

Università degli Studi di Napoli Federico II



Corso di Laurea Triennale in Informatica

Anno Accademico 2023/2024

Insegnamenti:

Basi di Dati e Sistemi Informativi I

Professore

Silvio Barra

Object Orientation

Professore

Porfirio Tramontana

Traccia 3: Sistema Informativo Dei Calciatori

Miranda Pasquale

Matricola: N86004643

Mennillo Vincenzo

Matricola: N86004494

27 gennaio 2024

Indice

1	Topic 3	3
1.1	Description Topic	3
2	Database	3
2.1	Analisi dei Requisiti	3
2.2	Progettazione Concettuale	3
2.2.1	Class Diagram non ristrutturato	4
2.2.2	Ristrutturazione del Class Diagram	5
2.2.3	Class Diagram ristrutturato	10
2.2.4	Dizionario	11

1 Topic 3

1.1 Description Topic

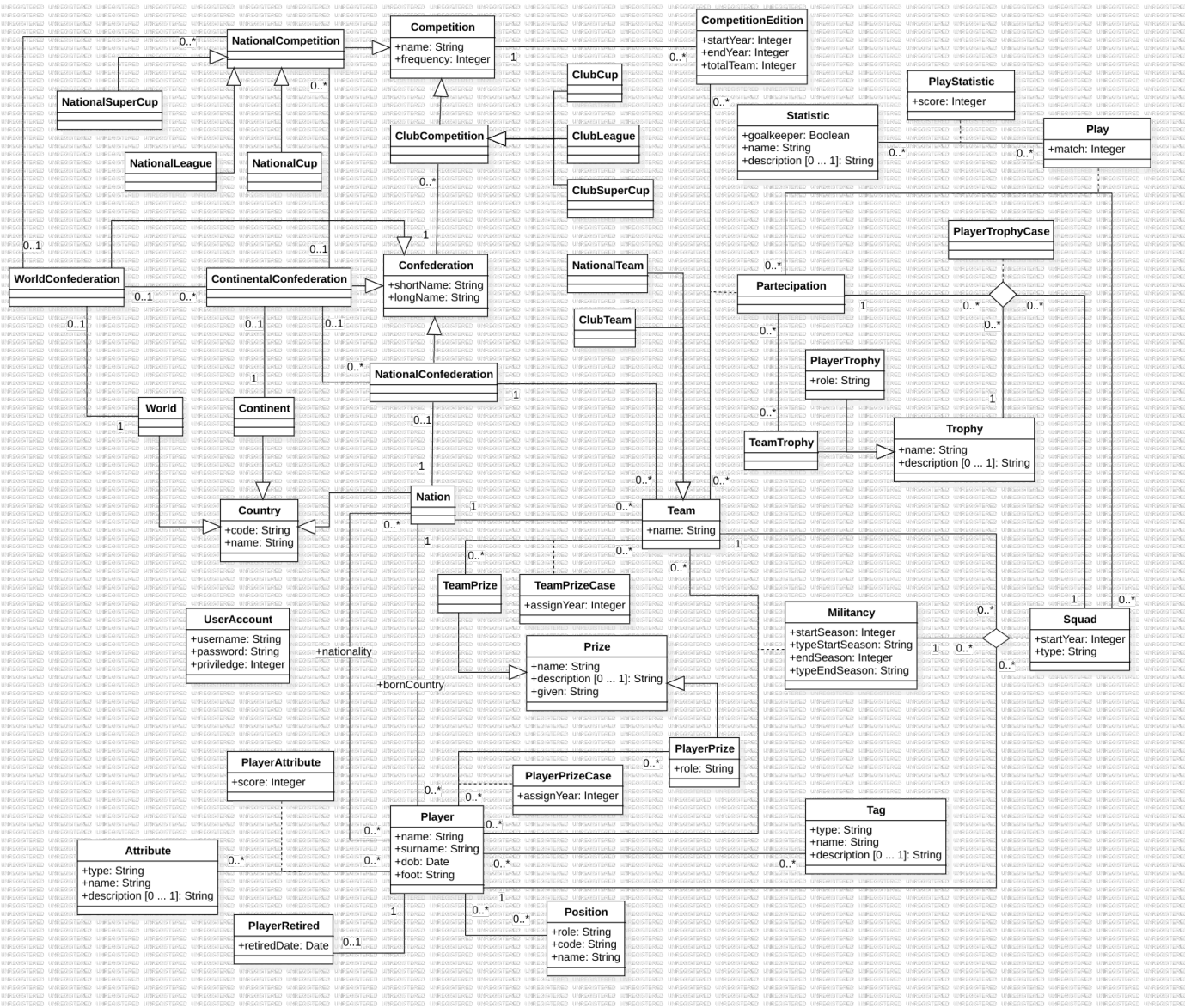
Si sviluppi un sistema informativo, composto da una base di dati relazionale e da un applicativo Java dotato di GUI (Swing o JavaFX), per la gestione di calciatori di tutto il mondo. Ogni calciatore è caratterizzato da nome, cognome, data di nascita, piede (sinistro, destro o ambidestro), uno o più ruoli di gioco (portiere, difensore, centrocampista, attaccante) e una serie di feature caratteristiche (ad esempio colpo di testa, tackle, rovesciata, etc.). Il giocatore ha una carriera durante la quale può militare in diverse squadre di calcio. La militanza in una squadra è caratterizzata da uno o più periodi di tempo nei quali il giocatore era in quella squadra. Ogni periodo di tempo ha una data di inizio ed una data di fine. Durante la militanza del giocatore nella squadra si tiene conto del numero di partite giocate, del numero di goal segnati e del numero di goal subiti (applicabile solo ai giocatori di ruolo portiere). Il giocatore può inoltre vincere dei trofei, individuali o di squadra. Il giocatore può avere anche una data di ritiro a seguito della quale decide di non giocare più. Le squadre di calcio sono specificate dal loro nome e nazionalità. L'amministratore del sistema si identifica con una login ed una password e ha il diritto di inserire nuovi giocatori nella base di dati, modificarne i dati, aggiungere ulteriori informazioni oppure eliminare un giocatore. L'utente generico può vedere l'elenco dei giocatori e le loro caratteristiche e può richiedere diverse ricerche, ad esempio filtrando i giocatori per nome, per ruolo, per piede, per numero di goal segnati, per numero di goal subiti, per età, per squadre di appartenenza. Per gruppi da 3 persone: I giocatori dopo la fine della carriera possono diventare allenatori o dirigenti. Il sistema continua a mantenere una parte delle informazioni (squadra, numero di partite e trofei vinti) anche per allenatori e dirigenti. Inoltre, può accedere al sistema anche un terzo tipo di utente, consistente nel Giocatore stesso. Egli ha una sua login e password e può modificare unicamente i dati relativi a sé stesso.

