

# Relazione progetto di **Basi di Dati e Sistemi Informativi** A.A. 2022/2023

Diciotti Matteo  
7072181

Mannucci Agostino  
7084379

25 agosto 2023 - 1 settembre 2023

---

**Titolo** Relazione progetto di *Basi di Dati e Sistemi Informativi* A.A. 2022/2023

**Autori** Lista degli autori ordinata per numero di matricola

Matricola	Cognome	Nome	e-mail
7072181	Diciotti	Matteo	matteo.diciotti@stud.unifi.it
7084379	Mannucci	Agostino	agostino.mannucci@stud.unifi.it

**Obiettivo** Obiettivo del progetto è la realizzazione, per fini accademici, di una base di dati a partire dall'identificazione di una richiesta realmente proponibile, seguita dalla fase di progettazione concettuale cui succederà la fase di progettazione logica e quindi la realizzazione in MySQL.

**Introduzione al progetto** La realizzazione di una base di dati prevede che sia definita rigorosamente una richiesta, pilastro attorno cui si sviluppa il database passando da alcune importanti fasi: una fase di studio ed analisi, suddivisa tra la fase di *progettazione concettuale* e la fase di *progettazione logica*, ed una fase di sviluppo, ovvero la fase di realizzazione in linguaggio *MySQL*.

La strutturazione del progetto parte dalla scrittura di una richiesta completa e precisa, affinché fosse plausibile che la fase preliminare, comprendente il confronto col richiedente della base, fosse già avvenuta proficuamente avendo quindi predefinito le specifiche necessarie ed i dettagli focali per il cliente. La progettazione si è quindi sviluppata a partire da una richiesta che non dovesse essere modificata ulteriormente ma che potesse considerarsi definitiva e quindi immediatamente appetibile per le fasi di progettazione suddette, bacino nozionistico del corso. La stesura della relazione segue a grandi linee il template proposto su [sito del corso](#).

Per la realizzazione dei vincoli e per la scelta del formato dei codici sono stati ripresi modelli reali.

# Richiesta

Una società che organizza tornei di calcio a 5, a 7 e a 11 è presente nel territorio toscano con tornei attivi nella città di Firenze e nelle zone limitrofe.

Di questi tornei la società desidera mantenere informazioni relative alle fasi di gioco, alle squadre partecipanti, ai giocatori che compongono le formazioni e alle partite.

I tornei sono identificati tramite un codice e sono caratterizzati da un nome, da un'edizione, dalla tipologia di gioco (calcio a 5, a 7 o a 11) e dalla categoria di genere (maschile, femminile o mista).

Ogni torneo può essere suddiviso in più fasi di gioco (tipicamente da 1 a 3), le quali si distinguono dal nome della fase e sono caratterizzate dalla modalità di organizzazione degli scontri tra le squadre (a gironi o ad eliminazione diretta) e da un numero rappresentante la quantità di scontri tra due squadre nella fase. Le fasi raggruppano uno o più insiemi di squadre partecipanti differenti tra loro.

Gli insiemi di squadre, identificati dalla fase a cui appartengono e dal nome dell'insieme, raggruppano le formazioni partecipanti a quella fase e definiscono un gruppo di giornate di gioco, che rappresentano i turni di gioco delle squadre nella suddetta fase.

Le giornate di gioco calendarizzano le partite e sono identificate dall'insieme di squadre a cui afferiscono e dal numero della giornata. In particolare è rilevante per ogni partita conoscere la squadra di casa e la squadra ospite, la data di gioco, il campo, l'arbitro che dirige la gara e il punteggio finale.

Per ogni partita la società ha interesse a mantenere anche le statistiche sui giocatori che hanno effettuato azioni rilevanti in quella partita (gol fatti, assist effettuati, espulsioni, ammonizioni).

Le squadre si distinguono per il nome, la tipologia di calcio a cui giocano e il genere dei giocatori che la compongono. La società richiede inoltre che sia mantenuta l'informazione sul campo di casa (preferenza della squadra) il quale è contraddistinto dagli altri campi attraverso l'indirizzo. I campi possiedono comunque un nome proprio e un recapito.

Le persone tesserate alla società si suddividono in due tipologie: i giocatori e gli arbitri i quali condividono un numero di tessera univoco per ogni tesserato e sequenzialmente generato.

Di tutti gli iscritti si conoscono nome e cognome, data di nascita e genere, mentre per i giocatori si conosce, qualora partecipino a qualche torneo, anche le squadre con le quali gareggiano ed i relativi numeri.

## Progettazione Concettuale

### Analisi della richiesta

Si procede all'analisi della richiesta evidenziando:

- in *corsivo* i termini ambigui o imprecisi;
- sottolineati i termini con i quali si esprimono concetti diversi;
- in **grassetto** i termini concettualmente equivalenti, distinguendo successivamente le equivalenze.

Una società che organizza tornei di calcio a 5, a 7 e a 11 è presente nel territorio toscano con tornei attivi nella città di Firenze e nelle zone limitrofe.

Di questi tornei la società desidera mantenere informazioni relative alle fasi di gioco, alle **squadre partecipanti**, ai giocatori che compongono le **formazioni** e alle **partite**.

I tornei sono identificati tramite un codice e sono caratterizzati da un nome, da un'edizione, dalla tipologia di gioco (calcio a 5, a 7 o a 11) e dalla categoria di genere (maschile, femminile o mista).

Ogni torneo può essere suddiviso in più fasi di gioco (tipicamente da 1 a 3), le quali si distinguono dal nome della fase e sono caratterizzate dalla modalità di organizzazione degli **scontri** tra le squadre (a gironi o ad eliminazione diretta) e da due numeri interi positivi: uno rappresentante la quantità di scontri tra due squadre nella fase, l'altro rappresenta l'*indice* della fase nel torneo. Le fasi raggruppano uno o più insiemi di squadre partecipanti differenti tra loro.

Gli insiemi di squadre, identificati dalla fase a cui appartengono e dal nome dell'insieme, raggruppano le formazioni partecipanti a quella fase e definiscono un gruppo di *giornate* di gioco, che rappresentano i *turni* di gioco delle squadre nella suddetta fase.

