COMP1314 - Data Management- (UoSM)-2025/26

Draft Coursework – LINUX and MySQL

Module Code:	COMP1314		
Module Title:	DATA MANAGEMENT		
Module Leader:	Assistant Professor, Dr Fairuz Safwan		
Assessment Type:	Coursework	**Weightage:**	**30%**
Submission Due Date:	**8th December 2025, 17:00 (Malaysia Time)**		
Method of Submission:	Handin via the Assignment Tab of Blackboard		
GenAI Tier Specification	**Tier 1: No GenAI use [applies All - Entire work]**		

This assessment relates to the following Module Learning Outcomes:

A. Numercy Skills	CLO2	A. Adopt a proactive approach towards entity-relationship modeling integrated with Unix-based tools, utilizing shell scripts, pipes, and filters to optimize software design and development processes (A3, PLO7)

COMP1314 Data Management (UoSM) 2025/2026

Coursework – LINUX and MySQL

This Coursework is worth **30%** of the total marks for this module. The deadline is **Monday, 8th December 2025, 17:00 (Malaysia Time), handin' via the Blackboard** site. Depending on your prior knowledge, the work will take 30-40 hours approximately.

Coursework Overview:

Submit to	:	**Blackboard submission link named "Coursework Submission".**
Feedback	:	2-3 weeks after the deadline

General notes:

- Keep your code clean, complete, and easy to understand
- You can show your distinctive skills by experimenting with new and different things!
- You are required to submit **ONE (1) Word document report** and **a ZIP File that contains the Script, MySQL Database Dump, and everything else** related to the project.
- Explain your script and your database in the Word document.
- You are expected to write a report with a minimum of **THREE (3)** pages which should include the **FOUR (4)** main components
  - **Unix Script for Data Collection**
  - **Unix Script for Plotting**
  - **Use of Git for Version Control**
  - **MySQL for Data Storage**

Rules:

- This is an **INDIVIDUAL COURSEWORK**.
- Late submissions will be penalised at 10% per working day
- No work can be accepted after feedback has been given
- You should expect to spend up to **30 hours** on this assignment
- Please note the University regulations regarding academic integrity
- Copying works from/to other students will be penalized under the regulation of the University of Southampton

Academic Integrity

"In this entire assignment, students are **prohibited** from **using any GenAI tools** for their **assessed work**. This includes entering any part of the assignment or your assessed work to GenAI, whether by pasting/typing, uploading files, or describing content directly or through plugins. Basic tools that assist spelling and grammar, translation and calculation without generating new content or ideas, can be used unless specified otherwise by the assessment setter. GenAI may be used to explain lecture slides and notes to enhance understanding of a relevant topic areas. Students are not required to complete a GenAI Declaration Form."

For more information on GenAI refer here

The focus of this coursework is learning. Sharing or reproducing others' ideas as your own is strictly prohibited. It's acceptable and encouraged for you to refer to certain web stuff in your notebook. However, in such cases, you must credit the original author's information or the source you referred to. If you cite/acknowledge any source that assists you in obtaining relevant knowledge for completing this coursework, it is highly appreciated, and no marks will be deducted for the same.

Help and Assistance
During lab hours, academics will be on hand to assist you. After the lab, you will need to continue on your own schedule. You are urged to speak with your colleagues about this work and share information and thoughts with them. Nevertheless, you need to record your own findings in the report.

Marking Guidelines & Feedback
This coursework is worth 30% of the total marks for this module. It will be assessed using the attached assessment rubrics. Marks with written feedback will be available within **THREE (3)** weeks of the coursework due date.

Coursework – LINUX and MySQL

The coursework is briefed in the following sections:

Data is often all over the place. It comes in many different forms and has many weird quirks - often it cannot be directly used for any real purpose. Enter data cleaning: the process of making data behave and manifest in a usable format. This is your task - take some (rather horrible) looking data and change it into an easy-to-use, useful and accessible format.

