



Politecnico
di Torino



Live Coding

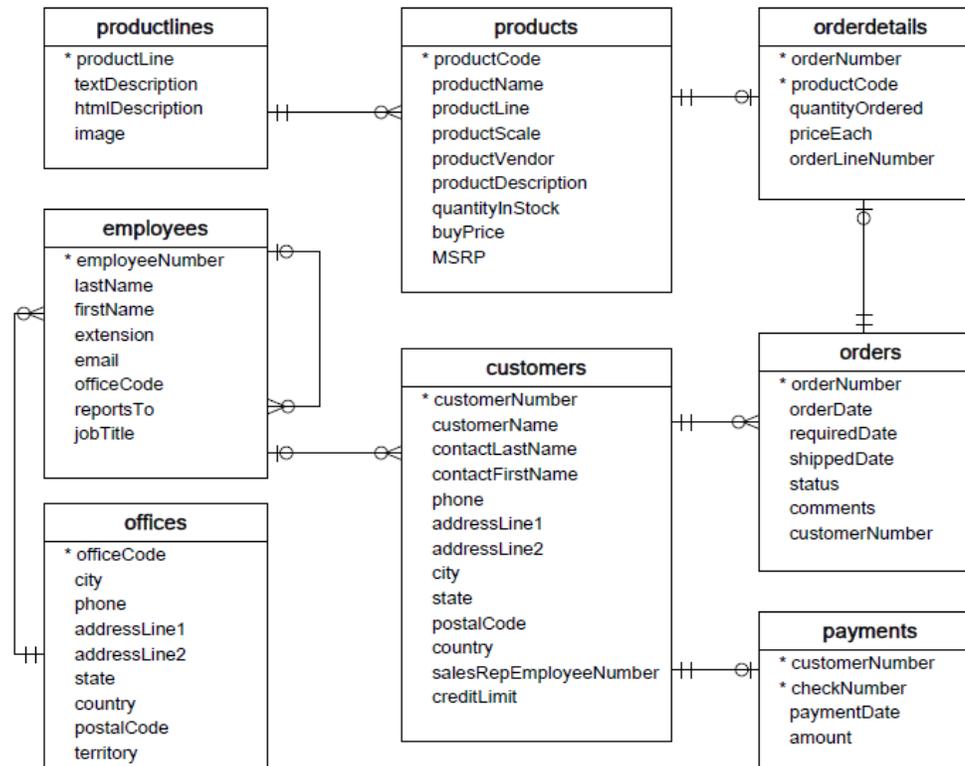
Sviluppo con Streamlit

Obiettivi

- Creare un'applicazione **multipage** in Streamlit step-by-step
- Implementare e conoscere i **principali widget** Streamlit
- Integrare le **interrogazioni SQL** in una web application per visualizzare e aggiornare i dati di un database MySQL
- Strutturare un progetto utilizzando i principali strumenti di sviluppo (i.e., GitHub, Docker)
- **Requisiti:**
 - Installare Git/GitHub
 - Installare Docker/Docker Compose
 - Clonare i repository

Database

- Database di un rivenditore di modellini in scala di automobili
- Si tratta di un Database MySQL di esempio chiamato *classicmodels*
- Raccoglie informazioni riguardo a prodotti, dipendenti, clienti, ordini...





Git e GitHub

- Git è un **sistema di controllo di versione** distribuito e registra le modifiche ai file nel tempo
- Permette a più persone di lavorare allo stesso progetto contemporaneamente
- GitHub è una piattaforma web che utilizza Git per la gestione dei **repository**: directory che raccolgono il codice dei progetti software
- Permette di collaborare, monitorare modifiche e gestire problemi



Git in a nutshell

- **Commit:** rappresenta un insieme di modifiche ai file del progetto, creando una nuova versione registrata nella cronologia
- **Pull e Push:** comandi per sincronizzare i repository locali con quelli remoti. *Pull* per scaricare gli aggiornamenti da remoto e *Push* per applicare i cambiamenti locali sul repository remoto
- **Branch:** ramo separato che permette di lavorare in parallelo a più modifiche isolate del ramo principale
- **Merge:** processo di combinazione dei contenuti di un branch in un altro

