

# How to create tests cases for judge42

# Problem statement [1/2]

Write a program that reads two integer numbers and output the result of they product and they addition.

## INPUTS

Two lines each one with an integer number.

## OUTPUTS

Two lines, first one must show «**A x B = C**» where **A** and **B** are the integer numbers readed and **C** is result of they product. Second line must show «**A + B = C**», where **A** and **B** are the integer numbers readed and **C** is result of they addition.

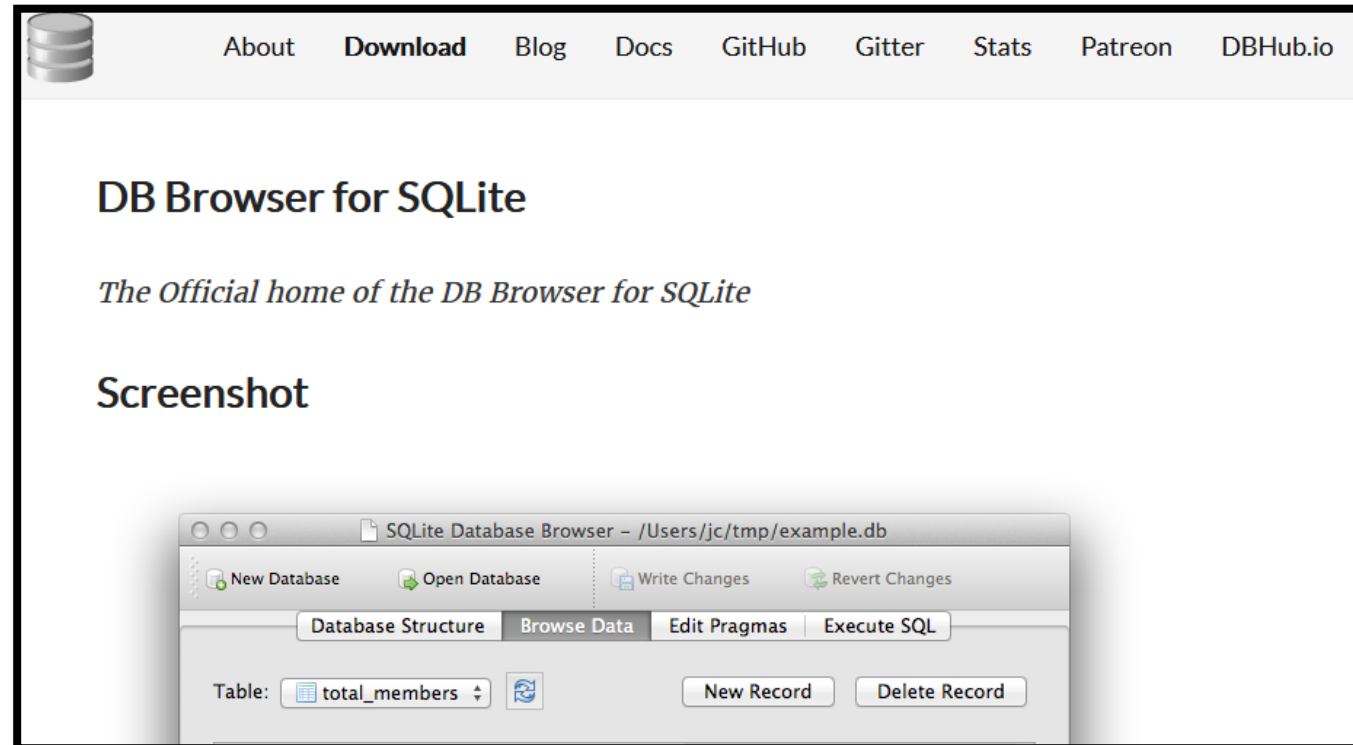
# Problem statement [2/2]

## INPUTS AND OUTPUTS EXAMPLES

INPUT	OUTPUT
2	$2 \times 3 = 6$
3	$2 + 3 = 5$
9	$9 \times 9 = 81$
9	$9 + 9 = 18$
30	$30 \times 20 = 60$
20	$30 + 20 = 50$

