

A large, stylized brain shape is formed by a dense collection of colorful icons. These icons represent a wide range of subjects and technologies, including science (satellite, microscope, globe), technology (laptop, smartphone, keyboard, Wi-Fi symbol), sports (soccer ball, car, airplane), arts (musical notes, camera, film strip), and general concepts (heart, gear, lightbulb, house). The brain is connected to a white game controller with a black cord, suggesting a link between gaming and education.

Studente: Caputo Luca
 Matricola: 4680479
 Corso: Informatica
 CFU: 12
 Parte 1

Indice

1. Requisiti ristrutturati	4
2. Progetto concettuale	
1. schema ER	7
2. dizionario dati ed entità	7
3. specifica gerarchie di generalizzazione	10
4. vincoli non esprimibili nel diagramma	10
3. Progetto logico	
1. schema ER ristrutturato	12
2. modifiche a domini degli attributi e attributi introdotti	12
3. eliminazione gerarchie di generalizzazione	13
4. schema logico	13
5. verifica di qualità e ottimizzazioni	14
6. modifiche all'elenco dei vincoli	15
4. Progetto fisico	
1. analisi del carico di lavoro	16
2. elenco degli indici scelti	17

1. Requisiti ristrutturati

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
- un numero massimo di squadre
- un'immagine di sfondo
- un certo numero di caselle.

Esso ha inoltre 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), il quale un **set** di icone (identificato da un nome) può essere utilizzato in più giochi.

Una **casella** può essere di gioco oppure podio ed è identificata dall'id del gioco e la posizione sull'immagine (coordinate X e Y). Ogni casella appartiene a un unico gioco.

Le **caselle di gioco** hanno

- 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.

Alcune caselle di gioco modificano la posizione della squadra che vi cade sopra, in particolare abbiamo

- caselle **serpente**, che fanno tornare a una casella precedente nella sequenza
- caselle **“scala”** che fanno avanzare a una casella successiva

Per queste caselle viene memorizzato anche il numero della casella di destinazione.

Le **caselle podio** invece non vengono percorse dai giocatori durante il gioco ma servono solo per visualizzare le icone delle tre squadre ai primi tre posti in classifica, e hanno un numero da 1 a 3 che indica il classificato

A ogni casella di gioco possono essere associati:

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

In una casella non possono essere presenti sia quiz che task

Un **quiz** a risposta multipla ha

- un identificatore
- un testo (html)
- può avere un'immagine associata
- diverse possibili risposte (da 2 a 9)

Ogni **risposta** a un quiz ha

- un numero di risposta
- un testo (html)
- può avere un'immagine associata
- un punteggio (positivo se giusta, nullo o negativo se sbagliata)

Ogni quiz deve prevedere almeno una risposta con punteggio positivo.

I **task** hanno

- un identificatore
- un testo (html)
- 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.

Al gioco si possono associare un certo numero di **dadi** (anche zero) ognuno dei quali ha un

- valore minimo (non inferiore a zero)
- 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 prevederà il lancio dei dadi. Ogni squadra inizia infatti il gioco con un certo numero di dadi a disposizione e questo può essere modificato nel corso della partita, purché non superi un limite massimo consentito.

Tali informazioni sono presenti se e solo se il numero dei dati richiesti in un gioco è diverso da 0.

Ogni **sfida** viene proposta (online)

- a una certa data e orario
- ha un identificatore univoco
- corrisponde all'esecuzione di un gioco
- ha una durata massima
- partecipano un certo numero di squadre

Ogni **squadra** ha

- un nome
- un'icona con cui viene visualizzata sul tabellone di gioco
- un certo numero (minimo 1) di utenti

Nomi e icone delle squadre che partecipano alla stessa sfida sono tutte distinte.

Ogni **utente** ha

- un indirizzo di e-mail
- un nickname
- nome (opzionale)
- cognome (opzionale)
- data di nascita (opzionale)

Un utente 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)
- **caposquadra** (nel qual caso è sia giocatore che moderatore della squadra)

Ovviamente un utente non può ricoprire entrambi i ruoli.

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.

Ogni **turno di gioco**, è identificato da

- la sfida effettuata relativa ad un determinato gioco
- la casella visitata
- la squadra che lo effettua

E include quindi

- Visualizzazione dell'eventuale video associato alla casella
- Risposta a eventuali domande o al task associati alla casella, con per ognuna di essi i relativi tempi
- 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 valore ottenuto con i dadi.

