Basi di Dati

SQL in Python & Streamlit - Esercitazione n. 8

Passi preliminari per lo svolgimento dell'esercitazione

Lo scopo di questa esercitazione è far interagire un database con un'applicazione web, progettata per eseguire interrogazioni in base alle interazioni dell'utente.

L'applicazione web è scritta in Python, e si serve della libreria Streamlit per visualizzare i dati. L'applicazione dialogherà con un database MySQL, creato e gestito dal software XAMPP.

Installazione del software XAMPP (solo studenti in aula)

- 1) Scaricare l'ultima versione di XAMPP da questo <u>link</u> scegliendo la versione corrispondente al sistema operativo utilizzato.
- 2) Al termine del download, eseguire il file di installazione senza cambiare alcuna opzione.

Installazione di Visual Studio Code ed estensione Python

Studenti in LAIB

Visual Studio Code è già installato sui PC dei LAIB. Dopo l'apertura del programma, verificare che le estensioni Python e Python Debugger siano installate tramite il pannello dedicato alle estensioni.



Studenti in aula

- 1) Scaricare l'ultima versione di Visual Studio Code a <u>questo link</u>.
- 2) Al termine del download, eseguire il file di installazione senza cambiare alcuna opzione.
- 3) Avviato Visual Studio Code, cercare e installare le estensioni "Python" e "Python Debugger" tramite il pannello dedicato alle estensioni.



Nota: In alcuni casi, installare l'estensione "Python" installa anche "Python Debugger" in automatico.

Installazione di Python 3.11.5 (solo studenti in aula)

- 1) Scaricare Python 3.11.5 da uno dei seguenti link, in base al sistema operativo utilizzato.
 - <u>Windows</u>
 <u>macOS</u>
 <u>RedHat</u>
 - <u>Debian/Ubuntu</u>
- 2) Al termine del download, eseguire il file di installazione, senza cambiare alcuna opzione.

Avvio di MySQL e Apache tramite XAMPP (studenti in LAIB + aula)

L'esecuzione degli script SQL contenenti i comandi SQL per la creazione e il popolamento della base di dati avviene tramite l'interfaccia web di MySQL.

Prima di aprire l'interfaccia web di MySQL è necessario:

- Avviare il server locale Apache
- Avviare il server locale MySQL

| 🙁 ХАМРР С | ontrol Panel v | 3.2.1 [Compiled | l: May 7th 2013] | | | | | |
|--|--|--|---|-----------------------|-------|----------|------|-----------|
| 8 | XAMPP Control Panel v3.2.1 | | | | | 🅜 Config | | |
| Modules Service | Module | PID(s) | Port(s) | Actions | | | | 🕜 Netstat |
| | Apache | | | Start | Admin | Config | Logs | Shell |
| | MySQL | | | Start | Admin | Config | Logs | Explorer |
| | FileZilla | | | Start | Admin | Config | Logs | Services |
| | Mercury | | | Start | Admin | Config | Logs | 🚯 Help |
| | Tomcat | | | Start | Admin | Config | Logs | Quit |
| 16:50:56 [16:50:56 [16:50:56 [16:50:56] 16:50:56 [16:50:56] 16:50:56 [16:50:56] | main] main] main] main] main] main] main] main] | XAMPP Instal Checking for r All prerequisit Initializing Mo The FileZilla n The Mercury r The Tomcat n Starting Chec Control Panel | lation Directory: ' orerequisites es found dules nodule is disableo nodule is disableo k-Timer Ready | 'd:\xampp\" I d | | | | E |

In particolare, eseguire i seguenti passi:

- 1) Avviare il programma "XAMPP Control Panel"
- 2) Cliccare su "Explorer" nella parte destra della finestra
 - a. Si aprirà la cartella di configurazione di XAMPP.
 - A questo punto, cliccare sulla cartella "apache" e poi sulla cartella "conf".