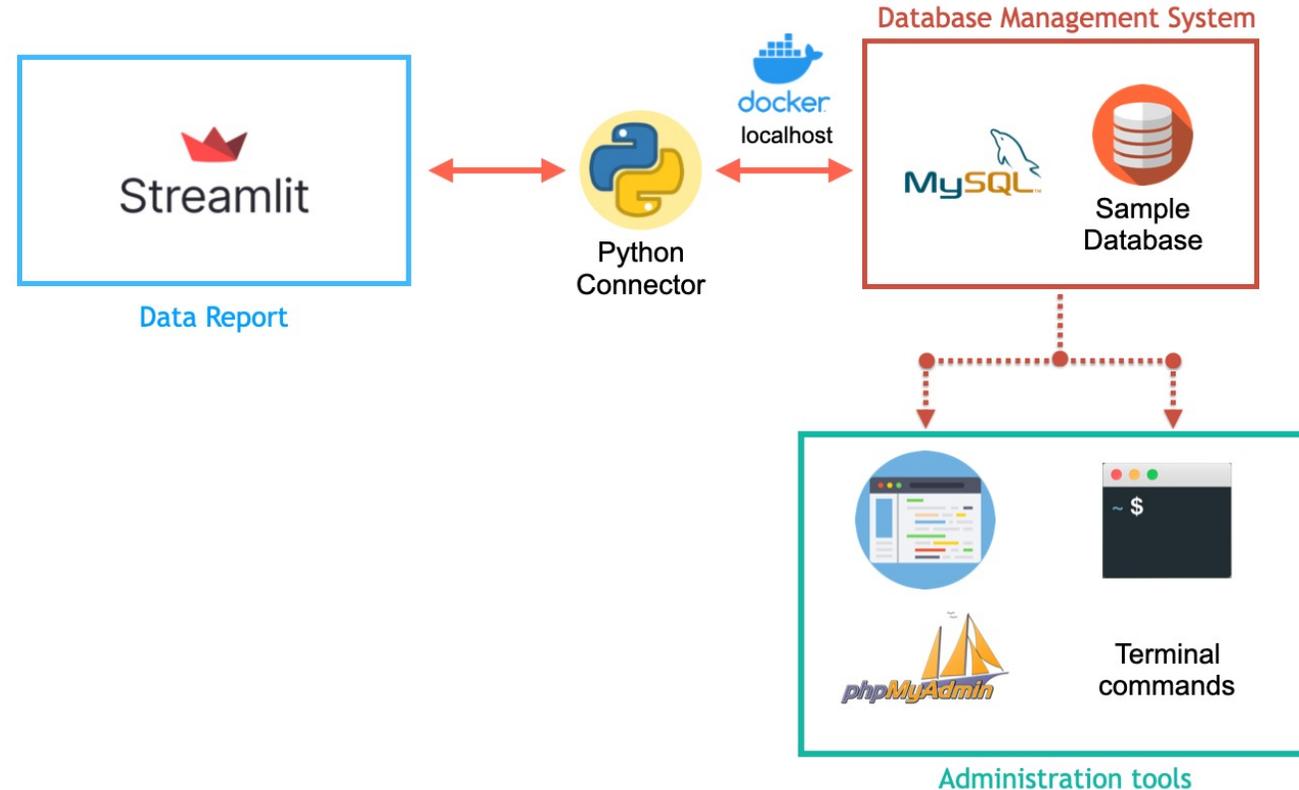
Docker

- **Docker:** Piattaforma open-source che semplifica la distribuzione ed esecuzione di applicazioni attraverso i **container**
- Un container è un'unità che contiene tutto il necessario per avviare l'applicazione (e.g., codice, librerie)
- I container sono leggeri e portabili, garantendo affidabilità in diversi ambienti di sviluppo

- **Docker Compose:** Utilizzato per definire e gestire applicazioni multi-container
- Viene configurato con un file YAML

