoft.Json offers significantly more functionality and flexibility, including advanced serialization capabilities, support for complex data types, and better performance for certain scenarios.

com.unity.editorcoroutines@1.0 (Editor Coroutines)

- **What is it?** Standard Unity coroutines (those started with StartCoroutine()) can only run in Play Mode. The Editor Coroutines package provides the ability to run coroutines in Editor Mode.
- **Why is it needed?** This library allows you to perform asynchronous, long-running operations in the Unity editor without blocking the main thread. In the context of our PackageManifestPatcher at UnityDatabaseTool, this could be useful for:
  - **Loading or processing large databases:** If database operations take a long time, using editor coroutines will keep the Unity interface responsive.
  - **Network requests:** If your tool needs to interact with remote services.
  - **Other long-running asset operations:** Any actions that take considerable time and might otherwise cause the editor to "freeze."

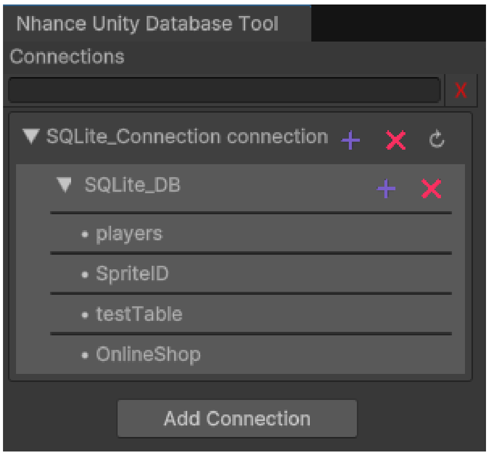
Interface Overview

The tool is divided into two main panels:

1. **Left Panel (Connections & Hierarchy):** Your central hub for managing database connections and navigating their structure.
2. **Right Panel (Workspace):** A dynamic area where you interact with your selected database or table through a system of tabs.

Left Panel: Connections & Hierarchy

This is the starting point for all your work. It allows you to manage connections and view their contents in a collapsible tree structure.



Features:

- **Search Bar:** Located at the top, this bar allows you to instantly filter connections, databases, and tables by name.
- **Add Connection Button:** Opens a form to create a new connection. Two types are supported:
  - **Local (SQLite):** For local SQLite database files. You can provide the path manually or use the "Browse" button to select a file.
  - **Remote (MySQL):** For connecting to a remote MySQL server. You must provide the Server Address, Database Name, User ID, Password, and SSL Mode.
- **Connection List:** Displays all added connections. Each connection can be expanded to show its databases, and each database can be expanded to show its tables.
- **Connection-Level Actions:**
  - ➕ **(Add Database):** Opens a modal window to create a new database within this connection.
  - ❌ **(Delete Connection):** Removes the selected connection from the list.
  - ⭮ **(Refresh):** Refreshes the hierarchy for the selected connection.
- **Database-Level Actions:**
  - ➕ **(Add Table):** Opens a window to create a new table in the selected database.
  - ❌ **(Delete Database):** Deletes the selected database.
- **Table-Level Actions:**
  - **Clicking on a table name:** Opens the table's content and structure in the Workspace panel on the right.
  - ❌ **(Delete Table):** Deletes the selected table.

Workspace Tabs Breakdown

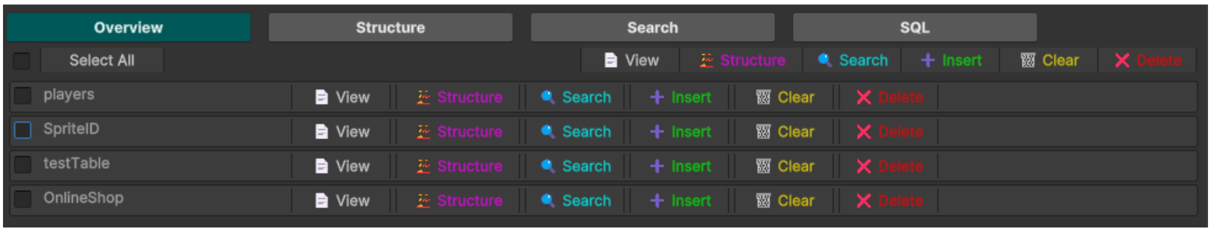
After selecting a database or table from the left panel, the right panel becomes active, offering four distinct tabs for interaction.

1. Overview Tab

This is the primary tab for data viewing and manipulation. Its content changes depending on whether you've selected a database or a specific table.

