Michele Frattini
S4878744

# Progetto Base di Dati

## Parte I

Michele Frattini
S4878744

# 1. Requisiti ristrutturati

*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.*

Creazione dell'entità Gioco, con attributi: identificatore (id), numero massimo di squadre (maxsq), plancia (immagine di sfondo).

Creazione dell'entità Icona, con attributi: dimensione (dim), nome (nomei) e tema (set).

Aggiunto anche l'attributo nome dentro l'entità Gioco.

Chiave primaria entità Gioco -> id.

Chiave primaria entità Icona -> nomei.

Ogni gioco ha n (un certo numero di) caselle, creazione relazione Appartiene che collega Casella e Gioco; Gioco-Casella (1,n).

---

*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.*

Creazione dell'entità Casella, con attributi: coordinata (attributo composto da coordinata x e y (cx e cy)), tipologia e numero d'ordine (nordine).

L'attributo tipologia potrà assumere come valori start (ST) e arrivo (AR).

Ogni casella appartiene a un unico gioco, Casella-Gioco (1,1).

Set di icone può essere utilizzato in più giochi, creazione relazione Contiene che collega Gioco con Icona; Icona-Gioco (1,n).

---

*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.*

L'attributo tipologia potrà assumere come valori primo posto (PP), secondo posto (SP) e terzo posto (TP).

---

*A ogni casella (non podio) possono essere associati: • Un breve video • Uno o più (massimo 5) quiz a risposta multipla • Un task*

Aggiunto attributo video in Casella.
Creazione gerarchia con entità padre Compito e entità figlie Quiz e Task.
Creazione relazione Puoavere che collega Casella a Compito; Casella-Compito (1,5).

---

*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.*

Aggiunto attributo testo a Compito (Sia Quiz che Task hanno in comune l'attributo testo, quindi ho riunito nell'entità padre).
Aggiunto attributo immagine a Quiz.
Creazione dell'entità Risposta, con attributi: testo, immagine (imm), punteggio (punt).
Aggiunto attributo punteggio (punt) a Task.

---

*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.*

Creazione dell'entità File. (Admin sarà successivamente creato, andando avanti nel testo, come entità figlia della gerarchia con entità padre Utente. Utente sarà poi collegata con una relazione a File).
Creazione dell'associazione Ha che collega Compito a Gioco che ha come attributo il tempo massimo di risposta (tempomax) (relazione molti a molti) (Squadra sarà successivamente creato, andando avanti nel testo, come entità).

---

*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*

L'attributo tipologia può assumere anche i valori serpente (SE) e scala (SC).
Aggiunto attributo casella di destinazione (casdes) in Casella.

---

*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 valore minimo e un valore massimo (non superiore a sei).*

Creazione entità Dado.
Creazione dell'associazione Usa che collega Gioco a Dado; Gioco-Dado (0,n) (Usa sarà un'associazione molti a molti).
Aggiunto attributi valore minimo (valmin) e valore massimo (valmax) a Dado.

---

*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 valore ottenuto con i dadi.*

Creazione entità Lancio, collegata, con la nuova associazione Tiro, a Casella; e con la nuova associazione Eseguito, a Dado.
Aggiunta attributo somma valori dei dadi lanciati (=valore del tiro dei dadi) (punt) a Lancio.
Creazione associazione Possiede che collega Dado e squadra (relazione molti a molti) (modificare numero dei dadi a disposizione della squadra).

---

*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.*

Creazione entità Sfida con attributo composto quando (formato da attributi data e ora).
Creazione associazione Basata che collega Sfida a Gioco (ogni sfida è basata su un gioco); Sfida-Gioco (1,1).
Aggiunta attributo durata massima (durmax) a Sfida.
Creazione associazione Partecipa che collega Sfida a Squadra (ogni sfida partecipano un certo numero di squadre); Sfida-squadra (2,n) (relazione molti a molti).
Creazione entità Squadra con attributo nome squadra (nomesq).
Creazione associazione Utilizza che collega Squadra a Icona (1,1) (relazione uno a uno).
Chiave primaria entità Squadra -> nomesq.

