

Overview [Lab work and Homework]

As discussed in the lecture we will start off in the module by presenting you with an opportunity to get setup with some of the basic tooling which will be used for the module. In this lab we will introduce some of our basic **tech stack** and begin exploring some basic use of the various elements of this tech stack.

Many of the exercises below are intended to prompt both **thinking** and **discussion**, don't just be satisfied with a **surface level of understanding** talk to your tutors in class and share your thoughts and ideas with us!

This tutorial makes use of the XAMPP software package, we recommend using a [portable version](#) of this software package The lab tutorial forms both the lab work and homework for the week 4 activities.¹

Objectives

- Familiarise ourselves with how to link MySQL/MariaDB with the server-side scripting system using mysqli
- Further familiarise ourselves with some basic SQL querying and consider how to use the results provided by said queries in our PHP scripts

¹ An additional homework exercise is also found as an extension task at the end of the week 1 lab talk / demo slides

Additional information: Tracking your learning progress.

To help track your progress early in the module, we will be using a little system in these first few weeks to **help you track your understanding so far**. This will give you an opportunity to show how you are getting on with understanding the material each week during the early stages of the module so we can **better guide you through developing your knowledge and understanding**. When attempting your answer also be honest and discuss with **us when you needed to do some additional background research²**, and **show us where you got your information from³**.

Progress checker

The progress checker is included at the end of our various exercises to allow you to keep a record of your learning, giving you an opportunity to quickly show us your progress and discuss in class during the following week. It also encourages you to keep a record of this progress to help you to reflect on your developing skills and understanding during the module.

Example "Progress checker"

Please select the answer you feel **best describes your level of understanding** for each **exercise** (an example progress checker can be seen in **figure 1**).

| Now let's check how well you have understood so far! 😊 | |
|--|----------------|
| How did you find attempting this exercise? | Choose one (X) |
| 😊 I'm very confident that I have understood and completed this task successfully | X |
| 🙂 I'm fairly confident that I have understood and completed this task successfully | |
| 😐 I think I'm getting there but I need to work on understanding this a little more | |
| 😓 I got a bit stuck and I will need to work on understanding this a lot more | |

Figure 1 An example 'progress checker'

² So we can keep track of whether the answer is primarily derived through your own existing understanding, or based on reviewing the work of someone else – this is not a negative at all when you are starting to learn a new tool or concept! Experienced developers started building their knowledge from somewhere, but you want to clearly reflect on how you are building those skills and knowledge and where you are getting it from

³ Referencing is very important, but a simple url, etc is fine for the lab content

Food for thought

The “Food for thought” messages boxes provide some prompts for you to work on deepening your understanding as you progress through the labs a good Computer Scientist / Data Scientist / Software Engineer recognises the value of picking apart the tools and concepts they are learning about to deepen their understanding. Fundamentally, we aim for this module to not just be about you “Developing Database-backed Web Applications” but about refining your approach to problem solving in Software Development, to help you become a more well-rounded Engineer 😊




To help this process along we will use the “Food for thought” message boxes to provide some suggested starting points to investigate various aspects of what you are doing in the labs further, moving you away from approaching development as simply “**remembering ‘recipes’ from a cookbook**” to “**understanding what tools/techniques/concepts to apply**”, “**when to apply them**”, “**how to apply them**” and “**why we tend to apply them in a particular way**”



Figure 2 The Swedish Chef (By Disney.com, Fair use, <https://en.wikipedia.org/w/index.php?curid=40291413>)

Think about it like this: developers are a bit like chefs, we don’t just memorise and follow a bunch of “recipes” (example programs) from someone else’s “cookbook” (tutorial / documentation / example code) we are experienced enough to understand **how** those recipes work and **why** they work so we can use that understanding to develop our own recipes and serve up our own tasty dishes to our customers (our own software applications)

Example “Food for thought”

| | |
|---|-------------------------|
|    | Food for thought |
| Let’s deepen our understanding by thinking a little deeper about this exercise and discussing our thoughts in class. | |
| We will have a sample prompt to think about / experiment with some aspect of the lab in this bit usually. | |

Be an active learner. Help us help you to grow.

As well as recording your progress with the lab work **so far**⁴ and discussing with us during the following week, please make sure you tell us **in class**⁵ about any aspects of the current week's work you are particularly struggling to understand. From what we have seen **so far** based on your progress at Level 4 many of you are still working your way up from the **lower levels** in your **understanding of programming / software development**. To turn you into an effective **Software Engineer / Software Developer** we need you to **work with us** to move away from those **basic, lower levels of understanding**, which will require **you** to work on **actively developing your skills** (with our guidance along the way 😊) to reach those **higher levels**⁶.

We want you to move from:

- **Memorising** recipes / definitions

By

- **Understanding** the tools, techniques and concepts underpinning our examples / exercises

Then working on

- **Applying** these tools, techniques, and concepts. Experimenting with them to gain deeper understanding.

So that you can eventually

- **Create** your own original software solutions through an informed approach to planning, designing, and implementing software.
- **Analyse** the different approaches (e.g tools, techniques and concepts) that could be applied to help solve a problem and how best to use them.
- **Evaluate** your approach (and others) to help develop your better judgment when creating software.

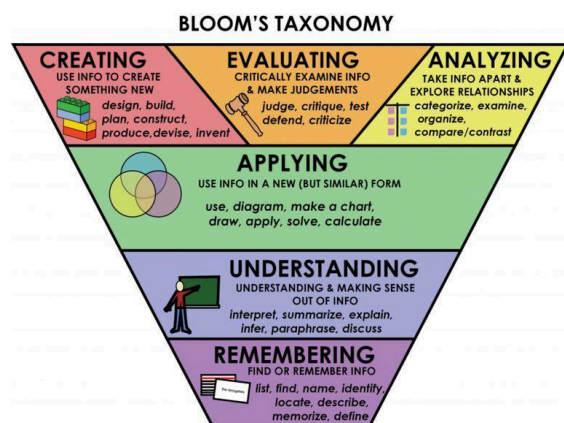


Figure 3 The different levels of learning, to discuss in class (ThoughtCo, 2019)

⁴ If you need to work on it further then please show us what you have done so far and then continue to work on it further, this allows us to ‘temperature check’ your progress as a group.