The coursework is divided into FOUR (4) components:

- **Unix Script for Data Collection**
  Create an automation tool that will collect data every certain period (daily or hourly) and keep those data in a MySQL database
- **Unix Script for Plotting**
  Once you have collected several data, you have to create a script to show graphs, such as day-to-day weather, gold price changes, price changes on Lazada, etc (see below on the project choices).
- **Use of Git for Version Control**

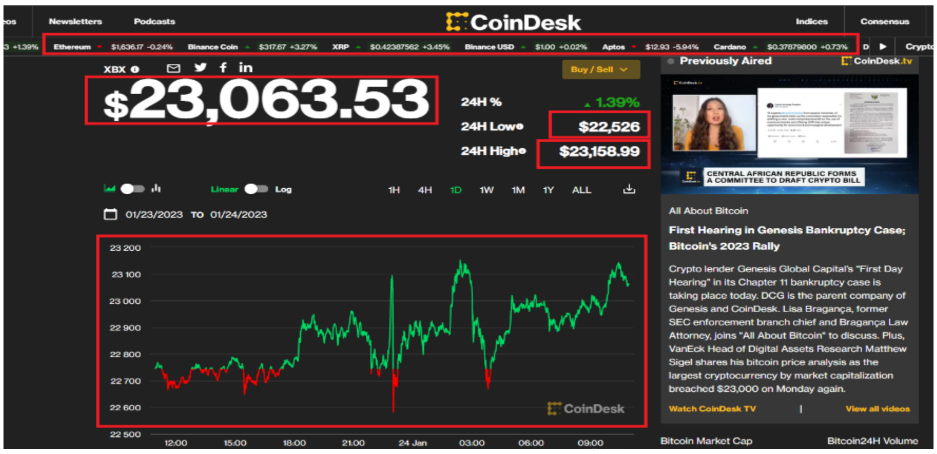
You must commit your scripts to a GitHub repository occasionally at a sensible rate. Your first commit must be at least a week before the deadline. For the submission, mention your GitHub URL and provide screenshots in the report.

- **MySQL for Data Storage**
  This will be your main data store; it stores historical data from the websites where you collect data. See the examples below.

The Project - Website Scraper/Tracker

Before you begin, **CHOOSE ONE** from the following possible scenarios (but not limited to):

- **Keep Track of BitCoin Price**
  Choose a website to track BitCoin price; possible data source:
- https://www.coindesk.com/price/bitcoin/
- https://coinmarketcap.com/currencies/bitcoin/
  What you can collect (basically, look at the numbers there, you can collect those!):
- Current BitCoin to USD rate
- 24H lowest and highest rate
- Other cryptocurrency rates
  Refer to the image below for what you can collect:

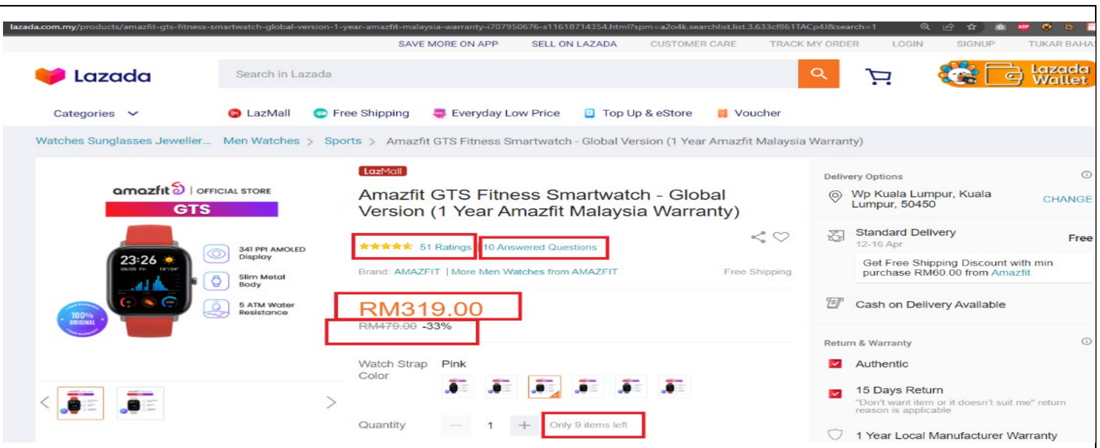


All About Bitcoin

First Hearing in Genesis Bankruptcy Case; Bitcoin's 2023 Rally

Crypto lender Genesis Global Capital's "First Day Hearing" in its Chapter 11 bankruptcy case is taking place today. DCG is the parent company of Genesis and CoinDesk. Lisa Bragança, former SEC enforcement branch chief and Bragança Law Attorney, joins "All About Bitcoin" to discuss. Plus, VanEck Head of Digital Assets Research Matthew Sigel shares his bitcoin price analysis as the largest cryptocurrency by market capitalization breached $23,000 on Monday again.

