

# **Gestione squadra professionistica**

## **Gruppo N° 12**

<b>Descrizione</b>	<b>2</b>
<b>Glossario</b>	<b>4</b>
<b>Schema Scheletro</b>	<b>6</b>
<b>Progetto Concettuale</b>	<b>7</b>
<b>Schema ER completo</b>	<b>13</b>
<b>Progetto Logico</b>	<b>14</b>
Eliminazione gerarchie ISA	14
Selezione chiavi primarie ed eliminazione chiavi esterne	16
Trasformazione attributi composti o multipli	18
Traduzione di entità e associazioni in schemi di relazione	19
Normalizzazione	26
Dato derivato	26
<b>Vincoli aggiuntivi</b>	<b>28</b>
<b>SQL</b>	<b>29</b>
Query di Creazione	29
Stored Procedure & Trigger	34
Query di inserimento	41
Query di Interrogazione	41
Query di Modifica	43
Query di Eliminazione	44

# Descrizione

Il progetto consiste nella creazione di un Database per la gestione di una squadra professionistica di basket.

Il database permette di gestire tutti gli aspetti di una squadra sportiva a partire dai giocatori e il proprio campionato fino ad aspetti più tecnici come i contratti, il mercato e gli infortuni dei propri giocatori. A noi interessa principalmente la gestione della squadra "Golden State Warriors".

La **squadra** è identificata dal nome (univoco) e come attributi aggiuntivi ha la città di appartenenza, i numeri di maglia ritirati e i campionati vinti.

Il **campionato** a cui la squadra partecipa è riconosciuto dalla stagione e dalla serie (possono essere fatte coppe e campionato contemporaneamente). Viene memorizzato anche il numero di squadre partecipanti e il numero di partite per squadra.

Le **persone** sono identificate da un codice fiscale, dal nome e cognome, dall'età e dalla nazionalità. In una squadra vi lavorano almeno 20 persone. Una persona può lavorare o meno per una squadra (es. un giocatore ritirato non ci lavora più). Fanno parte della gerarchia persone che è totale ed esclusiva, dato che ci interessano solo le persone che hanno rapporti con la squadra e una persona può svolgere solo un ruolo nella squadra.

L'**allenatore** oltre che dai classici dati anagrafici è identificato dal record personale di vittorie/sconfitte.

I **preparatori** sono vari all'interno di una squadra e per ognuno di essi deve essere specificato il tipo(atletico,tattico,ecc...) e il numero di tesserino, oltre che alla tessera del capo nel caso fosse un vice.

Il **giocatore**, oltre che dai dati anagrafici, è identificato dal n° di maglia, che all'interno della squadra è univoco. Ne è inoltre riportata l'altezza, il ruolo e lo status, il quale influisce nel contratto con relativi bonus e con il merchandising.

L'**allenamento** è preparato appunto da un preparatore, in una specific data, un preparatore può organizzare solamente un allenamento in quella data. L'allenamento è descritto dal tipo, che può essere atletico, video, tattico, ecc...

Le persone sono legate alla squadra da un **contratto**, in cui viene memorizzato lo stipendio, eventuali bonus l'anno della firma e la durata. Nel periodo che va dall'anno della firma fino alla scadenza lo stesso atleta non può firmare ulteriori contratti con la squadra.

Il **Budget** è identificato dalla stagione e dalla squadra a cui appartiene, di esso vengono memorizzati vari attributi come il budget massimo, quello disponibile per eventuali acquisti, il monte ingaggi e la presenza di eventuali sponsor a supporto. Il budget è disposto dalla squadra e viene influenzato dai contratti di quella stagione.

I giocatori possono subire degli **infortuni**, che vengono identificati dalla data di inizio, dalla descrizione ed è legato al giocatore (dato che un giocatore non può subire lo stesso tipo di infortunio nello stesso periodo). Di esso ne viene riportata la gravità e la data di rientro.

Lo **stadio** ha un nome, che è univoco, il numero di posti a sedere e il costo dei biglietti (si intende unico per tutti i posti).

Di una **partita** viene memorizzato il suo codice univoco oltre che la data, quest'ultima è legata allo stadio, infatti non vi possono essere 2 o più partite in una certa data nello stesso stadio. Viene anche riportato l'avversario incontrato, l'esito (Vittoria o sconfitta) e il risultato dell'incontro. Per essere valida vi devono partecipare almeno 5 giocatori ed un massimo di 12.

Tra l'associazione di partita e stadio si vuole conoscere il numero di biglietti venduti.

Per un giocatore, in base alla sua popolarità, viene prodotto del **merchandising** che i tifosi possono comprare. Viene identificato da un codice, unito al giocatore possessore, di tale merce si conosce la quantità e la quantità di riordino che è influenzata dallo status del giocatore.

Un Allenatore può decidere di effettuare dei **Trasferimenti**, descritti da un Codice univoco, un valore e una squadra di destinazione. Questi trasferimenti si suddividono in **Prestito**, a cui si aggiunge la durata, e la **Cessione**, a cui si aggiunge una eventuale contropartita.

# Glossario

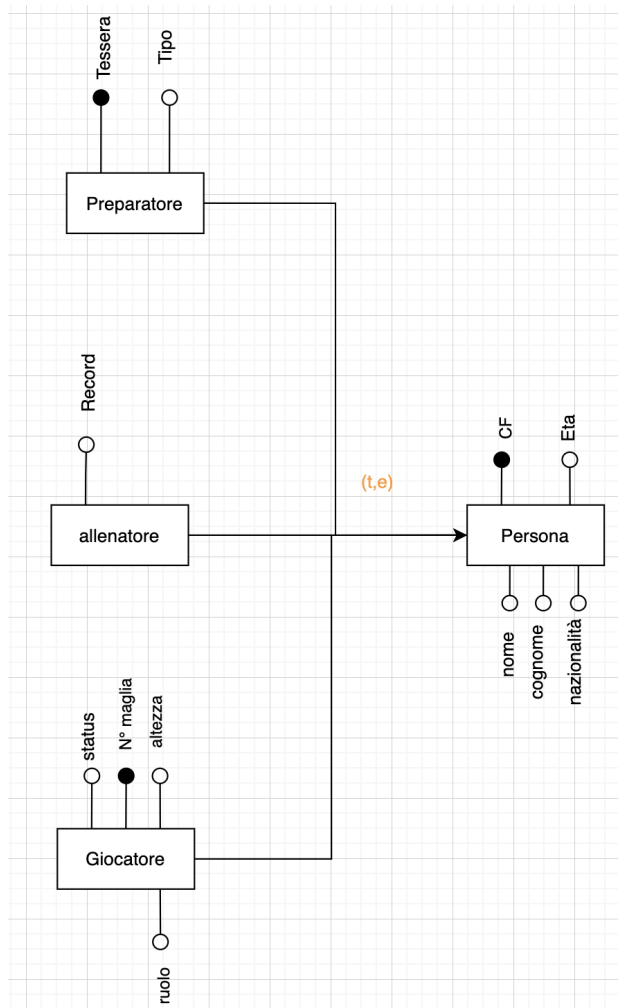
Nominativo	Descrizione	Sinonimi	Legami
Squadra	<ul style="list-style-type: none"> <li>• nome</li> <li>• città</li> <li>• campionati vinti</li> <li>• numero maglia ritirate (multipla)</li> </ul>	<ul style="list-style-type: none"> <li>• team</li> <li>• club</li> </ul>	<ul style="list-style-type: none"> <li>• budget</li> <li>• persona</li> <li>• campionato</li> </ul>
Persona	<ul style="list-style-type: none"> <li>• nome</li> <li>• cognome</li> <li>• CF</li> <li>• nazionalità</li> <li>• età</li> </ul>		<ul style="list-style-type: none"> <li>• contratto</li> <li>• squadra</li> <li>• giocatore</li> <li>• preparatore</li> <li>• allenatore</li> </ul>
Giocatore	<ul style="list-style-type: none"> <li>• altezza</li> <li>• CF</li> <li>• ruolo</li> <li>• status</li> <li>• numero maglia</li> </ul>	<ul style="list-style-type: none"> <li>• Atleta</li> </ul>	<ul style="list-style-type: none"> <li>• persona</li> <li>• infortunio</li> <li>• partita</li> <li>• merchandising</li> <li>• trasferimento</li> </ul>
Preparatori	<ul style="list-style-type: none"> <li>• tessera</li> <li>• tipo</li> <li>• CF</li> <li>• capo</li> </ul>	<ul style="list-style-type: none"> <li>• Organico</li> </ul>	<ul style="list-style-type: none"> <li>• persona</li> <li>• allenamento</li> </ul>
Allenatore	<ul style="list-style-type: none"> <li>• record</li> <li>• CF</li> </ul>	<ul style="list-style-type: none"> <li>• Coach</li> </ul>	<ul style="list-style-type: none"> <li>• persona</li> <li>• trasferimento</li> </ul>
Merchandising	<ul style="list-style-type: none"> <li>• QT riordino</li> <li>• QT</li> <li>• codice</li> <li>• CF</li> </ul>		<ul style="list-style-type: none"> <li>• giocatore</li> </ul>
Stadio	<ul style="list-style-type: none"> <li>• nome</li> <li>• num posti</li> <li>• costo biglietto (calcolato)</li> </ul>		<ul style="list-style-type: none"> <li>• partita</li> </ul>
Campionato	<ul style="list-style-type: none"> <li>• stagione</li> <li>• serie</li> <li>• numero di squadre</li> <li>• numero partite della squadra</li> </ul>	<ul style="list-style-type: none"> <li>• lega</li> </ul>	<ul style="list-style-type: none"> <li>• squadra</li> </ul>

Allenamento	<ul style="list-style-type: none"> <li>• tipo</li> <li>• data</li> </ul>		<ul style="list-style-type: none"> <li>• preparatori</li> </ul>
Infortunio	<ul style="list-style-type: none"> <li>• data inizio</li> <li>• data fine</li> <li>• gravità</li> <li>• descrizione</li> <li>• CF</li> </ul>		<ul style="list-style-type: none"> <li>• giocatore</li> </ul>
Contratto	<ul style="list-style-type: none"> <li>• CF</li> <li>• bonus</li> <li>• stipendio</li> <li>• anno firma</li> <li>• durata</li> </ul>		<ul style="list-style-type: none"> <li>• persona</li> <li>• budget</li> </ul>
Partita	<ul style="list-style-type: none"> <li>• codice partita</li> <li>• data</li> <li>• avversario</li> <li>• casa/trasfert a</li> <li>• risultato</li> <li>• esito</li> </ul>	<ul style="list-style-type: none"> <li>• match</li> </ul>	<ul style="list-style-type: none"> <li>• stadio</li> <li>• giocatore</li> </ul>
Budget	<ul style="list-style-type: none"> <li>• stagione</li> <li>• massimale</li> <li>• disponibile</li> <li>• monte ingaggi</li> <li>• sponsor</li> </ul>	<ul style="list-style-type: none"> <li>• liquidità</li> </ul>	<ul style="list-style-type: none"> <li>• contratto</li> <li>• squadra</li> </ul>
Trasferimento	<ul style="list-style-type: none"> <li>• cod. trasf</li> <li>• valore</li> <li>• squadra</li> <li>• destinazione</li> <li>• tipologia</li> </ul>	<ul style="list-style-type: none"> <li>• mercato</li> </ul>	<ul style="list-style-type: none"> <li>• allenatore</li> <li>• giocatore</li> </ul>
Prestito	<ul style="list-style-type: none"> <li>• cod. trasf</li> <li>• durata</li> </ul>		<ul style="list-style-type: none"> <li>• trasferimento</li> </ul>
Cessione	<ul style="list-style-type: none"> <li>• cod. trasf</li> <li>• contropartita</li> </ul>		<ul style="list-style-type: none"> <li>• trasferimento</li> </ul>

+



# Progetto Concettuale

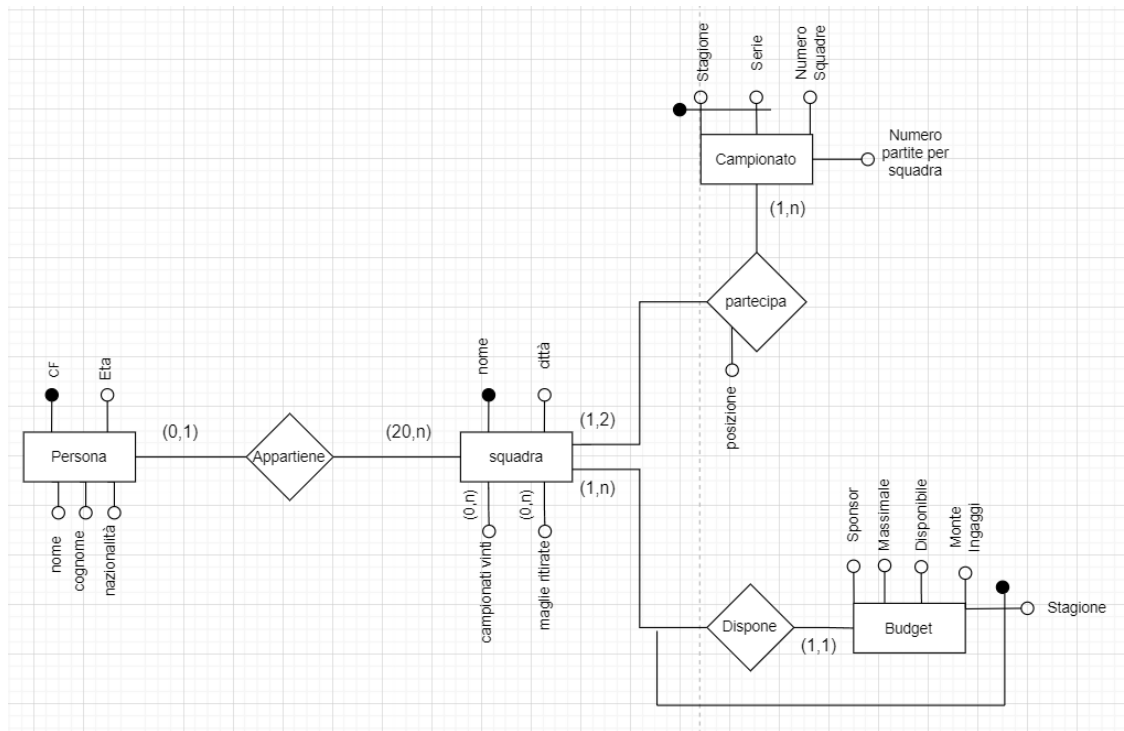


Una delle prime cose che abbiamo fatto è stata quella di modellare la gerarchia Persona. Infatti tutte le figure a fianco elencate hanno attributi di base molto simili, e quindi sembrava opportuno effettuare questa operazione.

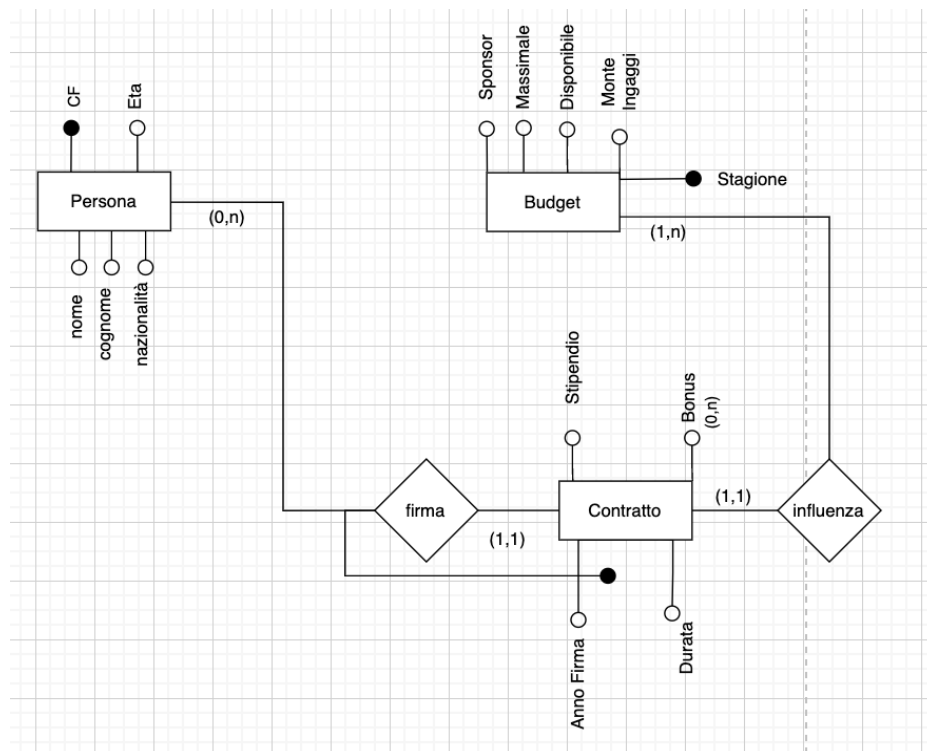
Ogni Persona all'interno della nostra squadra infatti è identificata da un Codice Fiscale, che ovviamente è univoco e i classici dati anagrafici.

Inoltre questa gerarchia è Totale, quindi ogni persona deve essere obbligatoriamente o un preparatore o un allenatore o un giocatore, ed esclusiva, quindi un preparatore non può essere anche un allenatore ecc...

In una squadra vi lavorano almeno 20 persone, ed essa partecipa al più a due campionati (Es: campionato nazionale e coppa). Inoltre possiede un budget per ogni stagione, maggiore è il budget maggiore sono i soldi che possono essere offerti ai giocatori.

