

Progetto Basi di Dati 2023-24

“UniGe Social Sport” (12 CFU)

Dominio applicativo – Specifiche

Si richiede di progettare e realizzare una (parte di una) base di dati a supporto di una piattaforma social “UniGE Social Sport” (ispirata a Fubles: <https://it.fubles.com/>) per studenti dell’Università di Genova. In particolare, la piattaforma consente agli studenti di organizzare partite di sport di squadra (calcio, volley, basket, tennis, beach volley, beach soccer, cricket, TAG Rugby, Minigolf, etc.) presso le strutture del CUS Genova, di partecipare ad eventi sportivi, di interagire con altri utenti attraverso news e meccanismi di rating.

La piattaforma gestisce gli eventi sportivi appartenenti a diverse categorie, tra cui ad esempio partite di sport di squadra, di diverse categorie¹. Per ogni categoria, si vogliono memorizzare: il regolamento (campo di testo), il numero di giocatori e una foto esplicativa.

Si vogliono inoltre gestire le informazioni relative agli utenti della piattaforma. Ogni utente dispone di username (univoco) e password, nome, cognome, anno e luogo di nascita, foto, recapito telefonico, numero di matricola, e nome del Corso di Studi cui afferisce. Gli utenti possono appartenere a due tipologie: utenti premium o utenti semplici. Ogni utente della piattaforma ricade in una delle due tipologie sopra indicate.

Un utente premium può organizzare un evento sportivo: ogni evento dispone di un identificativo univoco, una data di svolgimento, uno stato (APERTO/CHIUSO), e appartiene a una delle categorie tra quelle gestite dalla piattaforma. Ogni evento sportivo si svolge all’interno di un impianto del CUS Genova. Gli utenti semplici non possono organizzare eventi sportivi. Per ogni impianto, si vuole tenere traccia di: nome (univoco), via, numero di telefono, indirizzo e-mail, coordinate (latitudine/longitudine).

È prevista la possibilità per gli utenti –sia premium che semplici- di iscriversi agli eventi sportivi in stato APERTO presenti sulla piattaforma, in qualità di giocatore o di arbitro. Ogni iscrizione fa riferimento a un evento sportivo, è effettuata in una certa data, può specificare un ruolo e, una volta esaminata, è associata a uno stato (CONFERMATO/RIFIUTATO), a seconda che l’utente premium organizzatore dell’evento abbia approvato o meno la richiesta di iscrizione.

Il numero totale di utenti giocatori CONFERMATI per un certo evento sportivo non può eccedere il numero di partecipanti previsti dalla categoria cui afferisce l’evento. Il numero totale di utenti arbitri CONFERMATI non può essere maggiore di 1 per un dato evento. Lo stato di un evento sportivo diventa CHIUSO quando si raggiunge un numero di giocatori pari a quello previsto dalla categoria (la presenza dell’arbitro è opzionale); in tal caso, non è più possibile effettuare iscrizioni all’evento. Se non viene raggiunto in tempo utile il minimo numero di giocatori necessario a disputare l’evento, questo viene cancellato e gli utenti iscritti vengono avvisati. È possibile per un partecipante confermato disiscriversi entro un tempo limite indicato dall’organizzatore, dopo tale limite potrà solo indicare un sostituto che parteciperà al suo posto.

Inoltre, è prevista la possibilità di memorizzare l’esito di un evento sportivo. L’esito è immesso dall’utente organizzatore dell’evento stesso e contiene: il numero di goal/punti della prima squadra, il numero di goal/punti della seconda squadra, eventualmente il numero di goal/punti messi a segno da ciascun utente giocatore.

¹ Esempi di categorie sono quelle elencate sopra: calcio, volley, basket, tennis, beach volley, beach soccer, cricket, TAG Rugby, Minigolf, etc.)

La piattaforma consente inoltre agli utenti premium di organizzare tornei per le varie categorie. I tornei hanno una descrizione, eventuali sponsor e premi, eventuali restrizioni per la partecipazione. I tornei consistono in un certo numero di eventi e sono organizzati in accordo a diverse modalità (eliminazione diretta, gironi all'italiana, mista).