Le giornate di gioco calendarizzano le partite e sono identificate dall'insieme di squadre a cui afferiscono e dal numero della giornata. In particolare è rilevante per ogni partita conoscere la squadra di casa e la squadra ospite, la data di gioco, il campo, l'arbitro che dirige la gara e il punteggio finale.

In particolare è rilevante per ogni partita conoscere la squadra di casa e la squadra ospite, la data di gioco, il campo, l'arbitro che dirige la **gara** e il *punteggio* finale.

Per ogni partita la società ha interesse a mantenere anche le statistiche sui giocatori che hanno effettuato azioni rilevanti in quella partita (gol fatti, assist effettuati, *espulsioni*, *ammonizioni*).

Le squadre si distinguono per il nome, la tipologia di calcio a cui giocano e il genere dei giocatori che la compongono. La società richiede inoltre che sia mantenuta l'informazione sul campo di casa (preferenza dalla squadra) il quale è contraddistinto dagli altri campi attraverso l'*indirizzo*. I campi possiedono comunque un nome proprio e un recapito. Le **persone tesserate** alla società si suddividono in due tipologie: i giocatori e gli arbitri i quali condividono un numero di tessera univoco per ogni **tesserato** sequenzialmente generato.

Di tutti gli **iscritti** si conoscono nome e cognome, data di nascita e genere, mentre per i giocatori si conosce, qualora **partecipino** a qualche fase di gioco, anche le squadre con la quali **gareggiano** ed i relativi numeri.

torneo

## Analisi dei termini

### *Termini ambigui o imprecisi*

- *edizione*: si riferisce al numero in cui è stato ripetuto lo stesso torneo e si indica con un numero intero positivo diverso da zero;
- *indice*: si riferisce all'indicazione dell'ordine con cui si svolgono le fasi di un torneo;
- *punteggio*: si riferisce al binomio: (numero gol segnati dalla squadra di casa, numero segnati dalla squadra ospite);
- *indirizzo*: si riferisce all'indicazione del comune, della via in cui è situato e del numero civico;
- *recapito*: si riferisce al numero di telefono dei gestori del campo sportivo.
- *espulsioni*: booleano rappresentante il "cartellino rosso" a cui è associato il numero di giornate di espulsioni prese da un giocatore nella partita
- *ammonizioni*: numero di cartellini gialli presi in una partita. Se #ammonizioni fosse 2 allora il numero di giornate di espulsione sarebbe almeno 1.

### Termini concettualmente plurimi

- *partecipante*: il termine è utilizzato per indicare sia una squadra che partecipa ad un torneo ("squadra partecipante" o "formazione partecipante") sia per un giocatore membro di una squadra che partecipa ad un torneo ("giocatore partecipante");
- nome: il termine è utilizzato per indicare il nome del torneo, il nome delle fasi di un torneo, il nome di un insieme di squadre, il nome di una squadra, il nome di un campo e il nome proprio di un tesserato.
- tipologia: il termine sta ad indicare la tipologia di calcio giocato che un torneo prevede ("tipologia del torneo"), la tipologia di calcio che una squadra gioca ("tipologia di calcio della squadra") ed infine la tipologia di tesseramento di una persona alla società, se si iscrive come giocatore o come arbitro ("tipologia di tesseramento");
- genere: il termine è utilizzato per indicare il genere dei tesserati ("genere dei tesserati"), il genere dei giocatori componenti di una squadra ("genere dei giocatori della squadra") e la composizione di genere dei giocatori ammessi a partecipare ad un torneo ("categoria di genere");
- numero: il termine è utilizzato per indicare il numero rappresentante la quantità di scontri tra due squadre in una fase ("numero di scontri"), l'indice della fase ("indice"), il numero che identifica una giornata di gioco ("numero della giornata"), il numero di tessera di un iscritto ("numero di tessera") e il numero di maglia di un giocatore all'interno di una squadra ("numero di maglia");
- data: il termine è utilizzato per indicare il giorno e l'orario in cui si disputa una partita ("data di gioco") e per indicare la data di nascita di un tesserato ("data di nascita");

- campo: il termine è utilizzato per indicare un campo gestito dalla società organizzatrice dei tornei ("campo"), il campo di casa selezionato da una squadra come preferenza ("campo di casa") ed il campo in cui si svolge una partita ("campo di gioco").

#### Termini concettualmente equivalenti

Termine	Descrizione	Sinonimi
squadra	Insieme di giocatori che unitamente possono partecipare ad un torneo	formazione
partita	Evento nel quale due squadre si affrontano e il cui punteggio determina un unico vincitore oppure un pareggio	scontro, gara
tesserato	Persona iscritta alla società che gestisce i tornei	persona tesserata, iscritto
giornata	Raggruppamento di partite	turno

## Testo tradotto

Si evidenziano i concetti principali e i termini ad essi correlati:

Una società che organizza tornei di calcio a 5, a 7 e a 11 è presente nel territorio toscano con tornei attivi nella città di Firenze e nelle zone limitrofe.

Di questi tornei la società desidera mantenere informazioni relative alle fasi di gioco, alle squadre partecipanti, ai giocatori che compongono le squadre e alle partite.

I tornei sono identificati tramite un codice e sono caratterizzati da un nome del torneo, dal numero dell'edizione, dalla tipologia di gioco (calcio a 5, a 7 o a 11) e dalla categoria di genere del torneo (maschile, femminile o mista).

Ogni torneo può essere suddiviso in più fasi di gioco (tipicamente da 1 a 3), le quali si distinguono per il nome della fase e sono caratterizzate dalla modalità di organizzazione delle partite tra le squadre (a gironi o ad eliminazione diretta), dal numero di scontri, rappresentante la quantità di partite tra due squadre nella fase e dall'indice, indicatore dell'ordine tra le fasi. Le fasi raggruppano uno o più insiemi di squadre partecipanti differenti tra loro.

Gli insiemi di squadre, identificati dalla fase a cui appartengono e dal nome dell'insieme, raggruppano le squadre partecipanti a quella fase e definiscono un gruppo di giornate di gioco, che rappresentano i turni di gioco delle squadre nella suddetta fase.