# Create a tests cases [1/]

1. Download and install (or unzip) SQLiteDatabaseBrowser:
  - <https://sqlitebrowser.org/>



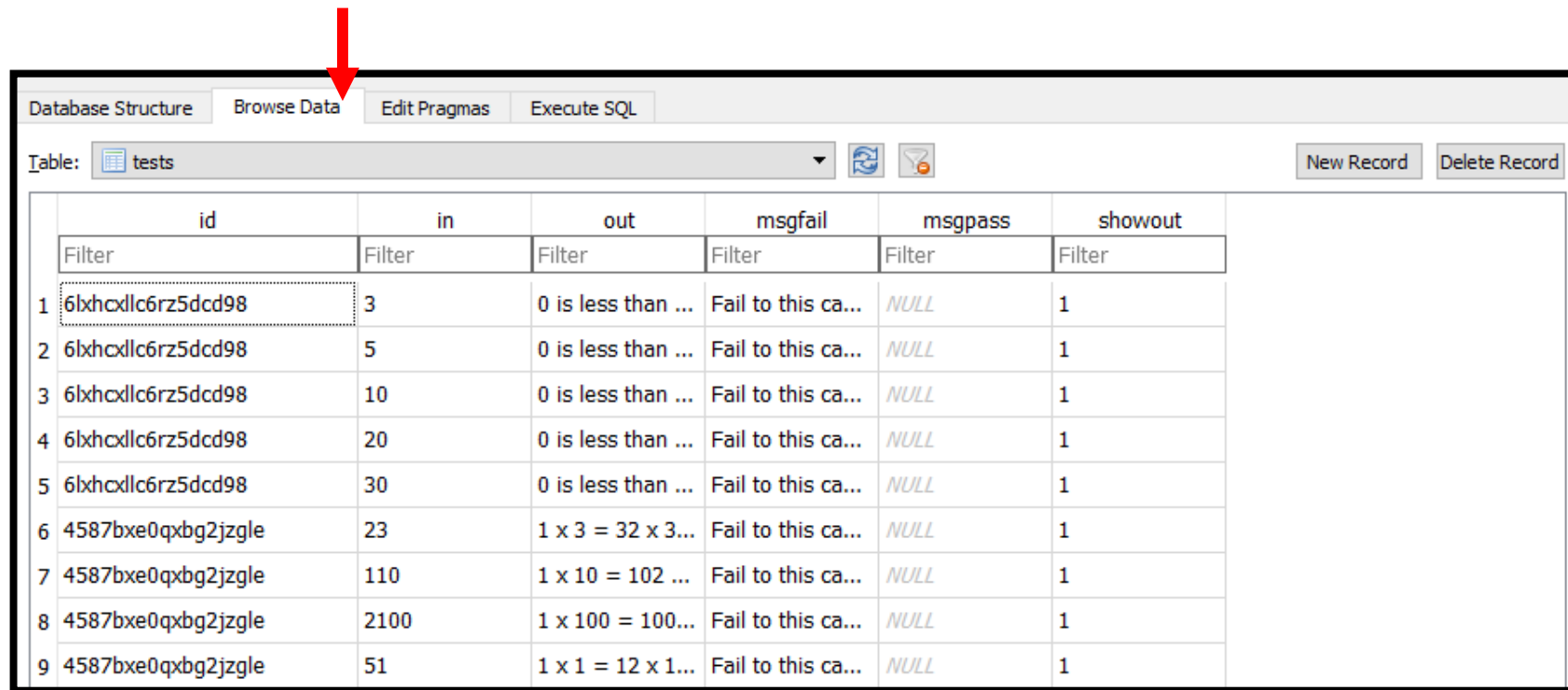
# Create a tests cases [2/]

2. Download **judge42.db** from GitHub repo of **judge42** project
  - <https://github.com/jfmunozf/judge42>
3. Open DB Browser for SQLite
  - Click File
  - Click Open Database...
4. Open database **judge42.db**