Per ogni sfida, si terrà conto di tutto ciò che specificato sopra, delle risposte date e delle soluzioni dei task consegnati dai singoli giocatori e dai team, approvate dai moderatori e dagli admin, per cui per ogni turno bisognerà specificare anche la **risposta data da ogni utente** che verrà confermata alla conclusione di un turno, in modo da poter determinare **punteggi** e classifiche non solo finali ma in ogni momento di svolgimento della sfida.

	delle determinate coordinate cartesiane		
Casella di podio	Spazio all'interno di un gioco contenente l'icona di una squadra in una delle prime tre posizioni	place	Gli stessi di Casella
Casella di gioco	Spazio all'interno di un gioco visitabile da una squadra e che può contenere un'attività, ossia video, task o quiz	num_ord, video	Gli stessi di Casella
Casella serpente	Casella di gioco che fa tornare la squadra ad una determinata posizione	dest	Gli stessi di Casella di gioco
Casella scala	Casella di gioco che fa tornare la squadra ad una determinata posizione	dest	Gli stessi di Casella di gioco
Quiz	Test a risposte multiple che i giocatori possono svolgere	Id, testo, immagine	Id
Risposta quiz	Opzione selezionabile all'interno di un quiz e con un determinato punteggio	num_risp, testo, immagine, punteggio	{num_risp, id (quiz)}
Task	Domanda a risposte aperte che può essere svolta dai giocatori	Id, testo, punteggio	Id
Sfida	Esecuzione di un gioco da parte di due o più squadre in un determinato orario	Id, orario, durata_max, moderata	{id, id (gioco)}
Squadra	Insieme di utenti che possono partecipare alle sfide	nome_squadra	nome_squadra
Utente	Giocatori che possono far parte di due o più squadre	email, nickname, nome, cognome, dataN	Email, nickname
Caposquadra	Utente che all'interno di una sfida moderata è sia giocatore che moderatore della squadra		Gli stessi di Utente
Coach	Utente che all'interno di una sfida moderata non dà risposte ma modera solo		Gli stessi di Utente
Turno	Visita di una casella da	numero_turno,	{id (sfida), id (gioco da

	parte di una squadra in una sfida e punteggio ottenuto da svolgimento di quiz o task	file_consegnato, punteggio_ottenuto, tempo_impiegato, numero_dadi_ottenuti, val_lancio_dadi	sfida) nome_squadra, X, Y, numero_turno}
Risposta utente	Le singole risposte a quiz e task date da un utente all'interno di un turno	file_consegnato	Gli stessi di Utente e Turno

Questa tabella riassume invece la specifica delle associazioni.

Nome	Descrizione	Attributi	Entità collegate
UsatoDa	Insieme di icone utilizzabili in un gioco		Set, Gioco
ParteDi	Set di appartenenza di un icona		Icona, Set
Contiene	Casella facente parte di un gioco		Gioco, Casella
PosizionatoIn	Posizionamento dell'icona di una squadra in una casella podio		Icona, Casella podio
Usa	Task usati in un gioco con tempo massimo	tempo_max	Gioco, Task
Usa	Quiz usati in un gioco con tempo massimo	tempo_max	Gioco, Quiz
Usa	Task contenuto nella casella di gioco		Task, Casella di gioco
Usa	Quiz contenuti nella casella di gioco		Quiz, Casella di gioco
CompostoDa	Risposte selezionabili in un quiz		Quiz, Risposta quiz
SceltaDa	Icona scelta da una squadra		Icona, Squadra
Punteggio	Punti ottenuti da una squadra in una sfida, insieme al numero di dadi ottenuti e dei turni effettuati	punti_tot, num_dadi, turni	Squadra, Sfida
CompostaDa	Squadre di appartenenza di un determinato utente		Squadra, Utente
Partecipa	Partecipazione di un utente ad una sfida		Utente, Sfida
DivisaIn	Insieme di mosse eseguite in un		Sfida, Turno

	determinato momento da una squadra in una sfida		
Esegue	Mosse eseguite da una squadra in una determinata sfida		Squadra, Turno
Visita	Casella visitata in un turno da una determinata squadra		Turno, Casella di gioco
Risponde	Risposte date ai quiz in un determinato turno		Turno, Risposta quiz
ParteDi	Risposte a quiz e task dati dagli utenti di una squadra in un determinato turno		Risposta utente, Turno
Risponde	Risposte date da utenti in una determinata sfida		Utente, Risposta utente
Contiene	Risposte ai quiz date dagli utenti		Risposta utente, Risposta quiz

2.3. Specifica gerarchie di generalizzazione

Una casella deve necessariamente essere una casella di podio oppure una casella di gioco. Tuttavia quest'ultima può non contenere una destinazione, così come un utente non deve necessariamente essere coach o caposquadra.

