EXPLORE AI ACADEMY

SQL in production

Notebooks as an IDE

What are IDEs and Python notebooks?

Integrated Development Environments (IDEs) are a collection of tools used for **software development**, while Python **notebooks** are a collection of tools for interactive **data analysis** and **visualisation**.

IDEs

Text editor interface for coding, debugging, and testing.

Often support **multiple languages** and come with built-in tools for **version control** and collaboration.

Additional **plugins** support software development tasks, including **building**, **compiling,** and **deploying** applications.

Notebooks

Interactive interface consisting of **cells** used for data analysis, visualisation, and documentation.

Libraries are imported for a range of functionalities, including **visualisation**, **modelling**, and more.

Supports the development of **tutorials**, **research,** and **storytelling** through data.

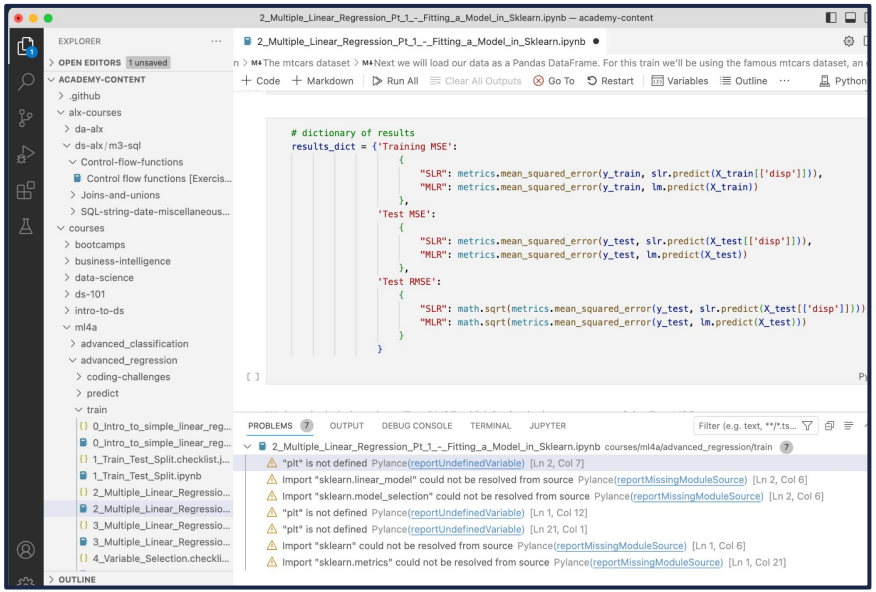
IDEs

> IDEs are a set of ‘traditional’ tools that help users develop software.

An IDE, or Integrated Development Environment, is a **digital toolkit for computer programmers**. It provides an engaging and organised environment for **writing, testing,** and **managing code**.

They are a one-stop shop where programmers can **create their code, edit it, debug it,** and even **run their programs**, all in one convenient location.

They often also incorporate functionality such as **code editors, debugging tools,** and **project management capabilities** to make the software development process more efficient.

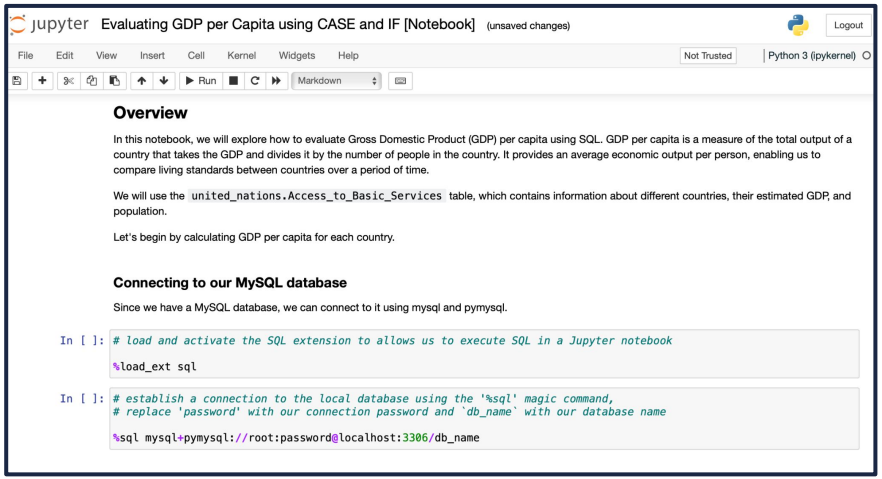


Notebooks

> Notebooks are used for **processing and visualising data**, where sections of code can be executed separately, with supporting text to describe what is happening.

