



Politecnico
di Torino



Live Coding

Development with Streamlit

Objectives

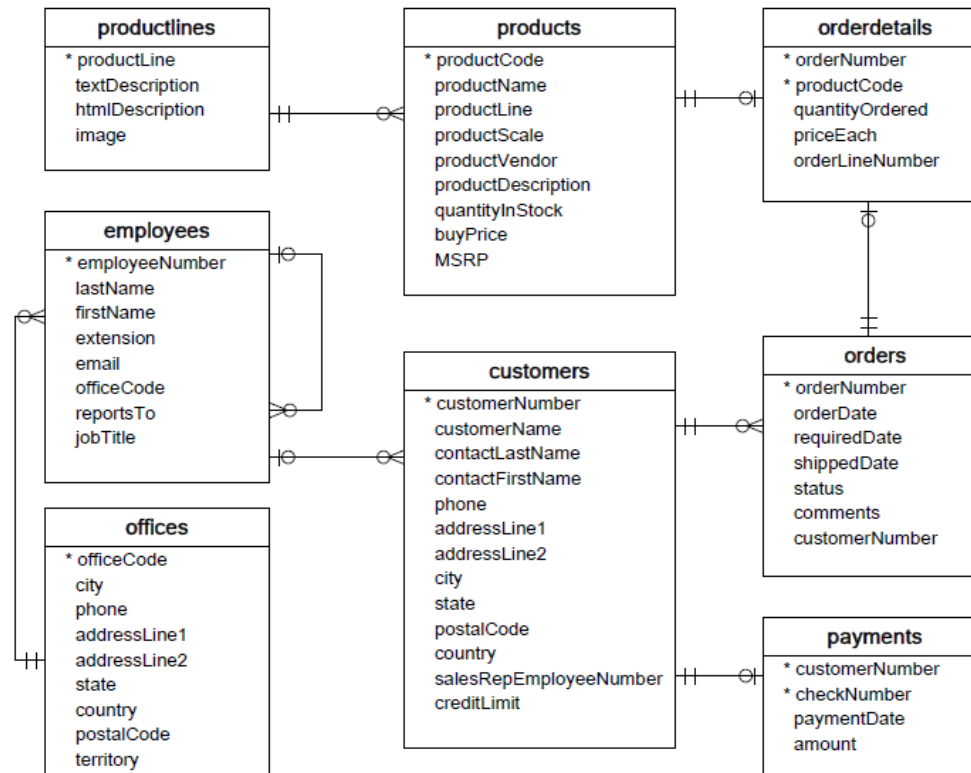
- Create a **multipage** application in Streamlit step-by-step
- Implement and learn about the **main** Streamlit **widgets**
- Integrate **SQL queries** into a web application to view and update data from a MySQL database
- Structure a project using the main development tools (i.e., GitHub, Docker)

Requirements:

- Install Git/GitHub
- Install Docker/Docker Compose
- Clone repositories

Database

- Database of a scale model car dealer
- This is a sample MySQL database called *classicmodels*
- It collects information about products, employees, customers, orders...



Git and GitHub



- Git is a distributed **version control system** and records changes to files over time
- Allows multiple people to work on the same project at the same time
- GitHub is a web platform that uses Git for **repository** management: directories that collect the code of software projects
- Allows to collaborate, to track changes, and to manage issues