---

*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.*

Creazione associazione Gioca che collega Squadra a Utente; Squadra-Utente (1,n).
Creazione gerarchia padre Utente con entità figlie Giocatore, Mod e Admin e avente come attributi: indirizzo e-mail (email), nickname (nick) e come attributo composto generalità, costituito da nome, cognome (cogn) e data di nascita (datan).
Ogni utente può appartenere a una o più squadre; Utente-Squadra (1,n) (relazione molti a molti).
L'attributo tiposfi in Sfida può assumere il valore Moderata (MO).
Mod è una gerarchia padre con entità figlie Coach e Caposquadra.
L'attributo tiposfi in Sfida può assumere il valore non Moderata (NM).

---

*Per ogni sfida, si terrà conto delle caselle visitate, dei lanci di dadi e relativi 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.*

Per le caselle visitate, lanci dei dadi e valori ottenuti, si considera la relazione Lancio con tutte le sue chiavi esterne.
Per le risposte date, si usa l'associazione molti a molti Vota, con l'attributo tempo; che collega Risposta a Giocatore.
Creazione dell'associazione InviaConfermaValida che collega Utente a File, e l'aggiunta in File degli attributi approvato (booleano) (valido) e tempo.

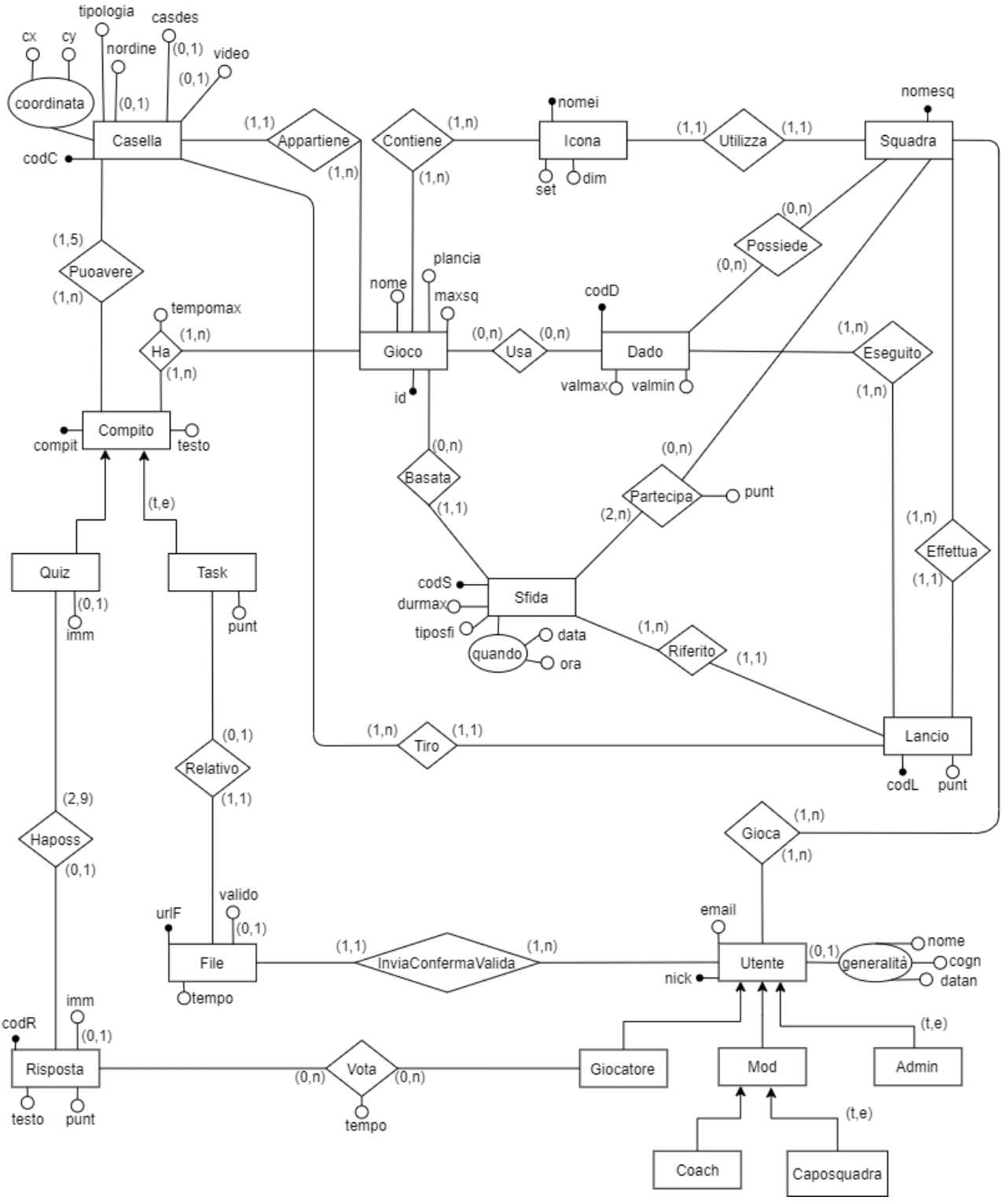
---

Michele Frattini
S4878744

# 2. Progetto Concettuale

a.

## Schema ER



Michele Frattini
S4878744

b.

# Dizionario delle Entità e domini

| Nome | Descrizione | Attributi | Identificatori |
|---|---|---|---|
| Gioco | Giochi disponibili nell'applicazione | id (NUMERIC(4)), nome(VARCHAR(100)), plancia(VARCHAR(100)), maxsq(DECIMAL(4)) | id |
| Casella | Caselle relati ai vari giochi | codC(NUMERIC(4)), nordine(NUMERIC(4)), video(VARCHAR(8)), tipologia(CHAR(2)), casedes(NUMERIC(4)), coordinata | codC |