2 Database

2.1 Analisi dei Requisiti

2.2 Progettazione Concettuale

2.2.1 Class Diagram non ristrutturato



2.2.2 Ristrutturazione del Class Diagram

Analisi delle Ridondanze

Possiamo constatare che non ci sono attributi ridondanti.

Notiamo però che vista l'alta frequenza di certe operazioni, che dovranno essere compiute sul database, vi è un impellente necessità di introdurre delle ridondanze.

Prima operazione su tutte, è la visualizzazione di un resoconto delle statistiche di un calciatore rispetto alla sua militanza (operazione richiesta anche dalla traccia). Quest'ultimo infatti si potrebbe ottenere facendo una somma delle statistiche riferite ai Play che sono associati allo specifico calciatore, alla squadra della militanza che si sta tenendo in considerazione e che siano avvenuti nell'arco di tempo in cui il calciatore in questione militava per quella certa squadra.

Per evitare che ad ogni richiesta di questa operazione si debba rifare il calcolo, soffermandoci nuovamente sul fatto che si suppone che questa operazione venga effettuata un numero molto elevato di volte, si vede la necessità di aggiungere in **Militanza** un attributo **match** che conservi il numero di presenze in campo di un calciatore durante quella specifica militanza.

Inoltre, vi è necessario aggiungere una classe **MilitancyStatistic** che sarà associata a **Statistic** e a **Militancy**, e che conterrà come attributo, **score** che ha lo scopo di conservare il valore di una certa statistica di un calciatore durante una specifica militanza.

