Part 1

A screenshot of a computer

Description automatically generated

Completed sections of SoloLearn

Part 2

**Step 2A**

[SELECT](http://localhost/phpMyAdmin/url.php?url=https://dev.mysql.com/doc/refman/5.7/en/select.html) customers.last\_name FROM customers WHERE customers.last\_name [LIKE](http://localhost/phpMyAdmin/url.php?url=https://dev.mysql.com/doc/refman/5.7/en/string-comparison-functions.html%23operator_like) '%Smith%' UNION ALL [SELECT](http://localhost/phpMyAdmin/url.php?url=https://dev.mysql.com/doc/refman/5.7/en/select.html) customers.last\_name FROM customers WHERE customers.last\_name [LIKE](http://localhost/phpMyAdmin/url.php?url=https://dev.mysql.com/doc/refman/5.7/en/string-comparison-functions.html%23operator_like) '%Hayes%';

A screenshot of a computer

Description automatically generated

Shows the result of combining two SELECT statements with a union.

**Step 2B**

[SELECT](http://localhost/phpMyAdmin/url.php?url=https://dev.mysql.com/doc/refman/5.7/en/select.html) customers.\* FROM customers INNER JOIN sale ON customers.customer\_numb = sale.customer\_numb WHERE sale.sale\_date <= '01-Aug-2021';

**A screenshot of a computer screen

Description automatically generated**

Can not figure out how to compare the dates without the month being a integer.

**Step 2C**

[SELECT](http://localhost/phpMyAdmin/url.php?url=https://dev.mysql.com/doc/refman/5.7/en/select.html) sale\_id, sale\_total\_amt, (sale\_total\_amt \* 0.1) as tax, (sale\_total\_amt \* 0.1 + sale\_total\_amt) / 12 as payment FROM sale;

**A screenshot of a computer

Description automatically generated**

Shows the result of calculating tax and then a monthly interest-free payment.

**Step 4C**

**A screenshot of a computer

Description automatically generated**

Shows the full table scan

A screenshot of a computer

Description automatically generated

Not sure if this is how you do use the created index, but here it is

Part 3

**Step 13**

**A screenshot of a computer

Description automatically generated**

Shows the WHERE statement

**Wildcard Screenshot 1**

**A screenshot of a computer

Description automatically generated**

Shows the result of using a wildcard to find all album titles that start with “A”.

**Wildcard Screenshot 2**

**A screenshot of a computer

Description automatically generated**

Shows the result of using a wildcard to find albums that start with “Y”.

Part 4

**Step 1 Screenshot**

**A screenshot of a computer

Description automatically generated**

Shows the successful loading of database items into WinForms.

**Step 2**

**A screenshot of a computer screen

Description automatically generated**

Shows the addition of a new album, with the image displayed as proof. Note, some of the images have the “copyright” thing attached to them so it crashes the program. Did not catch it when we first went through them all.

**Step 3**

A screenshot of a computer

Description automatically generated

Album art for Post Malone.

**Step 4**

**A screenshot of a computer

Description automatically generated**

Shows the search result of album titles that have the character “n”.

**Key Concepts Learned**

Learned wildcards, unions, and adding a new column for displaying “calculation” query results. Most exciting part was connecting a WinForms application to our album database. Definitely beats using .txt files!