

Studente: **Roberto Casadei**  
Matricola: **0000440575**  
Codice gruppo: **288**

# SportCore

Elaborato per il corso di **Sistemi Informativi**

Corso di Laurea in Ingegneria Elettronica, Informatica e  
Telecomunicazioni

Anno Scolastico 2010-2011

## Sommario

Introduzione.....	3
1. Analisi dei requisiti.....	4
1.1. Intervista: formulazione dei requisiti.....	4
1.2. Concetti principali, ambiguità, riformulazione dei requisiti.....	4
1.3. Tipologie di utenti ed estensione dei requisiti.....	5
1.4. Completamento dei requisiti: requisiti informativi, di processo, sui vincoli di integrità.....	7
2. Progettazione concettuale.....	9
2.1. Strategia di progetto.....	9
2.2. Glossario dei termini e schema scheletro.....	9
2.3. Correzioni e sviluppo dello schema E/R.....	10
2.4. Schema concettuale finale.....	11
2.5. Documentazione dello schema E/R.....	13
2.5.1. Dizionario dei dati.....	13
2.5.2. Regole di vincolo (e di derivazione).....	14
3. Progettazione logica.....	16
3.1. Stima del volume dei dati e del carico di lavoro.....	16
3.2. Descrizione delle operazioni principali.....	17
3.3. Raffinamento dello schema.....	17
3.4. Traduzione di entità e associazioni in relazioni.....	21
3.6. Schemi di navigazione.....	22
3.7. Traduzione delle operazioni in query SQL.....	24
4. Progettazione dell'interfaccia.....	26
4.1. Modifiche allo schema relazionale.....	26
4.2. Strategia di realizzazione.....	26

# Introduzione

*Un centro sportivo in forte espansione chiamato SportCore, dotato di diverse strutture e campi da gioco, desidera adottare un sistema che gli consenta di gestire in modo efficace le prenotazioni dei campi da gioco e di istituire tornei con premi.*

*In particolare, ciò che si vuole è un sistema informatizzato che semplifichi la gestione dei dati e dei processi, nonché dare la possibilità a persone ed utenti di accedere istantaneamente alle informazioni di interesse, ad esempio per visualizzare le tariffe dei campi o iscriversi ad un torneo.*

Questa prima descrizione del problema suggerisce la necessità di:

1. mantenere una **base di dati** per memorizzare tutte le informazioni finalizzate alla **gestione del centro sportivo** e al **miglioramento del servizio**
2. realizzare un'**interfaccia** verso l'utente che offra **informazioni aggiornate** e che consenta di interagire con il centro sportivo e **usufruire del servizio** in maniera semplice e diretta

Si vuole cioè effettuare il passaggio da un sistema di gestione disorganizzato, cartaceo e manuale a un sistema organizzato, informatizzato e automatizzato.

Inoltre, trattandosi di un *centro sportivo in forte espansione*, particolare attenzione è posta sulla **flessibilità** della soluzione, in modo da non precludere od ostacolare il soddisfacimento di requisiti futuri che potrebbero sorgere.

**Gli obiettivi strategici sono fissati.** Ora occorre esaminare in maniera più dettagliata il problema.

# 1. Analisi dei requisiti

## 1.1. Intervista: formulazione dei requisiti

Il primo passo dell'analisi è quello della **raccolta dei requisiti**. Questa fase può avvenire attraverso un'intervista al committente.

Dopo essersi incontrati con il committente, si è arrivati alla seguente descrizione in linguaggio naturale della realtà di interesse:

*SportCore è un centro sportivo che offre strutture e servizi per gli sportivi.*

*Dispone attualmente di diversi **campi da gioco**: quattro campi in sabbia (per beach tennis, beach volley, foot volley.), un campo da calcio a 5 e due campi da calcio a 7. E' in programma la costruzione di ulteriori campi ed impianti.*

*E' data la possibilità di **prenotare** campi da gioco secondo tariffe ben precise di tot/ora. Le prenotazioni possono essere effettuate anche via web.*

*Di tanto in tanto vengono istituiti **tornei**, che possono svolgersi su più campi di gioco, a cui le **squadre** possono iscriversi (anche via web) sostenendo una certa tariffa. Al termine di un torneo, i vincitori avranno diritto a un premio in denaro. Si vuole inoltre assegnare un arbitro ad ogni **partita**, il quale riporterà un referto con il risultato del match ed eventuali note (ammonizioni, penalità ecc..).*

*E' importante garantire che non vi sia sovrapposizione tra i tornei e le prenotazioni dei campi da gioco.*

## 1.2. Concetti principali, ambiguità, riformulazione dei requisiti

Si intende ora sottolineare i **concetti principali** che scaturiscono dalla precedente formulazione dei requisiti ed effettuare un **filtraggio** degli stessi per andare alla ricerca di **ambiguità** ed imprecisioni al fine di rendere i requisiti più chiari ed immediatamente fruibili dal progettista.

I concetti più importanti che subito saltano all'occhio sono i seguenti:

- Campo da gioco
- Prenotazione
- Torneo
- Squadra
- Partita

Partendo da questa base di conoscenza, rileggiamo i requisiti da un punto di vista di correttezza espressiva:

*SportCore è un centro sportivo che offre strutture e servizi per gli sportivi.*

*Dispone attualmente di diversi campi da gioco: quattro campi in sabbia (per beach tennis, beach volley, foot volley.), un campo da calcio a 5 e due campi da calcio a 7. E' in programma la costruzione di ulteriori campi ed impianti.*

*E' data la possibilità di prenotare campi da gioco secondo tariffe ben precise di tot/ora. Le*

*prenotazioni possono essere effettuate anche via web.*

*Di tanto in tanto vengono istituiti tornei, che possono **svolgersi su più campi di gioco (1)**, a cui le squadre possono iscriversi (anche via web) sostenendo una certa **tariffa (2)**. Al termine di un torneo, i **vincitori (3)** avranno diritto a un premio in denaro. Si vuole inoltre assegnare un arbitro ad ogni **partita (4)**, il quale riporterà un referto con il risultato del **match (5)** ed eventuali note (ammonizioni, penalità ecc..).*

*E' importante garantire che non vi sia sovrapposizione tra i **tornei (6)** e le prenotazioni dei campi di gioco.*

Alcune espressioni non sono esattamente corrette, perciò sarebbe opportuno effettuare le seguenti **sostituzioni**:

1. Il fatto di “svolgersi su più campi di gioco” non è esattamente una proprietà dei tornei, quanto piuttosto delle *partite* che appartengono alla programmazione dei tornei
2. Il termine “tariffa” si riferisce più propriamente alla tassa oraria per la prenotazione di un campo da gioco. Relativamente ai tornei, è meglio identificare questo concetto con il nome “tassa d'iscrizione”
3. Il termine “vincitori” può essere espresso più propriamente con “squadre vincitrici”
4. Il termine “partita” è generale e potrebbe essere impiegato anche qualora si stia parlando di prenotazioni, che di fatto non occorre arbitrare e di cui non si è interessati tener traccia sotto il punto di vista dell'esecuzione; in definitiva, per “partita” si intende la “partita di un torneo”
5. “Match” è un sinonimo di “partita (di un torneo)”
6. In questo caso sarebbe più corretto parlare di “partite (di un torneo)”

I requisiti possono ora essere riscritti eliminando le ambiguità. Si possono anche riformulare **raggruppando requisiti omogenei**:

*SportCore è un centro sportivo che offre strutture e servizi per gli sportivi.*

***(Campi da gioco)** Dispone attualmente di diversi campi da gioco: quattro campi in sabbia (per beach tennis, beach volley, foot volley..), un campo da calcio a 5 e due campi da calcio a 7. E' in programma la costruzione di ulteriori campi ed impianti. E' data la possibilità di prenotare campi da gioco, ai quali sono associate tariffe orarie. Le prenotazioni possono essere effettuate anche via web.*

***(Tornei)** Di tanto in tanto vengono istituiti tornei, a cui le squadre possono iscriversi (anche via web) sostenendo una certa tassa d'iscrizione. Al termine di un torneo, le squadre vincitrici avranno diritto a un premio in denaro.*

***(Partite)** Le partite di un torneo possono anche essere distribuite su più campi da gioco. Si vuole inoltre assegnare un arbitro ad ogni partita del torneo, il quale riporterà un referto con il risultato della stessa ed eventuali note (ammonizioni, penalità ecc..).*

*E' importante garantire che non vi sia sovrapposizione tra le partite previste dalla programmazione dei tornei e le prenotazioni dei campi di gioco.*

### **1.3. Tipologie di utenti ed estensione dei requisiti**

Si ritorna dal committente per risolvere alcuni dubbi e questioni che potrebbero essere scaturiti da

un primo ciclo di analisi dei requisiti. In particolare, i requisiti descrivono la possibilità di impiegare lo strumento **web** per interagire con SportCore, ad esempio per effettuare una prenotazione. Questo aspetto non è stato sviluppato durante il primo incontro con il committente, così come altri aspetti di gestione. Si vuole ora analizzare in dettaglio le parti mancanti.

Ad esempio si possono impiegare le **user stories** come metodo di analisi.

Ecco come potrebbe svolgersi la prenotazione di un campo da gioco via web:

1. Un visitatore si collega al sito di SportCore, visualizza le tariffe e clicca su “Effettua una prenotazione”
2. Il visitatore seleziona un campo da prenotare, il giorno, l'ora di inizio e l'ora di fine.
3. ... ???

Già a questo punto sorgono alcuni dubbi:

- Come si determina l'autenticità della prenotazione?

Si possono adottare svariati approcci. Ad esempio:

1. Il visitatore potrebbe fornire un numero di telefono. La segretaria allora telefonerebbe per effettuare un accertamento.
  2. Si potrebbe richiedere un pagamento anticipato.
  3. Si potrebbe richiedere la registrazione di un account alla segreteria (non sul web). Il visitatore allora potrebbe collegarsi sul sito attraverso un login e procedere con la prenotazione.
- Diventa chiaro che le prenotazioni non possono essere ritenute valide/agibili in automatico. Prima di tutto non possono sovrapporsi ad altre prenotazioni o alle partite dei tornei. Questo si può pensare possa venire gestito dal sistema, che risponderà all'utente con un messaggio d'errore. Tuttavia, potrebbero essere forniti orari in cui il centro è chiuso o in cui si sta effettuando la manutenzione dei campi da gioco: gestire in automatico tutta la casistica non sembra conveniente.  
Per questo problema, ad esempio si potrebbe delegare alla segreteria il compito di gestire l'accettazione delle prenotazioni.
  - Si desidera tenere traccia dei pagamenti nel sistema?

Queste considerazioni suggeriscono inoltre che potrebbe essere utile effettuare l'analisi secondo due punti di vista differenti (**tipologie d'utenza** del sistema informativo):

- “amministrativo”: quello della segreteria, che si occupa di confermare le prenotazioni, istituire tornei, gestire la programmazione dei tornei ecc...
- “utente”: quello di coloro che intendono effettuare una prenotazione od iscriverne la propria squadra ad un torneo

Dopo aver discusso nuovamente con il committente, abbiamo una descrizione di tutti questi aspetti e abbiamo un'idea un po' più precisa dei compiti di amministrazione e di come sono svolte le procedure e la fruizione del servizio:

*Le **prenotazioni** dei campi possono essere svolte solamente da un **utente** registrato. La **registrazione** deve avvenire in segreteria, con un documento, per garanzia. Gli utenti sono caratterizzati da un username e da una password (automaticamente generata) che possono essere impiegati per il **login** sul sito. Le prenotazioni devono essere **confermate***

*manualmente dalla segretaria, conferma che può essere notificata all'utente attraverso un recapito (email se presente o telefono).*

*I tornei sono creati dalla segretaria che dovrà effettuare la **programmazione** delle partite (specificando una data, la durata, il campo in cui si gioca e le squadre che si scontrano) una volta che si sono chiuse le iscrizioni. I tornei hanno un nome, una data di apertura delle iscrizioni, una data di chiusura delle iscrizioni, un certo numero di posti.*

*Un utente registrato può anche **iscrivere** la propria squadra ad un torneo. Una squadra è identificata da un nome all'interno di un singolo torneo ed è composta da un certo numero di **giocatori**.*

*I pagamenti sono gestiti manualmente: di essi non se ne occupa il sistema.*

Si noti la necessità di introdurre il concetto di “Utente” e di come siano stati espansi e rifiniti concetti già introdotti (come quello di “Squadra”).

Nella precedente formulazione sono inoltre stati indicati parecchi dei **processi** principali:

- Registrazione di utenti
- Login di utenti (per effettuare prenotazioni)
- Prenotazione di campi da gioco
- Conferma delle prenotazioni
- Istituzione di un torneo
- Definizione della programmazione di un torneo
- Iscrizione di una squadra di un torneo

#### **1.4. Completamento dei requisiti: requisiti informativi, di processo, sui vincoli di integrità**

Si effettua ora il punto della situazione e si cerca, assieme al committente, di formulare i requisiti in maniera completa:

*(**Obiettivi strategici**) SportCore è un centro sportivo che offre strutture e servizi per gli sportivi. Uno degli obiettivi è rendere fruibili alcuni di questi servizi attraverso il sito web.*

*(**Utenti**) Per poter usufruire dei servizi è necessario registrare un **account utente** presso la segreteria, esibendo un documento di riconoscimento. La creazione di un account utente è un'attività amministrativa; le attività amministrative sono svolte dalla segretaria o comunque dal personale. I dati richiesti al momento della registrazione sono: **nome**, **cognome**, **username**, **data di nascita**, **telefono**, **email** (opzionale). La password è automaticamente generata durante il processo di **registrazione**. Una volta registrati, gli utenti potranno **loggarsi** al sito attraverso le proprie credenziali (username e password) ed accedere al servizio desiderato.*

*(**Campi da gioco e prenotazioni**) SportCore dispone attualmente di diversi **campi da gioco**: quattro campi in sabbia (per beach tennis, beach volley, foot volley..), un campo da calcio a 5 e due campi da calcio a 7. E' in programma la costruzione di ulteriori campi ed impianti. Ad ogni campo da gioco, identificato da una **nome** che rispetta una nomenclatura interna, è associata una **tariffa oraria**. Gli utenti possono **prenotare** un campo da gioco, specificando una data e un'ora, e una durata. Le prenotazioni devono essere **confermate** dalla segreteria,*

*che effettuerà i dovuti controlli e deciderà se cancellare la prenotazione o confermarla, notificando la decisione all'utente.*

*(**Tornei, programmazione, squadre**) SportCore può anche istituire **tornei**. I tornei sono contraddistinti da un nome, una data di apertura delle iscrizioni e una data di chiusura delle iscrizioni. Si può anche stabilire un numero fisso di squadre partecipanti. Un utente può **iscrivere** la propria squadra ad un torneo sostenendo una certa tassa d'iscrizione. Le **squadre** sono identificate da un nome all'interno di un singolo torneo e sono formate da un certo numero di **giocatori**, di cui si memorizza codice fiscale, nome, cognome, data di nascita, numero di maglia. Una volta chiuse le iscrizioni, la segreteria realizzerà la **programmazione delle partite del torneo** (specificando la data, la durata, il campo in cui si gioca e le squadre che si scontrano). Le partite di un torneo possono anche essere distribuite su più campi da gioco. Si vuole inoltre **assegnare un arbitro** ad ogni partita del torneo, il quale riporterà un referto con il risultato della stessa ed eventuali note (ammonizioni, penalità ecc.). **E' importante che non vi siano sovrapposizioni tra le partite di un torneo e le prenotazioni.** Al termine di un torneo, le squadre vincitrici avranno diritto a un premio in denaro.*

*I pagamenti sono gestiti manualmente: di essi non se ne occupa il sistema.*

In **nero** e in grigio sono evidenziati i punti principali dei **requisiti informativi** (caratteristiche dei dati), in **rosso** quelli dei **requisiti sui processi** (operazioni sui dati) e in **verde** i punti più significativi dei **requisiti sui vincoli di integrità**.