- **Online Store Promotion Tracker**
  Have you ever wanted to buy something but decided to wait for the best promotional price on Lazada (or other sites)? Rather than constantly monitoring the price of an item, why not make a robot to do that? Here's what you can collect (not limited to):
- Daily price, stock, and rating changes
- You can track several items at once
- Send you an email notification when the price drops
  Refer to the image below for what you can collect:


lazada.com.my/products/amazfit-gts-fitness-smartwatch-global-version-1-year-amazfit-malaysia-warranty-i707950676-s11618714354.html?spm=a2o4k.searchlist.list.3.633cf861TACp4J&search=1

SAVE MORE ON APP | SELL ON LAZADA | CUSTOMER CARE | TRACK MY ORDER | LOGIN | SIGNUP | TUKAR BAHA

Lazada — Search in Lazada — Lazada Wallet

Categories | LazMall | Free Shipping | Everyday Low Price | Top Up & eStore | Voucher

Watches Sunglasses Jeweller... > Men Watches > Sports > Amazfit GTS Fitness Smartwatch - Global Version (1 Year Amazfit Malaysia Warranty)

LazMall

Amazfit GTS Fitness Smartwatch - Global Version (1 Year Amazfit Malaysia Warranty)

★★★★★ 51 Ratings | 10 Answered Questions

Brand: AMAZFIT | More Men Watches from AMAZFIT — Free Shipping

RM319.00

~~RM479.00~~ -33%

Watch Strap Color: Pink

Quantity: 1 — Only 9 items left

Delivery Options: Wp Kuala Lumpur, Kuala Lumpur, 50450 — CHANGE

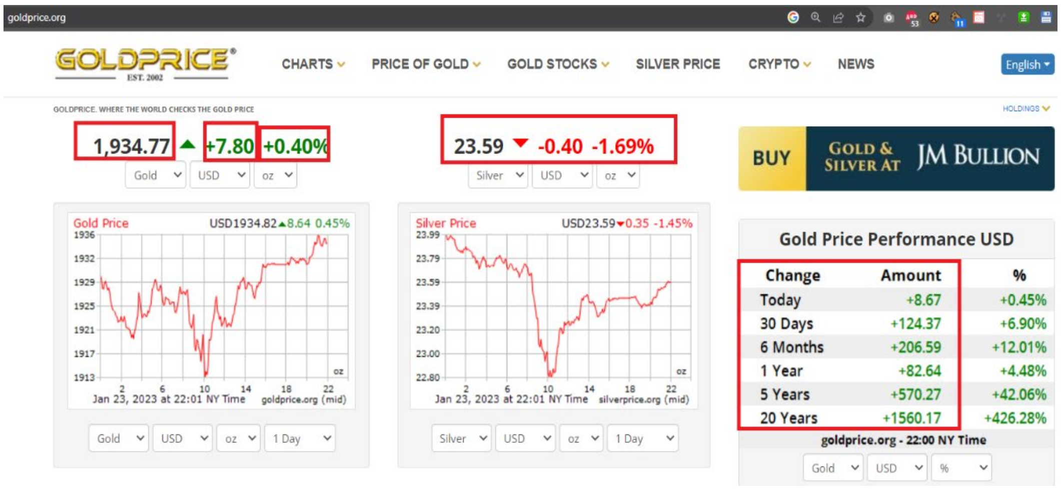
Standard Delivery, 12-16 Apr — Free

Get Free Shipping Discount with min purchase RM60.00 from Amazfit

Cash on Delivery Available

Return & Warranty: Authentic; 15 Days Return ("Don't want item or it doesn't suit me" return reason is applicable); 1 Year Local Manufacturer Warranty