Un'altra operazione che si suppone sia ricorrente, è il resoconto del numero di presenze in campo di un calciatore in una determinata posizione. Questo calcolo si potrebbe ottenere dalla somma dell'attributo **match** delle tuple in Play associate al calciatore in quella determinata posizione.

Per velocizzare l'operazione si vede necessario, dunque, introdurre nella classe **PlayPosition** un attributo **match** che conservi il numero di presenze in campo di un calciatore in una specifica posizione.

L'ultima operazione, ovvero la visualizzazione delle nazionalità di un giocatore, si potrebbe ottenere navigando entrambe le associazioni che **Player** ha con **Country**: **bornCountry** e **Nationality**.

Visto che il nostro database ha come scopo primario l'essere un sistema informativo per i calciatori, si vede necessario aggiungere alla relazione **Nationality** anche la nazione di nascita del calciatore così da velocizzare l'operazione, in quanto così facendo basta navigare soltanto l'associazione **Nationality**.

Eliminazione delle Generalizzazioni

Nonostante le numerose generalizzazioni presenti nel Class Diagram, tutte hanno un qualcosa in comune, ovvero sono generalizzazioni disgiunte totali, e in particolare le classi figlie non hanno attributi propri, ma al massimo associazioni con altre classi.

A seguito di ciò dunque, come metodo per eliminarle, si decise di adottare l'accorpamento delle figlie della generalizzazione nel padre, aggiungendo dove necessario un attributo **type** che distinguesse le varie istanze della classe padre, e facendo le dovute correzioni ad eventuali associazioni delle classi figlie con le altre classi.

Di seguito viene mostrato un elenco delle generalizzazioni e di come sono state ristrutturare:

Country

Alla classe **Country** è stato aggiunto un attributo **type** di tipo enum chiamato **enCountry**. Il tipo **enCountry** può assumere tre valori:

- Nation;
- Continental;
- World.

Per quanto riguarda le associazioni:

Le associazioni delle classi figlie di **Country** con le classi figlie di **Confederation** sono state accorpate, risultandone in un'unica associazione tra **Country** e **Confederation** con molteplicità invariata.

Per quanto riguarda le associazioni della classe figlia **Nation** con **Player** e **Team**, esse sono state ricollegate alla classe padre **Country** mantenendo la molteplicità invariata.

Confederation

Alla classe **Confederation** non è stato aggiunto un attributo **type** poiché si è sfruttato l'associazione 1 a 1 che le classi figlie di **Confederation** avevano con le classi figlie di **Country**. Dunque per comprendere a quale classe figlia una tupla di **Confederation** corrisponde, bisogna basarsi sull'attributo **type** della classe **Country**.

Per quanto riguarda le associazioni:

Le due associazioni che intercorrono tra le classi figlie di **Confederation**, ovvero l'associazione tra **NationalConfederation** e **ContinentalConfederation**, e tra **ContinentalConfederation** e **WorldConfederation**, sono state accorpate, risultandone in un'unica associazione tra la classe **Confederation** e sé stessa.

Le due associazioni delle classi figlie **ContinentalConfederation** e **WorldConfederation** con **NationalCompetition**, sono state accorpate, risultandone in un'unica associazione tra la classe **Confederation** e la classe **Competition**, classe padre di **NationalCompetition**.

Competition

Alla classe **Competition** sono stati aggiunti due attributi: **type** di tipo enum chiamato **enCompetition** e **teamType** di tipo enum chiamato **enTeam**. Questo perché anche le classi figlie di **Competition** erano una generalizzazione a loro volta.

Nota che ci si è ridotti a solo due tipi aggiunti, perché le classi **NationalCompetition** e **ClubCompetition** hanno le stesse classi figlie.

Il tipo enum **enTeam** può assumere i seguenti valori:

- National;
- Club.

Il tipo enum **enCompetition** può assumere i seguenti valori:

- Cup;
- League;
- Super Cup.

Per quanto riguarda le associazioni:

L'associazione tra **ClubCompetition** e **Confederation**, è stata ricollegata alla classe padre **Competition** mantenendone però la molteplicità invariata.