⁵ The earlier you tell us, the better 😊 Get our attention and discuss what you need help with so we can advise you

⁶ That’s where the **fun** stuff (and the **money**) is when it comes to Software Development 😊

Week 3 – Constructing Databases and PHP Scripting

Before attempting the below lab exercises ensure that you have set up an XAMPP instance capable of running the Apache web server and MySQL on the device you are using, work with your lab tutor to familiarise yourself with this process – they will likely demo this process as part of the introductory lab talk.

Exercise 1

You have been asked to create a website database similar to Amazon that stores information about books.

The information you need to store is:

- Title of book
- Author
- ISBN number (10)
- Publisher
- Year of publication
- Price
- Genre
- Type (hardcover or kindle)
- Customer reviews (5-star system)

Your task is to plan out a relational database with table. You first should **decide on the tables you would require** and **how these tables would be linked with one another** using primary and secondary keys.



Food for thought

Let's deepen our understanding by thinking a little deeper about this exercise and discussing our thoughts in class.

While we haven't introduced formal database design using ERD / ERM modelling yet, we can still think about the basics by considering what data we need to represent and why we need to represent it. Why are establishing relationships between different tables (relations) in a relational database important? What benefits does this provide us with?

My solution

Record your notes about how you attempted to answer the question / solve the exercise in this section (just edit the text bo


✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0008 seconds.)

`SELECT * FROM `books``

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

| BookID | Title | Author | ISBN | Publisher | YearOfPublication | Price | Genre | Type | AvgCustomerRating | TotalReviews |
|--------|-------|--------|------|-----------|-------------------|-------|-------|------|-------------------|--------------|
|--------|-------|--------|------|-----------|-------------------|-------|-------|------|-------------------|--------------|

Query results operations

 Create view

```

1 CREATE TABLE books (
2     BookID INT AUTO_INCREMENT PRIMARY KEY,
3     Title VARCHAR(255) NOT NULL,
4     Author VARCHAR(255) NOT NULL,
5     ISBN VARCHAR(20) UNIQUE,
6     Publisher VARCHAR(255),
7     YearOfPublication YEAR,
8     Price DECIMAL(10, 2),
9     Genre VARCHAR(100),
10    Type VARCHAR(100),
11    AvgCustomerRating DECIMAL(3, 2),
12    TotalReviews INT
13 );

```

x 😊)

Now let's check how well you have understood so far! 😊

| How did you find attempting this exercise? | Choose one (X) |
|--|----------------|
| 😊 I'm very confident that I have understood and completed this task successfully | |
| 😊 I'm fairly confident that I have understood and completed this task successfully | |
| 😐 I think I'm getting there but I need to work on understanding this a little more | |
| 😓 I got a bit stuck and I will need to work on understanding this a lot more | |

Exercise 2

You should then populate the database with at least 20 records with books from different authors, publishers, and genre. Ensure you have more than 2 books for each.

My solution 🧐

Record your notes about how you attempted to answer the question / solve the exercise in this section (just edit the text box 😊)

| SELECT * FROM `books` | | | | | | | | | | | |
|---|----------|---------------------------------------|-------------------|------------|--------------------|-------------------|-------|-----------------|-----------|-------------------|--------------|
| <input type="checkbox"/> Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh] | | | | | | | | | | | |
| <input type="checkbox"/> Show all Number of rows: 25 Filter rows: Search this table Sort by key: None | | | | | | | | | | | |
| Extra options | | | | | | | | | | | |
| ← T → | ▼ BookID | Title | Author | ISBN | Publisher | YearOfPublication | Price | Genre | Type | AvgCustomerRating | TotalReviews |
| <input type="checkbox"/> Edit Copy Delete | 1 | Atomic Habits | James Clear | 1234567890 | Penguin Books | 2023 | 19.99 | Self-help | Hardcover | 4.8 | 125 |
| <input type="checkbox"/> Edit Copy Delete | 2 | The Midnight Library | Matt Haig | 1234567891 | Canongate Books | 2022 | 15.49 | Fiction | Hardcover | 4.5 | 87 |
| <input type="checkbox"/> Edit Copy Delete | 3 | The Alchemist | Paulo Coelho | 1234567892 | HarperOne | 2022 | 12.99 | Fiction | Kindle | 4.7 | 113 |
| <input type="checkbox"/> Edit Copy Delete | 4 | Becoming | Michelle Obama | 1234567893 | Crown Publishing | 2021 | 22.99 | Memoir | Hardcover | 4.9 | 158 |
| <input type="checkbox"/> Edit Copy Delete | 5 | Educated | Tara Westover | 1234567894 | Random House | 2020 | 18.50 | Memoir | Hardcover | 4.6 | 134 |
| <input type="checkbox"/> Edit Copy Delete | 6 | Where the Crawdads Sing | Delia Owens | 1234567895 | G.P. Putnam's Sons | 2019 | 14.99 | Fiction | Kindle | 4.4 | 102 |
| <input type="checkbox"/> Edit Copy Delete | 7 | Dune | Frank Herbert | 1234567896 | Ace Books | 2021 | 10.99 | Science Fiction | Hardcover | 4.8 | 92 |
| <input type="checkbox"/> Edit Copy Delete | 8 | The Silent Patient | Alex Michaelides | 1234567897 | Celadon Books | 2023 | 19.95 | Thriller | Hardcover | 4.6 | 98 |
| <input type="checkbox"/> Edit Copy Delete | 9 | A Promised Land | Barack Obama | 1234567898 | Crown Publishing | 2021 | 24.50 | Memoir | Hardcover | 4.7 | 145 |
| <input type="checkbox"/> Edit Copy Delete | 10 | It Ends with Us | Colleen Hoover | 1234567899 | Atria Books | 2022 | 16.00 | Romance | Kindle | 4.3 | 77 |
| <input type="checkbox"/> Edit Copy Delete | 11 | 1984 | George Orwell | 1234567900 | Secker & Warburg | 2023 | 9.99 | Dystopian | Hardcover | 4.7 | 130 |
| <input type="checkbox"/> Edit Copy Delete | 12 | The Subtle Art of Not Giving a F*ck | Mark Manson | 1234567901 | Harper | 2022 | 15.49 | Self-help | Kindle | 4.4 | 89 |
| <input type="checkbox"/> Edit Copy Delete | 13 | Sapiens: A Brief History of Humankind | Yuval Noah Harari | 1234567902 | Harvill Secker | 2023 | 18.99 | History | Hardcover | 4.9 | 142 |