- **Gold Price**
  Choose a website to track gold price; possible data source:
- https://goldprice.org/
- https://www.kitco.com/charts/livegold.html
  What you can collect (basically, look at the numbers there, you can collect those!):
- Current gold/silver price
- The change rate
- Gold performance in the last X days, X months, etc.
  Refer to the image below for what you can collect:


goldprice.org

CHARTS | PRICE OF GOLD | GOLD STOCKS | SILVER PRICE | CRYPTO | NEWS | English

GOLDPRICE. WHERE THE WORLD CHECKS THE GOLD PRICE

Gold: 1,934.77 ▲ +7.80 +0.40% (Gold, USD, oz)

Silver: 23.59 ▼ -0.40 -1.69% (Silver, USD, oz)

Gold Price USD1934.82 ▲8.64 0.45% — Jan 23, 2023 at 22:01 NY Time, goldprice.org (mid) — Gold, USD, oz, 1 Day

Silver Price USD23.59 ▼0.35 -1.45% — Jan 23, 2023 at 22:01 NY Time, silverprice.org (mid) — Silver, USD, oz, 1 Day

Gold Price Performance USD

Change	Amount	%
Today	+8.67	+0.45%
30 Days	+124.37	+6.90%
6 Months	+206.59	+12.01%
1 Year	+82.64	+4.48%
5 Years	+570.27	+42.06%
20 Years	+1560.17	+426.28%

goldprice.org - 22:00 NY Time (Gold, USD, %)

- **Other Trackers!**
  You can choose to track other things as well, as long as the weightage is more/less the same with the ones mentioned above.

Important Notes

Scraping data using CURL can be tricky. Popular websites tend to have a protection that prevents a robot from getting or tracking data from their websites. Such a data tracker (like the one that we will build) can be misused to overload websites resources (CPU, RAM), and thus creates a DOS (denial of service) attack. Being ethical is important, as we collect data at a sensible rate that will not incur such a burden.

One quick tip, we should collect data from non-popular websites whenever possible. Avoid big names like Google, Facebook, etc.

Examples of such restrictions:

- **Shopee** has an **asynchronous loading process**, where a simple CURL won't be able to get the fully loaded HTML page
- **Instagram** has a **login restriction**, where we must log in to view any Instagram users, and thus crawling with CURL won't produce anything useful
- **Many forums and websites** have **Cloudflare protection** and/or **CAPTCHA protection** that will typically block CURL requests and other forms of DOS attacks

How to know if a website that we CURLed is protected?
After we do a CURL command, such as

```
curl https://shopee.com.my/product/35678246/8420261150 > data.html
```

Check the data.html file, does it match the one displayed on your browser? Nope.

1. Unix Script for Data Collection (65% Marks)

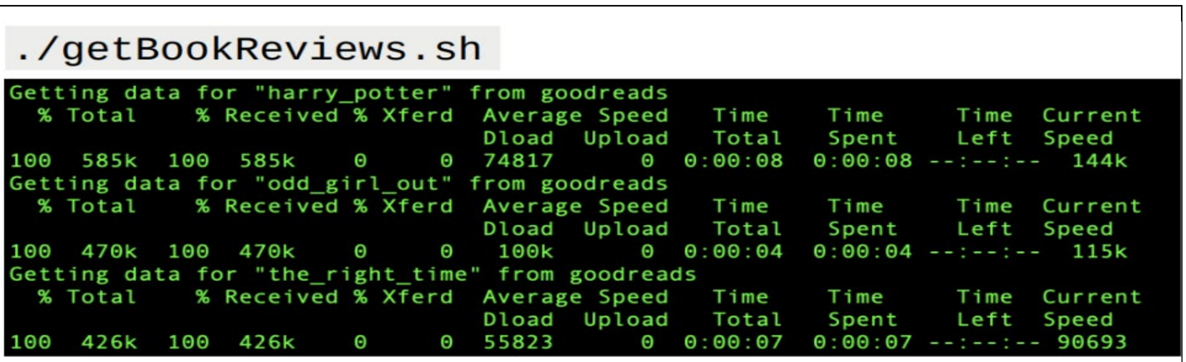
Based on the **ONE (1) scenario** that you have chosen before, you have to build your bash/awk script to achieve the stated objective. This typically includes:

- **Scraper/Tracker Script**: One or more .sh files that contain your tracker program
- **Data Input/Output**: Your .sh file need to collect data and store some data in MySQL database (refer to section 4)
- **Crontab**: A crontab setup to run your .sh file every certain period (daily or hourly). Learn about crontab here: https://www.geeksforgeeks.org/crontab-in-linux-with-examples/. Proceed to crontab once you think that your .sh file has worked properly

Assessment criterias:

- Data parsing complexity - Getting the right data and cleaning them (15%)
- Data manipulation complexity, such as arranging data to array, converting to number or date, etc (15%)
- Error handling, such as what if network is down or the website itself is down while the script is running, or what if the website blocks your script (10%)
- Inserting data to database (15%)
- Crontab setup (10%)

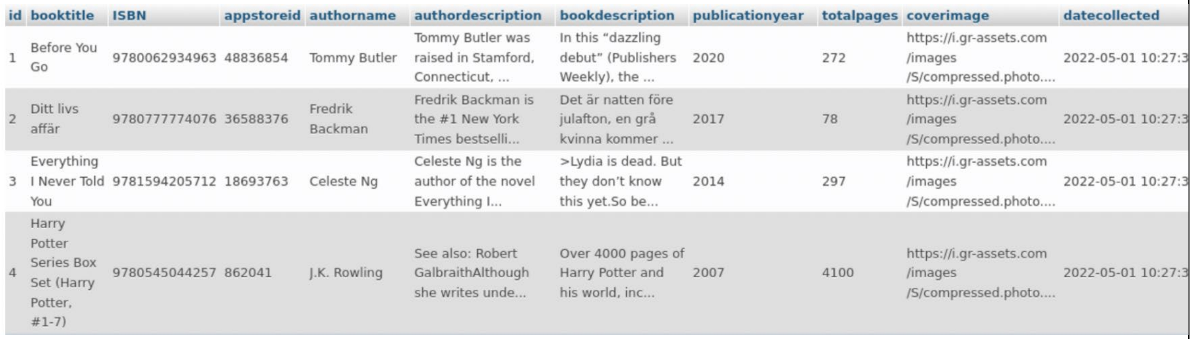
Refer to an example below, where a .sh script was built (Credit to *Jing Ru Ang*, Part 3), and it can be executed through terminal. It also shows some printouts for clarity of what is successful and not. Note that here, the script collects book information and reviews data from https://www.goodreads.com/.



```
./getBookReviews.sh
```

```
Getting data for "harry_potter" from goodreads
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  585k  100  585k    0     0  74817      0  0:00:08  0:00:08 --:--:--  144k
Getting data for "odd_girl_out" from goodreads
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  470k  100  470k    0     0   100k      0  0:00:04  0:00:04 --:--:--  115k
Getting data for "the_right_time" from goodreads
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  426k  100  426k    0     0  55823      0  0:00:07  0:00:07 --:--:-- 90693
```

The data will then be stored in a MySQL database, for example:



id	booktitle	ISBN	appstoreid	authorname	authordescription	bookdescription	publicationyear	totalpages	coverimage	datecollected
1	Before You Go	9780062934963	48836854	Tommy Butler	Tommy Butler was raised in Stamford, Connecticut, ...	In this “dazzling debut” (Publishers Weekly), the ...	2020	272	https://i.gr-assets.com /images /S/compressed.photo....	2022-05-01 10:27:3
2	Ditt livs affär	9780777774076	36588376	Fredrik Backman	Fredrik Backman is the #1 New York Times bestselli...	Det är natten före julafton, en grå kvinna kommer ...	2017	78	https://i.gr-assets.com /images /S/compressed.photo....	2022-05-01 10:27:3
3	Everything I Never Told You	9781594205712	18693763	Celeste Ng	Celeste Ng is the author of the novel Everything I...	>Lydia is dead. But they don’t know this yet.So be...	2014	297	https://i.gr-assets.com /images /S/compressed.photo....	2022-05-01 10:27:3
4	Harry Potter Series Box Set (Harry Potter, #1-7)	9780545044257	862041	J.K. Rowling	See also: Robert GalbraithAlthough she writes unde...	Over 4000 pages of Harry Potter and his world, inc...	2007	4100	https://i.gr-assets.com /images /S/compressed.photo....	2022-05-01 10:27:3

