

# Corso di Basi di Dati

## CdS Informatica per il Management

### TRACCIA di PROGETTO, A.A. 2023/2024

#### PREMESSA.

Si vuole realizzare la piattaforma **ESQL** per supportare la didattica del corso di dati, e, nello specifico, del modulo di programmazione SQL. La piattaforma è liberamente ispirata al tool Moodle SQL utilizzato nelle esercitazioni in aula e contenuto nel tool Virtuale. La piattaforma consente ai docenti di creare delle tabelle del modello relazionale, e di popolarne il contenuto. Inoltre, consente di creare test con domande a risposta chiusa o con sketch di codice. La piattaforma consente la registrazione da parte di studenti; ogni studente può visionare i test disponibili, completare un test, visionare l'esito o inviare messaggi al docente.

#### SPECIFICA DELLA PIATTAFORMA.

La piattaforma **ESQL** si basa sul database relazionale **ESQLDB**.

Tutti gli utenti della piattaforma dispongono di: **email**, **nome**, **cognome**, **eventuale recapito telefonico**. Gli **utenti** sono divisi in due tipologie: **docenti** e **studenti**. I primi dispongono anche di: **nome del dipartimento** di appartenenza e **nome del corso** di cui sono titolari. I secondi dispongono anche di un campo **anno di immatricolazione** e di un **codice alfanumerico** di lunghezza pari a 16 caratteri. I docenti possono creare delle tabelle SQL (definite **tabelle di esercizio**, nel seguito): ogni tabella di esercizio dispone di **nome**, **data di creazione**, un **campo num\_righe**. Inoltre, ogni tabella di **esercizio** dispone di un insieme di attributi: ogni **attributo** dispone di un **nome**, un **tipo**, e può far parte della chiave primaria della tabella di esercizio. Devono poter essere inseriti dai docenti anche gli vincoli di integrità tra attributi di diverse tabelle di esercizio. In aggiunta, ogni docente può creare dei **test**: ogni test dispone di un **titolo** univoco, una **data di creazione** ed un'eventuale **foto**. Ogni test può includere una serie di **quesiti**: ogni quesito dispone di un **numero progressivo** (univoco, ma solo all'interno di uno specifico test), un **livello di difficoltà** (campo enum con valori: Basso, Medio, Alto), un campo **descrizione**, un campo **#numrisposte** (ridondanza concettuale, vedere specifiche sotto) e fa riferimento ad una o più tabelle di esercizio tra quelle create dal docente. I quesiti possono appartenere esclusivamente a due categorie: **quesiti a risposta chiusa** o **quesiti di codice**. Nel primo caso, il quesito dispone di una serie di **opzioni di risposta**: ogni opzione dispone di una **numerazione** (univoca, ma solo all'interno di uno specifico quesito) ed un **campo testo**. Nel secondo caso, il quesito dispone **di una o più soluzioni** (sketch di codice SQL che implementano correttamente quanto richiesto dalla descrizione del quesito). Ogni test dispone di un campo booleano **VisualizzaRisposte**: se settato a True, le risposte dei quesiti diventano visibili agli studenti, altrimenti restano nascoste. Gli studenti possono svolgere un test, inserendo una o più risposte per ciascun quesito. Si vuole tenere traccia del **completamento del test**, ossia: **data di inserimento della prima risposta** (su scala temporale), **data di inserimento dell'ultima risposta** (su scala temporale), **stato** (campo enum con tre valori: Aperto, InCompleto, Concluso). Nel caso di quesiti a **risposta chiusa**, la risposta consiste nell'opzione scelta tra quelle disponibili. Nel caso di quesiti di codice, la risposta consiste in un campo testo (codice SQL che risolve l'esercizio). E' prevista la possibilità per lo studente di sottoporre più risposte per lo stesso quesito, in istanti diversi di tempo, ma solo se il test non è stato Concluso. Ogni **risposta** dispone di un **campo esito**, che può valere True o False a seconda che la risposta fornita dallo studente coincida con quella inserita dal docente (nel caso di quesiti a risposta chiusa) o che la risposta fornita dallo

studente produca lo stesso output di quella inserita dal docente (nel caso di quesiti di codice). Infine, è prevista la possibilità di inviare messaggi nella piattaforma. Ogni **messaggio** dispone di un **titolo**, un campo **testo**, una **data di inserimento**, e fa riferimento ad uno specifico test. Il messaggio può essere inviato da un docente: in tal caso, il messaggio viene ricevuto da tutti gli studenti. Viceversa, un messaggio può essere inviato da uno studente: in tal caso, il destinatario è uno specifico docente.