Now let's check how well you have understood so far! 😊

| How did you find attempting this exercise? | Choose one (X) |
|--|----------------|
| 😊 I'm very confident that I have understood and completed this task successfully | |
| 🙂 I'm fairly confident that I have understood and completed this task successfully | |
| 😐 I think I'm getting there but I need to work on understanding this a little more | |
| 😓 I got a bit stuck and I will need to work on understanding this a lot more | |

Exercise 3

Run some example queries such as:

- Displays all the data (as above)
- Displays books from each author
- Display only the title, price and author
- Display the review of books in descending order
- Display books from each publisher

My solution 🤖

Record your notes about how you attempted to answer the question / solve the exercise in this section (just edit the text box 😊)

```
SELECT * FROM `books`
```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

| | BookID | Title | Author | ISBN | Publisher | YearOfPublication | Price | Genre | Type | AvgCustomerRating | TotalReviews |
|---|--------|---------------------------------------|-------------------|------------|--------------------|-------------------|-------|-----------------|-----------|-------------------|--------------|
| <input type="checkbox"/> Edit Copy Delete | 1 | Atomic Habits | James Clear | 1234567890 | Penguin Books | 2023 | 19.99 | Self-help | Hardcover | 4.8 | 125 |
| <input type="checkbox"/> Edit Copy Delete | 2 | The Midnight Library | Matt Haig | 1234567891 | Canongate Books | 2022 | 15.49 | Fiction | Hardcover | 4.5 | 87 |
| <input type="checkbox"/> Edit Copy Delete | 3 | The Alchemist | Paulo Coelho | 1234567892 | HarperOne | 2022 | 12.99 | Fiction | Kindle | 4.7 | 113 |
| <input type="checkbox"/> Edit Copy Delete | 4 | Becoming | Michelle Obama | 1234567893 | Crown Publishing | 2021 | 22.99 | Memoir | Hardcover | 4.9 | 158 |
| <input type="checkbox"/> Edit Copy Delete | 5 | Educated | Tara Westover | 1234567894 | Random House | 2020 | 18.50 | Memoir | Hardcover | 4.6 | 134 |
| <input type="checkbox"/> Edit Copy Delete | 6 | Where the Crawdads Sing | Delia Owens | 1234567895 | G.P. Putnam's Sons | 2019 | 14.99 | Fiction | Kindle | 4.4 | 102 |
| <input type="checkbox"/> Edit Copy Delete | 7 | Dune | Frank Herbert | 1234567896 | Ace Books | 2021 | 10.99 | Science Fiction | Hardcover | 4.8 | 92 |
| <input type="checkbox"/> Edit Copy Delete | 8 | The Silent Patient | Alex Michaelides | 1234567897 | Celadon Books | 2023 | 19.95 | Thriller | Hardcover | 4.6 | 98 |
| <input type="checkbox"/> Edit Copy Delete | 9 | A Promised Land | Barack Obama | 1234567898 | Crown Publishing | 2021 | 24.50 | Memoir | Hardcover | 4.7 | 145 |
| <input type="checkbox"/> Edit Copy Delete | 10 | It Ends with Us | Colleen Hoover | 1234567899 | Atria Books | 2022 | 16.00 | Romance | Kindle | 4.3 | 77 |
| <input type="checkbox"/> Edit Copy Delete | 11 | 1984 | George Orwell | 1234567800 | Secker & Warburg | 2023 | 9.99 | Dystopian | Hardcover | 4.7 | 130 |
| <input type="checkbox"/> Edit Copy Delete | 12 | The Subtle Art of Not Giving a F*ck | Mark Manson | 1234567801 | Harper | 2022 | 15.49 | Self-help | Kindle | 4.4 | 89 |
| <input type="checkbox"/> Edit Copy Delete | 13 | Sapiens: A Brief History of Humankind | Yuval Noah Harari | 1234567802 | Harvill Secker | 2023 | 18.99 | History | Hardcover | 4.9 | 142 |