La piattaforma supporta la formazione delle squadre. Un utente premium può registrare una squadra, specificando nome, colore maglia, numero partecipanti minimo e massimo, descrizione ed eventuali note. L'utente creatore può inserire direttamente gli utenti componenti o, alternativamente, un utente può candidarsi a partecipare a una squadra in formazione (per un certo ruolo), con il meccanismo descritto sopra per gli eventi. L'utente premium che ha registrato la squadra accetta o rifiuta la candidatura. La formazione della squadra risulta conclusa quando l'utente ne cambia lo stato in definitiva o quando viene raggiunto il limite massimo di candidature accettate. L'utente organizzatore del torneo inserisce tutti gli eventi che lo compongono. Nel caso di tornei la squadra è formata inizialmente e non sono possibili sostituzioni di giocatori per singoli eventi.

Ogni utente –sia semplice che premium- che ha partecipato ad un evento può eventualmente valutare la prestazione degli altri utenti giocatori; la valutazione include la data, un punteggio (da 0 a 10), un eventuale commento.

Ogni utente-giocatore è caratterizzato, relativamente a un certo sport, da un livello. Il livello è un punteggio che, da 0 a 100, indica il valore di un giocatore in relazione alle partite che ha disputato e i voti che ha ricevuto. Il calcolo è legato ad un algoritmo che, partendo da un livello medio pari a 60, lo modifica partita per partita in base a diversi parametri di riferimento (voti ricevuti, esito partita, affidabilità voti, ...).

I giocatori che non si siano presentati ad almeno una partita per cui sono registrati tra le ultime 20 (no show), o che abbiano fatto un ritardo nelle ultime 10 partite, o che abbiano richiesto una sostituzione nelle ultime 10 partite vengono "indicati" come inaffidabili (informazione visibile a tutti gli utenti premium).

Attività da svolgere

Il progetto consiste nelle seguenti attività. Ogni attività dovrà produrre un deliverable (prodotto) di tipo software o di tipo report (documentazione) che illustri il processo di progettazione e le scelte effettuate. Per i prodotti attesi si dovranno completare i template forniti per le varie parti. In particolare, per i deliverable di tipo software i file SQL devono essere file SQL ... cioè formato solo testo e apribili ed eseguibili in pgAdmin.

I. Progettazione logica [*fare riferimento a e completare il template fornito per la parte 1 [vedi file `TemplatePartel.docx` (word) e `TemplatePartel.tex` (latex)].*]

Lo scopo di questa attività è la definizione di uno schema logico relativi a una base di dati relazionale per il dominio sopra descritto, ipotizzando un carico di lavoro a partire dalle specifiche sopra fornite (es. inserimento scuole, orti, repliche, misurazioni, analisi dei dati per attività di biomonitoraggio, ecc). Il progetto dovrà essere completo della specifica di tutti i vincoli di integrità. Si dovrà verificare la qualità dello schema relazionale progettato, applicando i principi della teoria della normalizzazione. In particolare, si dovrà stabilire se lo schema è normalizzato rispetto alla forma normale di Boyce Codd o rispetto alla terza forma normale. Se non è normalizzato rispetto alla terza forma normale, proporre una decomposizione senza perdita e che preserva le dipendenze per almeno una relazione non normalizzata e indicare quale forma normale soddisfa.

II. Realizzazione [*completare `TemplatePartell.sql` (variante per i CFU per cui seguite il corso)]*]

Lo scopo di questa attività è la definizione in PostgreSQL dello schema logico, il suo popolamento e la realizzazione di alcune operazioni di interrogazione sulla base di dati, nonché di alcune routine e trigger.

1. La definizione dello schema logico della base di dati progettato dovrà contenere la specifica di tutti i vincoli individuati in fase di progettazione. Per ogni vincolo andrà individuata e indicata l'opportuna modalità di implementazione (vincoli di tipo CHECK o trigger). È richiesta l'implementazione dei soli vincoli di tipo CHECK. Per quanto riguarda il popolamento, la base di dati creata dovrà essere popolata con dati sufficienti almeno a verificare che i vincoli di dominio espressi siano verificati e che in generale le operazioni di cui si richiede l'implementazione funzionino correttamente.