| Compito | Compito da svolgere quando si è in una casella | compit(NUMERIC(4)), testo(VARCHAR(200)) | compit |
| Task | Task da svolgere quando si è in una casella | punt(INT) | Gli stessi di Compito |
| Squadra | Squadra che partecipa a una sfida per un gioco | nomesq(VARCHAR(15)) | nomesq |
| Icona | Icona usata da una squadra come pedina per un gioco | nomei(VARCHAR(15)), set(VARCHAR(20)), dim(DECIMAL(5)) | nomei |
| Quiz | Quiz da rispondere quando si è in una casella | imm(VARCHAR(8)) | Gli stessi di Compito |
| Risposta | Possibili risposte a un quiz | codR(NUMERIC(4)), punt(INT), imm(VARCHAR(8)), testo(VARCHAR(200)) | codR |
| Sfida | Sfida online relativa a un gioco | codS(NUMERIC(4)), durmax(TIME), tiposfi(CHAR(2)), quando | codS |
| Dado | Dadi utilizzabili in un gioco e tenuti dalle squadre | codD(NUMERIC(4)), valmin(NUMERIC(1)), valmax(NUMERIC(1)) | codD |

| | | | |
|---|---|---|---|
| Lancio | Lancio effetuato in una casella da una squadra con dei dadi | codL(NUMERIC(4)), punt(DECIMAL(2,2)) | codL |
| Utente | Insieme di persone che interagiscono con l'applicazione | nick(VARCHAR(10)), email(VARCHAR(20)), generalità | nick |
| Giocatore | Giocatore che gioca una sfida | | Gli stessi di Utente |
| Mod | Moderatore che modera la sfida | | Gli stessi di Utente |
| Admin | Admin che valida il file consegnato dai giocatori | | Gli stessi di Utente |
| Coach | Moderatore che modera la sfida | | Gli stessi di Utente |
| Caposquadra | Moderatore che modera e gioca la sfida | | Gli stessi di Utente |
| File | File conseganto da un giocatore per eseguire il task | urlF(VARCHAR(30)), tempo(TIME), valido(BOOLEAN) | urlF |