2. Unix Script for Plotting (15% Marks)

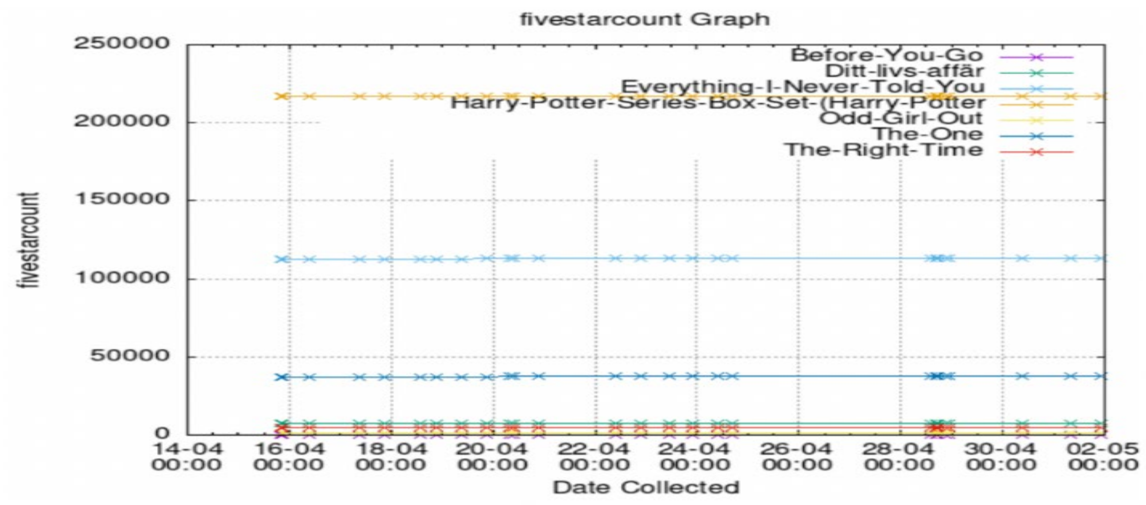
Once you have collected some data, you can start to do some plotting using GNU Plot and shell. A tutorial here https://www.youtube.com/watch?v=fLKEb-2cAzQ.

Create a minimum of **TEN (10) different plots**. You can plot things such as:

- BitCoin price changes
- Gold price changes
- Price changes on Lazada

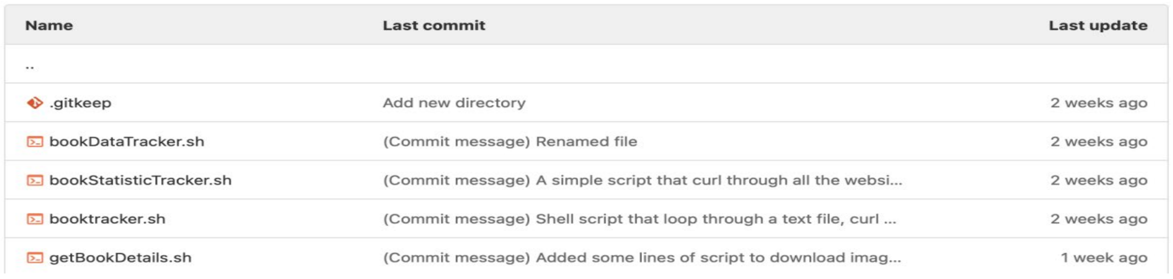
Create the scripts in a form of **functions** inside .sh file(s) that can be executed through terminal. The script is expected to be executable easily, example below (Credit to Jing Ru Ang). In the example, the script is executed by passing "fivestarcount" as the parameter, which is the plot type that we want to show.

./plotStatisticAll.sh “fivestarcount”