Console

```
SELECT Author, GROUP_CONCAT(Title SEPARATOR ', ') AS Books FROM Books GROUP BY Author;
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows: 25 Filter rows: Sort by key:

Extra options

| <div><div><div>←</div><div>→</div></div></div> | | | | Author | Books |
|--|---|---|---|---------------------|--|
| <input type="checkbox"/> |  |  |  | Alex Michaelides | The Silent Patient |
| <input type="checkbox"/> |  |  |  | Andy Weir | Project Hail Mary |
| <input type="checkbox"/> |  |  |  | Barack Obama | A Promised Land |
| <input type="checkbox"/> |  |  |  | Colleen Hoover | It Ends with Us, Verity |
| <input type="checkbox"/> |  |  |  | Delia Owens | Where the Crawdads Sing |
| <input type="checkbox"/> |  |  |  | Frank Herbert | Dune |
| <input type="checkbox"/> |  |  |  | George Orwell | 1984 |
| <input type="checkbox"/> |  |  |  | J.K. Rowling | Harry Potter and the Philosopher's Stone |
| <input type="checkbox"/> |  |  |  | James Clear | Atomic Habits |
| <input type="checkbox"/> |  |  |  | Madeline Miller | Circe |
| <input type="checkbox"/> |  |  |  | Mark Manson | The Subtle Art of Not Giving a F*ck |
| <input type="checkbox"/> |  |  |  | Markus Zusak | The Book Thief |
| <input type="checkbox"/> |  |  |  | Matt Haig | The Midnight Library |
| <input type="checkbox"/> |  |  |  | Michelle Obama | Becoming |
| <input type="checkbox"/> |  |  |  | Paulo Coelho | The Alchemist |
| <input type="checkbox"/> |  |  |  | Robert T. Kiyosaki | Rich Dad Poor Dad |
| <input type="checkbox"/> |  |  |  | Tara Westover | Educated |
| <input type="checkbox"/> |  |  |  | Taylor Jenkins Reid | The Seven Husbands of Evelyn Hugo |
| <input type="checkbox"/> |  |  |  | Yuval Noah Harari | Sapiens: A Brief History of Humankind |

`SELECT Title, Price, Author FROM Books;`

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows: Filter rows: Sort by key:

Extra options

| | | | | Title | Price | Author |
|--------------------------|------|------|--------|--|-------|---------------------|
| <input type="checkbox"/> | Edit | Copy | Delete | Atomic Habits | 19.99 | James Clear |
| <input type="checkbox"/> | Edit | Copy | Delete | The Midnight Library | 15.49 | Matt Haig |
| <input type="checkbox"/> | Edit | Copy | Delete | The Alchemist | 12.99 | Paulo Coelho |
| <input type="checkbox"/> | Edit | Copy | Delete | Becoming | 22.99 | Michelle Obama |
| <input type="checkbox"/> | Edit | Copy | Delete | Educated | 18.50 | Tara Westover |
| <input type="checkbox"/> | Edit | Copy | Delete | Where the Crawdads Sing | 14.99 | Delia Owens |
| <input type="checkbox"/> | Edit | Copy | Delete | Dune | 10.99 | Frank Herbert |
| <input type="checkbox"/> | Edit | Copy | Delete | The Silent Patient | 19.95 | Alex Michaelides |
| <input type="checkbox"/> | Edit | Copy | Delete | A Promised Land | 24.50 | Barack Obama |
| <input type="checkbox"/> | Edit | Copy | Delete | It Ends with Us | 16.00 | Colleen Hoover |
| <input type="checkbox"/> | Edit | Copy | Delete | 1984 | 9.99 | George Orwell |
| <input type="checkbox"/> | Edit | Copy | Delete | The Subtle Art of Not Giving a F*ck | 15.49 | Mark Manson |
| <input type="checkbox"/> | Edit | Copy | Delete | Sapiens: A Brief History of Humankind | 18.99 | Yuval Noah Harari |
| <input type="checkbox"/> | Edit | Copy | Delete | The Book Thief | 14.99 | Markus Zusak |
| <input type="checkbox"/> | Edit | Copy | Delete | Verity | 17.99 | Colleen Hoover |
| <input type="checkbox"/> | Edit | Copy | Delete | Circe | 16.99 | Madeline Miller |
| <input type="checkbox"/> | Edit | Copy | Delete | Rich Dad Poor Dad | 11.95 | Robert T. Kiyosaki |
| <input type="checkbox"/> | Edit | Copy | Delete | The Seven Husbands of Evelyn Hugo | 14.99 | Taylor Jenkins Reid |
| <input type="checkbox"/> | Edit | Copy | Delete | Project Hail Mary | 20.00 | Andy Weir |
| <input type="checkbox"/> | Edit | Copy | Delete | Harry Potter and the Philosopher's Stone | 12.99 | J.K. Rowling |

☒ Console ☐ Check all ☐ With selected: Edit Copy Delete Export

```
SELECT Title, AvgCustomerRating, TotalReviews FROM Books ORDER BY AvgCustomerRating DESC, TotalReviews DESC;
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows: Filter rows: Sort by key:

Extra options

| | | | | Title | AvgCustomerRating | 1 | TotalReviews | 2 |
|--------------------------|--|--|--|--|-------------------|---|--------------|---|
| <input type="checkbox"/> | | | | Harry Potter and the Philosopher's Stone | 4.9 | | 20000 | |
| <input type="checkbox"/> | | | | Becoming | 4.9 | | 15800 | |
| <input type="checkbox"/> | | | | Sapiens: A Brief History of Humankind | 4.9 | | 14200 | |
| <input type="checkbox"/> | | | | Atomic Habits | 4.8 | | 12500 | |
| <input type="checkbox"/> | | | | The Book Thief | 4.8 | | 11000 | |
| <input type="checkbox"/> | | | | Project Hail Mary | 4.8 | | 10000 | |
| <input type="checkbox"/> | | | | Dune | 4.8 | | 9200 | |
| <input type="checkbox"/> | | | | A Promised Land | 4.7 | | 14500 | |
| <input type="checkbox"/> | | | | 1984 | 4.7 | | 13000 | |
| <input type="checkbox"/> | | | | The Alchemist | 4.7 | | 11300 | |
| <input type="checkbox"/> | | | | The Seven Husbands of Evelyn Hugo | 4.7 | | 10700 | |
| <input type="checkbox"/> | | | | Circe | 4.7 | | 9400 | |
| <input type="checkbox"/> | | | | Educated | 4.6 | | 13400 | |
| <input type="checkbox"/> | | | | Rich Dad Poor Dad | 4.6 | | 12500 | |
| <input type="checkbox"/> | | | | The Silent Patient | 4.6 | | 9800 | |
| <input type="checkbox"/> | | | | The Midnight Library | 4.5 | | 8700 | |
| <input type="checkbox"/> | | | | Verity | 4.5 | | 8200 | |
| <input type="checkbox"/> | | | | Where the Crawdads Sing | 4.4 | | 10200 | |
| <input type="checkbox"/> | | | | The Subtle Art of Not Giving a F*ck | 4.4 | | 8900 | |
| <input type="checkbox"/> | | | | It Ends with Us | 4.3 | | 7700 | |