Le giornate di gioco calendarizzano le partite e sono identificate dall'insieme di squadre a cui afferiscono e dal numero della giornata. In particolare è rilevante per ogni partita conoscere la squadra di casa e la squadra ospite, la data di gioco, il campo di gioco, l'arbitro che dirige la partita e i gol segnati dalla squadra di casa e i gol segnati dalla squadra ospite.

Per ogni partita la società ha interesse a mantenere anche le statistiche sui giocatori che hanno effettuato azioni rilevanti in quella partita (gol fatti, assist effettuati, espulsioni, ammonizioni).

Le squadre si distinguono per il nome della squadra, la tipologia di calcio a cui gioca la squadra, il genere dei giocatori che la compongono e i colori della divisa della squadra (o colori sociali).

La società richiede inoltre che, per ogni squadra, sia mantenuta l'informazione sul campo di casa (preferenza della squadra) il quale è contraddistinto dagli altri campi attraverso l'indicazione del comune, della via e del numero civico in cui è situato. I campi possiedono comunque un nome del campo e un numero di telefono dei gestori del campo.

I tesserati alla società si suddividono in due tipologie di tesseramento: i giocatori e gli arbitri i quali condividono un numero di tessera univoco per ogni tesserato.

Di tutti i tesserati si conoscono il nome e cognome del tesserato, data di nascita e genere del tesserato, mentre per i giocatori si conosce, qualora siano membri di almeno una squadra che partecipa a qualche fase di gioco, anche le squadre con la quale partecipa ed i relativi numeri di maglia.

