Progetto Database Calciatori

Mario Ponticiello

Pasquale Sole

N86004563

N86004581

Anno Accademico
23/24

Indice

1 Progettazione Concettuale

1.1 Analisi dei Requisiti

La traccia chiede di sviluppare un sistema informativo per la gestione di calciatori di tutto il mondo; quindi, nel nostro progetto teniamo in considerazione sia i calciatori di leghe alte, come serie A, Premier League ecc. e sia calciatori di leghe inferiori/giovanili, sia maschili che femminili. In questa analisi andremo a individuare tutte le informazioni presenti nel testo che definiranno la struttura del nostro database dei calciatori, in questa fase andremo a definire le varie entità e relazioni tra di loro, ed eventuali vincoli.

"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.)"

Dal testo veniamo a conoscenza del fatto che i calciatori posseggono un Nome, Cognome, una data di nascita, qual è il loro piede preferito, abbiamo scelto di rappresentare queste informazioni come attributi dell'entità calciatore, mentre per quanto riguarda i ruoli creare un'entità a sé stante ci è sembrata la miglior opzione, andando a relazionare il giocatore con il ruolo, in questo modo un giocatore potrà ricoprire anche più ruoli diversi tra di loro. Stesso ragionamento per quanto riguarda le feature caratteristiche, abbiamo quindi creato una nuova entità che si andrà a relazionare con il calciatore permettendo a quest'ultimo di possedere zero, una o più feature, ovvero le loro migliori abilità come, ad esempio, il colpo di testa, rovesciata, pallonetto, punizioni dalla distanza ecc.

"Il giocatore può avere anche una data di ritiro a seguito della quale decide di non giocare più."

In questo progetto si chiede anche di ricordare i giocatori che hanno smesso di giocare, tener traccia di tutte le loro informazioni aggiungendo una data che sta ad indicare il giorno in cui hanno smesso di giocare, abbiamo quindi optato di aggiungere una generalizzazione a calciatore chiamandola Calciatore ritirato, come unico attributo la data ritiro.

“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).”

Sappiamo poi che i calciatori appartengono a una o più squadre, dobbiamo tenere traccia della loro data di inizio e fine militanza per ogni singola squadra in cui hanno giocato. Durante il periodo di permanenza per ogni squadra dobbiamo memorizzare tutti i gol segnati da quel calciatore e anche i gol subiti, statistica riservata solo ai giocatori che hanno ricoperto almeno una volta il ruolo di portiere. Tutte queste informazioni sono conservate nella classe associativa tra calciatore e squadra, tranne per i gol subiti, questa informazione deve apparire solo ai giocatori che hanno interpretato almeno una volta il ruolo di portiere, abbiamo quindi creato una generalizzazione di calciatore intitolata Portiere che non presenta attributi ma soltanto una classe associativa verso Squadra conservando i gol subiti e la data di inizio e fine militanza in quella squadra.

“Il giocatore può inoltre vincere dei trofei, individuali o di squadra”

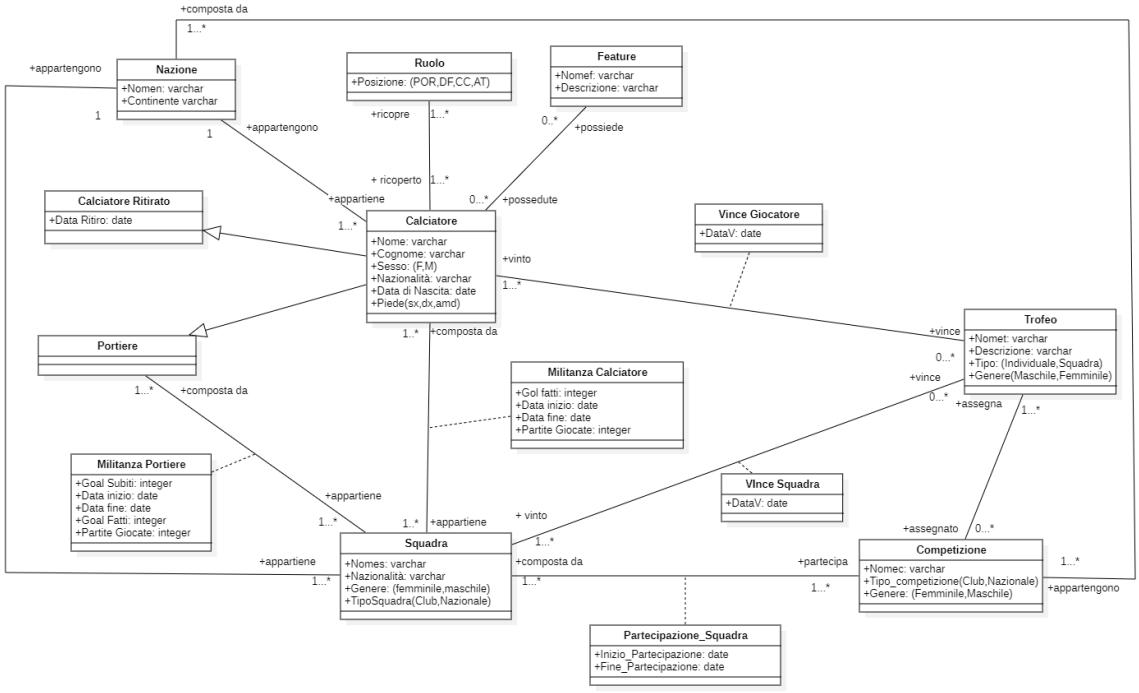
Per soddisfare questa richiesta c’è bisogno di creare una nuova entità trofeo, andandola a differenziare tra quelli singoli e di squadra tramite un attributo “Tipo”, creiamo una classe associativa da trofeo a giocatore conservando anche la data di quando il trofeo è stato assegnato, inoltre visto che sono presenti anche i trofei di squadra, tutti e soli questi trofei verranno collegati con l’entità squadra tramite un’altra classe associativa conservando anche qui la data di assegnazione.

“Le squadre di calcio sono specificate dal loro nome e nazionalità.”