Il prossimo step è quello della modellazione concettuale dei dati.



## 2. Progettazione concettuale

### 2.1. Strategia di progetto

Sebbene i requisiti siano la base da cui partire, è bene tenere a mente che i requisiti sono spesso per loro natura **mutevoli** e possono essere soggetti a modifiche. In questo caso il problema non richiede di considerare aspetti particolarmente complessi, altrimenti avrebbe più senso operare un approccio iterativo piuttosto che un approccio a cascata.

La fase di analisi dei requisiti è servita per identificare i concetti principali e costruire una base di conoscenza del problema applicativo.

Ora si intende **modellare concettualmente la realtà di interesse**, in particolar modo gli **aspetti statici** del dominio. L'enfasi è sui dati, e infatti si parla di progettazione *guidata dai dati*: i dati sono più stabili delle applicazioni e quindi ha senso costruire le soluzioni ai problemi business ponendo su di essi l'attenzione.

La strategia di progettazione segue un **approccio ibrido**: si parte da uno scheletro che modella i concetti più importanti e si prosegue per raffinamenti ed espansioni successive.

Questa fase di progettazione accetta in input i requisiti che sono stati raccolti e sviluppati assieme al committente nella fase precedente. In output si otterrà uno **schema concettuale secondo il modello E/R** (Entity/Relationship).

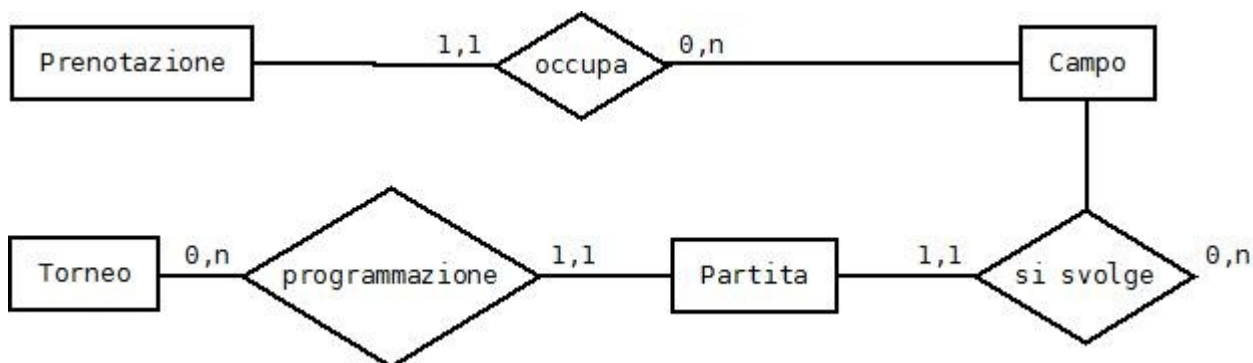
### 2.2. Glossario dei termini e schema scheletro

Prima di iniziare l'attività di modellazione, risulta utile – per meglio identificare i concetti e per visualizzare le relazioni che intercorrono tra essi – costruire un **glossario dei termini**.

Termine	Descrizione	Sinonimi	Collegamenti
Campo (da gioco)	Rappresenta un campo da gioco che può essere prenotato (ad una certa tariffa oraria) o in cui si possono svolgere le partite di un torneo.	–	Prenotazione, Partita
Utente	Persona iscritta al servizio SportCore attraverso la segreteria.	Account, membro	Prenotazione, Squadra, Torneo
Prenotazione	Operazione che consente ad un utente di impiegare, a una certa data e ora, un campo da gioco per una certa durata, sostenendo una tariffa specifica del campo da gioco.	–	Utente, Campo
Torneo	Evento che può essere istituito da SportCore a cui gli utenti possono iscrivere le proprie squadre.	–	Squadra, Utente, Partita
Squadra	Team fondato da un utente, formato da più giocatori.	Team	Utente, Torneo, Giocatore
Giocatore	Persona che rappresenta il	–	Squadra

	componente di una squadra.		
Partita	Sfida tra due squadre relativa alla programmazione di un torneo che si svolge in un campo, per una certa durata.	Match	Torneo, Campo, Squadra

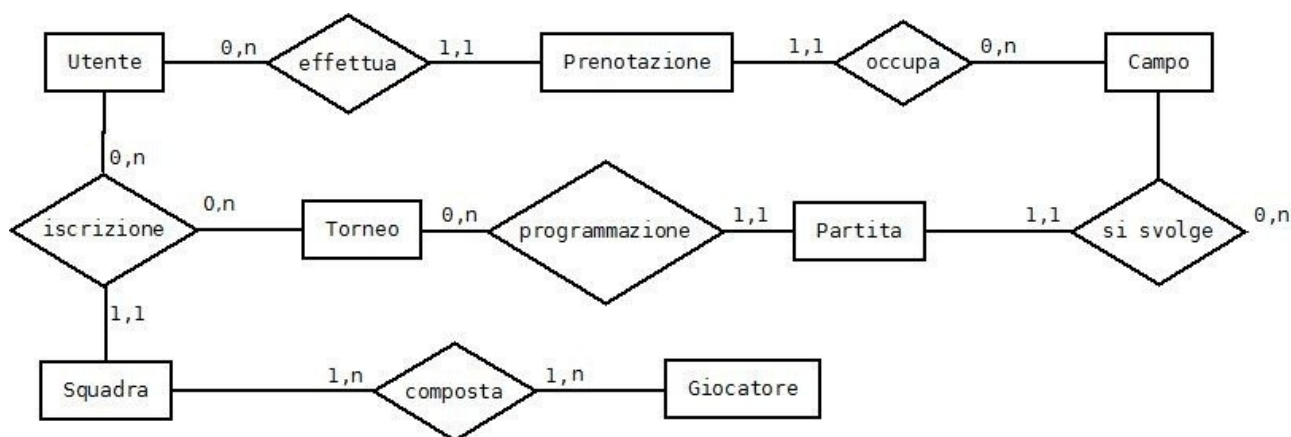
Lo schema scheletro da cui partire potrebbe modellare i concetti che si sono catturati al principio della fase di raccolta dei requisiti:



Da questo abbozzo si deduce subito chiaramente come i campi da gioco siano soggetti a prenotazioni e siano occupati dalle partite previste nella programmazione dei tornei.

Lo schema scheletro è incompleto. Esso può essere completato aggiungendo i concetti che sono stati evidenziati nella seconda fase, più approfondita, di raccolta dei requisiti. Da questo sviluppo scaturiscono inoltre quegli elementi che l'analisi dei processi ci ha indicato.