# Create a tests cases [2/]

## 5. Clic on **Browse Data** tab

- You can (and want) delete demo tests cases in a database



Database Structure Browse Data Edit Pragmas Execute SQL

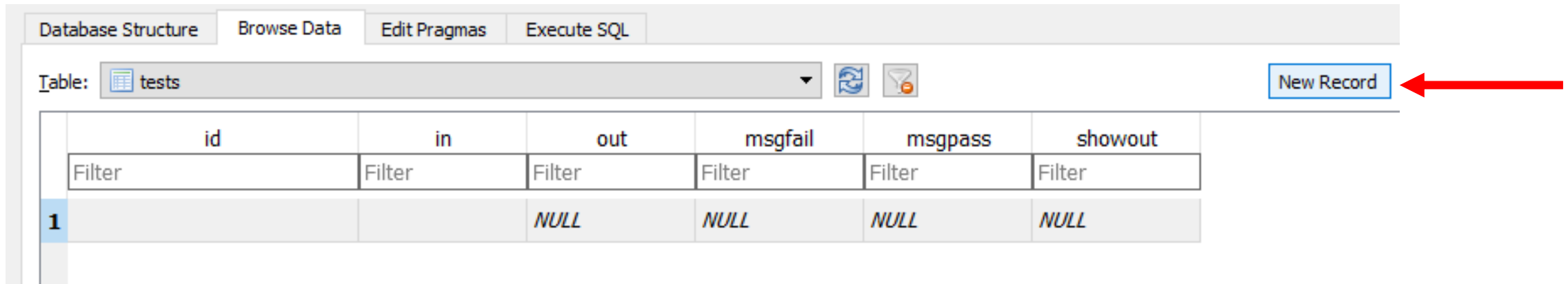
Table: tests

New Record Delete Record

	id	in	out	msgfail	msgpass	showout
	Filter	Filter	Filter	Filter	Filter	Filter
1	6lxhcxllc6rz5dcd98	3	0 is less than ...	Fail to this ca...	NULL	1
2	6lxhcxllc6rz5dcd98	5	0 is less than ...	Fail to this ca...	NULL	1
3	6lxhcxllc6rz5dcd98	10	0 is less than ...	Fail to this ca...	NULL	1
4	6lxhcxllc6rz5dcd98	20	0 is less than ...	Fail to this ca...	NULL	1
5	6lxhcxllc6rz5dcd98	30	0 is less than ...	Fail to this ca...	NULL	1
6	4587bxex0qxbg2jzgle	23	1 x 3 = 32 x 3...	Fail to this ca...	NULL	1
7	4587bxex0qxbg2jzgle	110	1 x 10 = 102 ...	Fail to this ca...	NULL	1
8	4587bxex0qxbg2jzgle	2100	1 x 100 = 100...	Fail to this ca...	NULL	1
9	4587bxex0qxbg2jzgle	51	1 x 1 = 12 x 1...	Fail to this ca...	NULL	1

# Create a tests cases [2/]

6. Begin with an empty **tests** table and clic in **New Record** to create a new blank record in a **tests** table.



The screenshot shows a database management interface with the following components:

- Navigation tabs: Database Structure, Browse Data, Edit Pragmas, Execute SQL.
- Table selection: A dropdown menu labeled "Table:" with "tests" selected.
- Actions: Two icons (refresh and filter) and a "New Record" button.
- Table structure: A table with columns: id, in, out, msgfail, msgpass, showout.
- Table data: The table contains one row with the following values: id (Filter), in (Filter), out (Filter), msgfail (Filter), msgpass (Filter), showout (Filter).
- Row selection: The first row is highlighted with a blue background.

A red arrow points to the "New Record" button.

# Database fields description

- **id**: Unique ID of this problem statement. Every problem statement can contains multiple tests cases, every test case is a record in a table. You could record multiple tests cases for multiple problem statements.
- **in**: Inputs of the this test case
- **out**: Expected outputs for this test case
- **msgfail**: A message that show to student when this test case fails.
- **msgpass**: A message that show to student when this test case pass.
- **showout**
  - **1**: Show expected output for this case when this test case fails.
  - **NULL**: Don't show expected output for this case when this test case fails.



# Create a tests cases [2/]

7. Create a first test case according to Inputs and Outputs examples in a problem statement at the beginning of this document.
- To record first test case use right panel of DB Browser for SQLite to edit cell. For example, to record Inputs of the first test case:
    - Clic on **in** cell
    - Type data: one line for each input according to problem statement
    - Clic **Apply** button to set this entry in cell.

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragmas Execute SQL

Table: tests

	id	in	out	msgfail	msgpass	showout
	Filter	Filter	Filter	Filter	Filter	Filter
1	Problem1				NULL	NULL

New Record Delete Record

Edit Database Cell

Mode: Text

Import Export Set as NULL

2  
3

Type of data currently in cell: Text / Numeric  
3 char(s)

Apply

Clic on **in** cell

Clic **Apply**

Write inputs of this test case: one input per line according to problem statement. No more breaklines at the end: just two lines.