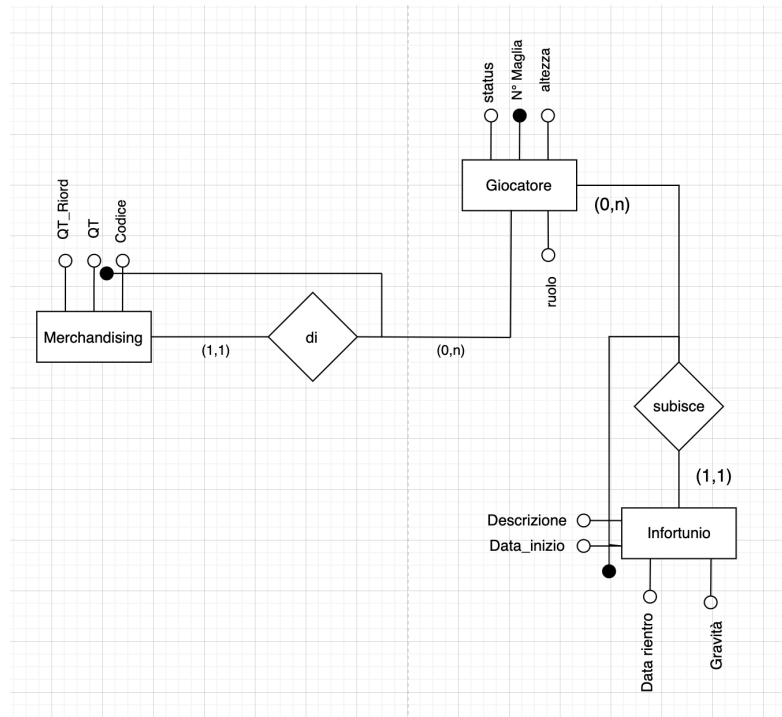


Come detto il Budget influenza i soldi che possono essere offerti ad un giocatore per la firma di un contratto. Una Persona può firmare 'N' contratti con la squadra ma può rinnovare solamente quando il contratto è in scadenza.

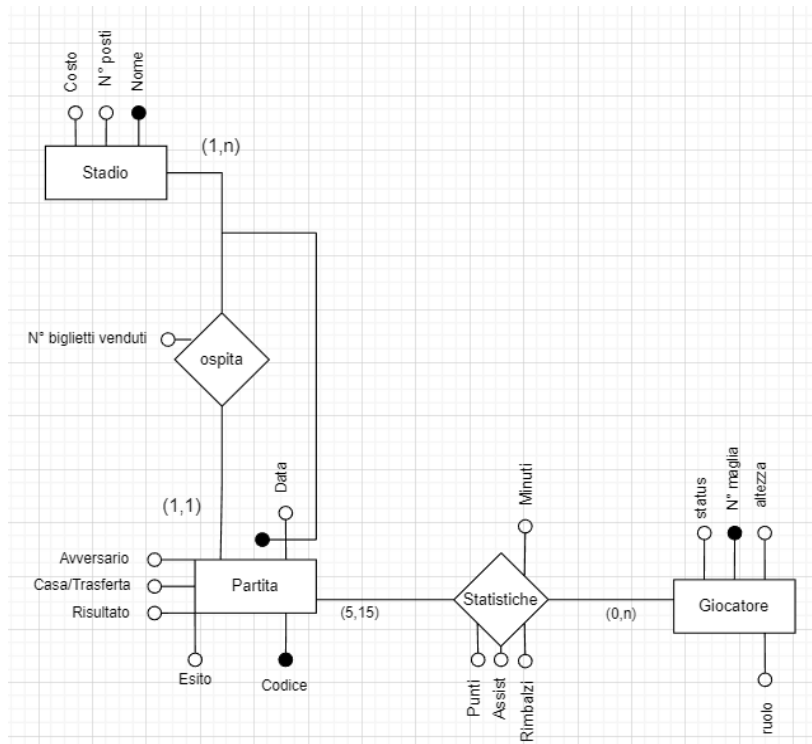




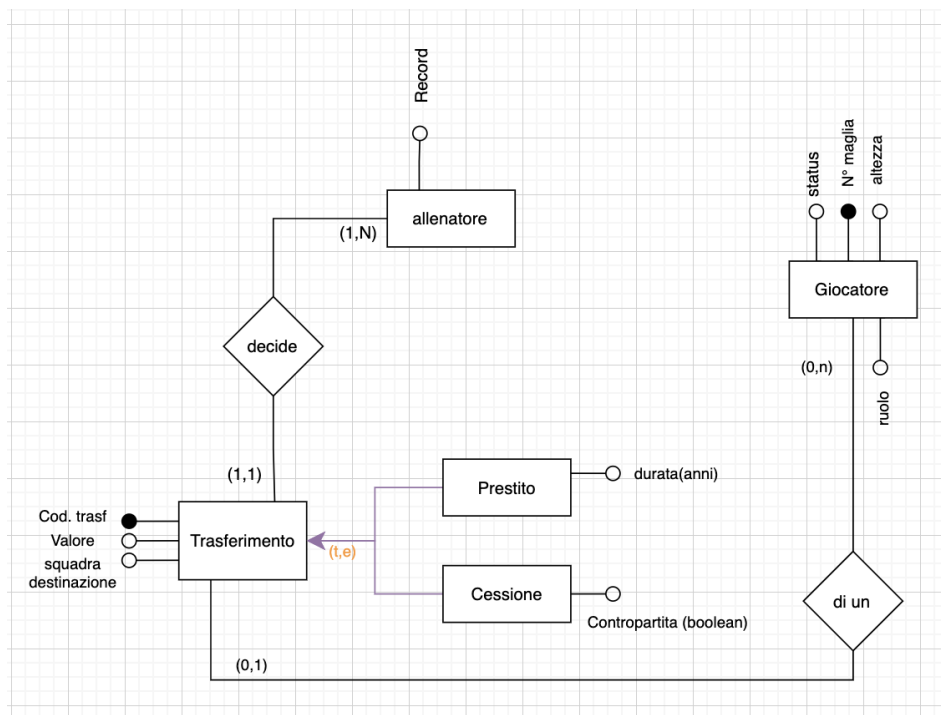
Durante la stagione vi possono essere degli infortuni e questi sono registrati tenendo nota della data di inizio e del tipo di infortunio, oltre che della sua gravità e della data di rientro prevista. Inoltre per un giocatore può essere prodotto del Merchandising generico, quando la quantità di merchandising è al di sotto di una certa quota chiamata QT\_r (quantità di riordino), viene fatto il riordino. La quantità di riordino è basata sullo status del giocatore, più lo status è importante più questa quantità sarà alta.



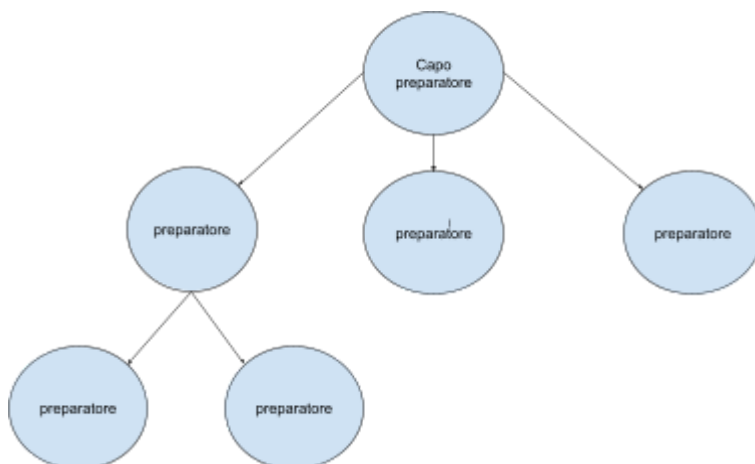
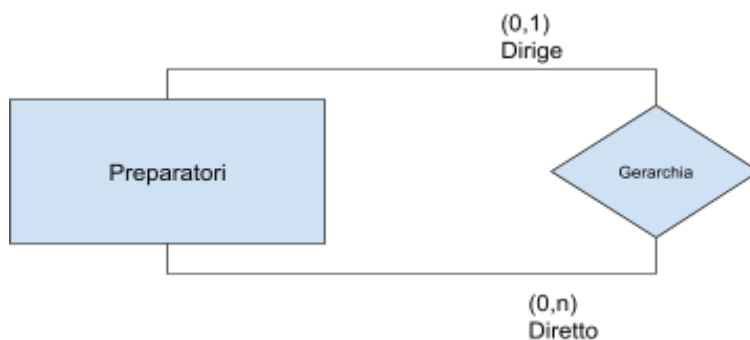
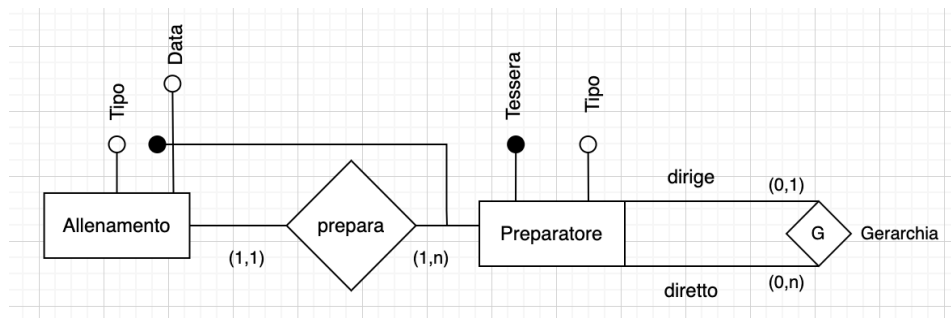
Uno stadio può ospitare “N” partite ma in una certa data ve ne può essere solamente una all’interno di esso. Inoltre, di uno stadio vengono riportati i numeri di posti e il costo del biglietto. Per i giocatori che partecipano ad una partita vengono mantenute le statistiche principali.



Abbiamo definito anche una nuova gerarchia per aiutarci nella gestione del “Mercato”. La nuova gerarchia Trasferimento è identificata da un Codice Trasferimento, che è univoco, e i suoi figli sono Prestito e Cessione i quali aggiungono qualche attributo specifico per ognuno di essi. La gerarchia è totale, infatti un trasferimento è o un prestito o una cessione, ed è esclusiva.



Un allenamento è preparato da un solo preparatore e quest'ultimo, in una certa data, ne può organizzare solamente uno. Con l'auto-associazione gestiamo i preparatori e la loro gerarchia. In particolare come radice abbiamo il capo preparatore che viene usato come identificativo per un determinato gruppo di preparatori per gli allenamenti.



Progetto realizzato da Luca Montanari e Davide Rivi



# Progetto Logico

il progetto logico relazionale serve per tradurre e trasformare lo schema E/R, per far sì che il DBMS sia in grado di operare. Queste operazioni in sviluppano in 5 fasi:

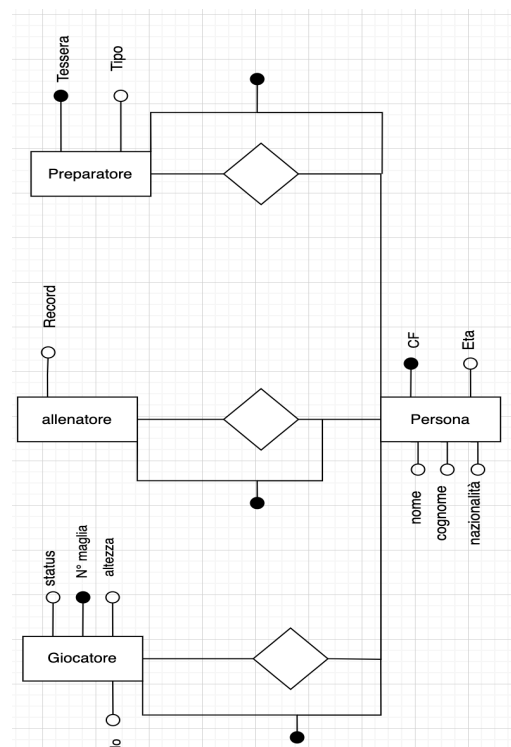
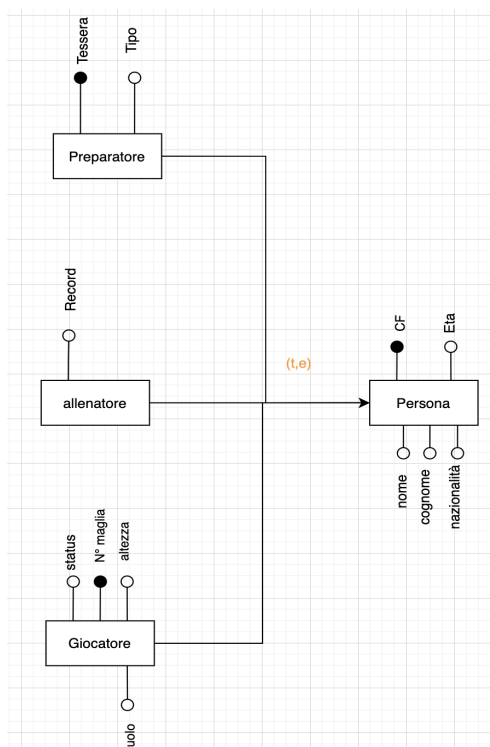
1. eliminazione gerarchie isa
2. selezione chiavi primarie ed eliminazione chiavi esterne
3. trasformazione attributi composti o multipli
4. traduzione di entità e associazioni in schemi di relazione
5. normalizzazione

## Eliminazione gerarchie ISA

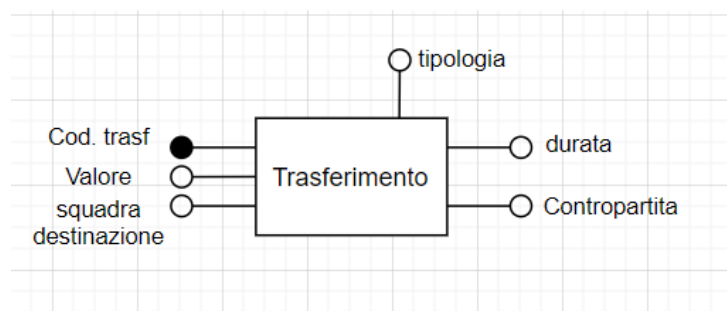
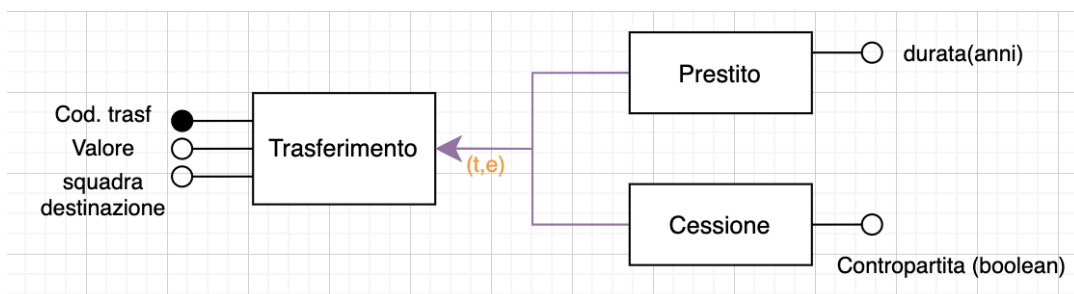
il modello relazionale non sopporta le gerarchie che vengono sostituite da entità e associazioni. Ci sono tre modi per gestire le gerarchie che sono: mantenimento delle entità con associazioni, collasso verso il basso, collasso verso l'alto.

Nel nostro database sono presenti 2 gerarchie da gestire, la gerarchia persona e quella trasferimento.

La gerarchia **persona** presenta l'entità padre e tre entità figlie con diverse associazioni collegate. Il modo migliore per gestire tale gerarchia è il **mantenimento delle entità** con associazioni perché anche se è totale ed esclusiva, effettuare un collasso verso il basso comporterebbe un numero alto di foreign keys all'entità collegate a persona, mentre un collasso verso l'alto darebbe all'entità persona molti attributi che sarebbero nulli.

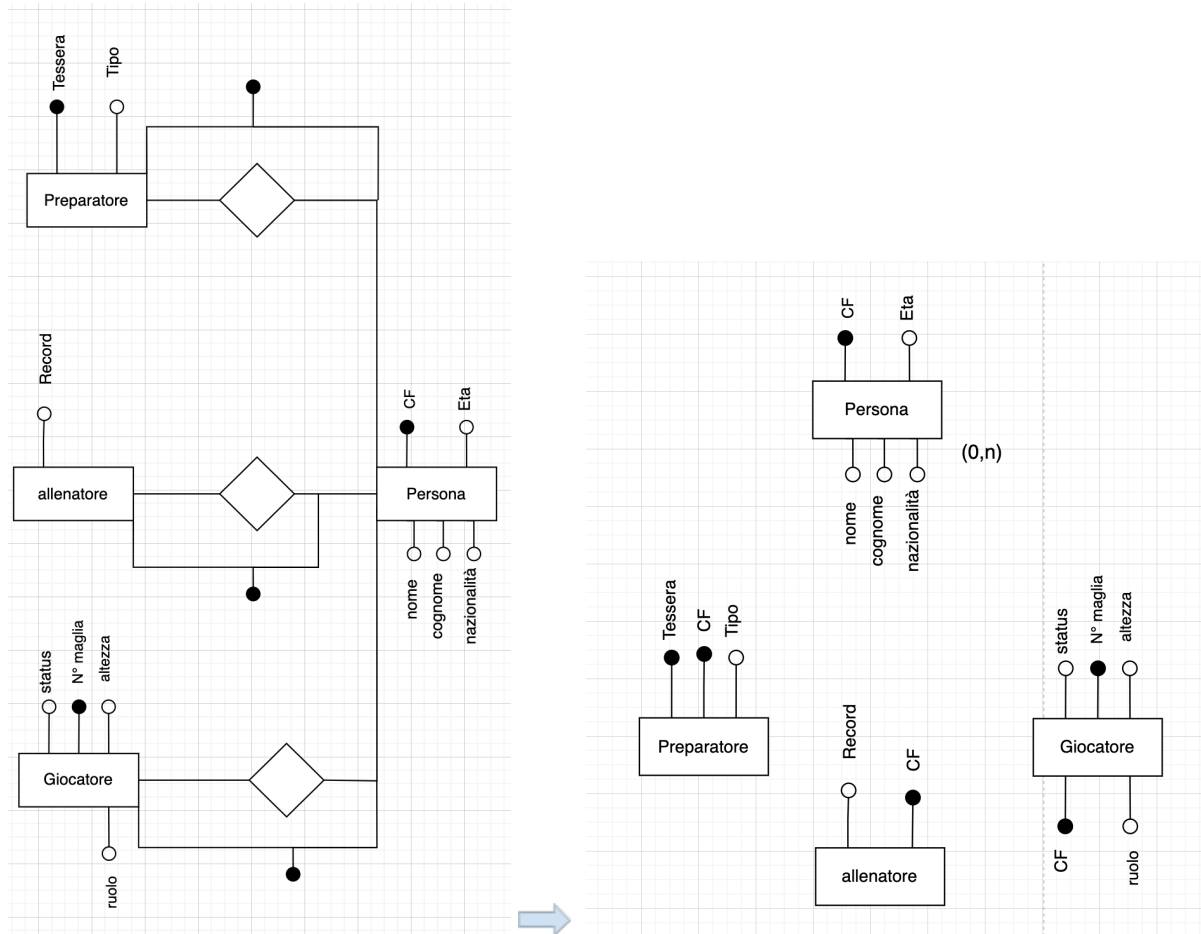


Mentre per la gerarchia **trasferimento**, è totale ed esclusiva, il collasso verso il basso non risulta efficace dato i pochi attributi dell'entità figlie e la mancanza di associazioni ad altre entità. Quindi il modo migliore di gestire questa gerarchia è con il **collasso verso l'alto** dove l'entità padre avrà anche gli attributi dell'entità figlie e in più l'attributo tipologia (selettore).