Entità padre	Entità figlie	Tipologia
Casella	Casella podio, Casella di gioco	Totale/esclusiva
Casella di gioco	Casella serpente, Casella scala	Parziale/esclusiva
Utente	Coach, Caposquadra	Parziale/esclusiva

2.4. Vincoli non esprimibili nel diagramma

Tenendo conto dei requisiti ristrutturati e il corrispondente schema concettuale, si estrapolano i seguenti vincoli.

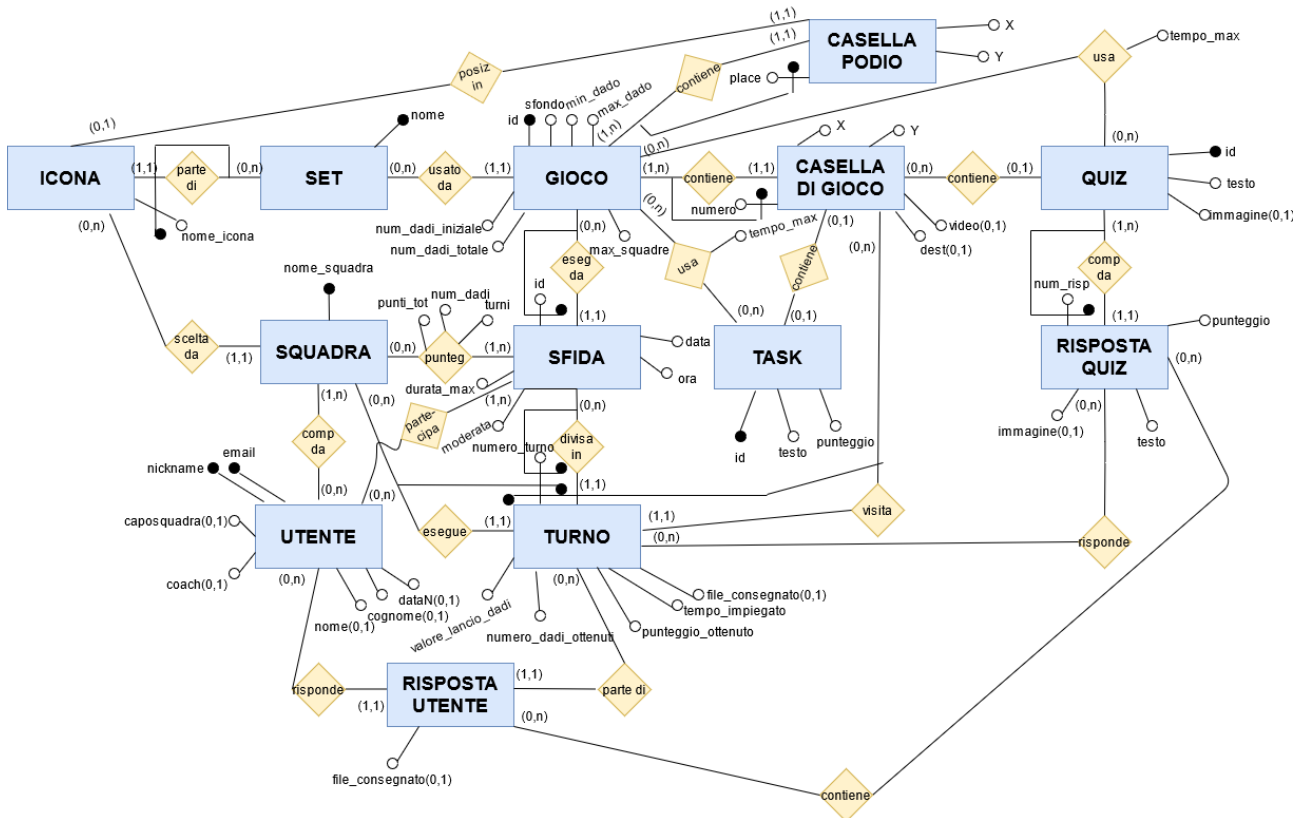
V01	Ad una sfida non possono partecipare più squadre di quelle consentite dal gioco
V02	Ad una casella di gioco non possono essere associati più di 5 quiz a risposte multiple
V03	Un quiz a risposte multiple deve avere almeno due risposte e non più di 9
V04	Ogni quiz deve avere almeno una risposta con punteggio positivo
V05	In una casella scala, la casella di destinazione non può andare oltre il numero massimo
V06	Ad una casella non possono essere associati sia quiz che task

