Nicole Lawrence
Database Design & Development
CS225 Week1 Ponder Assignment

Report on IDE and CLI Use with Consideration of the MySQL Command Line

It is an accepted fact that a programmer must use either an integrated development environment (IDE) or a command line interface (CLI) in the workplace. While many software engineers favor abandoning the more antiquated CLIs in favor of modern IDEs, a common practice within the programming community is to 'use what works best'. It is best to carefully consider each with your project requirements in mind to determine which to use.

An IDE, at its heart, is a command line dressed up in a graphical user interface (GUI). Early IDEs either specialized in a single language or generalized enough to moderately handle multiple languages. Modern IDEs are more versatile and robust, able to take on multiple languages with their specific requirements. This is partially due to open-source extensions. As well, increased storage and processing capacity in modern computers have allowed companies to load more features into their base IDE versions.

A beginning programmer will be familiar with most of the icons and menu designs in an IDE, having seen similar interfaces in operating systems and native programs while growing up. Furthermore, they will not have to memorize syntax and commands to effectively use it. Furthermore, the interface can be customized for optimal user efficiency. Other benefits of an IDE are multitasking and effortless remote networking. While an IDE seems ideal, there are drawbacks created by the GUI wrapper: it requires more memory, is slower since it relies on icons and mouse input, and needs more steps to access files and complete tasks. These drawbacks are virtually nonexistent with smaller projects, but they will adversely affect more complex projects.

While a CLI is not as visually appealing as its IDE counterpart, its banality is its best feature. Stripped of any GUI, the CLI directly connects the developer to the computer. Incidentally, this also results in faster run time and less memory usage. The learning requirements of CLI syntax and commands can be intimidating. However, once learned, the developer can utilize the CLI to interact in multiple languages, faced with an identical interface, syntax, and commands for each. Multitasking is, admittedly, limited within the CLI. Nevertheless, greater precision in creating and executing commands and more fine-tuned control over the system make this deficiency almost negligible. The CLI also offers the ability to script series of instructions instead of having to insert commands line by line.

With the value of the CLI clear, the prudent developer accessing an SQL database will favor the MySQL CLI over MySQL Workbench IDE. The MySQL CLI allows direct interaction with the database with faster response time, enabling the developer to efficiently complete assigned tasks with a high degree of accuracy.

References:

https://www.educba.com/gui-vs-cli/

https://hackernoon.com/why-you-should-be-developing-on-the-command-line-e3353bbdaf51

https://stackoverflow.blog/2020/11/09/modern-ide-vs-vim-emacs/ (an argument for modern IDEs)