Ciò che otteniamo è uno schema che descrive ad un alto livello di astrazione il sistema che desideriamo rappresentare: sono presenti tutte le entità che intervengono nel sistema e sono indicate le associazioni che intercorrono fra esse.



## 2.3. Correzioni e sviluppo dello schema E/R

In questo momento abbiamo uno schema scheletro piuttosto completo da cui possiamo partire per ottenere lo schema concettuale finale. Possiamo cercare di applicare **correzioni** e procedere successivamente **modellando più nel dettaglio** i concetti rappresentati.

Per prima cosa possiamo notare che “Utente” e “Giocatore” sono due entità che si riferiscono a persone fisiche e, in quanto tali, possiedono attributi comuni: nome, cognome, data di nascita, codice fiscale. Il codice fiscale è necessario per identificare univocamente queste persone, anche se, vedremo, identificheremo l’“Utente” attraverso il suo username. In virtù di questi **attributi comuni**, si può sviluppare una **generalizzazione**, in cui “Utente” e “Giocatore” sono specializzazioni di “Persona”. La *copertura è totale* (le classe generalizzata è l'unione delle classi specializzate), in quanto per “Persona” intendiamo le persone che “vengono a contatto” con SportCore (qualcosa come “Cliente”), e *sovrapposta* (gli insiemi delle specializzazioni non sono necessariamente disgiunti).

La seconda cosa da notare è relativa all'associazione ternaria che coinvolge “Utente”, “Squadra” e “Torneo”. Vediamo immediatamente che “Squadra” vi partecipa con cardinalità massima a 1: già questo potrebbe innescarci il dubbio che si tratti di una **falsa ternaria**. Di fatto essa può essere sostituita con una coppia di binarie. Un “Utente iscrive una Squadra” e una “Squadra partecipa a un Torneo”; un “Utente” può iscrivere 0 o n “Squadre”, una “Squadra” è iscritta da 1 e 1 solo utente, una “Squadra” partecipa ad 1 e 1 solo “Torneo” e ad un “Torneo” partecipano 2 o n “Squadra”.

Un'**ulteriore generalizzazione** che si può fare è quella relativa a “Prenotazione” e “Partita”. Infatti, esse sono, alla base, delle attività che tengono occupato un campo da gioco (si noti anche che le associazioni verso “Campo” hanno la stessa molteplicità), con le seguenti differenze:

- (A livello di associazioni) “Prenotazione” è effettuata da un “Utente”, mentre ogni “Partita” è relativa ad un “Torneo”
- (A livello di proprietà) Della “Partita” siamo interessati a stabilire quali sono le “Squadre” che si scontrano, a conoscere il risultato della partita, l'arbitro ed eventuali note. Della “Prenotazione” vogliamo sapere se questa è stata confermata o meno.

Ci possiamo riferire alla classe più generale con il nome “Prenotazione”, e quella che è l'effettiva prenotazione (per come è stata definita precedentemente) con il nome “PrenotazioneUtente”. La *copertura è totale ed esclusiva*, ovvero non ci sono “Prenotazioni” che non siano “PrenotazioniUtente” o “Partite”, e non ci sono “PrenotazioniUtente” che siano anche “Partite”, e viceversa.

Ultima cosa ma non meno importante, nello schema scheletro ci siamo dimenticati di modellare la relazione tra “Partita” e le due “Squadre” sfidanti.

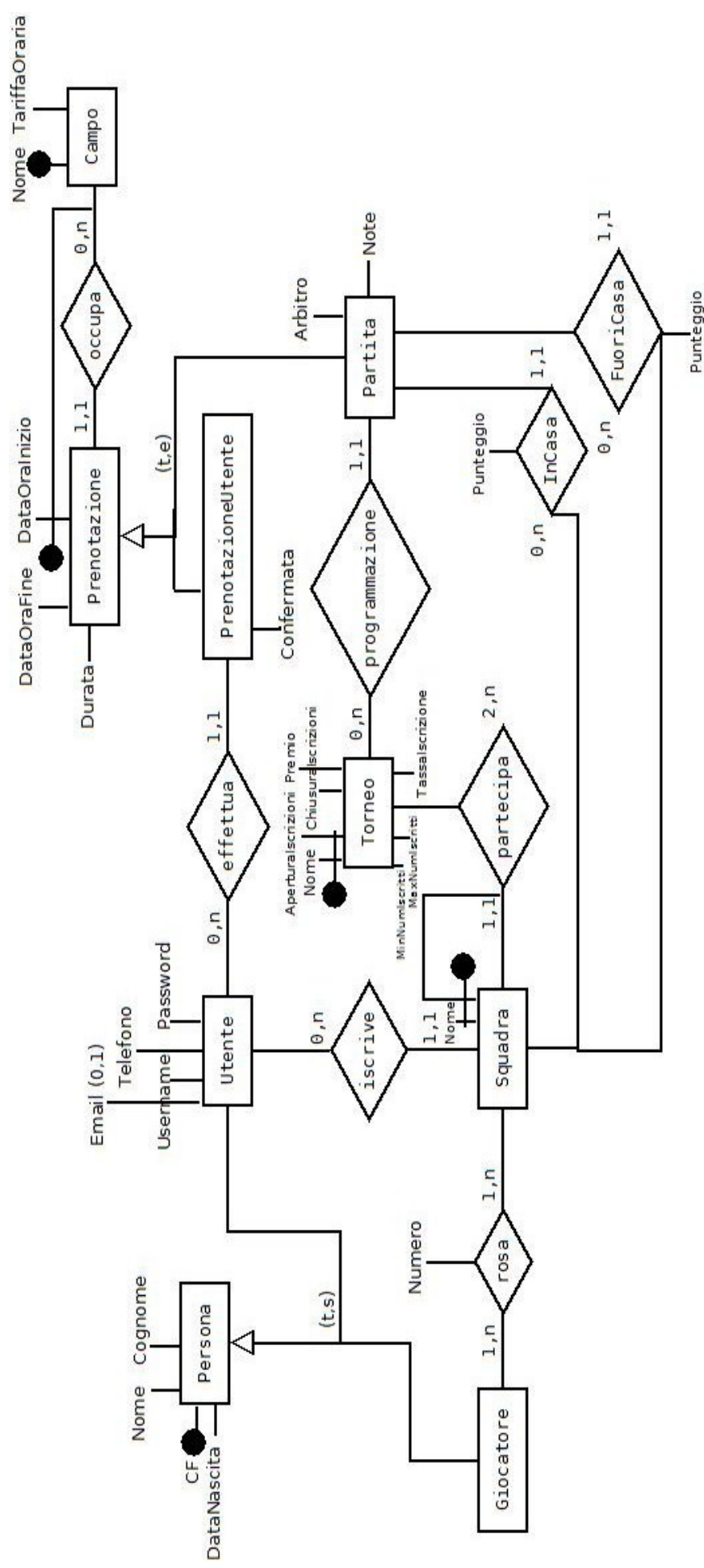
Dopo aver applicato queste correzioni, si sviluppa lo schema riportando gli **attributi** e gli **identificatori**.

Si noti come ogni “Prenotazione”, che sia una “Partita” o una “PrenotazioneUtente”, sia univocamente identificata attraverso la data, l'ora, e il campo da gioco. Si sfrutta in questo caso il vincolo di non sovrapposizione delle prenotazioni utente e delle partite dei tornei.

Ogni entità deve necessariamente avere un identificatore. Nelle gerarchie di generalizzazione l'identificatore è stato associato alla classe generale.