Panoramica del progetto

- Il repository [mysql-docker](#) contiene il database MySQL già configurato (*localhost:3306*) e l'interfaccia web phpMyAdmin (*localhost:8081*)
- Il repository [streamlitTutorial](#) contiene il progetto Streamlit che interagirà con il database



Repository mysql-docker

- Nel *docker-compose.yml* vengono definiti i container MySQL e phpMyAdmin
- Il file *data/mysqlsampledatabase.sql* contiene il codice SQL
- Il file *.env* contiene le credenziali (è buona norma modificarle)

```
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:
```

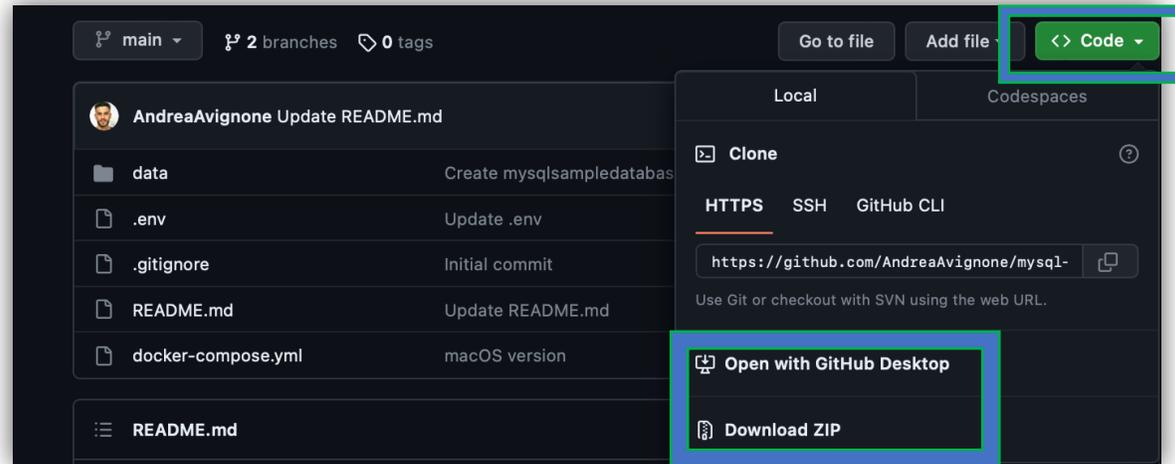
docker-compose.yml

```
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

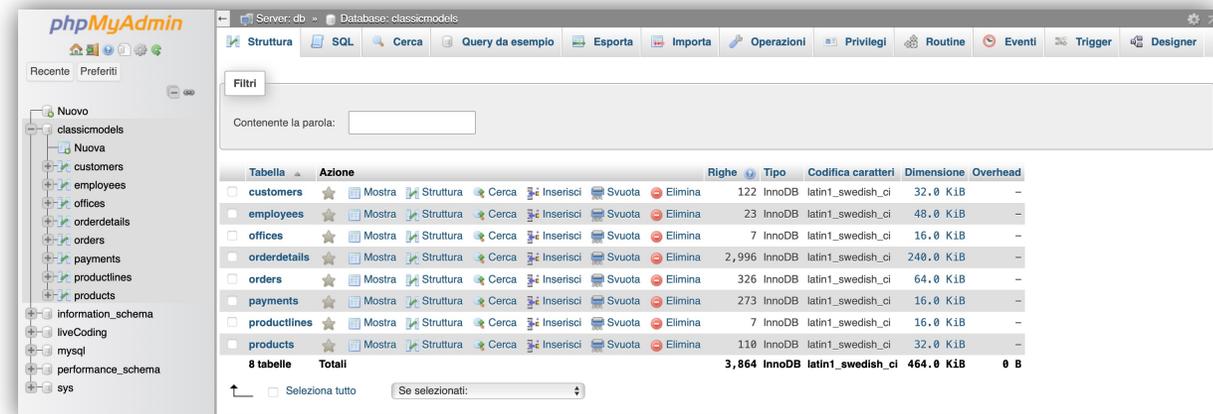
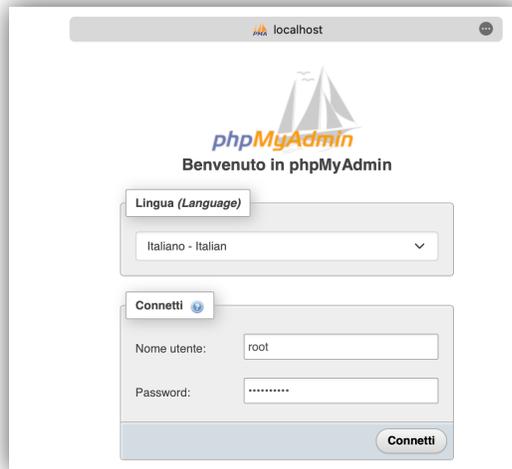
Repository mysql-docker

- Ogni repository ha un **README.md** che fornisce le informazioni essenziali sul progetto per capirlo e utilizzarlo
- Una volta installato git, il repository può essere clonato da terminale:
 - git clone <https://github.com/AndreaAvignone/mysql-docker.git>
- Oppure scaricando il file zip o aprendolo nell'applicazione GitHub Desktop



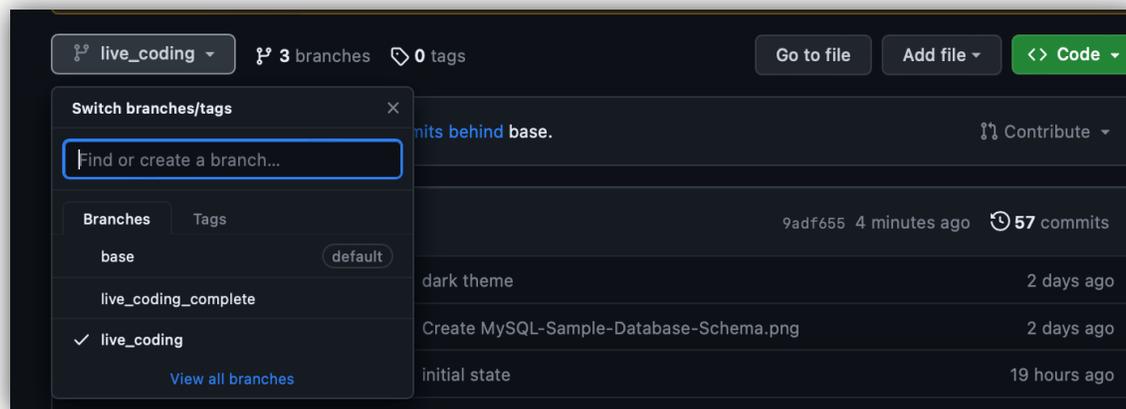