Le associazioni tra **NationalCompetition** e **NationalCompetitionEdition** e tra **ClubCompetition** e **ClubCompetitionEdition** sono state accorpate, risultandone in un'unica associazione (con molteplicità invariata) tra **Competition** e **CompetitionEdition**, la classe padre di **NationalCompetitionEdition** e **ClubCompetitionEdition**.

CompetitionEdition

Alla classe **CompetitionEdition** non è stato aggiunto un attributo **type** poiché si è sfruttata l'associazione che vi era tra le sue classi figlie e le classi figlie di **Competition**. Dunque per comprendere a quale classe figlia una tupla di **CompetitionEdition** corrisponde, bisogna basarsi sull'attributo **teamType** della classe **Competition**.

Per quando riguarda le associazioni:

Le associazioni tra **NationalCompetitionEdition** e **NationalPartecipation** e tra **ClubCompetitionEdition** e **ClubPartecipation**, sono state accorpate, risultandone in un'unica associazione (con molteplicità invariata) tra la classe **CompetitionEdition** e **Partecipation**, la classe padre di **NationalPartecipation** e **ClubPartecipation**.

Team

Alla classe **Team** è stato aggiunto un attributo **type** di tipo enum chiamato **enTeam** (lo stesso di **Competition**).

Per quanto riguarda le associazioni:

Le associazioni tra **National** e **NationalPartecipation** e tra **Club** e **ClubPartecipation**, sono state accorpate, risultandone in un'unica associazione tra la classe **Team** e **Partecipation** la classe padre di **NationalPartecipation** e **ClubPartecipation**.

Partecipation

Alla classe **Partecipation** non è stato aggiunto un attributo **type** poiché si è sfruttata l'associazione che vi era tra le sue classi figlie e le classi figlie di **Team**. Dunque per comprendere a quale classe figlia una tupla di **Partecipation** corrisponde, bisogna basarsi sull'attributo **type** della classe **Team**.

Per quanto riguarda le associazioni: Sono già state gestite precedentemente.

Trophy

Alla classe **Trophy** è stato aggiunto un attributo **type** di tipo enum chiamato **enAward**.

Il tipo **enAward** assume i seguenti valori:

- Player
- Team

Per quanto riguarda le associazioni:

L'associazione tra **TeamTrophy** e **Partecipation** è stata ricollegata alla classe padre **Trophy** mantenendo la molteplicità invariata.

Prize

Alla classe **Prize** è stato aggiunto un attributo **type** di tipo enum chiamato **enAward** (lo stesso tipo di Trophy).

Per quanto riguarda le associazioni:

L'associazione tra **TeamPrize** e **TeamPrizeCase** è stata ricollegata alla classe padre **Prize** mantenendo la molteplicità invariata.

L'associazione tra **PlayerPrize** e **PlayerPrizeCase** è stata ricollegata alla classe padre **Prize** mantenendo la molteplicità invariata.

Eliminazioni Attributi Multivalore

Vi sono tre attributi multivalore presenti nel Class Diagram:

- L'attributo **role** della classe **Prize**;
- L'attributo **role** della classe **Trophy**;
- L'attributo **role** della classe **Statistic**.

Questo attributo assume la stessa funzione in tutte e tre le Classi, ovvero quello di descrivere quella determinata classe a quali ruoli può essere associata.

Data l'importanza assunta da quest'attributo, si è deciso di creare un tipo enum che contiene tutte le combinazioni di ruoli possibili (ci possono essere massimo 4 ruoli).

Eliminazione Attributi Strutturati

Nel Class Diagram non sono presenti Attributi Strutturati.