## 2.4. Schema concettuale finale

Dopo aver sviluppato lo schema scheletro iniziale, apportandovi correzioni e modellando nel dettaglio tutti gli aspetti, siamo giunti allo schema concettuale finale.



## 2.5. Documentazione dello schema E/R

Alleghiamo ora allo schema E/R realizzato, già di per sé abbastanza espressivo, della utile documentazione che ne possa facilitare la comprensione e che sottolinei eventuali **vincoli di integrità inespresi**.

### 2.5.1. Dizionario dei dati

Riportiamo una tabella per le entità ed una tabella per le associazioni.

Entità	Descrizione	Attributi	Identificatore
Persona	Rappresenta una persona fisica.	CF, Nome, Cognome, DataNascita	CF
Persona → Giocatore	Componente di una squadra.	<i>CF, Nome, Cognome, DataNascita</i>	<i>CF</i>
Persona → Utente	Persona iscritta a SportCore.	<i>CF, Nome, Cognome, DataNascita, Username, Password, Telefono, Email (opz.)</i>	<i>CF</i>
Campo	Campo da gioco <ul style="list-style-type: none"><li>• può essere prenotato da un utente</li><li>• può ospitare le partite di un torneo</li></ul>	Nome, TariffaOraria	Nome
Prenotazione	Esprime l'occupazione di un certo campo ad una certa data e ora, per una certa durata in termini di ore.	<i>DataOraInizio, DataOraFine, Durata</i>	<i>DataOraInizio, Campo</i>
Prenotazione → Prenotazione Utente	Esprime la specifica prenotazione utente di un campo.	<i>DataOraInizio, DataOraFine, Durata, Confermata</i>	<i>DataOraInizio, Campo</i>
Prenotazione → Partita	Rappresenta una partita (in cui si sfidano due squadre, ad una certa data e ora, per una certa durata, su un certo campo) della programmazione di un torneo.	<i>DataOraInizio, DataOraFine, Durata, Note, Arbitro</i>	<i>DataOraInizio, Campo</i>
Squadra	Team composto da diversi giocatori che un utente ha iscritto ad un certo torneo.	Nome	Nome, <i>Torneo</i>
Torneo	Evento creato da SportCore in cui più squadre possono iscriversi e giocare fra loro nelle partite stabilite dalla programmazione del torneo.	Nome, AperturaIscrizioni, ChiusuraIscrizioni, TassaIscrizione, Premio, MinNumIscritti, MaxNumIscritti	Nome, AperturaIscrizioni

Relazione	Descrizione	Componenti	Attributi
Effettua (prenotazione)	Un utente può effettuare la prenotazione di un campo da gioco.	Utente, PrenotazioneUtente	–
Iscrive (squadra)	Un utente può iscrivere una squadra per farla partecipare ad un torneo.	Utente, Squadra	–
Partecipa (ad un torneo)	Una squadra partecipa ad un e un solo torneo.	Squadra, Torneo	–
Rosa	Una squadra è composta di un certo numero di giocatori.	Squadra, Giocatore	Numero
Programmazione	La programmazione di un torneo avviene assegnando al torneo delle partite da svolgersi ad una certa data, fra due squadre.	Torneo, Partita	–
InCasa	Una squadra (designata come “in casa”) ne sfida un'altra in una partita.	Squadra, Partita	Punteggio (della squadra in casa)
FuoriCasa	Una squadra (designata come “fuori casa”) ne sfida un'altra in una partita.	Squadra, Partita	Punteggio (della squadra fuori casa)
Occupi (un campo)	Ogni prenotazione impiega un campo ad una certa data e ora.	Prenotazione, Campo	–

### 2.5.2. Regole di vincolo (e di derivazione)

E' importante indicare quelle informazioni significative che non si è stati in grado di esprimere attraverso lo schema concettuale secondo il modello E/R.

E' inoltre opportuno indicare vincoli che, seppur espressi, sono particolarmente importanti nella realtà che si sta cercando di modellare.

Regole di vincolo
<ol style="list-style-type: none"> <li>Una “PrenotazioneUtente” e una “Partita” non possono svolgersi nella stessa data e ora sullo stesso campo, ovvero non possono sovrapporsi.</li> <li>Due “Squadre” iscritte a “Tornei” diversi non si possono sfidare in una “Partita”.</li> <li>Uno stesso “Giocatore” non può far parte di due “Squadre” iscritte allo stesso “Torneo”</li> </ol>
Regole di derivazione
<ol style="list-style-type: none"> <li>La data e l'ora finale, <i>DataOraFine</i>, si ottiene per le “Prenotazioni” semplicemente incrementando la data e ora d'inizio, <i>DataOraInizio</i>, per la <i>Durata</i>.</li> </ol>

Vi sono inoltre altri vincoli, intuitivi e banali, che si è deciso di non indicare. Ad esempio, la *Durata* di una prenotazione non può essere negativa; sarebbe anche opportuno che non sia frazionaria, ma questa è una considerazione atta a semplificare la gestione. O anche, le “Partite” della programmazione di un “Torneo” non possono essere programmate prima della *ChiusuraIscrizioni* di un torneo; ma l'attività di programmazione è svolta dalla segreteria ed è improbabile che ciò accada, salvo per sviste, ma non è un problema così significativo.

### 3. Progettazione logica

La fase di progettazione logica accetta in ingresso lo schema concettuale, le informazioni sul carico applicativo e il modello logico di progettazione, e produce uno **schema logico**. In generale, esso deve rappresentare fedelmente la realtà d'interesse e deve tenere in considerazione aspetti legati all'**efficienza**.

Nel nostro caso il modello logico è quello **relazionale**.

#### 3.1. Stima del volume dei dati e del carico di lavoro

La **stima del carico applicativo** è essenziale per effettuare valutazioni sull'efficienza.

Di seguito si riporta una tabella chiamata **tavola dei volumi**, che mostra il quantitativo dei dati e aiuta ad avere un'idea della dimensione della nostra base dati. I valori provengono da un processo di **stima**, in questo caso basato su valutazioni soggettive e intuitive, e rappresentano i *valori previsti a regime*.

Si noti che la presente realtà di interesse include una serie di concetti il cui volume dati è destinato a crescere in continuazione (a parte il numero di campi da gioco che rimane sensibilmente stabile).

Per questo motivo la tavola dei volumi è stata popolata su base mensile.

Concetto	Tipo	Volume
Campo	E	→ 7
PrenotazioneUtente	E	20 prenotazioni al giorno (qualcosa come una media di 2,xx per ogni campo), 6 giorni a settimana, 4 settimane al mese → 480
Utente	E	(Ad ogni prenotazione segue la creazione di un account, tuttavia si può pensare che uno stesso utente prenoti più e più volte. Ed inoltre si può pensare che il tasso di crescita del numero di utenti decresca nel tempo.) → 40
Torneo	E	(E' probabile che il numero di tornei sia più frequente in estate rispetto alle altre stagioni in cui potrebbero anche non essere istituiti) → 2
Squadra	E	16 squadre per torneo → 32
Giocatori	E	Supponiamo un torneo di beach tennis ed uno di calcio a 7, è $16 \cdot 2 + 16 \cdot 10$ (se le squadre di calcio a 7 hanno in media 10 giocatori, in fondo tre riserve servono) → 192
Partita	E	2 tornei a 16 squadre, supponiamo 4 gironi da 4, in ogni girone 6 partite, passano le prime 4, si svolgono infine le 2 semifinali e la finale



		→ 54
Effettua (una prenotazione)	R	→ 480
Occupi (un campo)	R	→ 480 + 54 = 534
Iscrive (una squadra)	R	→ 32
Composta (di giocatori)	R	→ 192
Partecipa (ad un torneo)	R	→ 32
Programmazione	R	→ 54
InCasa	R	→ 54
FuoriCasa	R	→ 54

### 3.2. Descrizione delle operazioni principali

Riportiamo la **tavola delle operazioni** per riassumere le operazioni principali che siamo interessati a svolgere e la relativa **frequenza**.