Michele Frattini
S4878744

# Dizionario delle associazioni e domini

| Nome | Descrizione | Attributi | Identificatori |
|---|---|---|---|
| Contiene | Associazione molti a molti che collega l'entità Gioco e Icona | | idG(id, Gioco), icona(nomei, Icona) |
| Puoavere | Associazione molti a molti che collega l'entità Casella e Compito | | codC(codC,Casellla), compit(compit,Compito) |
| Ha | Associazione molti a molti che collega l'entità Gioco e Compito | tempomax(TIME) | idG(id, Gioco), compit(compit, Compito) |
| Usa | Associazione molti a molti che collega l'entità Gioco e Dado | | codG(id,Gioco), codD(codD,Dado) |
| Possiede | Associazione molti a molti che collega l'entità Dado e Squadra | | codD(codD,Dado), nomesq(nomesq,Squadra) |
| Partecipa | Associazione molti a molti che collega l'entità Sfida e Squadra | punt(INT) | codS(codS,Sfida), nomesq(nomesq,Squadra) |
| Eseguito | Associazione molti a molti che collega l'entità Dado e Lancio | | codL(codL,Lancio), codD(codD,Dado) |
| Gioca | Associazione molti a molti che collega l'entità Squadra e Utente | | nomesq(nomesq,Squadra), nickgioc(nick,Utente) |
| Vota | Associazione molti a molti che collega l'entità Utente e Risposta | tempo(TIME) | nickgioc(nick,Utente), codR(codR,Risposta) |

Michele Frattini
S4878744

c.

# Vincoli non esprimibili nel diagramma

Una casella di tipologia ST deve avere come attributo nordine uguale a 0.

Una casella di tipologia AR deve avere come attributo nordine uguale al valore massimo per quel gioco.

Una casella di tipologia PP, SP, TP deve avere nessun valore come attributo nordine.

Ogni quiz deve prevedere almeno una risposta con punteggio positivo.

Se la squadra non fornisce risposta al quiz e al task entro il tempo massimo, il punteggio ottenuto è 0.

Una casella di tipologia SE o SC deve avere un valore nell'attributo casdes (non si accettano valori nulli se tipologia = SE o SC).

Una casella di tipologia SE deve avere un valore nell'attributo casdes minore rispetto a nordine.

Una casella di tipologia SC deve avere un valore nell'attributo casdes maggiore rispetto a nordine.

Il valmax che può assumere un dado deve essere < 6.

Una casella con casdes = valore non nullo, non si lanceranno i dadi.

Una casella può contenere o solo task o solo quiz, non possono essere presenti entrambi.

A una sfida devono partecipare almeno 2 squadre.

Le icone delle squadre che partecipano alla stessa sfida devono essere tutte diverse.

Un utente può appartenere a una sola squadra tra quelle che partecipano alla stessa sfida.

Se una sfida è MO, ogni squadra deve avere un solo utente coach o un solo utente caposquadra.

In sfide NM, le risposte della squadra al quiz sono ottenute prendendo dalla risposta più votata dagli utenti della squadra.

In sfide NM, si considerano i primi task consegnati da un giocatore per tutta la squadra.

In sfide MO, il moderatore conferma la soluzione proposta da un giocatore per il task.

Il numero di squadre che partecipano a una sfida, riferita a un gioco, deve essere inferiore o uguale al valore nell'attributo maxsq di quel gioco.

Una casella può avere al massimo 5 quiz o 1 solo task.

Nelle associazioni Partecipa e Gioca vengono mantenute solo istanze relative a utenti che fanno parte attualmente in quella squadra e squadre che stanno ancora effettivamente giocando una sfida. Utenti che non giocano più in una squadra e partite relative a sfide concluse vengono eliminate dalle corrispettive associazioni.

Un gioco può usare solo icone dello stesso set.

Michele Frattini
S4878744

d.

