

Un pasito más SQL

La prueba que la página web <https://sqlpd.com/> es bastante limitada y en el momento en el que quieres avanzar para aprender algo más avanzado que las primeras 11 misiones. Tras hablar con el compañero Jesus Diego (<https://github.com/Txus77/>) en la evaluación de la tarea, me recomendó seguir aprendiendo en la plataforma [Khan Academy](#) y así lo hice.

Descripción de la plataforma

Estos cursos siguen la siguiente dinámica:

A-Secuencia de:

-Vídeo explicativo

-Tarea

B-Proyecto final

Los vídeos incluyen subtítulos y se pueden ver en YouTube mientras estás realizando las tareas. Es muy cómodo.

Además, la herramienta facilita la tarea, ya que, hay un asistente que te va diciendo si lo que estás planteando está mal, incluso incluyendo pistas en cada paso.

The screenshot displays the Khan Academy SQL challenge interface. On the left, a code editor shows the following SQL commands:

```
1 CREATE TABLE books (id INTEGER PRIMARY KEY, name TEXT
2 , rating INTEGER);
3
4 INSERT INTO books VALUES (1, "Dialogue", 7) ;
5 INSERT INTO books VALUES (2, "Test Business Ideas",
6 7.5);
7 INSERT INTO books VALUES (3, "The Mondragon Myth",
8 8);
```

On the right, the 'DATABASE SCHEMA' section shows a table named 'books' with 3 rows. The schema is defined as follows:

books	3 rows
id (PK)	INTEGER
name	TEXT
rating	INTEGER

Below the schema, a congratulatory message reads: 'Congratulations! You earned 1050 points!' accompanied by a cartoon character wearing sunglasses. At the bottom, there are buttons for 'Undo', 'Start over', and a green smiley face icon. A progress indicator shows 'Step 1/2' with a green bar, and a 'Spin-off' button is visible on the right.

Figura 1. Recogido de: <https://www.khanacademy.org/computing/computer-programming/sql/sql-basics/pc/challenge-book-list-database>

A esto hay que añadirle que, en el proyecto final, hay una autoevaluación, ya que, como el reto es tan amplio, debes ir por tu parte viendo si cumples los requisitos que te piden:

Spin-off of "Project: Data dig"

We've curated a set of interesting data sets for you: [NASA astronauts](#), [Superbowl results](#), [Pokemon stats](#), [NBA players](#), [Top movies](#), [Top countries by population](#), [Solar system objects by size](#), [Marvel characters](#), [Furniture store sales](#), [Earned KA badges](#), [Winston's donut logs](#), [Card game results](#), and [NFL draft picks](#).

Pick one of those data sets or create a data set like that, and use advanced `SELECT` queries to discover things about the data. What sort of questions might one have about that data, like if they were using it for an app or a business idea? Here are some ideas:

- What are average, max, and min values in the data?
- What about those numbers per category in the data (using `HAVING`)?
- What ways are there to group the data values that don't exist yet (using `CASE`)?
- What interesting ways are there to filter the data (using `AND/OR`)?

Does your project meet these goals?

1. Queries a data set with at least 20 rows.
2. Has at least one query that uses `AND` or `OR`.
3. Has at least one query that uses `GROUP BY` with either `CASE` or `HAVING`.
4. Has at least one query that uses a math function like `MAX`, `MIN`, `AVG`, `SUM`, or `ROUND`.
5. Contains no syntax or logic errors.

Figura 2. Recogida de: <https://www.khanacademy.org/computing/computer-programming/sql/more-advanced-sql-queries/pp/project-data-dig>

Cursos realizados

1. [SQL basics](#)

The screenshot shows the Khan Academy interface for a SQL challenge. On the left is a sidebar with a list of challenges: 'Challenge: Book list database', 'Querying the table', 'Challenge: Box office hits database', 'Aggregating data', 'Challenge: TODO list database stats' (which is highlighted), 'S-Q-L or SEQUEL?', and 'Project: Design a store database'. The main area is titled 'Step 1' and contains the following text: 'Here's a table containing a TODO list with the number of minutes it will take to complete each item. Insert another item to your todo list with the estimated minutes it will take.' Below this text is a SQL editor with the following code:

```
1 CREATE TABLE todo_list (id INTEGER PRIMARY KEY, item
2 TEXT, minutes INTEGER);
3 INSERT INTO todo_list VALUES (1, "Wash the dishes",
4 15);
5 INSERT INTO todo_list VALUES (2, "vacuuming", 20);
6 INSERT INTO todo_list VALUES (3, "Learn some stuff on
7 KA", 30);
8 INSERT INTO todo_list VALUES (4, "Climb a hill", 40);
9 SELECT SUM (minutes) FROM todo_list
```

To the right of the editor is a 'DATABASE SCHEMA' section showing a table named 'todo_list' with 4 rows. The schema is as follows:

id (PK)	INTEGER
item	TEXT
minutes	INTEGER

Below the schema is a 'QUERY RESULTS' section showing the result of the `SUM (minutes)` query: 105. At the bottom right, there is a congratulatory message: 'Congratulations! You earned 2100 points!' with a cartoon character wearing sunglasses. At the bottom of the interface are buttons for 'Undo', 'Start over', and 'Spin-off', along with a progress indicator 'Step 1/2'.

Figura 3. Recogida de: <https://www.khanacademy.org/computing/sql/sql-basics/pc/challenge-todo-list-database-stats>

2. More advanced SQL queries

Computing > Computer programming > Intro to SQL: Querying and managing data > More advanced SQL queries

Restricting grouped results with HAVING

Challenge: The wordiest author

Who issues SQL queries?

Calculating results with CASE

Challenge: Gradebook

Project: Data dig

Next lesson
Relational queries in SQL

We've created a database to track student grades, with their name, number grade, and what percent of activities they've completed. In this first step, select all of the rows, and display the name, number_grade, and percent_completed, which you can compute by multiplying and rounding the fraction_completed column.

```
1 CREATE TABLE student_grades (  
2   id INTEGER PRIMARY KEY AUTOINCREMENT,  
3   name TEXT,  
4   number_grade INTEGER,  
5   fraction_completed REAL);  
6  
7 INSERT INTO student_grades (name, number_grade,  
8   fraction_completed)  
9   VALUES ("Winston", 90, 0.805);  
10 INSERT INTO student_grades (name, number_grade,  
11   fraction_completed)  
12   VALUES ("Winnefer", 95, 0.901);  
13 INSERT INTO student_grades (name, number_grade,  
14   fraction_completed)  
15   VALUES ("Winsteen", 85, 0.906);  
16 INSERT INTO student_grades (name, number_grade,  
17   fraction_completed)  
18   VALUES ("Wincifer", 66, 0.7054);  
19 INSERT INTO student_grades (name, number_grade,  
20   fraction_completed)  
21   VALUES ("Winsten", 76, 0.5013);  
22 INSERT INTO student_grades (name, number_grade,  
23   fraction_completed)  
24   VALUES ("Winstonia", 82, 0.9045);
```

DATABASE SCHEMA

student_grades	6 rows
id (PK)	INTEGER
name	TEXT
number_grade	INTEGER
fraction_completed	REAL

QUERY RESULTS

name	number_grade	percent_completed
Winston	90	81
Winnefer	Congratulations! You earned 2100 points!	
Winsteen		
Wincifer		
Winsten		
Winstonia		

Undo Start over

Step 1/2 Spin-off

Figure 4. <https://www.khanacademy.org/computing/computer-programming/sql/more-advanced-sql-queries/pc/challenge-gradebook>