# Create a tests cases [2/]

8. Example of outputs for this test case: just two lines according to (format) problem statement

The screenshot shows a database management interface. On the left, a table named 'tests' is displayed with the following columns: id, in, out, msgfail, msgpass, and showout. The first row of data shows 'Problem1' in the 'id' column, '23' in the 'in' column, and NULL values in the 'out', 'msgpass', and 'showout' columns. On the right, an 'Edit Database Cell' dialog is open, showing the text '2 x 3 = 6' and '2 + 3 = 5' in a text input field. The dialog also includes buttons for 'Import', 'Export', 'Set as NULL', and 'Apply'.

	id	in	out	msgfail	msgpass	showout
	Filter	Filter	Filter	Filter	Filter	Filter
1	Problem1	23			NULL	NULL

Table: tests

Mode: Text

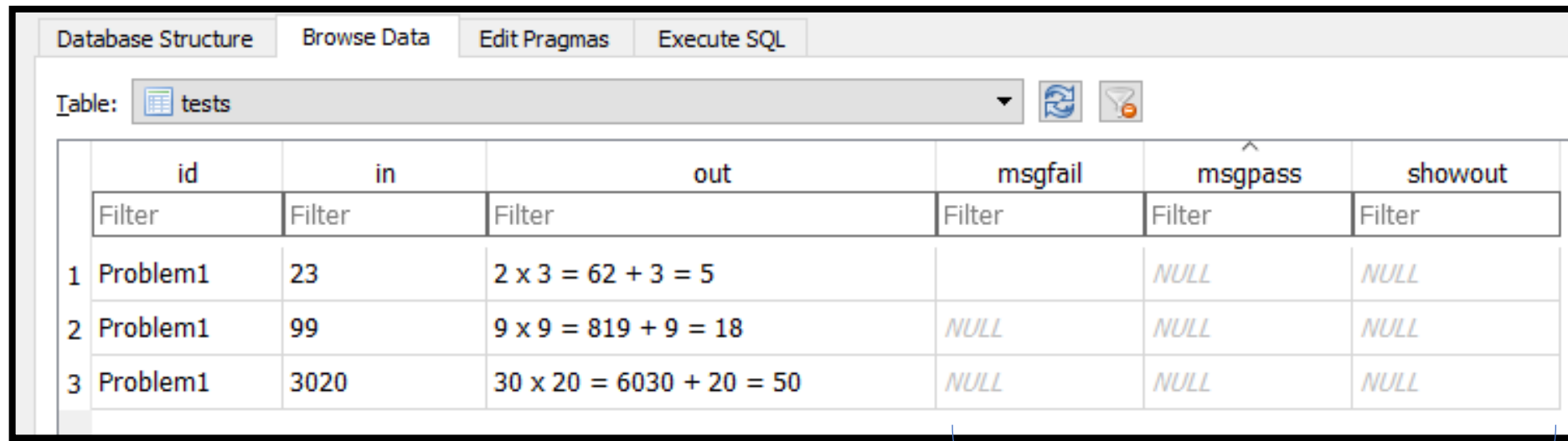
2 x 3 = 6  
2 + 3 = 5

Type of data currently in cell: Text / Numeric  
19 char(s)

Apply

# Create a tests cases [2/]

- At the end, the three test cases for problem statement:



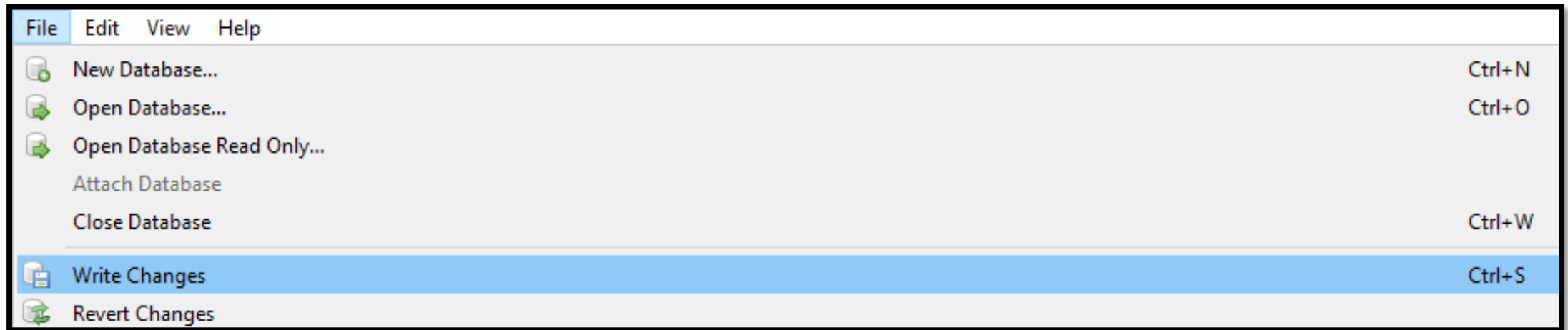
The screenshot shows a database management interface with tabs for 'Database Structure', 'Browse Data', 'Edit Pragmas', and 'Execute SQL'. The 'Table: tests' is selected, and its data is displayed in a table. The table has six columns: 'id', 'in', 'out', 'msgfail', 'msgpass', and 'showout'. The first row is a header with 'Filter' in each column. The next three rows represent test cases for 'Problem1'.

