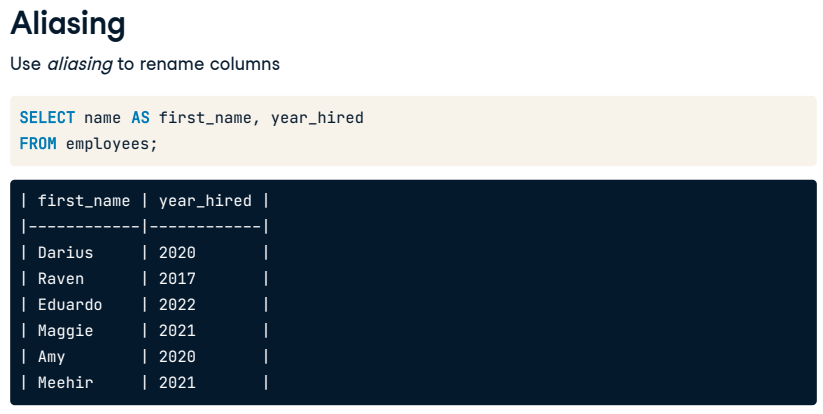
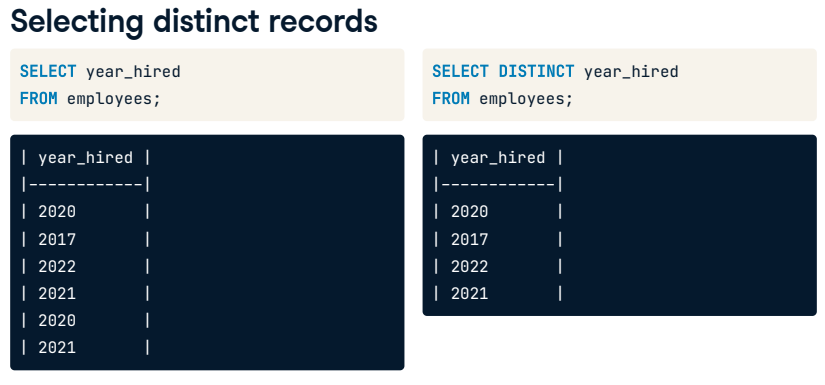
**1. Writing queries**

It's time to level up on our SQL queries by learning a few more commonly used keywords. Let's dive in.

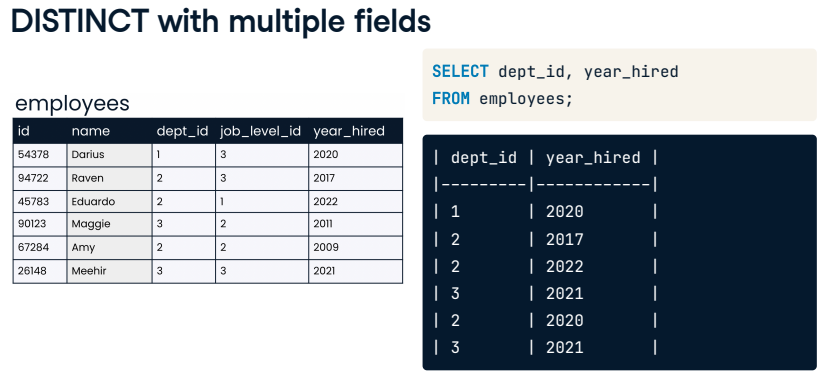
**2. Aliasing**

Sometimes it can be helpful to rename columns in our result set, whether for clarity or brevity. We can do this using aliasing. Perhaps we'd like to select the name and hire year for each record in the employees table. We could alias the name column as first\_name in the query by adding the AS keyword to indicate an alias of first\_name after selecting the name field. The result set now has first\_name rather than name as the column header. The alias only applies to the result of this particular query; in other words, the field name in the employees table itself is still name rather than first\_name.

**3. Selecting distinct records**

Some SQL questions require a way to return a list of unique values. Let's imagine that we are interested in getting a list of years in which we hired our current employees. If we select the year\_hired field from the employees table, the result set shows several years listed twice, which isn't what we are looking for. To get a list of years with no repeat values, we can add the DISTINCT keyword before the year\_hired field name in the SELECT statement. Now, we can see that all of our employees were hired in just four different years.