If a Database is Selected:

You will see a list of all tables within that database.



Overview | Structure | Search | SQL

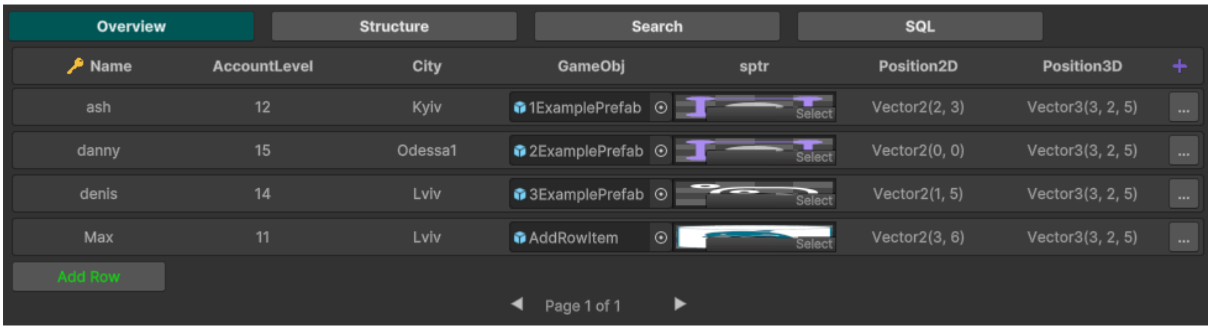
□ Select All	View	Structure	Search	Insert	Clear	Delete
□ players	View	Structure	Search	Insert	Clear	Delete
□ SpriteID	View	Structure	Search	Insert	Clear	Delete
□ testTable	View	Structure	Search	Insert	Clear	Delete
□ OnlineShop	View	Structure	Search	Insert	Clear	Delete

You can select one or more tables using checkboxes to perform bulk actions:

- 📄 **View:** Executes a `SELECT * FROM {tableName}` query and switches to the "SQL" tab to display the results.
- 🏗️ **Structure:** Switches to the "Structure" tab to view and edit the selected table's schema.
- 🔍 **Search:** Switches to the "Search" tab for the selected table.
- ➕ **Insert:** Opens a modal window to add a new row to the table.
- 🗑️ **Clear:** Deletes all data from the table (requires confirmation).
- ❌ **Delete:** Completely drops the table from the database (requires confirmation).

If a Table is Selected:

An interactive grid displays the table's data.



Overview | Structure | Search | SQL

🔑 Name	AccountLevel	City	GameObj	sptr	Position2D	Position3D	+
ash	12	Kyiv	1ExamplePrefab	Select	Vector2(2, 3)	Vector3(3, 2, 5)	...
danny	15	Odessa1	2ExamplePrefab	Select	Vector2(0, 0)	Vector3(3, 2, 5)	...
denis	14	Lviv	3ExamplePrefab	Select	Vector2(1, 5)	Vector3(3, 2, 5)	...
Max	11	Lviv	AddRowItem	Select	Vector2(3, 6)	Vector3(3, 2, 5)	...

Add Row

◀ Page 1 of 1 ▶

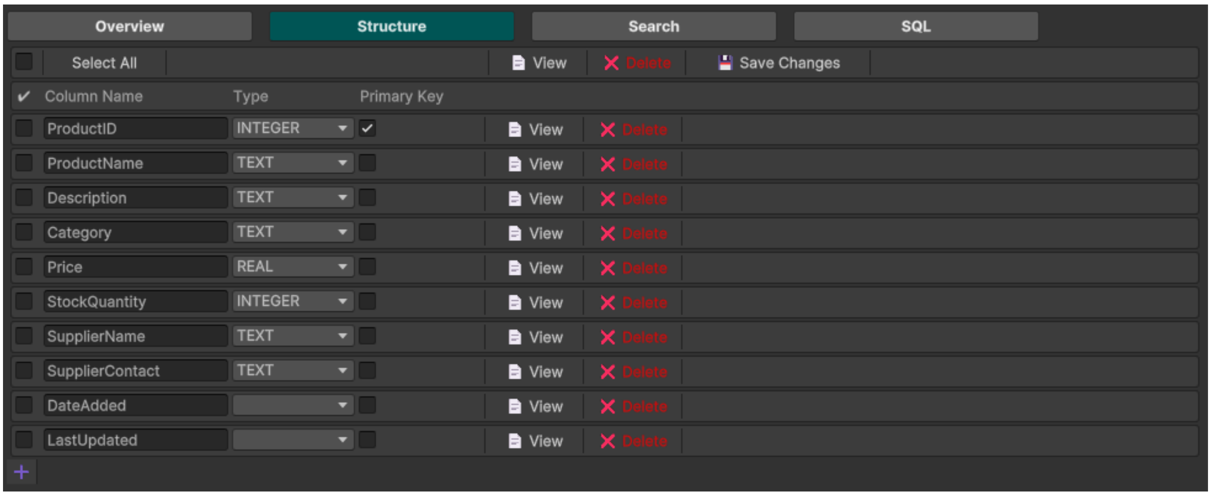
- **Viewing and Pagination:** Data is shown in pages of 10 rows to handle large tables efficiently. You can navigate between pages using the ◀ and ▶ buttons.