- b. Selezionare il file "httpd.conf" e premere Invio.
 Se il file non si apre automaticamente, cliccare su "Blocco note" (o "Notepad)" o su "More apps" e scegliere dall'elenco "Notepad" come programma con cui aprire il file.
- c. Cercare nel file (CTRL+F oppure Edit → Find) la stringa "Listen 80" e sostituirla con "Listen 8080".
- d. Cercare nel file la stringa "ServerName localhost:80" e sostituirla con "ServerName localhost:8080".
- e. Salvare il file (CTRL+S oppure File \rightarrow Save) e chiuderlo.
- 3) Avviare Apache premendo il tasto Start sul pannello di gestione nella riga relativa ad Apache
- 4) Avviare MySQL premendo il tasto Start sul pannello di gestione nella riga relativa a MySQL
- 5) Aprire l'interfaccia web di MySQL: dopo aver aperto un browser (es.: Google Chrome), andare all'indirizzo <u>http://localhost:8080/phpmyadmin/</u>.

| Firefox * | | | | | | |
|---|---|--|---|--|--|--|
| <u>///</u> | | | | | | |
| Ce localhost/phpmyadmin/#PMAU | RL-0:index.php?db=&table=&server=1⌖=&token=e50743e05cdc15f17144cc9e66aaa849 | $\langle c_7 \lor C \rangle$ $\langle c_7 \lor C_7 \rangle$ | ₽ 🚖 📴 | | | |
| phpMuAdmin | - 🗊 Sever 127.0.0.1 | | ⊼ _ | | | |
| <u>≙</u> ⊊ ⊌ ∅ ¢ | 🗊 Database 🖉 SQL 🐁 Stato 🖭 Utenti 🐺 Esporta 🐺 Importa 🥜 Impostazioni | 📱 Replicazione 💿 Variabili 🗮 Set di caratteri 🎲 Motori | | | | |
| Recente Preferiti | Impostazioni Generali | Server del Database | | | | |
| cdcol information_schema information_schema | ≣ Collation della connessione del server @: ut8mb4_general_ci | Server: 127.0.0.1 via TCP/IP Tipo di server: MySQL | | | | |
| mysql performance_schema | Impostazioni di Presentazione | Versione del server: 5.5.20 - MySQL Community Server (GPL) Versione protocollo: 10 Utente: root@localhost | one del server: 5.6.20 - MySQL Community Server (GPL) one protocollo: 10 le: root@localhost | | | |
| phpmyadmin test | Sungua - Language 🥹 Italiano - Italian | Codifica caratteri del server: UTF-8 Unicode (utf8) | | | | |
| €-@ webauth | Tema: pmahomme | Web server | | | | |
| | Ormensione font: 82% Oteveni impodazioni | Apache/2.4.10 (Win32) OpenSSL/1.0.1i PHP/5.5.15 Versione del client del database: biomysql - mysqlpd 5.0.11-de 201205003 - Sbir UBads3b11c9a57et01507592d73b928b8c5c7 Estersione PHP: mysqli @ | w - 7 \$ | | | |

- 6) Per questa esercitazione, sarà necessario importare il database *Hotel*:
 - a. Scaricare lo script di popolamento da <u>questo link</u> ed estrarre i contenuti dell'archivio
 - b. Selezionare il pannello "Importa" dall'interfaccia di phpMyAdmin
 - c. Selezionare il file hotel.sql e poi cliccare su Importa
- 7) Per rilanciare più volte lo script di creazione/popolamento, ricordarsi di cancellare eventuali istanze del database creato in precedenza dal pannello "*Database*" oppure includere all'inizio dello script i comandi per la cancellazione delle tabelle preesistenti