Console

[Check all](#)

[With selected:](#)

[Edit](#)

[Copy](#)

[Delete](#)

[Export](#)

`SELECT Publisher, GROUP_CONCAT(Title SEPARATOR ', ') AS Books FROM Books GROUP BY Publisher;`

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows: Filter rows: Sort by key:

Extra options

| | Publisher | Books |
|---|---------------------------|--|
| <input type="checkbox"/> Edit Copy Delete | Ace Books | Dune |
| <input type="checkbox"/> Edit Copy Delete | Atria Books | It Ends with Us, The Seven Husbands of Evelyn Hugo |
| <input type="checkbox"/> Edit Copy Delete | Ballantine Books | Project Hail Mary |
| <input type="checkbox"/> Edit Copy Delete | Bloomsbury | Harry Potter and the Philosopher's Stone |
| <input type="checkbox"/> Edit Copy Delete | Canongate Books | The Midnight Library |
| <input type="checkbox"/> Edit Copy Delete | Celadon Books | The Silent Patient |
| <input type="checkbox"/> Edit Copy Delete | Crown Publishing | Becoming, A Promised Land |
| <input type="checkbox"/> Edit Copy Delete | G.P. Putnam's Sons | Where the Crawdads Sing |
| <input type="checkbox"/> Edit Copy Delete | Grand Central Publishing | Verity |
| <input type="checkbox"/> Edit Copy Delete | Harper | The Subtle Art of Not Giving a F*ck |
| <input type="checkbox"/> Edit Copy Delete | HarperOne | The Alchemist |
| <input type="checkbox"/> Edit Copy Delete | Harvill Secker | Sapiens: A Brief History of Humankind |
| <input type="checkbox"/> Edit Copy Delete | Little, Brown and Company | Circe |
| <input type="checkbox"/> Edit Copy Delete | Penguin Books | Atomic Habits |
| <input type="checkbox"/> Edit Copy Delete | Picador | The Book Thief |
| <input type="checkbox"/> Edit Copy Delete | Plata Publishing | Rich Dad Poor Dad |
| <input type="checkbox"/> Edit Copy Delete | Random House | Educated |
| <input type="checkbox"/> Edit Copy Delete | Secker & Warburg | 1984 |

☐ Check all With selected: [Edit](#) [Copy](#) [Delete](#) [Export](#)

Now let's check how well you have understood so far! 😊

| How did you find attempting this exercise? | Choose one (X) |
|--|-------------------------------------|
| 😊 I'm very confident that I have understood and completed this task successfully | <input checked="" type="checkbox"/> |
| 🙂 I'm fairly confident that I have understood and completed this task successfully | <input type="checkbox"/> |
| 😐 I think I'm getting there but I need to work on understanding this a little more | <input type="checkbox"/> |
| 😓 I got a bit stuck and I will need to work on understanding this a lot more | <input type="checkbox"/> |


Exercise 4

In PHP create a table that has the following headings

```
<tr>
  <td><b>BookID</b></td>
  <td><b>Title</b></td>
  <td><b>Subject</b></td>
  <td><b>Publisher</b></td>
</tr>
```



You should then [populate this table](#) with the results of a SQL query that [pulls this data out of your database](#).



My solution

Record your notes about how you attempted to answer the question / solve the exercise in this section (just edit the text box )

| BookID | Title | Subject | Publisher |
|--------|--|--------------------|---------------------------|
| 1 | Atomic Habits | Self-help | Penguin Books |
| 2 | The Midnight Library | Fiction | Canongate Books |
| 3 | The Alchemist | Fiction | HarperOne |
| 4 | Becoming | Memoir | Crown Publishing |
| 5 | Educated | Memoir | Random House |
| 6 | Where the Crawdads Sing | Fiction | G.P. Putnam's Sons |
| 7 | Dune | Science Fiction | Ace Books |
| 8 | The Silent Patient | Thriller | Celadon Books |
| 9 | A Promised Land | Memoir | Crown Publishing |
| 10 | It Ends with Us | Romance | Atria Books |
| 11 | 1984 | Dystopian | Secker & Warburg |
| 12 | The Subtle Art of Not Giving a F*ck | Self-help | Harper |
| 13 | Sapiens: A Brief History of Humankind | History | Harvill Secker |
| 14 | The Book Thief | Historical Fiction | Picador |
| 15 | Verity | Romantic Suspense | Grand Central Publishing |
| 16 | Circe | Fantasy | Little, Brown and Company |
| 17 | Rich Dad Poor Dad | Personal Finance | Plata Publishing |
| 18 | The Seven Husbands of Evelyn Hugo | Fiction | Atria Books |
| 19 | Project Hail Mary | Science Fiction | Ballantine Books |
| 20 | Harry Potter and the Philosopher's Stone | Fantasy | Bloomsbury |

Now let's check how well you have understood so far!

| How did you find attempting this exercise? | Choose one (X) |
|---|----------------|
|  I'm very confident that I have understood and completed this task successfully | x |
|  I'm fairly confident that I have understood and completed this task successfully | |

| | |
|--|--|
|  I think I'm getting there but I need to work on understanding this a little more | |
|  I got a bit stuck and I will need to work on understanding this a lot more | |

Exercise 5

Make sure you read over the assessment brief and clarify any questions you have with your lab tutor and the module coordinator.

| Now let's check how well you have understood so far! 😊 | |
|--|----------------|
| How did you find attempting this exercise? | Choose one (X) |
| 😊 I'm very confident that I have understood and completed this task successfully | x |
| 🙂 I'm fairly confident that I have understood and completed this task successfully | |
| 🤔 I think I'm getting there but I need to work on understanding this a little more | |
| 😓 I got a bit stuck and I will need to work on understanding this a lot more | |

Additional Notes

Note: the following examples use the example Northwind database, we can create it using the following script:

[northwind.sql](#)

We can insert data into the database using the following script:

[northwind-data.sql](#)

Note running the scripts are not vital to understanding the content.