- **Cell Editing:** Cell values are directly editable. Standard data types use text fields, while special Unity types like `GameObject` and `Sprite` use an `ObjectField`, allowing you to drag and drop assets directly into the cell.
- **Context Menus (Right-Click):**
  - **On a Column Header:**
    - `Edit Column`: Modify the column's name and data type.
    - `Delete Column`: Remove the column.
    - `Make Primary Key`: Set this column as the primary key, marked with a 🔑 icon.
  - **On a Cell:**
    - `Change Value`: Open a dedicated window to edit the value.
    - `Copy`: Copy the cell's value to the clipboard.
- **Row and Column Management:**
  - **Add Row Button:** Opens a window to insert a new row.
  - `...` **Button (at the end of a row):** Opens a context menu to `Duplicate Row` or `Delete Row`.
  - ➕ **Button (in the header):** Allows you to add a new column to the table.

---

2. Structure Tab

This tab is for viewing and modifying the schema (structure) of the selected table.

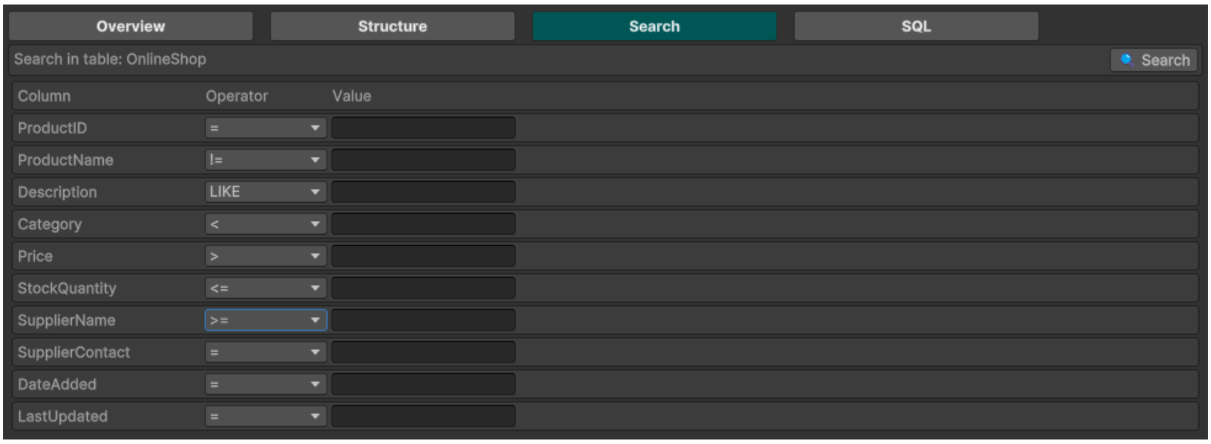


✔	Column Name	Type	Primary Key
□	ProductID	INTEGER	☑
□	ProductName	TEXT	□
□	Description	TEXT	□
□	Category	TEXT	□
□	Price	REAL	□
□	StockQuantity	INTEGER	□
□	SupplierName	TEXT	□
□	SupplierContact	TEXT	□
□	DateAdded		□
□	LastUpdated		□

- **Column List:** Displays all table columns with their name, type, and a toggle indicating if it's a primary key.
- **Schema Editing:** You can edit the name of any column, change its type from a dropdown list, and set a primary key. The available column types are: `TEXT`, `VARCHAR`, `INTEGER`, `REAL`, `BLOB`, `GameObject`, `Sprite`, `Vector2`, and `Vector3`.
- **Saving Changes:** Click the "Save Changes" button to apply your modifications to the database.
- **Bulk Actions:** You can select multiple columns via checkboxes to perform bulk operations, such as deleting them.