	id	in	out	msgfail	msgpass	showout
	Filter	Filter	Filter	Filter	Filter	Filter
1	Problem1	23	$2 \times 3 = 6$ $2 + 3 = 5$		NULL	NULL
2	Problem1	99	$9 \times 9 = 81$ $9 + 9 = 18$	NULL	NULL	NULL
3	Problem1	3020	$30 \times 20 = 600$ $30 + 20 = 50$	NULL	NULL	NULL

**This fields are optional**

# Write changes to a database

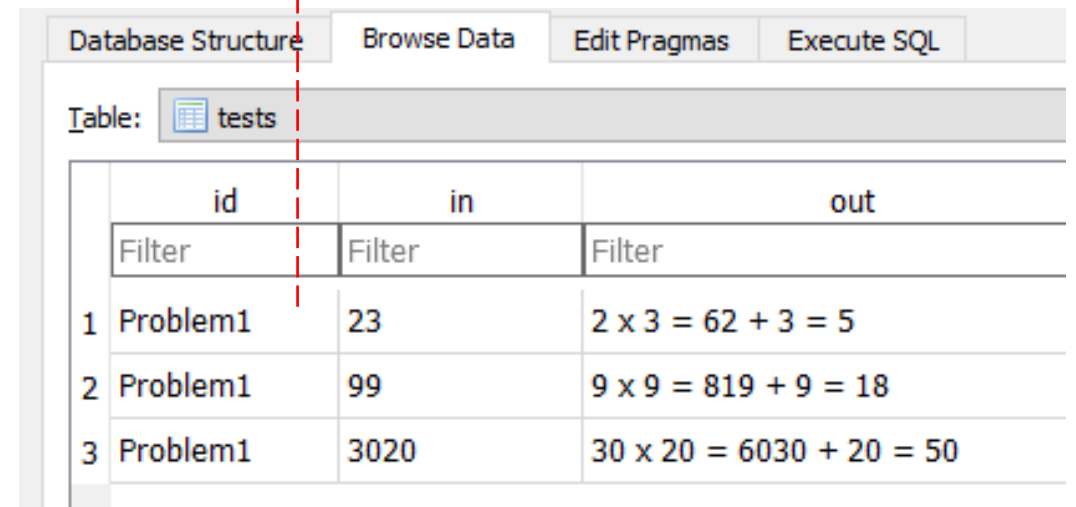
- Click on File
- Select Write Changes or CTRL + S



# Thats all

- Deploy **judge42.py** and **judge42.db**
- In this example, first line of code to test with this tests cases must be (exacly)

**# \_\_judge42\_\_id\_\_ = Problem1**



Database Structure   Browse Data   Edit Pragmas   Execute SQL			
Table:	tests		
	id	in	out
	Filter	Filter	Filter
1	Problem1	23	$2 \times 3 = 6$ $2 + 3 = 5$
2	Problem1	99	$9 \times 9 = 81$ $9 + 9 = 18$
3	Problem1	3020	$30 \times 20 = 600$ $30 + 20 = 50$

# Deployments examples

- <https://github.com/jfmunozf/judge42>
  - Google Colab
  - Replit in a Console application (fork the deployment to your account)
  - Replit in a Web application (fork the deployment to your account)
  - Python anywhere in a Web application

<https://github.com/jfmunozf/judge42>

Juan Felipe Muñoz Fernández