Abbiamo creato una nuova entità squadra che ha come attributi il suo nome, e la nazionalità da dove proviene, ogni squadra partecipa a diverse competizioni nazionali come il campionato o la coppa oppure europee/internazionali come la Champions League. Sia le squadre che le competizioni si dividono rispettivamente in sezione maschile e femminile tramite un attributo tipo.

1.2 Modello Concettuale

Schema concettuale del database costruito tramite le informazioni ottenute dall'analisi dei requisiti



1.3 Dizionario Entità

Entità	Descrizione	Attributi
Calciatore	un calciatore è un atleta che fa parte di una squadra di calcio.	**Nome**(string): è il nome della persona **Cognome**(string): è il cognome della persona **Sesso**(F,M): indica se la persona è un uomo o una donna. **DataNascita**(Date): indica la data di nascita del giocatore **Nazionalità**(string): indica il paese di provienza della persona **Piede**(SX,DX,AMD): indica il piede predominante della persona.
Calciatore Ritirato	sta a rappresentare tutti i calciatori che si sono ritirati dal calcio giocato.	**Data ritiro**(data): indica la data in cui la persona ha lasciato il calcio professionistico.
Portiere	Indica tutti i calciatori che giocano in porta, il loro compito è quello di non far subire gol alla squadra.	
Squadra	è un gruppo di calciatori che partecipano alle partite di calcio contro altre squadre.	**NomeS**(string): è il nome della squadra di calcio **Nazionalità**(string): indica in quale nazione è stata fondata la squadra di calcio. **Genere**(femminile,maschile): indica il sesso di tutti i calciatori che fanno parte della squadra. **Tipo Squadra**(Club,Nazionale): indica se è una nazionale o un club
Ruolo	indica in che posizione del campo gioca il giocatore, difesa, centrocampo e attacco, ogni posizione ha i propri compiti.	**Posizione**(POR,DF,CC,AT): indica in che posizione del campo.
Feature	elenco di caratteristiche possedute dai calciatori, sta ad indicare le proprie qualità individuali	**NomeF**(string): è il nome specifico della feature che riassume la caratteristica. **Descrizione**(string): fornisce una descrizione più dettagliata delle caraterristiche individuali
Trofeo	è un premio che viene assegnato ai giocatori e alle squadre che hanno eccelso in un torneo o in un'ambito particolare.	**NomeT**(string): indica il nome assegnato al trofeo **Descrizione**(string): fornisce una descrizione dei criteri di assegnazione del trofeo **Genere**(femminile,maschile): indica il genere del trofeo che può essere assegnato ai giocatori. **Tipo**(individuale,squadra): stabilisce se un trofeo è individuale o l'ha vinto una squadra
Competizione	è un insieme di squadre che si affronteranno nel corso della stagione calcistica	**NomeC**(string): è il nome assegnato alla competizione **Genere**(femminile,maschile): indica quali squadre vi partecipano se le maschili o femminili **Tipo Competizione**(Club,Nazionale): indica quali tipi di squadre vi partecipano se le nazionali o i club
Nazione	elenco delle nazioni a cui ad appartengono i giocatori	**NomeN**(string): è il nome specifico della Nazione. **Continente**(string): indica il continente in cui si trova la nazione.

1.4 Dizionario Associazioni

Associazione	Descrizione	Attributi
Militanza Calciatore	Rappresenta il periodo di permanenza di un giocatore in una determinata squadra. Associazione **molti-a-molti** sta a significare che un calciatore può appartenerea più squadre mentre una squadra può essere composta da più giocatori.	**Goal Fatti**(int): numero dei gol segnati in quel periodo **Data Inizio**(date): data di ingresso nella squadra **Data Fine** (date): data di fine militanza nella squadra **Partite Giocate**(int): numero di partite giocate in quel periodo.
Militanza Portieri	Rappresenta il periodo di permanenza di un giocatore avente il ruolo di portiere in una determinata squadra. Associazione **molti-a-molti** sta a significare che un portiere appartiene a una o più squadre e una squadra può essere composte da uno o più portieri.	**Goal Subiti**(int): numero dei goal subiti durante quel periodo **Data Inizio**(date): data di ingresso nella squadra **Data Fine** (date): data di fine militanza nella squadra **Partite Giocate**(int): numero di partite giocate in quel periodo. **Goal Fatti**(int): numero dei gol segnati in quel periodo
Partecipazione Squadra	Insieme di competizioni a cui la squadra partecipa. Associazione **molti-a-molti** sta a significare che una squadra può partecipare a una o più competizioni e una competizione può essere composta da più squadre.	**DataInizio**(date): indica l'anno di inizio partecipazione alla competizione **DataFine**(date): indica l'anno di fine partecipazione alla competizione
Vince Giocatore	Insieme di trofei vinti dal giocatore. Associazione **molti-a-molti**, un giocatore può vincere nessuno, uno o più tipi di trofei individuali, mentre quel tipo di trofeo(es. pallone d'oro) può essere vinto da uno o più calciatori.	**DataV**(date): indica la data di assegnazione del trofeo al giocatore
Vince Squadra	Insieme di trofei vinti dalla squadra. Associazione **molti-a-molti**, una squadra può vincere nessun, uno opiù trofei, mentre lo stesso tipo di trofeo (es. campionato italiano), può essere vinto da più squadre.	**DataV**(date): indica la data di assegnazione del trofeo alla squadra

2 Ristrutturazione Modello Concettuale

2.1 Analisi delle Ridondanze

Non sono presenti ridondanze nel nostro schema.

2.2 Eliminazione delle Generalizzazioni

Essendo una generalizzazione parziale non disgiunta, ovvero che un calciatore può essere sia portiere che ritirato andando a rompere il vincolo di disgiunzione, successivamente la definiamo parziale perchè un calciatore può non essere sia portiere che ritirato. Abbiamo deciso di accorpare i due figli al padre, Calciatore erediterà quindi l'attributo Data ritiro e una relazione, Militanza Portieri, con squadra che dovrà essere gestita in maniera opportuna con dei vincoli, ammettendo nella classe associativa solo i giocatori che abbiano ricoperto almeno una volta il ruolo di portiere.