In the context of coding, a notebook is similar to **a digital notepad** or **journal**. It is a software tool or platform that enables you to **create and organise documents that contain a combination of text, code,** and **visual elements.** These notebooks are often employed for **data analysis, scientific research,** and **interactive coding projects.**

Think of a notebook as a **versatile digital notebook** where you can **write notes, perform calculations, run code,** and **visualise data**, all in one place.



Notebooks vs IDEs

Notebooks and IDEs

Both notebooks and IDEs are tools for **creating** and **running code**, with IDEs focusing on traditional software development tools while notebooks are geared towards data analysis, research, and interactive documentation.

The **choice** between them is determined by the **individual task** and **user requirements**.

Synergy

In a software development project, **IDEs** and **notebooks** can **complement** one another.

Notebooks can be used for **data exploration** and **prototyping** to quickly test ideas and visualise results, as well as for documenting thought processes, experiments, and findings.

IDEs can be used for **rigorous testing and debugging** of software components, and their version control features can be used to **manage code revisions and collaboration**.

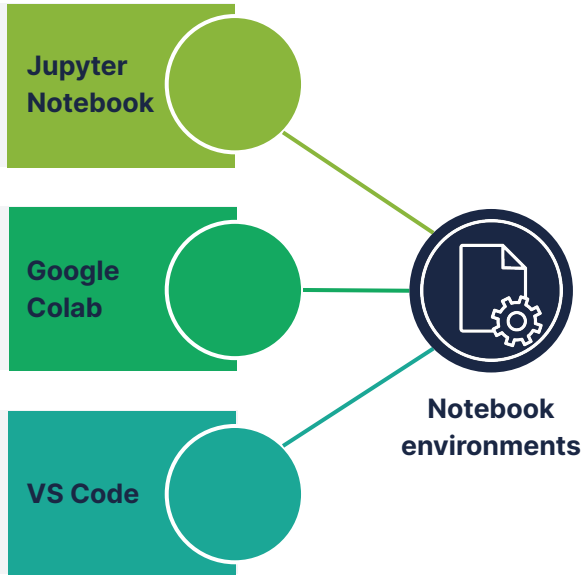
Notebook environments

> Anaconda, Jupyter Notebook, and Google Colab are **tools** that streamline data analysis in interactive notebook environments.

Jupyter Notebook is an open-source **web application** for creating documents with code, visualisations, and text.

Google Colab is a **cloud-based platform** provided by Google that allows users to write, run, and share code in an interactive and collaborative environment.

VS Code (Visual Studio Code) is a **versatile desktop application** that can be used for code development and data exploration with notebooks



SQL in production

Jupyter Notebook

> Jupyter Notebook runs as a **web application** where we can **mix code, visuals,** and **text** in one document. It allows us to generate and collect **real-time results**, making learning and experimentation very intuitive.

01. Code

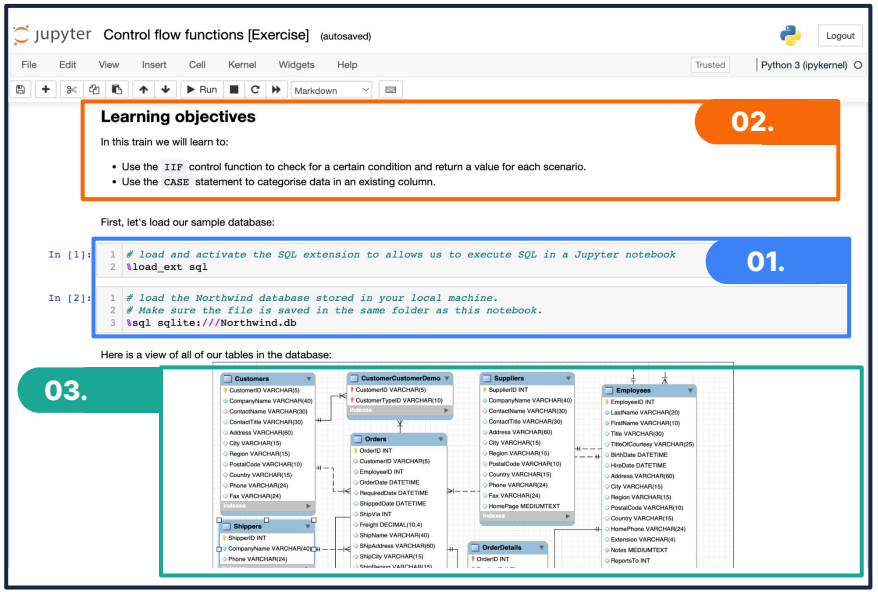
We can **write** and **execute code** in **small**, **manageable** chunks called cells, making it easier to experiment and iterate on ideas.