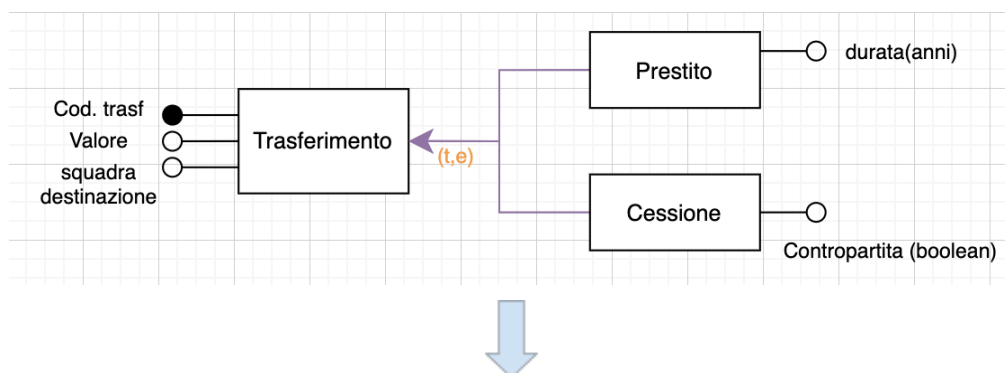


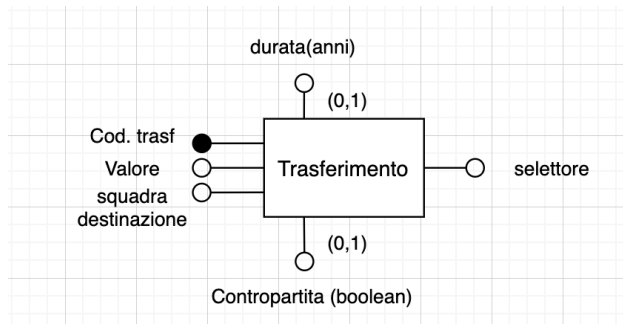
## Selezione chiavi primarie ed eliminazione chiavi esterne

Le entità Preparatore, Allenatore ed Atleta hanno una componente di identificazione esterna dall'entità persona e questo comporta le seguenti trasformazioni:

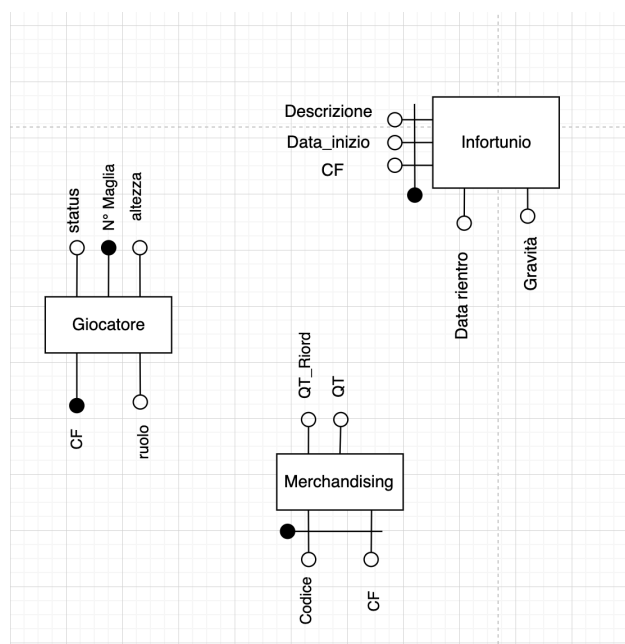
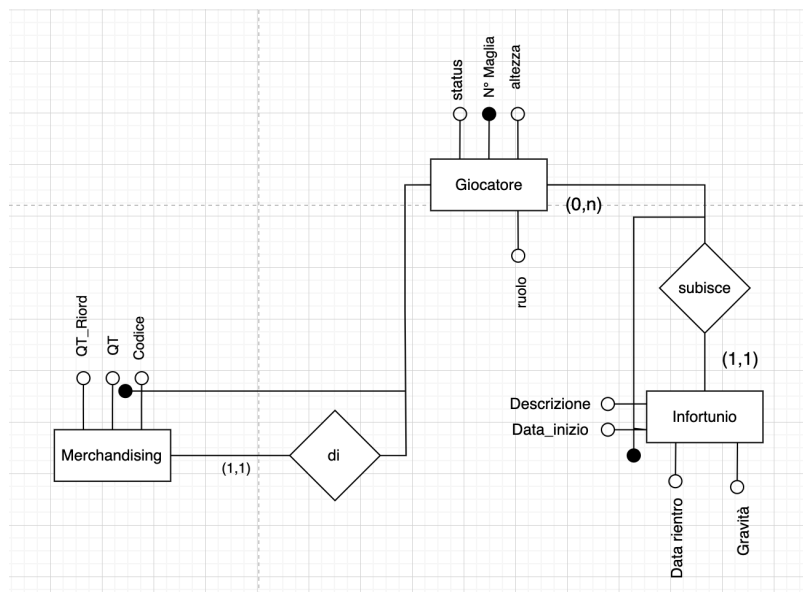


Per la nostra seconda gerarchia invece abbiamo scelto un altro approccio, in particolare abbiamo utilizzato il collasso verso l'alto. Questo perché le entità figlie non sono collegate tramite associazione a nessun'altra entità.



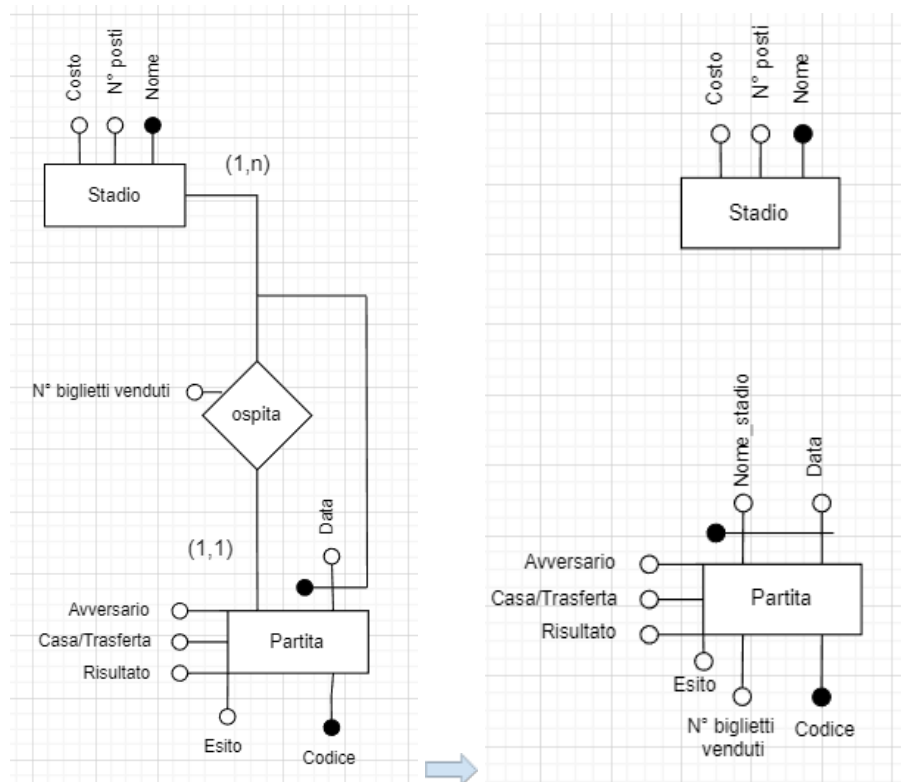


Le entità Infortunio e Merchandising hanno una componente esterna dell'entità Giocatore. Il risultato è il seguente:





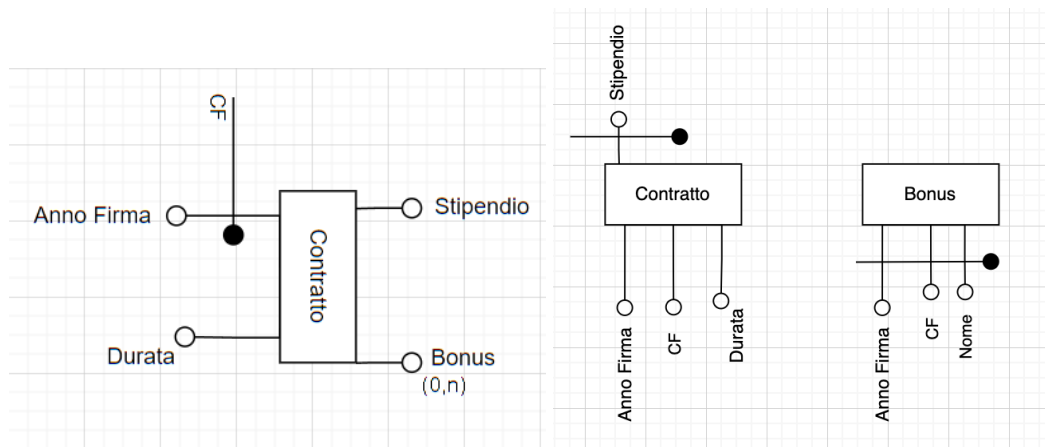
L'entità partita ha una componente esterna dell'entità Stadio. Il risultato è il seguente:



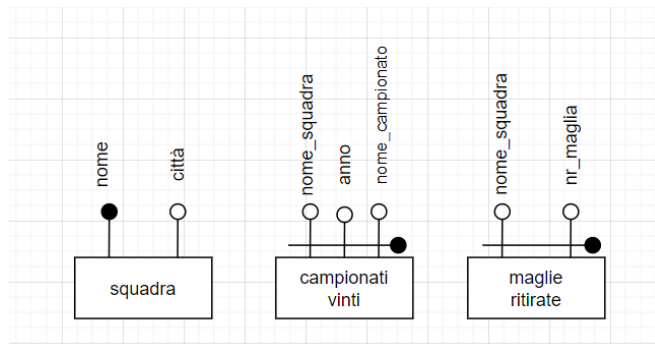
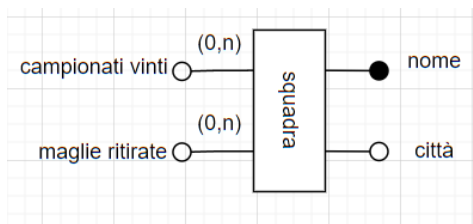
## Trasformazione attributi composti o multipli

Le relazioni possono avere solo attributi atomici, quindi vanno gestiti gli attributi composti/ripetuti. Nel nostro modello E/R sono presenti alcuni attributi ripetuti che diventeranno delle entità.

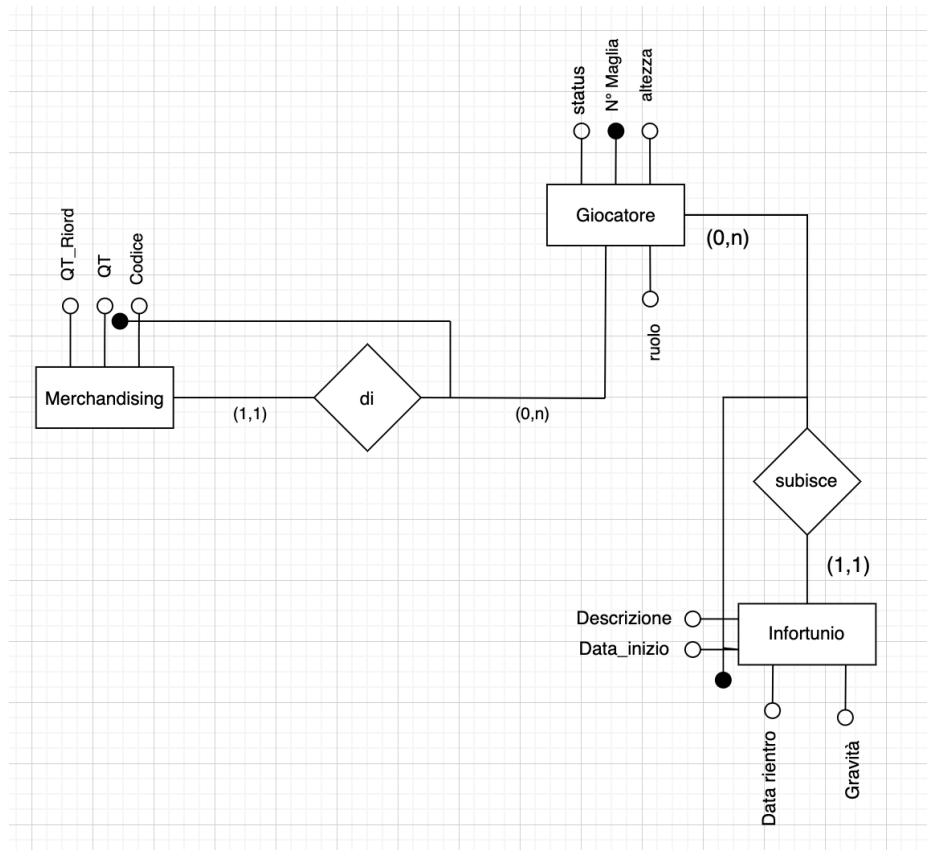
L'entità contratto presenta l'attributo ripetuto bonus, il quale diventa un'entità con chiave primaria composta dalla chiave di contratto più il nome.



L'entità squadra presenta 2 attributi ripetuti che diventano 2 relazioni con chiave primaria la chiave di squadra composta a, nel caso di campionati vinti, l'anno e il nome del campionato perché in un anno la squadra può vincere 2 campionati diversi, mentre l'attributo maglie ritirate con il numero di maglia essendo univoco nella squadra.



## Traduzione di entità e associazioni in schemi di relazione



L'entità Merchandising si collega con un'associazione 1:N all'entità Giocatore, perciò è possibile accorpare l'associazione "di" con Merchandising. Stessa situazione per quanto riguarda Infortunio e Giocatore.

**Giocatore** (CF, N° Maglia, ruolo, status, altezza)

FK: CF references Persona

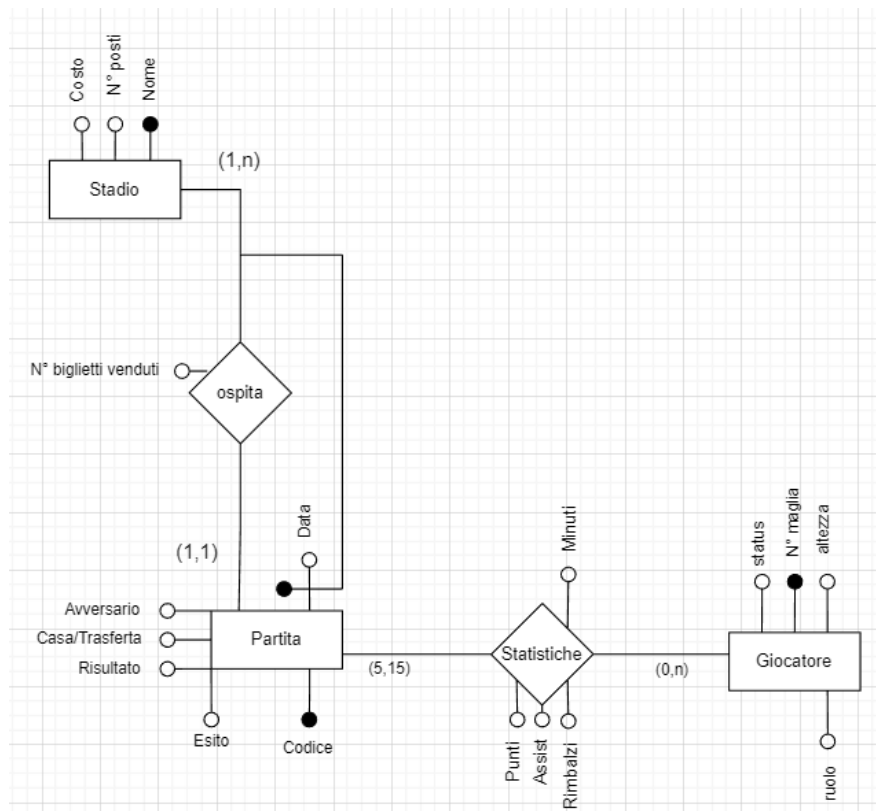
AK: N° Maglia

**Merchandising** (CF\_G, Cod, QT, QT\_R)

FK: CF\_G references Giocatore

**Infortunio** (CF\_G, Data\_Inizio, Descrizione, Data\_Rientro, Gravità)

FK: CF\_G references Giocatore



Giocatore si collega ancora con un'associazione N:M a Partita, per cui verranno fuori 2 relazioni per le entità e una per l'associazione. Partita poi si collega con un'associazione 1:N con l'entità Stadio, da cui ospita viene accorpata a Partita.