2.3 Eliminazione degli Attributi Multivalore

Non sono presenti Attributi Multivalore

2.4 Eliminazione degli Attributi Strutturati

Non sono presenti degli Attributi Strutturati

2.5 Partizionamento/Accorpamento di Entità e Associazioni

Nessun Partizionamento/Accorpamento di Entità e Associazioni effettuato

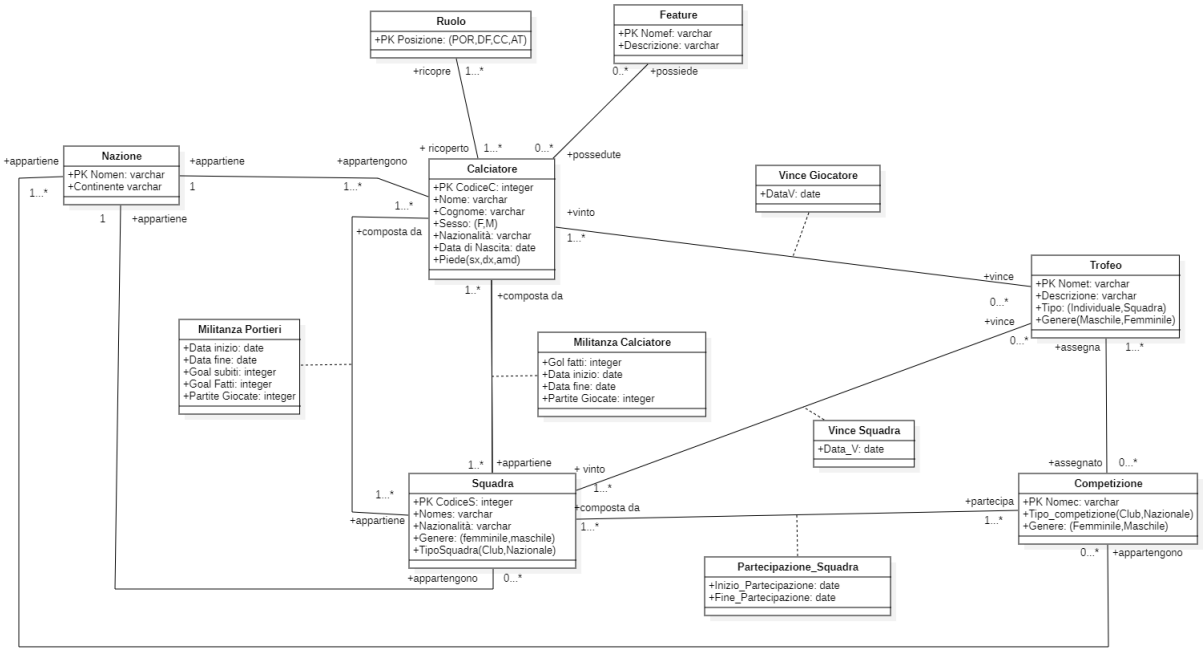
2.6 Scelta degli Identificatori Primari

In questa fase sceglieremo i nostri identificativi univoci delle nostre entità,in particolare:

1. L'entità **Calciatore** non aveva attributi univoci per essere chiave quindi è stato aggiunto CodiceC che indentifica un giocatore.
2. Per quando riguarda **Squadra** abbiamo una situazione analoga, ed abbiamo aggiunto CodiceS.
3. L'entità **Ruolo** ha solo un attributo ed è anche identificatore primario Ruolo..
4. L'entità **Feature** nel suo caso abbiamo l'attrbuto Nome che lo distingue.
5. L'entità **Trofeo** presenta al suo interno l'identificativo Nome.
6. L'entità **competizione** come identificativo abbiamo l'attributo Nome.

2.7 Modello UML Ristrutturato

Schema concettuale ottenuto costruito dopo aver effettuato una Ristrutturazione del modello



2.8 Dizionario delle Entità Ristrutturato

Entità	Descrizione	Attributi
Calciatore	un calciatore è un atleta che fa parte di una squadra di calcio.	**CodiceC**(int): codice univoco del calciatore **Nome**(string): è il nome della persona **Cognome**(string): è il cognome della persona **Sesso**(F,M): indica se la persona è un uomo o una donna. **DataNascita**(Date): indica la data di nascita del giocatore **Piede**(SX,DX,AMD): indica il piede predominante della persona. **Nazionalità**(string): indica il paese di provenienza della persona.
Squadra	è un gruppo di calciatori che partecipano alle partite di calcio contro altre squadre.	**CodiceS**(int): codice univoco della squadra **NomeS**(string): è il nome della squadra di calcio **TipoSquadra**(Club,Nazionale): indica il tipo di squadra da calcio. **Nazionalità**(string): indica in quale nazione è stata fondata la squadra di calcio **Genere**(femminile,maschile): indica il sesso di tutti i calciatori che fanno parte della squadra.
Ruolo	indica in che posizione del campo gioca il giocatore, difesa, centrocampo e attacco, ogni posizione ha i propri compiti.	**Posizione**(POR,DF,CC,AT): indica in che posizione del campo.
Feature	elenco di caratteristiche possedute dai calciatori, sta ad indicare le proprie qualità individuali	**NomeF**(string): è il nome specifico della feature che riassume la caratteristica. **Descrizione**(string): fornisce una descrizione più dettagliata delle caraterristiche individuali
Trofeo	è un premio che viene assegnato ai giocatori e alle squadre che hanno eccelso in un torneo o in un'ambito particolare.	**NomeT**(string): indica il nome assegnato al trofeo **Descrizione**(string): fornisce una descrizione dei criteri di assegnazione del trofeo **Tipo**(individuale,squadra): stabilisce se un trofeo è individuale o l'ha vinto una squadra **Genere**(femminile,maschile): indica il genere del trofeo che può essere assegnato ai giocatori.
Competizione	è un insieme di squadre che si affronteranno nel corso della stagione calcistica	**NomeC**(string): è il nome assegnato alla competizione **Genere**(femminile,maschile): indica quali tipi di squadre vi partecipano **TipoCompetizione**(Club,Nazionale): indica il tipo di competizione a cui le squadre partecipano.
Nazione	elenco delle nazioni a cui ad appartengono i giocatori	**NomeN**(string): è il nome specifico della Nazione. **Continente**(string): indica il continente in cui si trova la nazione.