Preparazione e avvio dell'applicazione web

- 1) Scaricare i file dell'applicazione a <u>questo link</u> ed estrarre l'archivio.
- Aprire la cartella appena estratta con Visual Studio Code. Per far ciò, con il programma aperto, andare su File → Open Folder... e selezionare la cartella estratta dall'archivio. Se richiesto, cliccare su "Yes, I trust the authors".
- 3) Creare un ambiente virtuale di Python
 - a. Aprire la palette dei comandi di Visual Studio Code, premendo CTRL+Shift+P o andando su View \rightarrow Command Palette...
 - b. Digitare "Python: Create Environment..." e premere Invio.

| >Python: Create Environment | |
|--|-----------------|
| Python: Create Environment | recently used 🍪 |
| Terminal: Show Environment Contributions | |

c. Selezionare "Venv".

| Şelect an environment type |
|---|
| Venv Creates a `.venv` virtual environment in the current workspace |
| Conda Creates a `.conda` Conda environment in the current workspace |

d. Selezionare "Python 3.11.5 64-bit".



e. Spuntare la casella di fianco a "requirements.txt" e premere Ok.

| | | 1 Selected | ОК |
|---|------------------|------------|----|
| 2 | requirements.txt | | |
| | | | |

- f. Si aprirà una finestra in basso a destra mostrando i progressi. Attendere qualche minuto finché la procedura non sarà completata.
- 4) Avviare l'applicazione web dal menu "Run and debug", cliccando sul pulsante "**Play**". Assicurarsi che accanto al pulsante sia selezionato "**Run Streamlit**".



5) Si aprirà una finestra di terminale che richiede un indirizzo email. Premere invio senza inserire nulla.



6) Si avvierà automaticamente il browser predefinito con l'applicazione aperta.

1



<u>Nota:</u> Una volta avviata l'applicazione, non serve riavviare ogni volta che si effettuano modifiche. I cambiamenti ai file vengono riflessi automaticamente nell'applicazione ogni volta che i file Python vengono salvati.

Descrizione del Database Hotel

La base di dati è stata progettata per gestire le prenotazioni delle stanze di un hotel.

La tabella **STANZA** contiene le informazioni sulle stanze disponibili per le prenotazioni. Le stanze sono identificate da un codice numerico (CodS) e sono caratterizzate dal piano, dalla superficie in metri quadrati e dall'elenco degli eventuali optional disponibili (minibar, jacuzzi, balcone, etc.) presenti nella tabella **HAS_OPTIONAL**. Le stanze possono essere singole, doppie, triple o suite. Tra le diverse stanze disponibili, per le suite è nota la lista di spazi disponibili (camera da letto, sala da pranzo, salotto, etc.) nella tabella **HAS_SPAZI**.

La tabella **AGENZIA** riporta le informazioni relative alle agenzie di viaggio che stipulano le prenotazioni. Le agenzie sono identificate da un codice alfanumerico e caratterizzate dall'indirizzo (via, numero, CAP, città e stato), dal numero di telefono e dal sito web (se disponibile).

Nella tabella **CITTA** sono presenti i dati sulle varie città, tra cui il nome, la regione, la nazione e le coordinate geografiche.

La tabella **PRENOTAZIONE** tiene traccia delle diverse prenotazioni per ogni stanza. Le prenotazioni possono essere stipulate con la stessa agenzia di viaggio o con agenzie di viaggio diverse. Una prenotazione è caratterizzata da una data di inizio e da una data di fine, da un costo. Una stanza può avere una sola prenotazione nello stesso periodo di tempo. Ogni prenotazione è associata ad una sola stanza e ad una sola agenzia.

La struttura del database è riportata nella figura seguente.