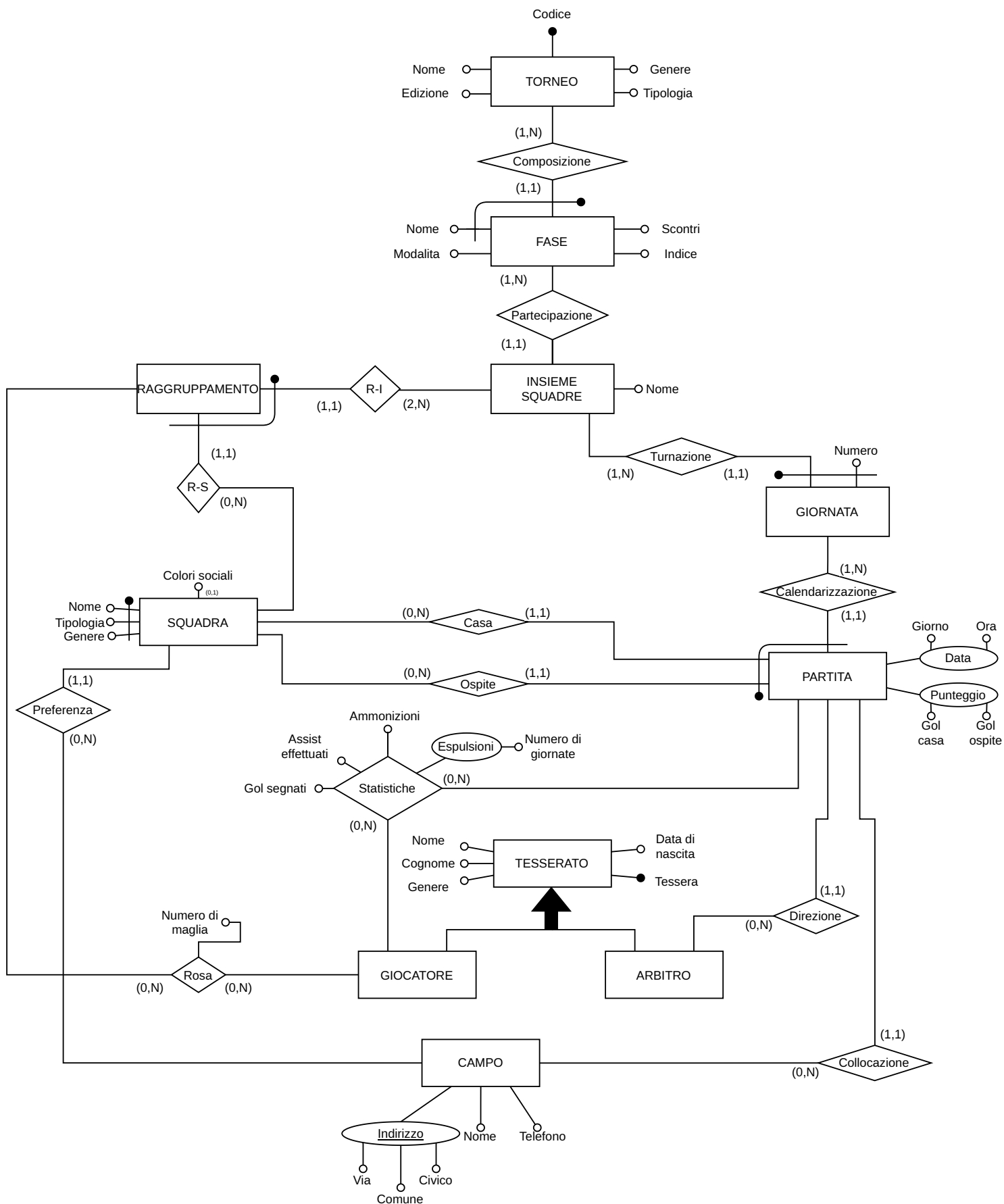
## Glossario dei termini

Nome	Descrizione	Termini relativi	Collegamenti
Torneo	Manifestazione sportiva organizzata dalla società gestrice	Codice del torneo, nome del torneo, numero dell'edizione, tipologia di gioco, categoria di genere del torneo	Fase
Fase	Parte del torneo che determina la modalità di organizzazione delle partite	Nome della fase, numero di scontri, modalità di organizzazione degli scontri, indice della fase	Insieme di squadre
Insieme di squadre	Gruppo di squadre che si affrontano nelle giornate della fase	Nome dell'insieme	Fase, Giornata
Giornata	Insieme di partite fra squadre dello stesso insieme di squadre	Numero della giornata	Insieme di squadre, Partita
Partita	Evento calcistico nel quale si affrontano due squadre	Data di gioco (giorno e orario), Punteggio (gol casa, gol ospite)	Giornata, Squadra casa, Squadra ospite, Arbitro, Campo
Squadra	Insieme di giocatori che unitamente possono partecipare ad un torneo	Nome, tipologia di calcio della squadra, genere dei giocatori della squadra, campo di casa, colori sociali	Insieme di squadre, Partita, Giocatore, Campo
Tesserato	Persona iscritta alla società che gestisce i tornei	Nome del tesserato, cognome, data di nascita, genere, numero di tessera	Squadra, Partita, Giocatore, Arbitro
Campo	Luogo in cui può essere disputata una partita	Nome del campo, indirizzo (comune, via, numero civico), recapito telefonico	Partita, Squadra

## Modello Entità-Relazione

Si mostra alla pagina seguente il modello entità-relazione derivante dalla richiesta e dallo studio svolto su di essa. Per la costruzione dello schema concettuale è stata adottata una strategia principalmente inside-out, ma complessivamente mista, partendo dalla raffinazione dello schema Torneo (strategia bottom-up), procedendo successivamente a macchia d'olio verso Fase, Insieme di squadre e Squadre, a cui sono stati associati successivamente gli attributi (top-down parziale), dopodiché sono stati definiti i concetti di Giornata e quello di Partita, il concetto di campo ed infine è stato creato, raffinato e collegato il concetto di tesserato come generalizzazione dei concetti Giocatore e Arbitro.

L'entità *Raggruppamento* era inizialmente un'associazione tra *Insieme di squadre* e *Squadra*, ma la creazione di insiemi di giocatori, ovvero *Rosa*, per ogni insieme di squadre richiedeva un'associazione tra le due entità suddette e *Giocatore*, creando quindi una associazione molto vicina concettualmente a *Raggruppamento*. Il motivo principale per cui è stato deciso di rendere *Raggruppamento* un'entità è il vincolo che sussiste tra giocatori, insiemi e squadre in *Rosa*, ovvero il fatto che un giocatore non può appartenere ad un insieme di squadre in qualche squadra se la stessa formazione non appartiene all'insieme delle squadre. Questo fatto ha reso necessaria l'introduzione dell'entità *Raggruppamento* permettendo all'associazione *Rosa* di mantenersi perennemente consistente all'interno della base di dati.



# Progettazione Logica

## Analisi delle prestazioni

### Tabella dei volumi

Vengono riportate nella tabella solo le associazioni aventi ciascuna cardinalità differente da (1,1).

Concetto	Tipo	Volume
Torneo	E	30
Fase	E	70
Insieme di squadre	E	250
Squadra	E	1500
Raggruppamento	E	2000
Giornata	E	3000
Partita	E	15000
Giocatore	E	15000
Arbitro	E	60
Tesserati	E	15060
Statistiche	R	75000
Rosa	R	20000
Campo	E	30

### Tabella delle operazioni

Dato il principio di Pareto (o regola 80:20) , si riportano le operazioni che richiedono più risorse al sistema.

- Operazione 1: Trova la classifica di un insieme di squadre ad una data giornata, eseguita circa 20000 volte a settimana;
- Operazione 2: Trova i giocatori di un insieme di squadre espulsi nella prossima giornata, eseguita circa 10000 volte a settimana.

Operazione	Tipo	Frequenza
Op. 1	B	20000 v. / sett.
Op. 2	B	10000 v. / sett.

### Tabelle degli accessi

#### Operazione 1

Supponiamo 10 squadre ad insieme e 2 scontri come caso peggiore.

Concetto	Costrutto	Accessi	Tipo
Insieme squadre	E	1	L
Raggruppamento	E	10	L
Giornata	E	18	L
Partita	E	90	L
Squadra	E	180	L

L'operazione risulta molto dispendiosa dal punto di vista delle richieste poste al database, per cui è stato deciso di introdurre una associazione tra *Squadra* e *Giornata* che mantiene il numero di vittorie, di pareggi e di sconfitte di ogni squadra al termine di ogni giornata di gioco. Questo introduce una ridondanza ma permette di operare il calcolo di classifica con soli 10 accessi a tale associazione.

#### Operazione 2

Supponiamo circa 10 squadre a insieme, 2 scontri e circa 15 giocatori a squadra, di cui 2 espulsi per squadra per ogni partita per circa una giornata.

Concetto	Costrutto	Accessi	Tipo
Insieme squadre	E	1	L
Rosa	R	150	L
Giornata	E	18	L
Partita	E	90	L
Statistiche	R	450	L
Giocatore	E	60	L

Ugualmente alla prima, anche questa operazione risulta molto costosa da punto di vista computazionale. Anche in questo caso quindi conviene introdurre una nuova associazione. L'associazione di più sempli-

ce implementazione per lo scopo risulta una associazione tra *Giocatore* e *Giornata* la quale mantiene l'associazione tra i giocatori espulsi in quali giornate.

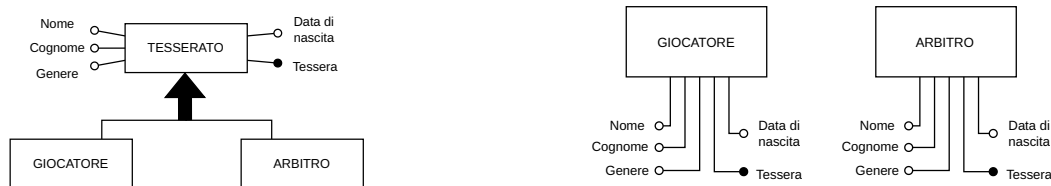
## Traduzione dello schema

### Analisi delle ridondanze

Come mostrato, sono state inserite alcune ridondanze al fine di agevolare le operazioni più costose, che quindi mantengono un ruolo centrale nel database. Escludendo le due ridondanze suddette, volutamente inserite nella base, non ve ne sono quindi di ulteriori.

### Eliminazione delle gerarchie

L'unica generalizzazione presente risulta essere composta da *Tesserato*, *Giocatore* e *Arbitro*. Nonostante le due entità figlie risultino identiche a livello di schema, come vedremo nello a pagina 11, l'aggregazione delle due in un'unica entità generale renderebbe più macchinoso e dispendioso il controllo necessario per distinguere a quale sotto-entità appartenga una chiave (ad esempio per poter assegnare *Giocatori* a squadre e non *Arbitri*) oltre a portare svantaggi in termini di spazio. In merito a questo è stato deciso di utilizzare due entità distinte ma identiche a livello di schema logico, cosicché i vincoli di integrità referenziale fossero sufficienti come controlli per le limitazioni suddette. Questa scelta comporta però un problema di implementazione legato all'inserimento dell'attributo "Tessera" il quale risulta condiviso tra le due entità. Si rimanda a questo proposito alla sotto-sezione *Trigger* del presente documento.

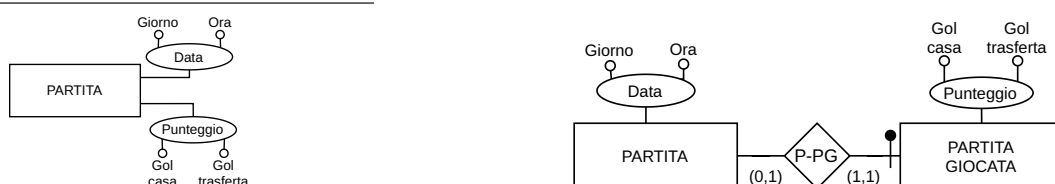


### Partizionamento/accorpamento di concetti

Un concetto che avrebbe potuto subire un partizionamento sarebbe potuto essere *Torneo* che, date le operazioni differenti tra i tornei terminati e quelli ancora in fase di esecuzione, tramite un partizionamento orizzontale, poteva suddividersi in *Torneo terminato* e *Torneo in esecuzione*, oppure tramite un partizionamento verticale in *Torneo* ed *Edizione*. L'introduzione di una nuova entità, in questo caso, è stato però ritenuto che in nessun caso agevolasse una qualunque operazione sufficientemente da giustificare questa introduzione.

Un ragionamento simile, ma con esito differente, è stato fatto per *Partita*. È stato operato un partizionamento verticale per distinguere le *Partite*, rappresentanti l'evento partita, e le *Partite giocate*, complete di punteggio. Questo permette di operare l'aggiornamento della classifica nel momento della definizione di una partita giocata, mantenendo le ridondanze presenti in classifica costantemente aggiornate.

Partita → Partita, Partita\_giocata





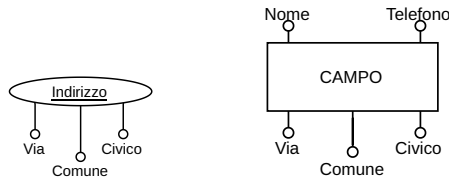
### Attributi composti partita



### Attributi composti partita giocata

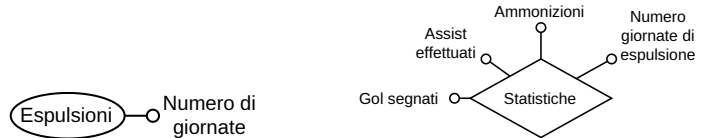


### Attributi composti campo



### Attributi composti statistiche

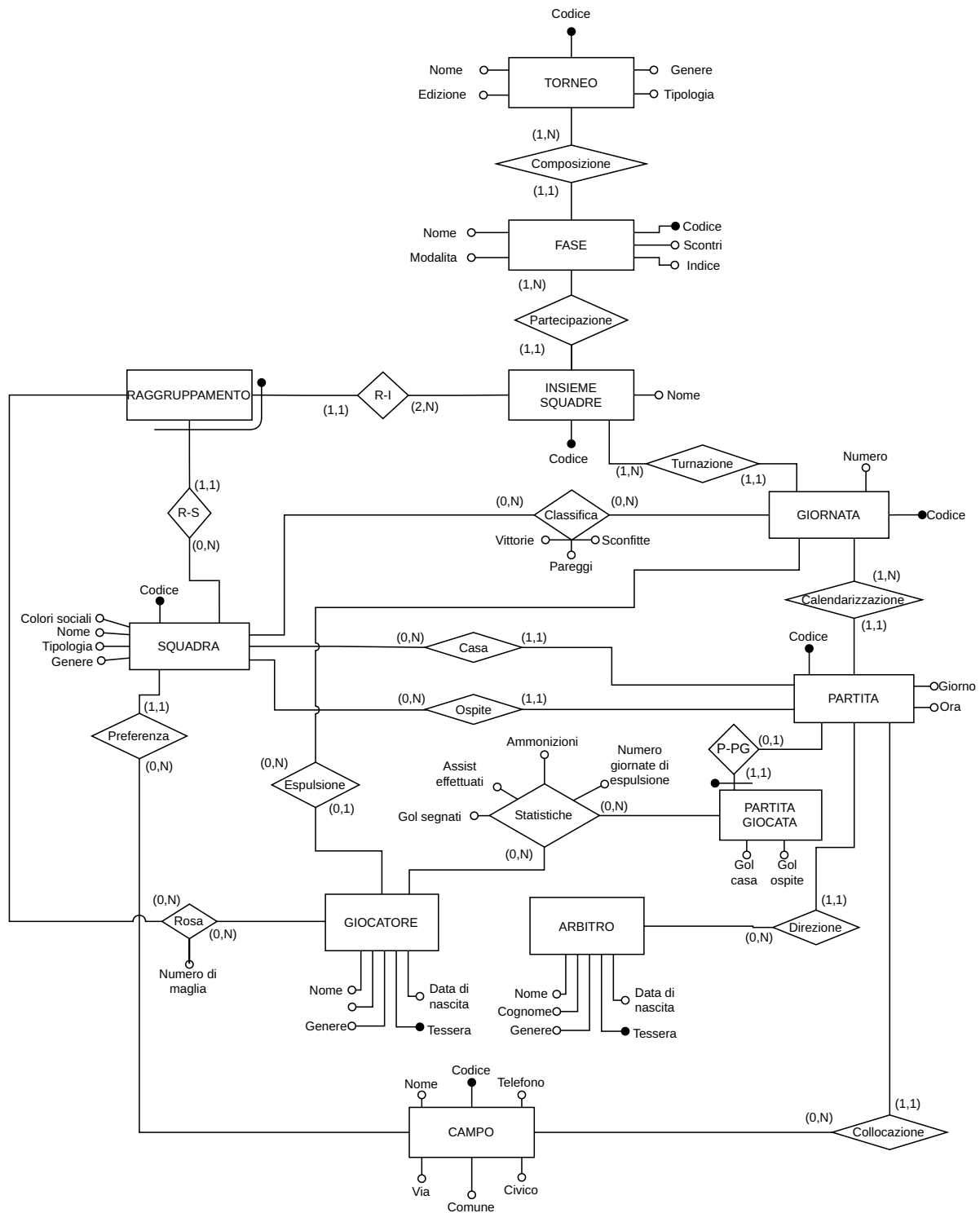
L'attributo comprende un booleano e un intero positivo in caso di "vero", che può essere sostituito con un numero naturale, (compreso lo zero, per rappresentare il caso "falso").



## Scelta degli identificatori principali

Data la dipendenza gerarchica dello schema, ovvero il fatto che varie entità risultano identificate parzialmente da un'entità a sua volta dipendente da un'ulteriore entità, risulta necessario introdurre dei codici per agevolare l'identificazione delle istanze di relazione. In particolare sono stati introdotti dei codici identificativi in *Fase*, *Giornata* e *Insieme squadre*, oltre che per *Squadra*, *Partita* e *Campo* per le cui entità la scelta di introdurre un codice è stata presa poiché le chiavi primarie risultavano composte da vari attributi. Invece, data la conformazione di *Raggruppamento*, è sembrato naturale lasciare come identificatore principale le due chiavi esterne che definiscono il raggruppamento.

## Modello ER ristrutturato



## Traduzione verso il modello relazionale

Si mostra nel seguito il risultato finale della traduzione, suddiviso in *Schema logico*, *Vincoli di integrità referenziale* e *Altri vincoli*. Tra i vincoli ne sono stati riportati alcuni assenti nella richiesta che tuttavia sono generalmente imposti nei contesti reali affini: è stato operato ciò affinché non risultasse contro-intuitivo l'utilizzo del database.

## Schema logico

**Torneo**(codice, nome, edizione, genere, tipologia)  
**Fase**(codice, torneo, nome, modalita, scontri, indice)  
**Insieme\_squadre**(codice, fase, nome)  
**Giornata**(codice, insieme\_squadre, numero)  
**Squadra**(codice, nome, tipologia, genere, colori, campo)  
**Raggruppamento**(insieme\_squadre, squadra)  
**Classifica**(giornata, squadra, vittorie, pareggi, sconfitte);  
**Partita**(codice, giornata, giorno, ora, squadra\_casa, squadra\_ospite, arbitro, campo)  
**Partita\_giocata**(partita, gol\_casa, gol\_ospite)  
**Giocatore**(tessera, nome, cognome, data, genere)  
**Rosa**(insieme\_squadre, squadra, giocatore, numero\_maglia)  
**Statistiche**(partita\_giocata, giocatore, gol, assist, ammonizioni, espulsione\_giornate)  
**Espulsione**(giornata, giocatore);  
**Arbitro**(tessera, nome, cognome, data, genere)  
**Campo**(codice, nome, telefono, comune, via, civico)

## Vincoli di integrità referenziale

Fase.torneo → Torneo.codice  
Insieme\_squadre.fase → Fase.codice  
Giornata.insieme\_squadre → Insieme\_squadre.codice  
Squadra.campo → Campo.codice  
Raggruppamento.insieme\_squadre →  
→ Insieme\_squadre.codice  
Raggruppamento.squadra → Squadra.codice  
Classifica.giornata → Giornata.codice  
Classifica.squadra → Squadra.codice  
Partita.giornata → Giornata.codice  
Partita.squadra\_casa → Squadra.codice

Partita.squadra\_ospite → Squadra.codice  
Partita.arbitro → Arbitro.codice  
Partita.campo → Campo.codice  
Rosa.insieme\_squadre →  
→ Raggruppamento.insieme\_squadre  
Rosa.squadra → Raggruppamento.squadra  
Rosa.giocatore → Giocatore.codice  
Statistiche.partita\_giocata → Partita\_giocata.partita  
Statistiche.giocatore → Giocatore.codice  
Espulsione.giornata → Giornata.codice  
Espulsione.giocatore → Giocatore.codice

## Altri vincoli

### Vincoli di unicità

Torneo(nome, genere, tipologia, edizione)  
Fase(torneo, nome)  
Fase(torneo, indice)  
Insieme\_squadre(fase, nome)  
Campo(comune, via, civico)  
Campo(nome, comune)  
Giornata(insieme\_squadre, numero)  
Giocatore(nome, cognome, data)  
Rosa(insieme\_squadre, squadra, numero\_maglia)  
Arbitro(nome, cognome, data)  
Partita(squadra\_casa, squadra\_ospite, giornata)

### Vincoli inter-relazionali

*Squadra*.genere uguale a *Torneo*.genere;  
*Giocatore*.genere corrispondente a *Squadra*.genere;  
La stessa *Squadra* può partecipare al più una volta ad una *Fase*;  
Tessera è un unico indicatore condiviso e sequenziale tra *Arbitri* e *Giocatori*;  
Somma dei gol delle *Statistiche* dei giocatori di una *Partita* corrispondente al punteggio della *Partita*;  
In un *Insieme di squadre* possono partecipare solo *Squadre* con giocatori differenti;  
Per ogni giornata di espulsione presa da un giocatore è presente un'istanza concorde in *Espulsione*;  
Per ogni *Giornata* giocata risulta presente in *Classifica* un'istanza riportante i dati della *Squadra* nella *Fase*

### Vincoli intra-relazionali e di dominio

*Squadra*.casa diversa da *Squadra*.ospite per ogni *Partita*;  
*Statistiche*.ammonizioni compreso in {0, 1, 2};  
Se *Statistiche*.ammonizioni = 2 allora *Statistiche*.espulsione\_giornate > 0;  
*Rosa*.numero\_maglia compreso in {0, ..., 99};

# Implementazione MySQL

## Premessa

Prima di dar spazio alla trattazione del codice è necessario anteporre una premessa: nello script sono stati inseriti volutamente alcuni commenti non necessari, errati secondo i canoni della buona programmazione. Le motivazioni che hanno portato alla decisione di ignorare i suddetti precetti sono principalmente due: lo script è il più utile strumento per illustrare il codice, senza il supporto di questo la comprensione delle spiegazioni date in un testo sconnesso risulterebbe più spinosa; la seconda motivazione deriva dal contesto in cui si inserisce l'elaborato, ovvero la verifica delle competenze. Le norme di buona programmazione occorrono principalmente nei contesti in cui il codice debba essere plausibilmente modificato in un futuro più o meno remoto da, possibilmente, altre persone. Nel caso in esame invece il codice non ha interesse nell'essere sviluppato ulteriormente dato il motivo per il quale è stato prodotto.

La compresenza dei due fatti esposti è stato motivo sufficiente per spingere gli autori ad inserire nello script vari commenti legati al contesto specifico ma superflui, per non dire nocivi, in contesti differenti.

## Creazione e popolamento delle tabelle

Data l'assenza di vincoli referenziali circolari, la creazione delle tabelle non ha avuto criticità in termini di implementazione. Il popolamento delle tabelle è stato eseguito tramite comandi INSERT e LOAD: per mostrare la procedura ma per non appesantire eccessivamente lo script è stato eseguito un unico inserimento tramite il comando INSERT, mentre tutti gli altri popolamenti sono stati eseguiti tramite il comando LOAD, dei quali uno da file non standard .in e gli altri da file formattati CSV<sup>1</sup>. Inoltre è stata creata una procedura per l'inserimento di istanze in *Statistiche* al fine di mostrare questo genere di procedure in MySQL, ma non ne sono state inserite di ulteriori per non inquinare eccessivamente lo script con procedure troppo simili tra loro, dato il contesto accademico.

Inoltre, data l'elevata presenza di codici all'interno del database è stato deciso che, affinché fossero distinguibili ad occhio umano, di inserire ad inizio stringa un carattere identificativo della relazione di appartenenza ed un separatore dagli indici (esempio: 'G-xx' per identificare l'istanza *Giornata* indicizzata XX, numero naturale).

Infine occorre dare una spiegazione riguardo alle scelte che definiscono la portata della base di dati, ovvero alla scelta dei tipi primitivi utilizzati o della lunghezza massima delle stringhe che definiscono i codici: data la presunta modesta dimensione della società gestrice dei Tornei (inserita "solo" su Firenze e sulle zone limitrofe) e soprattutto dato il contesto accademico, sono state preferite dimensioni massime esigue per i dati. Un eventuale ampliamento del database è permesso aumentando le dimensioni (in termini di byte) dei tipi interi e incrementando i limiti di caratteri di alcuni attributi definiti tramite VARCHAR (con particolare riguardo per i codici).

## Viste

Immediatamente dopo la creazione delle tabelle sono state implementate alcune viste ausiliare per il controllo di alcuni vincoli nei trigger. In particolare sono state create le viste per conoscere tutti i tesseramenti della società (per agevolare la generazione di nuove tessere), i tornei terminati nel database, le iscrizioni delle squadre ai vari insiemi di squadre (e quindi ai tornei), la suddivisione delle partite tra le giornate e la suddivisione delle partite giocate. Le viste generate per scopi ausiliari sono quindi in totale cinque.

Sono invece state introdotte viste per scopi non ausiliari nella sezione dello script dedicata agli utenti del database per un totale di altre cinque viste. In particolare di queste viene mostrato un esempio di funzionamento esplicito (commentato nello script alla sezione dedicata) e si suddividono in:

---

<sup>1</sup>Al fine di procedere al caricamento dei file per il popolamento devono essere inseriti nello script i percorsi assoluti dei file .csv e .in, ammenoché non venga lanciato il comando di apertura di mysql-workbench dalla cartella contenente lo script stesso. Lo script sarà così impostato perché si possano caricare i file tramite percorso relativo da quella directory. Questa procedura è stata testata per implementazioni di MySQL e MySQL-Workbench per sistemi Linux Debian-based. Per ulteriori informazioni consultare il manuale MySQL alla sezione Input File Location: [MySQL Manual LOAD DATA](#)

1. `classifica_punti_giornata`: mostra la classifica punti al termine delle singole giornate, dove i punti vengono calcolati attraverso lo schema di calcolo: 1 vittoria +3, 1 pareggio +1.;
2. `classifica_punti_aggiornata`: simile alla vista precedente ma specifica per il calcolo della classifica all'ultima giornata completa disponibile;
3. `prossime_partite`: mostra le prossime partite che avranno luogo dei tornei non terminati;
4. `squalificati_prossima_partita`: mostra la lista degli squalificati alla prossima partita (vedi vista precedente);
5. `somma_statistiche_giocatore`: mostra la somma di tutte le statistiche di un giocatore. Scopo preposto: calcolo delle classifiche dei giocatori con più gol di un dato torneo, giocatori con più assist, eccetera.

## Trigger

A causa dell'impossibilità di utilizzare meccanismi di SQL dinamico all'interno dei trigger, è stato deciso di implementare vari trigger reciprocamente molto simili tra loro per eseguire sulle tabelle controlli concettualmente e logicamente affini, come ad esempio il controllo della formattazione del codice. L'utilizzo di codici decimali avrebbe reso la gestione degli stessi immediata, affidando i doveri di generazione automatica dei codici al DBMS (AUTOINCREMENT), ma questo avrebbe reso difficile la comprensione delle tabelle per un essere umano, aumentando la probabilità di confusione durante le letture.

Complessivamente sono stati creati circa due trigger per tabella, uno per il controllo di vincoli durante gli inserimenti e l'altro per i controlli durante l'aggiornamento. La decisione di accorpare tutti i controlli in un unico trigger e di non separarli in vari trigger singoli che eseguissero controlli specifici o affini e che si susseguissero tramite i costrutti `FOLLOWS` e `PRECEDES` di MySQL è stata presa per coadiuvare la comprensione e la valutazione dal database, eseguendo così un'unica procedura per ogni evento d'innescio. Questa decisione ha delle conseguenze stilistiche dovute alla struttura rigida del linguaggio: le dichiarazioni di tutte le variabili utili nel trigger sono eseguite all'inizio, quindi potenzialmente separate dai controlli in cui queste variabili vengono utilizzate.

La struttura dei trigger è generalmente simile: in caso di condizioni non rispettate questi o impediscono le azioni dell'utente tramite la segnalazione dello stato '45000', oppure correggono l'inconsistenza portando le istanze ad uno stato accettabile. Generalmente negli update è stato preferito impedire un'azione illegale piuttosto che gestirla.

Per la creazione dei trigger sono state sfruttate alcune *viste* ausiliare, inserite precedentemente alla sezione dei trigger nello script, oltre che ai meccanismi dei *cursori* e degli *handler* oltre che, come già accennato, a quello delle *signal* e quello delle *query* per la ricerca nel database.

I controlli principali nei trigger sono:

- controllo di codice e tessera, generati in caso di formato sbagliato o valore nullo. In particolare per tessera si utilizza la vista *tesseramenti* per accorpare *Giocatori* e *Arbitri* e generare tessere sequenziali nell'unione dei due insiemi. Per gli inserimenti in queste tabelle è suggerito lasciare al sistema la generazione, limitando la creazione di "buchi" dovuti al fatto che la scelta della tessera successiva è fatta scegliendo il valore dell'ultima tessera assegnata incrementato di uno. Per i codici invece si assegna il primo indice disponibile;
- controllo del genere corrispondente tra *Torneo*, *Squadra* e *Giocatori*. Attraverso semplici query vengono eseguiti i confronti necessari;
- controllo della partecipazione singola di squadre e giocatori a fase ed insieme;
- controlli su i domini.

Oltre ai controlli necessari, i trigger eseguono anche alcune azioni per mantenere il database sempre coerente, come ad esempio gli aggiornamenti di *Classifica* e di *Espulsioni*.

## Procedure e funzioni

È stata implementata un'unica procedura per scopi meramente ausiliari che esegue la ricerca e la restituzione dei valori propri di classifica attraverso alcuni parametri. Le procedure orientate all'utilizzo degli utenti invece sono `inserimento_statistiche`, `raggruppamento_fase_successiva` e `riempimento_giornate`, per

un totale di tre procedure. La prima procedura dell'elenco è pensata per coadiuvare l'utente negli inserimenti di *Statistiche*, eseguendo i controlli necessari e guidandone lo svolgimento. La seconda procedura esegue gli inserimenti automatici in *Raggruppamento* partendo dalla classifica finale della fase a gironi precedente. La terza invece inserisce istanze in *Giornata* fino a raggiungere il numero minimo di giornate di cui un insieme di squadre ha sicuramente necessità per completarsi.

L'unica funzione destinata all'utilizzo dagli utenti mostra se, in una data partita, una squadra gioca nel campo di casa oppure no ed è denominata `is_campo_casa`.

## Interrogazioni

Le sezioni dello script successive successive a quella predisposta per la creazione di procedure per gli utenti, ovvero la sezione di esempi di utilizzo e la sezione dedicata alle interrogazioni, eseguono, commentando, varie query per alcune ricerche specifiche. In particolare si eseguono le richieste di:

### ESEMPI

- Classifica al termine della seconda giornata;
- Prossime partite da giocare dell'insieme 'I-2';
- Giocatori squalificati nella prossima partita nel torneo 'T-0';
- Classifica dei goleador e dei giocatori più ammoniti nel torneo 'T-0'. INTERROGAZIONI

### INTERROGAZIONI

- Trovare nome e cognome di tutti i giocatori aventi numero 10 nella squadra con la quale partecipano ad una fase e che hanno segnato un gol in almeno 2 partite distinte nella fase;
- Trovare tessera, nome e cognome dei giocatori non assegnati a nessuna squadra che attualmente partecipa ad un torneo in corso;
- Trovare tutte le squadre che hanno o potrebbero avere come colore sociale il bianco ma che non abbiano il colore blu.

---

## Indice

	1
<b>Richiesta</b>	<b>2</b>
<b>Progettazione Concettuale</b>	<b>2</b>
<b>Analisi della richiesta</b>	<b>2</b>
Analisi dei termini . . . . .	3
<b>Testo tradotto</b>	<b>4</b>
Glossario dei termini . . . . .	5
<b>Modello Entità-Relazione</b>	<b>5</b>
<b>Progettazione Logica</b>	<b>6</b>
<b>Analisi delle prestazioni</b>	<b>7</b>
Tabella dei volumi . . . . .	7
Tabella delle operazioni . . . . .	7
Tabelle degli accessi . . . . .	7
<b>Traduzione dello schema</b>	<b>8</b>
Analisi delle ridondanze . . . . .	8
Eliminazione delle gerarchie . . . . .	8
Partizionamento/accorpamento di concetti . . . . .	8
Scelta degli identificatori principali . . . . .	9
<b>Modello ER ristrutturato</b>	<b>10</b>
<b>Traduzione verso il modello relazionale</b>	<b>10</b>
Schema logico . . . . .	11
Vincoli di integrità referenziale . . . . .	11
Altri vincoli . . . . .	11
<b>Implementazione MySQL</b>	<b>11</b>
Creazione e popolamento delle tabelle . . . . .	12
Viste . . . . .	12
Trigger . . . . .	13
Procedure e funzioni . . . . .	13
Interrogazioni . . . . .	14