2.9 Dizionario delle Associazioni Ristrutturato

Associazione	Descrizione	Attributi
Militanza Calciatore	Rappresenta il periodo di permanenza di un giocatore in una determinata squadra. Associazione **molti-a-molti** sta a significare che un calciatore può appartenerea più squadre mentre una squadra può essere composta da più giocatori.	**Goal Fatti**(int): numero dei gol segnati in quel periodo **Data Inizio**(date): data di ingresso nella squadra **Data Fine** (date): data di fine militanza nella squadra **Partite Giocate**(int): numero di partite giocate in quel periodo.
Militanza Portieri	Rappresenta il periodo di permanenza di un giocatore avente il ruolo di portiere in una determinata squadra. Associazione **molti-a-molti** sta a significare che un portiere appartiene a una o più squadre e una squadra può essere composte da uno o più portieri.	**Goal Subiti**(int): numero dei goal subiti durante quel periodo **Data Inizio**(date): data di ingresso nella squadra **Data Fine** (date): data di fine militanza nella squadra **Partite Giocate**(int): numero di partite giocate in quel periodo. **Goal Fatti**(int): numero dei gol segnati in quel periodo
Partecipazione Squadra	Insieme di competizioni a cui la squadra partecipa. Associazione **molti-a-molti** sta a significare che una squadra può partecipare a una o più competizioni e una competizione può essere composta da più squadre.	**InizioPartecipazione**(Date): indica la in cui la squadra ha iniziato a partecipare ad una competizione in quell'anno **FinePartecipazione**(Date): indica la in cui la squadra ha finito di partecipare ad una competizione in quell'anno
Vince Giocatore	Insieme di trofei vinti dal giocatore. Associazione **molti-a-molti**, un giocatore può vincere nessuno, uno o più tipi di trofei individuali, mentre quel tipo di trofeo(es. pallone d'oro) può essere vinto da uno o più calciatori.	**DataV**(date): indica la data di assegnazione del trofeo al giocatore
Vince Squadra	Insieme di trofei vinti dalla squadra. Associazione **molti-a-molti**, una squadra può vincere nessun, uno opiù trofei, mentre lo stesso tipo di trofeo (es. campionato italiano), può essere vinto da più squadre.	**DataV**(date): indica la data di assegnazione del trofeo alla squadra

3 Modello Logico

Calciatore (CodiceC,Nome,Cognome,Piede,Datan,Sesso,DataRitiro,Nazionalità)
MilitanzaCalciatore (DataInizio,DataFine,GoalFatti,PartiteGiocate,CodiceC,CodiceS)

MilitanzaPortiere (DataInizio,DataFine,GoalFatti,PartiteGiocate,GoalSubiti,CodiceC,CodiceS)

Squadra (CodiceS,NomeS,Genere,TipoSquadra,Nazionalità)

StagioneSquadra (InizioPartecipazione,FinePartecipazione,NomeC,CodiceS)

Competizione (NomeC,Genere,TipoCompetizione)

VinceGiocatore (DataV,CodiceC,CodiceT)

VinceSquadra (DataV,CodiceS,CodiceT)
Trofeo (NomeT,Descrizione,Tipo,Genere)

Assegna (NomeC,NomeT)

Feature (NomeF,Descrizione)

PossiedeF (CodiceC,NomeF)

Nazione (NomeN,Continente)

Ruolo (Posizione)

Ricopre (CodiceC,Ruolo)

Accetta (NomeC,NomeN)

4 Modello Fisico

4.1 Creazione Tipi

Tipo-squadra:

```
create type tipo-squadra as enum(
    'Club',
    'Nazionale'
)
```

Sesso:

```
create type Sesso as enum(
    'Femmina',
    'Maschio'
)
```

Piede:

```
create type Piede as enum(
    'Sinistro',
    'Destro',
    'Ambidestro'
)
```

Genere:

```
create type genere as enum(
    'Femminile',
    'Maschile'
)
```

Posizione:

```
create type posizione as enum(
    'portiere',
    'difensore',
    'centrocampista',
    'attaccante'
)
```

Tipo-trofeo:

```
create type tipo-trofeo as enum(
    'individuale',
    'squadra'
)
```

4.2 Creazione Tabelle

Calciatore:

```
CREATE TABLE Calciatore (
    CodiceC SERIAL NOT NULL,
    Nome varchar(20) NOT NULL,
    Cognome varchar(25) NOT NULL,
    Piede piede NOT NULL,
    DataN date NOT NULL,
    Sesso sesso NOT NULL,
    Data-ritiro DATE,
    Nazionalità varchar(20) NOT NULL,
    PRIMARY KEY(CodiceC),
    FOREIGN KEY (nazionalità) REFERENCES nazione(nomen)
);
```

Squadra:

```
CREATE TABLE Squadra (
    CodiceS serial NOT NULL,
    NomeS varchar(20) NOT NULL,
    Nazionalità varchar(25) NOT NULL,
    genere genere,
```

```
    tipo-squadra tipo-squadra,
    PRIMARY KEY(CodiceS),
    FOREIGN KEY (Nazionalità) REFERENCES nazione(nomen)
);
```

Feature:

```
CREATE TABLE Feature (
    NomeF varchar(20) NOT NULL,
    Descrizione varchar(200),
    PRIMARY KEY(NomeF)
);
```

Ruolo:

```
CREATE TABLE Ruolo (
    Posizione posizione NOT NULL,
    PRIMARY KEY(Posizione)
);
```

Trofeo:

```
CREATE TABLE Trofeo (
    NomeT varchar(50) NOT NULL,
    Descrizione varchar(200),
    Tipo tipo-trofeo NOT NULL,
    Genere genere NOT NULL,
    PRIMARY KEY (NomeT)
);
```