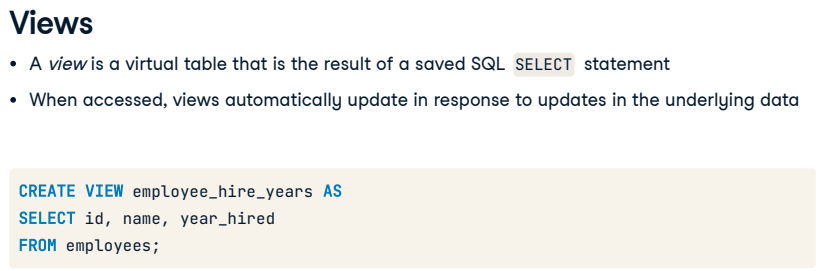
**4. DISTINCT with multiple fields**

It's possible to return the unique combinations of multiple field values by listing multiple fields after the DISTINCT keyword. Take a look at the employees table. Perhaps we'd like to know the years that different departments hired employees. We could use this SQL query to look at this information, selecting the dept\_id and year\_hired from the employees table. Looking at the results, we see that department three hired two employees in 2021.

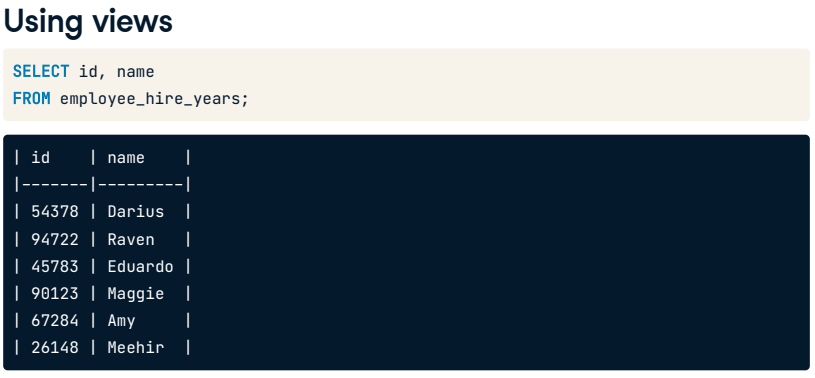
**5. DISTINCT with multiple fields**

To avoid repeating this information, we could add the DISTINCT keyword before the fields to select. Notice that the department id and year\_hired fields still have repeat values individually, but none of the records are the same: they are all unique combinations of the two fields.

**6. Views**

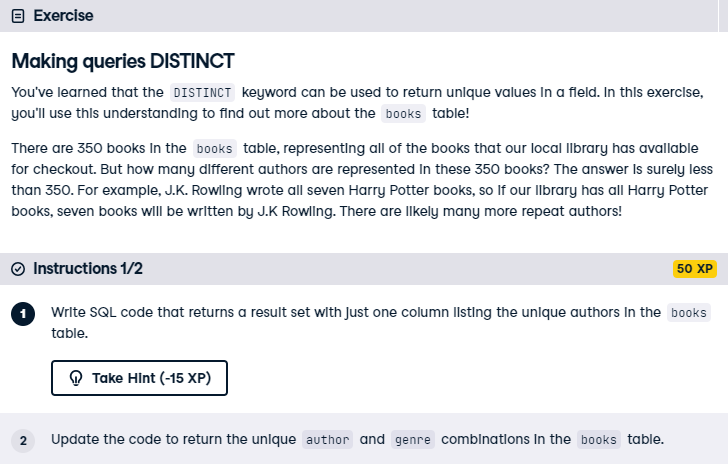
Finally, let's discuss saving SQL result sets. In SQL, a view refers to a table that is the result of a saved SQL SELECT statement. Views are considered virtual tables, which means that the data a view contains is not generally stored in the database. Rather, it is the query code that is stored for future use. A benefit of this is that whenever the view is accessed, it automatically updates the query results to account for any updates to the underlying database. To create a view, we'll add a line of code before the SELECT statement: CREATE VIEW, then the name we'd like for the new view, then the AS keyword to assign the results of the query to the new view name. Here, we create a view called employee\_hire\_years by assigning the results of a query selecting three fields from the employees table to a new view. There is no result set when creating a view.