Queste informazioni possono risultare particolarmente utili per un'analisi ai fini delle valutazioni di efficienza e per decidere se mantenere o meno delle ridondanze.

Operazione	Tipo	Frequenza (/mese)
1) Controllare se ad una certa data e ora iniziale e finale è già stata registrata una “PrenotazioneUtente” o una “Partita”	I	480 + 54 (al minimo)
2) Controllare il numero di posti disponibili per un “Torneo”.	I	Stimato: 100
3) Determinare il “Campo” su cui sono state effettuate più “Prenotazioni”.	I	4 (una volta a settimana)

Le operazioni 1) e 2) sono classificabili come operazioni “di gestione” o “controllo vincoli”.

L'operazione 3) assume un ruolo “strategico”, in quanto consente di trovare il “prodotto più richiesto” e potrebbe ad esempio suggerire un aggiustamento delle tariffe.

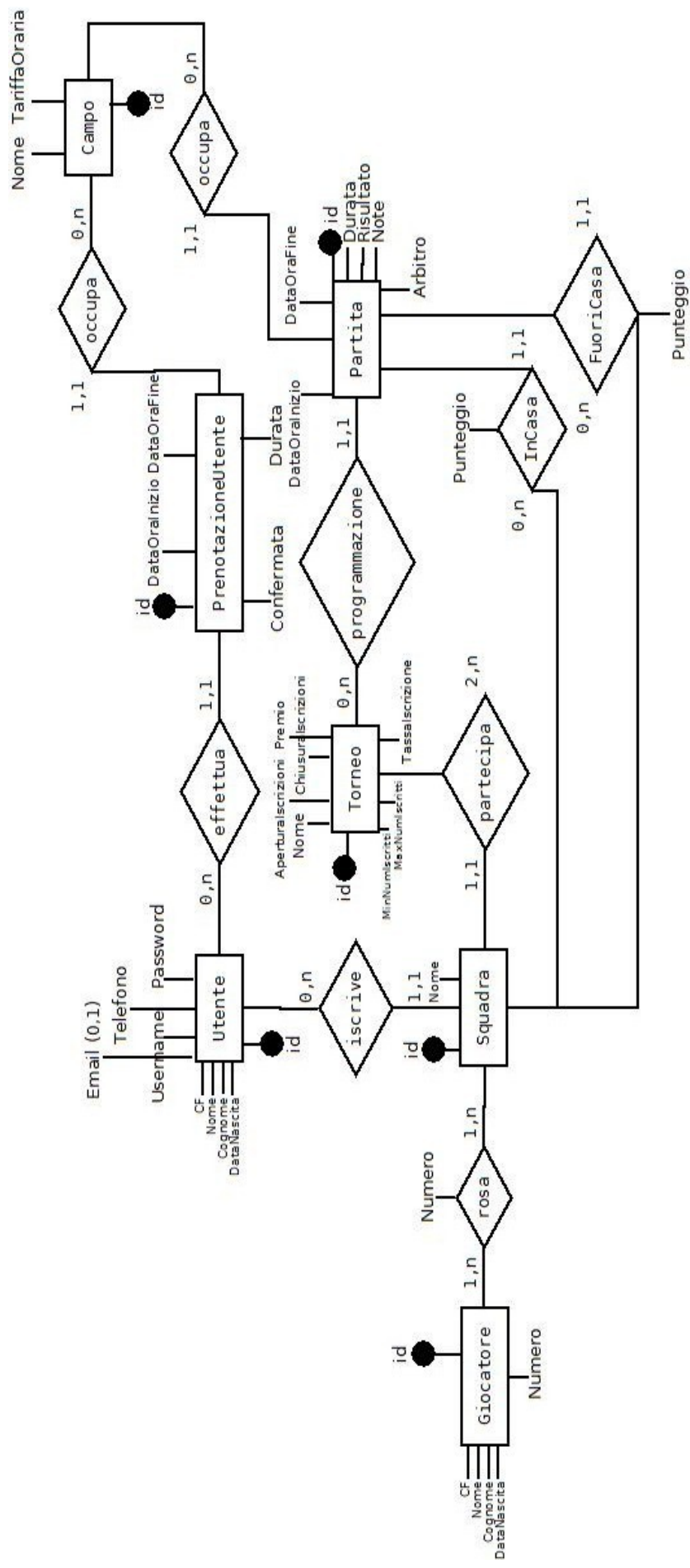
Non abbiamo particolari requisiti prestazionali a cui prestare attenzione.

### 3.3. Raffinamento dello schema

Per la traduzione nel modello relazionale occorre eliminare o convertire dallo schema concettuale tutto ciò che non può essere espresso nel modello logico, in particolar modo gerarchie di generalizzazione, attributi composti e attributi multivalore.

Prima però è opportuno effettuare un'**analisi delle ridondanze**, che consiste appunto nel decidere se mantenere o meno le ridondanze. Da una parte, mantenerle può comportare interrogazioni della base dati più semplici, dall'altra si occupa più spazio ed in generale si appesantiscono le operazioni di aggiornamento.

Relativamente alle ridondanze, una nota che si era fatta a proposito era relativa alla *DataOraFinale* delle “Prenotazioni”. Per semplicità, si decide di mantenere la ridondanza, sebbene la *DataOraFinale* si potrebbe ottenere dalla *DataOraInizio* semplicemente sommandovi la *Durata*.



Questa decisione è stata effettuata senza analisi approfondite in quanto già sappiamo che il numero di operazioni in termini di frequenza è il medesimo (non cambia la navigazione – l'attributo derivato è nella stessa relazione), e l'occupazione in termini di spazio è stimata poco significativa.

Nel nostro schema abbiamo due gerarchie di generalizzazione:

1. “Utente” e “Giocatore” sono “Persone”, con copertura totale e sovrapposta
2. “PrenotazioneUtente” e “Partita” sono “Prenotazioni”, con copertura totale ed esclusiva

Decidiamo per entrambe di accorpare il genitore nelle entità figlie, cioè operiamo un **collasso verso il basso**.

La gerarchia di “Persona” ha copertura sovrapposta, perciò il collasso verso il basso *introduce ridondanza*. Tuttavia si stima poco significativa la ridondanza, ed inoltre si preferisce marcare la distinzione delle due classi specializzate, “Giocatore” e “Utente”.

Lo schema non presenta attributi composti, né attributi multivalore.

Per il principio di località, a volte potrebbe risultare utile effettuare partizionamenti o accorpamenti di entità e associazioni. Il nostro schema sembra equilibrato, perciò non procediamo con simili operazioni.

L'ultimo passaggio, prima della traduzione, è quello relativo alla scelta degli **identificatori primari**.

Nel nostro schema attuale notiamo identificatori formati da più attributi e spesso identificatori esterni (vi è anche un'entità debole). Gli identificatori primari non possono essere opzionali e dovrebbero essere semplici, possibilmente interni. Se nessun identificatore soddisfa tali requisiti si introducono per convenienza nuovi attributi allo scopo. La necessità di avere identificatori principali **semplici** deriva dal fatto che poi questi sono utilizzati di frequente e per diverse funzioni. Quest'ultimo fatto trova riscontro soprattutto nello sviluppo web; ad esempio *ActiveRecord*, il *framework O/RM* impiegato di default in *Ruby on Rails*, richiede per default (il principio *Convention over Configuration* è usato in maniera estensiva) che le tabelle abbiano una colonna di interi chiamata *id* come chiave primaria.