3. Use of Git for Version Control (10% Marks)

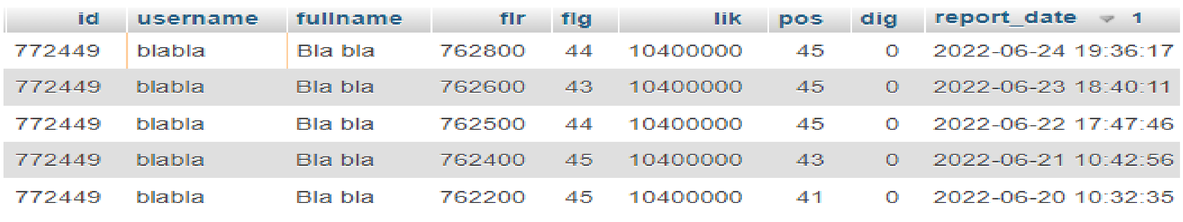
To simulate working on a real IT project, we ask you to do revision control using a Git repository. You need to commit to changes every now and then (at a sensible rate), so we can assess your progress from time to time. Furthermore, Git exercises will help you in the future to keep backups and keep track of changes you made.
You should have at least **TEN (10) commits** across at least **ONE (1) week** period.
This is an expected result of your Git commits in the report (Credit to Jing Ru Ang):



Name	Last commit	Last update
..		
.gitkeep	Add new directory	2 weeks ago
bookDataTracker.sh	(Commit message) Renamed file	2 weeks ago
bookStatisticTracker.sh	(Commit message) A simple script that curl through all the websi...	2 weeks ago
booktracker.sh	(Commit message) Shell script that loop through a text file, curl ...	2 weeks ago
getBookDetails.sh	(Commit message) Added some lines of script to download imag...	1 week ago

4. MySQL for Data Storage (10% Marks)

You have to use a **MySQL database** to keep your data. Since we will collect the data daily, we expect to see some daily historical data such as the one below. This is an example from Dr. Suresh's project, where data was collected from TikTok from 4,000 influencers every day. The changes in the TikTok users' statistics (*flr*=followers, *flg*=following, *lik*=number of likes, *pos*=number of posts) can be clearly seen in the table. Note that some fields are obscured for privacy. **You will also need to design an ERD in draw.io and export as .pdf**



id	username	fullname	flr	flg	lik	pos	dig	report_date ▼ 1
772449	blabla	Bla bla	762800	44	10400000	45	0	2022-06-24 19:36:17
772449	blabla	Bla bla	762600	43	10400000	45	0	2022-06-23 18:40:11
772449	blabla	Bla bla	762500	44	10400000	45	0	2022-06-22 17:47:46
772449	blabla	Bla bla	762400	45	10400000	43	0	2022-06-21 10:42:56
772449	blabla	Bla bla	762200	45	10400000	41	0	2022-06-20 10:32:35

Note: You typically need more than one table, because your **database must be normalized**.

MySQL (XAMPP) has been installed in the Lubuntu VirtualBox package. Here's how to run the server:

```
sudo /opt/lampp/lampp restart
/opt/lampp/bin/mysql -u root
```

Similar to Windows' and Mac's XAMPP, you can access PHPMyAdmin using localhost/phpmyadmin inside VirtualBox.
Here's how to issue a command from a Bash script to MySQL. In order to make easier calls, you should wrap this in a functions.

```
#!/bin/bash
$(/opt/lampp/bin/mysql -u root -e "create database john_db; use john_db")
$(/opt/lampp/bin/mysql -u root -e "use john_db; create table user (id int, name varchar(50))")
username="john"
$(/opt/lampp/bin/mysql -u root -e "use john_db; insert into user values (15,'$username')"
```

COMP1314 - Data Management (UoSM)-2025/26

Marks Criteria for Coursework-UNIX & MySQL (This coursework is worth 30% of the total marks for this module)