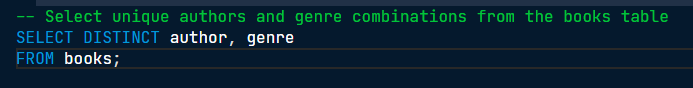
**7. Using views**

Once a view is created, however, we can query it just as we would a normal table by selecting FROM the view.

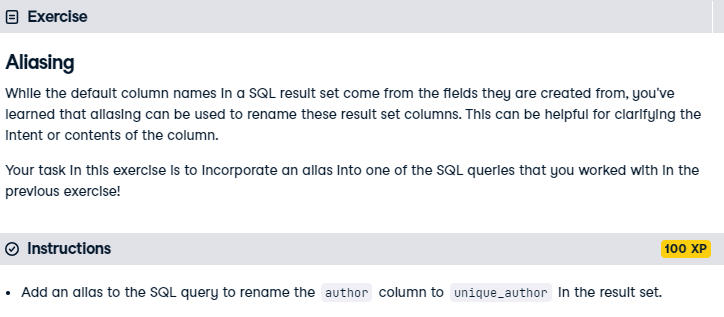
**8. Let's practice!**

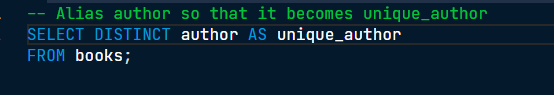
Time to practice refining and saving queries with these new keywords!



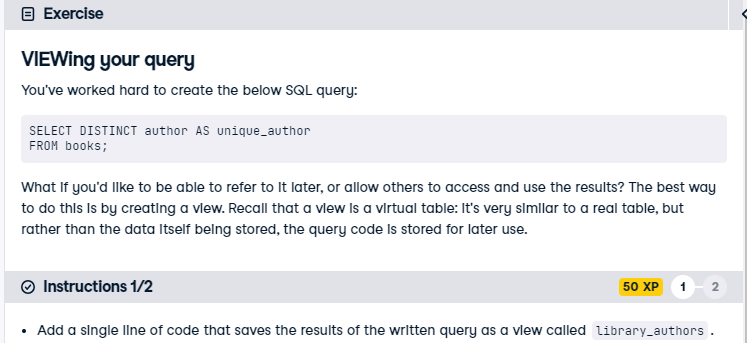


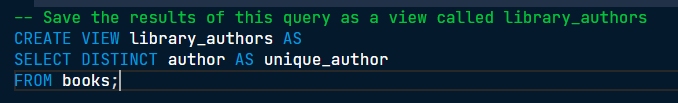
You've passed this exercise with DISTINCTion! Notice that you found 247 unique authors in the books table overall but 249 unique combinations of authors and genres. This means there are one or two authors who have written books in multiple genres!

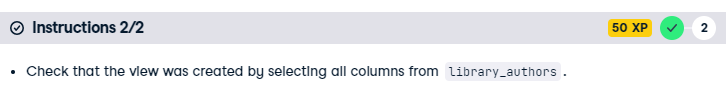


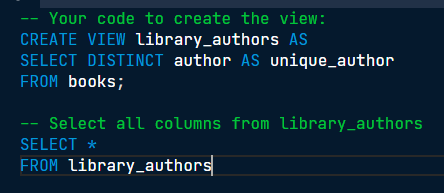


It's AS easy AS that! Great work. The alias you just implemented makes it clear that only unique authors are listed in the results and that there are no duplicates. This is clear even to someone who is reading only the result set and does not know the SQL code behind the results.









Amazing! As your SQL queries become long and complex, you'll want to be able to save your queries for referencing later. Views can also be useful when the information contained in a database table isn't quite what you need. You can create your own custom view with exactly the information you are looking for, without needing to edit the database itself, which you may not have permission to do. Creating views is a valuable skill to have, and you've mastered it!