Ci conviene per molte entità introdurre nuovi attributi che agiscano come identificatori principali. Per altre, come “Campo” o “Giocatore”, si potrebbero scegliere “Nome” e “CF”, rispettivamente. Tuttavia:

1. Sono stringhe, e ad esempio riportare tali valori come parametri GET nelle query string può non essere troppo comodo (escape caratteri ..), in caso vi siano spazi o caratteri strani.

O anche, memorizzare un codice fiscale nella sessione (per mantenere lo stato tra le varie richieste HTTP), non è esattamente la cosa più confortevole.

Nulla di così tragico, ma perché complicarsi le cose?

2. Meglio adottare un approccio (stilistico) consistente

In virtù di queste considerazioni, introduciamo per ogni entità un attributo “id” intero che agisce come identificatore primario.

Segue lo schema dopo avervi applicato le operazioni di adattamento precedentemente discusse:

Come precedentemente descritto, le generalizzazioni sono state eliminate attraverso un collasso verso il basso, e sono stati aggiunti nuovi attributi che agiscano come identificatori primari.

### 3.4. Traduzione di entità e associazioni in relazioni

Dopo aver tolto dallo schema concettuale tutto ciò che non era esprimibile nel modello relazionale, possiamo operare una **traduzione** delle entità e associazioni in relazioni.

L'operazione è piuttosto diretta:

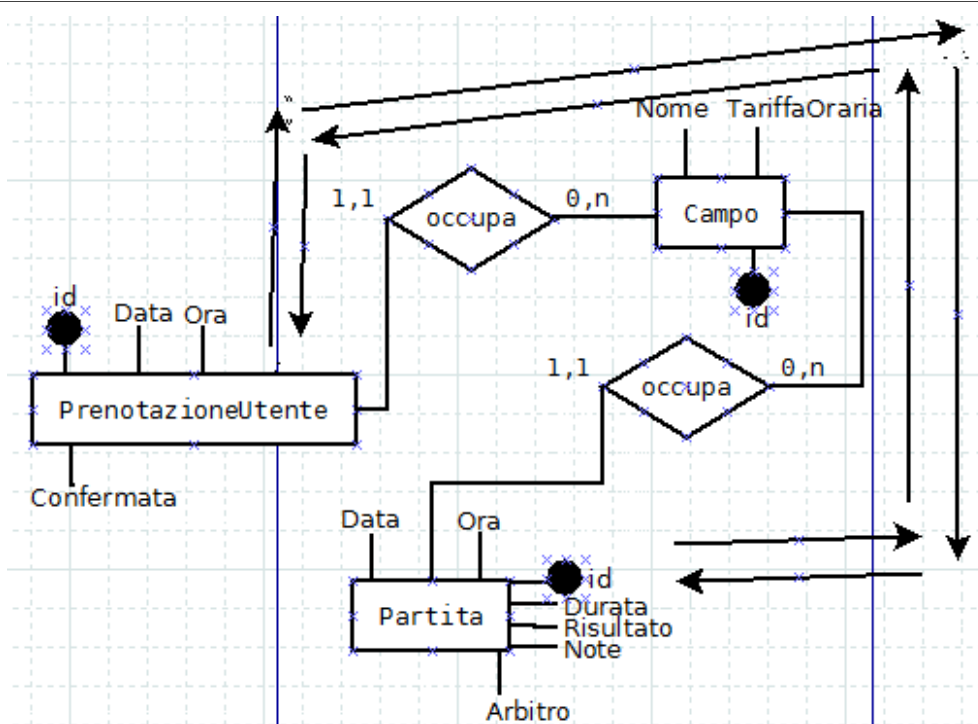
Entità/Associazione	Traduzione nel modello relazionale
Utente	Utenti (Id, CF, Nome, Cognome, DataNascita, Username, Password, Telefono, Email*) PK: Id  AK: CF AK: Username AK: Email
Campo	Campi (Id, Nome, TariffaOraria) PK: Id  AK: Nome
PrenotazioneUtente	PrenotazioniUtente (Id, DataOraInizio, DataOraFine, Durata, Confermata, Utente, Campo) PK: Id FK: Utente REFERENCES Utenti(Id) FK: Campo REFERENCES Campi(Id)  AK: DataOraInizio, Campo
Torneo	Tornei (Id, Nome, AperturaIscrizioni, ChiusuraIscrizioni, TassaIscrizione, Premio, MinNumIscritti, MaxNumIscritti) PK: Id  AK: Nome, AperturaIscrizioni
Squadra	Squadre (Id, Nome, Utente, Torneo) PK: Id FK: Utente REFERENCES Utente(Id) FK: Torneo REFERENCES Torneo(Id)  AK: Nome, Torneo
Partita	Partite (Id, DataOraInizio, DataOraFine, Campo, Durata, Arbitro, Note,

	Torneo, SquadraInCasa, SquadraFuoriCasa, PunteggioSquadraInCasa, PunteggioSquadraFuoriCasa) PK: Id FK: Campo REFERENCES Campo(Id) FK: Torneo REFERENCES Torneo(Id) FK: SquadraInCasa REFERENCES Squadra(Id) FK: SquadraFuoriCasa REFERENCES Squadra(Id)  AK: DataOraInizio, Campo AK: DataOraInizio, SquadraInCasa AK: DataOraInizio, SquadraFuoriCasa
Giocatore	Giocatori (Id, CF, Nome, Cognome, DataNascita) PK: Id  AK: CF
Rosa	Rosa (Giocatore, Squadra, Numero) PK: Giocatore, Squadra FK: Giocatore REFERENCES Giocatore(Id) FK: Squadra REFERENCES Squadra(Id)

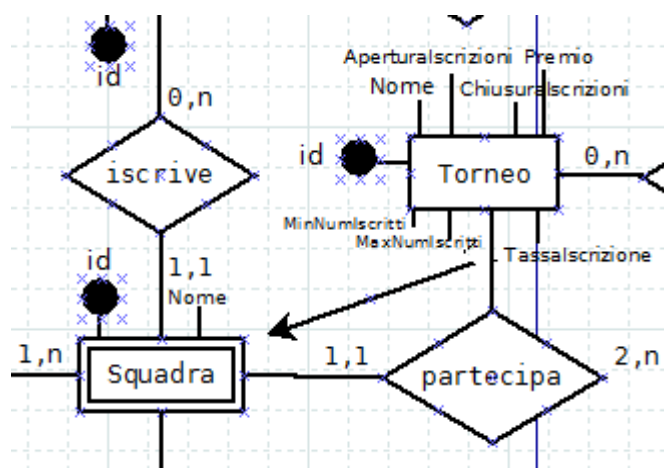
### 3.6. Schemi di navigazione

Gli schemi di navigazione mostrano in uno schema concettuale E/R il cammino “logico” impiegato per accedere alle informazioni di interesse.