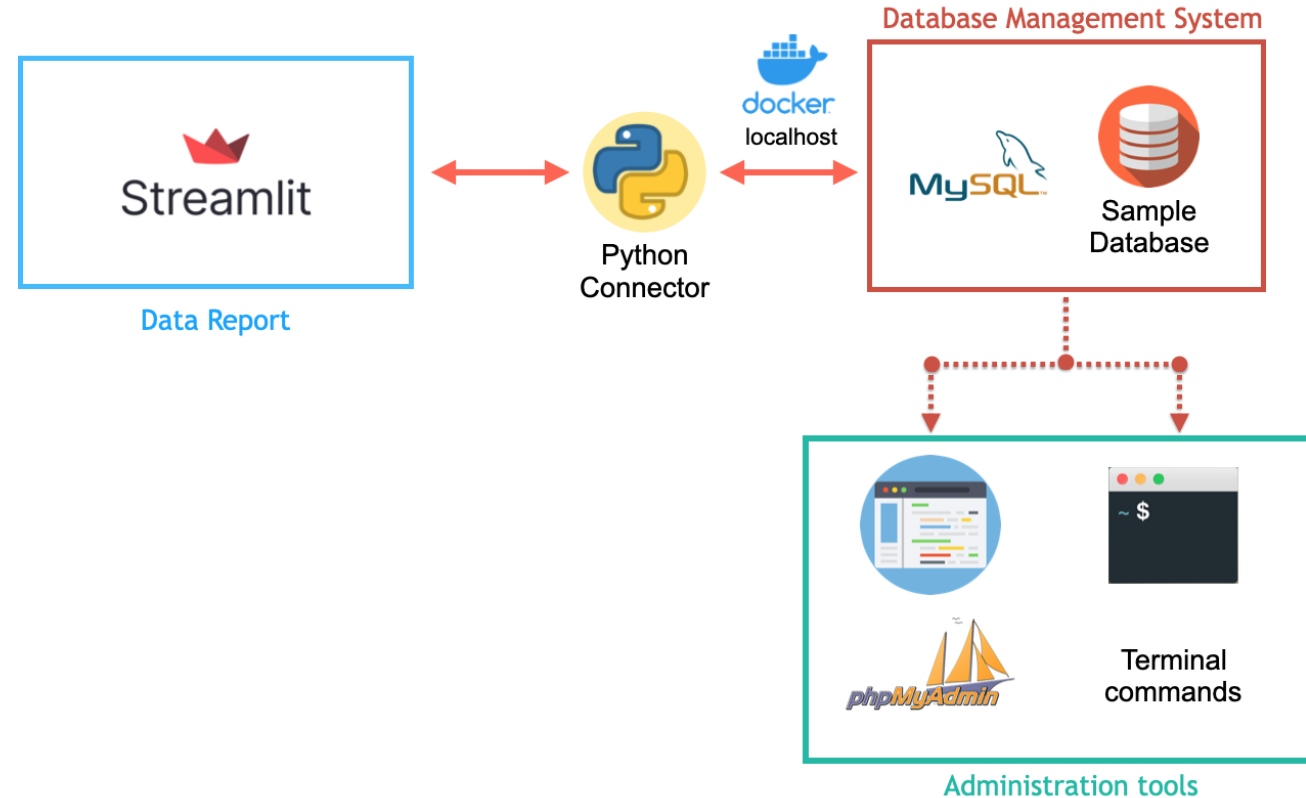


- **Commit:** represents a set of changes to the project files, creating a new version recorded in the history
- **Pull e Push:** commands to synchronize local repositories with remote repositories. *Pull* to download updates remotely and *Push* to apply local changes to the remote repository
- **Branch:** separate branch that allows you to work in parallel to multiple isolated changes of the main branch
- **Merge:** the process of combining the contents of one branch into another

- **Docker:** open-source platform that simplifies the deployment and execution of applications through **containers**
- A container is a unit that contains everything you need to start your application (e.g., code, libraries)
- Containers are lightweight and portable, ensuring reliability in different development environments
- **Docker Compose:** used to define and manage multi-container applications
- Is configured with a YAML file

Project Overview

- The repository [mysql-docker](#) contains the MySQL database already configured (*localhost:3306*) and the phpMyAdmin web interface (*localhost:8081*)
- The repository [streamlitTutorial](#) contains the Streamlit project that will interact with the database



mysql-docker Repository

- The *docker-compose.yml* defines the MySQL and phpMyAdmin containers
- The *data/mysqlsampledatabase.sql* file contains SQL code
- The *.env* file contains the credentials (it is good practice to change them)

```
Code Blame 27 lines (25 loc) · 483 Bytes
1  version: '3'
2
3  services:
4    db:
5      image: mysql:latest
6      container_name: db
7      env_file:
8        - .env
9      volumes:
10       - dbdata:/var/lib/mysql
11       - ./data/mysqlsampledatabase.sql:/docker-entrypoint-initdb.d/start.sql
12      ports:
13       - 3306:3306
14    phpmyadmin:
15      container_name: phpmyadmin
16      #platform: linux/amd64
17      image: phpmyadmin:latest
18      restart: always
19      ports:
20       - "8081:80"
21      environment:
22        PMA_HOST: db
23      depends_on:
24       - db
25
26  volumes:
27    dbdata:
```