Competizione:

```
CREATE TABLE Competizione (
    NomeC varchar(50) NOT NULL,
    genere genere,
    tipo-competizione tipo-squadra,
    PRIMARY KEY(NomeC)
);
```

Militanza-Portiere:

```
CREATE TABLE Militanza-Portiere(
    Data-Inizio date NOT NULL,
    Data-Fine date,
    Goal-Fatti int NOT NULL,
    Partite-Giocate int NOT NULL,
    Goal-Subiti int NOT NULL,
    CodiceC int NOT NULL,
    CodiceS int NOT NULL,
    FOREIGN KEY (CodiceC) REFERENCES calciatore(CodiceC),
    FOREIGN KEY (CodiceS) REFERENCES squad
```

Vince-Giocatore:

```
CREATE TABLE Vince-Giocatore(
    Data-Vittoria date,
    CodiceC int NOT NULL,
    nomet varchar(25) NOT NULL,
    FOREIGN KEY (nomet) REFERENCES trofeo(nomet),
    FOREIGN KEY (CodiceC) REFERENCES calciatore(CodiceC)
);
```

Vince-Squadra:

```
CREATE TABLE Vince-Squadra(
    Data-Vittoria date,
    Codices int NOT NULL,
    nomet varchar(25) NOT NULL,
    FOREIGN KEY (nomet) REFERENCES trofeo(nomet),
    FOREIGN KEY (Codices) REFERENCES squadra(Codices)
);
```

Stagione-Squadra:

```
CREATE TABLE Stagione-Squadra(
    inizio-partecipazione date NOT NULL,
```

```
    fine-partecipazione date,
    nomec varchar(25) NOT NULL,
    CodiceS int NOT NULL,
    FOREIGN KEY (CodiceS) REFERENCES squadra(CodiceS),
    FOREIGN KEY (nomec) REFERENCES competizione(nomec)
);
```

Militanza-Calciatore:

```
CREATE TABLE Militanza-Calciatore(
    Data-Inizio date NOT NULL,
    Data-Fine date,
    Goal-Fatti int NOT NULL,
    Partite-Giocate int NOT NULL,
    CodiceC int NOT NULL,
    CodiceS int NOT NULL,
    FOREIGN KEY (CodiceC) REFERENCES calciatore(CodiceC),
    FOREIGN KEY (CodiceS) REFERENCES squadra(CodiceS)
);
```

Ricopre:

```
CREATE TABLE Ricopre (
    CodiceC int NOT NULL,
    Ruolo posizione NOT NULL,
    FOREIGN KEY (CodiceC) REFERENCES Calciatore(CodiceC),
    FOREIGN KEY (Ruolo) REFERENCES Ruolo(Posizione)
);
```

Possiedef:

```
CREATE TABLE Possiedef (
    CodiceC int NOT NULL,
    NomeF varchar(15) NOT NULL,
    FOREIGN KEY (CodiceC) REFERENCES Calciatore(CodiceC),
    FOREIGN KEY (NomeF) REFERENCES Feature(NomeF)
);
```

Assegna:

```
CREATE TABLE Assegna (
    nomet varchar(25) NOT NULL,
    nomec varchar(25) NOT NULL,
    FOREIGN KEY (nomet) REFERENCES Trofeo(NomeT),
    FOREIGN KEY (nomec) REFERENCES Competizione(NomeC)
);
```

Accetta:

```
CREATE TABLE Accetta (
    nomec varchar(25) NOT NULL,
    nomen varchar(25) NOT NULL,
    FOREIGN KEY (nomen) REFERENCES Nazione(Nomen),
    FOREIGN KEY (nomec) REFERENCES Competizione(NomeC)
);
```

4.3 Creazione Vincoli

1. ```
   ALTER TABLE squadra ADD CONSTRAINT squadtype
   UNIQUE(nomes, genere)
   ```

2. ```
   ALTER TABLE militanza_calciatore
   ADD CONSTRAINT didf CHECK (data_fine > data_inizio);

   ALTER TABLE militanza_portiere
   ADD CONSTRAINT didf CHECK (data_fine > data_inizio);
   ```

3. ```
   ALTER TABLE accetta ADD UNIQUE(nomec, nomen);
   ```

4. ```
   ALTER TABLE militanza_portiere
   ADD UNIQUE(codicec, codices, data_inizio);

   ALTER TABLE militanza_calciatore
   ADD UNIQUE(codicec, codices, data_inizio);
   ```

5. ```
   ALTER TABLE vince_squadra
   ADD UNIQUE(data_vittoria,nomet);
   ```
   ```

```
ALTER TABLE vince_giocatore
ADD UNIQUE(data_vittoria,nomet);
```

4.4 Creazione Trigger

Trigger1:

Questo trigger controllerà che la data di nascita del giocatore inserito sia minore del suo inizio carriera.

```
CREATE OR REPLACE TRIGGER datanmin
BEFORE INSERT
ON public.calciatore
FOR EACH ROW
EXECUTE FUNCTION public.datanminf();

CREATE OR REPLACE FUNCTION datanminf()
RETURNS trigger AS $$
DECLARE
mindi date;
BEGIN
select min(data_inizio) into mindi from (select min(data_inizio)
as data_inizio from militanza_calciatore  where codicec = new.codicec
union
select min(data_inizio) as data_inizio from militanza_portiere
where codicec = new.codicec);
if mindi is null then
    return new;
end if;
if(new.datan < mindi) then
    return new;
end if;
raise notice 'data di nascita non valida';
return null;
end;
$$ language plpgsql
```