Repository mysql-docker

- Una volta avviato Docker, all'interno della cartella *mysql-docker* digitare il comando per avviare i container (-d per avviarli in background):
 - `docker-compose up -d`
- Per fermare i container:
 - `docker-compose down`
- (Opzionale) entrare in phpMyAdmin su *localhost:8081*

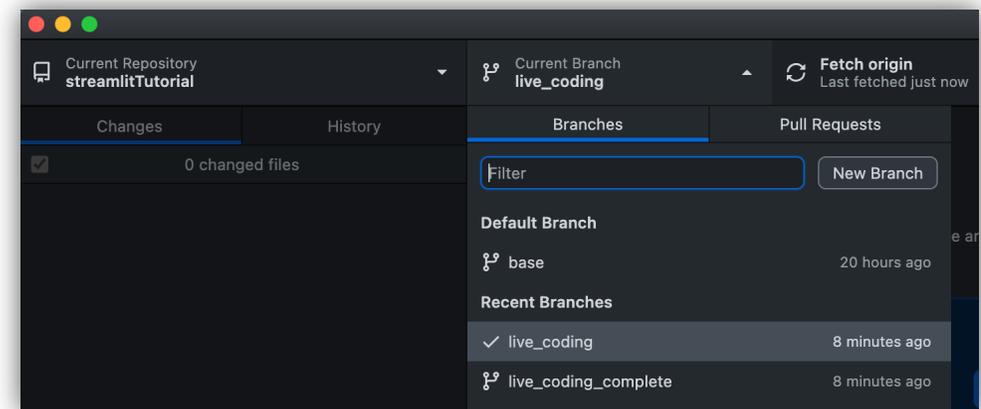


Repository streamlitTutorial

- Formato da 3 branch:
 - **base**: struttura di un progetto multipagina generico (può essere usato come punto di partenza per laboratorio, quaderno)
 - **live_coding**: punto di partenza del live coding
 - **live_coding_complete**: applicazione finale completa
- Per cambiare branch da terminale:
 - `git checkout branch_name`
- Oppure:



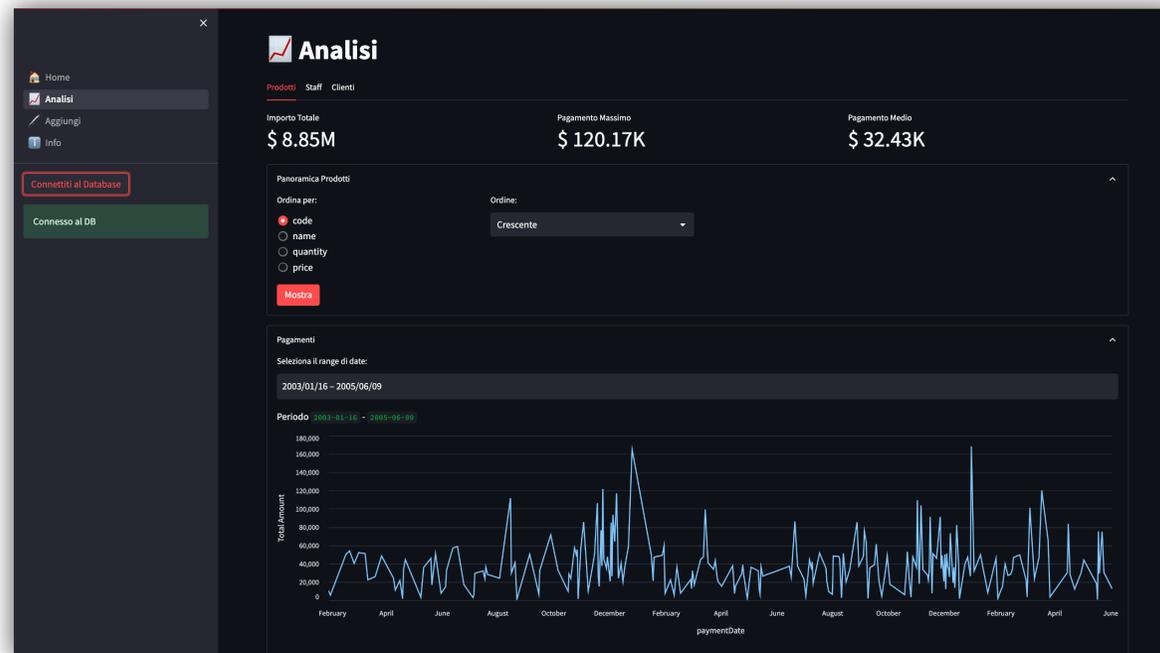
GitHub web



GitHub Desktop

Dashboard

- Creare una dashboard finale con funzione di visualizzazione e di aggiunta dati
- Saranno sviluppate 4 pagine: Home, Analisi, Aggiungi, Info
- Vengono inclusi widget di input dell'utente per personalizzare le interrogazioni e la visualizzazione dati



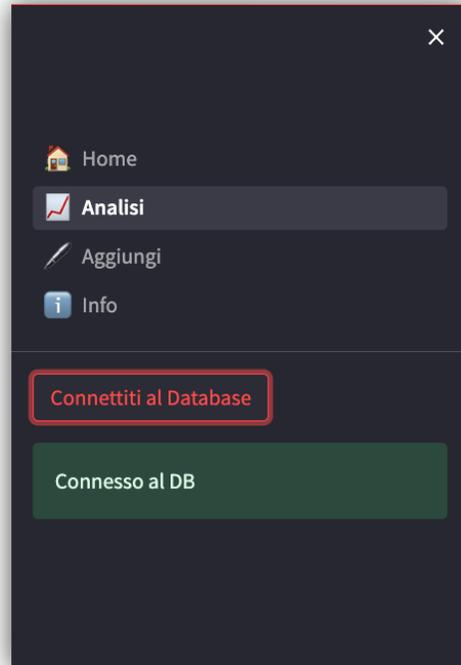
Personalizzazione Homepage

