

# MotoGP Championship

Progetto di una base di dati per la gestione di un  
campionato di MotoGP.

Nicola Montanari

Nicola.Montanari14@studio.unibo.it

0000970119

<b>1</b>	<b>Analisi dei Requisiti</b>	<b>2</b>
1.1	Intervista . . . . .	3
1.2	Estrazione dei concetti Principali . . . . .	4
<b>2</b>	<b>Progettazione Concettuale</b>	<b>6</b>
2.1	Team e Componenti . . . . .	6
2.1.1	Schema scheletro . . . . .	6
2.1.2	Raffinamenti proposti . . . . .	6
2.2	Campionato e Gare . . . . .	8
2.2.1	Schema scheletro . . . . .	8
2.2.2	Raffinamenti proposti . . . . .	8
2.3	Autodromi e Biglietti . . . . .	10
2.3.1	Schema scheletro . . . . .	10
2.3.2	Raffinamenti proposti . . . . .	10
2.4	Schema concettuale finale . . . . .	11
<b>3</b>	<b>Progettazione Logica</b>	<b>14</b>
3.1	Stima del volume dei dati . . . . .	14
3.2	Descrizione delle operazioni principali e stima della loro frequenza . . . . .	15
3.3	Schemi di navigazione e tabelle degli accessi . . . . .	16
3.4	Raffinamento dello schema . . . . .	26
<b>4</b>	<b>Analisi Delle Ridondanze</b>	<b>27</b>
4.1	Traduzione di entità e associazioni in relazioni . . . . .	28
4.2	Schema relazionale finale . . . . .	29
4.3	Traduzione delle operazioni in query SQL . . . . .	32
<b>5</b>	<b>Progettazione dell'applicazione</b>	<b>39</b>
5.1	Descrizione dell'architettura dell'applicazione real- izzata . . . . .	39

# Analisi dei Requisiti

Si vuole creare un applicativo che permetta la gestione e l'automatizzazione di un Campionato di MotoGP, ovvero una serie di gare di moto nell'arco di un anno solare. La base di dati dovrà immagazzinare e gestire le informazioni relative alle squadre partecipanti e alle gare effettuate, comprendendo anche la gestione degli autodromi con relativi visitatori, oltre al sistema di punteggio che decreterà i vincitori del campionato piloti e di quello dei costruttori.

## 1.1 Intervista

Un primo testo ottenuto dall'intervista è il seguente:

L'attuale gestore e organizzatore dei più famosi campionati mondiali di Moto (Dorna) richiede che venga creato un applicativo utile a gestire un'intero campionato di MotoGP. In un anno solare verranno svolte numerose gare, ospitate dai vari autodromi in giro per il mondo. Ogni gara è caratterizzata dal numero di giri da effettuare, è richiesto inoltre di memorizzare la classifica di arrivo generale e il tempo totale che ogni pilota impiega ad effettuare i predeterminati giri di gara. La base di dati dovrà contenere informazioni relative sia ai vari team partecipanti al campionato che ai loro relativi componenti. Ogni team ha una struttura precisa e definita in cui presenta esattamente due piloti, un gommista, un telemetrista e tre Ingegneri Meccanici. Ogni pilota parteciperà ad ogni gara presente nel calendario del campionato grazie ai restanti componenti del team che prepareranno la moto per correre. Alla fine di ogni anno questo campionato è caratterizzato da un mercato dei vari componenti di ogni team, è richiesto quindi di memorizzare uno storico relativo a quest'ultimi. All'inizio di ogni anno ogni team dovrà scegliere uno tra i vari costruttori disponibili, quest'ultimo diventerà il fornitore ufficiale del team per tutta la durata della stagione. Siccome Dorna possiede il diritto alla vendita dei vari biglietti per ogni evento, sarà necessario mantenere in memoria e gestire le informazioni riguardanti i posti disponibili per ogni autodromo, i biglietti da vendere/venduti ai vari visitatori che verranno a vedere le gare di MotoGP. Per ogni gara effettuata verrà assegnato un punteggio ai primi 15 classificati di tale evento con un pattern preciso: 25, 20, 16, 13, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1. Utilizzando questo sistema di punti, a fine stagione si dovrà decretare il vincitore del campionato piloti e quello dei costruttori.

## 1.2 Estrazione dei concetti Principali

A seguito della lettura e comprensione dei requisiti richiesti dal cliente, si procede sviluppando un testo che ne riassume tutti i concetti e in particolare ne estragga quelli principali, risultando essere in questo modo meglio fruibile per la realizzazione della base di dati. Si tiene conto delle seguenti correzioni di ambiguità.

Termine	Breve Descrizione	Eventuali sinonimi
Campionato	Serie di competizioni ufficiali tra i vari team partecipanti	Stagione, Mondiale.
Gara	Singolo evento in cui i Piloti di ogni Team gareggiano con lo scopo di effettuare un prestabilito numero di giri nel minor tempo possibile	Evento, Competizione, Weekend.
Autodromo	Grande struttura di edifici che comprende il tracciato, le tribune e i box.	Pista, Tracciato.
Spettatore	Persona fisica che compra un biglietto per vedere una gara di MotoGP.	Visitatore, Fan.
Team	Gruppo di persone che lavorano insieme per massimizzare le performance per ogni evento.	Squadra, Gruppo.
Pilota	Persona fisica che guida la moto durante ogni competizione.	Corridore, Componente di un team.
Telemetrista	Persona fisica che analizza i dati ottenuti durante la guida del pilota.	Componente di un team.
Gommista	Persona fisica che gestisce i dati e le decisioni riguardanti le gomme.	Componente di un team.
Ingegnere Meccanico	Persona fisica che lavora sulla moto per massimizzare le performance.	Componente di un team.
Costruttore	Azienda che sviluppa, produce e fornisce tutti i componenti di una moto.	Produttore.

## Testo Riformulato

Per ogni **componente di un team** vengono memorizzati nome, cognome, data di nascita e codice fiscale. Ogni team, relativi componenti e **costruttori** possiedono un codice univoco fornitogli al momento della registrazione al sistema. Ogni **team** ha una struttura ben definita e non possono esserci componenti diversi da: 2 **Piloti**, 3 **Ingegneri meccanici**, 1 **Telemetrista** e 1 **Gommista**. Ogni struttura dei vari team dovrà essere memorizzata tramite uno **storico**. Ciascun team farà affidamento a un singolo **costruttore** iscritto al Campionato. Una **Stagione** comprende numerose **gare**, tutte ufficiali siccome incideranno sulle **due classifiche finali**. Ogni Team può partecipare a tutte le gare e ogni competizione può far partecipare un numero infinito di team. Ogni gara è ospitata da un **Autodromo** preciso, un autodromo può ospitare una sola gara ogni anno. Per ogni evento, l'Autodromo disporrà dei **biglietti** da vendere ai vari **visitatori** che vorranno vedere la gara. Alla fine del Mondiale, verranno decretati i **due vincitori**, il pilota con maggior punti e il costruttore con maggior punti.

### Segue un elenco delle principali azioni richieste:

1. Registrare un nuovo componente.
2. Creare un nuovo team.
3. Sostituire un componente di un team.
4. Aggiungere un costruttore.
5. Assegnare un costruttore a un team.
6. Visualizzare uno storico dei componenti di un team.
7. Creare un nuovo Campionato.
8. Iscrivere un team ad un Campionato.
9. Aggiungere un Autodromo.
10. Aggiungere una nuova gara ad un campionato.
11. Registrare l'esito di un evento.
12. Visualizzare una classifica di arrivo di un evento.
13. Calcolare il punteggio di un pilota in un evento.
14. Calcolare il punteggio di un costruttore in un evento.
15. Decretare il vincitore del campionato piloti a fine stagione.
16. Decretare il vincitore del campionato costruttori a fine stagione.
17. Vendere un biglietto per un evento.

# Progettazione Concettuale

## 2.1 Team e Componenti

### 2.1.1 Schema scheletro

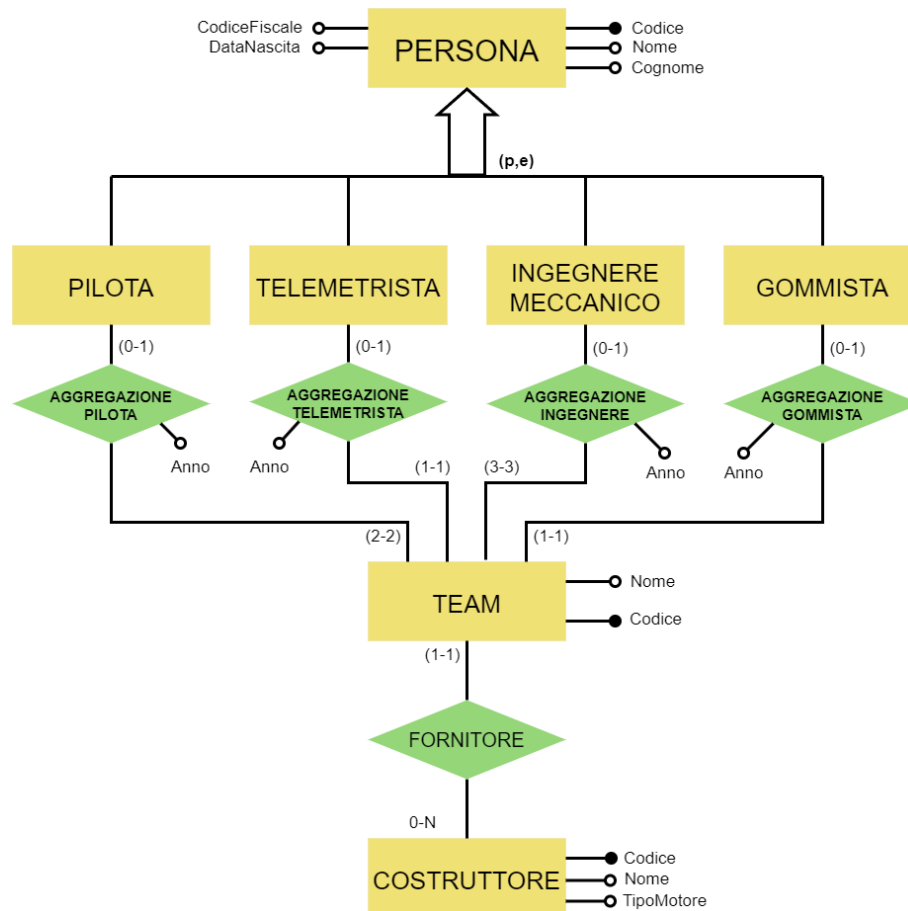


Figura 1 - Schema scheletro per i Team e i propri componenti.

### 2.1.2 Raffinamenti proposti

Ciascun entità **componente di un team** è implementata come una generalizzazione dell'entità **persona**, identificata con un codice univoco per evitare problemi di omocodia. Dallo studio del dominio si nota come la struttura di ogni team debba essere **precisa e ben definita**, infatti ciascun team sarà un'aggregazione di esattamente 7 membri:

- 2 Piloti
- 3 Ingegneri Meccanici
- 1 Telemetrista
- 1 Gommista

Ciascun elemento avrà la possibilità di poter **far parte o meno** di un team, quindi è possibile avere in memoria dati di componenti che **non fanno parte di nessuna squadra**.