**Statistiche** (Cod\_Partita, CF\_G, punti, assist, rimbalzi, minuti)

FK: Cod\_Partita references Partita

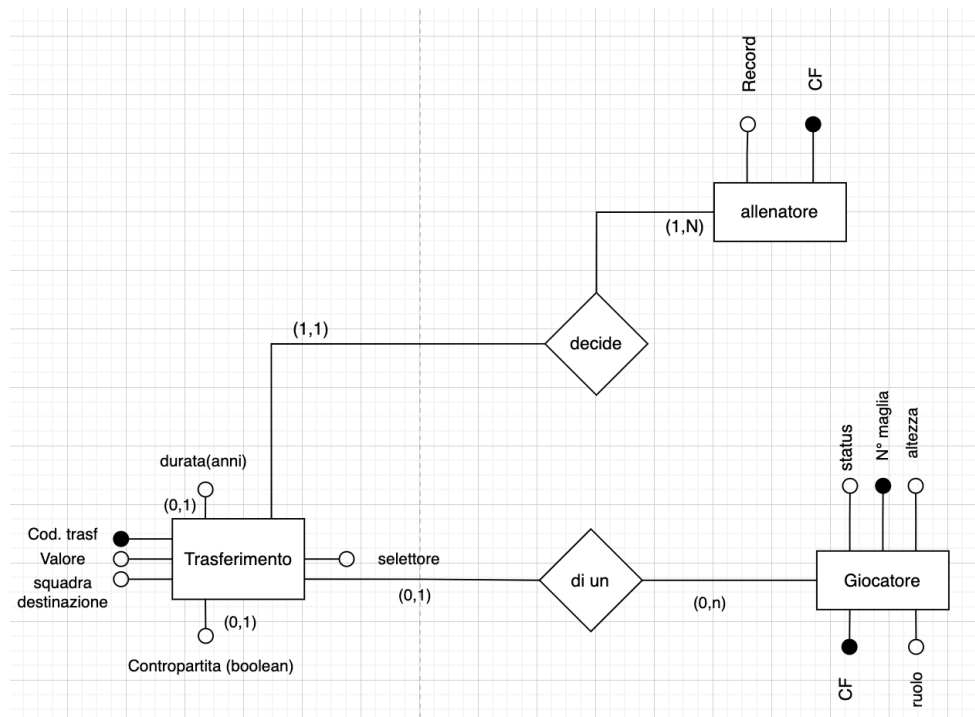
FK: CF\_G references Giocatore

**Stadio** (Nome, Costo, N° Posti)

**Partita** (Cod, Data, nome\_stadio, avversario, tipo, risultato, esito, biglietti\_venduti)

FK: nome\_stadio references Stadio

AK: Data, nome\_stadio



Infine Giocatore si collega anche con trasferimento tramite un'associazione 1:N, dove quindi "di un" verrà accorpato a trasferimento. E trasferimento è collegato ad Allenatore sempre con un'associazione 1:N che verrà anch'essa accorpata in trasferimento

**Trasferimento** (Cod. Trasf., Valore, Squadra\_Dest, CF\_G, CF\_A, Durata, Contropartita, Tipologia)

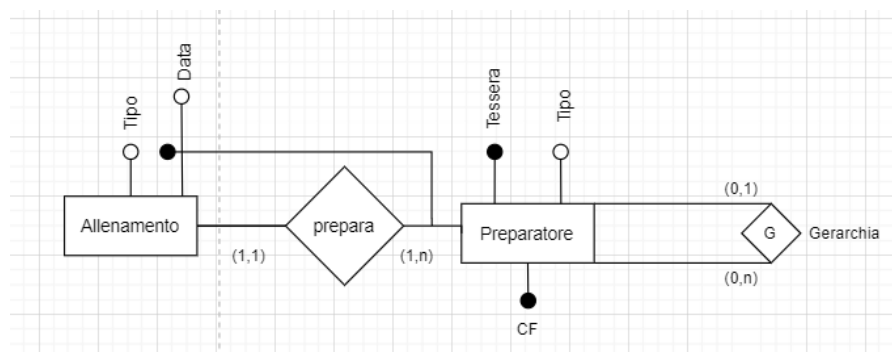
FK: CF\_G references Giocatore

FK: CF\_A references Allenatore

**Allenatore** (CF, record)

FK: CF references Persona

Preparatore e Allenamento sono collegati da un' associazione 1:N e quindi "prepara" viene accorpato ad Allenamento.



Progetto realizzato da Luca Montanari e Davide Rivi

**Preparatore** (CF, Tessera, tipo, tessera\_capo)

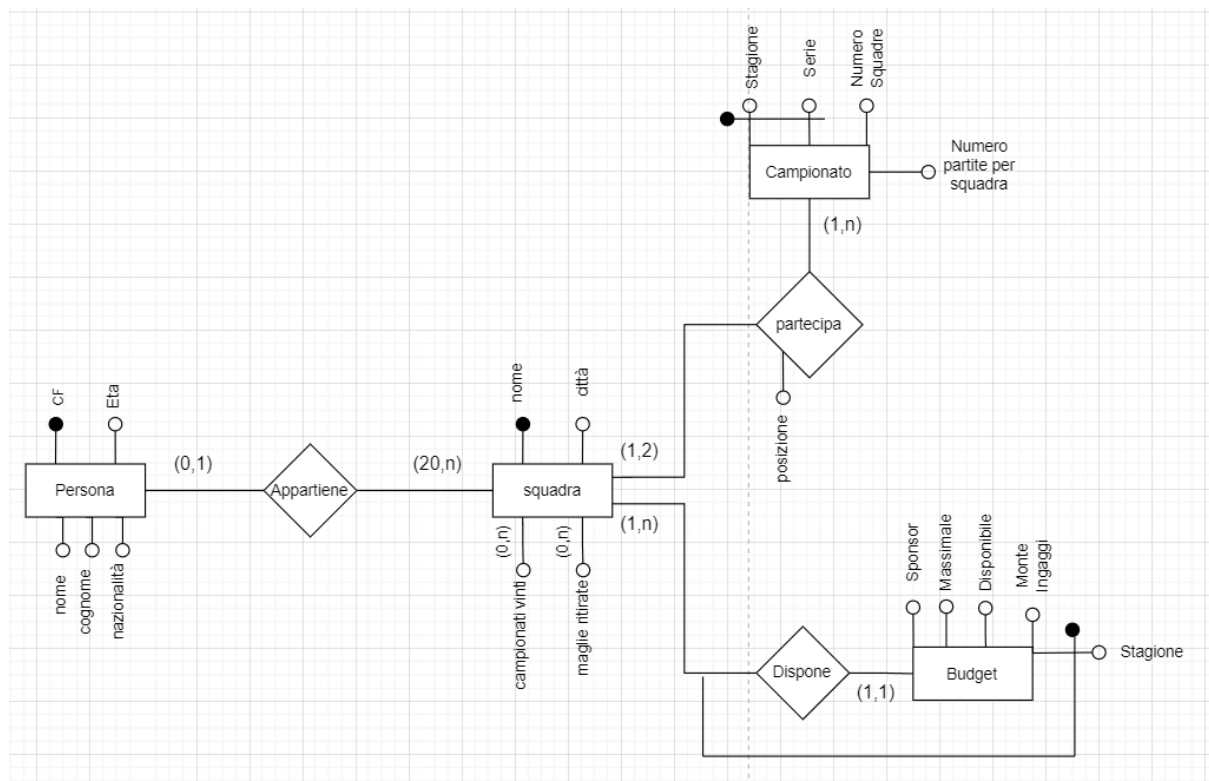
FK: CF references Persona

FK: tessera\_capo references preparatore

AK: Tessera

**Allenamento** (Data, Tessera, tipo)

FK: Tessera references Preparatore



Persona e squadra sono collegati da un'associazione 1:N e quindi ci saranno 3 relazioni in totale. Squadra poi è collegata 1:N con Budget, in questo caso abbiamo scelto di mantenere 2 entità per comodità e non collassare tutto in una unica. Sempre Squadra poi è collegata a Campionato da una relazione N:M, e quindi vengono anche qui mantenute 3 relazioni.

**Persona** (CF, età, nome, cognome, nazionalità, nome\_squadra)

FK: nome\_squadra references Squadra

**Squadra** (Nome, città)

**Partecipa** (Stagione, Serie, Nome Squadra, posizione)

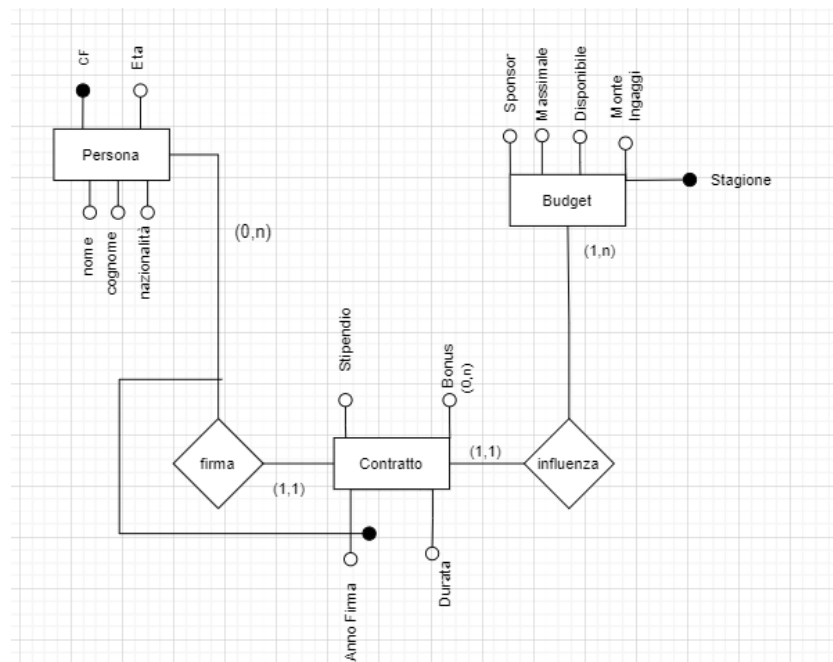
FK: Stagione, Serie references Campionato

FK: Nome\_Squadra references Squadra

**Budget** (Stagione, Nome\_Squadra, sponsor, massimale, disponibile, monte ingaggi)

**Campionato** (Stagione, Serie, numero squadre, numero partite per squadra)

Budget Persona sono poi collegate a Contratto da due associazioni 1:N. Quindi contratto assorbe le due associazioni “firma” e “influenza” così da far rimanere solo 3 relazioni.



**Contratto** (CF\_P, anno firma, durata, stipendio, Stagione\_B)

FK: CF\_P references Persona

FK stagione\_B references Budget

*Riassumendo:*

**Squadra** (Nome, città)

**Persona** (CF, nome, cognome, età, nazionalità, nome\_squadra)

FK: nome\_squadra references Squadra

**Allenatore** (CF, record)

FK: CF references Persona

**Preparatore** (CF, Tessera, tipo, tessara\_capo)

FK: CF references Persona

FK: tessera\_capo references preparatore

AK: Tessera

**Giocatore** (CF, N° Maglia, ruolo, status, altezza)

Progetto realizzato da Luca Montanari e Davide Rivi

FK: CF references Persona  
AK: N° Maglia

**Infortunio** (CF\_G, Data\_Inizio, Descrizione, Data\_Rientro, Gravità)  
FK: CF\_G references Giocatore

**Allenamento** (Data, Tessera, tipo)  
FK: Tessera references Preparatore

**Stadio** (Nome, Costo, N° Posti)

**Partita** (Cod, Data, nome\_stadio, avversario, tipo, risultato, esito, biglietti\_venduti)  
FK: nome\_stadio references Stadio  
AK: Data, nome\_stadio

**Statistiche** (Cod\_Partita, CF\_G, punti, assist, rimbalzi, minuti)  
FK: Cod\_Partita references Partita  
FK: CF\_G references Giocatore

**Budget** (Stagione, Nome\_Squadra, sponsor, massimale, disponibile, monte ingaggi,)  
FK Nome\_Squadra references Squadra

**Contratto** (CF\_P, anno\_firma, durata, stipendio, Stagione\_B)  
FK: CF\_P references Persona  
FK stagione\_B references Budget

**Bonus** (CF\_P, anno\_firma, nome)  
FK: CF\_P, anno\_firma references Contratto

**Campionato** (Stagione, Serie, numero squadre, numero partite per squadra)

**Partecipa** (Stagione, Serie, Nome\_Squadra, posizione)  
FK: Stagione, Serie references Campionato  
FK: Nome\_Squadra references Squadra

**Maglie ritirate** (Nome\_squadra, N° Maglia)  
FK: Nome\_squadra references Squadra

**Campionati Vinti** (Nome\_squadra, anno, nome\_campionato)  
FK: Nome\_squadra references Squadra

**Merchandising** (CF\_G, Cod, QT, QT\_R)  
FK: CF\_G references Giocatore

**Trasferimento** (Cod\_Trasf, Valore, Squadra\_Dest, CF\_G, CF\_A, Durata, Contropartita, Tipologia)  
FK: CF\_G references Giocatore

Progetto realizzato da Luca Montanari e Davide Rivi



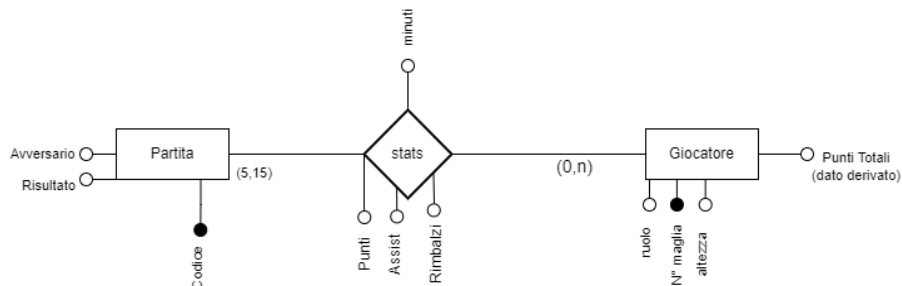
FK: CF\_A references Allenatore

## Normalizzazione

Dopo un controllo, abbiamo verificato che le relazioni sono in forma normale.

## Dato derivato

Il dato derivato è un dato che può essere ottenuto attraverso una serie di operazioni da altri dati. Sulla base delle operazioni e delle loro frequenze è possibile valutare se conviene averlo oppure no. Nel nostro database abbiamo considerato se utilizzare un dato derivato o meno per sapere i punti totali di un giocatore, durante una stagione.



Abbiamo ipotizzato le seguenti tabelle e operazioni:

### Tabella dei volumi

Concetto	Tipo	Volume Dati
Giocatore	E	15
Partita	E	100
Statistiche	R	1500

### Tabella delle Operazioni

**op.1:** inserimento punto in una partita.

**op.2:** visualizzazione punti totali in stagione.

Operazione	Tipo	Frequenza
Op. 1	I	60/Giorno
Op. 2	I	5/giorno

abbiamo svolto i calcoli per vedere se è utile o meno il dato derivato punti totali:

Progetto realizzato da Luca Montanari e Davide Rivi

Con dato derivato

Op. 1		
<b>Concetto</b>	<b>Accesso</b>	<b>Tipo</b>
Partita	1	L
Statistiche	1	L
	1	S
Giocatore	1	L
	1	S
		$3L * 60 + 2S * 60 = 420/\text{giorno}$
Op. 2		
Giocatore	1	L
		$1L * 5 = 5/\text{giorno}$
<b>totale</b>		1205/giorno

Senza dato derivato

Op. 1		
<b>Concetto</b>	<b>Accesso</b>	<b>Tipo</b>
Partita	1	L
Statistica	1	L
Statistica	1	S
		$2L * 60 + 1S * 60 = 240/\text{giorno}$
Op. 2		
Giocatore	1	L
Statistiche	100	L
		$101L * 5 = 505/\text{giorno}$
<b>totale</b>		745/giorno

**Conclusione:**

Con i calcoli effettuati siamo arrivati alla conclusione che è meglio non avere tale dato derivato con un totale di 745 accessi al giorno contro 1205 accessi al giorno nel caso ci fosse, per eseguire le 2 operazioni.

## Vincoli aggiuntivi

attraverso il database non è possibile gestire determinati vincoli, che verranno gestiti tramite trigger e checks. I vincoli da gestire sono:

1. La tessera di preparatore deve essere formata da una lettera e due numeri (Es: A22, B32)
2. Per un allenatore il record è formato da Vittorie - Sconfitte (es: 1-10, 12-5, 102-28)
3. Per un Giocatore il Numero di Maglia è compreso tra 0 e 99 inclusi, il Ruolo può solamente essere uno di quelli sotto:
  - Ala Grande
  - Ala Piccola
  - Playmaker
  - Guardia
  - Centro

Lo status può essere solo uno di quelli sotto:

- Starter
  - Rookie
  - Bencher
  - All-Star
4. Per lo stadio il Numero di posti deve essere  $> 2000$  e il Costo del Biglietto deve essere  $\geq 0$
  5. In Partita il Tipo può essere "Casa" o "Trasferta" e l'Esito 'W' o 'L'
  6. In Campionato Serie può essere "Campionato" o "Coppa". Stagione è nel formato 2020/21
  7. In Trasferimento la Tipologia può essere "Prestito" o "Cessione"
  8. In Infortunio Data rientro  $>$  Data Inizio. La Gravità può essere "Lieve" o "Grave"
  9. In Budget Disponibile = Massimale+Sponsor-Monte\_Ingaggi
  10. In Contratto Durata può essere compreso tra 1 e 5 anni
  11. Una Persona non può firmare un nuovo contratto se l'Anno Firma del nuovo contratto è all'interno del range Anno firma-Durata del vecchio contratto
  12. Una Persona non può subire un nuovo infortunio dello stesso tipo (Descrizione) se la Data Iniziale del nuovo infortunio è nel range Data iniziale-Data rientro del vecchio infortunio dello stesso tipo
  13. Non si può inserire un nuovo contratto se il Budget Disponibile non è sufficiente
  14. Il Numero di Biglietti venduti non può superare il numero di posti dello Stadio

15. In Merchandising la QT di riordino è influenzata dallo status del Giocatore (All Star -> default \*2, Rookie -> default \*1, Becher -> default \*0.75 , Starter -> default \*1,25)
16. In Merchandising se la QT Disponibile è minore della QT\_Riordino si riordina il Merchandising relativo a quel giocatore
17. L'inserimento di un nuovo numero di maglia non può essere uguale ad un numero di maglia ritirato
18. Un giocatore non può giocare una partita se è infortunato