Su tale basi di dati si richiede di realizzare:

2. La definizione di una vista *Programma* che per ogni impianto e mese riassume tornei e eventi che si svolgono in tale impianto, evidenziando in particolare per ogni categoria il numero di tornei, il numero di eventi, il numero di partecipanti coinvolti e di quanti diversi corsi di studio, la durata totale (in termini di minuti) di utilizzo e la percentuale di utilizzo rispetto alla disponibilità complessiva (minuti totali nel mese in cui l'impianto è utilizzabile).
3. Le seguenti interrogazioni
 - a. Determinare gli utenti che si sono candidati come giocatori e non sono mai stati accettati e quelli che sono stati accettati tutte le volte che si sono candidati;
 - b. determinare gli utenti che hanno partecipato ad almeno un evento di ogni categoria;
 - c. determinare per ogni categoria il corso di laurea più attivo in tale categoria, cioè quello i cui studenti hanno partecipato al maggior numero di eventi (singoli o all'interno di tornei) di tale categoria.
4. Le seguenti procedure/funzioni:
 - a. funzione che effettua la conferma di un giocatore quale componente di una squadra, realizzando gli opportuni controlli;
 - b. funzione che dato un giocatore ne calcoli il livello (se si è deciso di memorizzare il livello, definire il trigger che mantenga aggiornato il valore dell'attributo livello come trigger E.b. e definire invece la funzione corrispondente alla seguente query parametrica: data una categoria e un corso di studi, determinare la frazione

di partecipanti a eventi di quella categoria di genere femminile sul totale dei partecipanti provenienti da quel corso di studi.)

5. I seguenti trigger:

- a. verifica del vincolo che non è possibile iscriversi a eventi chiusi e che lo stato di un evento sportivo diventa CHIUSO quando si raggiunge un numero di giocatori pari a quello previsto dalla categoria
- b. mantenimento dell'attributo derivato LIVELLO, se si è deciso di memorizzarlo, altrimenti del trigger che gestisce la sede di un evento: se la sede è disponibile nel periodo di svolgimento dell'evento la sede viene confermata altrimenti viene individuata una sede alternativa: tra gli impianti disponibili nel periodo di svolgimento dell'evento si seleziona quello meno utilizzato nel mese in corso (vedi vista *Programma*).

III. Progettazione fisica, elaborazione delle interrogazioni e controllo dell'accesso

Lo scopo di questa attività è: (i) la progettazione fisica della base di dati, la sua creazione in PostgreSQL, un popolamento di dimensioni significative, la verifica dei piani di esecuzione scelti dal sistema per alcune interrogazioni prima e dopo la creazione dello schema fisico; (ii) la definizione di un modello di controllo dell'accesso per il dominio considerato.

1. Per la parte di progettazione fisica, si richiede di (per ogni punto, P indica una attività di progettazione, SQL indica un'attività di traduzione in SQL):
 - a. Individuare un carico di lavoro composto da tre interrogazioni a vostra scelta, specificandole in linguaggio naturale; la prima interrogazione deve contenere una singola condizione di selezione e nessun join, la seconda deve contenere una condizione di selezione complessa e nessun join, la terza deve contenere almeno un join e almeno una condizione di selezione.
 - b. Per ogni relazione R coinvolta nel carico di lavoro, creare una nuova relazione R_CL (relazione per analisi Carico di Lavoro) in SQL con schema equivalente a R ma senza vincoli di chiave primaria, secondaria o esterna.
 - c. Tradurre le interrogazioni del carico di lavoro, proposte al punto (a), in SQL, considerando le relazioni definite al punto (b).
 - d. Progettare lo schema fisico della base di dati a partire dal carico di lavoro individuato; lo schema fisico dovrà contenere l'elenco degli indici che si intendono creare per le interrogazioni contenute nel carico di lavoro, specificando relazione di riferimento, chiave di ricerca, il loro tipo (ordinato/hash, clusterizzato/non clusterizzato), motivando opportunamente, in modo sintetico, le scelte effettuate.
 - e. Tradurre lo schema fisico progettato in SQL per le tabelle appena create.
 - f. Inserire un numero di tuple significativo che portino ad occupare, per ciascuna relazione appena creata, qualche decina di pagine (in PostgreSQL ogni pagina, per default, ha dimensione 8 kb). Per far ciò, e *solo per le relazioni coinvolte nel carico di lavoro*, si suggerisce di utilizzare un generatore automatico di dati (ad esempio www.datanamic.com). Per raggiungere più velocemente l'occupazione di spazio disco indicata, si suggerisce di aggiungere solo a queste relazioni un attributo semanticamente inutile (dummy), definito con tipo char(n) o text. Questo permetterà di estendere la dimensione di ciascuna tupla e di arrivare all'occupazione di spazio disco indicata con un numero inferiore di tuple.
 - g. Individuare ed esaminare il piano di esecuzione scelto dal sistema e il tempo di esecuzione per le interrogazioni nel carico di lavoro *prima e dopo la creazione dello schema fisico*; confrontare i piani e i tempi ottenuti nei due casi, fornendo una giustificazione per i risultati ottenuti.
2. Per la parte di controllo dell'accesso, si richiede di definire i ruoli *amministratore di UniGe social sport*, *utente premium*, *gestore impianto*, *utente semplice*, individuando una opportuna gerarchia tra di essi. Individuare quindi i privilegi che ritenete ragionevole attribuire ai ruoli, tenendo in considerazione quanto specificato nella descrizione del dominio e la gerarchia considerata, motivando le scelte effettuate. Riassumere la politica di controllo dell'accesso con una tabella contenente una riga per ogni relazione e una colonna per ogni ruolo. Assegnare i ruoli definiti a 4 utenti a vostra scelta. Specificare in SQL la politica di controllo dell'accesso individuata.

