

# Basi di Dati Mod. 2 - Progetto A.A. 2020/2021

Stefano Calzavara

April 16, 2021

## 1 Introduzione

L'obiettivo del progetto è lo sviluppo di una web application che si interfaccia con un database sottostante. Il progetto deve essere sviluppato in Python, utilizzando le librerie Flask e SQLAlchemy. La scelta del DBMS da utilizzare è invece libera e lasciata ai singoli gruppi. Siete invitati a leggere interamente questo documento con attenzione ed a chiarire col docente eventuali punti oscuri prima dello sviluppo del progetto.

## 2 Temi per il Progetto

Il progetto può essere svolto liberamente su uno dei seguenti due temi. La scelta del tema non avrà nessuno impatto sulla valutazione finale del progetto: entrambi i temi possono garantire il punteggio massimo all'esame. I temi sono deliberatamente presentati ad alto livello e si prestano allo sviluppo di progetti più o meno complicati, a seconda dei gusti, della fantasia e dell'abilità dei componenti del gruppo. Considerate questi temi sostanzialmente come semplici spunti da cui partire: potete utilizzare la vostra esperienza con applicazioni simili per identificare un insieme di funzionalità interessanti da implementare.

### 2.1 Creazione di Questionari Online

Vi viene chiesto di curare il design e l'implementazione di una web application per la creazione di questionari a risposta multipla nello stile di Google Forms. L'applicazione deve permettere la creazione di nuove domande, possibilmente di tipo diverso (es. risposta singola o risposta multipla), da combinare all'interno di un unico questionario da rendere accessibile ad una community di utenti. Il creatore di un questionario deve avere accesso ad un'interfaccia di analisi dei dati per estrarre statistiche sulle risposte fornite dalla community.

Vengono forniti alcuni spunti possibili per arricchire il progetto, senza pretesa di esaustività:

- Definire categorie di domande, permettendo l'importazione di gruppi di domande per categoria.
- Gestire l'attivazione di domande sulla base delle risposte fornite alle domande precedenti.
- Implementare una funzionalità di esportazione dei dati per analisi offline, es. via CSV.

### 2.2 Gestione di una Palestra durante il Covid-19

L'emergenza Covid-19 richiede accessi contingentati alle palestre, quindi avete deciso di sviluppare un sistema di controllo degli accessi tramite prenotazioni online. L'applicazione deve presentare un calendario suddiviso in un numero di slot prenotabili e deve essere possibile imporre un limite al numero di accessi per ciascuno slot. Oltre alla classica sala pesi, la palestra offre anche una serie di corsi, ciascuno dei quali è gestito da un istruttore e può avere un numero massimo di iscritti.

Vengono forniti alcuni spunti possibili per arricchire il progetto, senza pretesa di esaustività:

- Le palestre sono molto variegata in termini di numero e dimensioni delle sale pesi, quindi ha senso sviluppare un'architettura generale configurabile rispetto alla specifica palestra.
- Ciascun gestore può definire le proprie politiche interne e declinare i protocolli di sicurezza ministeriali in modo diverso, senza contare che tali protocolli tendono a cambiare con una certa frequenza, quindi cercate di offrire flessibilità in questo senso. Per esempio alcuni gestori potrebbero volere imporre dei limiti sul numero di accessi settimanali o sul tempo complessivo di allenamento giornaliero.
- Le prenotazioni nella stessa fascia oraria potrebbero automaticamente generare un calendario di accessi scaglionati per evitare assembramenti all'ingresso. Inoltre potrebbe essere utile mantenere un log di tutte le prenotazioni effettuate a fini di contact tracing.

### 3 Requisiti del Progetto

Il progetto richiede come minimo lo svolgimento dei seguenti punti:

1. Progettazione concettuale e logica dello schema della base di dati su cui si appoggerà all'applicazione, opportunamente commentata e documentata.
2. Creazione di un database, anche artificiale, tramite l'utilizzo di uno specifico DBMS. La creazione delle tabelle e l'inserimento dei dati può essere effettuato anche con uno script esterno al progetto.
3. Implementazione di un front-end minimale basato su HTML e CSS. E' possibile utilizzare framework CSS esistenti come W3.CSS, Bootstrap o altri. E' inoltre possibile fare uso di JavaScript per migliorare l'esperienza utente, ma non è strettamente necessario e non influirà sulla valutazione finale.
4. Implementazione di un back-end basato su Flask e SQLAlchemy (o Flask-SQLAlchemy).