## SQL

### Query di Creazione

```
CREATE TABLE IF NOT EXISTS SQUADRA
(
  Nome VARCHAR(50) PRIMARY KEY,
  Città VARCHAR(50) NOT NULL
);
```

```
CREATE TABLE IF NOT EXISTS PERSONA
(
  CF CHAR(10) PRIMARY KEY,
  Nome VARCHAR(30) NOT NULL,
  Cognome VARCHAR(30) NOT NULL,
  Eta INT NOT NULL,
  Nazionalita VARCHAR(30) NOT NULL,
  Nome_squadra VARCHAR(50) NOT NULL,
  FOREIGN KEY(Nome_Squadra) REFERENCES Squadra(Nome)
  ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS PREPARATORE
```

```
(
  CF CHAR(10) PRIMARY KEY,
  Tesserata VARCHAR(3) NOT NULL UNIQUE,
  Tipo VARCHAR(30) NOT NULL,
  Tesserata_capo VARCHAR(3),
  FOREIGN KEY(CF) REFERENCES PERSONA(CF)
    ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY(Tesserata_capo) REFERENCES PREPARATORE(Tesserata)
    ON DELETE SET NULL ON UPDATE CASCADE,
  CHECK
  (
    Tesserata~*'[A-Z][0-9][0-9]'
  )
);
```

```
CREATE TABLE IF NOT EXISTS ALLENATORE
```

```
(
  CF CHAR(10) PRIMARY KEY,
  Record VARCHAR(7) DEFAULT '0-0',
  FOREIGN KEY(CF) REFERENCES PERSONA
    ON DELETE CASCADE ON UPDATE CASCADE,
  CHECK
  (
    Record~*'[0-9]?[0-9]?[0-9]-[0-9]?[0-9]?[0-9]'
  )
);
```

```
CREATE TABLE IF NOT EXISTS GIOCATORE
```

```
(
  CF CHAR(10) PRIMARY KEY,
  N_Maglia INT UNIQUE NOT NULL,
  Ruolo VARCHAR(20) NOT NULL,
  Status VARCHAR(20) DEFAULT null,
  Altezza DECIMAL DEFAULT null,
  FOREIGN KEY(cf) REFERENCES PERSONA
    ON DELETE CASCADE ON UPDATE CASCADE,
  CHECK
  (
    N_Maglia>=0 AND N_Maglia<100 AND
    Ruolo='Ala Grande' OR Ruolo='Ala Piccola' OR Ruolo='PlayMaker' OR
    Ruolo='Guardia' OR Ruolo='Centro' AND
    Status='Starter' OR Status='Rookie' OR Status='Bencher' OR
    Status='All-Star'
  )
);
```

```
CREATE TABLE IF NOT EXISTS ALLENAMENTO
```

```
(
```

Progetto realizzato da Luca Montanari e Davide Rivi

```

Data DATE,
Tessera VARCHAR(3),
Tipo VARCHAR(30),
PRIMARY KEY(Data,Tessera),
FOREIGN KEY(Tessera) REFERENCES PREPARATORE(Tessera)
ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE IF NOT EXISTS STADIO
(
Nome VARCHAR(50) PRIMARY KEY,
Costo DECIMAL DEFAULT 80.00 NOT NULL,
N_posti INT default 5000,
CHECK
(
Costo>=0 AND N_posti>2000
)
);

CREATE TABLE IF NOT EXISTS PARTITA
(
Cod CHAR(5) PRIMARY KEY,
Data DATE NOT NULL UNIQUE,
Nome_Stadio VARCHAR(50) NOT NULL,
Avversario VARCHAR(50) NOT NULL,
Tipo VARCHAR(6) NOT NULL,
Risultato VARCHAR(7) DEFAULT '-',
Esito CHAR(1),
Biglietti_venduti INT,
FOREIGN KEY(Nome_Stadio) REFERENCES Stadio(Nome)
ON UPDATE CASCADE,
CHECK
(
Tipo='Casa' OR Tipo='Trasf' AND
Risultato~'*'[0-9]?[0-9][0-9]-[0-9]?[0-9][0-9]' AND
Biglietti_venduti>=0 AND
Esito='W' OR Esito='L'
)
);

CREATE TABLE IF NOT EXISTS STATISTICHE
(
Cod_P CHAR(5),
CF_G VARCHAR(10),
Punti INT DEFAULT 0,
Assist INT DEFAULT 0,
Rimbalzi INT DEFAULT 0,
Minuti INT DEFAULT 0,
PRIMARY KEY(Cod_P,CF_G),
FOREIGN KEY(CF_G) REFERENCES GIOCATORE(CF)
ON DELETE CASCADE ON UPDATE CASCADE,

```

Progetto realizzato da Luca Montanari e Davide Rivi

```

FOREIGN KEY(Cod_P) REFERENCES Partita(Cod)
    ON DELETE CASCADE ON UPDATE CASCADE,
CHECK
(
    Minuti>=0 AND Punti>=0 AND Assist>=0 AND Rimbalzi>=0
)
);

```

```

CREATE TABLE IF NOT EXISTS CAMPIONATO
(
    Stagione CHAR(7),
    Serie VARCHAR(20),
    N_squadre INT DEFAULT 20,
    N_partite_squadra INT DEFAULT 82,
    PRIMARY KEY(Stagione,Serie),
    CHECK
    (
        N_squadre>0 AND
        N_partite_squadra>0 AND
        Serie='Campionato' OR Serie='Coppa' AND
        Stagione~*'[1-2][0-9][0-9][0-9]/[0-9][0-9]'
    )
);

```

```

CREATE TABLE IF NOT EXISTS PARTECIPA
(
    Stagione CHAR(7),
    Serie VARCHAR(20),
    Nome_Squadra VARCHAR(50),
    Posizione int,
    PRIMARY KEY(Stagione,Serie,Nome_Squadra),
    FOREIGN KEY(Nome_Squadra) REFERENCES Squadra(Nome)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY(Stagione,Serie) REFERENCES Campionato
        ON DELETE CASCADE ON UPDATE CASCADE,
    CHECK
    (
        Posizione>0 AND
        Serie='Campionato' OR Serie='Coppa' AND
        Stagione~*'[1-2][0-9][0-9][0-9]/[0-9][0-9]'
    )
);

```

```

CREATE TABLE IF NOT EXISTS MAGLIE_RITIRATE
(
    Nome_Squadra VARCHAR(50),
    N_maglia int,
    PRIMARY KEY(N_maglia,Nome_Squadra),
    FOREIGN KEY(Nome_Squadra) REFERENCES Squadra(Nome)

```

Progetto realizzato da Luca Montanari e Davide Rivi

```

        ON DELETE CASCADE ON UPDATE CASCADE,
CHECK
(
    N_Maglia>=0 AND N_Maglia<100
)
);

CREATE TABLE IF NOT EXISTS CAMPIONATI_VINTI
(
    Nome_Squadra VARCHAR(50),
    Anno int,
    Nome_Camp VARCHAR(20),
    PRIMARY KEY(Anno, Nome_Camp, Nome_Squadra),
    FOREIGN KEY(Nome_Squadra) REFERENCES Squadra(Nome)
        ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE IF NOT EXISTS TRASFERIMENTO
(
    Cod CHAR(5) PRIMARY KEY,
    Valore DECIMAL DEFAULT 0.00 NOT NULL,
    Squadra_Dest VARCHAR(50),
    CF_G CHAR(10) NOT NULL,
    CF_A CHAR(10) NOT NULL,
    Durata INT,
    Contropartita BOOL,
    Tipologia VARCHAR(20),
    FOREIGN KEY(CF_G) REFERENCES GIOCATORE(CF)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY(CF_A) REFERENCES ALLENATORE(CF)
        ON DELETE SET NULL ON UPDATE CASCADE,
CHECK
(
    Valore>=0 AND
    Durata>=0 AND
    Tipologia='Prestito' OR Tipologia='Cessione'
)
);

CREATE TABLE IF NOT EXISTS MERCHANDISING
(
    CF_G CHAR(10),
    Cod CHAR(5),
    Qt INT DEFAULT 50 NOT NULL,
    Qt_R INT DEFAULT 50,
    PRIMARY KEY(CF_G, Cod),
    FOREIGN KEY(CF_G) REFERENCES GIOCATORE(CF)
        ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE IF NOT EXISTS INFORTUNIO

```

Progetto realizzato da Luca Montanari e Davide Rivi



```

(
  CF_G CHAR(10),
  Data_Inizio DATE,
  Descrizione VARCHAR(20),
  Data_Rientro DATE,
  Gravita VARCHAR(20) NOT NULL,
  PRIMARY KEY(CF_G,Data_Inizio,Descrizione),
  FOREIGN KEY(CF_G) REFERENCES GIOCATORE(CF)
    ON DELETE CASCADE ON UPDATE CASCADE,
  CHECK
  (
    Data_Rientro>Data_Inizio AND
    Gravita='Lieve' OR Gravita='Grave'
  )
);

CREATE TABLE IF NOT EXISTS BUDGET
(
  Stagione CHAR(7) UNIQUE,
  Nome_squadra VARCHAR(50),
  Sponsor DECIMAL DEFAULT 0.00,
  Massimale DECIMAL NOT NULL,
  Disponibile DECIMAL,
  Monte_Ingaggi DECIMAL NOT NULL,
  PRIMARY KEY(Stagione,Nome_Squadra),
  FOREIGN KEY(Nome_Squadra) REFERENCES Squadra(Nome)
    ON DELETE CASCADE ON UPDATE CASCADE,
  CHECK
  (
    Massimale>0 AND Monte_Ingaggi>=0 AND
    Stagione~*' [1-2] [0-9] [0-9] [0-9] / [0-9] [0-9] '
  )
);

CREATE TABLE IF NOT EXISTS CONTRATTO
(
  CF_P CHAR(10),
  Anno_Firma INT,
  Durata INT NOT NULL,
  Stipendio DECIMAL NOT NULL,
  Stagione_B CHAR(7) NOT NULL,
  PRIMARY KEY(CF_P,Anno_Firma),
  FOREIGN KEY(CF_P) REFERENCES PERSONA(CF)
    ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY(Stagione_B) REFERENCES Budget(Stagione)
    ON DELETE CASCADE ON UPDATE CASCADE,
  CHECK
  (
    Stipendio>0 AND
    Durata>0 AND Durata<6 AND
    Stagione_B~*' [1-2] [0-9] [0-9] [0-9] / [0-9] [0-9] '

```

Progetto realizzato da Luca Montanari e Davide Rivi

```

)
);

CREATE TABLE IF NOT EXISTS BONUS
(
  CF_P CHAR(10),
  Anno_Firma INT,
  Nome VARCHAR(20),
  PRIMARY KEY (CF_P, Anno_Firma, Nome),
  FOREIGN KEY (CF_P, Anno_Firma) REFERENCES Contratto (CF_P, Anno_Firma)
    ON DELETE CASCADE ON UPDATE CASCADE
);

```

## Stored Procedure & Trigger

**Una Persona non può firmare un nuovo contratto se l'Anno Firma del nuovo contratto è all'interno del range Anno firma-Durata del vecchio contratto**

```

CREATE FUNCTION RINNOVO() RETURNS TRIGGER AS $$
DECLARE
  Q1 INT;
BEGIN

  select COUNT(*) into Q1
  from CONTRATTO C
  where C.CF_P=NEW.CF_P AND NEW.ANNO_FIRMA BETWEEN C.ANNO_FIRMA AND
C.ANNO_FIRMA+C.DURATA;

  IF Q1 IS NOT NULL THEN
    IF Q1>0 THEN
      RAISE EXCEPTION 'GIOCATORE GIA SOTTO CONTRATTO';
    ELSE
      RETURN NEW;
    END IF;
  ELSE
    RETURN NEW;
  END IF;

END

$$ LANGUAGE 'plpgsql';

CREATE TRIGGER GESTIONE_RINNOVO
BEFORE
INSERT ON CONTRATTO

```

```
FOR EACH ROW WHEN (NEW.ANNO_FIRMA >= 0) EXECUTE FUNCTION RINNOVO();
```

**Una Persona non può subire un nuovo infortunio dello stesso tipo (Descrizione) se la Data Iniziale del nuovo infortunio è nel range Data iniziale-Data rientro del vecchio infortunio dello stesso tipo**

```
CREATE FUNCTION INFORTUNIO() RETURNS TRIGGER AS $$
DECLARE
    D1 int;
BEGIN
    Select count(*) into D1
    from Infortunio
    Where Descrizione=NEW.Descrizione and CF_G=new.CF_G AND
    Data_Rientro Between new.data_inizio and new.data_rientro;

    IF D1 is not null then
        if D1 > 0 then
            raise exception 'Inserimento Impossibile';
        else
            return NEW;
        END IF;
    end if;
    return new;
END
$$ LANGUAGE'plpgsql';
```

```
CREATE TRIGGER GESTIONE_INFORTUNIO
BEFORE
INSERT
OR
UPDATE ON INFORTUNIO
FOR EACH ROW EXECUTE FUNCTION INFORTUNIO();
```

**Non si può inserire un nuovo contratto se il Budget Disponibile non è sufficiente**

```
CREATE FUNCTION CONTROLLO_BUDGET() RETURNS TRIGGER AS $$
DECLARE

Q1 DECIMAL;
Q2 DECIMAL;

BEGIN
```

```

Select Disponibile into Q1
From Budget
where stagione=new.stagione_b;

Q2=new.stipendio;

if Q1-NEW.Stipendio < 0 then
    raise exception 'Budget insufficiente';
else
    update budget
    set monte_ingaggi=monte_ingaggi+Q2
    where stagione=new.stagione_b;
    return NEW;
END IF;

END
$$ LANGUAGE 'plpgsql';

```

```

CREATE TRIGGER CONTROLLO_BUDGET
BEFORE
INSERT
OR
UPDATE ON CONTRATTO
FOR EACH ROW EXECUTE FUNCTION CONTROLLO_BUDGET();

```

**Il Numero di Biglietti venduti non può superare il numero di posti dello Stadio**

```

CREATE FUNCTION CONTROLLO_BIGLIETTI() RETURNS TRIGGER AS $$
DECLARE

Q1 INTEGER ;

BEGIN

    Select A.N_posti into Q1
    From stadio A
    Where A.Nome = new.Nome_stadio;

    if Q1-NEW.biglietti_venduti < 0 then
        raise exception 'Numero di posti insufficiente';
    else
        return NEW;
    end if;
END;

```

```

        END IF;

    END
    $$ LANGUAGE 'plpgsql';

CREATE TRIGGER CONTROLLO_BIGLIETTI AFTER
INSERT
OR
UPDATE ON PARTITA
FOR EACH ROW WHEN (NEW.BIGLIETTI_VENDUTI > 0) EXECUTE FUNCTION
CONTROLLO_BIGLIETTI();

```

In Merchandising la QT di riordino è influenzata dallo status del Giocatore (All Star -> default \*2, Rookie -> default \*1, Becher -> default \*0.75 , Starter -> default \*1,25)

```

CREATE FUNCTION QT_STATUS() RETURNS TRIGGER AS $$
DECLARE
    Q1 CHAR(5);
    Q2 VARCHAR(20);
BEGIN

    Select M.COD into Q1
    From MERCHANDISING M
    Where  M.CF_G=NEW.CF;

    Select G.Status into Q2
    from GIOCATORE G
    where G.CF=new.CF;
- funzione per generare stringa casuale
    SELECT array_to_string(ARRAY(SELECT chr((97 + round(random()
* 25)) :: integer) into Q1
    FROM generate_series(1,5)), '');

    DELETE FROM MERCHANDISING WHERE CF_G=NEW.CF;

    IF Q2='All-Star' THEN
        INSERT INTO MERCHANDISING VALUES (NEW.CF,Q1,50,50*2);
    ELSIF Q2='Rookie' THEN
        INSERT INTO MERCHANDISING VALUES (NEW.CF,Q1,50,50*1);
    ELSIF Q2='Starter' THEN
        INSERT INTO MERCHANDISING
VALUES (NEW.CF,Q1,50,50*1.25);

```

```

        ELSE
            INSERT INTO MERCHANDISING
VALUES (NEW.CF,Q1,50,50*0.75);
        END IF;

        RETURN NEW;

END
$$ LANGUAGE 'plpgsql';

```

```

CREATE TRIGGER QT_DEFAULT AFTER
INSERT
OR
UPDATE ON GIOCATORE
FOR EACH ROW EXECUTE PROCEDURE QT_STATUS();

```

**In Merchandising se la QT Disponibile è minore della QT Riordino, dopo un acquisto si riordina il Merchandising relativo a quel giocatore**

```

CREATE FUNCTION ACQUISTO(PROD CHAR(10), QUANT INTEGER) RETURNS INTEGER
AS $$
DECLARE
Q1 INTEGER;
Q2 INTEGER;
BEGIN
SELECT QT INTO Q1
FROM MERCHANDISING
WHERE CF_G = PROD;

Q2 := Q1 - QUANT;

UPDATE MERCHANDISING
SET QT = Q2
WHERE CF_G = PROD;

RETURN Q2;

END
$$ LANGUAGE 'plpgsql';

```

```

CREATE FUNCTION RIORDINO() RETURNS TRIGGER AS $$

```

```

BEGIN
IF NEW. QT < NEW. QT_R THEN
IF NEW. QT < 0 THEN
RAISE EXCEPTION 'Modifica impossibile';
ELSE

UPDATE MERCHANDISING
SET QT = QT_R*2
WHERE COD = OLD.COD;
END IF;
END IF;

RETURN NEW;

END;
$$ LANGUAGE 'plpgsql';

CREATE TRIGGER GESTIONE_RIORDINO AFTER
UPDATE ON MERCHANDISING
FOR EACH ROW EXECUTE PROCEDURE RIORDINO();

```

**L'inserimento di un nuovo numero di maglia non può essere uguale ad un numero di maglia ritirato**

```

CREATE FUNCTION MAGLIE_RIT() RETURNS TRIGGER AS $$
DECLARE
    C1 INT;
BEGIN

    SELECT G.N_Maglia INTO C1
    FROM GIOCATORE G
    WHERE G.N_MAGLIA=new.N_MAGLIA and G.N_MAGLIA in
    ( SELECT N_MAGLIA
      FROM MAGLIE_RITIRATE);

    IF C1>=0 THEN
        DELETE FROM GIOCATORE WHERE n_maglia=C1;
        RAISE EXCEPTION 'MAGLIA RITIRATA';
    else
        RETURN NEW;
    END IF;

```

```

END
$$ LANGUAGE 'plpgsql';

CREATE TRIGGER GESTIONE_MAGLIE AFTER
INSERT
OR
UPDATE ON GIOCATORE
FOR EACH ROW EXECUTE PROCEDURE MAGLIE_RIT();

```

### Calcolo del disponibile quando si modifica un contratto

```

CREATE FUNCTION DISPONIBILE() RETURNS TRIGGER AS $$
    DECLARE

    Q1 DECIMAL;
    Q2 DECIMAL;
    Q3 DECIMAL;

    BEGIN

        Select massimale,sponsor,monte_ingaggi into Q1,Q2,Q3
        From budget
        where Stagione=new.Stagione_b;

        if Q1-Q3+Q2<0 THEN
            raise exception 'Disponibile insufficiente';
        else
            update budget
            set disponibile=Q1-Q3+Q2
            where stagione=new.Stagione_b;
        END IF;

        return NEW;
    END
    $$ LANGUAGE 'plpgsql';

CREATE TRIGGER CALCOLO_DISPONIBILE
BEFORE
INSERT
OR
UPDATE ON CONTRATTO
FOR EACH ROW EXECUTE FUNCTION DISPONIBILE();