Trigger2:
Il seguente trigger farà in modo che non venga inserita una data di ritiro minore della sua fine carriera, comparando le date delle varie militanze.

```
CREATE OR REPLACE TRIGGER dataritmax
BEFORE INSERT
ON public.calciatore
FOR EACH ROW
WHEN (new.data_ritiro IS NOT NULL)
EXECUTE FUNCTION public.dataritmaxf();

CREATE OR REPLACE FUNCTION dataritmaxf()
RETURNS trigger AS $$
DECLARE
dfmaxc date;
dfmaxp date;
BEGIN
select MAX(data_fine) into dfmaxc from militanza_calciatore m where m.codicec =
select MAX(data_fine) into dfmaxp from militanza_portiere m
where m.codicec = new.codicec;
if dfmaxc >= dfmaxp then
  if (new.data_ritiro >= dfmaxc) then
    return new;
  end if;
else
  if (new.data_ritiro >= dfmaxp)then
     return new;
  end if;
end if;
raise notice 'non è possibile inserire questa data ritiro';
return null;
end;
$$ language plpgsql
```

Trigger3:
Identico al trigger2 ma viene eseguito quando un giocatore viene aggiornato.

```
CREATE OR REPLACE TRIGGER updataritmax
BEFORE UPDATE OF data_ritiro
ON public.calciatore
FOR EACH ROW
WHEN (new.data_ritiro IS NOT NULL)
EXECUTE FUNCTION public.updataritmaxf();

CREATE OR REPLACE FUNCTION updataritmaxf() RETURNS trigger AS $$
DECLARE
dfmaxc date;
dfmaxp date;
BEGIN
select MAX(data_fine) into dfmaxc from militanza_calciatore m
where m.codicec = new.codicec;
select MAX(data_fine) into dfmaxp from militanza_calciatore m
where m.codicec = new.codicec;
if dfmaxc >= dfmaxp then
    if (new.data_ritiro >= dfmaxc) then
        return new;
    end if;
end if;
if dfmaxp >= dfmaxc then
    if (new.data_ritiro >= dfmaxp)then
        return new;
    end if;
end if;
raise notice 'non è possibile inserire questa data ritiro';
return null;
end;
$$ language plpgsql
```

Trigger4:

Il seguente trigger si assicurerà che i calciatori di movimento maschi vengano inseriti nelle squadre maschili e le femmine nelle squadre femminili.

```
CREATE OR REPLACE TRIGGER distinzione_calciatori
BEFORE INSERT
ON public.militanza_calciatore
FOR EACH ROW
EXECUTE FUNCTION public.inserimento_calciatori_genere();

CREATE OR REPLACE FUNCTION inserimento_calciatori_genere()
RETURNS trigger AS $$
DECLARE
persona_sesso varchar(15);
squadra_genere varchar(15);
BEGIN
select sesso into persona_sesso from calciatore where codicec = new.codicec;
select genere into squadra_genere from squadra  where codices = new.codices;
if (persona_sesso = 'Maschio' and squadra_genere = 'Maschile') OR
   (persona_sesso = 'Femmina' and squadra_genere = 'Femminile') then
   return new;
else
    raise notice 'non puoi inserire un giocatore % in una squadra
    %',persona_sesso,squadra_genere;
    return null;
end if;
end;
$$ language plpgsql
```

Trigger5:

Il seguente trigger si assicurerà che vengano messe le giuste date in militanza calciatore, la prima cosa che controllerà è se il giocatore avrà giocato anche come portiere nella stessa squadra, se questa cosa è vera allora la data inizio e data fine nuovi dovranno coincidere con quelli in militanza portiere se si vuole dire che ad esempio meret ha giocato una partita da titolare durante la sua militanza a napoli dalla stagione x a y, ma se mettiamo caso che meret abbia giocato a napoli da attaccante in una militanza diversa il trigger lo prenderà lo stesso affinche le date non si intersechino con le altre

militanze.
questo trigger aggiornerà automaticamente la data fine della precedente militanza se essa è null con la data inizio di quella appena inserita. Il funzionamento di questo trigger è tale grazie soprattutto all'ordine specifico delle casistiche che abbiamo controllato.

```
CREATE OR REPLACE TRIGGER porcalc
BEFORE INSERT
ON public.militanza_calciatore
FOR EACH ROW
EXECUTE FUNCTION public.inserimento_calciatore();

CREATE OR REPLACE FUNCTION inserimento_calciatore()
RETURNS trigger AS $$

DECLARE
curs1 refcursor;
df date;
di date;
maxdf date;
maxdi date;
BEGIN
/*controlliamo per prima cosa se il calciatore ha giocato anche da portiere
nella stessa squadra*/
if (new.codicec = (select codicec from  militanza_portiere where
codicec = new.codicec and codices = new.codices limit 1)) then
     open curs1 for
     select data_inizio,data_fine
     from militanza_portiere
     where codicec = new.codicec and codices = new.codices;
     loop
     fetch curs1 into di,df;
     if (new.data_inizio = di and new.data_fine = df ) then
          return new;
     end if;
     EXIT when not found;
     end loop;
     close curs1;
```

```
    open curs1 for
    select data_inizio,data_fine
    from militanza_portiere
    where codicec = new.codicec
    union
    select data_inizio,data_fine
    from militanza_calciatore
    where codicec = new.codicec;
    loop
    fetch curs1 into di,df;
    if ((new.data_inizio <= df and new.data_inizio >= di) or
   (new.data_fine <= df and new.data_fine >= di) or
   (new.data_inizio <= di and new.data_fine >= df) ) then
           raise notice 'hai inserito una militanza durante
           il periodo di un altra militanza';
           return null;
    end if;
    EXIT when not found;
    end loop;
    close curs1;
    if maxdf is null and new.data_inizio > maxdi then
           update militanza_calciatore set data_fine = new.data_inizio
           where data_fine is null and codicec = new.codicec;
    end if;
    return new;
end if;
select Max(data_fine),Max(data_inizio) into maxdf,maxdi
from (select data_fine,data_inizio
   from  militanza_portiere
   where codicec = new.codicec
   union
   select data_inizio,data_fine
   from  militanza_calciatore
   where codicec = new.codicec);
/* inserimento per la prima volta del calciatore*/
if maxdi is null then
       return new;
```

```
end if;
/* inserimento per la seconda o successiva volta del calciatore*/
open curs1 for
select data_inizio,data_fine
from  militanza_portiere
where codicec = new.codicec
union
select data_inizio,data_fine
from
militanza_calciatore where codicec = new.codicec;
loop
EXIT when not found;
fetch curs1 into di,df;
if ((new.data_inizio <= df and new.data_inizio >= di) or
(new.data_fine <= df and new.data_fine >= di) or
(new.data_inizio <= di and new.data_fine >= df) ) then
       raise notice 'hai inserito una militanza durante
       il periodo di un altra militanza';
       return null;
end if;
end loop;
close curs1;
select data_fine into df
from militanza_calciatore
where codicec = new.codicec and data_inizio = maxdi;
if df is null and new.data_inizio > maxdi then
   update militanza_calciatore
   set data_fine = new.data_inizio
   where data_fine is null and codicec = new.codicec;
end if;
select data_fine into df
from militanza_portiere
where codicec = new.codicec and data_inizio = maxdi;
if df is null and new.data_inizio > maxdi then
   update militanza_portiere
   set data_fine = new.data_inizio
   where data_fine is null and codicec = new.codicec;
```

```
 end if;
 return new;
 end;
 $$ language plpgsql
```