Modalità di svolgimento e consegna

Svolgimento

Il progetto deve essere svolto **a gruppi**. La composizione dei gruppi (tre persone) va dichiarata su AulaWeb (sezione Progetto) al momento dell'inizio del lavoro e **al più tardi una settimana prima dell'appello in cui si consegna il progetto**.

Scadenze per la consegna

Sono previste consegne **in corrispondenza di ogni appello di esame**, con consegna da effettuarsi **via AulaWeb entro le ore 9.00 della data dell'esame scritto**.

Modalità di consegna su AulaWeb

Si deve consegnare, utilizzando il compito Consegna progetto su AulaWeb, un singolo file **zip** con nome = GruppoXX con XX numero del gruppo (sarà impostata la consegna di gruppo) contenente la documentazione e il codice richiesti ai punti precedenti, opportunamente organizzata in file.

Il file zip dovrà contenere

- un documento pdf per la parte I, redatto a partire dai template forniti (TemplatePartel.docx e TemplatePartel.tex), e chiamato Partel.pdf.
- un file SQL per la parte II, redatto a partire dal template fornito (TemplatePartell.sql), chiamato Partell.sql.
- un documento pdf per la parte III, redatto a partire dai template forniti (TemplatePartelll.docx e TemplatePartelll.tex), chiamato Partelll.pdf;
- un primo file SQL per la parte III, redatto a partire dal template fornito (TemplatePartelll-a.sql), chiamato Partelll-a.sql;
- un secondo file SQL contenente i soli comandi di inserimento dati "in the large" delle relazioni coinvolte nel carico di lavoro, chiamato Partelll-b.sql.

Il progetto consegnato deve essere completo (rispetto alle specifiche) e funzionante!

Attenzione: sono permesse consegne ripetute, ma farà fede solo l'ultima e, se sufficiente, verrà valutato un solo progetto (quindi, se avete ambizioni di voto, consegnate solo se siete convinti di aver svolto il progetto in maniera per voi soddisfacente, perché non potrete rifarlo per alzarvi il voto). In caso di progetto con valutazione insufficiente, verrà richiesto di consegnare un nuovo progetto (una sola volta per ogni anno accademico), che dovrà tenere conto dei commenti/correzioni ricevuti sul progetto insufficiente.

Criteri di valutazione

- Correttezza e completezza dei deliverable (per esempio, lo schema concettuale, lo schema ristrutturato e lo schema logico devono essere singolarmente corretti).
- Consistenza complessiva dei deliverable e corretta applicazione delle metodologie (per esempio, lo schema concettuale, lo schema ristrutturato e lo schema logico devono essere tra di loro coerenti).
- Qualità della documentazione in termini di chiarezza e coerenza con le richieste.