V07	Non possono esserci più di tre caselle podio per gioco
V08	Il valore minimo in un dado dev'essere minore o uguale del valore massimo
V09	Il valore massimo in un dado non dev'essere superiore a 6
V10	Il numero dei dadi ottenibili da una squadra non deve andare oltre il numero massimo consentito
V11	Nomi e icone delle squadre che partecipano alla stessa sfida sono tutti distinti
V12	Un utente non può partecipare a sfide contemporanee

3. Progetto logico

3.1. Schema ER ristrutturato

Sulla base del precedente schema concettuale, si ottiene il seguente schema ER ristrutturato.



3.2. Modifiche al dominio degli attributi e attributi introdotti

Relativamente agli attributi composti, nello schema concettuale erano presenti i seguenti:

- **num_dadi**: contenente il numero di dadi iniziale in un gioco (num_dadi_iniziale) e il limite massimo (num_dadi_totale). Si è deciso rispettivamente di scorporarli in due attributi diversi mantenendo i nomi indicati tra parentesi
- **val_dadi**: esso riguarda sempre l'entità Gioco, e contiene il valore minimo ammissibile in un dado (min_dado) e il valore massimo (max_dado). Anch'essi scorporati negli attributi di cui i nomi tra parentesi
- **orario**: relativo all'entità Sfida, era composto dalla data e dall'ora, con i rispettivi nomi di attributo. Anch'esso è stato diviso in due diversi attributi.

Viste le modifiche fatte sopra, la tabella dei vincoli andrà quindi ad allargarsi in modo tale da preservare il fatto che in origine si trattava di attributi composti.

3.3. Eliminazione gerarchie di generalizzazione

La gerarchia **Casella**, essendo di tipo totale, fa sì che una casella all'interno di un gioco sia una casella visitabile (quindi “di gioco”) oppure una casella in cui inserire l'icona di una squadra in classifica (quindi “podio”), ed essendo che queste hanno entrambe in comune solo le coordinate nel piano (X e Y), si è deciso di eliminare il padre e far sì che un gioco possa avere distintamente caselle di gioco o caselle podio. Inoltre, per evitare chiavi composte da troppi attributi (vedi relazione Turno), entrambe non saranno più identificate dal gioco e le coordinate, ma bensì dal gioco e, rispettivamente, dal numero d'ordine e la posizione del classificato. Bisognerà quindi introdurre un vincolo per evitare due caselle di gioco o podio aventi le stesse coordinate.

La gerarchia **Casella di gioco**, essendo parziale, ed essendo che non è importante da un punto di vista computazionale che si vada indietro con le caselle oppure avanti, porta all'eliminazione di entrambe le entità figlie Casella serpente e Casella scala e l'introduzione di un attributo non obbligatorio “dest” all'interno dell'entità padre.

Ciò che fatto finora porta anche ad eliminare l'attributo “tipo” in quanto basta effettuare un controllo sugli attributi.

La gerarchia **Utente**, essendo anch'essa parziale, porta all'eliminazione delle entità figlie e all'introduzione di due attributi di tipo booleano “coach” e “caposquadra”, i quali porteranno all'introduzione di un vincolo che non porterà un utente a ricoprire entrambi i ruoli.

3.4. Schema logico

Sulla base dei punti precedenti, si ottiene il seguente schema relazionale.