- Inserire nell'homepage elementi di testo in markdown personalizzati
- Aggiungere il logo Polito
- Inizializzare il *session state*



Connessione al DB

- Aggiungere le funzioni per connettersi al DB ed eseguire le interrogazioni
- Aggiungere una funzione che controlli lo status della connessione
- Aggiungere un tasto sulla sidebar per connettersi e stampare a schermo l'esito (successo o errore)



Analisi prodotti

- Aggiungere al tab *Prodotti* della pagina *Analisi*:
 - **3 Metriche**: importo totale, pagamento massimo, pagamento medio;
 - **2 Expander**: panoramica prodotti, pagamenti;
 - **3 Input utente**: radio button (attributo prodotto su cui fare l'ordinamento), select box (crescente o decrescente) e date range (intervallo in cui visualizzare l'andamento dei pagamenti)

Panoramica Prodotti

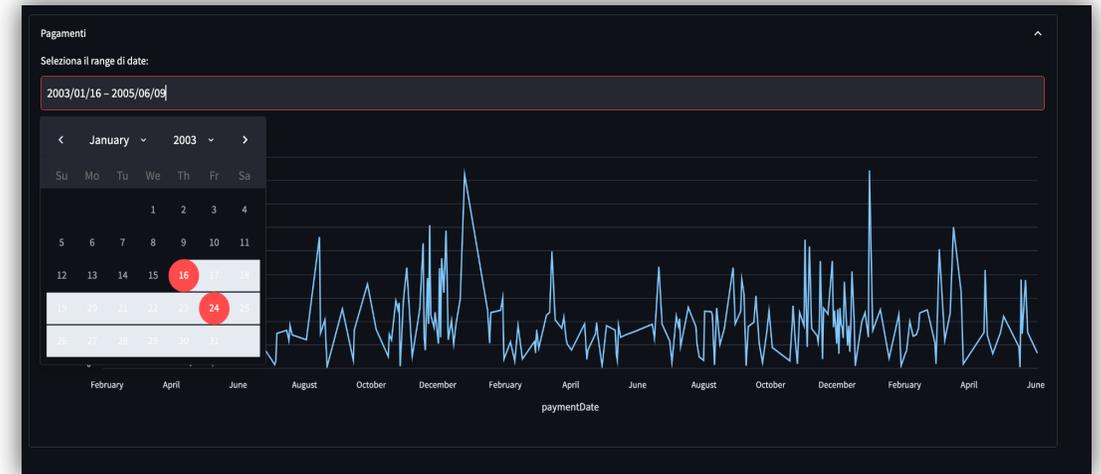
Ordina per:

- code
- name
- quantity
- price

Mostra

Ordine: Decrescente

	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 B757-200	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



Analisi staff

- Visualizzare Nome e cognome di **President** e **VP Sales**
- Rappresentare con un **bar chart** la distribuzione dei dipendenti nei vari ruoli