```



funzione che gestisce i trasferimenti, se la durata é 0 imposta cessione, se è >0 imposta prestito

```
CREATE PROCEDURE TRASFERIMENTO(COD CHAR(5), VAL DECIMAL, SQUADRA_DEST
VARCHAR, CF_G CHAR(10), CF_A CHAR(10), DURATA INT, CONTROPARTITA
BOOLEAN) AS $$
    DECLARE

    Q1 INTEGER;

    BEGIN

        select COUNT(*) into q1
        from GIOCATORE G, PERSONA P
        where G.CF=P.CF and P.nome_squadra = 'Golden State Warriors';

        IF q1-1 < 5 THEN
            RAISE EXCEPTION 'Avresti meno di 5 giocatori!!
Trasferimento annullato';
        END IF;

        IF(durata>0) THEN
            insert into TRASFERIMENTO values (COD, VAL, Squadra_Dest,
CF_G, CF_A, DURATA, Contropartita, 'Prestito');
        ELSE
            insert into TRASFERIMENTO values (COD, VAL,
Squadra_Dest, CF_G, CF_A, DURATA, Contropartita, 'Cessione');
        END IF;

        UPDATE PERSONA
        SET nome_squadra = Squadra_Dest
        where CF = CF_G;

    END
    $$ LANGUAGE'plpgsql';
```

**Se Un Giocatore È Infortunato Non Può Giocare Una Partita**

```
CREATE FUNCTION INDISPONIBILITA() RETURNS TRIGGER AS $$
DECLARE
Q1 DATE;
Q2 DATE;
Q3 DATE;

BEGIN

    SELECT P.DATA INTO Q1
    FROM PARTITA p
```

```

WHERE COD=NEW.COD_P;

SELECT I.DATA_RIENTRO,I.DATA_INIZIO INTO Q2,Q3
FROM INFORTUNIO I
WHERE I.CF_G=NEW.CF_G AND I.DATA_INIZIO>=ALL(
SELECT DATA_INIZIO
FROM INFORTUNIO P
WHERE P.CF_G=NEW.CF_G);

IF Q3 IS NOT NULL THEN
    IF Q1<=Q2 AND Q1>=Q3 THEN
        RAISE EXCEPTION 'Giocatore infortunato, non può
giocare';
    ELSE
        RETURN NEW;
    END IF;
END IF;

RETURN NEW;

END
$$ LANGUAGE'plpgsql';

CREATE TRIGGER GESTIONE_PARTITE
BEFORE
INSERT
OR
UPDATE ON STATISTICHE
FOR EACH ROW EXECUTE FUNCTION INDISPONIBILITA();

```

## Query di inserimento

```

INSERT INTO squadra VALUES('Boston Celtics','Boston');
INSERT INTO squadra VALUES('Brooklyn Nets','New York');
INSERT INTO squadra VALUES('New York Knicks','New York');
INSERT INTO squadra VALUES('Philadelphia 76ers','Filadelfia');
INSERT INTO squadra VALUES('Toronto Raptors','Toronto');
INSERT INTO squadra VALUES('Chicago Bulls','Chicago');
INSERT INTO squadra VALUES('Cleveland Cavaliers','Cleveland');
INSERT INTO squadra VALUES('Detroit Pistons','Detroit');
INSERT INTO squadra VALUES('Milwaukee Bucks','Milwaukee');
INSERT INTO squadra VALUES('Atlanta Hawks','Atlanta');
INSERT INTO squadra VALUES('Charlotte Hornets','Charlotte');
INSERT INTO squadra VALUES('Miami Heat','Miami');
INSERT INTO squadra VALUES('Orlando Magic','Orlando');
INSERT INTO squadra VALUES('Washington Wizards','Washington');
INSERT INTO squadra VALUES('Denver Nuggets','Denver');

```

```

INSERT INTO squadra VALUES('Minnesota Timberwolves','Minneapolis');
INSERT INTO squadra VALUES('Oklahoma City Thunder','Oklahoma City');
INSERT INTO squadra VALUES('Portland Trail Blazers','Portland');
INSERT INTO squadra VALUES('Utah Jazz','Utah');
INSERT INTO squadra VALUES('Golden State Warriors','San Francisco');
INSERT INTO squadra VALUES('Los Angeles Clippers','Los Angeles');
INSERT INTO squadra VALUES('Los Angeles Lakers','Los Angeles');
INSERT INTO squadra VALUES('Phoenix Suns','Phoenix');
INSERT INTO squadra VALUES('Sacramento Kings','Sacramento');
INSERT INTO squadra VALUES('Dallas Mavericks','Dallas');
INSERT INTO squadra VALUES('Houston Rockets','Houston');
INSERT INTO squadra VALUES('Memphis Grizzlies','Memphis');
INSERT INTO squadra VALUES('New Orleans Pelicans','New Orleans');
INSERT INTO squadra VALUES('San Antonio Spurs','San Antonio');
insert into squadra values('Estero','Estero');
INSERT INTO persona VALUES('2VQL7NA3Uh','Nemanja','Bjelica','33','Serbia','Golden State Warriors');
INSERT INTO persona VALUES('3LVdbbwjyQ','Chris','Chiozza','27','USA','Golden State Warriors');
INSERT INTO persona VALUES('bMSzjEuUWr','Stephen','Curry','33','USA','Golden State Warriors');
INSERT INTO persona VALUES('sRELjEAY4w','Draymond','Green','32','USA','Golden State Warriors');
INSERT INTO persona VALUES('VD0sEGq4gv','Andre','Iguodala','36','USA','Golden State Warriors');
INSERT INTO persona VALUES('0lbP86MJ55','Jonathan','Kuminga','20','Congo','Golden State Warriors');
INSERT INTO persona VALUES('MvXBlu4TxJ','Damion','Lee','30','USA','Golden State Warriors');
INSERT INTO persona VALUES('62QyuZSCtJ','Kevon','Looney','26','USA','Golden State Warriors');
INSERT INTO persona VALUES('mix2pWfebX','Moses','Moody','20','USA','Golden State Warriors');
INSERT INTO persona VALUES('cf9Xf2hz5w','Gary','Payton II','30','USA','Golden State Warriors');
INSERT INTO persona VALUES('bqm7YByEKB','Jordan','Poole','23','USA','Golden State Warriors');
INSERT INTO persona VALUES('fWqaexj9a9','Otto','Porte Jr','29','USA','Golden State Warriors');
INSERT INTO persona VALUES('ZUsJLQ2Rdd','Klay','Thompson','32','USA','Golden State Warriors');
INSERT INTO persona
VALUES('m2rHYWpbFY','Juan','Toscano-Anderson','29','USA','Golden State Warriors');
INSERT INTO persona VALUES('pSi62RRLtc','Quinn','Weatherspoon','26','USA','Golden State Warriors');
INSERT INTO persona VALUES('asoLihn79I','Andrew','Wiggins','27','Canada','Golden State Warriors');

```

```

INSERT INTO persona VALUES('8MHVmuuZRn','James','Wiseman','21','USA','Golden State Warriors');
INSERT INTO persona VALUES('t70r53H0I9','Steve','Kerr','50','USA','Golden State Warriors');
INSERT INTO persona VALUES('oOpgvlzsaH','Ron','Adams','37','USA','Golden State Warriors');
INSERT INTO persona VALUES('5pVpFNIEgM','Kenny','Atkinson','40','USA','Golden State Warriors');
INSERT INTO persona VALUES('rGhMRRv4Yx','Bobby','Brown','42','USA','Golden State Warriors');
INSERT INTO persona VALUES('mdVFRkdPLg','Milos','Slukas','36','Grecia','Golden State Warriors');
INSERT INTO persona VALUES('mjDEJkdClp','DeJan','Milojevic','39','Croazia','Golden State Warriors');
INSERT INTO persona VALUES('mdVFRkdPDy','Drew','Yoder','29','Canada','Golden State Warriors');
INSERT INTO persona VALUES('lnBRBggOSr','Leandro','Barbosa','21','USA','Golden State Warriors');
INSERT INTO persona VALUES('yhv7F2vbQv','Mark','Jackson','55','USA','Golden State Warriors');
INSERT INTO persona VALUES('dKOasXiVAz','Kevin','Durant','34','USA','Golden State Warriors');
INSERT INTO persona VALUES('nmlweSiPZz','Niccolò','Mannion','21','Italia','Golden State Warriors');
INSERT INTO persona VALUES('dcSojPIACq','DeMarcus','Cousins','31','USA','Golden State Warriors');
INSERT INTO budget VALUES('2021/22','Golden State Warriors','30000000.00','80000000.00',NULL,'0.00');
INSERT INTO budget VALUES('2020/21','Golden State Warriors','20000000.00','60000000.00',NULL,'0.00');
INSERT INTO budget VALUES('2019/20','Golden State Warriors','22000000.00','75000000.00',NULL,'0.00');
INSERT INTO budget VALUES('2018/19','Golden State Warriors','10000000.00','70000000.00',NULL,'0.00');
INSERT INTO budget VALUES('2017/18','Golden State Warriors','12000000.00','67000000.00',NULL,'0.00');
INSERT INTO budget VALUES('2016/17','Golden State Warriors','10000000.00','70000000.00',NULL,'0.00');
INSERT INTO budget VALUES('2015/16','Golden State Warriors','15000000.00','50000000.00',NULL,'0.00');
INSERT INTO budget VALUES('2014/15','Golden State Warriors','5000000.00','45000000.00',NULL,'0.00');
INSERT INTO campionato VALUES('2021/22','Campionato','20','82');
INSERT INTO campionato VALUES('2020/21','Campionato','20','82');
INSERT INTO campionato VALUES('2019/20','Campionato','20','82');
INSERT INTO campionato VALUES('2018/19','Campionato','20','82');
INSERT INTO campionato VALUES('2017/18','Campionato','20','82');
INSERT INTO campionato VALUES('2016/17','Campionato','20','82');

```

Progetto realizzato da Luca Montanari e Davide Rivi

```

INSERT INTO campionato VALUES('2015/16','Campionato','20','82');
INSERT INTO campionato VALUES('2014/15','Campionato','20','82');
INSERT INTO campionato VALUES('2021/22','Coppa','20','82');
INSERT INTO campionato VALUES('2020/21','Coppa','20','82');
INSERT INTO campionato VALUES('2019/20','Coppa','20','82');
INSERT INTO campionato VALUES('2018/19','Coppa','20','82');
INSERT INTO campionato VALUES('2017/18','Coppa','20','82');
INSERT INTO campionato VALUES('2016/17','Coppa','20','82');
INSERT INTO campionato VALUES('2015/16','Coppa','20','82');
INSERT INTO campionato VALUES('2014/15','Coppa','20','82');
INSERT INTO partecipa VALUES('2021/22','Campionato','Golden State Warriors','1');
INSERT INTO partecipa VALUES('2020/21','Campionato','Golden State Warriors',null);
INSERT INTO partecipa VALUES('2019/20','Campionato','Golden State Warriors',null);
INSERT INTO partecipa VALUES('2018/19','Campionato','Golden State Warriors','2');
INSERT INTO partecipa VALUES('2017/18','Campionato','Golden State Warriors','1');
INSERT INTO partecipa VALUES('2016/17','Campionato','Golden State Warriors','1');
INSERT INTO partecipa VALUES('2015/16','Campionato','Golden State Warriors','2');
INSERT INTO partecipa VALUES('2014/15','Campionato','Golden State Warriors','1');
INSERT INTO partecipa VALUES('2021/22','Coppa','Golden State Warriors','3');
INSERT INTO partecipa VALUES('2020/21','Coppa','Golden State Warriors','9');
INSERT INTO partecipa VALUES('2019/20','Coppa','Golden State Warriors','15');
INSERT INTO partecipa VALUES('2018/19','Coppa','Golden State Warriors','1');
INSERT INTO partecipa VALUES('2017/18','Coppa','Golden State Warriors','1');
INSERT INTO partecipa VALUES('2016/17','Coppa','Golden State Warriors','1');
INSERT INTO partecipa VALUES('2015/16','Coppa','Golden State Warriors','1');
INSERT INTO partecipa VALUES('2014/15','Coppa','Golden State Warriors','1');
INSERT INTO campionati_vinti VALUES('Golden State Warriors','1947','Titolo Nba');
INSERT INTO campionati_vinti VALUES('Golden State Warriors','1956','Titolo Nba');
INSERT INTO campionati_vinti VALUES('Golden State Warriors','1975','Titolo Nba');
INSERT INTO campionati_vinti VALUES('Golden State Warriors','2015','Titolo Nba');
INSERT INTO campionati_vinti VALUES('Golden State Warriors','2017','Titolo Nba');
INSERT INTO campionati_vinti VALUES('Golden State Warriors','2018','Titolo Nba');
INSERT INTO campionati_vinti VALUES('Golden State Warriors','1947','Conference');
INSERT INTO campionati_vinti VALUES('Golden State Warriors','1948','Conference');
INSERT INTO campionati_vinti VALUES('Golden State Warriors','1956','Conference');
INSERT INTO campionati_vinti VALUES('Golden State Warriors','1964','Conference');
INSERT INTO campionati_vinti VALUES('Golden State Warriors','1967','Conference');
INSERT INTO campionati_vinti VALUES('Golden State Warriors','1975','Conference');
INSERT INTO campionati_vinti VALUES('Golden State Warriors','2015','Conference');
INSERT INTO campionati_vinti VALUES('Golden State Warriors','2016','Conference');
INSERT INTO campionati_vinti VALUES('Golden State Warriors','2017','Conference');
INSERT INTO campionati_vinti VALUES('Golden State Warriors','2018','Conference');
INSERT INTO campionati_vinti VALUES('Golden State Warriors','2019','Conference');
INSERT INTO campionati_vinti VALUES('Golden State Warriors','2022','Conference');
INSERT INTO maglie_ritirate VALUES('Golden State Warriors','13');
INSERT INTO maglie_ritirate VALUES('Golden State Warriors','14');
INSERT INTO maglie_ritirate VALUES('Golden State Warriors','16');
INSERT INTO maglie_ritirate VALUES('Golden State Warriors','17');

```

Progetto realizzato da Luca Montanari e Davide Rivi

```

INSERT INTO maglie_ritirate VALUES('Golden State Warriors','24');
INSERT INTO maglie_ritirate VALUES('Golden State Warriors','42');
INSERT INTO contratto VALUES('yHv7F2vbQv','2014','3','5000000','2014/15');
INSERT INTO contratto VALUES('bMSzjEuUWr','2016','4','12000000','2015/16');
INSERT INTO contratto VALUES('mjDEJkdClp','2015','4','810000','2015/16');
INSERT INTO contratto VALUES('lnBRBggOSr','2015','4','100000','2015/16');
INSERT INTO contratto VALUES('dKOasXiVAz','2016','3','35000000','2016/17');
insert into contratto values('ZUsJLQ2Rdd','2016','3','17000000','2016/17');
insert into contratto values('sRELjEAY4w','2017','3','12000000','2017/18');
INSERT INTO contratto VALUES('mdVFRkdPDy','2017','3','500000','2017/18');
INSERT INTO contratto VALUES('dcSojPIACq','2018','1','9700000','2018/19');
INSERT INTO contratto VALUES('t70r53H0I9','2019','5','10000000','2019/20');
INSERT INTO contratto VALUES('nmlweSiPZz','2020','3','700000','2020/21');
INSERT INTO contratto VALUES('VD0sEGq4gv','2020','1','2500000','2020/21');
INSERT INTO contratto VALUES('MvXBlu4TxJ','2020','3','2000000','2020/21');
INSERT INTO contratto VALUES('ZUsJLQ2Rdd','2020','5','17000000','2020/21');
INSERT INTO contratto VALUES('2VQL7NA3UH','2018','5','2000000','2018/19');
INSERT INTO contratto VALUES('3LVdbbwjyQ','2019','1','1500000','2019/20');
INSERT INTO contratto VALUES('bMSzjEuUWr','2021','5','11000000','2021/22');
INSERT INTO contratto VALUES('sRELjEAY4w','2021','4','7000000','2020/21');
INSERT INTO contratto VALUES('62QyuZSCtJ','2021','3','3000000','2021/22');
INSERT INTO contratto VALUES('bqm7YByEKB','2021','4','7000000','2021/22');
INSERT INTO contratto VALUES('fWqaexj9a9','2021','2','2000000','2021/22');
INSERT INTO contratto VALUES('m2rHYWpbFY','2021','3','2000000','2021/22');
INSERT INTO contratto VALUES('asoLihn79I','2021','3','12000000','2021/22');
INSERT INTO contratto VALUES('8MHVmuuZRn','2021','3','8000000','2021/22');
INSERT INTO contratto VALUES('5pVpFNIEgM','2021','3','200000','2021/22');
INSERT INTO contratto VALUES('rGhMRRv4Yx','2020','3','250000','2020/21');
INSERT INTO contratto VALUES('mdVFRkdPLg','2021','3','150000','2021/22');
INSERT INTO contratto VALUES('oOpgvlzsaH','2022','3','200000','2021/22');
INSERT INTO contratto VALUES('0IbP86MJ55','2022','4','3000000','2021/22');
INSERT INTO contratto VALUES('pSi62RRLtc','2022','3','1000000','2021/22');
INSERT INTO contratto VALUES('mix2pWfebx','2022','4','1500000','2021/22');
INSERT INTO contratto VALUES('cf9Xf2hz5w','2016','3','3000000','2016/17');
INSERT INTO bonus VALUES('bMSzjEuUWr','2021','Triple');
INSERT INTO bonus VALUES('bMSzjEuUWr','2021','All-Star');
INSERT INTO bonus VALUES('bMSzjEuUWr','2021','Top-Player');
INSERT INTO bonus VALUES('sRELjEAY4w','2021','Top-Player');
INSERT INTO bonus VALUES('sRELjEAY4w','2021','Difensore');
INSERT INTO bonus VALUES('dKOasXiVAz','2016','Top-Player');
INSERT INTO bonus VALUES('ZUsJLQ2Rdd','2020','Triple');
INSERT INTO bonus VALUES('ZUsJLQ2Rdd','2020','Top-Player');
INSERT INTO bonus VALUES('ZUsJLQ2Rdd','2020','Ritorno');
INSERT INTO bonus VALUES('t70r53H0I9','2019','Titoli');
INSERT INTO allenatore VALUES('t70r53H0I9','376-171');
INSERT INTO allenatore VALUES('yHv7F2vbQv','121-109');
INSERT INTO preparatore VALUES('oOpgvlzsaH','A01','Atletico',null);
INSERT INTO preparatore VALUES('5pVpFNIEgM','A02','Tattico',null);