Trigger6:

Trigger Analogo al 5 ma dalla parte di militanza portiere.

```
CREATE OR REPLACE TRIGGER calcpor
BEFORE INSERT
ON public.militanza_portiere
FOR EACH ROW
EXECUTE FUNCTION public.inserimento_portiere();

CREATE OR REPLACE FUNCTION inserimento_portiere()
RETURNS trigger AS $$
DECLARE
curs1 refcursor;
df date;
di date;
maxdf date;
maxdi date;
BEGIN
/*controlliamo per prima cosa se il portiere ha giocato
anche da calciatore nella stessa squadra*/
if (new.codicec = (select codicec from  militanza_calciatore
where codicec = new.codicec and codices = new.codices limit 1)) then
    open curs1 for
    select data_inizio,data_fine
    from militanza_calciatore
    where codicec = new.codicec and codices = new.codices;
    loop
    fetch curs1 into di,df;
    if (new.data_inizio = di and new.data_fine = df ) then
        return new;
    end if;
    EXIT when not found;
```

```
    end loop;
    close curs1;
    open curs1 for
    select data_inizio,data_fine
    from militanza_portiere
    where codicec = new.codicec
    union
    select data_inizio,data_fine
    from  militanza_calciatore
    where codicec = new.codicec;
    loop
    fetch curs1 into di,df;
    if ((new.data_inizio <= df and new.data_inizio >=di) or
    (new.data_fine <= df and new.data_fine >=di) or
    (new.data_inizio <= di and new.data_fine >=  df)) then
        raise notice 'hai inserito una militanza durante
        il periodo di un altra militanza';
        return null;
    end if;
    EXIT when not found;
    end loop;
    close curs1;
    if maxdf is null and new.data_inizio > maxdi then
    update militanza_portiere
    set data_fine = new.data_inizio
    where data_fine is null and codicec = new.codicec;
    end if;
    return new;
end if;
select Max(data_fine),Max(data_inizio) into maxdf,maxdi
from (select data_fine,data_inizio
from militanza_portiere
where codicec = new.codicec
union
select data_inizio,data_fine
from  militanza_calciatore
where codicec = new.codicec);
```

```
/* inserimento per la prima volta del portiere*/
if maxdi is null  then
    return new;
end if;
/* inserimento per la seconda o successiva volta del portiere*/
open curs1 for
select data_inizio,data_fine
from  militanza_portiere
where codicec = new.codicec
union
select data_inizio,data_fine
from  militanza_calciatore
where codicec = new.codicec;
loop
EXIT when not found;
fetch curs1 into di,df;
if ((new.data_inizio <= df and new.data_inizio >=di) or
(new.data_fine <= df and new.data_fine >=di) or
(new.data_inizio <= di and new.data_fine >= df) ) then
    raise notice 'hai inserito una militanza durante
    il periodo di un altra militanza';
    return null;
end if;
end loop;
close curs1;
select data_fine into df
from militanza_calciatore
where codicec = new.codicec and data_inizio = maxdi;
if df is null and new.data_inizio > maxdi then
    update militanza_calciatore
    set data_fine = new.data_inizio
    where data_fine is null and codicec = new.codicec;
end if;
select data_fine into df
from militanza_portiere
where codicec = new.codicec and data_inizio = maxdi;
if df is null and new.data_inizio > maxdi then
```

```
    update militanza_portiere
    set data_fine = new.data_inizio
    where data_fine is null and codicec = new.codicec;
end if;
return new;
end;
$$ language plpgsql
```

Trigger7: Il seguente trigger si assicurerà che i portieri maschi vengano inseriti nelle squadre maschili e le femmine nelle squadre femminili.

```
CREATE OR REPLACE TRIGGER distinzione_portieri
BEFORE INSERT
ON public.militanza_portiere
FOR EACH ROW
EXECUTE FUNCTION public.inserimento_portieri_genere();

CREATE OR REPLACE FUNCTION inserimento_portieri_genere()
RETURNS trigger AS $$

DECLARE
persona_sesso varchar(15);
squadra_genere varchar(15);
BEGIN
select sesso into persona_sesso
from calciatore
where codicec = new.codicec;
select genere into squadra_genere
from squadra
where codices = new.codices;
if (persona_sesso = 'Maschio' and squadra_genere = 'Maschile') OR
(persona_sesso = 'Femmina' and squadra_genere = 'Femminile') then
    return new;
else
    raise notice 'non puoi inserire un giocatore % in una squadra
    %',persona_sesso,squadra_genere;
```

```
    return null;
end if;
end;
$$ language plpgsql
```

Trigger8:
Il seguente trigger farà in modo che in militanza portiere vengano inseriti solo giocatori che hanno giocato in porta;

```
CREATE OR REPLACE TRIGGER onlyport
BEFORE INSERT
ON public.militanza_portiere
FOR EACH ROW
EXECUTE FUNCTION public.onlyporf();

CREATE OR REPLACE FUNCTION onlyporf()
RETURNS trigger AS $$
BEGIN
if('portiere' in (select ruolo from ricopre r
where r.codicec = new.codicec)) then
    return new;
end if;
raise notice 'il giocatore inserito non ha mai giocato come portiere';
return null;
end;
$$ language plpgsql
```