Analisi clienti

- Visualizzare informazioni riguardanti i clienti
- Inserire una tabella che descriva l'origine dei clienti, raggruppandoli per paese e ordinandoli per il numero
- Inserire una tabella che mostri i clienti statunitensi con *creditLimit*>100000

The screenshot shows a web application interface with a dark theme. On the left is a sidebar with navigation links: Home, Analisi (selected), Aggiungi, and Info. Below these is a 'Connetti al Database' button and a green 'Connesso al DB' indicator. The main content area is titled 'Analisi' and has sub-tabs for 'Prodotti', 'Staff', and 'Clienti' (selected). There are two data tables displayed:

Distribuzione clienti nel mondo

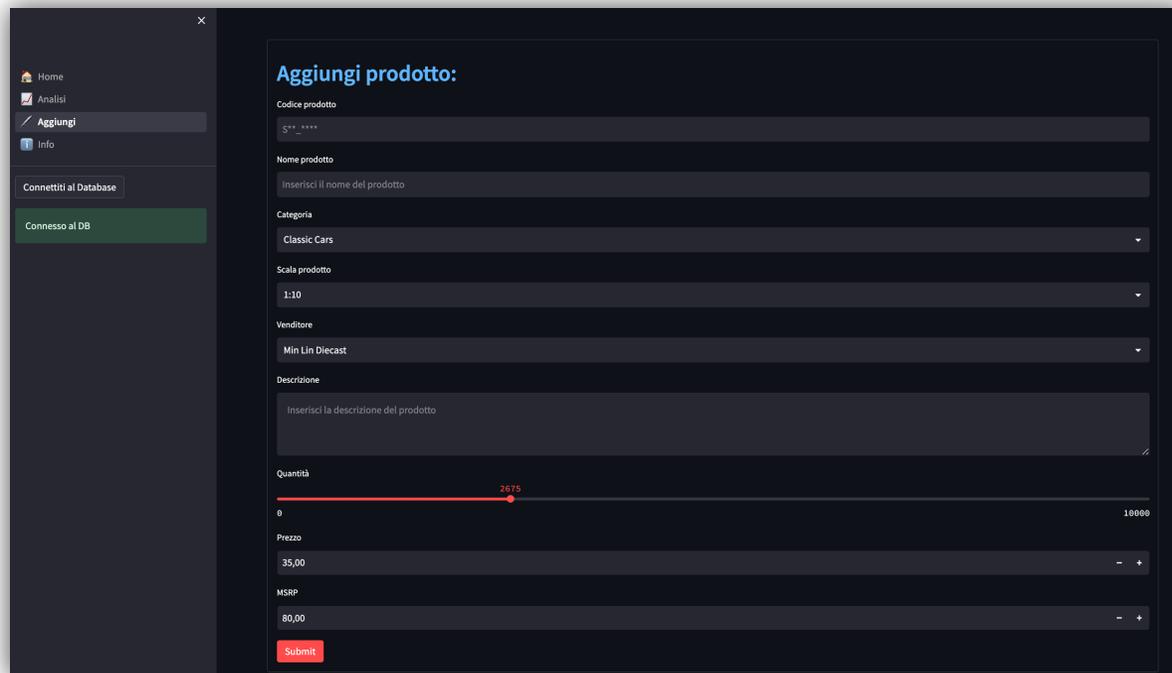
	numeroClienti	country
0	36	USA
1	13	Germany
2	12	France
3	7	Spain
4	5	UK
5	5	Australia
6	4	New Zealand
7	4	Italy
8	3	Norway

Clienti con maggior *credit limit* negli USA

	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

Aggiunta di un prodotto

- Creare un form per l'aggiunta di un nuovo prodotto nel database
- Utilizzare widget di inserimento testo, selectbox, slider e input numerico
- Aggiungere un tasto submit, eseguire la query e verificare che sia stata eseguita con successo
- Acquisire i seguenti parametri: *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 menu with options: Home, Analisi, Aggiungi (selected), and Info. Below the menu are buttons for 'Connettili al Database' and 'Connesso al DB'. The main content area is titled 'Aggiungi prodotto:' and contains the following form fields:

- Codice prodotto:** A text input field with a mask '***-****'.
- Nome prodotto:** A text input field with the placeholder 'Inserisci il nome del prodotto'.
- Categoria:** A dropdown menu with 'Classic Cars' selected.
- Scala prodotto:** A dropdown menu with '1:10' selected.
- Venditore:** A dropdown menu with 'Min Lin Diecast' selected.
- Descrizione:** A large text area with the placeholder 'Inserisci la descrizione del prodotto'.
- Quantità:** A slider control with a red bar and a value of 2675. The range is from 0 to 10000.
- Prezzo:** A numeric input field with the value 35,00 and minus/plus buttons.
- MSRP:** A numeric input field with the value 80,00 and minus/plus buttons.
- Submit:** A red button at the bottom of the form.