3. Search Tab

This tab allows you to query data from a table using a simple form, without writing any SQL code.



Search in table: OnlineShop

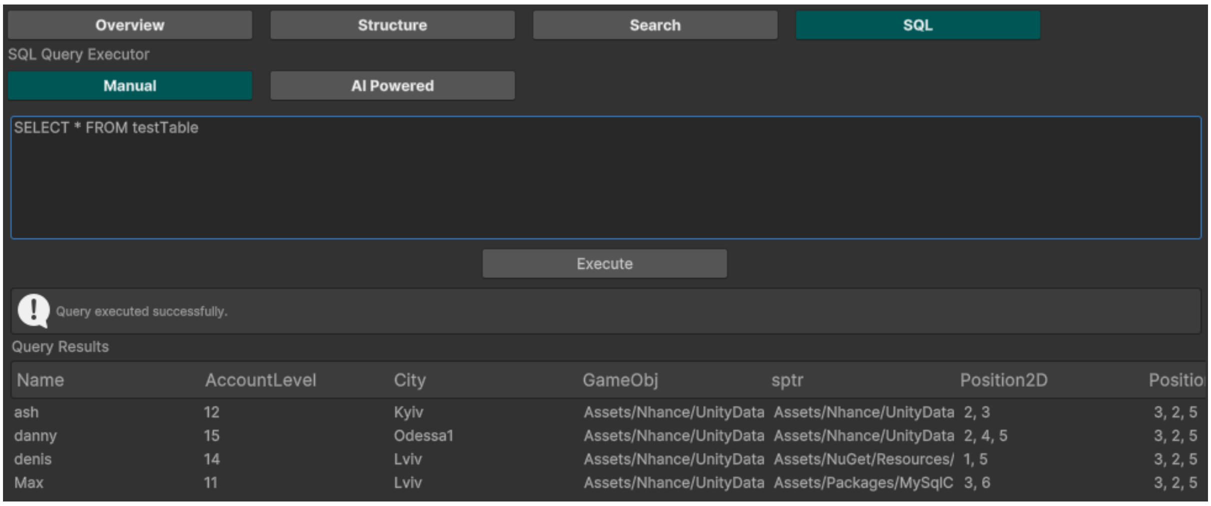
Column	Operator	Value
ProductID	=	
ProductName	!=	
Description	LIKE	
Category	<	
Price	>	
StockQuantity	<=	
SupplierName	>=	
SupplierContact	=	
DateAdded	=	
LastUpdated	=	

- **Filter Builder:** For each column in the table, you can define a search condition.
- **Conditions:** Choose an operator from the dropdown list (`=`, `!=`, `LIKE`, `<`, `>`, `<=`, `>=`) and enter a value to search for.
- **Execute Search:** Click the "Search" button to run the query. The results will be displayed in a table at the bottom of the tab.

4. SQL Tab

A powerful tool for executing custom SQL queries, featuring two distinct modes.

Manual Mode



SQL Query Executor

```
SELECT * FROM testTable
```

Query executed successfully.