# Specifica dei tipi di gerarchie di generalizzazione

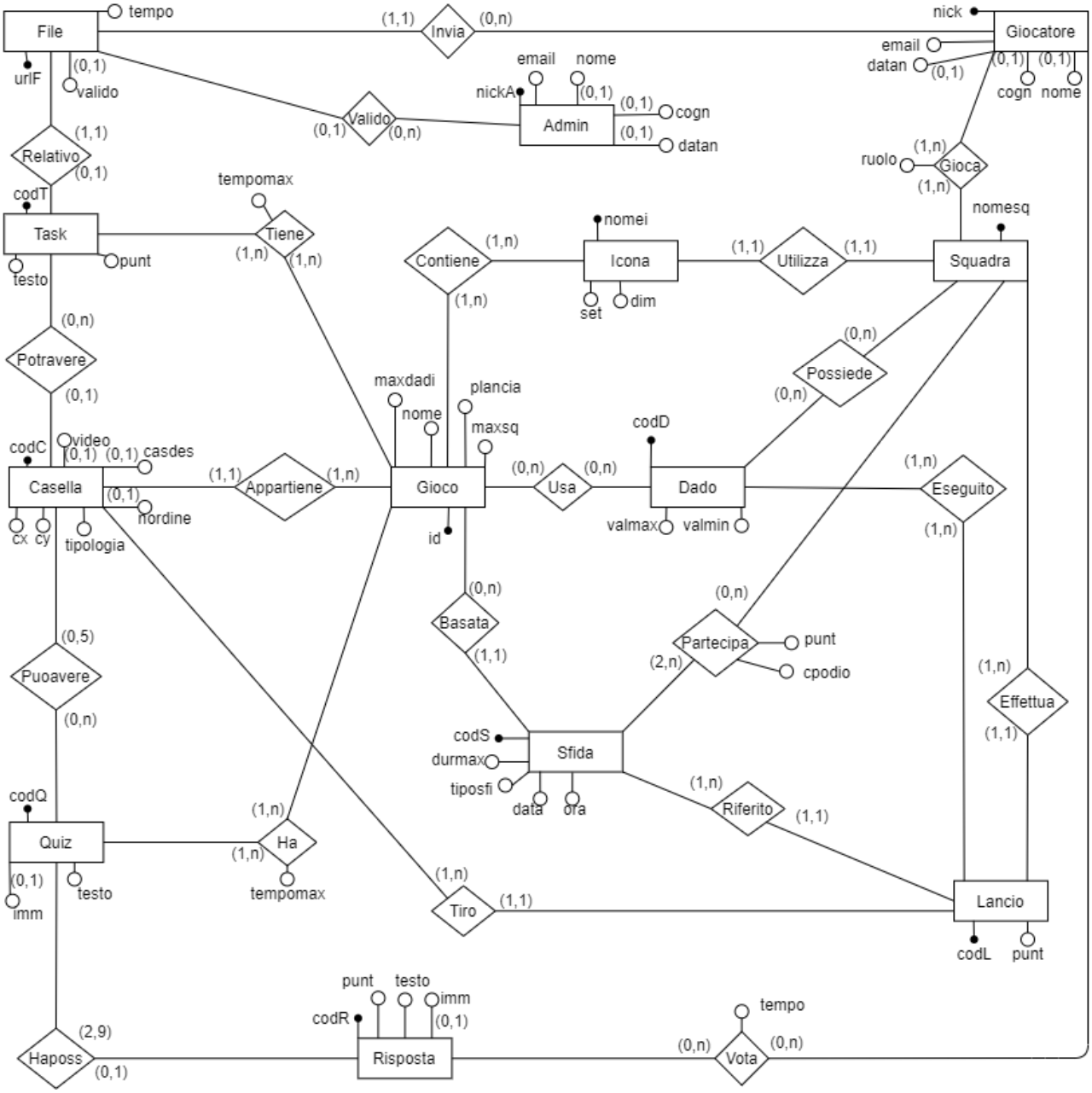
| Entità padre | Entità figlie | Tipologia |
|---|---|---|
| Compito | Quiz, task | totale/esclusiva |
| Utente | Giocatore, mod, admin | totale/esclusiva |
| Mod | Coach, Caposquadra | totale/esclusiva |

Michele Frattini
S4878744

# 3. Progetto logico

a.

## Schema ER ristrutturato



Michele Frattini
S4878744

b.