Trigger9:
Il seguente trigger farà in modo che la squadra partecipi nelle competizione che la accettano, ad esempio la champions accetterà solo le squadre provenienti da certe nazioni, mentre le competizioni per le nazionali accetterà solo le nazionali.

```
CREATE OR REPLACE TRIGGER associazione_competizione
BEFORE INSERT
ON public.stagione_squadra
FOR EACH ROW
EXECUTE FUNCTION public.inserimento_stagione_squadra();
```

```
CREATE OR REPLACE FUNCTION inserimento_stagione_squadra()
RETURNS trigger AS $$
DECLARE
curs1 refcursor;
competizione_genere varchar(15);
squadra_genere varchar(15);
nazionalità_squadra varchar (15);
nazionalità_competizione varchar (15);
tipocomp varchar(15);
tiposquad varchar(15);
BEGIN
select genere into competizione_genere
from competizione
where nomec = new.nomec;

select genere into squadra_genere
from squadra
where codices = new.codices;

select nazionalità into nazionalità_squadra
from squadra
where codices = new.codices;

select tipo_squadra into tiposquad
from squadra
where codices = new.codices;

select tipo_competizione into tipocomp
from competizione
where nomec = new.nomec;
if competizione_genere = squadra_genere then
    if tipocomp = tiposquad then
        open curs1 for
        select nomen
        from competizione c join accetta a  on c.nomec = a.nomec
        where c.nomec = new.nomec;
```

```
        LOOP
        fetch curs1 into nazionalità_competizione;
        if(nazionalità_squadra = nazionalità_competizione or
        nazionalità_competizione is null ) then
            return new;
        end if;
        EXIT when not found;
        end loop;
        close curs1;
    end if;
end if;
raise notice 'inserimento errato la squadra non può
partecipare a questa competizione';
return null;
end;
$$ language plpgsql
```

Trigger10:
Il seguente trigger controlla che la data di vittoria del trofeo sia maggiore dell'inizio della sua carriera.

```
CREATE OR REPLACE TRIGGER dataminwin
BEFORE INSERT
ON public.vince_giocatore
FOR EACH ROW
EXECUTE FUNCTION public.dataminwinf();

CREATE OR REPLACE FUNCTION dataminwinf()
RETURNS trigger AS $$

BEGIN
if (new.data_vittoria >= (select MIN(data_inizio)
                          from calciatore c
                          join militanza_calciatore mc on
                          c.codicec = mc.codicec
                          where c.codicec = new.codicec))
then
```

```
    return new;


elsif (new.data_vittoria >= (select MIN(data_inizio)
                             from calciatore c
                             join militanza_portiere mc on
                             c.codicec = mc.codicec
                             where c.codicec = new.codicec))
then
    return new;
end if;
raise notice 'data_vittoria non valida';
return null;
end;
$$ language plpgsql
```

Trigger11:
Il seguente trigger controllerà che il genere del trofeo e il sesso del calciatore coincidano

```
CREATE OR REPLACE TRIGGER trofeigenere
BEFORE INSERT
ON public.vince_giocatore
FOR EACH ROW
EXECUTE FUNCTION public.trofeigeneref();

CREATE OR REPLACE FUNCTION trofeigeneref()
RETURNS trigger AS $$
BEGIN
if('Femmina'= (select sesso
               from calciatore
               where codicec=new.codicec ) AND
 'Femminile' = (select genere
                from trofeo
```

```
                    where nomet = new.nomet ))
    then
        return new;
    end if;

    if('Maschio'= (select sesso
                   from calciatore
                   where codicec=new.codicec ) AND
       'Maschile' = (select genere
                      from trofeo
                      where nomet = new.nomet ))
    then
        return new;
    end if;
    return null;
    end;
    $$ language plpgsql;
```

Trigger12:
Il seguente trigger controllerà che il trofeo assegnato al giocatore sia un trofeo individuale

```
CREATE OR REPLACE TRIGGER trofeogiocatore
BEFORE INSERT ON vince_giocatore
for each row
execute function trofeogiocatoref();

CREATE OR REPLACE FUNCTION trofeogiocatoref()
RETURNS TRIGGER AS $$
BEGIN
if ('individuale' = (select tipo from trofeo where nomet = new.nomet)) then
    return new;
end if;
return null;
end;
$$ language plpgsql;
```

Trigger13:
Il seguente trigger controllerà che il trofeo assegnato alla squadra sia un trofeo di squadra ed inoltre controllerà che abbia effettivamente partecipato a quella competizione nel periodo della data vittoria, in questa manierà scagionerà anche eventuali errori come l'inserimento della vittoria della Coppa del Mondo al Napoli per esempio perchè il Napoli non potendo mai partecipare non avrà data da controllare e non verrà inserito il valore.

```
CREATE OR REPLACE TRIGGER trofeisquadra
BEFORE INSERT ON vince_squadra
for each row
execute function trofeisquadraf();


CREATE OR REPLACE FUNCTION trofeisquadraf() RETURNS TRIGGER AS $$
DECLARE
curs refcursor;
datai date;
dataf date;
BEGIN
if ('squadra' = (select tipo from trofeo where nomet = new.nomet)) then
    open curs for
    SELECT inizio_partecipazione, fine_partecipazione
    FROM stagione_squadra
    WHERE codices = NEW.codices AND nomec = (select nomec
                                             from assegna
                                             where nomet = new.nomet);
    loop
    fetch curs into datai,dataf;
    if dataf is null then
        if new.data_vittoria >= datai then
            CLOSE curs;
            return new;
        end if;
    else
        if new.data_vittoria >= datai AND new.data_vittoria <= dataf then
```

```
            CLOSE curs;
            return new;
        end if;
    end if;
    EXIT WHEN NOT FOUND;
    end loop;
    close curs;
    return null;
end if;
end;
$$ language plpgsql;
```