```

Progetto realizzato da Luca Montanari e Davide Rivi

```

INSERT INTO preparatore VALUES('rGhMRRv4Yx','A03','Video',null);
INSERT INTO preparatore VALUES('mdVFRkdPLg','B01','Atletico','A01');
INSERT INTO preparatore VALUES('mjDEJkdClp','B02','Tattico','A01');
INSERT INTO preparatore VALUES('mdVFRkdPDy','B05','Tattico','A02');
INSERT INTO preparatore VALUES('lnBRBgOSr','D01','In prova','B05');
INSERT INTO allenamento VALUES('17/02/2019','A01','Preparazione fisica');
INSERT INTO allenamento VALUES('01/02/2016','A02','Schemi di gioco');
INSERT INTO allenamento VALUES('09/12/2017','A03','Highlights partita');
INSERT INTO allenamento VALUES('07/06/2019','A03','Review post sconfitta');
INSERT INTO allenamento VALUES('11/08/2018','B01','Preparazione');
INSERT INTO allenamento VALUES('15/02/2022','D01','Partitella');
INSERT INTO allenamento VALUES('04/04/2017','A02','Schemi di gioco');
INSERT INTO allenamento VALUES('25/04/2019','B02','Mobilita');
INSERT INTO allenamento VALUES('27/02/2019','A03','Schemi di gioco');
INSERT INTO allenamento VALUES('14/03/2017','A02','Schemi di gioco');
INSERT INTO allenamento VALUES('23/07/2020','A01','Atletica');
INSERT INTO giocatore VALUES('2VQL7NA3Uh','8','Ala Piccola','Bencher','2.10');
INSERT INTO giocatore VALUES('3LVdbbwjyQ','2','Guardia','Bencher','1.55');
INSERT INTO giocatore VALUES('bMSzjEuUWr','30','Playmaker','All-Star','1.89');
INSERT INTO giocatore VALUES('sRELjEAY4w','23','Ala Grande','All-Star','2.01');
INSERT INTO giocatore VALUES('VD0sEGq4gv','9','Ala Piccola','Bencher','2.01');
INSERT INTO giocatore VALUES('0lbP86MJ55','99','Ala Grande','Rookie','2.04');
INSERT INTO giocatore VALUES('MvXBlu4TxJ','1','Guardia','Bencher','1.98');
INSERT INTO giocatore VALUES('62QyuZSCtJ','5','Centro','Starter','2.10');
INSERT INTO giocatore VALUES('mix2pWfebX','4','Playmaker','Rookie','1.98');
INSERT INTO giocatore VALUES('cf9Xf2hz5w','0','Playmaker','Bencher','1.92');
INSERT INTO giocatore VALUES('bqm7YByEKB','3','Guardia','Starter','1.95');
INSERT INTO giocatore VALUES('fWQaexj9a9','32','Ala Piccola','Bencher','2.07');
INSERT INTO giocatore VALUES('ZUsJLQ2Rdd','11','Guardia','All-Star','2.01');
INSERT INTO giocatore VALUES('m2rHYWpbFY','95','Ala Piccola','Bencher','2.01');
INSERT INTO giocatore VALUES('pSi62RRLtc','15','Playmaker','Bencher','1.92');
INSERT INTO giocatore VALUES('asoLihn79l','22','Ala Piccola','Starter','2.04');
INSERT INTO giocatore VALUES('8MHVmuuZRn','33','Centro','Starter','2.13');
INSERT INTO giocatore VALUES('dKOasXiVAz','7','Ala Grande','All-Star','2.08');
INSERT INTO giocatore VALUES('nmlweSiPZz','12','Playmaker','Rookie','1.91');
INSERT INTO giocatore VALUES('dcSojPIACq','19','Centro','Starter','2.08');
INSERT INTO infortunio
VALUES('8MHVmuuZRn','2022/03/26','Ginocchio','2022/10/24','Grave');
INSERT INTO infortunio
VALUES('8MHVmuuZRn','2021/09/18','Ginocchio','2022/01/30','Grave');
INSERT INTO infortunio VALUES('ZUsJLQ2Rdd','2020/11/18','Tendine di
Achille','2022/01/08','Grave');
INSERT INTO infortunio
VALUES('ZUsJLQ2Rdd','2019/07/01','Crociato','2020/01/08','Grave');
INSERT INTO infortunio VALUES('dKOasXiVAz','2019/06/22','Tendine di
Achille','2019/09/12','Grave');
INSERT INTO infortunio
VALUES('bMSzjEuUWr','2021/03/12','Ginocchio','2021/06/10','Grave');

```

Progetto realizzato da Luca Montanari e Davide Rivi

```

INSERT INTO infortunio
VALUES('sRELJEAY4w','2022/06/11','Ginocchio','2022/06/13','Lieve');
INSERT INTO infortunio VALUES('fWqaexj9a9','2022/04/17','Schiena','2022/06/12','Lieve');
call trasferimento('NM21B', 700000,'Estero', 'nmlweSiPZz', 't70r53H0I9', 3, false);
call trasferimento('KD07A', 35000000,'Brooklyn Nets', 'dKOasXiVAz', 't70r53H0I9', 0, false);
call trasferimento('DC00S', 1000000,'Houston Rockets', 'dcSojPIACq', 't70r53H0I9', 0, false);
INSERT INTO stadio VALUES('Chase Center','100','15000');
INSERT INTO stadio VALUES('Staples Center','150','20000');
INSERT INTO stadio VALUES('Madison Square Garden','120','20000');
INSERT INTO stadio VALUES('Barclays Center','90','19000');
INSERT INTO stadio VALUES('Scotiabank Arena','90','15000');
INSERT INTO stadio VALUES('United Center','100','21000');
INSERT INTO stadio VALUES('Little Caesar Arena','60','15000');
INSERT INTO stadio VALUES('TD Banknorth Garden','95','19000');
INSERT INTO stadio VALUES('Quicken Loans Arena','75','15500');
INSERT INTO stadio VALUES('Wells Fargo Center','85','21000');
INSERT INTO stadio VALUES('American Airlines Center','65','18700');
INSERT INTO stadio VALUES('American Airlines Arena','90','16700');
INSERT INTO stadio VALUES('Amway Center','55','18500');
INSERT INTO stadio VALUES('Moda Center','65','16750');
INSERT INTO stadio VALUES('Chesapeake Energy Arena','70','18200');
INSERT INTO stadio VALUES('Pepsi Center','97','19500');
INSERT INTO stadio VALUES('AT&T Center','67','18300');
INSERT INTO stadio VALUES('Toyota Center','55','14600');
INSERT INTO stadio VALUES('Vivint Smart Home Arena','65','16700');
INSERT INTO stadio VALUES('Talking Stick Resort Arena','90','18000');
INSERT INTO stadio VALUES('Bankerslife Field House','50','14890');
INSERT INTO stadio VALUES('Philips Arena','78','18560');
INSERT INTO stadio VALUES('Target Center','80','17590');
INSERT INTO stadio VALUES('Smoothie King Center','64','18000');
INSERT INTO stadio VALUES('Fedex Forum','85','18250');
INSERT INTO stadio VALUES('Spectrum Center','58','16500');
INSERT INTO stadio VALUES('Capital One Arena','55','20300');
INSERT INTO stadio VALUES('Golden 1 Center','45','17600');
INSERT INTO stadio VALUES('Fiserv Forum','100','17800');
INSERT INTO partita VALUES('W4tWQ','2022/04/22','United Center','Chicago Bulls','Trasf','85-102','L','19250');
INSERT INTO partita VALUES('jfinN','2022/05/12','Chase Center','Milwaukee Bucks','Casa','94-97','W','14200');
INSERT INTO partita VALUES('LXTXT','2022/01/21','Madison Square Garden','New York Knicks','Trasf','122-102','W','17300');
INSERT INTO partita VALUES('DatSa','2021/12/21','Barclays Center','Brooklyn Nets','Trasf','92-112','L','13890');
INSERT INTO partita VALUES('EQ6QP','2022/01/09','Chase Center','Miami Heat','Casa','122-121','L','14950');
INSERT INTO partita VALUES('qNUKG','2022/01/12','Chase Center','Charlotte Hornets','Casa','88-95','W','13000');

```



```

INSERT INTO partita VALUES('VZFgW','2021/10/19','Scotiabank Arena','Toronto
Raptors','Trasf','128-101','W','13870');
INSERT INTO partita VALUES('7gnWW','2022/03/02','Little Caesar Arena','Detroit
Pistons','Trasf','103-102','W','12400');
INSERT INTO partita VALUES('hpPZe','2022/01/10','Chase Center','Detroit
Pistons','Casa','115-89','L','11340');
INSERT INTO partita VALUES('3Byqj','2022/02/25','TD Banknorth Garden','Boston
Celtics','Trasf','111-109','W','17930');
INSERT INTO partita VALUES('2jVVw','2022/01/30','Chase Center','Boston
Cetics','Casa','98-95','L','12400');
INSERT INTO partita VALUES('PBjx5','2022/04/10','Quicken Loans Arena','Cleveland
cavaliers','Trasf','107-112','L','15200');
INSERT INTO partita VALUES('chPf4','2022/03/10','Chase Center','Cleveland
Cavaliers','Casa','121-95','L','13210');
INSERT INTO partita VALUES('2vHDk','2021/11/10','Wells Fargo Center','Philadelphia
79ers','Trasf','97-88','W','20300');
INSERT INTO partita VALUES('u4eBf','2021/11/13','Chase Center','Philadelphia
79ers','Casa','103-101','L','11890');
INSERT INTO partita VALUES('STJd4','2021/10/30','American Airlines Center','Dallas
Mavericks','Trasf','87-95','L','12400');
INSERT INTO partita VALUES('JCdbX','2022/04/21','Chase Center','Dallas
Mavericks','Casa','123-95','L','13200');
INSERT INTO partita VALUES('6puVb','2021/12/31','American Airlines Arena','Miami
Heat','Trasf','113-93','W','16690');
INSERT INTO partita VALUES('uFwWn','2021/12/24','Amway Center','Orlando
Magic','Trasf','98-99','L','18000');
INSERT INTO partita VALUES('pXUH8','2022/02/17','Moda Center','Portland Trail
Blazers','Trasf','97-105','L','16300');
INSERT INTO partita VALUES('tiFem','2022/02/09','Chase Center','Portland Trail
Blazers','Casa','101-99','L','13279');
INSERT INTO partita VALUES('Yzq3y','2022/03/03','Chase Center','Toronto
Raptors','Casa','99-104','W','12567');
INSERT INTO partita VALUES('hExFP','2022/02/07','Chesapeake Energy Arena','Oklahoma
City Thunder','Trasf','97-105','L','18110');
INSERT INTO partita VALUES('TdTz','2022/04/01','Pepsi Center','Denver
Nuggets','Trasf','89-104','L','19230');
INSERT INTO partita VALUES('cCmZt','2022/01/29','Chase Center','Denver
Nuggets','Casa','104-115','W','12560');
INSERT INTO partita VALUES('krBcS','2020/11/17','Chase Center','San Antonio
Spurs','Casa','118-113','L','12490');
INSERT INTO partita VALUES('dGh3v','2020/04/16','Toyota Center','Houston
Rockets','Trasf','97-105','L','14200');
INSERT INTO partita VALUES('KmPZc','2020/12/06','Vivint Smart Home Arena','Utah
Jazz','Trasf','102-96','W','16500');
INSERT INTO partita VALUES('ztWEn','2020/10/13','Chase Center','Utah
Jazz','Casa','112-87','L','10230');
INSERT INTO partita VALUES('gdZiT','2020/03/10','Bankerslife Field House','Detroit
Pistons','Trasf','98-105','L','9550');

```

Progetto realizzato da Luca Montanari e Davide Rivi

```

INSERT INTO partita VALUES('DcapQ','2020/02/29','Chase Center','Detroit
Pistons','Casa','121-119','L','9230');
INSERT INTO partita VALUES('wBDyh','2020/12/26','Philips Arena','Atlanta
Hawks','Trasf','100-92','W','18000');
INSERT INTO partita VALUES('3bDWd','2019/11/01','Chase Center','Atlanta
Hawks','Casa','110-87','L','13200');
INSERT INTO partita VALUES('eE3Pn','2019/12/31','Target Center','Minnesota
Timberwolves','Trasf','78-86','L','12400');
INSERT INTO partita VALUES('g5Z7P','2019/04/14','Chase Center','Minnesota
Timberwolves','Casa','127-121','L','10123');
INSERT INTO partita VALUES('mkx5T','2019/03/19','Smoothie King Center','New Orleans
Pelicans','Trasf','97-105','L','12345');
INSERT INTO partita VALUES('KpuYX','2019/02/02','Chase Center','New Orleans
Pelicans','Casa','99-98','L','10278');
INSERT INTO partita VALUES('Da83V','2019/01/25','Fedex Forum','Memphis
Grizzlies','Trasf','89-103','L','18200');
INSERT INTO partita VALUES('u3LuW','2018/12/25','Chase Center','Memphis
Grizzlies','Casa','106-96','L','11560');
INSERT INTO partita VALUES('k2Mt5','2018/11/24','Spectrum Center','Charlotte
Hornets','Trasf','97-108','L','10210');
INSERT INTO partita VALUES('XNaw8','2018/10/10','Chase Center','Charlotte
Hornets','Casa','91-99','W','10900');
INSERT INTO partita VALUES('A892m','2018/04/01','Capital One Arena','Washington
Wizards','Trasf','96-99','L','19230');
INSERT INTO partita VALUES('chcaG','2018/03/09','Chase Center','Washington
Wizards','Casa','102-99','L','11250');
INSERT INTO partita VALUES('VUB7Q','2018/03/05','Golden 1 Center','Sacramento
Kings','Trasf','72-103','L','10768');
INSERT INTO partita VALUES('VrK4p','2018/01/10','Chase Center','Sacramento
Kings','Casa','100-92','L','8900');
INSERT INTO partita VALUES('6D8Wb','2017/12/04','Fiserv Forum','Milwaukee
Bucks','Trasf','94-110','L','17700');
INSERT INTO partita VALUES('Sxxtx','2017/11/30','Staples Center','Los Angeles
Lakers','Trasf','91-87','W','19900');
INSERT INTO partita VALUES('h6U77','2017/10/21','Chase Center','Los Angeles
Lakers','Casa','91-101','W','10234');
INSERT INTO partita VALUES('cubCj','2017/09/09','Staples Center','Los Angeles
Clippers','Trasf','113-102','W','19780');
INSERT INTO partita VALUES('8iKQR','2017/09/27','Chase Center','Los Angeles
Clippers','Casa','111-109','L','13890');
INSERT INTO partita VALUES('wCCCF','2017/03/29','Barclays Center','Brooklyn
Nets','Trasf','91-121','L','18700');
INSERT INTO partita VALUES('dLbL8','2017/03/15','Chase Center','Brooklyn
Nets','Casa','102-89','L','13900');
INSERT INTO partita VALUES('Bftpb','2017/02/11','Madison Square Garden','New York
Knicks','Trasf','99-100','L','19900');
INSERT INTO partita VALUES('yuyjq','2017/01/04','Chase Center','New York
Knicks','Casa','103-90','L','14900');

```

Progetto realizzato da Luca Montanari e Davide Rivi

```

INSERT INTO partita VALUES('BhJyL','2017/12/03','TD Banknorth Garden','Boston
celtics','Trasf','100-121','L','18750');
INSERT INTO partita VALUES('6hitj','2017/02/03','Chase Center','Boston
Celtics','Casa','117-115','L','14900');
INSERT INTO partita VALUES('WxfbY','2017/01/20','Pepsi Center','Denver
Nuggets','Trasf','80-89','L','19000');
INSERT INTO partita VALUES('4DZyE','2016/12/27','Chase Center','Denver
Nuggets','Casa','90-87','L','12450');
INSERT INTO partita VALUES('RYvMi','2016/11/10','AT&T Center','San Antonio
Spurs','Trasf','102-87','W','18000');
INSERT INTO partita VALUES('LVxmp','2016/10/25','Chase Center','San Antonio
Spurs','Casa','90-99','W','12100');
INSERT INTO partita VALUES('XY9Bh','2016/09/20','Amway Center','Orlando
Magic','Trasf','90-120','L','15230');
INSERT INTO partita VALUES('iAray','2016/04/02','Moda Center','Portland Trail
Blazers','Trasf','97-99','L','16000');
INSERT INTO partita VALUES('r2TjF','2016/03/02','Chase Center','Portland Trail
Blazers','Casa','107-101','L','13100');
INSERT INTO partita VALUES('p4adU','2016/02/12','American Airlines Arena','Miami
Heat','Trasf','78-100','L','16200');
INSERT INTO partita VALUES('AVnqR','2016/01/17','Chase Center','Miami
Heat','Casa','98-97','L','10900');
INSERT INTO partita VALUES('i8kJz','2015/12/19','Wells Fargo Center','Philadelphia
79ers','Trasf','103-111','L','20870');
INSERT INTO partita VALUES('zEH9a','2015/11/23','Chase Center','Philadelphia
79ers','Casa','100-102','W','14900');
INSERT INTO partita VALUES('DcBfm','2015/10/15','American Airlines Center','Dallas
Mavericks','Trasf','98-92','W','18000');
INSERT INTO partita VALUES('kPwUi','2015/09/21','Chase Center','Dallas
Mavericks','Casa','100-97','L','13740');
INSERT INTO partita VALUES('uyMRz','2018/1/14','Scotiabank Arena','Toronto
Raptors','Trasf','125-127','L','14500');
INSERT INTO partita VALUES('TRM46','2017/10/18','Chase Center','Houston
Rockets','Casa','121-122','W','13290');
INSERT INTO partita VALUES('y8dHQ','2021/03/01','Staples Center','Los Angeles
Lakers','Trasf','91-117','L','16900');
INSERT INTO partita VALUES('AnMPT','2019/05/21','Moda Center','Portland Trail
Blazers','Trasf','119-117','W','16203');
INSERT INTO statistiche VALUES('W4tWQ','0lbP86MJ55','2','2','3','15');
INSERT INTO statistiche VALUES('W4tWQ','62QyuZSCtJ','12','2','10','20');
INSERT INTO statistiche VALUES('W4tWQ','sRELjEAY4w','20','3','15','25');
INSERT INTO statistiche VALUES('W4tWQ','bqm7YByEKB','14','5','9','25');
INSERT INTO statistiche VALUES('W4tWQ','ZUsJLQ2Rdd','19','6','6','30');
INSERT INTO statistiche VALUES('W4tWQ','bMSzjEuUWr','35','9','7','35');
INSERT INTO statistiche VALUES('jfinN','bqm7YByEKB','25','5','9','25');
INSERT INTO statistiche VALUES('jfinN','ZUsJLQ2Rdd','20','6','6','30');
INSERT INTO statistiche VALUES('jfinN','bMSzjEuUWr','29','9','7','35');
INSERT INTO statistiche VALUES('jfinN','VD0sEGq4gv','10','2','2','15');