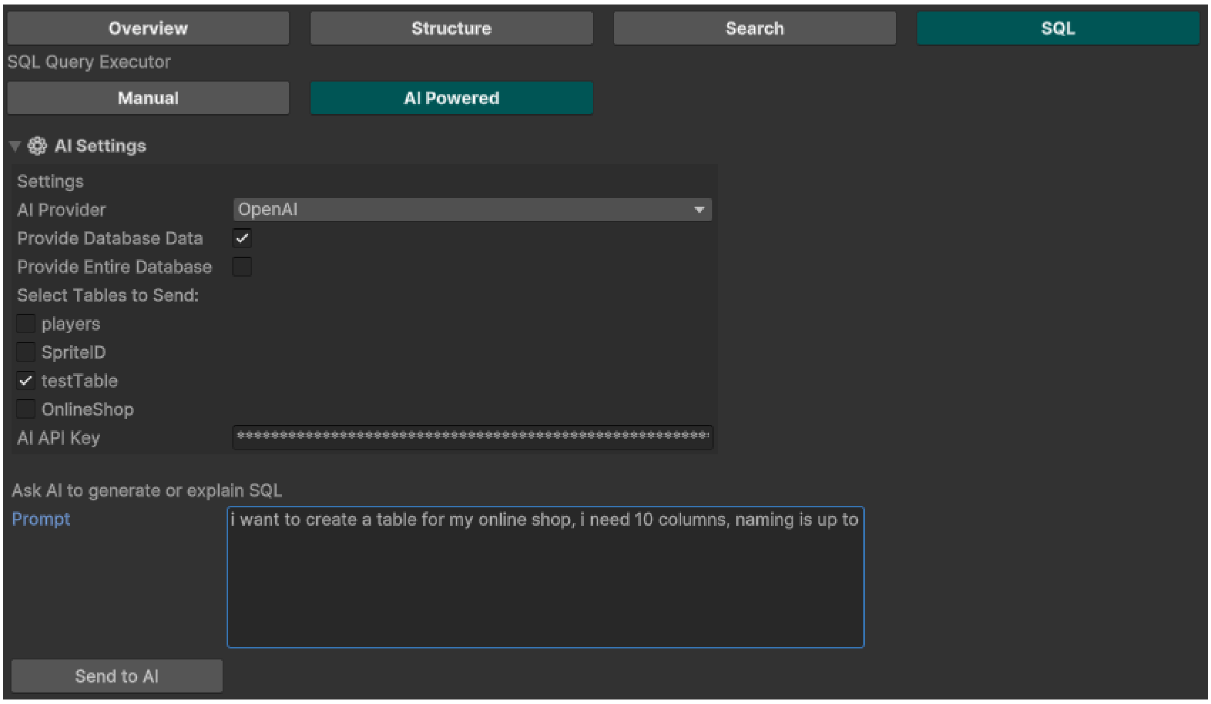
Query Results

Name	AccountLevel	City	GameObj	sptr	Position2D	Positio
ash	12	Kyiv	Assets/Nhance/UnityData	Assets/Nhance/UnityData	2, 3	3, 2, 5
danny	15	Odessa1	Assets/Nhance/UnityData	Assets/Nhance/UnityData	2, 4, 5	3, 2, 5
denis	14	Lviv	Assets/Nhance/UnityData	Assets/NuGet/Resources/	1, 5	3, 2, 5
Max	11	Lviv	Assets/Nhance/UnityData	Assets/Packages/MySqlC	3, 6	3, 2, 5

- **Query Editor:** A large text area for writing your SQL queries.
- **Execute Button:** Runs the query in the editor.
- **Results Display:**
  - For `SELECT` queries, a results table will appear below.
  - For other queries (`INSERT`, `UPDATE`, `DELETE`, etc.), a status message will show the outcome and the number of affected rows.
  - Errors are displayed in red text for easy debugging.

AI Powered Mode

This innovative feature uses natural language to generate SQL queries.



⚙ AI Settings

Settings

AI Provider: OpenAI

Provide Database Data ☑

Provide Entire Database ☐

Select Tables to Send:

☐ players
☐ SpriteID
☑ testTable
☐ OnlineShop

AI API Key: **

Ask AI to generate or explain SQL

Prompt: i want to create a table for my online shop, i need 10 columns, naming is up to

- **AI Settings (⚙️ `AI Settings`):**
  - **AI Provider:** Choose your AI service provider from a list including OpenAI, Groq, DeepSeek, and Nvidia.
  - **AI API Key:** Enter your API key for the selected service.
  - **Provide Database Data:** When enabled, the tool sends information about your tables to the AI as context.
    - `Provide Entire Database`: Sends the schema and the first 50 rows of data from *all* tables. **Warning:** This can consume a significant number of tokens.
    - If the above is disabled, you can select specific tables to send as context.
- **Prompt Field:** Enter your request in plain English. For example: *"show all users with a level greater than 10"* or *"write a query to add a new item named 'Sword' with a damage of 25"*.
- **Send and Receive:** Click "Send to AI". The AI's response will appear below. If the AI returns a valid SQL code block (formatted with ` ```sql `), it will be automatically extracted and placed into the Manual Mode editor, ready for you to execute.
- **AI Behavior:** The AI is guided by a system prompt to be a helpful SQL assistant. It is instructed to return raw data values if asked for a specific value (e.g., "what is the first name?") and to return a SQL query if asked to write one (e.g., "write a query to get all users").

Additional Features

- **Session Persistence:** The tool automatically saves your workspace state, including your connection list, the expanded state of each item in the hierarchy, and your last opened table. When you reopen the window, you can pick up right where you left off.
- **Unity Data Type Support:** Native support for `GameObject`, `Sprite`, `Vector2`, and `Vector3` makes it seamless to manage game-specific data directly within the database.

Special cases breakdown

- Installation of Newtonsoft can be failed depending on your firewall or antivirus (by some cause…). In this case only thing you can do – install it by yourself.