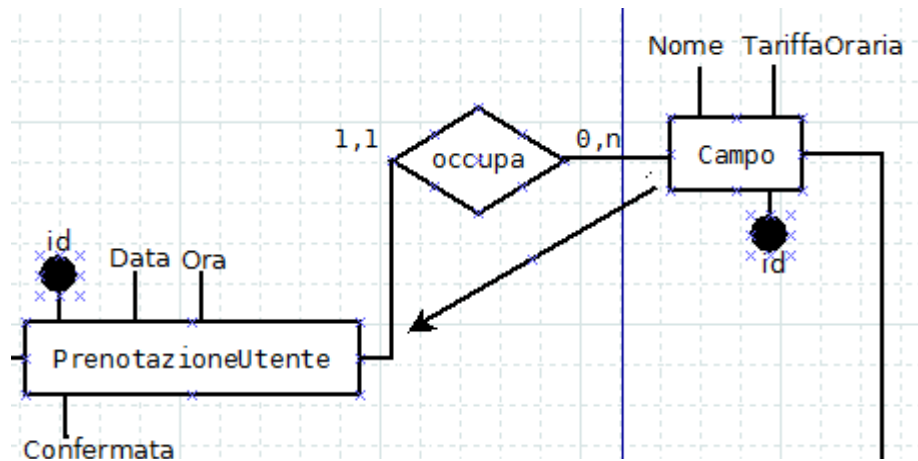
Operazione
1) Controllare se ad una certa data e ora iniziale e finale, su un certo campo, è già stata registrata una “PrenotazioneUtente” o una “Partita”. (Lo schema qui è generale, e suppone che tali informazioni di input vengano prelevate da “PrenotazioneUtente” o “Partita”, ma in generale potrebbero essere parametri stabiliti da un'utente attraverso un'apposita interfaccia.)



2) Controllare il numero di posti disponibili per un “Torneo”.



3) Determinare il “Campo” su cui sono state effettuate più “Prenotazioni”.



Queste informazioni possono essere riassunte nella cosiddetta **tavola degli accessi**.

Op.	Concetto	Costrutto	Accessi	Tipo
1) Controllare se ad una certa data e ora, su un certo campo, è già stata registrata una “PrenotazioneUtente” o una “Partita”.	PrenotazioneUtente (Partita) Campo	E (E) E	480 (54) 1	L (L) L
2) Controllare il numero di posti disponibili per un “Torneo”.	Torneo Squadra	E E	1 16	L L
3) Determinare il “Campo” su cui sono state effettuate più “Prenotazioni”.	Campo PrenotazioneUtente	E E	1 480	L L

### 3.7. Traduzione delle operazioni in query SQL

1) Controllare se ad una certa data e ora iniziale e finale, su un certo campo, è già stata registrata una “PrenotazioneUtente” o una “Partita”.

Sicuramente non ci sono sovrapposizioni nel caso la *DataOraFine* di una “Partita” o “PrenotazioneUtente” anticipi la *DataOraInizio* di una prenotazione da effettuare. Stessa cosa nel caso la *DataOraInizio* posticipi la *DataOraFine* della prenotazione da registrare.

```

SELECT DataOraInizio, DataOraFine, Campo
FROM PrenotazioneUtente
WHERE DataOraFine > @DataOraInizio AND DataOraInizio <
@DataOraFine AND Campo = @Campo
  
```

---

UNION

```
SELECT DataOraInizio, DataOraFine, Campo
FROM Partita
WHERE DataOraFine > @DataOraInizio AND DataOraInizio <
@DataOraFine AND Campo = @Campo
```

---

2) Controllare il numero di posti disponibili per un “Torneo”.

Qui occorre sottrarre al numero massimo di posti disponibili (*MaxNumIscritti*) il numero di “Squadre” iscritte.

```
SELECT (MaxNumIscritti - (
    SELECT COUNT(*)
    FROM Squadra
    WHERE Torneo = @IdTorneo
)) AS PostiDisponibili
FROM Torneo
WHERE Id = @IdTorneo
```

---

3) Determinare il “Campo” (o i “Campi”) su cui sono state effettuate più “PrenotazioniUtente”.

```
CREATE VIEW NumPrenotazioni(Campo, Num) AS
SELECT Campo, COUNT(*)
FROM PrenotazioneUtente
WHERE Confermata='true'
GROUP BY Campo
```

```
SELECT C.*
FROM NumPrenotazioni N, Campo C
WHERE Num = (
    SELECT MAX(Num)
    FROM NumPrenotazioni
) AND N.Campo = C.Id
```

---



## 4. Progettazione dell'interfaccia

### 4.1. Modifiche allo schema relazionale

Durante la fase di implementazione può capitare di accorgersi che qualcosa manca, che il sistema informativo potrebbe offrire ulteriori informazioni, ad esempio finalizzate a rendere l'interfaccia più *user-friendly* o semplicemente completare l'accesso informativo agli utenti.

Non è neppure il modo sbagliato di procedere. Nell'ingegneria del software sono noti i **problemi** e le difficoltà che comporta un **approccio a cascata**, che derivano sostanzialmente dall'impossibilità di conoscere completamente i requisiti all'inizio del progetto, per il fatto che non si possa conoscere esattamente tutto nei minimi particolari, dal momento che le persone non sono infallibili, possono commettere errori, e non si può dare per scontato che gli analisti e i progettisti abbiano compreso a fondo le esigenze del cliente, il quale potrebbe tra l'altro comunicarle in maniera poco precisa, o potrebbe anch'esso cambiare nel tempo le proprie idee.

Per questo motivo sono stati preferiti nel tempo **approcci iterativi**, basati sul concetto di *iterazione*. Ogni iterazione include una fase di raccolta di requisiti, di analisi, di progettazione, di implementazione (e di testing, di integrazione, di rilascio). In questo modo è possibile ritardare le decisioni, e sfruttare per l'analisi le informazioni che sono scaturite dalla progettazione e dall'implementazione. Inoltre, ci si focalizza su un piccolo sottoinsieme di funzionalità alla volta.

Riassumendo, i requisiti sono soggetti al cambiamento, perciò occorre adottare tecniche che prendano in considerazione questo aspetto.

(Parentesi) A proposito, è stato sviluppato nel tempo il concetto di **agility**, e sono nate le cosiddette metodologie agili (XP, SCRUM, Crystal...). L'Agile Manifesto riassume perfettamente la filosofia alla base:

- Individuals and interactions over processes and tools.
- Working software over comprehensive documentation.
- Customer collaboration over contract negotiation.
- **Responding to change over following a plan.**

Le modifiche allo schema relazionale sono riportate nella seguente tabella.

Tabella	Variazione
Campi	Aggiunta di un attributo "NomeUserFriendly" (ricordo che "Nome" seguiva una nomenclatura interna).
Campi	Aggiunta di un attributo "Descrizione", che offra informazioni aggiuntive sul campo in questione (dimensioni, info sulla struttura, il terreno...).
Campi	Aggiunta di un attributo "ImgUrl", che contenga l'URL relativo a un'immagine del campo.

## 4.2. Strategia di realizzazione

Attraverso il progetto logico abbiamo dato una struttura alla nostra base di dati.

Ora è necessario fornire un'**interfaccia** che offra un più alto livello di astrazione sui nostri dati. Da un punto di vista utente vogliamo dare la possibilità di accedere in modo semplice ed immediato al servizio; da un punto di vista amministrativo vogliamo migliorare l'efficacia della gestione.

La nostra interfaccia dovrà essere **usabile** e nascondere i dettagli implementativi.

L'interfaccia è **web-based**. L'environment è composto da

- **Apache** come web server
- **PHP** come linguaggio di programmazione server-side
- **MySQL** come RDBMS

Come sottolineato durante la fase di analisi dei requisiti, il nostro sistema informativo prevede due tipologie di utenza: utente e amministratore. Conseguentemente, l'interfaccia sarà composta di due parti:

- **Front-end**: costituito dalle pagine web attraverso cui l'utente interagisce con il servizio
- **Back-end**: area di amministrazione in cui viene svolta la gestione