```

Progetto realizzato da Luca Montanari e Davide Rivi

```

INSERT INTO statistiche VALUES('jfinN','MvXBlu4TxJ','6','6','8','13');
INSERT INTO statistiche VALUES('jfinN','cf9Xf2hz5w','4','2','2','10');
INSERT INTO statistiche VALUES('LXTXT','bqm7YByEKB','25','6','9','22');
INSERT INTO statistiche VALUES('LXTXT','ZUsJLQ2Rdd','30','3','2','29');
INSERT INTO statistiche VALUES('LXTXT','bMSzjEuUWr','30','9','7','36');
INSERT INTO statistiche VALUES('LXTXT','fWqaexj9a9','5','2','9','19');
INSERT INTO statistiche VALUES('LXTXT','m2rHYWpbFY','10','2','3','15');
INSERT INTO statistiche VALUES('LXTXT','pSi62RRLtc','2','1','3','7');
INSERT INTO statistiche VALUES('DatSa','bqm7YByEKB','25','3','6','29');
INSERT INTO statistiche VALUES('DatSa','bMSzjEuUWr','30','3','6','25');
INSERT INTO statistiche VALUES('DatSa','asoLihn79I','15','2','7','24');
INSERT INTO statistiche VALUES('DatSa','62QyuZSCtJ','6','0','10','14');
INSERT INTO statistiche VALUES('DatSa','2VQL7NA3Uh','6','2','2','10');
INSERT INTO statistiche VALUES('EQ6QP','bqm7YByEKB','20','5','12','32');
INSERT INTO statistiche VALUES('EQ6QP','ZUsJLQ2Rdd','33','3','2','26');
INSERT INTO statistiche VALUES('EQ6QP','bMSzjEuUWr','40','10','3','35');
INSERT INTO statistiche VALUES('EQ6QP','3LVdbbwjyQ','9','4','2','13');
INSERT INTO statistiche VALUES('EQ6QP','0IbP86MJ55','13','4','10','20');
INSERT INTO statistiche VALUES('EQ6QP','fWqaexj9a9','7','1','2','8');
INSERT INTO statistiche VALUES('qNUKG','bqm7YByEKB','12','3','6','21');
INSERT INTO statistiche VALUES('qNUKG','ZUsJLQ2Rdd','20','0','8','24');
INSERT INTO statistiche VALUES('qNUKG','bMSzjEuUWr','25','7','5','25');
INSERT INTO statistiche VALUES('qNUKG','8MHVmuuZRn','15','3','12','20');
INSERT INTO statistiche VALUES('qNUKG','asoLihn79I','14','3','3','19');
INSERT INTO statistiche VALUES('qNUKG','pSi62RRLtc','2','0','2','5');
INSERT INTO statistiche VALUES('VZFgW','bqm7YByEKB','29','4','2','28');
INSERT INTO statistiche VALUES('VZFgW','bMSzjEuUWr','39','11','5','34');
INSERT INTO statistiche VALUES('VZFgW','asoLihn79I','6','1','7','10');
INSERT INTO statistiche VALUES('VZFgW','fWqaexj9a9','7','0','4','7');
INSERT INTO statistiche VALUES('VZFgW','2VQL7NA3Uh','2','2','0','6');
INSERT INTO statistiche VALUES('7gnWW','bqm7YByEKB','25','2','10','26');
INSERT INTO statistiche VALUES('7gnWW','ZUsJLQ2Rdd','20','6','2','23');
INSERT INTO statistiche VALUES('7gnWW','bMSzjEuUWr','31','6','2','27');
INSERT INTO statistiche VALUES('7gnWW','3LVdbbwjyQ','12','2','3','17');
INSERT INTO statistiche VALUES('7gnWW','MvXBlu4TxJ','9','5','4','24');
INSERT INTO statistiche VALUES('7gnWW','mix2pWfebx','5','5','3','12');
INSERT INTO statistiche VALUES('hpPZe','bqm7YByEKB','25','5','9','27');
INSERT INTO statistiche VALUES('hpPZe','ZUsJLQ2Rdd','20','6','6','24');
INSERT INTO statistiche VALUES('hpPZe','bMSzjEuUWr','31','9','7','20');
INSERT INTO statistiche VALUES('hpPZe','cf9Xf2hz5w','12','2','4','20');
INSERT INTO statistiche VALUES('hpPZe','m2rHYWpbFY','13','3','6','18');
INSERT INTO statistiche VALUES('hpPZe','pSi62RRLtc','14','2','3','25');
INSERT INTO statistiche VALUES('3Byqj','bqm7YByEKB','32','5','9','23');
INSERT INTO statistiche VALUES('3Byqj','ZUsJLQ2Rdd','29','6','6','29');
INSERT INTO statistiche VALUES('3Byqj','bMSzjEuUWr','15','9','7','32');
INSERT INTO statistiche VALUES('3Byqj','mix2pWfebx','7','8','4','27');
INSERT INTO statistiche VALUES('3Byqj','0IbP86MJ55','13','2','9','17');
INSERT INTO statistiche VALUES('3Byqj','VD0sEGq4gv','13','6','6','19');

```

Progetto realizzato da Luca Montanari e Davide Rivi

```

INSERT INTO statistiche VALUES('2jVVw','bqm7YByEKB','18','5','9','28');
INSERT INTO statistiche VALUES('2jVVw','ZUsJLQ2Rdd','22','6','6','17');
INSERT INTO statistiche VALUES('2jVVw','bMSzjEuUWr','23','9','7','29');
INSERT INTO statistiche VALUES('2jVVw','mix2pWfebx','5','2','6','10');
INSERT INTO statistiche VALUES('2jVVw','cf9Xf2hz5w','10','7','3','16');
INSERT INTO statistiche VALUES('2jVVw','asoLihn79I','20','2','7','25');
INSERT INTO statistiche VALUES('PBJx5','bqm7YByEKB','30','8','4','29');
INSERT INTO statistiche VALUES('PBJx5','ZUsJLQ2Rdd','22','6','4','22');
INSERT INTO statistiche VALUES('PBJx5','bMSzjEuUWr','35','10','6','28');
INSERT INTO statistiche VALUES('PBJx5','3LVdbbwjyQ','6','8','2','19');
INSERT INTO statistiche VALUES('PBJx5','0lbP86MJ55','9','3','3','20');
INSERT INTO statistiche VALUES('PBJx5','m2rHYWpbFY','10','10','7','28');
INSERT INTO statistiche VALUES('2vHdK','bqm7YByEKB','20','2','4','27');
INSERT INTO statistiche VALUES('2vHdK','2VQL7NA3Uh','15','1','7','19');
INSERT INTO statistiche VALUES('2vHdK','bMSzjEuUWr','18','6','2','22');
INSERT INTO statistiche VALUES('2vHdK','MvXBlu4TxJ','10','2','5','18');
INSERT INTO statistiche VALUES('2vHdK','62QyuZSCtJ','20','3','12','27');
INSERT INTO statistiche VALUES('2vHdK','fWqaexj9a9','5','5','8','15');
INSERT INTO statistiche VALUES('u4eBf','bqm7YByEKB','25','8','4','27');
INSERT INTO statistiche VALUES('u4eBf','nmlweSiPZz','12','6','4','19');
INSERT INTO statistiche VALUES('u4eBf','bMSzjEuUWr','50','10','6','22');
INSERT INTO statistiche VALUES('u4eBf','8MHVmuuZRn','8','4','3','15');
INSERT INTO statistiche VALUES('u4eBf','asoLihn79I','4','4','6','10');
INSERT INTO statistiche VALUES('u4eBf','2VQL7NA3Uh','4','2','1','8');
INSERT INTO statistiche VALUES('STJd4','bqm7YByEKB','17','6','2','19');
INSERT INTO statistiche VALUES('STJd4','asoLihn79I','19','9','6','25');
INSERT INTO statistiche VALUES('STJd4','bMSzjEuUWr','22','13','2','28');
INSERT INTO statistiche VALUES('STJd4','2VQL7NA3Uh','7','3','5','12');
INSERT INTO statistiche VALUES('STJd4','VD0sEGq4gv','19','6','5','24');
INSERT INTO statistiche VALUES('STJd4','62QyuZSCtJ','11','3','9','19');
INSERT INTO statistiche VALUES('6puVb','bqm7YByEKB','18','4','4','22');
INSERT INTO statistiche VALUES('6puVb','bMSzjEuUWr','25','7','4','27');
INSERT INTO statistiche VALUES('6puVb','MvXBlu4TxJ','5','2','5','10');
INSERT INTO statistiche VALUES('6puVb','8MHVmuuZRn','12','3','6','20');
INSERT INTO statistiche VALUES('6puVb','asoLihn79I','10','3','9','16');
INSERT INTO statistiche VALUES('uFwWn','bqm7YByEKB','20','2','1','26');
INSERT INTO statistiche VALUES('uFwWn','bMSzjEuUWr','37','2','3','31');
INSERT INTO statistiche VALUES('uFwWn','m2rHYWpbFY','11','2','4','19');
INSERT INTO statistiche VALUES('uFwWn','MvXBlu4TxJ','5','3','2','9');
INSERT INTO statistiche VALUES('uFwWn','62QyuZSCtJ','15','4','3','22');
INSERT INTO statistiche VALUES('uyMRz','bMSzjEuUWr','24','9','6','34');
INSERT INTO statistiche VALUES('uyMRz','dKOasXiVAz','25','5','6','37');
INSERT INTO statistiche VALUES('uyMRz','sRELjEAY4w','14','8','3','38');
INSERT INTO statistiche VALUES('uyMRz','ZUsJLQ2Rdd','26','0','4','39');
INSERT INTO statistiche VALUES('uyMRz','dcSojPIACq','4','0','12','21');
INSERT INTO statistiche VALUES('TRM46','bMSzjEuUWr','22','4','5','30');
INSERT INTO statistiche VALUES('TRM46','dKOasXiVAz','20','7','5','37');
INSERT INTO statistiche VALUES('TRM46','ZUsJLQ2Rdd','16','3','6','38');

```

Progetto realizzato da Luca Montanari e Davide Rivi

```

INSERT INTO statistiche VALUES('TRM46','sRELjEAY4w','14','8','12','31');
INSERT INTO statistiche VALUES('TRM46','dcSojPIACq','4','0','8','11');
INSERT INTO statistiche VALUES('y8dHQ','bMSzjEuUWr','16','7','4','26');
INSERT INTO statistiche VALUES('y8dHQ','sRELjEAY4w','6','2','1','13');
INSERT INTO statistiche VALUES('y8dHQ','fWqaexj9a9','12','3','6','33');
INSERT INTO statistiche VALUES('y8dHQ','62QyuZSCtJ','3','5','15','36');
INSERT INTO statistiche VALUES('y8dHQ','asoLihn79I','19','7','5','32');
INSERT INTO statistiche VALUES('AnMPT','bMSzjEuUWr','37','11','13','47');
INSERT INTO statistiche VALUES('AnMPT','sRELjEAY4w','18','11','14','43');
INSERT INTO statistiche VALUES('AnMPT','ZUsJLQ2Rdd','17','2','6','46');
INSERT INTO statistiche VALUES('AnMPT','cf9Xf2hz5w','7','2','1','11');
INSERT INTO statistiche VALUES('AnMPT','dKOasXiVAz','25','12','5','34');

```

## Query di Interrogazione

-- Visualizzare i giocatori che sono alti più di 205 cm con ruolo "ala piccola"

```

SELECT P.COGNOME,P.NOME,G.*
FROM GIOCATORE G, PERSONA P
WHERE ALTEZZA>2.05 AND RUOLO='Ala Piccola' AND P.CF=G.CF;

```

-- Visualizzare gli stadi in cui si è giocato una partita con almeno 12000 spettatori, vinta, e ordinarli per capienza

```

SELECT DISTINCT S.NOME,S.N_POSTI
FROM STADIO S, PARTITA P
WHERE S.NOME=P.NOME_STADIO AND P.BIGLIETTI_VENDUTI>=12000 AND
P.ESITO='W'
ORDER BY 2 DESC;

```

-- Visualizzare il budget disponibile di ogni stagione

```

SELECT STAGIONE,DISPONIBILE
FROM BUDGET
ORDER BY STAGIONE;

```

-- Visualizzare il nome del giocatore, il numero di maglia e le sue statistiche medie

```

SELECT P.COGNOME,P.NOME,G.N_MAGLIA, AVG(S.PUNTI) AS PPP, AVG(S.ASSIST)
AS APP, AVG(S.RIMBALZI) AS RPP, AVG(S.MINUTI) AS MPP
FROM GIOCATORE G, PERSONA P, STATISTICHE S
WHERE P.CF=G.CF AND S.CF_G=G.CF
GROUP BY P.COGNOME,P.NOME,G.N_MAGLIA;

```

-- Visualizzare il nome del giocatore che ha giocato più partite nella squadra

```

SELECT P.COGNOME,P.NOME,G.N_MAGLIA
FROM GIOCATORE G, PERSONA P, STATISTICHE S
WHERE P.CF=G.CF AND S.CF_G=G.CF
GROUP BY P.COGNOME,P.NOME,G.N_MAGLIA
HAVING COUNT(*) >= ALL
(
  SELECT COUNT(*)
  FROM STATISTICHE
  GROUP BY CF_G
);

```

-- Visualizzare per ogni preparatore, il numero di sotto-preparatori e gli allenamenti svolti

```

SELECT P.TESSERA, (SELECT COUNT(*) FROM PREPARATORE P1 WHERE
P1.TESSERA_CAPO=P.TESSERA) AS SOTTOPREP, A.TIPO,A.DATA
FROM PREPARATORE P, ALLENAMENTO A
WHERE P.TESSERA=A.TESSERA
GROUP BY P.TESSERA,A.TIPO,A.DATA
ORDER BY P.TESSERA;

```

-- Visualizzare gli infortuni e ordinarli per gravità

```

SELECT *
FROM INFORTUNIO
ORDER BY GRAVITA;

```

-- Visualizzare il numero dei giocatori con status "Bencher" e il totale dei loro stipendi in tutte le stagioni

```

SELECT COUNT(DISTINCT G.CF), SUM(C.STIPENDIO)
FROM GIOCATORE G, CONTRATTO C
WHERE STATUS='Bencher' AND G.CF=C.CF_P;

```

-- Visualizzare i giocatori che non hanno mai realizzato una doppia doppia (doppia cifra in 2 statistiche)

```

SELECT P.CF,P.NOME,P.COGNOME,G.N_MAGLIA
FROM GIOCATORE G, PERSONA P
WHERE G.CF=P.CF AND G.CF NOT IN(
  SELECT CF_G
  FROM STATISTICHE
  WHERE (PUNTI>9 AND ASSIST>9) OR (PUNTI>9 AND RIMBALZI>9) OR (RIMBALZI>9
AND ASSIST>9)) ;

```

-- Visualizzare i giocatori ceduti e raggrupparli per ruolo

```

SELECT P.*,G.RUOLO

```

Progetto realizzato da Luca Montanari e Davide Rivi

```
FROM PERSONA P, GIOCATORE G, TRASFERIMENTO T
WHERE T.TIPOLOGIA='Cessione' AND P.CF=G.CF AND T.CF_G=G.CF
GROUP BY G.RUOLO,P.CF ;
```

-- Visualizzare i campionati vinti con i rispettivi allenatori

```
SELECT P.NOME,P.COGNOME, CV.ANNO, CV.NOME_CAMP
FROM ALLENATORE A, PERSONA P, CONTRATTO C, CAMPIONATI_VINTI CV
WHERE A.CF=P.CF AND P.CF=C.CF_P AND CV.ANNO BETWEEN C.ANNO_FIRMA
AND C.ANNO_FIRMA+DURATA;
```

-- Visualizzare i giocatori che hanno giocato almeno 10 minuti in tutte le partite del 2021

```
SELECT *
FROM GIOCATORE G
WHERE NOT EXISTS(
SELECT *
FROM PARTITA P
WHERE DATA BETWEEN '2021/01/01' and '2021/12/31' AND NOT EXISTS(
SELECT *
FROM STATISTICHE S
WHERE S.MINUTI>=10 AND P.COD=S.COD_P AND S.CF_G = G.CF ));
```

## Query di Modifica

-- Aumento di stipendio per la persona con il cf seguente nel 2021

```
UPDATE Contratto
SET stipendio = stipendio + 10000
WHERE CF_P='asoLihn79I' and anno_firma=2021;
```

-- Cambio di status per il giocatore con la maglia numero 23

```
UPDATE Giocatore
SET status = 'Starter'
where N_Maglia=23;
```

-- Aumento di punti del giocatore con CF='fWQaexj9a9' nella partita con  
cod\_Partita='VZFgW'

```
UPDATE Statistiche
SET Punti = Punti+5
WHERE Cod_P='VZFgW' and CF_G='fWQaexj9a9';
```

-- Acquisto del merchandising del giocatore con cf 'bMSzjEuUWr' di 50 qta, utilizzando  
funzione acquisto

```
select Acquisto('bMSzjEuUWr',50);
Progetto realizzato da Luca Montanari e Davide Rivi
```



## Query di Eliminazione

-- Eliminazione del giocatore che ha il numero 19

```
Delete FROM Giocatore WHERE N_Maglia = 19;
```

-- Eliminazione della partita giocata il 21/01/2022 al 'Madison Square Garden'

```
Delete FROM Partita WHERE Data ='2022/01/21' AND Nome_Stadio = 'Madison Square Garden';
```

-- Eliminazione di tutti i contratti che non superano i 750000 di stipendio

```
Delete FROM Contratto WHERE Stipendio<750000;
```