GIOCO (id, max_squadre, sfondo, num_dadi_iniziale, num_dadi_totale, min_dado, max_dado, set_icone^{Set})
SET (nome)
ICONA (nome_icona, nome_set^{Set})
CASELLA_DI_GIOCO (numero, gioco^{Gioco}, X, Y, video_o, dest_o, task_o^{Task})
CASELLA_PODIO (place, gioco^{Gioco}, X, Y, nome_icona_o^{Icona}, nome_set_o^{Icona})
QUIZ (id, testo, immagine_o)
RISPOSTA_QUIZ (num_risp, id_quiz^{Quiz}, testo, immagine_o, punteggio)
TASK (id, testo, punteggio)
GIOCO_HA_QUIZ (gioco^{Gioco}, quiz^{Quiz}, tempo_max)
GIOCO_HA_TASK (gioco^{Gioco}, task^{Task}, tempo_max)
CASELLA_HA_QUIZ (numero^{Casella_di_gioco}, gioco^{Casella_di_gioco}, quiz^{Quiz})
SFIDA (id, gioco^{Gioco}, data, ora, durata_max, moderata)
SQUADRA (nome_squadra, nome_icona_o^{Icona}, nome_set_o^{Icona})
PUNTEGGIO (sfida^{Sfida}, gioco^{Gioco}, squadra^{Squadra}, punti_tot, num_dadi, turni)
UTENTE (email, nickname, nome_o, cognome_o, dataN_o, coach_o, caposquadra_o)
COMPOSIZIONE_SQUADRA (squadra^{Squadra}, utente^{Utente})
UTENTE_PARTECIPA (utente^{Utente}, sfida^{Sfida}, gioco^{Sfida})
TURNO (squadra^{Squadra}, sfida^{Sfida}, gioco^{Sfida}, numero_casella^{Casella_di_gioco}, numero_turno, file_consegnato_o, tempo_impiegato, punteggio_ottenuto, numero_dadi_ottenuti, valore_lancio_dadi)
TURNO_RISPOSTA_QUIZ (squadra^{Turno}, sfida^{Turno}, gioco^{Turno}, numero_casella^{Turno}, numero_turno^{Turno}, risposta^{Risposta_quiz}, quiz^{Risposta_quiz})
RISPOSTA_UTENTE (squadra^{Turno}, sfida^{Turno}, gioco^{Turno}, numero_casella^{Turno}, numero_turno^{Turno}, utente^{Utente},

file_consegnato,)

RISPOSTA_UTENTE_QUIZ (squadra^{Risposta_utente}, sfida^{Risposta_utente}, gioco^{Risposta_utente}, numero_casella^{Risposta_utente}, numero_turno^{Risposta_utente}, risposta^{Risposta_quiz}, quiz^{Risposta_quiz})

3.5. Verifica di qualità e ottimizzazioni

Una prima ottimizzazione può essere considerabile la chiave di **CASELLA_DI_GIOCO** definita precedentemente, in quanto permette, laddove essi siano chiave esterna (ad esempio su **TURNO**), di non doversi “portare dietro” entrambe le coordinate, ma solo il numero di casella.

La relazione **UTENTE_PARTECIPA** è inutile in quanto è possibile ottenere lo stesso risultato tramite operazioni di JOIN, oltre a costringere ad una (costosa) operazione di inserimento dei rispettivi utenti ad ogni sfida, e considerato che la verifica del vincolo che due utenti non possono partecipare a due sfide contemporanee non è un'operazione del carico di lavoro (ma bensì un trigger) la si può eliminare.

La relazione **PUNTEGGIO** ha attributi i cui valori sono ricavabili dai turni, ma essendo prevista l'implementazione di un trigger che calcola i dati delle partite (punteggio totale, numero dadi...) la si tiene.

La relazione **TURNO** ha l'attributo file_consegnato che è ripetuto in quanto esso è per forza presente in **RISPOSTA_UTENTE**, ma essendo che lo svolgimento di un turno consiste, rispettivamente nel

- visitare una casella, la quale porta ad una relazione di inserimento nella tabella **TURNO**
- salvare le risposte di quiz e task degli utenti in un determinato turno, il quale porta ad un inserimento in **RISPOSTA_UTENTE**
- salvare in **TURNO** il task e determinare punteggi, dadi ottenuti e valore lancio dei dati, la quale porta ad un'operazione di aggiornamento (e di conseguenza nell'attivazione del trigger)

La si può lasciare così per evitare di “complicare” lo svolgimento di un gioco. Notare che in caso di sfida non moderata, è possibile eventualmente non utilizzare le risposte degli utenti, in quanto si propone il primo task caricato, non portando quindi sempre a ripetere l'informazione, il che può essere considerato un compromesso.

Lo stesso concetto vale per i quiz nel caso delle relazioni **TURNO_RISPOSTA_QUIZ** e **RISPOSTA_QUIZ_UTENTE**, anche relativamente alle sfide non moderate.

La relazione **GIOCO** può essere decomposta, questo a causa delle (tante) informazioni relative ai dadi, le quali dipendono dall'esistenza dei dadi nel gioco, ossia il numero minimo di dadi a squadra. Quindi ipotizzando una relazione con le seguenti dipendenze funzionali:

id → max_squadre, sfondo, set_icone

id, num_dadi_iniziale → num_dadi_totale, min_dado, max_dado

Si potrebbero creare due diverse relazioni corrispondenti alle dipendenze funzionali di cui sopra, in modo da normalizzare in BCNF.

Tuttavia, il carico di lavoro (vedere sezione 4) contiene due operazioni sulla relazione **GIOCO**, di cui una già con il join, per cui se si mantenesse questo schema si avrebbero due tabelle in join per la prima operazione e tre tabelle in join per la terza, per cui si alzerebbe troppo il costo di tali operazioni. Allo stesso tempo però, mantenere tutti i dati relativi ai dadi su gioco può portare a spreco di memoria in caso di giochi che non usano dadi, per cui si decide di creare un compromesso, nel quale si mantiene in maniera ridondante il numero di dadi richiesti sulla tabella gioco, e si crea una relazione per le altre informazioni sui dadi, qualora questi siano effettivamente presenti.

La porzione di schema relazionale diventa quindi come segue.

GIOCO (id, max_squadre, sfondo, dadi_richiesti, set_icone^{Set})

INFO_DADI (gioco^{Gioco}, numero_iniziale, limite_massimo, valore_minimo, valore_massimo)

Ovviamente ad ogni gioco corrisponderà al massimo un solo insieme di informazioni sui dadi e si introdurrà nella tabella, oltre ai vincoli relativi al limite massimo e ai valori dei dadi, un vincolo che controlla l'uguaglianza del numero di dadi richiesti in entrambe le relazioni ($\text{dadi_richiesti} = \text{numero_iniziale}$).

Il numero di dadi richiesti non sarà un attributo chiave ma se, alla creazione di un gioco questo sarà impostato a valore 0, non ci sarà bisogno di creare la corrispondente istanza di INFO_DADI.

3.6. Modifiche all'elenco dei vincoli

Sulla base delle ristrutturazioni fatte in precedenza, si introducono nella precedente tabella i seguenti vincoli, a cui si ne aggiungono due ulteriori per evitare valori senza senso nel dominio considerato, tutti relativi a possibili valori negativi o nulli.

V13	Nello stesso gioco non possono esserci caselle con le stesse coordinate, indipendentemente dal fatto che siano di gioco oppure podio
V14	Un utente non può ricoprire sia il ruolo di coach che di caposquadra
V15	Se nel gioco sono presenti dadi, il numero di dadi richiesti nelle informazioni del gioco deve essere uguale al numero iniziale nelle informazioni sui dadi relativi allo stesso gioco
V16	Durate massime delle sfide, numeri di caselle di gioco, punteggi massimi dei task e turni non possono essere nulli o negativi
V17	Le coordinate X devono essere comprese tra 0 e 1080 mentre le coordinate Y tra 0 e 720

4. Progetto fisico

4.1. Analisi del carico di lavoro

La specifica prevede il seguente carico di lavoro, di cui se ne riporta sia la descrizione in linguaggio naturale che la traduzione in SQL, sulla base del risultato dello schema relazionale, sulle relazioni Gioco e Sfida.