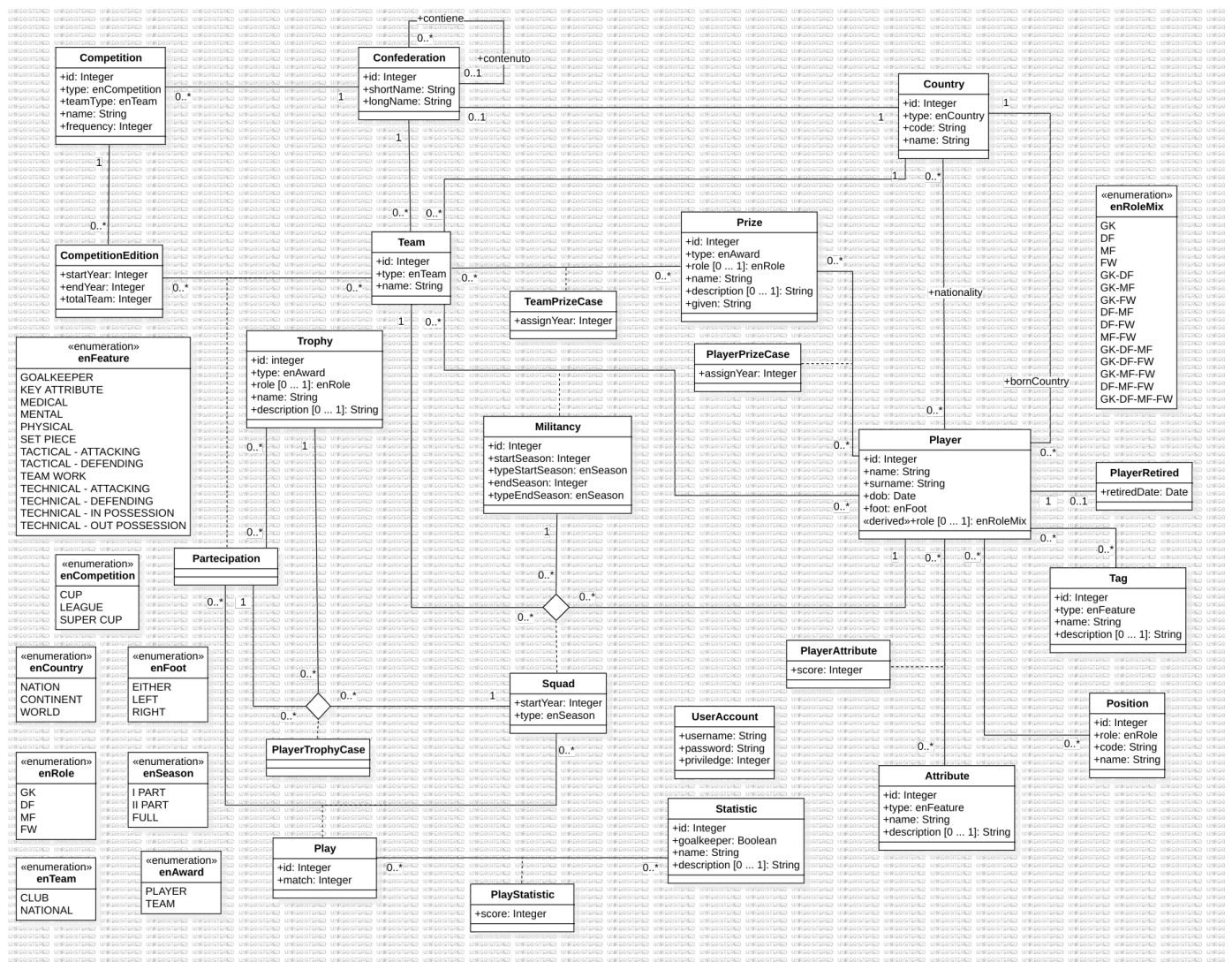
Partizionamento/Accorpamento di Entità e Associazioni

L'unica accorpamento che potrebbe essere effettuato nel Class Diagram è quello tra **Country** e **Confederation**, poiché tra esse vi è un'associazione 1 a 1.

Si è arrivati però alla conclusione che c'è bisogno che le due classi siano separate, perché nelle altre associazioni che esse hanno con le altre classi, svolgono un ruolo centrale.

