

# Basi di Dati 2020-21 – Progetto “Online Challenge Activity”

## Dominio applicativo – Specifiche

Si richiede di progettare e realizzare una (parte di una) base di dati a supporto di un’applicazione per la gestione di sfide a squadre online (ad esempio per approccio gamification in ambito didattico – game based learning).

Ogni gioco ha un identificatore, è proposto per un numero massimo di squadre e consiste di una plancia di gioco con un’immagine di sfondo, un insieme di icone (immagini della stessa dimensione con un nome associato, che vengono utilizzate come segnaposto della posizione della squadra sulla plancia di gioco) a tema e un certo numero di caselle. Le caselle hanno una posizione sull’immagine (coordinate X e Y), una tipologia, un numero d’ordine nel percorso (sequenza progressiva). La casella start (punto di partenza) è quella con numero 0 e la casella arrivo è quella con numero massimo. Ogni casella appartiene a un unico gioco mentre lo stesso set di icone (identificato da un nome) può essere utilizzato in più giochi.

In aggiunta, nello sfondo possono inoltre essere indicate tre “caselle” (cioè aree con una certa posizione X e Y) *podio* che non vengono percorse dai giocatori durante il gioco ma servono solo per visualizzare le icone delle tre squadre ai primi tre posti in classifica.

A ogni casella (non podio) possono essere associati:

- Un breve video
- Uno o più (massimo 5) quiz a risposta multipla
- Un task

Un quiz a risposta multipla ha un testo (html), può avere un’immagine associata e ha diverse possibili risposte (da 2 a 9). Ogni risposta ha un testo (html), può avere un’immagine associata e ha un punteggio (positivo se giusta, nullo o negativo se sbagliata). Ogni quiz deve prevedere almeno una risposta con punteggio positivo.

I task hanno un testo (html) e un punteggio. La risposta al task avviene caricando un file. La risposta al task deve essere validata manualmente da un admin che decide se assegnare o meno il punteggio associato al task.

Sia quiz che task hanno all’interno dei giochi in cui vengono inseriti un tempo massimo di risposta, che può variare da gioco a gioco. Se la squadra non fornisce risposta al quiz e al task entro il tempo massimo queste scadono e il punteggio ottenuto è nullo.

Alcune caselle modificano la posizione della squadra che vi cade sopra, in particolare ad es. potremmo avere caselle “serpente” che fanno tornare a una casella precedente nella sequenza, caselle “scala” che fanno avanzare a una casella successiva, come illustrato in figura. Per queste caselle viene memorizzato anche il numero della casella destinazione.

Nel gioco può essere presente un elemento aleatorio, in particolare si possono associare al gioco un certo numero di dadi (anche zero) ognuno dei quali ha un ~~punteggio~~ **valore** minimo e un ~~punteggio~~ **valore** massimo (non superiore a sei).



Il comportamento della casella può includere il lancio dei dadi, il punteggio che si ottiene rispondendo alle domande/al task posti sulla casella può modificare il numero di dadi a disposizione della squadra.

Nel caso la casella preveda una casella destinazione, allora non prevedrà il lancio dei dadi [ma può essere comunque modificato il numero di dadi a disposizione della squadra per le caselle successive]. Ogni squadra inizia infatti il gioco con un certo numero di dadi a disposizione e questo può essere modificato nel corso della partita.

Ogni turno di gioco include quindi

- Visualizzazione dell'eventuale video associato alla casella
- Risposta a eventuali domande o al task associati alla casella (non possono essere presenti entrambi)
- Eventualmente a seconda del punteggio conseguito: modifica del numero di dadi a disposizione
- Se è specificata casella successiva: raggiungimento della casella successiva
- Se non è specificata casella successiva: lancio dei dadi e avanzamento di un numero di caselle pari al ~~punteggio~~ *valore* ottenuto con i dadi.

Ogni sfida viene proposta (online) in una certa data e orario e corrisponde all'esecuzione di un gioco. Ogni sfida è basata su un gioco e ha una durata massima. A ogni sfida partecipano un certo numero di squadre. Ogni squadra ha un nome e un'icona con cui viene visualizzata sul tabellone di gioco. Nomi e icone delle squadre che partecipano alla stessa sfida sono tutte distinte.

Alle sfide partecipano squadre costituite da un certo numero (minimo 1) di utenti. Ogni utente ha un indirizzo di e-mail, un nickname, opzionalmente nome, cognome, data di nascita e può appartenere a una o più squadre, ma appartiene a un'unica squadra tra quelle che partecipano alla stessa sfida. Alcune sfide possono essere moderate, in tal caso ogni squadra avrà un utente designato come coach (nel qual caso non dà risposte ma modera solo) o caposquadra (nel qual caso è sia giocatore che moderatore della squadra).

Nel caso di sfida non moderata le risposte della squadra al quiz sono ottenute prendendo la risposta più votata dagli utenti della squadra. Per i task si considera la prima risposta sottomessa. Nel caso di sfida moderata è il moderatore che conferma la soluzione proposta da un giocatore per il task.

Per ogni sfida, si terrà conto delle caselle visitate, dei lanci di dadi e relativi ~~punteggi~~ *valori ottenuti*, delle risposte date e delle soluzioni dei task consegnati dai singoli giocatori e dai team, approvate dai moderatori e dagli admin, con per ogni operazione i relativi tempi, in modo da poter determinare punteggi e classifiche non solo finali ma in ogni momento di svolgimento della sfida.

## Attività da svolgere

Il progetto consiste nelle seguenti attività (per ognuna di esse il materiale da consegnare è dettagliato nella sezione seguente). **Le parti indicate in blu devono essere svolte solo per chi ha in piano Basi di Dati 12 CFU.** Le parti indicate in verde devono essere svolte solo per chi ha in piano Basi di Dati 9 CFU o 12 CFU.

I. **Progettazione:** Lo scopo di questa attività è la definizione di uno schema logico **e di uno schema fisico** relativi a una base di dati relazionale per il dominio sopra descritto e il carico di lavoro sotto descritto. Si ricordi che una base di dati a supporto di un'applicazione deve fornire accesso incapsulato ai dati da gestire in maniera che l'applicazione (ed eventuali altre che vengano in futuro ad insistere sulla stessa base di dati) debbano implementare solo la logica specifica dell'applicazione, non siano forzate a fare controlli o operazioni di manutenzione dei dati che possano essere demandati al DBMS. Il progetto dovrà quindi essere completo della specifica di tutti i vincoli di integrità.

**Il carico di lavoro contiene le tre seguenti interrogazioni, considerate frequenti, definite sulle relazioni corrispondenti alle entità sfida e gioco (nel caso nel vostro schema queste entità siano radici di una gerarchia di specializzazione, ristrutturare la gerarchia mantenendo le entità padre):**

- **Determinare l'identificatore dei giochi che coinvolgono al più quattro squadre e richiedono l'uso di due dadi.**
- **Determinare l'identificatore delle sfide relative a un gioco A di vostra scelta (specificare direttamente l'identificatore nella richiesta) che, in alternativa:**
  - **hanno avuto luogo a gennaio 2021 e durata massima superiore a 2 ore, o**
  - **hanno avuto luogo a marzo 2021 e durata massima pari a 30 minuti.**
- **Determinare le sfide, di durata massima superiore a 2 ore, dei giochi che richiedono almeno due dadi. Restituire sia l'identificatore della sfida sia l'identificatore del gioco.**

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. Se non è normalizzato, proporre una decomposizione senza perdita e che preserva le dipendenze per almeno una relazione non normalizzata.

**Lo schema fisico dovrà essere progettato a partire dal carico di lavoro proposto, motivando opportunamente, in modo sintetico, le scelte effettuate.**

II. **Realizzazione:** Lo scopo di questa attività è la definizione in PostgreSQL dello schema logico **e dello schema fisico della base di dati**, il suo popolamento (vedi punto 4.), **il suo completamento con routine e trigger (vedi punti 6. e 7., solo per  $\geq 9$  CFU)**, la realizzazione del carico di lavoro e di alcune ulteriori operazioni (ritenute poco frequenti) di interrogazione sulla base di dati (vedi punto 5), **la verifica dei piani di esecuzione scelti dal sistema per le interrogazioni del carico di lavoro prima e dopo la creazione dello schema fisico.**

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 l'opportuna modalità di implementazione (vincoli di tipo CHECK o trigger). I vincoli di tipo CHECK andranno implementati mentre non è richiesta l'implementazione degli altri.

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.

Per le relazioni coinvolte nel carico di lavoro, si richiede di inserire un numero di tuple significativo che portino ad occupare, per ciascuna relazione, qualche decina di pagina (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 automatica di dati (ad esempio [www.datanamic.com](http://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.

Per le operazioni di manipolazione si richiede di realizzare le operazioni incluse nel carico di lavoro definito al punto (I) e di prevedere, in aggiunta:

1. La definizione di una vista che fornisca alcune informazioni riassuntive per ogni gioco: il numero di sfide relative a quel gioco disputate, la durata media di tali sfide, il numero di squadre e di giocatori partecipanti a tali sfide, i punteggi minimo, medio e massimo ottenuti dalle squadre partecipanti a tali sfide;
2. Le seguenti interrogazioni
  - a. Determinare i giochi che contengono caselle a cui sono associati task;
  - b. Determinare i giochi che non contengono caselle a cui sono associati task;
  - c. Determinare le sfide che hanno durata superiore alla durata media delle sfide relative allo stesso gioco.

Per ogni operazione (inclusa o meno nel carico di lavoro), si richiede inoltre di applicare i principi del tuning delle interrogazioni e proporre, se necessario, eventuali riscritture, motivando le scelte effettuate.

Si richiede, inoltre, di specificare:

3. Le seguenti procedure/funzioni:
  - a. Funzione che realizza l'interrogazione 2b/l'interrogazione 2c in maniera parametrica rispetto all'ID del gioco (*cioè determina le sfide che hanno durata superiore alla durata medie delle sfide di un dato gioco, prendendo come parametro l'ID del gioco*);
  - b. Funzione di scelta dell'icona da parte di una squadra in una sfida: possono essere scelte solo le icone corrispondenti al gioco cui si riferisce la sfida che non siano già state scelte da altre squadre.
4. I seguenti trigger:
  - a. Verifica del vincolo che nessun utente possa partecipare a sfide contemporanee;
  - b. Mantenimento del punteggio corrente di ciascuna squadra in ogni sfida e inserimento delle icone opportune nella casella podio.

### III: Controllo dell'accesso ed elaborazione delle interrogazioni

Si richiede di individuare il piano di esecuzione scelto dal sistema e il tempo di esecuzione per le interrogazioni contenute nel carico di lavoro **prima e dopo la creazione dello schema fisico**; confrontare i tempi ottenuti nei due casi e fornire una giustificazione per i risultati ottenuti.

Si richiede inoltre di definire i ruoli utente, `giocatore`, `gameadmin`, `gamecreator` individuando una gerarchia tra di essi.

Assegnare ai ruoli sopra descritti i privilegi che ritenete ragionevoli nel dominio applicativo considerato, tenendo in considerazione quanto specificato nella descrizione del dominio e il fatto che un game admin è colui che attiva

le sfide e approva i task e un game creator è colui che crea i giochi, motivando le scelte effettuate.

## Deliverable (prodotti da consegnare)

Ogni attività al punto precedente dovrà produrre un deliverable (prodotto) di tipo software o di tipo report (documentazione) che illustri il processo di progettazione e le scelte effettuate. In particolare, i prodotti attesi per ogni attività sono i seguenti (S=codice SQL, D=documentazione in pdf):

### PARTE I

1. [D] **Requisiti ristrutturati** in modo da eliminare ambiguità.
2. [D] **Progetto concettuale**, articolato in
  - (a) schema ER;
  - (b) documentazione relativa ai domini degli attributi (dizionario dati ed entità);
  - (c) vincoli non esprimibili nel diagramma;
  - (d) specifica dei tipi di gerarchie di generalizzazione.

Per i gruppi va effettuata la consegna completa, includendo anche i punti 1. e 2. che devono riferirsi alle specifiche di dominio qui fornite, e quindi non devono essere quelli consegnati il 20 maggio. Devono, invece, essere consistenti con i documenti di progettazione relativi ai passi successivi.