Allo stesso momento, componente è vincolato a far parte di al massimo **un solo team alla volta**.

Dopo aver quindi strutturato lo schema affinché **ciascun team** rispetti la formazione richiesta, si definiscono i singoli attributi dei Team. Ciascun team avrà come attributi il proprio **nome** e un **codice univoco**; quest'ultimo sarà utilizzato come **identificativo** per differenziare ciascun team.

Ciascun team dovrà relazionarsi con un singolo **Costruttore** affinché possa partecipare al campionato. I Costruttori sono aziende private che offrono il servizio di vendita di qualsiasi componentistica necessaria per preparare una moto da competizione per partecipare al campionato, motore e telaio compresi. **Risulta essere essenziale** quindi che ciascun team abbia sottoscritto un contratto con **un singolo costruttore**, affinché possa ricevere le moto per competere durante la stagione. Ogni costruttore è caratterizzato da un nome e una tipologia di motore sviluppata da tale azienda oltre ad un **codice univoco** che verrà utilizzato per identificare ciascuna istanza.

Nonostante ci sia un vincolo da parte di ogni team affinché debba riferirsi ad un singolo costruttore, **nulla vieta ai costruttori** di poter fornire le proprie componenti **a numerosi team partecipanti allo stesso campionato**.



## 2.2 Campionato e Gare

### 2.2.1 Schema scheletro

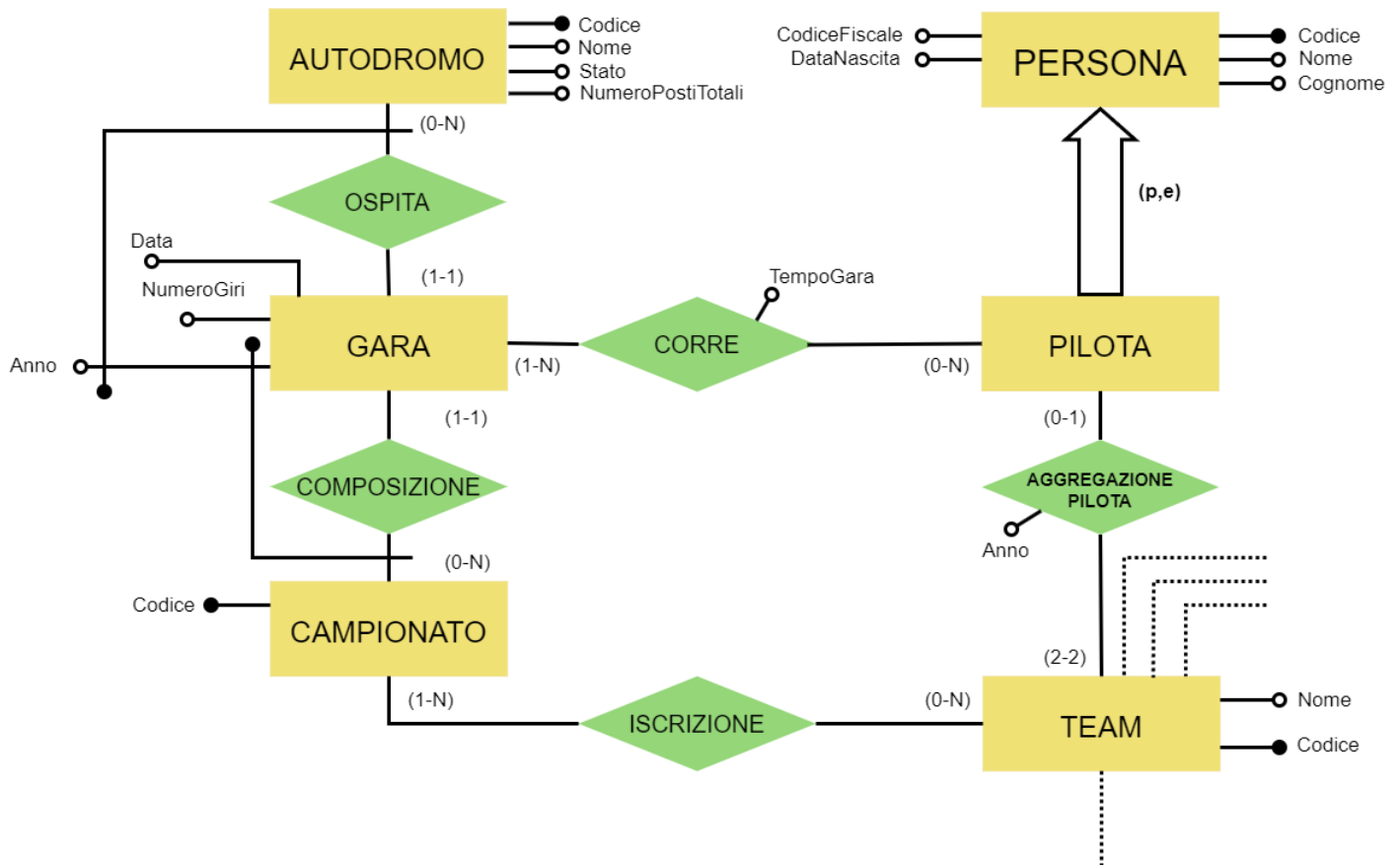


Figura 2 - Schema scheletro per i Campionati e le gare.

### 2.2.2 Raffinamenti proposti

Un campionato di MotoGP è una **composizione** di numerose gare (eventi) svolte durante un anno solare. Durante l'anno viene svolto **un solo campionato**, e ciascuna edizione è caratterizzata da un **codice univoco identificativo** per poter differenziare i mondiali tra di essi. **All'inizio** di ogni campionato, affinché possano partecipare e competere al titolo, ogni team dovrà **isciversi** a tale edizione, **definendo così i propri componenti per quella stagione**. Così facendo, saremo in grado di tenere traccia in memoria uno storico dei componenti di ciascun team.

Una gara è un'evento, caratterizzato da un **numero di giri totali**, dalla **data** e **anno** in cui viene svolta tale esibizione. Per identificare ciascuna gara, ho optato utilizzare una terna di elementi: **L'attributo Anno**, **un'entità "autodromo"** e **un'entità "campionato"**.

Così facendo, **non sarà possibile** far ospitare allo **stesso autodromo**, una gara di campionato **due volte lo stesso anno**. Questa struttura fa in modo che **molteplici gare** dello stesso campionato possano essere ospitate da **molteplici autodromi** in giro per il mondo. Allo stesso momento (nonostante venga svolto un solo campionato all'anno) è possibile far ospitare ad un singolo autodromo molteplici gare, queste però **dovranno essere tutte componenti di campionati differenti**.

L'attributo **data** è stato implementato con lo scopo di ordinare i dati in un futuro momento in cui sarà necessario richiedere l'elenco delle gare contenute all'interno di un campionato.

**Un Autodromo** è un complesso di strutture in grado di poter ospitare una gara del mondiale grazie al suo tracciato. Ogni Autodromo è identificato da un suo **codice univoco**, inoltre è caratterizzato dagli attributi **Nome**, **Stato di appartenenza** (e.g. Italia) e dal **numero di posti totali** nelle proprie tribune. Quest'ultimo campo sarà utile alla **gestione della vendita dei biglietti**.

**Ciascun pilota** partecipante al campionato, **potrà prender parte ad ogni gara** e disporrà al sistema il **tempo totale a lui necessario** per percorrere la totalità dei giri di gara predisposti.

Grazie a questi dati, confrontando i vari tempi registrati, sarà possibile andare a **generare le classifiche di ciascun evento**, **calcolare i punteggi** e quindi definire i vincitori alla fine di ogni campionato.