Scelta degli Identificatori Primari

2.2.3 Class Diagram ristrutturato



2.2.4 Dizionario

Dizionario delle Classi

Classe	Descrizione	Attributo
Attribute	Rappresenta gli attributi di un calciatore.	id(Integer) [chiave surrogata]: Rappresenta l'identificativo di un Attributo. type(enFeature) : Rappresenta il tipo di un Attributo. name(String) [chiave naturale]: Rappresenta il nome di un Attributo. description(String) [parziale]: Rappresenta la descrizione di un Attributo.
Competition	Rappresenta le competizioni calcistiche.	id(Integer) [chiave surrogata]: Rappresenta l'identificativo di una Competizione. type(enCompetition) : Rappresenta il tipo di una Competizione. teamType(enTeam) : Rappresenta il tipo di squadra che può partecipare alla Competizione. name(String) [chiave naturale]: Rappresenta il nome di una Competizione. frequency(Integer) : Rappresenta la frequenza di una Competizione.
CompetitionEdition	Rappresenta le edizioni delle competizioni calcistiche.	startYear(Integer) [chiave parziale]: Rappresenta l'anno di inizio di un'Edizione. endYear(Integer) [chiave parziale]: Rappresenta l'anno di fine di un'Edizione. totalTeam(Integer) : Rappresenta il numero di team che partecipano in un'Edizione.
Confederation	Rappresenta le confederazioni calcistiche.	id(Integer) [chiave surrogata]: Rappresenta l'identificativo di una Confederazione. shortName(String) : Rappresenta il nome abbreviato di una Confederazione. longName(String) [chiave naturale]: Rappresenta il nome esteso di una Confederazione.

Country	Rappresenta i paesi in cui si gioca ufficialmente a calcio.	id(Integer) [chiave surrogata]: Rappresenta l'identificativo di un Paese. type(enCountry) : Rappresenta il tipo di un Paese. code(String) [chiave naturale]: Rappresenta il codice ISO 3166-1 alpha-3 di un Paese. name(String) [chiave naturale]: Rappresenta il nome di un Paese.
Player	Rappresenta i calciatori.	id(Integer) [chiave surrogata]: Rappresenta l'identificativo di un Calciatore. name(String) : Rappresenta il nome di un Calciatore. surname(String) : Rappresenta il cognome di un Calciatore. dob(Date) : Rappresenta la data di nascita. foot(enFoot) : Rappresenta il piede preferito di un Calciatore. role(enRoleMix) [derivato, parziale]: Rappresenta i possibili ruoli di gioco di un Calciatore.
PlayerRetired	Rappresenta i calciatori che sono ritirati.	retiredDate(Date) : Rappresenta la data di ritiro di un calciatore.
Position	Rappresenta le posizioni di gioco di un Calciatore.	id(Integer) [chiave surrogata]: Rappresenta l'identificativo di una Posizione. role(enRole) : Rappresenta il ruolo associato ad una Posizione. code(String) [chiave naturale]: Rappresenta il nome abbreviato di una Posizione. name(String) [chiave naturale]: Rappresenta il nome di una Posizione.

Prize	Rappresenta i premi calcistici.	id (Integer)[chiave surrogata]: Rappresenta l'identificativo del Premio. type (enAward): Rappresenta il tipo del Premio. role (enRole)[parziale]: Rappresenta il ruolo a cui è associato un Premio. name (String)[chiave naturale]: Rappresenta il nome del Premio. description (String)[parziale]: Rappresenta la descrizione del Premio. given (String): Rappresenta il nome della società calcistica che conferisce il Premio.
Statistic	Rappresenta le statistiche di un calciatore.	id (Integer)[chiave surrogata]: Rappresenta l'identificativo della Statistica. goalkeeper (Boolean): Rappresenta se true che la Statistica è associata soltanto al portiere, altrimenti a tutti i ruoli. name (String): Rappresenta il nome della Statistica. description (String)[parziale]: Rappresenta la descrizione della Statistica.
Tag	Rappresenta i tag di un calciatore.	id (Integer)[chiave surrogata]: Rappresenta l'identificativo del Tag. type (enFeature): Rappresenta il tipo del Tag. name (String)[chiave naturale]: Rappresenta il nome del Tag. description (String)[parziale]: Rappresenta la descrizione del Tag.
Team	Rappresenta le squadre di calcio.	id (Integer)[chiave surrogata]: Rappresenta l'identificativo della Squadra. type (enTeam): Rappresenta il tipo della Squadra. name (String)[chiave naturale]: Rappresenta il nome della Squadra.