Il contenuto di ogni tabella può essere visualizzato dalla pagina di *phpMyAdmin*, espandendo il database "hotel" dal pannello laterale e selezionando la tabella desiderata.

| phpMuAdmin | 🗧 📬 Server 127 0 0 1 -> 👩 Database: hotel = 🌃 fabella: agenzia | | | | | |
|---------------------------|---|---|--|--|--|--|
| 210000 | 🖪 Mostra 📝 Struttura 📄 SQL 🔍 Cerca | 📑 Inserisci 🔜 Esporta 🔜 Importa 🎫 | Privilegi 🎤 Operazioni 💌 Monitoraggio 🚟 Trig | | | |
| ente Preferiti | 🖋 Mostro le righe 0 - 10 (11 del totale, La query ha implegato 0,0002 secondi.) | | | | | |
| Nuovo hotel | SELECT * FROM 'agenzia' | | | | | |
| 🐻 Nuova | Profiling [Modifica inline] [Modifica] [Spiega SQL] [| Crea il codice PHP] [Aggiorna] | | | | |
| 🖌 agenzia 📝 citta | 🗆 Mostra tutti Numero di righe: 25 👻 Fi | iltra righe: Cerca nella tabella Ordina per chi | ave: Nessuno 🗸 | | | |
| has_optional has_spazi | Opzioni extra | | | | | |
| prenotazione | ← T→ ▼ CodA SitoW | Veb Tel Via_Indirizzo CAP_Indirizzo | Citta_Indirizzo Numero_Indirizzo Stato_Indirizzo | | | |
| riormation scheme | 🗌 🗋 🥜 Modifica 👫 Copia 🥥 Elimina 🔰 1 www.e | ag1.it 070-123456 Via Stretta 1234 | Milano 12 Italia | | | |
| nvsal | 🔲 🥜 Modifica 🙀 Copia 🥥 Elimina 🛛 2 www.e | ag2.it 070-234567 Via Corta 2345 | Roma 24 Italia | | | |
| erformance_schema | 🗌 🥔 Modifica 👫 Copia 🥥 Elimina 3 NULL | 070-345678 Via Lunga 3456 | Napoli 38 Italia | | | |
| hpmyadmin | 🔲 🥜 Modifica 🙀 Copia 🥥 Elimina 4 www.r | ag4.it 070-23232 Via Leonardo 2345 | Roma 11 Italia | | | |
| est | 🗌 🥜 Modifica 🙀 Copia 🥥 Elimina 5 www.r | ag5.it 070-89821 Via Rimini 1234 | Milano 5 Italia | | | |
| | 🗌 🥜 Modifica 🙀 Copia 🥥 Elimina 6 www.a | ag6.it 070-77623 Via Cavour 1234 | Milano 89 Italia | | | |
| | Modifica Si Copia GElimina 7 www.c | ag7.it 070-14521 Via dei Maritri 1234 | Milano 43 Italia | | | |

Struttura del progetto Streamlit

I componenti del progetto Streamlit principali per questa esercitazione sono elencati di seguito.

- Il file **Home.py** contiene il codice dell'homepage.
- La cartella **pages** contiene le altre pagine dell'applicazione. Ogni file al suo interno corrisponde ad una diversa pagina.
- Il file **utils/connection.py** contiene le funzioni necessarie per dialogare con il database MySQL. Le funzioni
 - o connect_db(...)

Effettua la connessione al database.

• execute_query(sql)

Esegue una query all'interno del database. Il parametro deve essere una stringa contenente il codice SQL relativo alla query.

• check_connection()

Determina se si è attualmente connessi al database. Restituisce True se si è connessi, False altrimenti.

Esercizi

0 – Connessione al database MySQL

All'interno del file **utils/connection.py**, implementare la funzione **connect_db(...)** in modo da aprire una connessione con il database.

La funzione riceve i seguenti parametri (già forniti):

- dialect: nome del linguaggio che verrà utilizzato per la connessione (e.g., "mysql").
- **username** e **password**: nome e password dell'utente nel database.
- host: nome della macchina che ospita il DBMS (in questo caso, la macchina locale).
- **dbname**: nome del database.

La funzione deve restituire un oggetto di connessione valido.

Per verificare l'implementazione, cliccare il bottone "Connetti al database" nel pannello laterale dell'applicazione. Se è corretta, apparirà un messaggio di avvenuta connessione.