# Modifiche domini attributi precedenti

| Nome | Descrizione | Attributi | Identificatori |
|---|---|---|---|
| Gioco | Giochi disponibili nell'applicazione | id (NUMERIC(4)), nome(VARCHAR(100)), plancia(VARCHAR(100)), maxsq(DECIMAL(4)), **maxdadi (INT)** | id |
| Casella | Caselle relati ai vari giochi | codC(NUMERIC(4)), nordine(NUMERIC(4)), video(VARCHAR(8)), tipologia(CHAR(2)), casedes(NUMERIC(4)), **cx(DECIMAL(15)), cy(DECIMAL(15))** | codC |
| Task | Task da svolgere quando si è in una casella | **codT(NUMERIC(4)), testo(VARCHAR(200)), punt(INT)** | **codT** |
| Quiz | Quiz da rispondere quando si è in una casella | **codQ(NUMERIC(4)), imm(VARCHAR(8)), testo(VARCHAR(200))** | **codQ** |
| Sfida | Sfida online relativa a un gioco | codS(NUMERIC(4)), durmax(TIME), tiposfi(CHAR(2)), **data(DATE), ora(TIME)** | codS |
| Giocatore | Giocatore che gioca una sfida | **nick(VARCHAR(10)), email(VARCHAR(20)), dataN(DATE), cognome(VARCHAR(20)), nome(VARCHAR(20))** | **nick** |
| Admin | Moderatore che modera la sfida | **nickA(VARCHAR(10)), email(VARCHAR(20)), dataN(DATE), cognome(VARCHAR(20)), nome(VARCHAR(20))** | **nickA** |
| Puoavere | Associazione molti a molti che collega l'entità Casella e **Quiz** | | codC(codC,Casellla), **codQ(codQ,Quiz)** |

| | | | |
|---|---|---|---|
| Ha | Associazione molti a molti che collega l'entità Gioco e **Quiz** | | idG(id, Gioco), **codQ(codQ,Quiz)** |
| Partecipa | Associazione molti a molti che collega l'entità Sfida e Squadra | punt(INT), **cpodio(INT)** | codS(codS,Sfida), nomesq(nomesq,Squadra) |
| Gioca | Associazione molti a molti che collega l'entità Squadra e **Giocatore** | **ruolo(CHAR(2))** | nomesq(nomesq,Squadra), **nickgioc(nick,Giocatore)** |
| Vota | Associazione molti a molti che collega l'entità **Giocatore** e Risposta | | **nickgioc(nick,Giocatore),** codR(codR,Risposta) |

Gli altri domini sono rimasti invariati rispetto alla versione precedente.

(Le parole in **grassetto** sono le modifiche effettuate nel dizionario).

c.

# Modifiche ai vincoli precedenti

Il vincolo “Una casella può avere al massimo 5 quiz o 1 solo task” viene eliminato per via dell’eliminazione della gerarchia Compito e l’introduzione delle due associazioni che collegano Casella a Task e Casella a Quiz.

Con queste due associazioni il vincolo è già specificato nello schema ER ristrutturato.

Aggiunto ulteriore vincolo per via dell’aggiunta dell’attributo maxdadi nell’entità Gioco:

“Ogni Gioco deve usare un numero di dadi non superiore a quanto specificato nell’attributo maxdadi dell’entità Gioco”.

Gli altri vincoli rimangono invariati.

Michele Frattini
S4878744

d.

# Scelte fatte per eliminare gerarchie

Per la gerarchia Compito-Quiz-Task ho deciso di eliminare l'entità padre Compito e lasciare intatte le due entità figlie Quiz e Task nello schema ristrutturato. Ho collegato con una nuova associazione Puoavere, l'entità Casella e Quiz con: Casella-Quiz (0,5) e Quiz-Casella (0,n). Poi ho creato un'altra associazione Potravere che collega l'entità Casella e Task con: Casella-Task (0,1) e Task-Casella (0,n). L'entità padre Compito era troppo generica e non portava alcuna informazione in più rispetto alle due entità figlie Task e Quiz.

Per la gerarchia Utente ho deciso di tenere nello schema ristrutturato solo l'entità Giocatore e Admin. Inoltre ho introdotto all'interno dell'associazione Gioca, che ora è collegata a Squadra e Giocatore; l'attributo ruolo, che identifica se un giocatore, per una determinata Sfida; è un utente, un coach o un caposquadra. Invece l'entità Admin, che ora è solo collegata con l'associazione Valido a File; ha gli stessi attributi di Giocatore. L' entità padre Utente, nel progetto concettuale, raggruppava sotto di se troppe entità figlie, che nell'applicazione svolgevano attività diverse tra loro. Quindi per rendere lo schema più chiaro ho deciso di dividere la gerarchia, lasciando solo l'entità più importanti.