**Infine, si vuole tenere traccia di tutti gli eventi che occorrono nella piattaforma, relativamente all'inserimento di nuovi dati (es. nuovi utenti, nuovi test, nuovi quesiti, etc). Tali eventi vanno inseriti, sotto forma di messaggi di testo, all'interno di un log, implementato in un' apposita collezione MongoDB.**

### **Operazioni sui dati<sup>1</sup>:**

#### **Operazioni che riguardano tutti gli utenti:**

- Autenticazione/registrazione sulla piattaforma
- Visualizzazione dei test disponibili
- Visualizzazione dei quesiti presenti all'interno di ciascun test

#### **Operazioni che riguardano SOLO i docenti:**

- Inserimento di una nuova tabella di esercizio, con relativi meta-dati
- Inserimento di una riga per una tabella di esercizio definita dal docente.
- Creazione di nuovo test
- Creazione di un nuovo quesito con le relative risposte
- Abilitare / disabilitare la visualizzazione delle risposte per uno specifico test
- Inserimento di un messaggio

#### **Operazioni che riguardano SOLO gli studenti:**

- Inserimento di una nuova risposta ad un quesito
- Visualizzazione dell'esito della risposta
- Inserimento di un messaggio

#### **Statistiche (visibili da tutti gli utenti):**

- Visualizzare la classifica degli studenti, sulla base del numero di test completati (un test si considera completato se il suo stato è pari a "Concluso"). Nella classifica NON devono apparire i dati sensibili dello studente (nome, cognome, email) ma solo il codice alfanumerico.
- Visualizzare la classifica degli studenti, sulla base del numero di risposte corrette inserite rispetto al numero totale di risposte inserite. Nella classifica NON devono apparire i dati sensibili dello studente (nome, cognome, email) ma solo il codice alfanumerico.
- Visualizzare la classifica dei quesiti, in base al numero di risposte inserite dagli studenti.

### **Popolamento della piattaforma:**

Non richiesta, bastano i dati sufficienti per la demo in sede d'esame.

---

<sup>1</sup> La lista contiene le operazioni di base: può essere estesa/modificata a discrezione dello studente.

### Vincoli sull'implementazione:

- Implementare tutte le operazioni sui dati (ove possibile) attraverso **stored procedure**.
- Implementare le tre statistiche menzionate sopra mediante **viste**.
- Utilizzare un **trigger** per implementare l'operazione cambio di stato un test. Un test diventa InCompleto per uno studente, quando questi inserisce la prima risposta.
- Utilizzare un **trigger** per implementare l'operazione cambio di stato un test. Un test diventa Concluso per uno studente, quando: (i) ha inserito una risposta a tutti i quesiti del test; (ii) tutte le risposte inserite hanno come esito True.
- Utilizzare un **trigger** per implementare l'operazione cambio di stato un test. Un test diventa Concluso per TUTTI gli studenti quando il docente setta il campo VisualizzaRisposte a True per quel test.
- Utilizzare un **trigger** per incrementare il campo num\_righe quando un docente inserisce una riga all'interno di una tabella
- **Tutti i meta-dati delle tabelle di esercizio (nomi tabelle, lista attributi, vincoli di chiave primaria ed integrità referenziale, etc) create dai docenti devono essere modellati a livello di diagramma E-R e memorizzati nel database ESQldb, sfruttando la riflessività del modello relazionale. In aggiunta, le tabelle devono essere fisicamente generate all'interno del database ESQldb, in modo da supportare operazioni di inserimento righe da parte del docente e l'esecuzione delle query inserite dagli studenti.**
- La piattaforma deve supportare l'esecuzione del codice SQL fornito dallo studente ed il confronto con la/le soluzioni salvate fornite dal docente. Deve poter mostrare eventuali errori derivanti da errori di sintassi SQL. Nel caso di quesiti di codice, deve inoltre calcolare l'esito sulla base del confronto tra il risultato della/e soluzioni fornita dal docente ed il risultato ottenuto dall'esecuzione del codice dello studente.
- Nelle traduzioni delle generalizzazioni (e solo in esse), **utilizzare la soluzione che minimizzi il numero di valori NULL.**

### Tabelle dei volumi:

- Valutare se la seguente **ridondanza:**  
**campo #numrisposte relativo ad un quesito**  
debba essere **tenuta o eliminata**, sulla base delle seguenti operazioni:
  - Aggiungere una nuova risposta ad un quesito esistente (10 volte/mese, interattiva)
  - Rimuovere un quesito e tutte le risposte ottenute (2 volte/mese, batch)
  - Visualizzare tutti gli utenti presenti nella piattaforma (1 volta/mese, batch)
  - Contare il numero di risposte per ciascun quesito presente nella piattaforma (2 volte/mese, interattiva)
- **Coefficienti per l'analisi:**  $w_I = 1$ ,  $w_B = 0.5$ ,  $\alpha = 2$
- **Tabella dei volumi:** 10 risposte per quesito, 20 quesiti, 50 utenti

### Bonus:

Il punteggio massimo ottenibile è **30/30** se si implementano correttamente tutte le specifiche menzionate fin qui. E' previsto il seguente bonus:

- **(per la lode, solo se i punti precedenti sono stati sviluppati correttamente)** Utilizzo di librerie CSS per la realizzazione del front-end Web (es. Bootstrap <https://getbootstrap.com>)