## 2.3 Autodromi e Biglietti

### 2.3.1 Schema scheletro

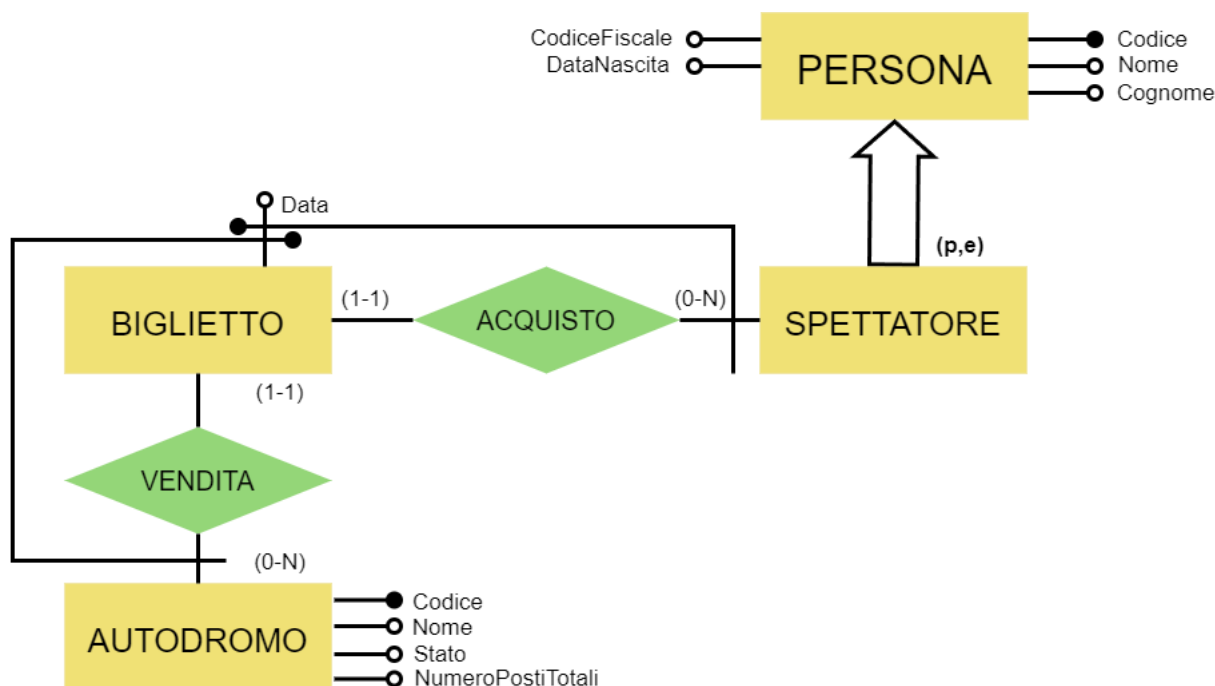


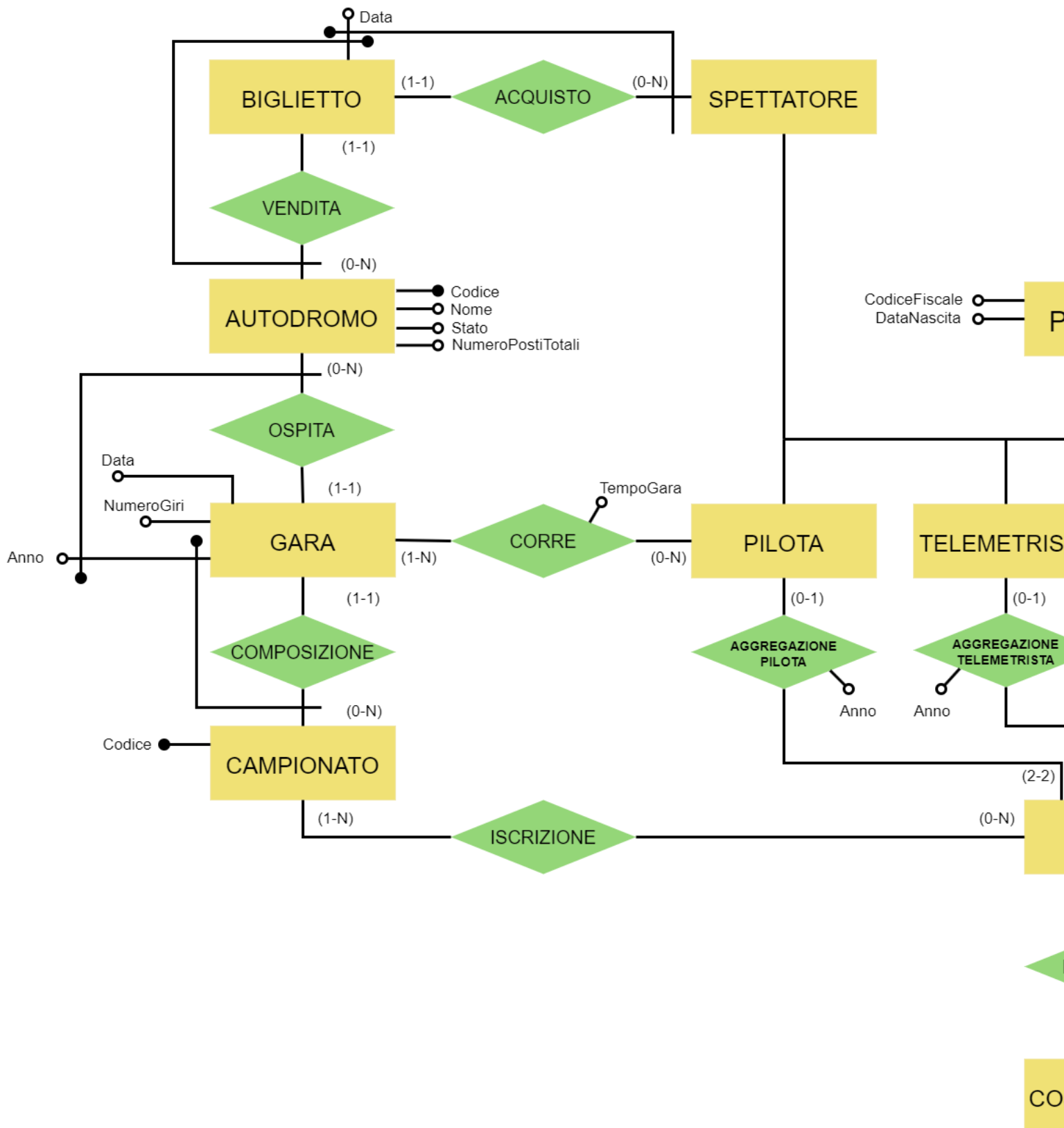
Figura 3 - Schema scheletro per gli Autodromi e i Biglietti.

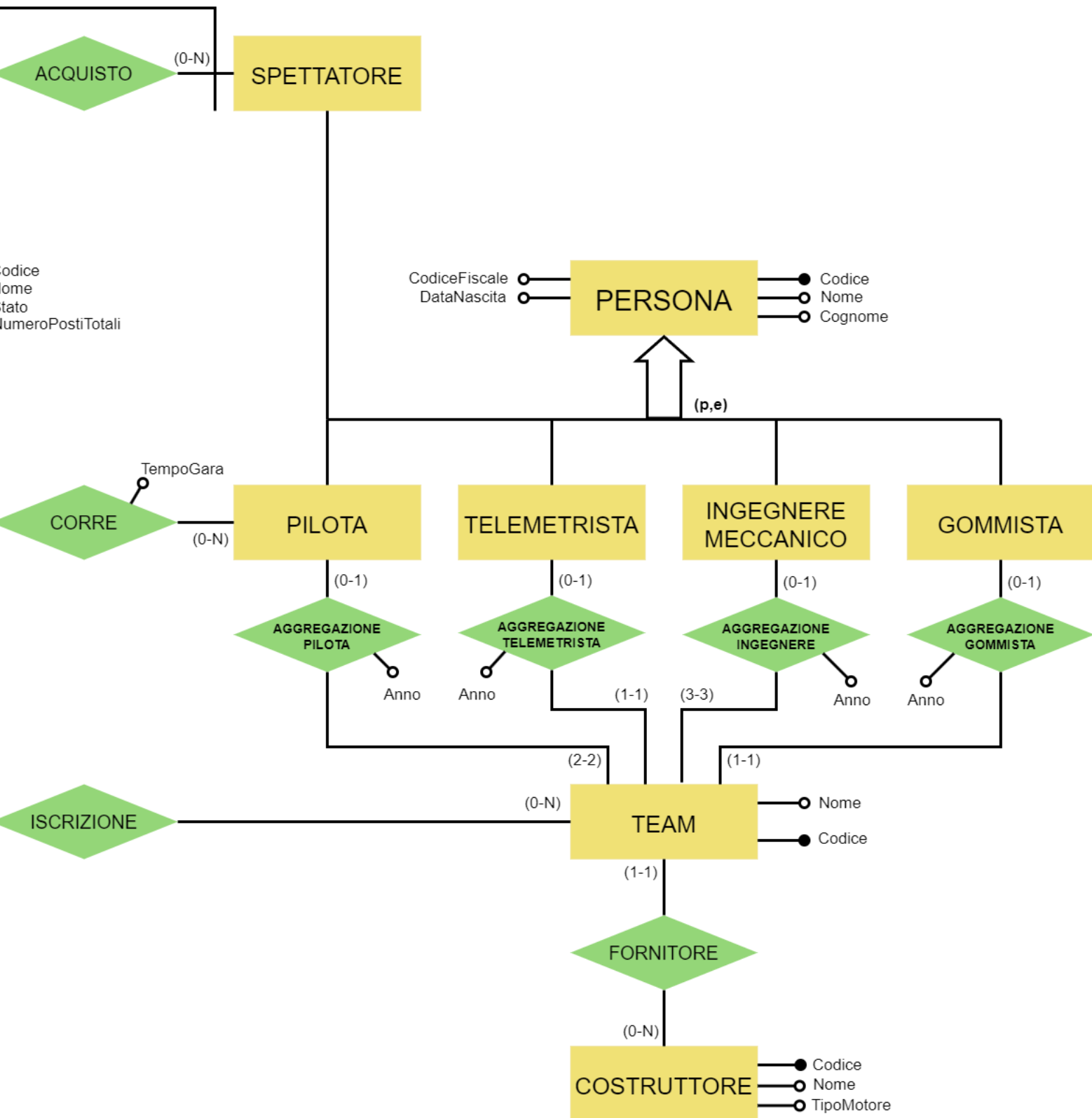
### 2.3.2 Raffinamenti proposti

Ogni autodromo predisporre nel suo complesso, numerose tribune per far assistere agli spettatori i numerosi eventi ospitati dal tracciato. Ogni **spettatore** è anch'esso una generalizzazione di **persona**, quindi identificato con un codice univoco.

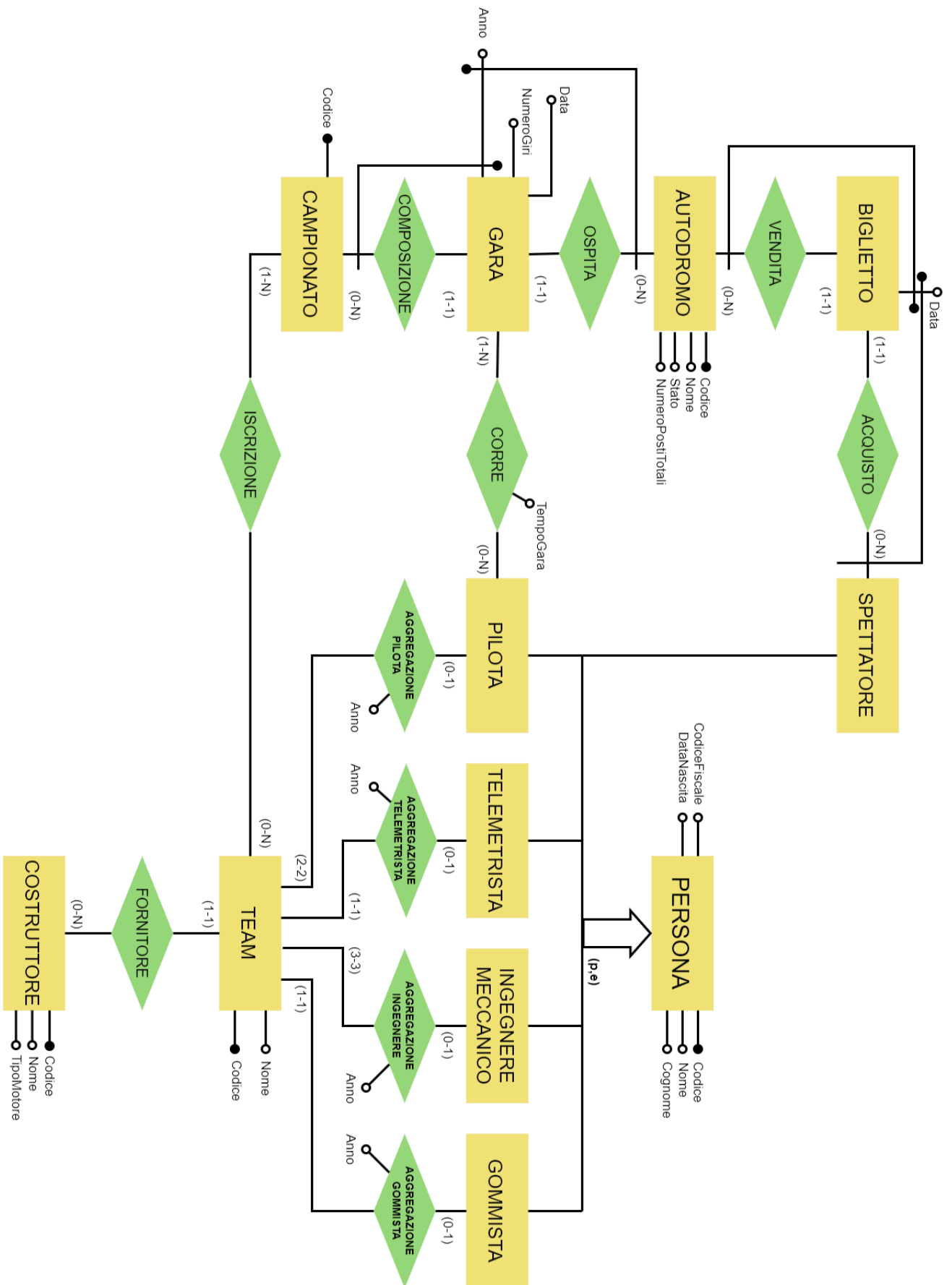
Per ogni evento, verranno venduti agli spettatori dei **biglietti** necessari ad accedere alla struttura. Ciascun autodromo predispone di un numero preciso e limitato di posti a sedere, quindi la vendita dei biglietti sarà **limitata** da essi; nel momento in cui i biglietti venduti pareggiano il numero di posti a sedere, **non sarà piu possibile acquistarne ulteriori**. Ciascun biglietto sarà **nominativo** e vincolato dalla **data** e dall'**autodromo** in cui lo si ha acquistato, questa terna verrà utilizzata per **identificare i vari biglietti**. Un biglietto quindi, potrà essere comprato da un singolo spettatore, utilizzato in un singolo autodromo in una precisa data. Nulla vieta ad uno spettatore di comprare molteplici biglietti, per esempio in **altri autodromi** o per **altre date**.

## 2.4 Schema concettuale finale





## Schema finale Unico



# Progettazione Logica

## 3.1 Stima del volume dei dati

Tabella dei Volumi

Concetto	Costrutto	Volume
Pilota	E	30
Telemetrista	E	15
Gommista	E	15
Ingegnere Meccanico	E	45
Team	E	15
Costruttore	E	5
Fornitore	R	15
AggregazionePilota	R	30
AggregazioneTelemetrista	R	15
AggregazioneGommista	R	15
AggregazioneIngegnere	R	45
Iscrizione	R	15
Campionato	E	1
Gara	E	20
Autodromo	E	20
ComposizioneCampionato	R	20
Ospita	R	20
Corre	R	600
Spettatore	E	1.500.000
Biglietto	E	2.000.000
Vendita	R	2.000.000
Acquisto	R	2.000.000

### 3.2 Descrizione delle operazioni principali e stima della loro frequenza

Le operazioni da effettuare sono quelle già elencate nella fase di analisi. Segue una tabella riportante la relativa **frequenza annuale**:

Codice	Operazione	Frequenza annuale
1	Registrare un nuovo componente	10
2	Creare un nuovo team	2
3	Sostituire un componente di un team	10
4	Aggiungere un costruttore	1
5	Assegnare un costruttore a un team	15
6	Visualizzare uno storico dei componenti di un team	1
7	Creare un nuovo Campionato	1
8	Iscrivere un team ad un Campionato	15
9	Aggiungere un Autodromo	1
10	Aggiungere una nuova gara ad un campionato	20
11	Registrare l'esito di un evento	20
12	Visualizzare una classifica di arrivo di un evento	20
13	Calcolare il punteggio di un pilota	600
14	Calcolare il punteggio di un costruttore	100
15	Decretare il vincitore del campionato piloti a fine stagione	1
16	Decretare il vincitore del campionato costruttori a fine stagione	1
17	Vendere un biglietto per un evento	2.000.000



### 3.3 Schemi di navigazione e tabelle degli accessi

Sono riportate in seguito le tabelle degli accessi delle operazioni sopra riportate; inoltre, ove non risulti banale, sono stati inseriti i relativi **schemi di navigazione**. Al fine del calcolo degli costi, si considerano di **peso doppio** gli accessi in scrittura rispetto a quelli in lettura.

#### NOTA:

Alcune operazioni possono essere effettuate allo stesso modo per **ciascun componente di un team**, per chiarezza elencati qua sotto:

- Pilota
- Ingegnere Meccanico
- Telemetrista
- Gommista

Per evitare inutili ripetizioni, nelle tabelle si utilizzerà il termine generico **"Componente"**.

#### OP-1: Registrare un nuovo componente

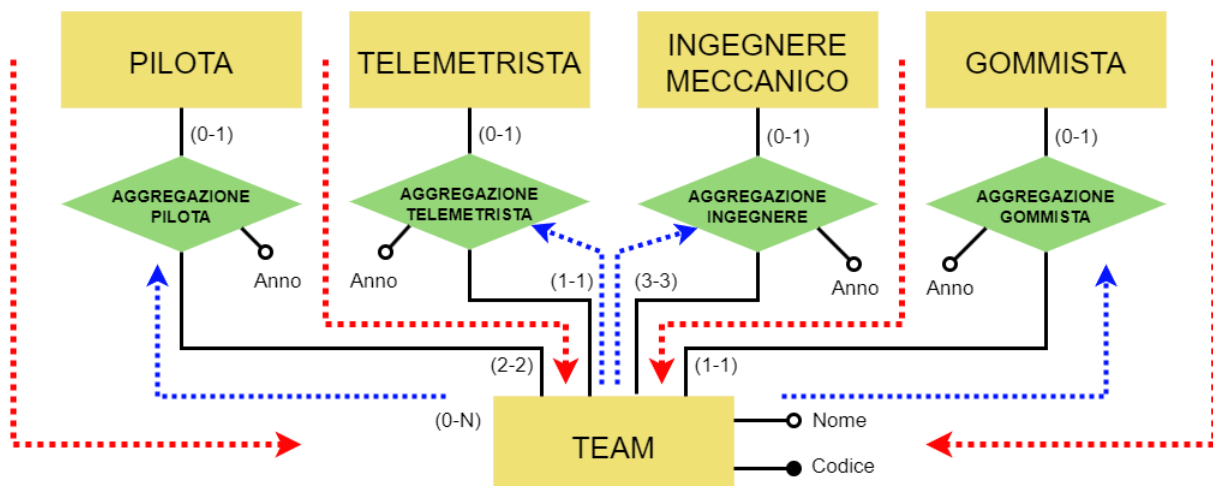
Questa operazione può essere effettuata allo stesso modo per ciascun componente di un team.

Concetto	Costrutto	Accessi	Tipo
Componente	E	1	S
Totale: $1S + 0L \rightarrow 2$			
Frequenza: 10 all'anno			
Costo Totale: 20 all'anno			

## OP-2: Creare un nuovo team

Per poter portare a termine questa operazione, è necessario che tutti i componenti **siano già registrati** all'interno del sistema.

Una volta letti i dati dei componenti e creato effettivamente l'istanza dell'entità "Team" si andranno a generare anche le varie relazioni di aggregazione per ogni componente.



Concetto	Costrutto	Accessi	Tipo
Pilota	E	2	L
Telemetrista	E	1	L
Gommista	E	1	L
Ingegnere	E	3	L
Team	E	1	S
Aggr.Pilota	R	2	S
Aggr.Telemetrista	R	1	S
Aggr.Ingegnere	R	3	S
Aggr.Gommista	R	1	S
Totale: 8S + 7L → 23			
Frequenza: 2 all'anno			
Costo Totale: 46 all'anno			

### OP-3: Sostituire un componente di un team

Anche in questo caso, questa operazione può essere effettuata allo stesso modo per ciascun componente di un team.

La relazione chiamata letteralmente "Aggr.Componente" non esiste, ma indica la **relativa relazione in base al componente da sostituire**.

Supponendo di conoscere sia il **codice del componente** che quello del **team** da inserire nella tabella sarà sufficiente una scrittura **dell'entità "Aggr.Componente"**.

Prima di effettuare questa operazione è essenziale assicurarsi che il nuovo componente **non faccia già parte di qualche team**.

Concetto	Costrutto	Accessi	Tipo
Aggr.Componente	R	1	S
Totale: $1S + 0L \rightarrow 2$			
Frequenza: 10 all'anno			
Costo Totale: 20 all'anno			

### OP-4: Aggiungere un costruttore

Concetto	Costrutto	Accessi	Tipo
Costruttore	E	1	S
Totale: $1S + 0L \rightarrow 2$			
Frequenza: 1 all'anno			
Costo Totale: 2 all'anno			

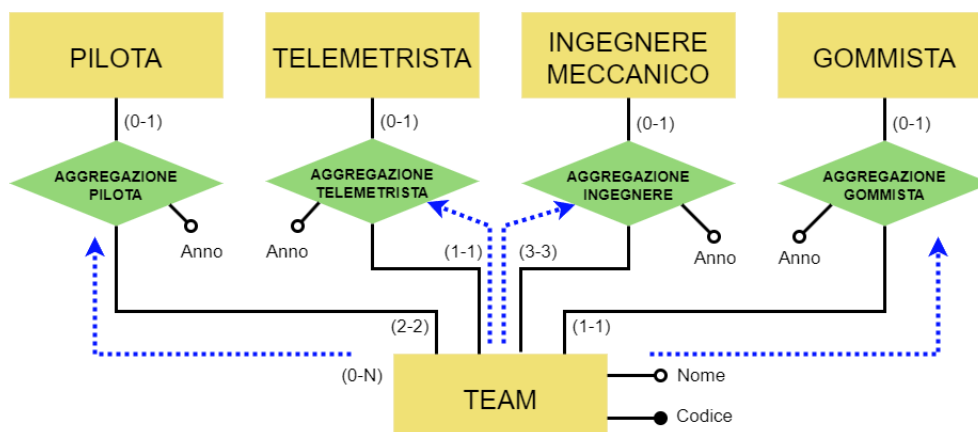
**OP-5: Assegnare un costruttore a un team**

Supponendo di conoscere sia il **codice del costruttore** che del **team** da inserire nella tabella sarà sufficiente una scrittura dell'entità **Fornitore**.

Concetto	Costrutto	Accessi	Tipo
Fornitore	R	1	S
Totale: $1S + 0L \rightarrow 2$			
Frequenza: 15 all'anno			
Costo Totale: 30 all'anno			

**OP-6: Visualizzare uno storico dei componenti di un team**

Si suppone di conoscere il **codice del team** preso in considerazione e l'anno.



Concetto	Costrutto	Accessi	Tipo
Aggr.Pilota	R	2	L
Aggr.Telemetrista	R	1	L
Aggr.Ingegnere	R	3	L
Aggr.Gommista	R	1	L
Totale: $0S + 7L \rightarrow 7$			
Frequenza: 1 all'anno			
Costo Totale: 7 all'anno			

### OP-7: Creare un nuovo Campionato

Concetto	Costrutto	Accessi	Tipo
Campionato	E	1	S
Totale: $1S + 0L \rightarrow 2$			
Frequenza: 1 all'anno			
Costo Totale: 2 all'anno			

### OP-8: Iscrivere un team ad un Campionato

Supponendo di conoscere sia il **codice del team** che quello del **campionato** da inserire nella tabella sarà sufficiente una scrittura **dell'entità Iscrizione**.

Concetto	Costrutto	Accessi	Tipo
Iscrizione	R	1	S
Totale: $1S + 0L \rightarrow 2$			
Frequenza: 15 all'anno			
Costo Totale: 30 all'anno			

### OP-9: Aggiungere un Autodromo

Concetto	Costrutto	Accessi	Tipo
Autodromo	E	1	S
Totale: $1S + 0L \rightarrow 2$			
Frequenza: 1 all'anno			
Costo Totale: 2 all'anno			

### OP-10: Aggiungere una nuova gara ad un campionato

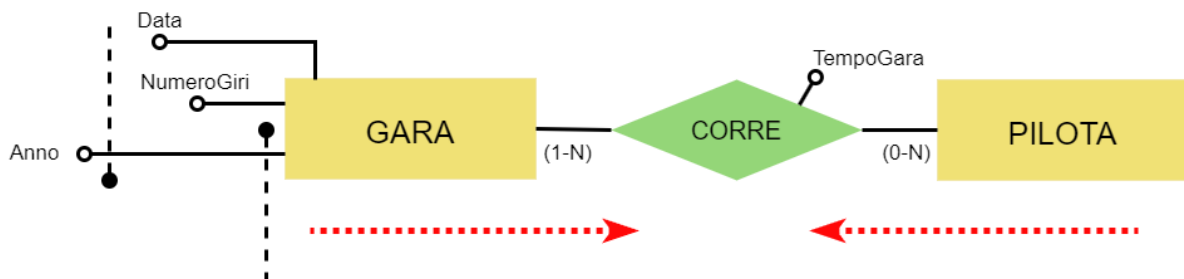
Supponendo di conoscere sia il **codice dell'autodromo** che quello del **campionato** da inserire nella tabella sarà sufficiente una scrittura dell'entità "Gara".

Concetto	Costrutto	Accessi	Tipo
Gara	E	1	S
Totale: $1S + 0L \rightarrow 2$			
Frequenza: 20 all'anno			
Costo Totale: 40 all'anno			

### OP-11: Registrare l'esito di un evento

Si suppone di conoscere il **codice della gara** presa in considerazione.

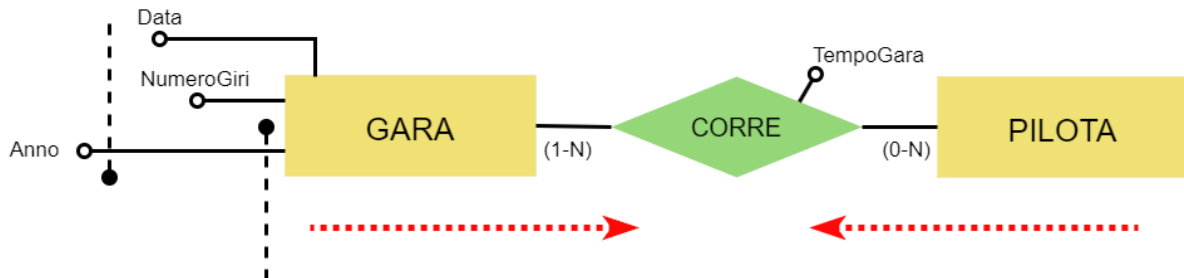
**Ciascun pilota** fa registrare il proprio tempo necessario a percorrere la totalità dei giri di gara all'interno dell'entità "Corre", grazie a questo valore si riuscirà a ordinare i vari piloti partecipanti e quindi creare una classifica di arrivo.



Concetto	Costrutto	Accessi	Tipo
Pilota	E	30	L
Corre	R	30	S
Totale: $30S + 30L \rightarrow 90$			
Frequenza: 20 all'anno			
Costo Totale: 1800 all'anno			

### OP-12: Visualizzare una classifica di arrivo di un evento

Si suppone di conoscere il **codice della gara** presa in considerazione.  
Si utilizza l'attributo "**TempoGara**" all'interno dell'entità "**Corre**" per calcolare l'ordine di arrivo.



Concetto	Costrutto	Accessi	Tipo
Corre	R	30	L
Pilota	E	30	L
Totale: 0S + 60L → 60			
Frequenza: 20 all'anno			
Costo Totale: 1200 all'anno			

### OP-13: Calcolare il punteggio di un pilota in un evento

Si suppone di conoscere sia il **codice della gara** che quello del **pilota** preso in considerazione.

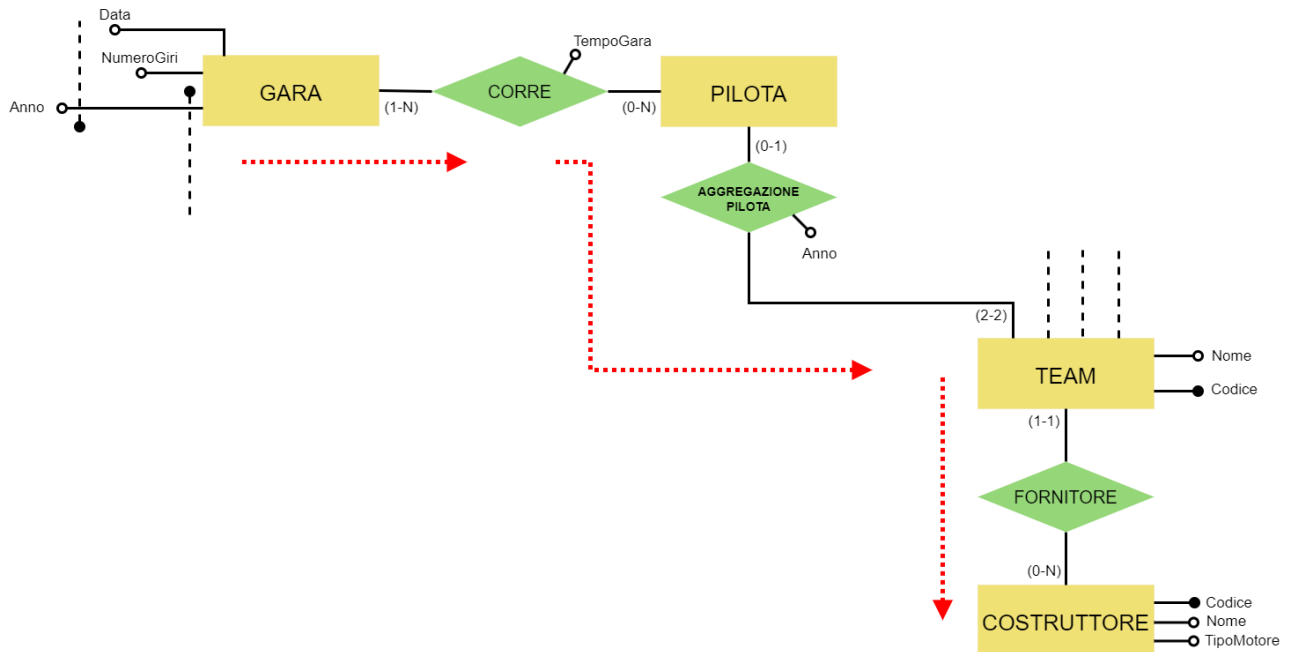
Risulta essere necessario leggere solo i primi 15 arrivati in una gara siccome il **restante gruppo non guadagna punti**.

Concetto	Costrutto	Accessi	Tipo
Corre	R	15	L
Totale: 0S + 15L → 15			
Frequenza: 600 all'anno			
Costo Totale: 9000 all'anno			

### OP-14: Calcolare il punteggio di un costruttore in un evento

Si suppone di conoscere sia il **codice della gara** che quello del **costruttore** preso in considerazione.

Risulta essere necessario leggere solo i primi 15 arrivati in una gara siccome il **restante gruppo non guadagna punti**.



Concetto	Costrutto	Accessi	Tipo
Corre	R	15	L
Pilota	E	15	L
Aggr.Pilota	E	15	L
Team	E	15	L
Fornitore	R	15	L
Costruttore	E	15	L
Totale: 0S + 90L → 90			
Frequenza: 100 all'anno			
Costo Totale: 9000 all'anno			



**OP-15: Decretare il vincitore del campionato piloti a fine stagione**

Si suppone di conoscere il **codice del campionato** preso in considerazione.

Questa operazione risulta essere **esattamente l'OP-13**, ma **ripetuta per ogni pilota** partecipante al campionato, **sommando i vari punti** di ogni gara all'interno del mondiale. E' necessario leggere solo i primi 15 piloti nella classifica di ogni gara siccome i **restanti non guadagnano punti**. **Il vincitore sarà il pilota con più punti.**

Concetto	Costrutto	Accessi	Tipo
Gara	E	20	L
Corre	R	300	L
Pilota	E	30	L
Totale: 0S + 350L → 350			
Frequenza: 1 all'anno			
Costo Totale: 350 all'anno			

### OP-16: Decretare il vincitore del campionato costruttori a fine stagione

Si suppone di conoscere il **codice del campionato** preso in considerazione.

Questa operazione risulta essere **esattamente l'OP-14**, ma **ripetuta per ogni pilota** partecipante al campionato, **sommando i vari punti** di ogni gara all'interno del mondiale **ai vari costruttori** presenti nell'elenco. **Il vincitore sarà il costruttore con più punti.**

Concetto	Costrutto	Accessi	Tipo
Gara	E	20	L
Corre	R	300	L
Pilota	E	30	L
Aggr.Pilota	E	30	L
Team	E	30	L
Fornitore	R	30	L
Costruttore	E	30	L
Totale: 0S + 470L $\rightarrow$ 470			
Frequenza: 1 all'anno			
Costo Totale: 470 all'anno			

### OP-17: Vendere un biglietto per un evento

Supponendo di conoscere sia il **codice dell'autodromo** che quello dello **spettatore** da inserire nella tabella sarà sufficiente una scrittura dell'entità **"Biglietto"**.

Concetto	Costrutto	Accessi	Tipo
Biglietto	E	1	S
Totale: 1S + 0L $\rightarrow$ 2			
Frequenza: 2.000.000 all'anno			
Costo Totale: 4.000.000 all'anno			

### 3.4 Raffinamento dello schema

#### Eliminazione delle gerarchie

Per l'eliminazione della gerarchia persona si è scelto di adottare l'approccio del **collasso verso il basso**, replicando così gli attributi in spettatore e in ciascun componente di un team. Si è adottata questa strategia in quanto si deve interagire con gli spettatori molto più spesso che con il resto, e non si ha la necessità che l'identificatore per tali entità sia globalmente univoco.

#### Scelta delle chiavi primarie

Nello schema sono già evidenziate senza ambiguità tutte le chiavi primarie per la maggior parte delle entità; per quanto riguarda l'entità **GARE**, si sceglie di usare come chiave primaria **l'anno, l'associazione con AUTODROMO e l'associazione con CAMPIONATO**. Infine, per l'entità **BIGLIETTO**, la chiave primaria sarà data dalla **data, dall'associazione con SPETTATORE e con l'associazione con AUTODROMO**.

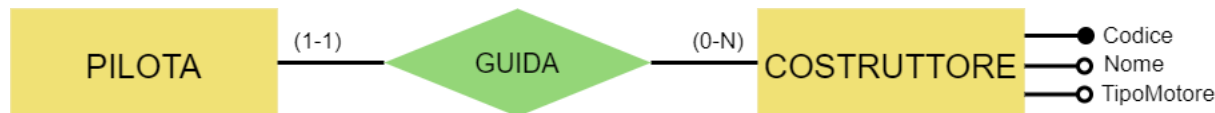
#### Eliminazione di identificatori esterni

Nello schema E/R sono state effettuate le seguenti operazioni di rimozione:

- Relazione "**Ospita**" tra **AUTODROMO** e **GARA**; con conseguente importazione della chiave esterna "CodiceAutodromo" nell'entità GARA
- Relazione "**ComposizioneCampionato**" tra **CAMPIONATO** e **GARA**; con conseguente importazione della chiave esterna "CodiceCampionato" nell'entità GARA
- Relazione "**Vendita**" tra **AUTODROMO** e **BIGLIETTO**; con conseguente importazione della chiave esterna "CodiceAutodromo" nell'entità BIGLIETTO
- Relazione "**Acquisto**" tra **SPETTATORE** e **BIGLIETTO**; con conseguente importazione della chiave esterna "CodiceSpettatore" nell'entità BIGLIETTO

# Analisi Delle Ridondanze

Una **possibile** ridondanza applicabile su questo dominio, sarebbe la creazione di una nuova relazione **"GUIDA"** che unisca le entità **"PILOTA"** e **"COSTRUTTORE"**. Così facendo, in fase di calcolo del punteggio per ogni costruttore, che sia dopo una gara o a fine campionato, si risparmierebbero parecchi passaggi siccome **non è più necessario** ricercare il costruttore attraverso il team di appartenenza di ogni pilota. Osservando però lo schema di navigazione e la tabella degli accessi proposti per le varie operazioni, si evince che l'aggiunta di questa nuova relazione gioverebbe nell'effettuazione della singola operazione di calcolo del punteggio, ma **comporterebbe una spesa di tempo ulteriore** sia in fase di creazione di un nuovo team, sia nel momento in cui un pilota volesse cambiare squadra, soprattutto per il fatto che in quest'ultimo caso ciascun pilota **non potrà più essere "trattato" come un qualsiasi componente** di un team siccome avrà necessità di operazioni aggiuntive. Per questi motivi l'aggiunta **non viene fatta**.



## OP-14: Calcolare il punteggio di un costruttore in un evento

Si suppone di conoscere sia il **codice della gara** che quello del **costruttore** preso in considerazione.

Concetto	Costrutto	Accessi	Tipo
Corre	R	15	L
Pilota	E	15	L
<b>GUIDA</b>	R	15	L
Costruttore	E	15	L
Totale: 0S + 60L → 60			
Frequenza: 100 all'anno			
Costo Totale: 600 all'anno			

## 4.1 Traduzione di entità e associazioni in relazioni

### Entità

Piloti(**CodicePilota**, Nome, Cognome, CodiceFiscale, DataNascita)

Telemetrismi(**CodiceTelemetrista**, Nome, Cognome, CodiceFiscale, DataNascita)

IngegneriMeccanici(**CodiceIngegnere**, Nome, Cognome, CodiceFiscale, DataNascita)

Gommisti(**CodiceGommista**, Nome, Cognome, CodiceFiscale, DataNascita)

Spettatori(**CodiceSpettatore**, Nome, Cognome, CodiceFiscale, DataNascita)

Team(**CodiceTeam**, Nome)

Costruttori(**CodiceCostruttore**, Nome, TipoMotore)

Autodromi(**CodiceAutodromo**, Nome, Stato, NumeroPostiTotali)

Campionati(**CodiceCampionato**)

Biglietti(**Data**, **CodiceSpettatore** : Spettatore, **CodiceAutodromo** : Autodromo)

Gare(**Anno**, **CodiceCampionato** : Campionato, **CodiceAutodromo** : Autodromo, Data, NumeroGiri)

### Associazioni

Corre((**Anno**, **CodiceCampionato**, **CodiceAutodromo**) : Gara, **CodicePilota** : Pilota, TempoGara)

Iscrizioni(**CodiceCampionato** : Campionato, **CodiceTeam** : Team)

Fornitori(**CodiceCostruttore** : Costruttore, **CodiceTeam** : Team)

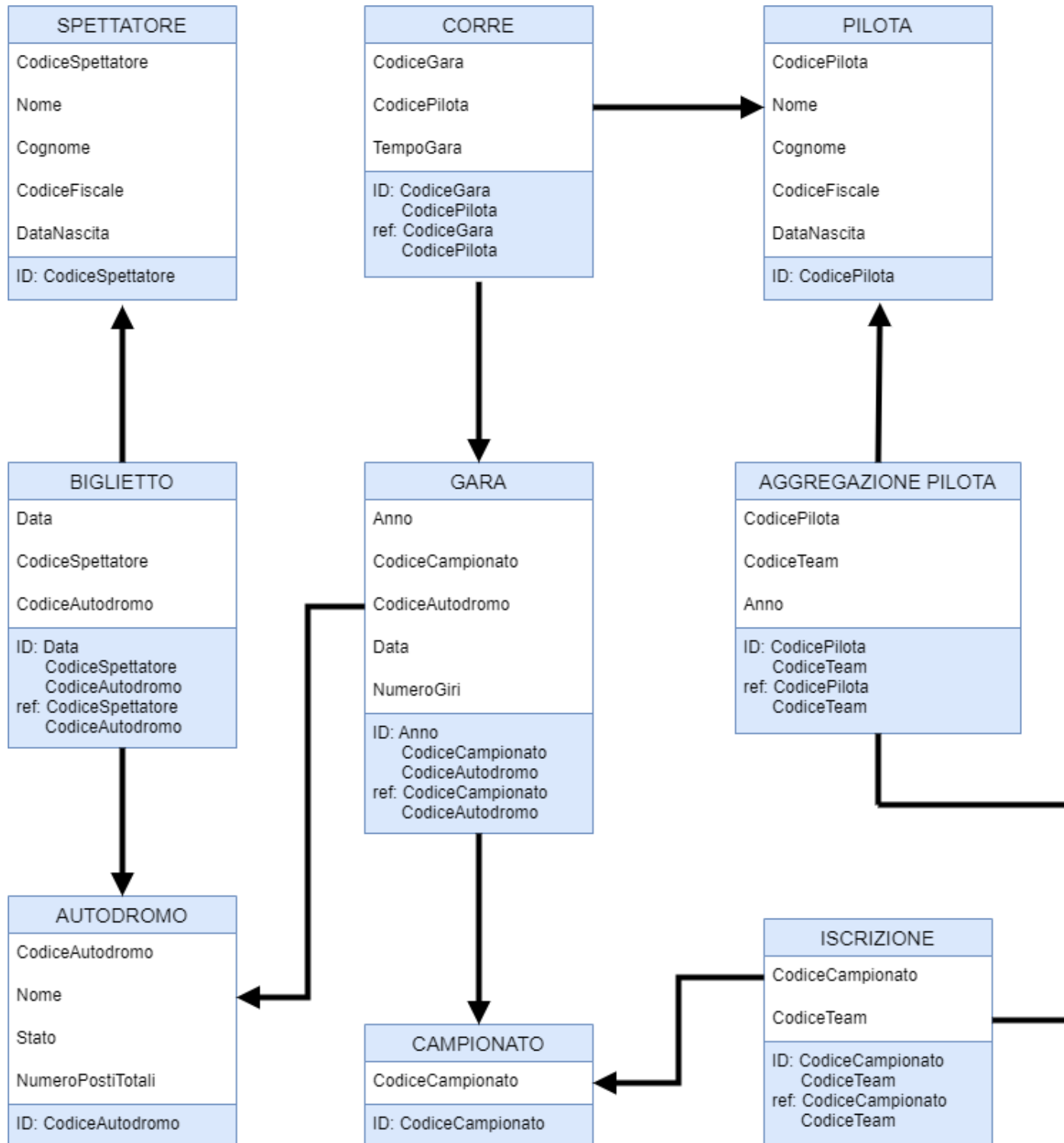
AggregazionePiloti(**CodicePilota** : Pilota, **CodiceTeam** : Team, Anno)

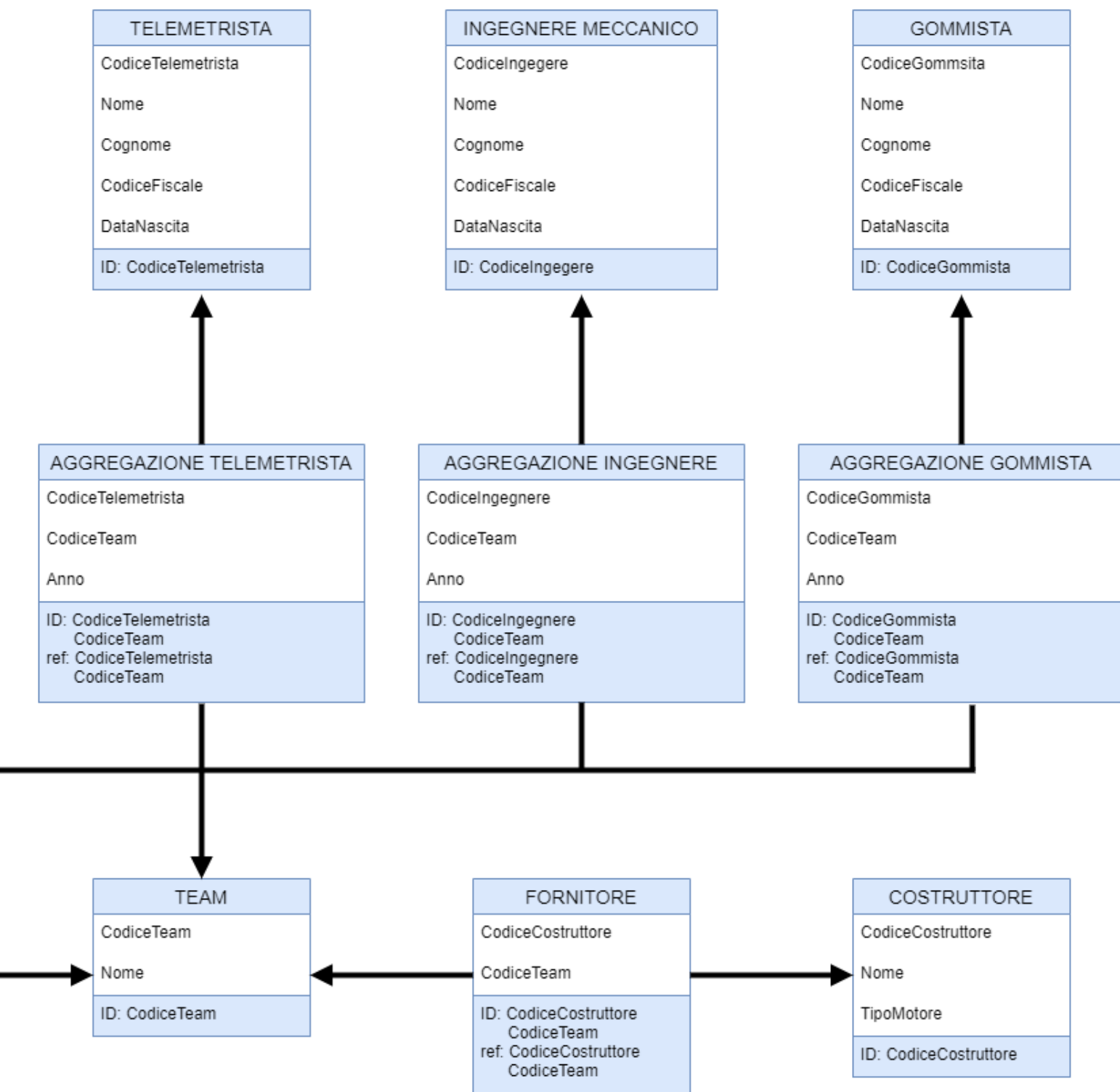
AggregazioneTelemetrismi(**CodiceTelemetrista** : Telemetrista, **CodiceTeam** : Team, Anno)

AggregazioneIngegneri(**CodiceIngegnere** : Ingegnere, **CodiceTeam** : Team, Anno)

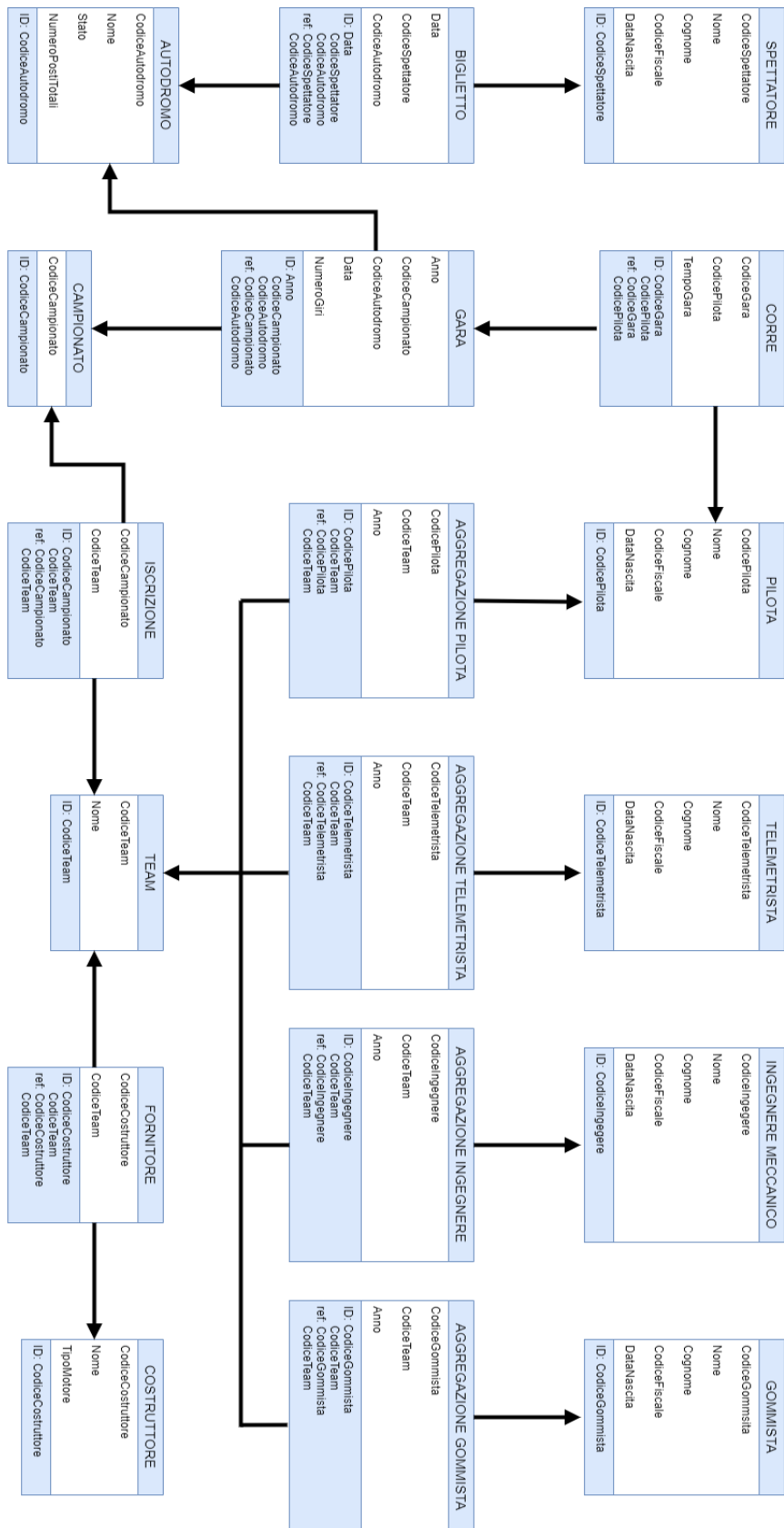
AggregazioneGommisti(**CodiceGommista** : Gommista, **CodiceTeam** : Team, Anno)

## 4.2 Schema relazionale finale





## Schema relazionale finale Unico





### 4.3 Traduzione delle operazioni in query SQL

#### OP-1: Registrare un nuovo componente.

Questa operazione può essere effettuata **allo stesso modo per ciascun componente di un team.**

In questo caso verrà utilizzata un'entità "Pilota" come **esempio.**

```
INSERT INTO piloti (nome, cognome, codiceFiscale, dataNascita)
VALUES (?, ?, ?, ?)
```

#### OP-2: Creare un nuovo team.

```
INSERT INTO Team (nome)
VALUES (?)
```

#### OP-3: Sostituire un componente di un team.

Questa operazione può essere effettuata **allo stesso modo per ciascun componente di un team.**

In questo caso verrà utilizzata un'entità "Pilota" come **esempio.**

Supponendo di sapere già il **codice del componente da aggiungere.**

Prima di tutto si verifica che il nuovo componente **non faccia già parte di un team.**

Si verificherà quindi che la seguente query **non produci risultati:**

```
SELECT *
FROM piloti JOIN aggregazionepiloti
ON piloti.CodicePilota = aggregazionepiloti.CodicePilota
WHERE piloti.CodicePilota = ? AND aggregazionepiloti.Anno =
YEAR(NOW())
```

Una volta che la verifica risulti essere **confermata**, verrà aggiunta la nuova relazione. Siccome le modifiche alla struttura di ogni team potranno essere fatte solo **all'inizio della stagione**, non sarà necessario eliminare la vecchia relazione, in quanto risulterà essenziale in un secondo momento in cui si vorrà **visualizzare uno storico del team.**

```
INSERT INTO aggregazionepiloti (codicePilota, codiceTeam, anno)
VALUES (?, ?, YEAR(NOW()))
```

**OP-4: Aggiungere un costruttore.**

```
INSERT INTO Costruttori (nome, tipoMotore)
VALUES (?, ?)
```

**OP-5: Assegnare un costruttore a un team.**

Supponendo di sapere già il **codice del costruttore** e del **team**.

Prima di tutto si **rimuove** un eventuale contratto tra il team e **un altro costruttore**.

```
DELETE FROM fornitori
WHERE CodiceTeam = ?;
```

Poi si andrà ad **aggiungere la nuova relazione** tra il team e il costruttore preso in considerazione.

```
INSERT INTO fornitori (codiceCostruttore, codiceTeam)
VALUES (?, ?)
```

**OP-6: Visualizzare uno storico dei componenti di un team.**

Questa operazione può essere effettuata **allo stesso modo per ciascun componente di un team**.

In questo caso verrà utilizzata un'entità "Pilota" come **esempio**.

```
SELECT *
FROM team
JOIN aggregazionepiloti ON aggregazionepiloti.CodiceTeam =
team.CodiceTeam
WHERE team.CodiceTeam = ? AND aggregazionepiloti.Anno = ?
```

**OP-7: Creare un nuovo Campionato.**

```
INSERT INTO Campionati ()
VALUES ()
```

**OP-8: Iscrivere un team ad un Campionato.**

```
INSERT INTO iscrizioni (codiceCampionato, codiceTeam)
VALUES (?, ?)
```

**OP-9: Aggiungere un Autodromo.**

```
INSERT INTO Autodromi (nome, stato, numeroPostiTotali)
VALUES (?, ?, ?)
```

**OP-10: Aggiungere una nuova gara ad un campionato.**

Supponendo di sapere già il **codice dell'autodromo e del campionato**.

```
INSERT INTO Gare (Anno, codiceCampionato, codiceAutodromo, numeroGiri, data)
VALUES (?, ?, ?, ?, ?)
```

**OP-11: Registrare l'esito di un evento.**

Supponendo di sapere già sia il **codice del pilota che quelli dell'autodromo e del campionato**; questi ultimi due grazie ad una **gara** già presente all'interno del Database.

```
INSERT INTO corre (anno, codiceCampionato, codiceAutodromo, codicePilota, tempoGara)
VALUES (YEAR(NOW()), ?, ?, ?, ?)
```

Questa operazione verrà **ripetuta per ogni pilota partecipante alla gara**.

**OP-12: Visualizzare una classifica di arrivo di un evento.**

Supponendo di sapere già il **codice dell'autodromo e del campionato**, oltre che **l'anno di svolgimento** della gara.

```
SELECT *
FROM corre
WHERE corre.codiceCampionato = ? AND corre.codiceAutodromo = ?
AND corre.anno = ?
ORDER BY corre.tempoGara ASC
```

### OP-13: Calcolare il punteggio di un pilota in un evento.

Supponendo di sapere già il **codice del pilota** e i dati per individuare la gara, ovvero il **codice dell'autodromo**, del **campionato** e l'**anno di svolgimento**.

*Solo i primi 15 arrivati ottengono punti, secondo lo standard stabilito nell'analisi.*

```
WITH Classifica AS
(
SELECT corre.TempoGara AS 'Tempo', corre.CodicePilota,
ROW_NUMBER() OVER (ORDER BY corre.TempoGara ASC ) AS 'Po-
sizione'
FROM corre
WHERE corre.CodiceCampionato = ? AND corre.CodiceAutodromo = ?
AND corre.Anno = ?
)
SELECT Classifica.Posizione AS 'Posizione', CASE
WHEN Classifica.Posizione = 1 THEN '25'
WHEN Classifica.Posizione = 2 THEN '20'
WHEN Classifica.Posizione = 3 THEN '16'
WHEN Classifica.Posizione = 4 THEN '13'
WHEN Classifica.Posizione <= 15 THEN 16-Classifica.Posizione
ELSE 0
END AS 'Punti',
piloti.Nome AS 'Nome', piloti.Cognome AS 'Cognome', Classifica.Tempo
AS 'Tempo Gara', piloti.CodicePilota AS 'Codice Pilota'
FROM Classifica INNER JOIN piloti ON piloti.CodicePilota = Classi-
fica.codicePilota
WHERE Classifica.CodicePilota = ?
```

*Questa query può essere utilizzata anche per generare le classifiche CON PUNTI di ogni singola gara, l'unica differenza è il vincolo "WHERE" finale, utilizzato per specificare il codice del pilota richiesto.*

#### OP-14: Calcolare il punteggio di un costruttore in un evento.

Supponendo di sapere già il **codice del costruttore** e i dati per individuare la **gara**, ovvero il **codice dell'autodromo**, del **campionato** e l'**anno di svolgimento**.