Michele Frattini
S4878744

e.

# Schema logico

Gioco(<u>id</u>, nome, plancia, maxsq, maxdadi)

Casella(<u>codC</u>, nordine$_O$, *codG$^{gioco}$*, video$_O$, tipologia, casdes$_O$ ,*cx, cy*, codT$^{Task}$)

Task(<u>codT</u>, testo, punt)

Squadra(<u>nomesq</u>, nomei$^{Icona}$)

Icona(<u>nomei</u>, set, dim)

Quiz(<u>codQ</u>, testo, imm$_O$)

Risposta(<u>codR</u>, punt, imm$_O$, testo, codQ$^{Quiz}$)

Sfida(<u>codS</u>, durmax, tiposfi, data, ora, codG$^{Gioco}$)

Dado(<u>codD</u>, valmin, valmax)

Lancio(<u>codL</u>, punt, squad$^{Squadra}$, codC$^{Casella}$, codS$^{Sfida}$)

Giocatore(<u>nick</u>, email, nome$_O$, cogn$_O$, datan$_O$)

File(<u>urlF</u>, valido$_O$, codT$^{Task}$, nickgioc$^{Giocatore}$, Admin$^{Admin}$, tempo)

Admin(<u>nickA</u>, email, nome$_O$, cogn$_O$, datan$_O$)

Contiene(<u>idg$^{Gioco}$, nomei$^{Icona}$</u>)

Ha(<u>id$^{Gioco}$, codQ$^{Quiz}$</u>, tempomax)

Tiene(<u>id$^{Gioco}$, codT$^{Task}$</u>, tempomax)

Usa(<u>codG$^{Gioco}$, codD$^{Dado}$</u>)

Puoavere(<u>codC$^{Casella}$, codQ$^{Quiz}$</u>)

Possiede(<u>codD$^{Dado}$, nomesq$^{Squadra}$</u>)

Eseguito(<u>codL$^{Lancio}$, codD$^{Dado}$</u>)

Partecipa(<u>codS$^{Sfida}$, nomesq$^{Squadra}$</u>, punt, cpodio)

Vota(<u>nickgioc$^{Giocatore}$, codR$^{Rsiposta}$</u>, tempo)

Gioca(<u>nickgioc$^{Giocatore}$, nomesq$^{Squadra}$,</u> ruolo )

Michele Frattini
S4878744

f.

# Verifica di qualità dello schema

Gioco(id, nome, plancia, maxsq, maxdadi)
Id -> nome, plancia, maxsq, maxdadi
Chiave id

Casella(codC, nordine$_O$, *codG$^{gioco}$*, video$_O$, tipologia, casdes$_O$ ,*cx, cy*, codT$^{Task}$)
codC -> nordine, id, video, tipologia, casdes ,cx, cy, codT
Chiave codC

Task(codT, testo, punt)
codT -> testo, punt
Chiave codT

Squadra(nomesq, nomei$^{Icona}$)
nomesq -> nomei
Chiave nomesq

Icona(nomei, set, dim)
nomei -> set, dim

Quiz(codQ, testo, imm$_O$)
codQ -> testo, imm
Chiave codQ

Risposta(codR, punt, imm$_O$, testo, codQ$^{Quiz}$)
codR -> punt, imm, testo, codQ
Chiave codR

Sfida(codS, durmax, tiposfi, data, ora, codG$^{Gioco}$)
codS -> durmax, tiposfi, data, ora, id
Chiave codS

Dado(codD, valmin, valmax)
codD -> valmin, valmax
Chiave codD

Lancio(codL, punt, squad$^{Squadra}$, codC$^{Casella}$, codS$^{Sfida}$)
codL -> punt, nomesq, codC, codS

Chiave codL

Giocatore(nick, email, nome$_O$, cogn$_O$, datan$_O$)
nick -> email, nome, cogn, datan
Chiave nick

File(urlF, valido$_O$, codT$^{Task}$, giocat$^{Giocatore}$, Admin$^{Admin}$, tempo)
urlF -> valido, codT, nick, nickA, tempo
Chiave urlF