Select

What is it?

The select statement is used to retrieve records from a database, you are selecting the attributes that you want to retrieve for each record as the DBMS scans through the database.

Why might I use it?

SELECT statements are one of the simplest queries we can construct and are used as the basis for most queries that involve retrieving records from one or more tables.

For instance, we may wish to retrieve all the record tuples that are within a Customers table.

How can it be done?

Example

```
SELECT *  
FROM Customer;
```

Table Aliases

What is it?

SQL table aliases are used to give a table a temporary name that it can be referred to by. Tables are created by following the **FROM** clause with the **AS** keyword, which will create an alias for the duration of that query, outside of the specific execution of that query the alias will not exist.

The upshot of this is that aliases are almost like local variables in programming, which can be used to refer to your tables, they only exist within a given executing statement (in other words we cannot reference the alias in another statement) and the alias will be destroyed at the end of that statement execution (again somewhat like a local variable in programming)

Why might I use it?

Table aliases are often used to make table names easier to refer to within a SQL statement. For example, we may wish to use the ALIAS 'c' instead of Customer as shorthand to refer to the customer table.

This has somewhat more limited usage in queries that only involve one table, but in larger queries that may involve multiple tables aliases are very useful to allow us to keep track of which table we are referring to at any given point via our shorthand alias value.

How can it be done?

Example

```
SELECT c.*  
FROM Customer AS c;
```

Qualified select: Qualifying attributes (vertical qualification)

What is it?

A vertically qualified select statement allows us to specify individual attributes (columns) that we wish to extract from a given query result. It is 'vertically qualified' because attributes (columns) are (at least conceptually) thought of as the vertical labels in our results set, therefore it specifies ways to extract or 'slice' or result data "vertically".

Why might I use it?

How can it be done?

We use an appropriately constructed SQL select statement to tell the DBMS to perform this task.

We can assume (for the time being) that the database engine will understand that it needs to first **access the data object** representing the specific **table** in question it will **scan through each record** for the entire table, but only note down the data within the **specifically selected attributes** for that record.

For instance, if we take the following table

| Attribute1 | Attribute2 | Attribute3 | Attribute4 |
|---------------------|---------------------|---------------------|---------------------|
| Record 1 Attribute1 | Record 1 Attribute2 | Record 1 Attribute3 | Record 1 Attribute4 |
| Record 2 Attribute1 | Record 2 Attribute2 | Record 2 Attribute3 | Record 2 Attribute4 |

Figure 3 ExampleTable

And execute this SELECT statement

```
SELECT Attribute1, Attribute3
FROM ExampleTable;
```

The DBMS will understand that it only needs to extract Attribute1 and Attribute3 for each record in the result set.

| Attribute1 | Attribute2 | Attribute3 | Attribute4 |
|---------------------|---------------------|---------------------|---------------------|
| Record 1 Attribute1 | Record 1 Attribute2 | Record 1 Attribute3 | Record 1 Attribute4 |
| Record 2 Attribute1 | Record 2 Attribute2 | Record 2 Attribute3 | Record 2 Attribute4 |

Figure 4 Data that is read and extracted by the database engine is highlighted in blue

So it will return the following result

| Attribute1 | Attribute3 |
|---------------------|---------------------|
| Record 1 Attribute1 | Record 1 Attribute3 |
| Record 2 Attribute1 | Record 2 Attribute3 |

Figure 5 Data that the database engine has read and extracted is highlighted in blue

Example

```
SELECT Id, CompanyName, ContactName, Country, City  
FROM Customer;
```

Attribute aliases

What is it?

Attribute aliases allow us to take a specific attribute that we are retrieving from our intermediate result set and give it a new temporary name / identifier in the result set that is returned.

Why might I use it?

Attribute aliases allow us to give more meaningful / readable names to particular attributes that are being retrieved. It is particularly useful when using tables where the meaning of the attribute names is not entirely clear, thus by given the attributes an alias we can give additional context as to their meaning within the set of results retrieved.

Attribute aliases are also useful when performing JOINS between two or more tables, as they allow us to rename attributes within the context of our result set rather than within the context of an individual table, a very simple example of this is that you could have an ID attribute on the Customer table as well as an ID attribute on the Order table without aliasing it will not be clear within the result set of a join between the two tables which ID belongs to which relation / table.

How can it be done?

Example

```
SELECT Id AS 'Customer ID', CompanyName AS 'Name of Company', Address
AS 'Company Address', Country, City
FROM Customer;
```

Exercise

Construct 5 SQL statements that use this technique, try to create a variety of queries using this technique.

When constructing your statements you should include a summary of:

- What your statement or query does
 - o A summary of the query in terms of **what is being done** within the SQL statement itself from the **perspective of the database**
 - This would discuss the query from the perspective of what it is doing in with the data objects (tables, attributes, etc) within the database from a more technical perspective
 - o A summary of the query in terms of the **real-life information it is trying to extract / change** from the **perspective of a user in a business setting**
 - This would discuss the query from the **perspective of what it is doing in terms of the actual useful business information / data it is trying to find** from the **perspective of a user trying to extract / change further information about an aspect of the business**

- Why your statement does this
 - o An explanation of **why your statement might be used in the real world** for the purpose of **extracting / changing information relevant to aspects of the business**
- How your statement operates
 - o Explain each **clause** in your **statement** and discuss how each of those clauses/keywords contribute to constructing the overall statement

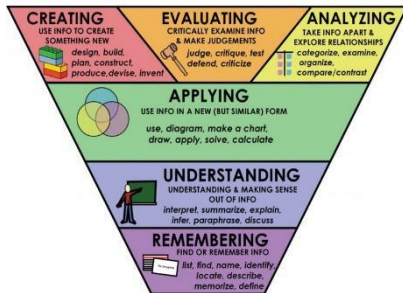


Figure 7 Taxonomy of Learning. Image courtesy of Rawia Inaim / Kwantlen Polytechnic University

This exercise should be seen as an opportunity to consolidate your thought process and high-level understanding of the basics of this SQL technique. With this exercise you should focus on strengthening the lower levels of your learning stack: **remembering**, **understanding** and **applying** the **knowledge** and **skills** gained.