Per migliorare il progetto e la relativa valutazione è raccomandato gestire anche i seguenti aspetti:

1. Integrità: definizione di vincoli e trigger per garantire l'integrità dei dati gestiti dall'applicazione.
2. Sicurezza: definizione di opportuni ruoli e politiche di autorizzazione, oltre che di ulteriori meccanismi atti a migliorare il livello di sicurezza dell'applicazione (es. difese contro SQL injection).
3. Performance: definizione di indici o viste materializzate sulla base delle query più frequenti previste.
4. Transazioni: utilizzo di transazioni per garantire la correttezza di funzionalità strutturate.
5. Astrazione dal DBMS sottostante: uso di Expression Language o ORM per astrarre dal dialetto SQL.

E' possibile focalizzarsi solo su un sottoinsieme di questi aspetti, ma i progetti eccellenti cercheranno di coprirli tutti ad un qualche livello di dettaglio. E' meglio approfondire adeguatamente solo alcuni di questi aspetti piuttosto che coprirli tutti in modo insoddisfacente.

### 4 Documentazione

Il progetto deve essere corredato da una relazione in formato PDF opportunamente strutturata, che discuta nel dettaglio le principali scelte progettuali ed implementative. Una struttura plausibile per la relazione può essere la seguente:

1. Introduzione: descrizione ad alto livello dell'applicazione e struttura del documento.
2. Funzionalità principali: una descrizione delle principali funzionalità fornite dall'applicazione.
3. Progettazione concettuale e logica della basi di dati, opportunamente spiegate e motivate.

4. Query principali: una descrizione di una selezione delle query più interessanti che sono state implementate all'interno dell'applicazione, utilizzando una sintassi SQL opportuna.
5. Principali scelte progettuali: politiche di integrità e come sono state garantite in pratica, definizione di ruoli e politiche di autorizzazione, uso di indici, ecc. Tutte le principali scelte progettuali devono essere opportunamente commentate e motivate.
6. Ulteriori informazioni: scelte tecnologiche specifiche e qualsiasi altra informazione sia necessaria per apprezzare il progetto.

Il codice del progetto deve essere inoltre opportunamente strutturato e commentato per favorirne la manutenzione e la leggibilità.

## 5 Consegna e Valutazione

Ciascun gruppo deve consegnare il progetto tramite un unico file ZIP caricato tramite Moodle nelle finestre dedicate, tipicamente in prossimità delle sessioni di esame. Il file ZIP deve contenere sia il progetto che la documentazione (in un unico file in formato PDF). Ogni componente del gruppo è responsabile di chiarire il proprio contributo al progetto durante la presentazione orale dello stesso, ma la valutazione del progetto sarà unica per l'intero gruppo a meno di casi eccezionali (es. copiatore). Durante la presentazione orale del progetto uno o più componenti del gruppo mostreranno all'opera l'applicazione sviluppata, facendo vedere le funzionalità principali e rispondendo alle domande del docente. E' sufficiente che l'applicazione funzioni localmente su una delle macchine del gruppo: non è richiesto che sia accessibile tramite Internet.

Il progetto verrà valutato rispetto ai seguenti parametri:

1. Documentazione: qualità e completezza della documentazione allegata.
2. Database: qualità della progettazione ed uso appropriato degli strumenti presentati nel corso.
3. Funzionalità: quantità e qualità delle funzionalità implementate dall'applicazione.
4. Codice: qualità complessiva del codice prodotto (robustezza, leggibilità, riuso...).

Si noti che eventuali progetti artificialmente complicati potrebbero essere penalizzati: implementare funzionalità complesse, ma non appropriatamente pensate o motivate, non è una buona strategia per migliorare la valutazione del proprio progetto.

**Nota a questa edizione.** I migliori gruppi che svilupperanno il primo fra i due temi suggeriti potranno essere contattati per portare in produzione una versione opportunamente rivista ed estesa del loro progetto. In particolare c'è un forte interesse del gruppo di ricerca del prof. Wilmer Pasut nello sviluppo di un'applicazione di questo tipo, di conseguenza potrebbe essere possibile sviluppare ulteriormente il progetto come parte dell'attività di tesi triennale.