Trophy	Rappresenta i trofei calcistici.	id (Integer)[chiave surrogata]: Rappresenta l'identificativo del Trofeo. type (enAward): Rappresenta il tipo del Trofeo. role (enRole)[parziale]: Rappresenta il ruolo a cui è associato un Trofeo. name (String)[chiave naturale]: Rappresenta il nome del Trofeo. description (String)[parziale]: Rappresenta la descrizione del Trofeo.
UserAccount	Rappresenta gli utenti dell'applicativo.	username (String)[chiave naturale]: Rappresenta l'username dell'Account dell'Utente. password (String): Rappresenta la password dell'Account dell'Utente. privilege (Integer): Rappresenta i privilegi dell'Account dell'Utente.

Dizionario delle Associazioni

Nome	Descrizione	Classe in Relazione	Attributo
Militancy	Esprime le militanze di un calciatore in una squadra.	Player [0 ... *]: Indica che un Calciatore può militare in più Squadre. Team [0 ... *]: Indica che una Squadra può essere associata a più Calciatori.	id(Integer)[chiave surrogata]: Rappresenta l'identificativo di una Militanza. startSeason(Integer): Rappresenta la stagione d'inizio di una Militanza. typeStartSeason(enSeason): Rappresenta in che parte della stagione è iniziata la Militanza. endSeason(Integer): Rappresenta la stagione in cui termina la Militanza. typeEndSeason(enSeason): Rappresenta in che parte della stagione è terminata la Militanza.
PlayerPrizeCase	Esprime i premi di un calciatore.	Player [0 ... *]: Indica che un Calciatore può essere associato a più Premi. Prize [0 ... *]: Indica che un Premio può essere associato, in anni diversi, a più Premi.	assignYear(Integer): Rappresenta l'anno di assegnazione del Premio al Calciatore.
Nationality	Esprime le nazionalità di un calciatore.	Country [0 ... *]: Indica che uno stesso paese può essere associato a più calciatore. Player [0 ... *]: Indica che un calciatore può avere più nazionalità.	
bornCountry	Esprime il paese di nascita di un calciatore.	Country [0 ... *]: Indica che un paese può essere il paese di nascita di più calciatori. Player [1]: Indica che un calciatore ha uno e un solo paese di nascita.	
Player-PlayerRetired	Esprime i calciatori ritirati.	Player [0 ... 1]: Indica che un Calciatore può essere o non essere ritirato. PlayerRetired [1]: Indica che una data di ritiro si riferisce ad uno e un solo Calciatore.	

Player-Tag	Esprime i tag di un calciatore.	Player [0 ... *]: Indica che un Calciatore può essere associato a più Tag. Tag [0 ... *]: Indica che uno Tag può essere associato a più Calciatori.	
Player-Position	Esprime le posizioni di gioco di un calciatore.	Player [0 ... *]: Indica che un Calciatore può essere associato a più Posizioni. Position [0 ... *]: Indica che una Posizione può essere associata a più Calciatori.	
PlayerAttribute	Esprime gli attributi di un calciatore.	Player [0 ... *]: Indica che un Calciatore può essere associato a più Attributi. Attribute [0 ... *]: Indica che un Attributo può essere associato a più Calciatori.	score(Integer): Rappresenta il valore di un Attributo associato al Calciatore.
Squad	Esprime la rosa in un determinato anno di una squadra.	Militancy [0 ... *]: Indica che una Militanza può essere associato a più Rose. Player [0 ... *]: Indica che un Calciatore può essere associato a più Rose. Team [0 ... *]: Indica che una Squadra può essere associata a più Rose. Squad -> Team [1]: Indica che una Rosa può essere associata ad una e una sola Squadra. Squad -> Militancy [1]: Indica che una Rosa può essere associata ad una e una sola Militanza. Squad -> Player [1]: Indica che una Rosa può essere associata ad un e un solo Calciatore.	startYear(Integer): Rappresenta la stagione di riferimento di una Rosa. type(enSeason): Rappresenta fino a che parte della stagione un Giocatore era in Rosa.
TeamPrizeCase	Esprime i premi di una squadra.	Team [0 ... *]: Indica che una Squadra può essere associata a più Premi. Prize [0 ... *]: Indica che un Premio, in anni diversi, a più Squadre.	assignYear(Integer): Rappresenta l'anno di assegnazione di un Premio ad una Squadra.