Criteria/ Sub-criteria		CLOs	Score/ Wtg	Poor 0-1	Satisfactory 2	Good 3	Very Good 4	Excellent 5	Mark (≈Wtg * Score)
1. Unix Script for Data Collection (65%)		CLO2		The script has no any key components, and basic functionalities are missing. There is no any data parsing, manipulation, error handling, database interaction, and crontab setup.	The script lacks key components, and basic functionalities are missing. Significant issues in data parsing, manipulation, error handling, database interaction, and crontab setup.	The script includes basic components but lacks refinement. Data parsing is rudimentary, and there is minimal consideration for error handling. Basic database interactions are implemented, and the crontab setup is partially functional.	The script demonstrates competency in key areas. Data parsing shows moderate complexity, with some efforts in cleaning and organizing data. Adequate error handling is implemented, considering common scenarios. Effective data insertion into the database, and the crontab setup is reasonably configured.	The script surpasses expectations in every aspect. Data parsing is highly intricate, showcasing advanced techniques. Comprehensive error handling covers a wide range of scenarios, ensuring script robustness. Data insertion into the database is seamless, and the crontab setup is optimized for efficiency and resource utilization.	
	a) Data Parsing Complexity (15%):		3	No data extraction with limited cleaning.	Basic data extraction with limited cleaning.	Moderate data parsing with some cleaning.	Advanced data parsing with effective cleaning.	Highly intricate data parsing, demonstrating advanced techniques.	/15
	b) Data Manipulation Complexity (15%):		3	No manipulation with minimal considerations.	Basic manipulation with minimal considerations.	Moderate manipulation efforts, showing some complexity.	Advanced manipulation, demonstrating sophistication.	Exceptional data manipulation, showcasing advanced techniques.	/15
	c) Error Handling (10%):		2	No error handling with major oversights.	Limited error handling with major oversights.	Basic error handling for common scenarios.	Thorough error handling covering various scenarios.	Comprehensive error handling, ensuring script resilience.	/10
	d) Inserting Data to Database (15%):		3	There is no any data insertion with major issues.	Inadequate data insertion with major issues.	Basic data insertion with some shortcomings.	Efficient data insertion with minor improvements possible.	Seamless data insertion, demonstrating efficiency and optimization.	/15
	e) Crontab Setup (10 %):		2	There is no any Crontab setup	Crontab setup with major configuration issues.	Basic crontab setup with partial functionality.	Well-configured crontab setup, meeting most requirements.	Optimized crontab setup, ensuring efficiency and resource utilization.	/10
2.Unix Script for Plotting (15%)		CLO2	3	The script has no any key components, and basic functionalities are missing. Plots are	The script lacks key components, and basic functionalities are missing. Plots	The script includes basic components but lacks refinement. Plots are basic and show limited	The script demonstrates competency in key areas. Plots show moderate complexity and diversity. Script organization	The script exhibits a high level of sophistication. Plots are diverse, complex, and well-implemented. Script organization and execution	/15

			either missing or poorly implemented. Major issues in script organization and execution.	are either missing or poorly implemented. Major issues in script organization and execution.	diversity. Issues in script organization and execution.	and execution are adequate but may have minor issues.	are efficient, with minor improvements possible.	
3.Use of Git for Version Control (10%)	**CLO2**	**2**	There is no any use of Git or nonexistent. Less than 10 commits, and commits are clustered within a short timeframe	The use of Git is minimal or nonexistent. Less than 10 commits, and commits are clustered within a short timeframe (less than a week). Significant issues in commit messages and repository structure.	Basic use of Git is evident. At least 10 commits, but with limited spread over a week. Commits may lack meaningful messages, and the repository structure is somewhat disorganized.	Competent use of Git is demonstrated. At least 10 commits spread over a week, with meaningful commit messages. The repository structure is organized, but there might be minor issues.	Git usage is highly effective. More than 10 commits strategically spread over a week with detailed and meaningful commit messages. The repository is well-structured with minor improvements possible.	**/10**
4. MySQL for Data Storage (10%)	**CLO2**	**2**	There is no any MySQL or absent. The database structure lacks normalization, without table or poorly defined relationships. Major issues in database design and usage. No ERD is submitted	MySQL usage is minimal or absent. The database structure lacks normalization, with only one table or poorly defined relationships. Major issues in database design and usage. Simple ERD has been submitted	Basic use of MySQL is evident. There is an attempt to use multiple tables, but normalization might be incomplete or inconsistent. Issues in the design of tables and relationships. ERD shows good normalization and sufficient relationship with some errors which can be improved	Competent use of MySQL for data storage. Multiple tables are used, and normalization is evident. Relationships between tables are well-defined, but there might be minor issues. ERD shows very good normalization and very good relationship and cardinality with very minimal errors	MySQL usage is highly effective. Multiple tables demonstrate thorough normalization. Relationships between tables are well-established, with minor improvements possible. ERD has perfect normalization and perfect relationship and cardinality	**/10**
Total Marks		**20**						**/100**

Note: Any work submitted after the deadline's time will be subject to the standard University late penalties unless an extension has been granted, in writing by the Senior Tutor, in advance of the deadline. Details on the University's late penalties can be found here.