Note

If you are unsure about how to approach attempting aspects of this exercise, discuss with the tutor as soon as possible – I am here to help! 😊

Qualified select: Qualifying records (horizontal qualification)

What is it?

Qualified select allows us to tell the DBMS to filter our results so that only specific records are retrieved from the database.

Why might I use it?

We use horizontal qualification when we only want records that meet a specific condition (predicate). For instance, we may want to only retrieve Customer details for our UK customers.

How can it be done?

We use the WHERE clause to allow us to filter the result set retrieved from a table based on a specific condition provided.

As well as SELECT statements we can also use the WHERE clause to qualify UPDATE and DELETE operations.

Example

```
SELECT *  
FROM Customer  
WHERE Country = 'UK';
```


Aggregate functions: Count

What is it?

The Count aggregate function allows us to ask the DBMS to calculate how many records are provided in our set of results for a given query.

Why might I use it?

We can use the Count function when we want to count the amount of complete records retrieved from the database. For instance, we may want to count the amount of Customers that we have.

How can it be done?

Example

```
SELECT Count (*)  
FROM Customer;
```

Unqualified Join / Cross Join: The Cartesian Product

What is it?

A way of pairing the data within each records for one table with the data for each record in another table.

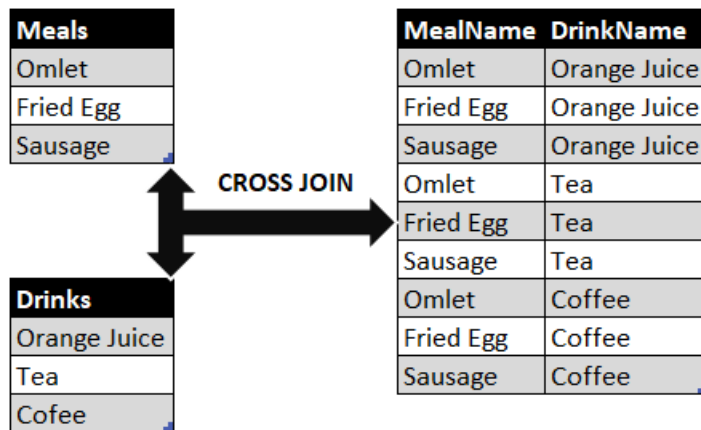


Figure 10 Performing a CROSS JOIN on two example tables to retrieve the cartesian product of the two tables

Why might I use it?

Because they join all the data for each record in one table with all the data for each record in another table CROSS JOIN operations are usually limited in their practical usefulness. One of their major uses is as an intermediary step to demonstrate how INNER JOIN operations construct their result sets.

Another major use of the CROSS JOIN operation is to test the performance of a given DBMS (and the database it is managing) when handling queries that need to combine data from multiple tables.

How can it be done?

There are two major ways of performing a CROSS JOIN in SQL. One is simply by performing a SELECT operation on two tables, without a WHERE clause which attempts to match relevant data from one table with another.

The other method is to simply perform a SELECT operation on one table and use the CROSS JOIN keyword to introduce the second table that should be included in the result set.

Example

```
SELECT o.Id AS 'Order ID',  
       o.CustomerID AS 'Customer ID on Order table',  
       c.Id AS 'Customer ID on Customer Table',  
       c.CompanyName AS 'Company Name'  
FROM 'Order' AS o, Customer AS c;
```

Figure 11 Selecting records from two tables will automatically produce an unqualified join

```
SELECT o.Id AS 'Order ID',  
       o.CustomerID AS 'Customer ID on Order table',  
       c.Id AS 'Customer ID on Customer Table',  
       c.CompanyName AS 'Company Name'  
FROM 'Order' AS o  
CROSS JOIN Customer AS c;
```

Figure 12 An unqualified join can also be performed using the CROSS JOIN keyword

Inner Join

What is it?

An inner join allows you to select records from both tables that **match** based on a specific criterion (called a join condition) provided by one or more **predicate** conditions.

Why might I use it?

One of the main benefits of database systems is the ability to find related data records in different locations (e.g. tables). An inner join allows us to better identify which records in a given table also have related data in another table, the relationship under investigation is what has been specified in the join condition.

For instance, say we wanted to find all the orders that have been placed for each customer, we would join the Order table with the Customer table and use the join criteria that for each record in the order table we only retrieve the company record(s) for the company that placed that order (based on the company ID).

How can it be done?

Conceptually the inner join can be thought of as a horizontally qualified cross join, in other words it states which records from the cartesian product of two tables to extract as part of the result set of records.

Example

```
SELECT o.Id AS 'Order ID',
       o.CustomerID AS 'Customer ID on Order table',
       c.Id AS 'Customer ID on Customer Table',
       c.CompanyName AS 'Company Name'
FROM 'Order' AS o, Customer AS c
WHERE o.CustomerID = c.Id;
```

Figure 14 Inner Join as a qualified select query of multiple tables

```
SELECT o.Id AS 'Order ID',
       o.CustomerID AS 'Customer ID on Order table',
       c.Id AS 'Customer ID on Customer Table',
       c.CompanyName AS 'Company Name'
FROM 'Order' AS o
CROSS JOIN Customer AS c
WHERE o.CustomerID = c.Id;
```

Figure 15 Inner join as a qualified select query of the cartesian product of two tables

```
SELECT o.Id AS 'Order ID',  
       o.CustomerID AS 'Customer ID on Order table',  
       c.Id AS 'Customer ID on Customer Table',  
       c.CompanyName AS 'Company Name'  
FROM 'Order' AS o  
INNER JOIN Customer AS c  
ON o.CustomerID = c.Id;
```

Figure 16 Inner join using the explicit Inner Join syntax

Food for thought

Taking our knowledge of programming languages and the fact that tools like compilers and interpreters can automatically analyse our code to identify optimisations that can be applied to our programs, which of the above examples of inner join do you believe would likely be the easiest for the database engine to identify optimisations for?