Admin(nickA, email, nome$_O$, cogn$_O$, datan$_O$)
nickA -> email, nome, cogn, datan
Chiave nickA

Ha(id$^{Gioco}$, codQ$^{Quiz}$, tempomax)
id, codQ -> tempomax
Chiave id, codQ

Tiene(id$^{Gioco}$, codT$^{Task}$, tempomax)
id, codT -> tempomax
Chiave id, codT

Partecipa(codS$^{Sfida}$, nomesq$^{Squadra}$, punt, cpodio)
codS, nomesq, -> punt, cpodio
Chiave codS, nomesq

Vota(nickgioc$^{Giocatore}$, codR$^{Rsiposta}$, tempo)
nick, codR -> tempo
Chiave nick, codR

Gioca(nickgioc$^{Giocatore}$, nomesq$^{Squadra}$, ruolo )
nick, nomesq -> ruolo
Chiave nick, nomesq

Lo schema è normalizzato in forma normale di Boyce Codd perché ogni dipendenza contiene a sinistra la chiave della relazione a cui si riferisce.

Un'ottimizzazione effettuata tenendo in considerazione il carico di lavoro è l'introduzione dell'attributo maxdadi nell'entità Gioco. Questo per ottimizzare le query considerate più frequenti che richiedevano l'uso di un certo numero di dadi, quindi evitando di utilizzare nelle query la tabella Dado, il join tra l'entità Gioco e Usa e le fuzioni count e group by.

Michele Frattini
S4878744

# 4. Progetto fisico

Elenco indici:

1. Indice ordinato (maxsq) su Gioco non clusterizzato
2. Indice ordinato (maxdadi) su Gioco clusterizzato

Questi primi due indici servono per ottimizzare la prima interrogazione del carico di lavoro. Entrambi questi due attributi, nelle interrogazioni del carico di lavoro, vengono utilizzati per fare ricerche su un certo intervallo (maxsq <= 4 e, ora no, ma nell'ultima query anche maxdadi >=2) quindi, dalla teoria, si possono usare solo indici ordinati e non quelli hash. Visto che è possibile clusterizzare solo un indice per tabella, ho deciso di clusterizzare l'indice ordinato maxdadi perché viene utilizzato per ben due query del carico di lavoro (anche nella terza) rispetto all'indice ordinato maxsq che viene usato solo in questa.

3. Indice ordinato (codG) su Sfida non clusterizzato
4. Indice ordinato (data) su Sfida non clusterizzato
5. Indice ordinato (durmax) su Sfida clusterizzato

Questi ulteriori tre indici servono invece per ottimizzare la seconda interrogazione del carico di lavoro. Volevo utilizzare per l'attributo codG l'indice hash perché, questa query usa una condizione di selezione (codG='0001') e, dalla teoria in questo caso, un indice hash può garantire migliori prestazioni rispetto un indice ordinato (se non c'è la presenza di overflow, ma non è questo il caso). Inoltre volevo anche sceglierlo con questa tipologia in vista dell'ultima query (la prossima/la terza) la quale c'è la presenza di un join di uguaglianza che fa sempre uso dell'attributo codG di Sfida (S.codG=G.id), e anche in quest'ultimo caso l'indice hash può garantire migliori prestazioni rispetto un indice ordinato (senza presenza di overflow). Ma, il compilatore di pgAdmin 4 mi ha sollevato un warning dicendo che questo indice hash non è ottimale per l'esecuzione delle funzioni che ne fanno uso in seguito. Per ciò ho cambiato la tipologia in ordinato. Gli altri due indici invece sono ordinati perché i corrispettivi attributi vengono utilizzati per fare ricerche su un certo intervallo e quindi la scelta della tipologia è analoga alla prima interrogazione. Ho deciso di clusterizzare l'indice ordinato durmax per lo stesso motivo già sopra citato per cui ho clusterizzato l'indice maxdadi.

Per la terza interrogazione non servono creare nuovi indici perché sono riutilizzati il secondo, il terzo e il quinto indice per le analoghe motivazioni già descritte precedentemente.