1 – Pagina Agenzie

La pagina, il cui codice si trova in **pages/1_Agenzie.py**, permette di visualizzare una lista di agenzie e relative metriche.

Modificare il codice della pagina in modo da soddisfare le seguenti richieste.

- 1. Mostrare le seguenti metriche:
 - a. Numero totale di agenzie
 - b. Numero di città univoche in cui sono presenti agenzie
 - c. Città con il maggior numero di agenzie

2. Il progetto iniziale mostra già una lista totale di agenzie. Modificare la query SQL in modo da permettere ad un utente di filtrare le agenzie per città.

Opzionale

3. Mostrare su una mappa la posizione geografica delle città in cui sono presenti agenzie. Utilizzare la funzione <u>st.map(df)</u>. L'oggetto df è un DataFrame di Pandas avente due colonne "LAT" e "LON", corrispondenti rispettivamente a latitudine e longitudine (che devono essere obbligatoriamente presenti) della città in cui è presente almeno un'agenzia.

2 – Pagina Stanze

La pagina **pages/2_Stanze.py**, mostra la lista totale delle stanze, insieme ai relativi spazi e optional.

Modificare il codice della pagina in modo da implementare i seguenti filtri:

- Filtro sul **tipo di stanza**: mostrare solo le stanze del tipo selezionato dall'utente. Se l'opzione "Tutte" è selezionata, mostrare le stanze appartenenti a qualsiasi tipo.
- Filtro sugli **optional**: l'utente può selezionare 0 o più optional. La query deve restituire le stanze che contengono tutti gli optional selezionati dall'utente. Se nessun optional viene selezionato, visualizzare le stanze con qualsiasi (o nessun) optional.
- Filtro sulla **cucina**: se l'utente seleziona la casella "Voglio la cucina", filtrare solo le stanze che contengono una cucina tra i suoi spazi. Altrimenti, visualizzare tutte le stanze (con e senza cucina).

Nota:

L'utente può utilizzare più filtri contemporaneamente. In questo caso, restituire le stanze che soddisfano tutti i filtri applicati.

Suggerimento:

Costruire le condizioni di WHERE in maniera incrementale, aggiungendole progressivamente alla query SQL.

3 – Pagina Prenotazioni

La pagina **pages/3_Prenotazioni.py** mostra su un grafico l'andamento dei costi delle stanze che hanno avuto costo giornaliero più alto, per ogni mese dell'anno 2023.

In altri termini, la pagina deve presentare il risultato della seguente query: "Per ciascun mese dell'anno 2023, visualizzare il codice ed il costo giornaliero medio della stanza che ha il costo giornaliero medio più alto".

Nota:

Per permettere la corretta visualizzazione del grafico, il risultato deve contenere obbligatoriamente dei campi chiamati "Mese" e "MediaGiornaliera".

Suggerimenti:

Per estrarre giorno, mese e anno da una data, è possibile utilizzare le funzioni DAY (Data), MONTH (Data) e YEAR (Data) di MySQL.

Per calcolare la differenza tra due date (in termini di giorni trascorsi) è possibile utilizzare la funzione DATEDIFF(Data2, Data1) di MySQL.

Risultato

| | Prenotaz | zioni | |
|---|-------------|-------------------------|------------------------|
| | Mese | STANZA_CodS | MediaGiornaliera |
| | 1 | 2 | 139.3333 |
| | 2 | 3 | 89.4444 |
| 2 | 3 | 13 | 88.8889 |
| | 4 | 10 | 220 |
| 4 | 7 | 10 | 169.2857 |
| | 8 | 10 | 316.6667 |
| | 9 | 7 | 113.6364 |
| | 12 | 8 | 236.6667 |
| 350 300 250 200 150 100 50 0 | | | |
| 0 | 1.0 2.0 3.0 | 4.0 5.0 6.0 7.0 Mese | 8.0 9.0 10.0 11.0 12.0 |

atteso: