

WikiSerieA

1. ABSTRACT

Da ormai più di un secolo il calcio è considerabile come lo sport più seguito e apprezzato in Italia. La Serie A (il campionato calcistico italiano d'eccellenza) conta oggi giorno milioni di appassionati in tutto il mondo ed è sicuramente tra i campionati sportivi più seguiti.

"WikiSerieA" si presta ad essere un'applicazione e sito Web che permettono a un qualsiasi utente, italiano o non, di iscriversi tramite una veloce registrazione, per rimanere collegato al mondo della Serie A tutti i giorni e a tutte le ore.

Un vero e proprio portale, che prevede come progetto iniziale quello di creare una base di dati contenente tutte le informazioni riguardanti il campionato di Serie A: calciatori, partite, squadre, acquisto di biglietti e con inoltre una sezione dedicata al Fantacalcio.

Quest'ultimo è un gioco in cui gruppi di amici, iscrivendosi, creano delle Leghe in cui ogni partecipante crea una sua Fantasquadra formata dai giocatori reali delle varie squadre del campionato di Serie A. L'andamento dei giocatori reali nelle partite del week-end comporta dei voti che vengono assegnati ai giocatori della propria Fantasquadra (con l'aggiunta di bonus per goal o assist, oppure malus per autogol o cartellini presi).

La somma dei punteggi ottenuti dai propri giocatori ogni fine settimana determina, con il passare dei mesi, la propria posizione nella Fantaclassifica della Lega.

WikiSerieA può essere quindi l'applicazione di riferimento per chiunque voglia vivere la Serie A a pieno, indirettamente da casa con statistiche aggiornate e il Fantacalcio, o direttamente con la possibilità di comprare i biglietti delle partite e viverle allo stadio.

2. DESCRIZIONE ED ANALISI DEI REQUISITI

Il progetto in esame vuole quindi creare una base di dati accessibile ai soli utenti iscritti a WikiSerieA.

L'iscrizione può essere effettuata da sito o da applicazione indistintamente, così da fornire a ciascun utente la piattaforma di utilizzo che più gli aggrada.

La **registrazione semplice** serve solo a fini identificativi e richiede le seguenti informazioni:

- Username, Password, e-mail, Nome, Cognome, Sesso, Data di nascita

Un utente semplice non può però acquistare i biglietti per le partite. Per questa funzione sarà necessaria una **registrazione completa**, con l'aggiunta dei seguenti campi:

- Metodo di pagamento (per l'acquisto effettivo)
 - Tipologia di pagamento, Codice della carta
- Indirizzo di residenza (per la spedizione a casa del biglietto)
 - Via, Numero Civico, Provincia, Codice Postale (CAP), Regione, Nazione

La registrazione (semplice o completa) permette di usufruire liberamente di tutte le altre parti della Wiki. Una sezione dedicata alle **partite** della Serie A:

- Squadra casa, Squadra ospite, Data e orario dell'incontro, Stadio, Goal fatti squadra casa (se l'incontro e` gia` concluso), Goal fatti squadra ospite (se l'incontro e` gia` concluso)

Una sezione dedicata alla **classifica** attuale:

- Squadra, Punti, Vittorie, Sconfitte, Pareggi, Goal fatti, Goal subiti

Una sezione dedicata ai **giocatori** di ogni squadra:

- Nome, Cognome, Nazionalità, Data di nascita, Ruolo, Squadra, Goal Fatti, Media voto, Media Fantavoto (aggiunta nel voto di possibili bonus o malus)

Una sezione dedicata alle **squadre** del campionato:

- Nome, Allenatore, Capitano, Stadio di gioco, Anno di fondazione

Per vivere le partite allo stadio c'è la sezione **biglietti**. Ciascuno di essi e` così memorizzato nella base di dati:

- Squadra casa, Squadra ospite, Data e orario dell'incontro, Stadio , Numero posto (che comprende: numero dell'anello, numero di fila, il numero del posto nella fila), Codice identificativo, Prezzo

Come già detto, sarà poi disponibile una sezione relativa al Fantacalcio, in cui, sfruttando le proprie conoscenze su giocatori e relative statistiche, ogni utente potrà creare e partecipare a una Lega privata con i propri amici.

Bisogna quindi distinguere nella base di dati ogni **Lega**, attraverso i seguenti campi:

- Codice identificativo, Nome Lega, Data di creazione, Numero di partecipanti, Username admin (l'utente capo della Lega, colui che decide le regole della Lega, come per esempio il valore da dare a ogni bonus o malus)

Ogni partecipante creerà la sua **Fantasquadra**, così memorizzata nella base di dati:

- Username (del creatore della Fantasquadra), Nome, Codice Lega (Lega unica in cui partecipa la Fantasquadra)

Ogni Fantasquadra verrà creata in seguito ad un'asta tra gli utenti di ciascuna Lega.

I giocatori da acquistare sono 25 per ciascuna Fantasquadra e il tutto avviene con l'utilizzo di Fantamilioni (una valuta virtuale) fornita a ugual numero a tutti i partecipanti della Lega.

L'**asta** di ogni Lega e` così memorizzata nella base di dati:

- Codice della Lega in cui si e` svolta l'asta, Giocatore acquistato, Nome Fantasquadra che lo ha acquistato, Prezzo d'acquisto (in Fantamilioni: la valuta del Fantacalcio)

A questo punto, creata la propria Fantasquadra attraverso l'asta, potrà iniziare il vero e proprio confronto tra le varie squadre create dagli utenti in ogni rispettiva Lega. Ogni weekend ciascun utente totalizzerà un certo punteggio (corrispondente alla somma dei Fantavoti degli 11 giocatori che ha scelto di schierare in campo tra la sua rosa di 25 giocatori scelti all'asta) che, partita dopo partita, si sommerà con i punteggi delle successive giornate. Ci sarà quindi una sezione dedicata alla **Fantaclassifica**, memorizzata nella base di dati con i seguenti campi:

- Codice della Lega, Nome Fantasquadra, Somma punteggi

Glossario dei termini

TERMINE	DESCRIZIONE	SINONIMI	COLLEGAMENTI
Utente	Colui che si iscrive al portale di WikiSerieA attraverso registrazione semplice, inserendo i suoi dati personali a fini identificatori.	Iscritto, Utente semplice	Lega, Fantasquadra
UtenteCliente	Specializzazione dell'utente. E' colui che ha intenzione di acquistare biglietti per le partite e inserisce ulteriori dati personali per questo fine.	Utente completo	Biglietto
Biglietto	Metodo di accesso a ciascuna partita.	Ticket	UtenteCliente, Partita
Partita	Confronto in campo tra due squadre (una di casa e una ospite).Il risultato equivale al numero di goal fatti da entrambe le squadre.	Match, Incontro	Biglietto, Squadra
Squadra	Gruppo di giocatori uniti in un unico team.	Team	Classifica, Giocatore
Classifica	Graduatoria tra le squadre della Serie A, ordinata secondo i punti totalizzati dalle vittorie (3 punti) e pareggi (1 punto) ottenuti.	Graduatoria	Squadra
Giocatore	Colui che gioca in una delle squadre di Serie A. Puo` essere portiere, difensore, centrocampista, attaccante a seconda della sua posizione in campo (puo` essere anche capitano della squadra in cui gioca).	Calciatore	Squadra, Lega, Fantasquadra
Fantasquadra	Squadra virtuale creata da un utente e formata da 25 calciatori reali giocanti nella Serie A.	Squadra virtuale	Utente, Lega, Giocatore
Lega	Campionato virtuale di Fantasquadre create da utenti che si sfidano a chi totalizza piu` punti con i voti dei propri giocatori giornata dopo giornata.	Campionato virtuale	Utente, Giocatore, Fantasquadra
Asta	Metodo attraverso il quale gli utenti di una Lega si contendono i calciatori giocanti nella Serie A per creare la propria Fantasquadra.	Trattativa	Fantasquadra, Lega, Giocatore
Fantaclassifica	Graduatoria tra gli utenti di una Lega, ordinata secondo la somma dei Fantavoti dei giocatori di ciascuna Fantasquadra, partita dopo partita.	Graduatoria virtuale	Lega, Fantasquadra

Strutturazione dei requisiti

Frasi di carattere generale
Si vuole realizzare una base di dati che raccoglie informazioni sul campionato calcistico italiano di Serie A, che fornisce la possibilità di acquistare biglietti per le partite e di giocare al Fantacalcio con altri utenti.

Frase relative agli utenti

Gli utenti iscritti a WikiSerieA sono identificati attraverso una registrazione semplice che richiede il nome, il cognome, la data di nascita, il sesso, un username, una email e una password. Per la registrazione completa (richiesta per l'acquisto di biglietti) bisogna inserire anche un metodo di pagamento e il proprio indirizzo di residenza.

Frase relative alle partite di Serie A

Le partite di Serie A prevedono il nome della squadra di casa, il nome della squadra ospite, la data e orario dell'incontro, lo stadio di gioco e il risultato (nel caso la partita sia già terminata).

Frase relative alla classifica di Serie A

Nella classifica della Serie A è distinguibile per ogni squadra partecipante il numero di punti che ha totalizzato, il numero di vittorie, sconfitte e pareggi, i goal fatti ed i goal subiti.

Frase relative ai giocatori della Serie A

Ogni giocatore è distinguibile dagli altri nella base di dati attraverso un Codice identificativo (per evitare problemi con giocatori possibilmente omonimi), il suo nome, cognome, squadra in cui gioca, ruolo in campo, goal da lui realizzati, la sua età, la sua nazionalità, il suo voto medio e fantavoto medio

Frase relative alle squadre della Serie A

Le squadre sono identificate con il loro nome, il loro allenatore, il loro capitano, il loro stadio di gioco e l'anno in cui sono state fondate.

Frase relative ai biglietti delle partite

Ciascun biglietto è unico attraverso il codice identificativo e distinguibile attraverso il nome della squadra di casa, il nome della squadra ospite, la data e orario dell'incontro, lo stadio in cui si svolgerà l'incontro, il numero del posto a sedere nello stadio e il relativo prezzo.

Frase relative alla Lega di Fantacalcio

Ogni Lega di Fantacalcio si distingue per il suo nome, la sua data di creazione, il suo unico codice identificativo, il numero di utenti nella Lega e lo username dell'admin della Lega.

Frase relative alla Fantasquadra di Fantacalcio

Ogni Fantasquadra si distingue per lo username dell'utente che l'ha creata, il nome della Fantasquadra stessa e il nome della Lega in cui partecipa (ogni Fantasquadra può partecipare a una sola Lega).

Frase relative all'asta di ciascuna Lega

L'asta è unica tra gli utenti di una Lega e viene memorizzata attraverso un campo giocatore, un campo che individua la Fantasquadra che lo ha acquistato e un campo che identifica il prezzo d'acquisto.

Frasi relative alla Fantaclassifica di ciascuna Lega

La Fantaclassifica è unica in ogni Lega ed è formata da un campo che identifica il nome della Fantasquadra, la relativa Lega in cui partecipa e il suo punteggio fino a quel momento.

3. PROGETTAZIONE CONCETTUALE

Lista delle Entità

Le entità del database sono qui sotto riportate. Tutti gli attributi sono **not null**, tranne quelli specificati:

- **Utente:** rappresenta un utente generico della Wiki
 - Username: varchar(20) unique,
 - Password: varchar(16),
 - e-mail: varchar(50),
 - Nome: varchar(20),
 - Cognome: varchar(20),
 - Sesso: (M o F),
 - Data di nascita: date

L'utente si specializza in due sottocategorie con una generalizzazione totale:

- **Non acquista:** rappresenta un utente semplice, che non effettua acquisti
- **UtenteCliente:** rappresenta un utente che vuole anche acquistare dei biglietti
 - Metodo di pagamento: (Carta di credito o Carta di debito),
 - Codice carta: int(16),
 - Via: varchar(20),
 - NumCivico: varchar(5),
 - Provincia: varchar(20),
 - CAP: int(5),
 - Regione: varchar(20),
 - Nazione: varchar(20)
- **Giocatore:**
 - Codice: int unique auto_increment,
(Si utilizza un codice per identificare univocamente un giocatore. Usare solo nome e cognome potrebbe creare problemi se due giocatori sono omonimi)
 - Nome: varchar(20),
 - Cognome: varchar(20),
 - Nazionalità: varchar(20),
 - DataNascita: date,
 - Squadra: varchar(20),
 - Goal fatti: int(2),
 - MediaVoto: int(2),
 - MediaFantavoto: int(2)

Attraverso una generalizzazione totale abbiamo distinto i giocatori per il loro ruolo in campo:

- **Portiere**
- **Difensore**
- **Centrocampista**
- **Attaccante**

- **Squadra:**
 - Nome: varchar(20) unique,
 - Allenatore: varchar(20),
 - Capitano: int,
 - AnnoFondazione: int(4),
 - Stadio: varchar(20)
- **Classifica:**
 - Squadra: varchar(20) unique,
 - Punti: int(3),
 - Vittorie: int(2),
 - Sconfitte: int(2),
 - Pareggi: int(2),
 - GoalFatti: int(3),
 - GoalSubiti: int(3)
- **Biglietto:**
 - Codice: int unique,
 - SquadraCasa: varchar(20),
 - SquadraOspite: varchar(20),
 - DataOraincontro: timestamp,
 - NumeroAnello: int(1),
 - NumeroFila: int(3),
 - NumeroPosto: int(4),
 - Prezzo: decimal(3,2)
- **Partita:**
 - SquadraCasa: varchar(20),
 - SquadraOspite: varchar(20),
 - DataOraincontro: timestamp,
 - Stadio: varchar (20)

Le partite si dividono in:

- **Partita futura** : partita che deve ancora iniziare
- **Partita passata** : partita già conclusa, di cui si conosce quindi il risultato finale
 - Goal squadra di casa: int(2),
 - Goal squadra ospite: int(2)
- **F_Lega:**
 - Codice: int unique,
 - Nome: varchar(20),
 - DataCreazione: date,
 - NumeroPartecipanti: int(3),
 - Admin: varchar(20),
- **F_Fantasquadra:**
 - Nome: varchar (20) unique,
 - Codice Lega: int unique,
 - UsernameCreatore: varchar(20)

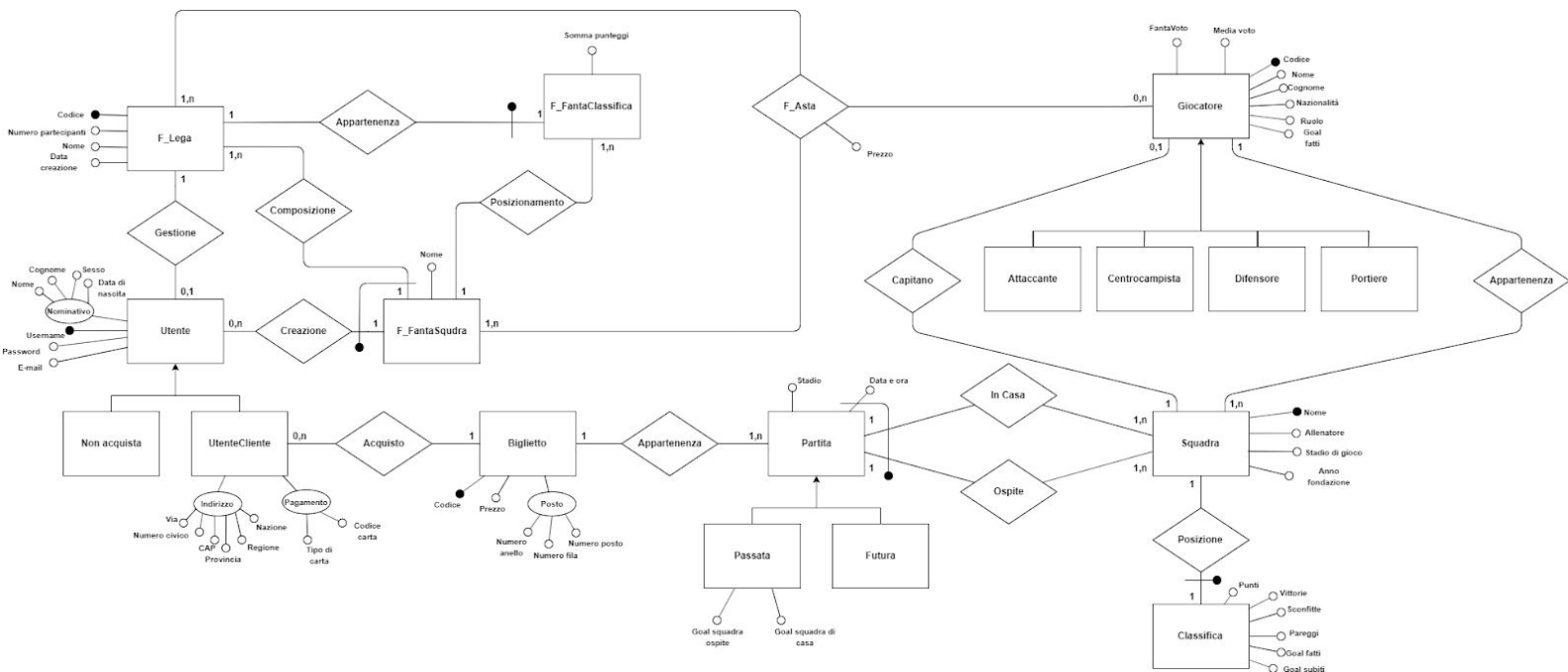
- **F_Fantaclassifica:**
 - CodiceLega: int unique,
 - Nome Fantasquadra: varchar(20) unique,
 - Punteggio: integer(4)

Lista delle Relazioni

Le entità sopra descritte sono collegate dalle seguenti relazioni:

- **UtenteCliente - Biglietto: Acquisto**
 - ❑ Un cliente può acquistare zero o più biglietti (0,N)
 - ❑ Un singolo biglietto viene acquistato da uno e un solo utente (1,1)
- **Biglietto - Partita: Appartenenza**
 - ❑ Un biglietto appartiene a una e una sola partita (1,1)
 - ❑ Per una singola partita sono disponibili più biglietti (1,N)
- **Partita - Squadra: In casa**
 - ❑ Una partita ha una e una sola squadra di casa (1,1)
 - ❑ Una squadra può giocare in casa in più partite del campionato (1,N)
- **Partita - Squadra: Ospite**
 - ❑ Una partita ha una e una sola squadra ospite (1,1)
 - ❑ Una squadra può essere ospite in più partite del campionato (1,N)
- **Squadra - Classifica: Posizione**
 - ❑ Una squadra è posizionata in una e una sola classifica (1,1)
 - ❑ Una classifica mostra la posizione di tutte le squadre del campionato (1,N)
- **Squadra - Giocatore: Appartenenza**
 - ❑ In una squadra appartengono più giocatori (1,N)
 - ❑ Un giocatore appartiene a una e una sola squadra (1,1)
- **Squadra - Giocatore: Capitano**
 - ❑ Ogni squadra ha uno e un solo capitano (1,1)
 - ❑ Un giocatore può essere capitano di una squadra, o non esserlo (0,1)
- **F_Fantasquadra - Utente: Creazione**
 - ❑ Una Fantasquadra viene creata da uno e un solo utente (1,1)
 - ❑ Un utente può creare nessuna o più Fantasquadre (0,N)
- **Utente - F_Lega: Gestione**
 - ❑ Un utente può gestire una sola Lega, o nessuna (0,1)
 - ❑ Una Lega ha uno e un solo utente admin (1,1)
- **F_Lega - F_Fantasquadra: Partecipazione**
 - ❑ In una Lega partecipano più Fantasquadre (1,N)
 - ❑ Una Fantasquadra può partecipare a una sola Lega (1,1)
- **F_Lega - F_Fantaclassifica: Appartenenza**
 - ❑ In una Lega è presente una e una sola Fantaclassifica (1,1)
 - ❑ Una Fantaclassifica appartiene a una e una sola Lega (1,1)
- **F_Fantaclassifica - F_Fantasquadra: Posizionamento**
 - ❑ In una Fantaclassifica partecipano tutte le Fantasquadre della relativa Lega (1,N)
 - ❑ Una Fantasquadra partecipa a una e una sola Fantaclassifica (1,1)
- **F_Lega - F_Fantasquadra - Giocatore: F_Asta**
 - ❑ Ad una Lega appartengono più Fantasquadre e più Giocatori (1,N)
 - ❑ Una Fantasquadra appartiene a solo una Lega, ci appartengono più giocatori (1,N)
 - ❑ Un giocatore è in più Leghe e in più Fantasquadre, ma solo di Leghe diverse (0,N)

Schema concettuale (E-R)



4. PROGETTAZIONE LOGICA

RISTRUTTURAZIONE DELLO SCHEMA

Analisi ridondanza

Una ridondanza si può trovare nell'entità 'Classifica', in cui si trova il campo 'GoalFatti' il quale potrebbe essere rappresentato dalla somma dei GoalFatti di ciascun giocatore appartenente a una specifica squadra. Studiamo la ridondanza su due operazioni:

- **Operazione 1:** stampa dei goal di una squadra composta da 20 giocatori (100 volte al giorno)
- **Operazione 2:** aggiornamento dei goal di due giocatori di una squadra (10 volte al giorno)

Concetto	Tipo	Volume
Giocatore	Entità	1000
Squadra	Entità	50
Classifica	Entità	50

Con ridondanza

Operazione 1

Concetto	Costrutto	Accessi	Tipo
Classifica	Entità	1	L

Costo: 100 accessi in lettura

Costo: $2 \cdot 10 + 10 + 10$ accessi in lettura e $2 \cdot 10 + 10$ accessi in scrittura → 40 letture e 30 scritture

Operazione 2

Concetto	Costrutto	Accessi	Tipo
Giocatore	Entità	2	L
Giocatore	Entità	2	S
Squadra	Entità	1	L
Classifica	Entità	1	L
Classifica	Entità	1	S

Senza ridondanza

In questo caso l'attributo 'GoalFatti' dell'entità 'Squadra' è raggiungibile sommando tutti i goal fatti di un giocatore della stessa squadra, utilizzando quindi gli attributi 'GoalFatti' e 'Squadra' dell'entità 'Giocatore'

Operazione 1

Concetto	Costrutto	Accessi	Tipo
Giocatore	Entità	20	L

Costo: 100*20 accessi in lettura
→ 2000 accessi in lettura

Operazione 2

Concetto	Costrutto	Accessi	Tipo
Giocatore	Entità	2	L
Giocatore	Entità	2	S

Costo: 10*2 accessi in lettura e 10*2 accessi in scrittura → 20 letture e 20 scritture

Guardando i risultati degli accessi relativi alle operazioni giornaliere abbiamo deciso di mantenere l'attributo 'GoalFatti' in 'Classifica' e quindi mantenere la ridondanza.

Eliminazione delle generalizzazioni

Utente

Questa generalizzazione totale prevede due entità figlie che identificano gli utenti che non acquistano biglietti a quelli che li acquistano. Per la ristrutturazione della generalizzazione verso lo schema logico abbiamo ragionato su due aspetti:

- *Attributi*: gli attributi dell'entità padre appartengono anche alle entità figlie, con l'aggiunta di ulteriori attributi per gli utenti clienti, che intendono acquistare biglietti.
- *Operazioni*: le operazioni che possono compiere gli utenti che non acquistano e gli utenti che acquistano sono le medesime, tranne per la possibilità di acquistare biglietti (operazione unicamente disponibile all'utente cliente).

Per questi motivi abbiamo deciso di accorpare l'entità figlia 'Non Acquista' con quella genitore 'Utente' e di mantenere separata l'entità 'Cliente', collegandola con una relazione 'Specializzazione' ad 'Utente'.

- *Utente - Cliente: Specializzazione*
 - ☐ Un utente può essere o non essere specializzato nell'acquisto biglietti (0,1)
 - ☐ Un cliente è una specializzazione di uno e un solo utente (1,1)

Partita

Questa generalizzazione totale prevede due entità figlie che distinguono le partite in passate e future. Per la ristrutturazione della generalizzazione verso lo schema logico abbiamo ragionato su due aspetti:

- *Attributi*: gli attributi dell'entità padre appartengono anche alle entità figlie. L'entità figlia che rappresenta le partite passate possiede anche gli attributi riguardanti il risultato finale.
- *Operazioni*: non ci sono operazioni che distinguono le due entità figlie.

Per questi motivi abbiamo deciso di accorpare entrambe le entità figlie nell'entità genitore 'Partita', aggiungendole gli attributi 'GoalSquadraCasa' e 'GoalSquadraOspite' dell'entità 'PartitaPassata' e un attributo booleano 'Passata' per distinguere le due tipologie di partite.

Giocatore

Questa generalizzazione totale prevede quattro entità figlie che identificano i giocatori per il loro ruolo (portiere, difensore, centrocampista, attaccante). Per la ristrutturazione della generalizzazione verso lo schema logico abbiamo ragionato su due aspetti:

- *Attributi*: gli attributi dell'entità genitore appartengono a ciascuna entità figlia. Inoltre le entità figlie non hanno attributi propri che le distinguono l'una dall'altra.
- *Operazioni*: le entità figlie non sono legate a nessuna associazione. Ciascuna operazione compiuta nell'entità genitore appartiene a ogni entità figlia.

Per questi motivi abbiamo deciso di accorpare le entità figlie nell'entità genitore 'Giocatore', aggiungendo a quest'ultima un attributo 'Ruolo' che distingua i vari giocatori per la loro posizione in campo.

Accorpamento di entità e relazioni

Classifica

Abbiamo deciso di accorpare la relazione 'Posizione' che univa le entità Classifica e Squadra in un'unica entità Squadra, eliminando l'entità Classifica e aggiungendo tutti gli attributi di quest'ultima a Squadra. Così facendo lo schema ristrutturato risulta essere più comprensibile visto che ogni attributo di Classifica riguardava singole squadre, distinguibili ora anche per i punti totalizzati in campionato fino a quel momento, goal fatti e goal subiti.

F_Fantaclassifica

Per gli stessi motivi abbiamo deciso di accorpare la relazione 'Posizionamento' che univa le entità F_Fantaclassifica e Fantasquadra in un'unica entità Fantasquadra, eliminando l'entità F_Fantaclassifica e aggiungendo tutti gli attributi di quest'ultima a Fantasquadra.

Ogni Fantasquadra risulta essere ora distinguibile anche per il suo punteggio nella classifica della Lega di appartenenza direttamente dalla entità Fantasquadra.

Relazione F_Asta

Abbiamo poi trasformato la relazione ternaria F_Asta in una entità, fornendogli l'attributo prezzo della relazione e avente come chiave primaria il codice del giocatore e il codice della lega, visto che in ogni Lega non può essere stato acquistato lo stesso giocatore più volte.

Scelta degli identificatori principali

Abbiamo deciso di aggiungere un attributo alle entità **F_FantaSquadra** e **Partita**. Questo lo abbiamo chiamato **Codice**, ossia un valore numerico che identifica ciascuna Fantasquadra e ciascuna partita (Codice è rispettivamente chiave primaria delle due entità).

Nel caso della Fantasquadre è stato necessario per l'identificazione di esse nella tabella F_Asta, per evitare di dover utilizzare tutti i valori di Nome Fantasquadra e il codice della Lega di partecipazione (ne consegue risparmio di memoria con l'aumentare del volume delle tabelle). Per le partite è stato necessario per evitare di trasportare gli attributi SquadraCasa, SquadraOspite e DataOraincontro nella tabella dei Biglietti per identificare la partita in ciascun biglietto (sempre per un generale risparmio di memoria).

Schema logico

Utente(Username, Password, e-mail, Nome, Cognome, Sesso, DataNascita)

Cliente(Username, MetodoPagamento, CodiceCarta, Via, NumCivico, Provincia, CAP, Regione, Nazione)

Giocatore(Codice, Nome, Cognome, Nazionalità, DataNascita, Ruolo, Squadra, GoalFatti, MediaVoto, MediaFantavoto)

Giocatore.Squadra -> Squadra.Nome

Squadra(Nome, Allenatore, Capitano, AnnoFondazione, Stadio, Punti, Vittorie, Sconfitte, Pareggi, GoalFatti, GoalSubiti)

Squadra.Capitano -> Giocatore.Codice

Partita(Codice, SquadraCasa, SquadraOspite, DataOraincontro, Stadio, Passata, GoalSquadraCasa, GoalSquadraOspite)

Partita.SquadraCasa -> Squadra.Nome

Partita.SquadraOspite -> Squadra.Nome

Biglietto(Codice, CodicePartita, NumeroAnello, NumeroFila, NumeroPosto, Prezzo, Cliente)

Biglietto.Cliente -> Cliente.Username

Biglietto.CodicePartita -> Partita.Codice

F_Lega(Codice, Nome, DataCreazione, NumeroPartecipanti, Admin)

F_Lega.Admin -> Utente.Username

F_Fantasquadra(Codice, Nome, UsernameCreatore, CodiceLega, Punteggio)

F_Fantasquadra.CodiceLega -> F_Lega.Codice

F_Fantasquadra.UsernameCreatore -> Utente.Username

F_Asta(CodiceLega, CodiceGiocatore, Fantasquadra, Prezzo)

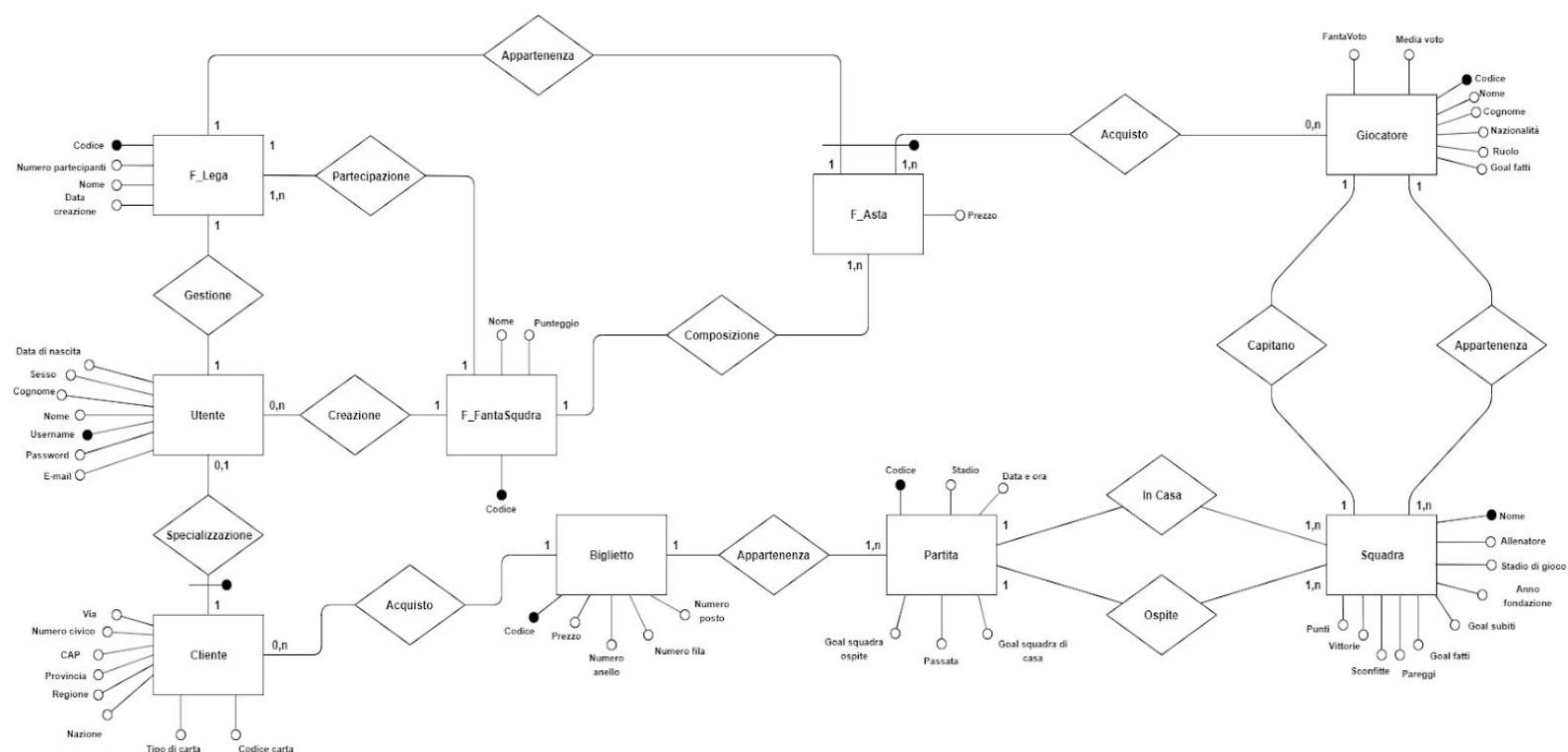
F_Asta.CodiceLega -> F_Lega.Codice

F_Asta.CodiceGiocatore -> Giocatore.Codice

F_Asta.Fantasquadra -> F_Fantasquadra.Codice

Per garantire la correttezza dei dati, bisognerebbe aggiungere alcuni trigger, al posto di questi abbiamo aggiunto degli update dopo aver inserito i dati corretti.

Schema concettuale ristrutturato



5. QUERY E INDICI

Query

1) Visualizzare l'indirizzo di tutti gli utenti che hanno acquistato un biglietto di una partita dove giocava la Juventus.

```
SELECT UtenteCliente.Username, Nazione, Provincia, CAP, Via, NumeroCivico
FROM Biglietto, UtenteCliente
WHERE Biglietto.Cliente=UtenteCliente.Username AND Biglietto.CodPartita IN (
    SELECT Partita.Codice
    FROM Partita
    WHERE SquadraCasa="Juventus" OR SquadraOspite="Juventus"
);
```

Username	Nazione	Provincia	CAP	Via	NumeroCivico
Ale.Pugliesi	Italia	TN	38017	De Varda	45
Carla.Costa	Italia	FM	63900	Maranesi	6
Lui.Ferri	Italia	BO	40037	Ponte Albano	8

2) Selezionare nome, cognome e goal fatti dei giocatori che hanno realizzato almeno 5 goal, sono stati gli unici ad averne fatti quel numero e sono stati acquistati in almeno una F_Asta.

```
SELECT G.Nome, G.Cognome, G.GoalFatti
FROM Giocatore G
WHERE NOT EXISTS (SELECT *
    FROM Giocatore G1
    WHERE G.GoalFatti = G1.GoalFatti AND G.Codice <> G1.Codice
    GROUP BY G1.GoalFatti)
AND G.GoalFatti > 4 AND (G.Codice) IN (SELECT Giocatore
    FROM F_Asta)
ORDER BY G.GoalFatti;
```

Nome	Cognome	GoalFatti
Romelu	Lukaku	7

3) L'utente che ha acquistato nella propria fantasquadra il giocatore che ha segnato di più in campionato (tra quelli acquistati).

```
SELECT Utente
FROM F_Fantasquadra JOIN F_Asta ON F_Fantasquadra.Codice=F_Asta.CodFantasquadra
WHERE F_Asta.Giocatore = (
    SELECT Codice
    FROM Giocatore
    WHERE GoalFatti = (
        SELECT max(GoalFatti)
        FROM Giocatore
        WHERE Codice IN (
            SELECT Giocatore
            FROM F_Asta
        )
    )
)
AND Codice IN (
    SELECT Giocatore
    FROM F_Asta
)
);
```

Utente
Silvio.Botros

4) Trovare le squadre vincitrici delle partite relative ai biglietti acquistati dal proprietario della fantasquadra in prima posizione nella lega Fantamici.

```
SELECT IF(GoalSquadraCasa>GoalSquadraOspite, SquadraCasa,
IF(GoalSquadraOspite>GoalSquadraCasa, SquadraOspite, "Pareggio")) AS
SquadraVincitrice FROM Partita
WHERE Codice IN (
    SELECT CodPartita
    FROM Biglietto
    WHERE Cliente = (
        SELECT Utente
        FROM F_Fantasquadra
        WHERE Punteggio = (
            SELECT max(Punteggio)
            FROM F_Fantasquadra
            WHERE CodLega IN (
                SELECT Codice
                FROM F_Lega
                WHERE Nome="Fantamici"
            )
        )
    )
);
```

SquadraVincitrice
Inter
Pareggio

5) Vedere il risultato delle partite in cui hanno vinto i giocatori acquistati da Ale.Pugliesi.

```
SELECT SquadraCasa, GoalSquadraCasa, GoalSquadraOspite, SquadraOspite
FROM Partita
WHERE (SquadraCasa IN (
    SELECT Giocatore.Squadra
    FROM (Giocatore JOIN F_Asta ON Giocatore.Codice = F_Asta.Giocatore) JOIN
F_Fantasquadra ON F_Asta.CodFantasquadra = F_Fantasquadra.Codice
    WHERE F_Fantasquadra.Utente="Ale.Pugliesi") AND
GoalSquadraCasa>GoalSquadraOspite) OR
(SquadraOspite IN (
    SELECT Giocatore.Squadra
    FROM (Giocatore JOIN F_Asta ON Giocatore.Codice = F_Asta.Giocatore)
JOIN F_Fantasquadra ON F_Asta.CodFantasquadra = F_Fantasquadra.Codice
    WHERE F_Fantasquadra.Utente="Ale.Pugliesi") AND
GoalSquadraCasa<GoalSquadraOspite);
```