*Solo i primi 15 arrivati ottengono punti, secondo lo standard stabilito nell'analisi.*

```
WITH Classifica AS
(
SELECT corre.TempoGara AS 'Tempo', corre.CodicePilota AS 'CodicePilota', costruttori.Nome 'Casa', piloti.Nome AS 'Nome', piloti.Cognome AS 'Cognome', costruttori.Nome 'Costruttore',
costruttori.CodiceCostruttore AS 'CodiceCostruttore',
ROW_NUMBER() OVER ( ORDER BY corre.TempoGara ASC ) AS
'Posizione'
FROM corre INNER JOIN piloti ON piloti.CodicePilota =
corre.CodicePilota
INNER JOIN aggregazionepiloti ON piloti.CodicePilota = aggregazionepiloti.CodicePilota
INNER JOIN team ON aggregazionepiloti.CodiceTeam =
team.CodiceTeam
INNER JOIN fornitori ON fornitori.CodiceTeam = team.CodiceTeam
INNER JOIN costruttori ON fornitori.CodiceCostruttore = costruttori.CodiceCostruttore
WHERE corre.CodiceCampionato = ? AND corre.CodiceAutodromo = ?
AND corre.Anno = ?
), elencoArrivi AS
(
SELECT Classifica.Posizione AS 'Posizione', CASE
WHEN Classifica.Posizione = 1 THEN '25'
WHEN Classifica.Posizione = 2 THEN '20'
WHEN Classifica.Posizione = 3 THEN '16'
WHEN Classifica.Posizione = 4 THEN '13'
WHEN Classifica.Posizione <= 15 THEN 16-Classifica.Posizione
ELSE 0
```

```

END AS 'Punti',
Classifica.Nome AS 'Nome', Classifica.Cognome AS 'Cognome', Classi-
fica.Tempo AS 'Tempo Gara', Classifica.CodicePilota AS 'Codice Pilota',
Classifica.Casa 'Costruttore', Classifica.CodiceCostruttore AS 'Codice-
Costruttore'
FROM Classifica
)
SELECT DISTINCT elencoArrivi.Costruttore AS 'Costruttore',
(SELECT SUM(punti) FROM elencoArrivi WHERE elencoar-
rivi.CodiceCostruttore = ?) AS 'Punti'
FROM elencoArrivi
WHERE elencoArrivi.CodiceCostruttore = ?

```

*Questa query risulta essere molto simile a quella dell'OP-13, l'unica differenza è il collegamento delle varie tabelle per trovare i dati relativi del costruttore partendo dal pilota per poi sommare tutti i punti guadagnati ai relativi costruttori presenti nel campionato.*

**OP-15: Decretare il vincitore del campionato piloti a fine stagione.**

Supponendo di sapere già il **codice del campionato** preso in considerazione.

Per questa operazione **non sarà necessaria eseguire una query specifica**, siccome si dovranno eseguire **N query quante le gare all'interno del campionato** preso in analisi. Si andranno quindi a **sommare i punti guadagnati da ogni pilota** con lo scopo di trovare **colui** che ha totalizzato il **maggior numero di punti**. Questo calcolo verrà svolto dall'applicazione e non dal database.

**OP-16: Decretare il vincitore del campionato costruttori a fine stagione.**

Supponendo di sapere già il **codice del campionato** preso in considerazione.

Per questa operazione **non sarà necessaria eseguire una query specifica**, siccome si dovranno eseguire **N query quante le gare all'interno del campionato** preso in analisi. Si andranno quindi a **sommare i punti guadagnati da ogni pilota** con lo scopo di trovare **il costruttore** che ha totalizzato il **maggior numero di punti**. Questo calcolo verrà svolto dall'applicazione e non dal database.

**OP-17: Vendere un biglietto per un evento.**

Supponendo di sapere già il **codice dell'autodromo** e dello **spettatore**.

```
INSERT INTO Biglietti (Data, codiceSpettatore, codiceAutodromo)
VALUES (?, ?, ?)
```

# Progettazione dell'applicazione

## 5.1 Descrizione dell'architettura dell'applicazione realizzata

L'applicazione per interfacciarsi al database è stata realizzata in **Java**. Il database gestito risiede in **locale** e il DBMS usato è **PhpMyAdmin(mysql)**.

L'applicazione risulta avere un **pannello principale** con lo scopo di dividere a metà la **gestione degli spettatori e dei biglietti** con le operazioni più importanti quali la **gestione del campionato e delle gare effettuate**.



*Schermata iniziale dell'applicazione.*



L'applicazione quindi **due sezioni distinte**:

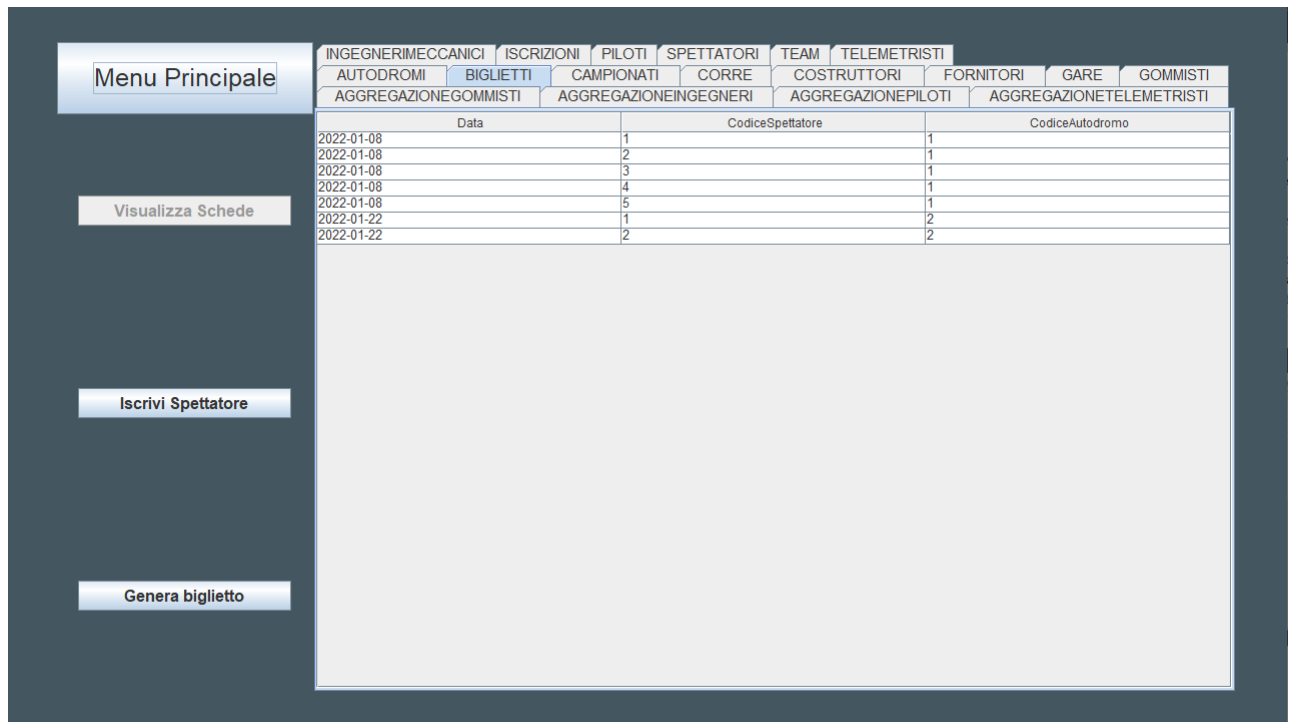
La prima sezione chiamata **"Autorizzati"** comprende la visualizzazione di tutte le tabelle del database oltre che tutte le possibili operazioni che possono riguardare la **modifica, creazione o gestione del campionato di MotoGP**.

Menu Principale	INGEGNERIMECCANICI   ISCRIZIONI   PILOTI   SPETTATORI   TEAM   TELEMETRISTI															
	AUTODROMI		BIGLIETTI		CAMPIONATI		CORRE		COSTRUTTORI		FORNITORI		GARE		GOMMISTI	
	AGGREGAZIONE GOMMISTI				AGGREGAZIONE INGEGNERI				AGGREGAZIONE PILOTI				AGGREGAZIONE TELEMETRISTI			
Visualizza Schede	CodicePilota	Nome			Cognome			CodiceFiscale			DataNascita					
Nuovo TEAM	1	Andrea	Dovizioso									1986-03-23				
Nuovo Pilota	2	Johann	Zarco									1990-07-16				
Nuovo Telemetrista	3	Stefan	Bradl									1989-11-29				
Nuovo Gommista	4	Maverick	Vinales									1995-01-12				
Nuovo Ingegnere	5	Enea	Bastianini									1997-12-30				
Nuovo Costruttore	6	Takaaki	Nakagami									1992-02-09				
Nuovo Autodromo	7	Lorenzo	Savadori									1993-04-04				
Nuova Gara	8	Brad	Binder									1995-08-11				
Assegna Componente	9	Joan	Mir									1997-09-01				
Assegna Costruttore	10	Luca	Marini									1997-08-10				
Iscrivi Team	11	Fabio	Quartararo									1999-04-20				
Nuovo Campionato	12	Franco	Morbidelli									1994-12-04				
Registra Tempo di Arrivo	13	Raul	Fernandez									2000-10-23				
Classifiche Singole gare	14	Darryn	Binder									1998-01-21				
Class. Campionato Piloti	15	Alex	Espargaro									1989-07-30				
Class. Campionato Costrutt...	16	Alex	Rins									1995-12-08				
Storico Team	17	Jack	Miller									1995-01-18				
	18	Pol	Espargaro									1991-06-10				
	19	Fabio	Di Giannantonio									1998-10-10				
	20	Michele	Pirro									1986-07-05				
	21	Francesco	Bagnaia									1997-01-14				
	22	Marco	Bezzecchi									1998-11-12				
	23	Alex	Marquez									1996-04-23				
	24	Remy	Gardner									1998-02-24				
	25	Miguel	Oliveira									1995-01-04				
	26	Jorge	Martin									1998-01-29				
	27	Marc	Marquez									1993-02-17				
	28	Cameron	Beaubier									1992-12-06				
	29	Dylan	Kelly									2002-05-17				
	30	Joe	Roberts									1997-06-16				

*Schermata "Autorizzati".*

Nella scheda relativa alla **"Registrazione Tempo Arrivo"** è stato inserito, per completezza, un form per poter registrare nuovi risultati da parte dei piloti; tuttavia questa operazione dovrebbe essere effettuata **automaticamente** tramite la connessione a un dispositivo **"transponder"**. (presente nelle moto dei piloti)

La seconda e ultima schermata è chiamata **"Spettatori"** e contiene le uniche due operazioni che possono essere effettuate nel **dominio degli spettatori**, ovvero la **registrazione di un nuovo spettatore** e la **generazione di un nuovo biglietto**.



*Schermata "Spettatori".*

*In entrambe le schermate, l'applicazione andrà a verificare il corretto inserimento di ciascun campo prima di inviare una qualsiasi query al database.*

*L'applicazione per portare a termine le query una volta ottenuti i dati dall'utente utilizza la classe "DBConnection".*

*L'applicazione fornisce nel complesso le funzionalità richieste evidenziate nella fase di progettazione*