Team-Country	Esprime la nazionalità di una squadra.	Team [1]: Indica che una Squadra può essere associata ad un e un solo Paese. Country [0 ... *]: Indica che un Paese può essere associato a più Squadre.	
Team-Confederation	Esprime la confederazione di cui è membro una squadra.	Team [1]: Indica che una Squadra può essere associata ad un e un solo Paese. Confederation [0 ... *]: Indica che una Confederazione può essere associata a più Squadre.	
Partecipation	Esprime la partecipazione di una squadra ad un'edizione di una competizione.	Team [0 ... *]: Indica che una Squadra può partecipare a più Edizioni. CompetitionEdition [0 ... *]: Indica che una Edizione può essere associata a più squadre.	
Competition-CompetitionEdition	Esprime le edizioni di una competizione. È una relazione identificante.	Competition[0 ... *]: Indica che una Competizione può essere associata a più Edizioni. CompetitionEdition[1]: Indica che un'Edizione può essere associata ad una e una sola Competizione.	
Competition-Confederation	Esprime la confederazione che organizza la competizione.	Competition [1]: Indica che una Competizione può essere associata ad una e una sola Confederazione. Confederation [0 ... *]: Indica che una Confederazione può essere associata a più Competizioni.	
Confederation-Confederation	Esprime la possibilità di una confederazione di avere come membri altre confederazioni, o essere membro a sua volta.	Confederation [0 ... 1] ruolo (contenuto): Indica che una Confederazione può o non può essere membra di un'altra Confederazione. Confederation [0 ... *] ruolo (contiene): Indica che una Confederazione può essere associata a più Confederazioni.	
Confederation-Country	Esprime l'appartenza di una confederazione ad un unico paese.	Confederation [1]: Indica che una Confederazione può essere associata ad un e un solo Paese. Country [0 ... 1]: Indica che un Paese può essere associata a nessuna o ad una Confederazione.	
Partecipation-Trophy	Esprime la bacheca dei trofei di una squadra.	Trophy [0 ... *]: Indica che un Trofeo può essere associata a più Partecipazioni di una Squadra ad una Edizione. Partecipation [0 ... *]: Indica che una Partecipazione può essere associata a più Trofei.	

PlayTrophyCase	Esprime la bacheca dei trofei di un calciatore.	<p>Partecipation [0 ... *]: Indica che una Partecipazione può essere associata a più Bacheche.</p> <p>Trophy [0 ... *]: Indica che un Trofeo può essere associato a più Bacheche.</p> <p>Squad [0 ... *]: Indica che una Rosa può essere associata a più Bacheche.</p> <p>PlayTrophyCase -> Partecipation [1]: Indica che una Bacheca può essere associata ad una e una sola Partecipazione.</p> <p>PlayTrophyCase -> Trophy [1]: Indica che una Bacheca può essere associata ad un e un solo Trofeo.</p> <p>PlayTrophyCase -> Squad [1]: Indica che una Bacheca può essere associata ad una e una sola Rosa.</p>	
Play	Esprime in quale edizione un calciatore, in una squadra e in una certa stagione, ha giocato.	<p>Partecipation [0 ... *]: Indica che una Partecipazione può essere associata a più Rose.</p> <p>Squad [0 ... *]: Indica che una Rosa può essere associata a più Partecipazioni.</p>	<p>id(Integer)[chiave surrogata]: Rappresenta l'identificativo di un Gioco.</p> <p>match(Integer): Rappresenta il numero di presenze di un calciatore in un Gioco.</p>
PlayStatistic	Esprime le statistiche di un gioco.	<p>Play [0 ... *]: Indica che un Gioco può essere associato a più Statistiche.</p> <p>Statistic [0 ... *]: Indica che una Statistica può essere associata a più Giochi.</p>	<p>score(Integer): Rappresenta il valore di una Statistica per un Gioco.</p>

Dizionario dei Vincoli

[illegible]