Food for thought

Can you turn the explicit inner join example found above into a cross join by modifying the join condition only? If so, how would you do it and why does / doesn't it work?

Left Join

What is it?

The **LEFT JOIN** returns **all those records that match the join criterion** between the first “Left” table ‘Customer’ and the second “Right” table ‘Order’

AND

It also **returns any records in the “Left table” that do not meet the join criteria.**

Why might I use it?

Let’s say that we wanted to retrieve data which identifies the specific orders that have been placed by a company AND we also wanted to include those companies which have not placed any orders at all.

How can it be done?

We use the LEFT JOIN keyword to specify that we want to produce a set of results that joins records from two tables together, based on our join criterion. But the LEFT JOIN will mean that it will also include records in the first table that do not have related data in the second table (in other words it will still include companies who haven’t placed an order).

Example

Let’s first insert a record into our Customer table that we know will have no orders assigned to it.

```
INSERT INTO Customer
VALUES ('NOORDER', 'DoesntOrderInc', 'Jaen Hghar',
       'Faceless Man', 'House of Black and White', 'Braavos',
       NULL, 'GOT5', 'Essos', NULL, NULL);
```

Then let’s write our Left Join query

```
SELECT c.CompanyName, o.ID
FROM Customer AS c
LEFT JOIN 'Order' AS o
ON c.ID = o.CustomerID;
```

If we look through the result set we would be able to find a record for ‘Doesn’t Order Inc’ where their Order ID is “None” or NULL, this is because the **LEFT JOIN** has returned **all those records that match the join criterion** between the first “Left” table ‘Customer’ and the second “Right” table ‘Order’

AND

It also **returns any records in the “Left table” that do not meet the join criteria.**

In this way we can produce a result set that contains a list of all Customers and the Orders they have placed that also includes Customers who have not placed an Order yet.

Food for thought

Compare the above LEFT JOIN example with a similarly structured inner join, what do you notice about the set of company results?

Group by

What is it?

The Group By statement allows us to group records that have the same values together, to provide a summary of attributes for that group.

Why might I use it?

We tend to use the group by statement when we have multiple records that are related to a particular value.

For instance, we may wish to work out the amount of orders that have been placed by a particular

customer to do this we would group our result set by an attribute that allows us to uniquely identify each

company we would then count the amount of records within the database that are Orders related to that

Company (this may involve a Join condition if the information is contained in multiple tables

How can it be done?

We use the Group By statement to identify the attribute(s) within our result set that we wish to group

together. For instance, if we wanted to group together records for each Company we would select an

attribute that can be used to identify each Company and tell the DBMS to group the results based on that

attribute.

If we wanted to group together records of our customers for each Country in order to find the amount of

Customers in each country we would select an attribute that can be used to identify the Country of each

customer (this could be a name, an ID, a country code, etc) and tell the DBMS to group the results

based on that attribute.

Group By is commonly used with one or more aggregate functions to allow an aggregated summary to

be calculated for each group of records that the result set has been divided into.

Example


```
SELECT c.CompanyName, Count(o.ID) AS 'Amount of Orders'
FROM Customer AS c
LEFT JOIN 'Order' AS o
ON c.ID = o.CustomerID
GROUP BY c.CompanyName;
```

```
SELECT Country, COUNT(ID) AS 'Amount of Customers'

FROM Customer

GROUP BY Country;
```

Order by

What is it?

Allows us to tell the DBMS to sort our result set in a particular alphanumeric order based on some attribute (column)

Why might I use it?

We may wish to look at the set of all orders for each Customer, to ensure that the orders for a Customer are all together in our result set we might choose to Order our results based on a unique identifier for the Customer.

How can it be done?

We use the Order By clause to tell the database what attribute it must use to sort the records. Essentially an initial result set of relevant records will be retrieved by the database which will then have some sorting algorithm applied to the results, the sorting algorithm will use the specific attribute chosen to decide the sorting order of the results set retrieved. In this way we go from the usual notion of having an unordered collection of results to an ordered collection of results.

Example

Let's ensure that we have inserted the record for Doesn'tOrderInc into the Customer table.

```
INSERT INTO Customer
VALUES ('NOORDER', 'DoesntOrderInc', 'Jaen Hghar',
      'Faceless Man', 'House of Black and White', 'Braavos',
      NULL, 'GOT5', 'Essos', NULL, NULL);
```

Then let's perform our previous left join to retrieve the Orders associated with each company, where is DoesntOrderInc in the result set?

```
SELECT c.CompanyName, o.ID AS 'Order ID'
FROM Customer AS c
LEFT JOIN 'Order' AS o
ON c.ID = o.CustomerID;
```

It is likely near the bottom of the result set, this is because records in a database table are not stored in any specific order by default as a table is just a an unordered collection of records.

Now let's order our query by company name, this means that the result set of records will be sorted into alphanumeric order by the Company Name attribute.

```
SELECT c.CompanyName, o.ID AS 'Order ID'
FROM Customer AS c
LEFT JOIN 'Order' AS o
ON c.ID = o.CustomerID
ORDER BY c.CompanyName;
```

Food for thought

Why did I believe that it was likely that the DoesntOrderInc record was near the bottom of the result set? What does this tell us about how records might be stored within our database?

Extension task



Finished early? Excellent! ☺

Use this opportunity to discuss with your lab tutor what you have understood so far and ask them to assign some tasks that you could attempt to further refine your skills and knowledge.