

Progetto di Esonero – Laboratorio di Ingegneria Informatica: Interrogazione di Database tramite Linguaggio Naturale

Emanuele Cocchi

Sapienza Università di Roma

cocchi.1967642@studenti.uniroma1.it

1 Obiettivo del Progetto

L'obiettivo del progetto è lo sviluppo di un'applicazione web in grado di permettere agli utenti di interrogare un database relazionale utilizzando il linguaggio naturale. L'applicazione si occupa di tradurre automaticamente interrogazioni in italiano in query SQL, eseguire tali query su un database e restituire i risultati in modo leggibile. Nel contesto dell'esonero, la conversione da linguaggio naturale a SQL non è implementata tramite tecniche di intelligenza artificiale, ma si basa su una lista predeterminata di coppie (interrogazione in italiano, query SQL corrispondente).

2 Organizzazione del Database

Il database **movie_catalog**, all'interno del DBMS MariaDB, è strutturato attorno a tre tabelle principali:

- **directors** (id, name, age): contiene le informazioni che riguardano il singolo regista, identificato univocamente da id (primary key) e anche da name (unique).
- **platforms** (id, name): contiene le informazioni che riguardano la singola piattaforma streaming, identificata univocamente da id (primary key) e anche da name (unique).
- **movies** (id, title, year, genre, id_director, id_platform1, id_platform2): contiene le informazioni che riguardano il singolo film, identificato univocamente da id (primary key) e anche da title (unique).

Sono presenti tre foreign key all'interno della tabella **movies**: `id_director` si riferisce al campo `id` della tabella **directors**, `id_platform1` e `id_platform2` si riferiscono al campo `id` della tabella **platforms** (un film può essere su al più due piattaforme) per garantire l'integrità referenziale.

3 Organizzazione del codice

Il progetto è suddiviso in tre componenti principali: `backend`, `frontend` e `mariadb_init`, ciascuno organizzato in modo modulare e coerente con le rispettive responsabilità.

3.1 Backend

La cartella `backend` è strutturata come segue:

- `backend/src/backend/`:
 - `connection_manager.py`: definisce la classe `ConnectionManager`, responsabile della gestione della connessione al database MariaDB e dell'esecuzione delle query SQL.
 - `models.py`: contiene le classi di modello (derivate da `BaseModel` di Pydantic) utilizzate per definire le strutture dati delle API tra backend e frontend.
 - `backend.py`: punto di ingresso dell'applicazione FastAPI. Gestisce gli endpoint REST, interagisce con il database tramite `ConnectionManager` e comunica con il frontend.

3.2 Frontend

La cartella `frontend` è organizzata come segue:

- `frontend/src/frontend/`:
 - `frontend.py`: gestisce gli endpoint web e comunica con il backend usando la libreria `requests`. Interagisce con le pagine HTML tramite `Jinja2Templates`.
- `frontend/templates/`:

- `index.html`: pagina principale in cui l'utente può inserire una domanda in linguaggio naturale e accedere alle altre funzionalità.
- `schema-summary.html`: visualizza lo schema del database.
- `search.html`: mostra i risultati ottenuti dall'esecuzione della query SQL corrispondente alla domanda.
- `add.html`: consente l'inserimento di nuovi dati nel database.

3.3 Mariadb_init

La cartella `mariadb_init` contiene lo script di inizializzazione del database:

- `init.sql`: definisce la struttura delle tabelle per il database all'interno di MariaDB. Lo script viene eseguito automaticamente all'avvio del container per predisporre il database.