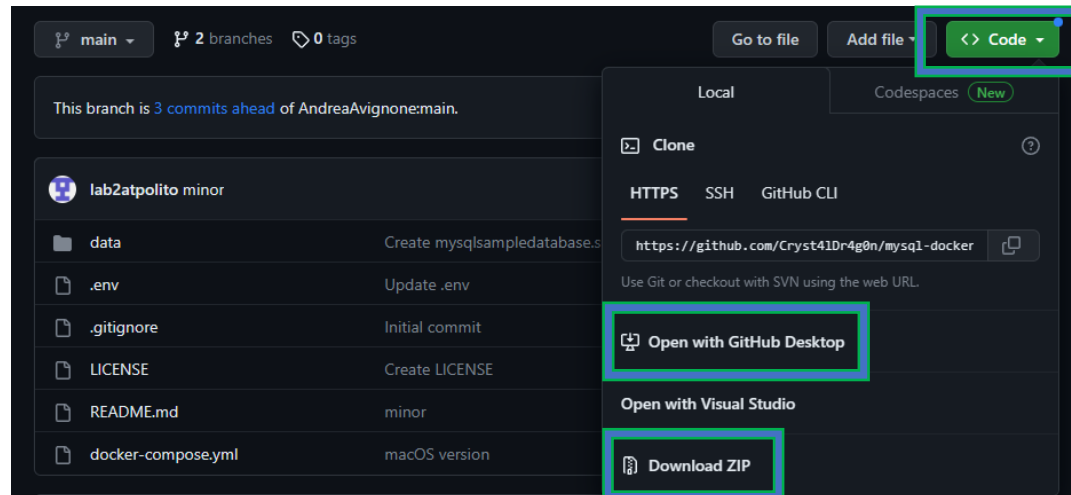
docker-compose.yml

```
Code Blame 4 lines (4 loc) · 99 Bytes
1  MYSQL_USER=student
2  MYSQL_ROOT_PASSWORD=mypassword
3  MYSQL_PASSWORD=user_pwd
4  MYSQL_DATABASE=liveCoding
```

.env

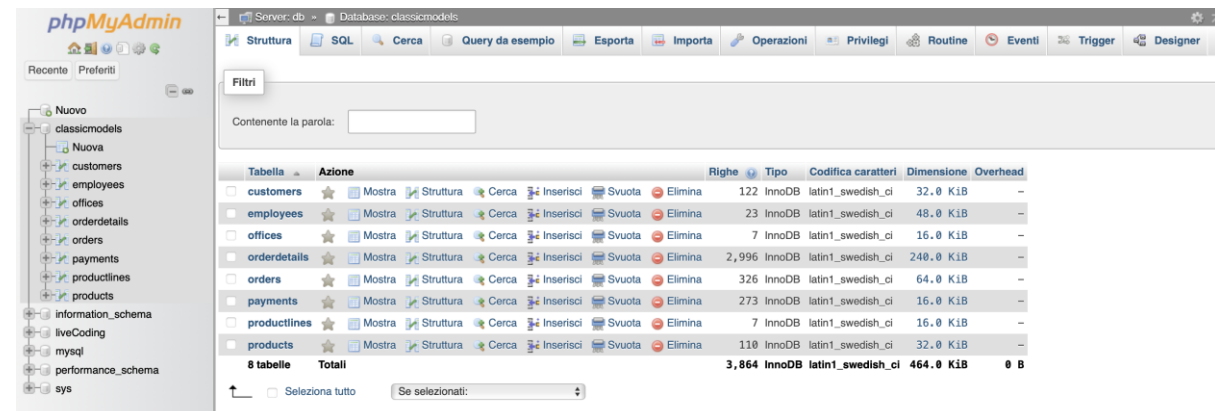
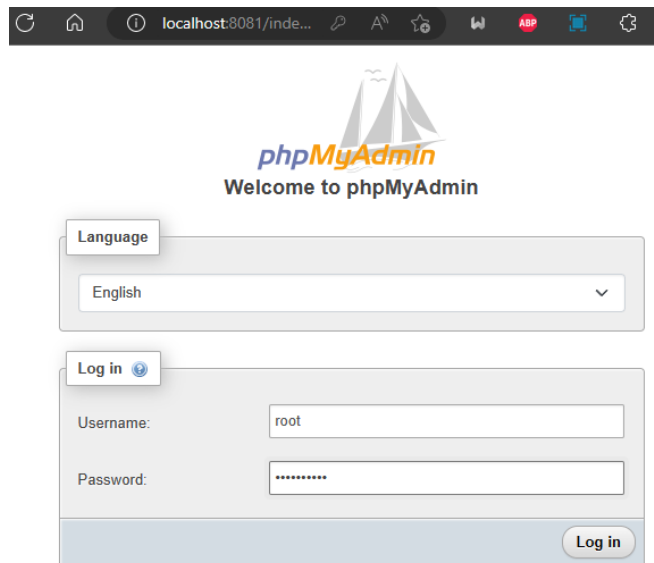
mysql-docker Repository

- Each repository has a **README.md** that provides essential information about the project to understand and use it
- Once git is installed, the repository can be cloned from the terminal:
 - git clone <https://github.com/Cryst4lDr4g0n/mysql-docker-eng.git>
- Or by downloading the zip file or opening it in the GitHub Desktop application



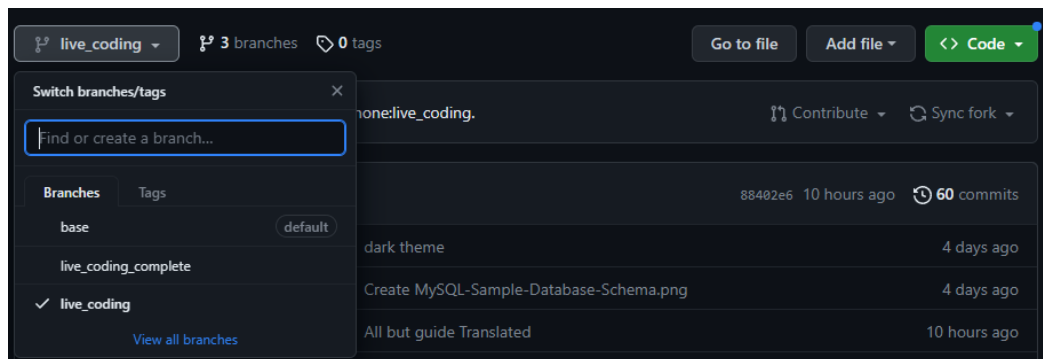
mysql-docker Repository

- Once you start Docker, in the *mysql-docker* folder, type the command to start the containers (-d to start them in the background):
 - `docker-compose up -d`
- To stop containers:
 - `docker-compose down`
- (Optional) enter phpMyAdmin on *localhost:8081*

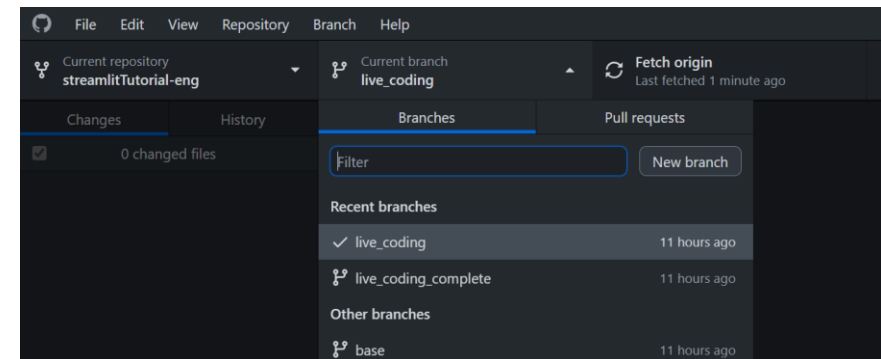


streamlitTutorial Repository

- git clone <https://github.com/Cryst4lDr4g0n/streamlitTutorial-eng.git>
- Consisting of 3 branches:
 - **base**: structure of a generic multi-page project (can be used as a starting point for laboratory, notebook)
 - **live_coding**: starting point for live coding
 - **live_coding_complete**: complete final application
- To change branches from terminal:
 - git checkout *branch_name*
- Or:



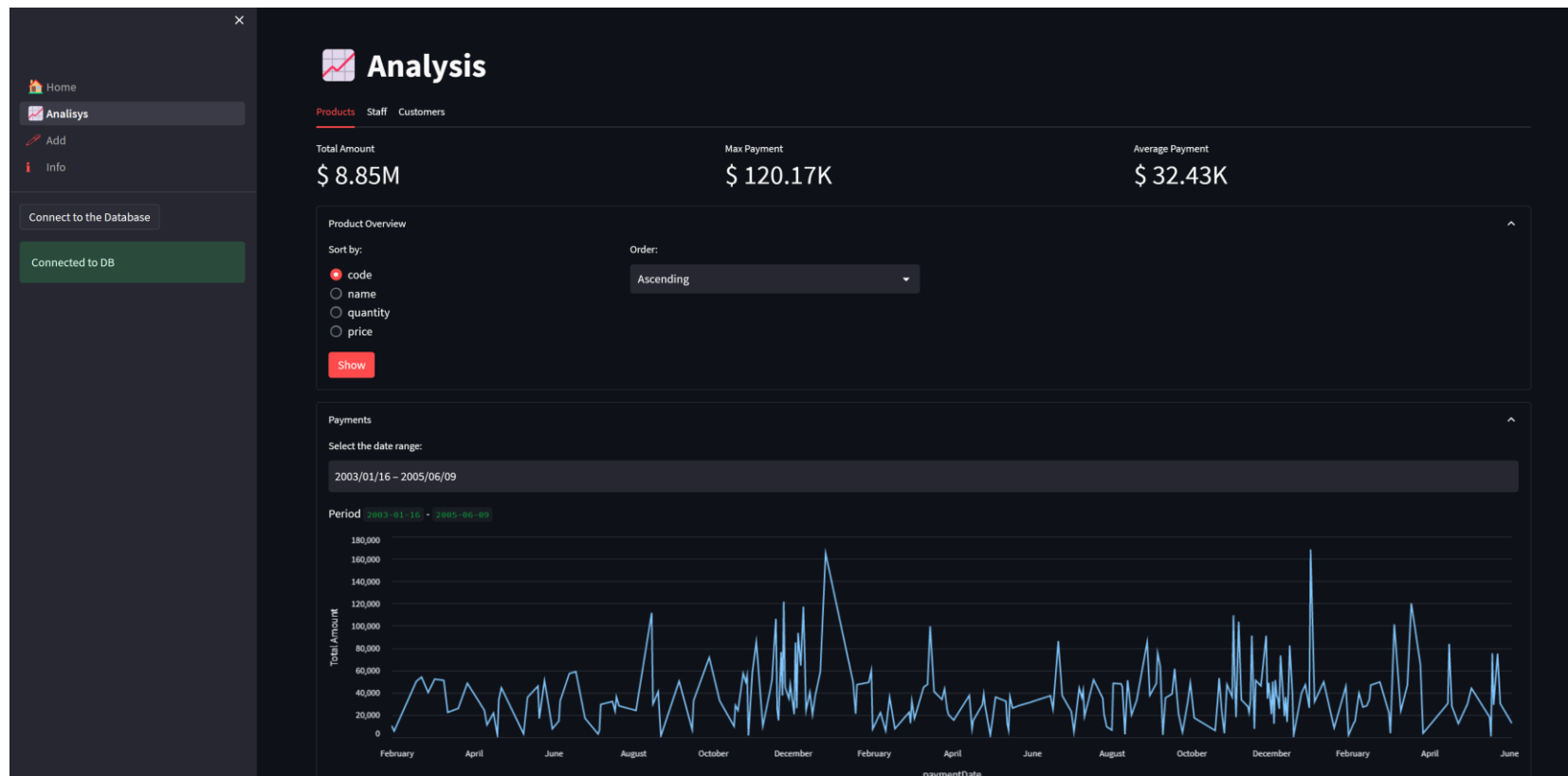
GitHub web



GitHub Desktop

Dashboard

- Create a final dashboard with visualization and data addition function
- 4 pages will be developed: Home, Analysis, Add, Info
- Includes user input widgets to customize queries and data display



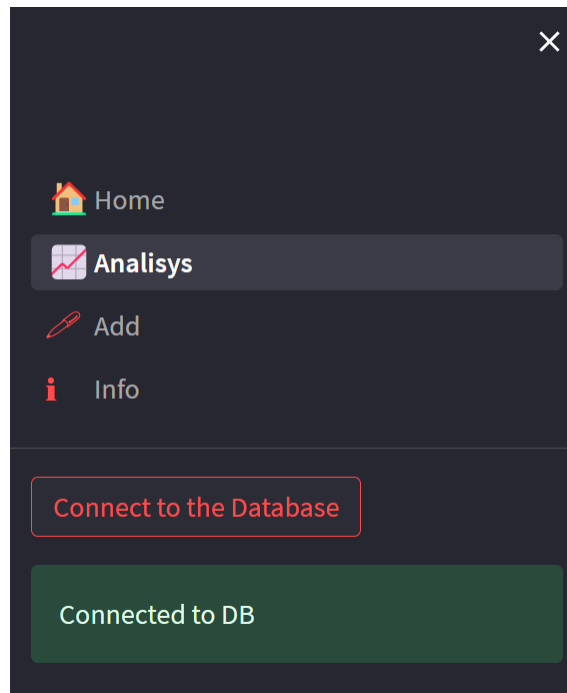
Homepage Customization

- Insert text elements in custom markdowns on the homepage
- Add the Polito logo
- Initialize the *session* state



Connecting to the DB

- Add functions to connect to the DB and execute queries
- Add a function that checks the status of the connection
- Add a button on the sidebar to connect and print the result (success or error) on the screen



Product analysis

- Add to the *Products* tab of the *Analysis* page:
 - **3 Metrics:** total amount, maximum payment, average payment;
 - **2 Expanders:** product overview, payments;
 - **3 Input utente:** radio button (product attribute to sort), select box (ascending or descending) and date range (interval in which to view the progress of payments)

Product Overview

Sort by:

☐ code

☐ name

☒ quantity

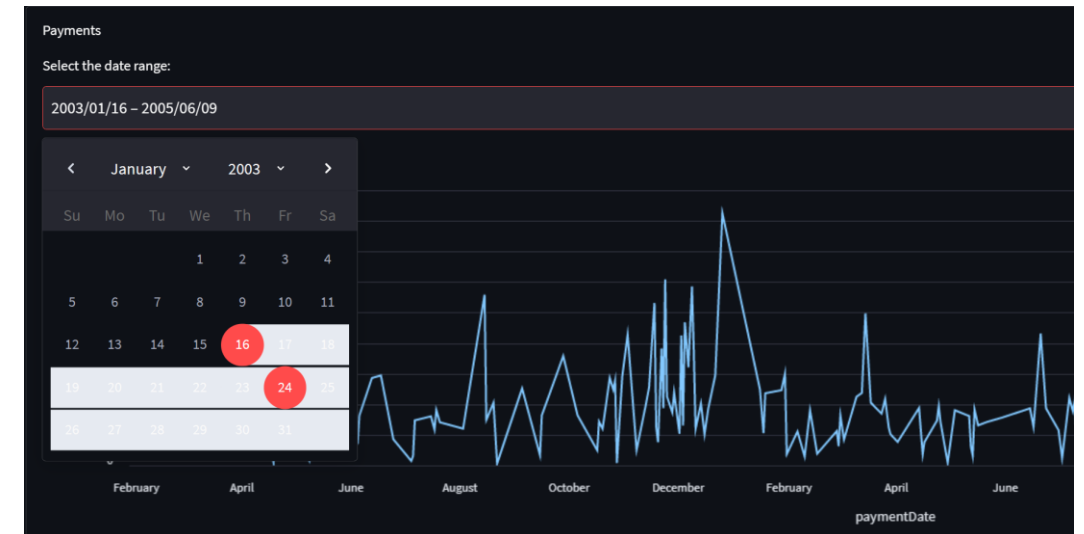
☐ price

Show

Order:

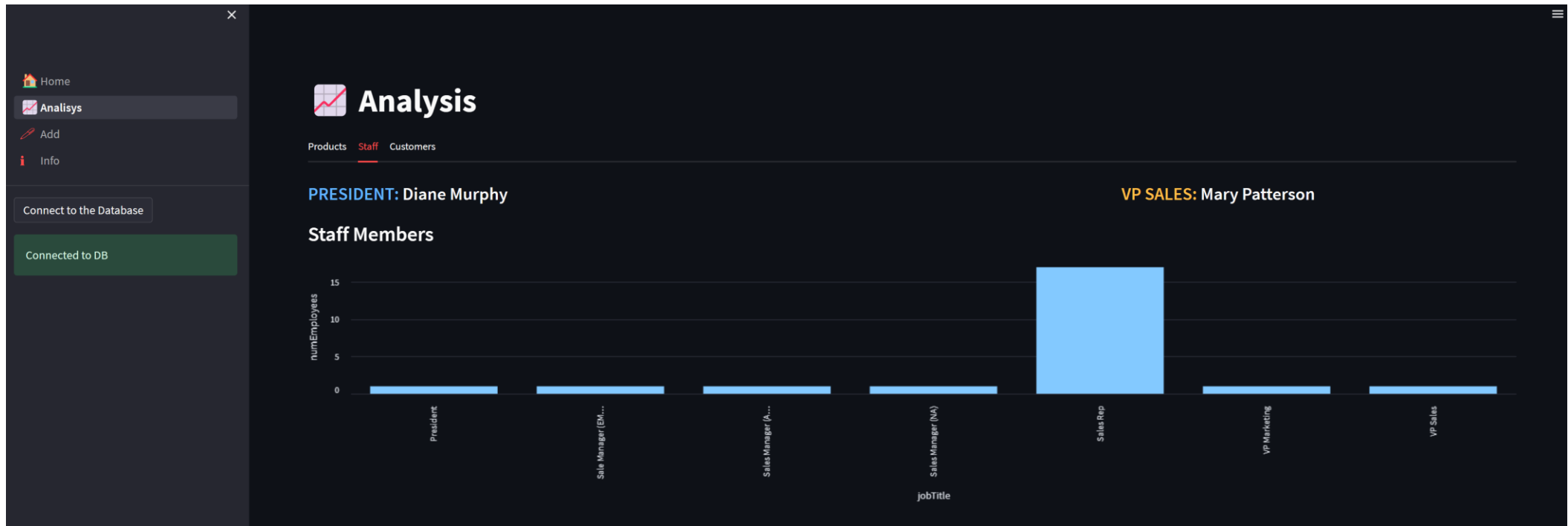
Descending

	code	name	quantity	price	MSRP
0	S12_2823	2002 Suzuki XREO	9,997	66.27	150.62
1	S18_1984	1995 Honda Civic	9,772	93.89	142.25
2	S700_2466	America West Airlines B757-300	9,653	68.80	99.72
3	S24_3432	2002 Chevy Corvette	9,446	62.11	107.08
4	S18_2325	1932 Model A Ford J-Coupe	9,354	58.48	127.13
5	S32_2206	1982 Ducati 996 R	9,241	24.14	40.23
6	S24_3151	1912 Ford Model T Delivery Wagon	9,173	46.91	88.51
7	S18_3482	1976 Ford Gran Torino	9,127	73.49	146.99
8	S12_3380	1968 Dodge Charger	9,123	75.16	117.44
9	S18_1589	1965 Aston Martin DB5	9,042	65.96	124.44



Staff analysis

- View the first and last names of ***President*** and ***VP Sales***
- Represent with a **bar chart** the distribution of employees in the various roles



Customer analysis

- View customer information
- Insert a table that describes the origin of customers, grouping them by country and sorting them by number
- Insert a table showing U.S. customers with *creditLimit*>100000

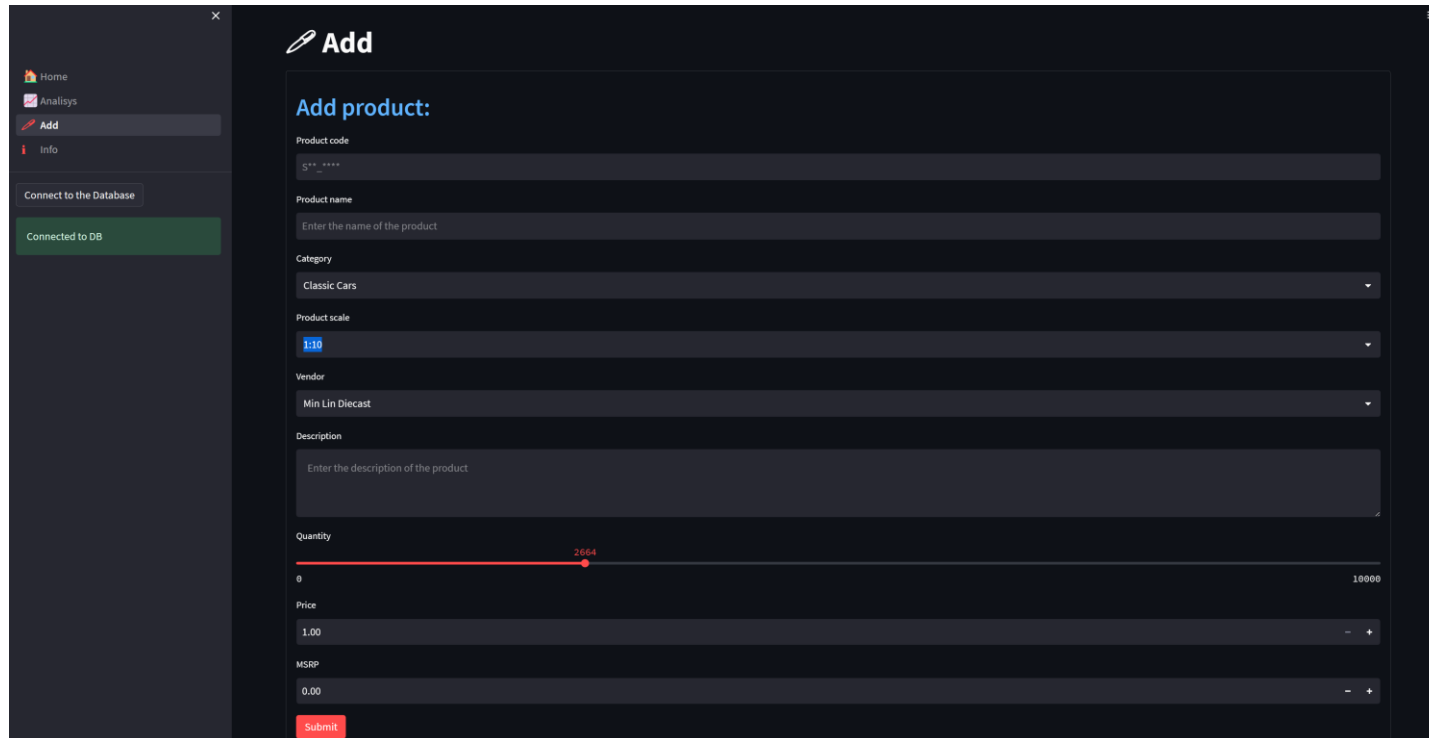
The screenshot shows a web application interface with a dark theme. On the left is a sidebar with navigation links: Home, Analysis (selected), Add, and Info. Below these is a 'Connect to the Database' button and a green status bar indicating 'Connected to DB'. The main content area is titled 'Analysis' and has three tabs: Products, Staff, and Customers (selected). Under the 'Customers' tab, there are two tables. The first table, 'Worldwide customers distribution', shows customer counts by country. The second table, 'Customers with higher credit limit in the USA', shows details for high-credit-limit customers in the US.

	customersNumber	country
0	36	USA
1	13	Germany
2	12	France
3	7	Spain
4	5	UK
5	5	Australia
6	4	New Zealand
7	4	Italy
8	3	Norway

	customername	state	creditLimit
0	Mini Gifts Distributors Ltd.	CA	210500.00
1	Muscle Machine Inc	NY	138500.00
2	Marta's Replicas Co.	MA	123700.00
3	Land of Toys Inc.	NY	114900.00
4	Online Diecast Creations Co.	NH	114200.00
5	Collectable Mini Designs Co.	CA	105000.00
6	Corporate Gift Ideas Co.	CA	105000.00
7	Mini Classics	NY	102700.00
8	Diecast Classics Inc.	PA	100600.00

Adding a product

- Create a form to add a new product to the database
- Use text insertion, selectbox, slider, and numeric input widgets
- Add a submit button, run the query, and verify that it was successfully executed
- Capture the following parameters: *productCode*, *productName*, *productLine*, *productScale*, *ProductVendor*, *ProductDescription*, *quantityInStock*, *buyPrice*, *MSRP*



The screenshot shows a web application interface with a dark theme. On the left is a sidebar with navigation links: Home, Analysis, Add (highlighted), and Info. Below these are buttons for 'Connect to the Database' and a green status bar indicating 'Connected to DB'. The main content area is titled 'Add' and contains a form titled 'Add product:'. The form fields are: 'Product code' (text input), 'Product name' (text input with placeholder 'Enter the name of the product'), 'Category' (dropdown menu showing 'Classic Cars'), 'Product scale' (dropdown menu showing '1:10'), 'Vendor' (dropdown menu showing 'Min Lin Diecast'), 'Description' (text area with placeholder 'Enter the description of the product'), 'Quantity' (slider ranging from 0 to 10000 with a value of 2664), 'Price' (numeric input with value 1.00), and 'MSRP' (numeric input with value 0.00). A red 'Submit' button is at the bottom of the form.