02. Text

These cells can be interspersed with **rich-text** explanations, allowing us to provide **context**, **documentation**, and **insights** alongside our code.

03. Visualisations

By embedding charts, graphs, and interactive plots alongside our code, we can **illustrate our findings** and **convey complex information** in an accessible manner.



Google Colab

> Google Colab is a **cloud-based** platform where you can write and run code. It offers free access to powerful machines (**Virtual Servers**) and integrates easily with **Google Drive** as a data source.

01. Powerful computing and storage

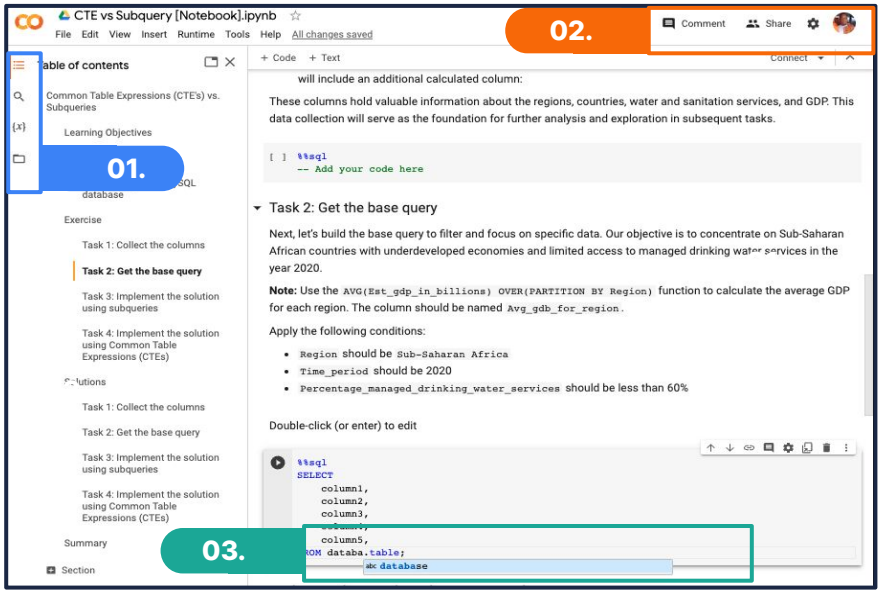
High-performance servers are available for **data processing** and **modelling**, with seamless Google Drive integration.

02. Collaborative environment

Enables **real-time collaboration** with team members through shareable links, comments, and more collective editing.

03. Enhanced coding experience

The workspace has some **IDE-like features** including auto complete, syntax highlighting, variable inspector, and the ability to format visualisations.



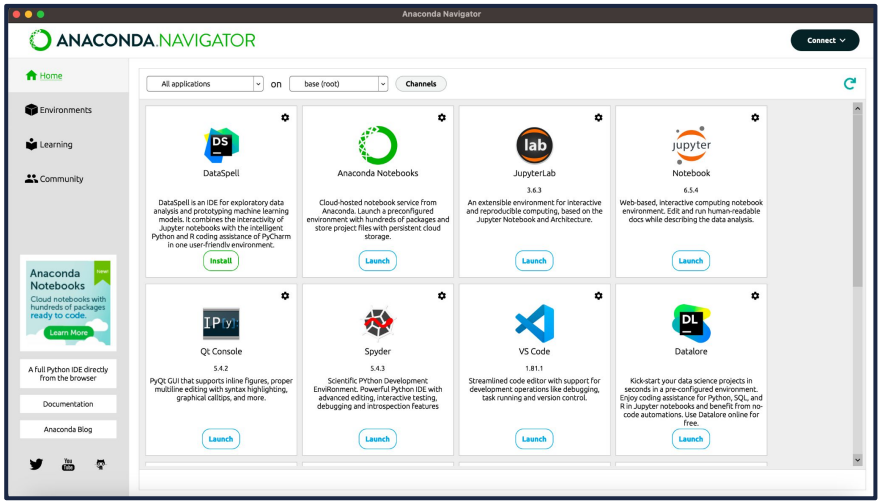
Anaconda

> Anaconda is an **all-in-one** data science toolkit that bundles together many tools for data analysis and science.

Anaconda Navigator is a Graphical User Interface (GUI) included with the Anaconda distribution. It is designed to help users **manage** and **interact** with their Anaconda **environments, packages,** and **projects**.

It comes **pre-packed** with over 1,500 tools and **platforms** such as Jupyter, Pandas, and Matplotlib, which helps with the use and management of these tools in one central place.

Anaconda can be installed on different operating systems (e.g. Mac or Windows) and environments (a computer or cloud provider) making it **platform independent**.

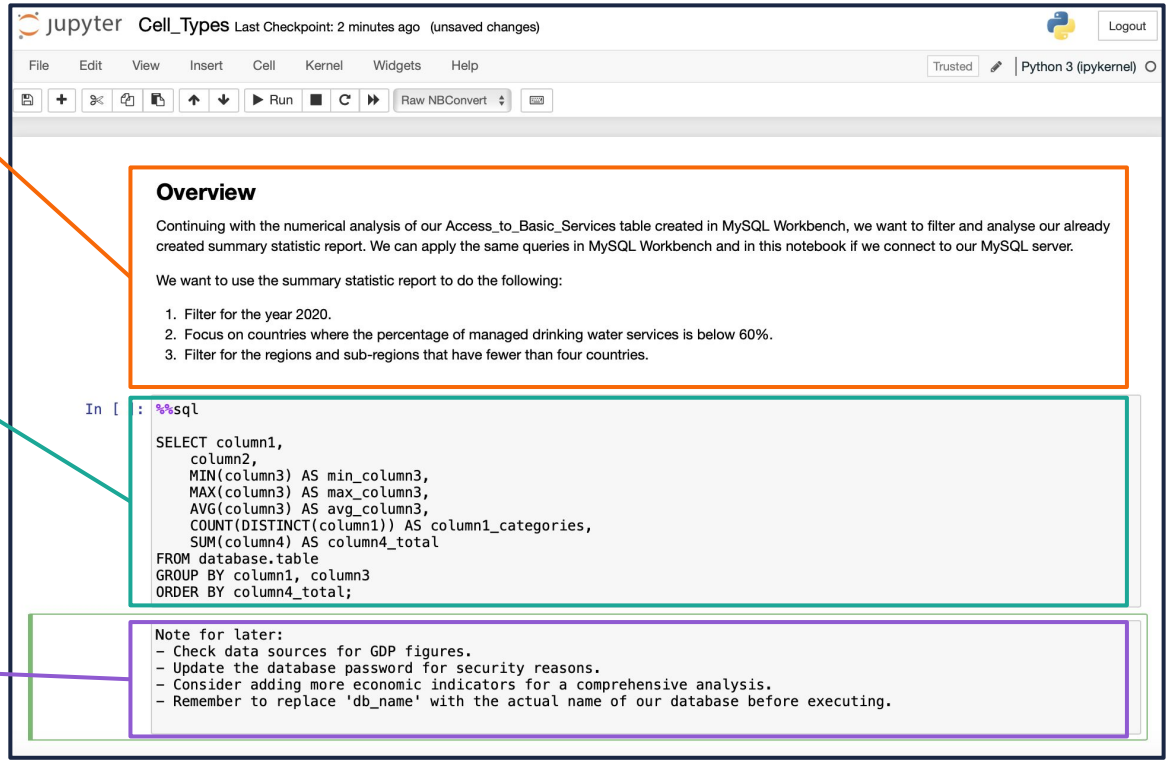


Notebook cell types

Markdown cells: Used for documentation. We can write text, add headers, and format the content to explain code.

Code cells: Used for input commands and scripts. When executed, the notebook processes the code and displays the output.

Raw cells: Unprocessed content. It's a space to input data or notes without the notebook interpreting or executing it.



Other programming languages used in notebooks

Notebook environments like Jupyter Notebook and JupyterLab **support a variety of programming languages, not just SQL**. Some commonly supported programming languages in Jupyter Notebook include:

Python: One of the most popular languages for data science and general-purpose programming used for data analysis, machine learning, and scientific computing.

R: Commonly used for statistical analysis, data visualisation, and machine learning.

JavaScript: Useful for web development and interactive data visualisation tasks.

These are just some examples of the many languages we can use in Jupyter Notebook. This versatility makes Jupyter Notebook a powerful tool for various programming and data analysis tasks, and it enables **polyglot programming**, where we can combine **multiple languages in a single notebook.**

Notebooks help us **combine the capabilities** of SQL, Python, and other languages to perform data exploration and analysis.