SquadraCasa	GoalSquadraCasa	GoalSquadraOspite	SquadraOspite
Juventus	3	0	Lazio

6) Vedere la classifica con le relative posizioni delle squadre appartenenti alla stessa lega degli utenti che hanno acquistato più di biglietti le cui partite sono terminate pari

```
DROP VIEW IF EXISTS
Users;
CREATE VIEW Users AS
SELECT Cliente
FROM (SELECT Biglietto.Codice, Cliente, Biglietto.CodPartita
```

```

FROM Biglietto JOIN Partita ON Biglietto.CodPartita=Partita.Codice
WHERE Partita.GoalSquadraCasa=Partita.GoalSquadraOspite)
BigliettiPartitePareggiate
GROUP BY Cliente HAVING (count(*)>1);

SELECT Utente, Punteggio, @riga := IF(@precedente = CodLega, @riga + 1, 1) AS
Posizione, @precedente := CodLega AS CodLega
FROM F_Fantasquadra, (SELECT @riga := 0) x, (SELECT @precedente := '') y
WHERE CodLega IN (SELECT CodLega
FROM F_Fantasquadra
WHERE Utente IN (SELECT * FROM Users)
)
ORDER BY CodLega, Punteggio DESC;

```

Utente	Punteggio ▾ 2	Posizione	CodLega ▲ 1
Luca.Esposito	1054	1	8
Giu.Repezza	961	2	8
Giuse.Bianchi	989	1	10
Silvio.Botros	915	2	10
Jose.Ghidara	870	3	10

7) Selezionare le squadre di SerieA la cui media dei propri giocatori nell'attributo 'MediaFantaVoto' e' >= 7. Per ciascuna di esse selezionare inoltre nome, cognome, e goal fatti del miglior realizzatore.

```

DROP VIEW IF EXISTS
MediaSquadre;
CREATE VIEW MediaSquadre AS
SELECT Squadra, AVG(MediaFantaVoto) AS MediaGiocatori
FROM Giocatore
GROUP BY Squadra;

DROP VIEW IF EXISTS
TopScorer;
CREATE VIEW TopScorer AS
SELECT Squadra, MAX(GoalFatti) AS MigliorMarcatore
FROM Giocatore
GROUP BY Squadra;

SELECT T.Squadra, MediaGiocatori, Nome, Cognome, MigliorMarcatore
FROM Giocatore G JOIN TopScorer T ON (G.GoalFatti = T.MigliorMarcatore)
AND (G.Squadra = T.Squadra) JOIN MediaSquadre M ON T.Squadra = M.Squadra
WHERE MediaGiocatori >= 7
ORDER BY MediaGiocatori DESC;

```

Squadra	MediaGiocatori ▾ 1	Nome	Cognome	MigliorMarcatore
Lazio	7.7500	Ciro	Immobile	9
Atalanta	7.2500	Luis	Muriel	8
Juventus	7.0000	Cristiano	Ronaldo	5

Indici

L'utilizzo reale della base di dati comporta inequivocabilmente l'aumento dei volumi di ciascuna delle tabelle. La conseguenza principale di ciò è l'aumento dei tempi di accesso alle informazioni contenute in esse. Questo aspetto può creare delle criticità nella base di dati, in particolare nelle tabelle in cui il numero di tuple aumenta maggiormente.

Nel nostro caso le tabelle che più si espandono dal punto di vista di numero di tuple, possono essere **Utente** (ogni tupla rappresenta un singolo utente), **Biglietto** (considerando che ogni UtenteCliente potrebbe comprare anche più biglietti ogni settimana) e **F_Asta** (considerando che ogni Fantasquadra deve comprare 25 giocatori nell'asta della sua Lega).

Indice 'utenti'

Le tuple della tabella Utente difficilmente necessiteranno di modifiche dei loro attributi o la loro cancellazione totale. Per questo risulta ancora più utile avere un indice che permetta di evitare, una volta che una query richieda informazioni sugli utenti, di scansionare l'intera tabella e arrivare ai dati necessari consultando semplicemente l'indice.

In SQL quest'ultimo può essere creato attraverso questa istruzione:

```
CREATE INDEX utenti ON Utente (Username);
```

Indice 'biglietti'

Lo stesso può valere per la tabella Biglietto che non richiederà modifiche con il passare del tempo. Una volta che viene acquistato un biglietto da un Cliente infatti, questo viene memorizzato per soli scopi di coerenza, ma non si avrà probabilmente mai l'interesse di modificare o cancellare l'acquisto. La lettura delle tuple invece può essere necessaria per monitorare gli acquisti di alcuni clienti. L'indice può quindi essere così creato:

```
CREATE INDEX biglietti ON Biglietto (Codice);
```

Indice 'aste'

Considerando che ogni Fantasquadra crea la sua squadra di giocatori a inizio campionato ed essa rimane tale per tutto l'anno, anche questa tabella risulta facilmente indicizzabile attraverso l'indice 'aste', così creato:

```
CREATE INDEX aste ON F_Asta (CodiceLega, CodiceGiocatore);
```