3. [D] **Progetto logico**, articolato in
  - (a) schema ER ristrutturato;
  - (b) eventuali modifiche dei domini degli attributi e informazioni sui domini di eventuali attributi introdotti;
  - (c) modifiche all'elenco di vincoli del modello concettuale (nuovi vincoli, eventuali vincoli eliminati o modificati);
  - (d) documentazione relativa alle scelte fatte per eliminare le gerarchie di generalizzazione;
  - (e) schema logico;
  - (f) verifica di qualità dello schema e eventuali ottimizzazioni applicate tenendo in considerazione il carico di lavoro
4. [D] **Progetto fisico**, contenente l'elenco degli indici che si intendono creare per le interrogazioni contenute nel carico di lavoro (specificando relazione di riferimento e chiave di ricerca), il loro tipo (ordinato/hash, clusterizzato/non clusterizzato) e la motivazione che ha portato alla loro creazione.

### PARTE II

5. [S] Script SQL per la **creazione dello schema** logico della base di dati in accordo allo schema relazionale ottenuto alla fine della fase di progettazione logica, per la porzione necessaria per i punti successivi (cioè le tabelle coinvolte dalle interrogazioni nel carico di lavoro, nella definizione della vista, nelle interrogazioni, in funzioni, procedure e trigger). Lo schema dovrà essere comprensivo dei vincoli esprimibili con check, e per il **popolamento** di tale base di dati. Per ogni tabella della base di dati, riportare, come commento nello script, il numero di tuple in essa contenute e il numero di blocchi occupati su disco.
6. [D] diagramma che visualizza lo script SQL in forma grafica ottenuto con DataGrip (vedi <https://2020.aula-web.unige.it/mod/resource/view.php?id=83558> per come crearla da 5.).
7. [S] Script SQL per la **creazione dello schema fisico** della base di dati.
8. [S] Il codice SQL sviluppato per implementare le **interrogazioni del carico di lavoro**, le **ulteriori interrogazioni** richieste e la **vista** richieste, inserendo, come commento nello script, la corrispondente richiesta in linguaggio naturale e la **descrizione delle eventuali riscritture applicate in base ai principi del tuning delle interrogazioni**.
9. [S] Il codice PL/pgSQL sviluppato per implementare le **funzioni e procedure** richieste, inserendo, come commento nello script, la specifica relativa a ciascuna funzione o procedura.

10. [S] Il codice PL/pgSQL sviluppato per implementare i **trigger** richiesti, inserendo, come commento nello script, la specifica relativa a ciascun trigger.

### PARTE III

11. [D] Descrizione dei **piani di esecuzione** scelti dal sistema (prima e dopo la creazione dello schema fisico) per le interrogazioni contenute nel carico di lavoro e giustificazione di tale scelta sulla base dello schema fisico realizzato e della dimensione della base di dati.
12. [D] Descrizione della politica di **controllo dell'accesso** scelta, motivando le scelte effettuate (utilizzare una tabella per chiarire quali privilegi si intendono assegnare ai vari ruoli).
13. [S] Script SQL per l'implementazione della politica di **controllo dell'accesso**.

## Modalità di svolgimento e consegna

### *Svolgimento*

Il progetto deve essere svolto **singolarmente**, con l'eccezione dei **Gruppi patto d'aula 2021**: il progetto **può essere svolto in gruppi** dagli studenti che abbiano completato tutte le attività legate al patto d'aula (singole e a gruppi) durante il semestre dell'anno accademico in corso, numero e composizione dei gruppi sono quelli presenti nel modulo AulaWeb del corso. **In caso di progetto svolto in gruppo si chiede di utilizzare GitHub <https://classroom.github.com/g/3vtf1WDL> durante lo svolgimento del progetto per condividere codice e materiale.**

### *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= NNCC con NN nome dello studente e CC cognome dello studente per i progetti singoli e
- con nome = GruppoXX con XX numero del gruppo per i progetti in gruppo (è sufficiente che uno dei componenti del consegna il file);
- contenente la documentazione e il codice richiesti ai punti precedenti, opportunamente organizzata in file. Il file zip dovrà contenere un unico file SQL e un unico documento pdf per ogni parte (quindi in totale, due/**tre** file SQL e due/**tre** file pdf). Ogni file deve essere organizzato in sezioni, il titolo di ciascuna sezione dovrà corrispondere al deliverable corrispondente (il titolo dovrà essere inserito come commento nei file SQL). I file SQL devono essere file SQL ... cioè formato solo testo e apribili ed eseguibili in pgAdmin! Per ogni interrogazione, operazione, routine e trigger riportarne come commento nel file SQL la specifica.

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.