- Determinare l'identificatore dei giochi che coinvolgono al più quattro squadre e richiedono l'uso di due dadi.

```
SELECT id
FROM Gioco
WHERE max_squadre = 4 AND dadi_richiesti = 2
```

- Determinare l'identificatore delle sfide relativo a un gioco A di vostra scelta 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.

```
SELECT id
FROM Sfida
WHERE gioco = A AND ((data BETWEEN '01/01/2021' AND '31/01/2021' AND durata_max > 120) OR
(data BETWEEN '01/03/2021' AND '31/03/2021' AND durata_max = 30))
```

- 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.

```
SELECT Sfida.id, Gioco
FROM Sfida JOIN Gioco ON gioco = Gioco.id
WHERE durata_max > 120 AND dadi_richiesti >= 2
```

Considerato che tra le interrogazioni ce n'è una che agisce su entrambe le tabelle tramite join, l'obiettivo è quello di equilibrare tutte e tre le interrogazioni, assumendo che tra di esse non ci siano operazioni che vengono eseguite più di altre, per cui non può essere una buona scelta ottimizzare un'interrogazione "troppo" a discapito di un'altra.

La **prima** interrogazione prevede due condizioni di selezione, le quali vanno soddisfatte entrambe, una sul numero di squadre, e una sul numero minimo di dadi richiesti. Nel dominio considerato, si presume che entrambe abbiano un fattore di selettività simile, questo perché il numero di squadre sarà sicuramente 2, ma non sarà troppo grande (5 o 6 sono un numero più che sufficiente per un gioco). Lo stesso discorso può essere relativo ai dadi, che possono non esserci (0 quindi), ma essendo questo un numero minimo si presume che non sia troppo grande (3 o 4). Per cui sicuramente può convenire un indice multiattributo, ma per la clusterizzazione è necessario aspettare l'ultima interrogazione.

La **seconda** interrogazione prevede una condizione di selezione su gioco (che dev'essere soddisfatta, e verrà sostituito con un identificatore a scelta nel file SQL corrispondente) e una sulla data e la durata, la cui prima che varia sempre in un intervallo, ma entrambe possono soddisfare due diversi intervalli.

Essa ha un fattore di selettività in ordine decrescente: in particolare, il gioco è molto selettivo (considerato che la specifica richiede di riempire molto la tabella gioco), così come per la data si hanno due range di 31

giorni per entrambi i mesi. Per quanto riguarda la durata, si presume che verranno privilegiati numeri multipli di 10, e che non vadano (troppo) oltre le due ore di gioco, per cui si assume un fattore di selettività più vicino all'1 rispetto al mese.

Sulla base di queste assunzioni, un indice multiattributo sulle tre condizioni nell'ordine proposto è conveniente, così come sicuramente anche la clusterizzazione.

La **terza** interrogazione è sicuramente quella più complicata da ottimizzare, se si considerano le scelte fatte sulle prime due interrogazioni sulle stesse tabelle. In questo caso abbiamo due condizioni di selezione i cui attributi sono inclusi negli indici multiattributo ipotizzati in precedenza. Può quindi essere utile creare un indice per entrambi, di cui quello su `durata_max` non clusterizzato, vista la scelta di clusterizzare quello multiattributo.

Riguardo invece all'indice sul numero di dadi richiesti, si può decidere se clusterizzare questo oppure l'indice multiattributo creato per la prima interrogazione. Considerato che la creazione di un indice e la sua clusterizzazione porta al sistema a considerare diversi algoritmi di join (tra cui l'index nested loop, o anche il merge join), ed essendo che il fattore di selettività dell'indice multiattributo è più o meno equivalente, si decide quindi di clusterizzare l'indice sul numero di dadi richiesti.

4.2. Elenco degli indici scelti

Viste le considerazioni fatte nel punto precedente, si decide quindi di creare i seguenti indici:

- Un indice multiattributo su Gioco, definito sugli attributi `max_squadre` e `dadi_richiesti`, non clusterizzato
- Un indice a singolo attributo su Gioco, definito sull'attributo `dadi_richiesti`, clusterizzato
- Un indice multiattributo su Sfida, definito sugli attributi `gioco`, `data` e `durata_max`, clusterizzato
- Un indice a singolo attributo su Sfida, definito sull'attributo `durata_max`, non clusterizzato

Tutti gli indici in questione saranno creati come indici ad albero, indipendentemente dal fatto che possano essere considerati indici hash. La decisione è da ricondursi principalmente al fatto che, essendo che le relazioni Gioco e Sfida hanno bisogno di molti dati per verificare i piani di esecuzione scelti dal sistema, alcune tuple potrebbero essere inserite in area di overflow, il che porterebbe a non avere una stima molto precisa delle scelte effettuate.