# Progetto di Sistema Informativo per Centro Sportivo

## 1. Analisi e progettazione concettuale

In questa sezione si descrive il modello concettuale dei dati rappresentato attraverso il diagramma ER. Si identificano le principali entità, le relazioni e i vincoli fondamentali.

### Descrizione del contesto

Il progetto riguarda la **digitalizzazione della gestione di un centro sportivo polifunzionale**, una struttura in cui vengono messi a disposizione diversi impianti (come campi da calcio, tennis, padel, basket, pallavolo) per l’uso da parte di clienti registrati.

Il centro sportivo non si limita solo all’affitto degli spazi, ma organizza anche **eventi sportivi**, gestisce un **team di personale con ruoli distinti** (manutenzione, reception, istruttori, manager, ecc.) e necessita di un sistema informativo in grado di **monitorare prenotazioni, iscrizioni e disponibilità impianti.**

L’obiettivo principale del sistema è quello di:

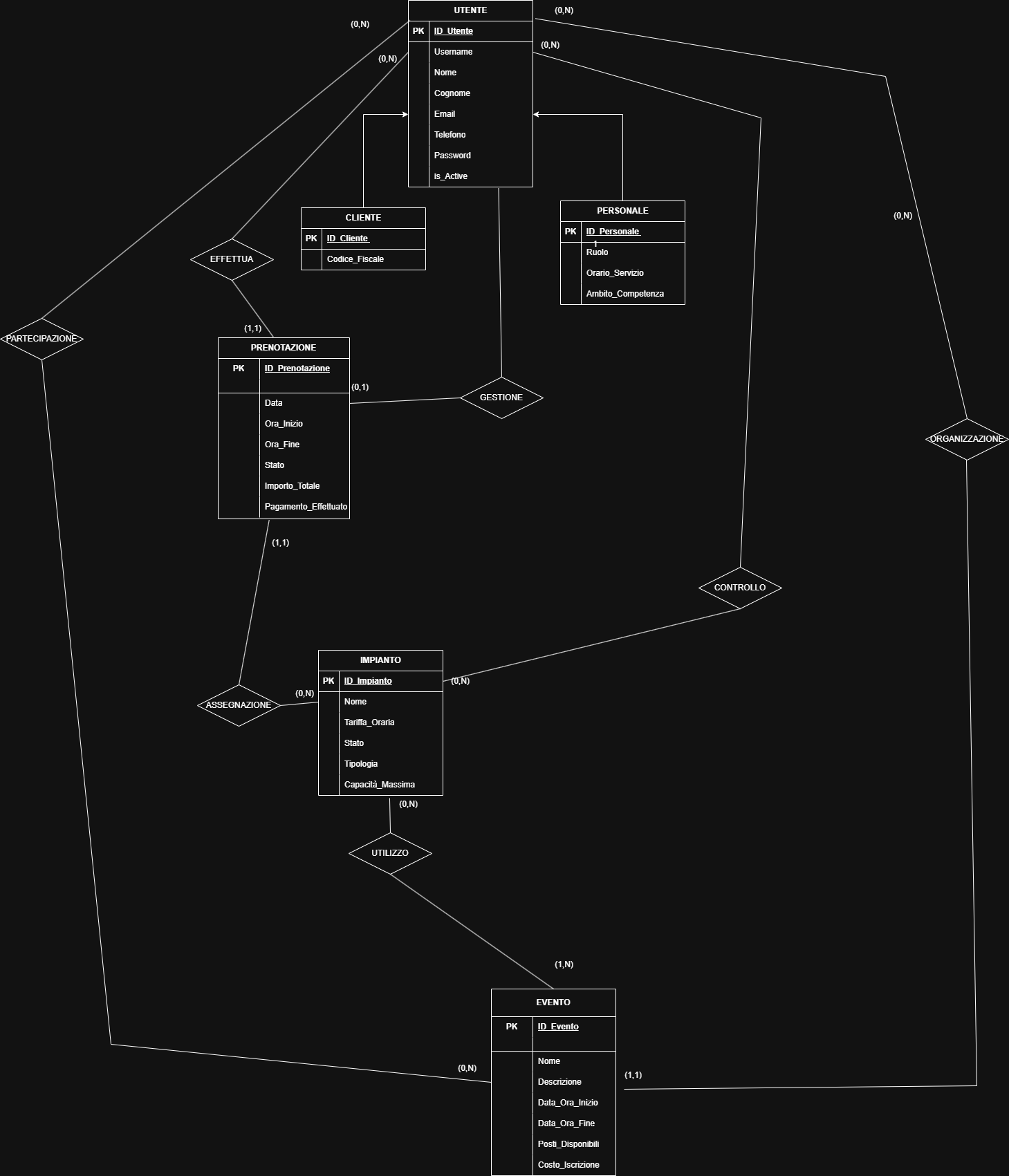
* consentire ai **clienti** di prenotare impianti sportivi e partecipare a eventi;
* permettere al **personale** di organizzare eventi, gestire gli impianti e monitorare lo stato delle prenotazioni;
* garantire un controllo **efficiente, centralizzato e tracciabile** di tutte le attività del centro.

L’analisi del dominio ha evidenziato le seguenti aree funzionali fondamentali:

* **Gestione utenti**: distinzione tra clienti e personale, con dati anagrafici e accesso autenticato.
* **Prenotazioni**: processo di richiesta, verifica disponibilità e conferma dell’utilizzo di impianti.
* **Organizzazione eventi**: attività gestite dal personale con iscrizioni aperte ai clienti.
* **Gestione impianti**: monitoraggio della disponibilità, assegnazione del personale e uso durante eventi.

Questa analisi ha portato alla definizione delle principali entità del modello informativo, illustrate nei diagrammi ER successivi.

### Diagramma ER



**Spiegazione delle relazioni principali:**

* Un **cliente** può effettuare molte prenotazioni.
* Ogni **prenotazione** è riferita a un solo cliente e a un solo impianto.
* Un **impianto** può essere prenotato da più clienti e gestito da più membri del personale.
* Gli **eventi** sono organizzati da membri del personale e coinvolgono più clienti.

Nel primo diagramma ER, è stata modellata un’entità astratta **Utente**, da cui derivano per specializzazione le entità **Cliente** e **Personale**.

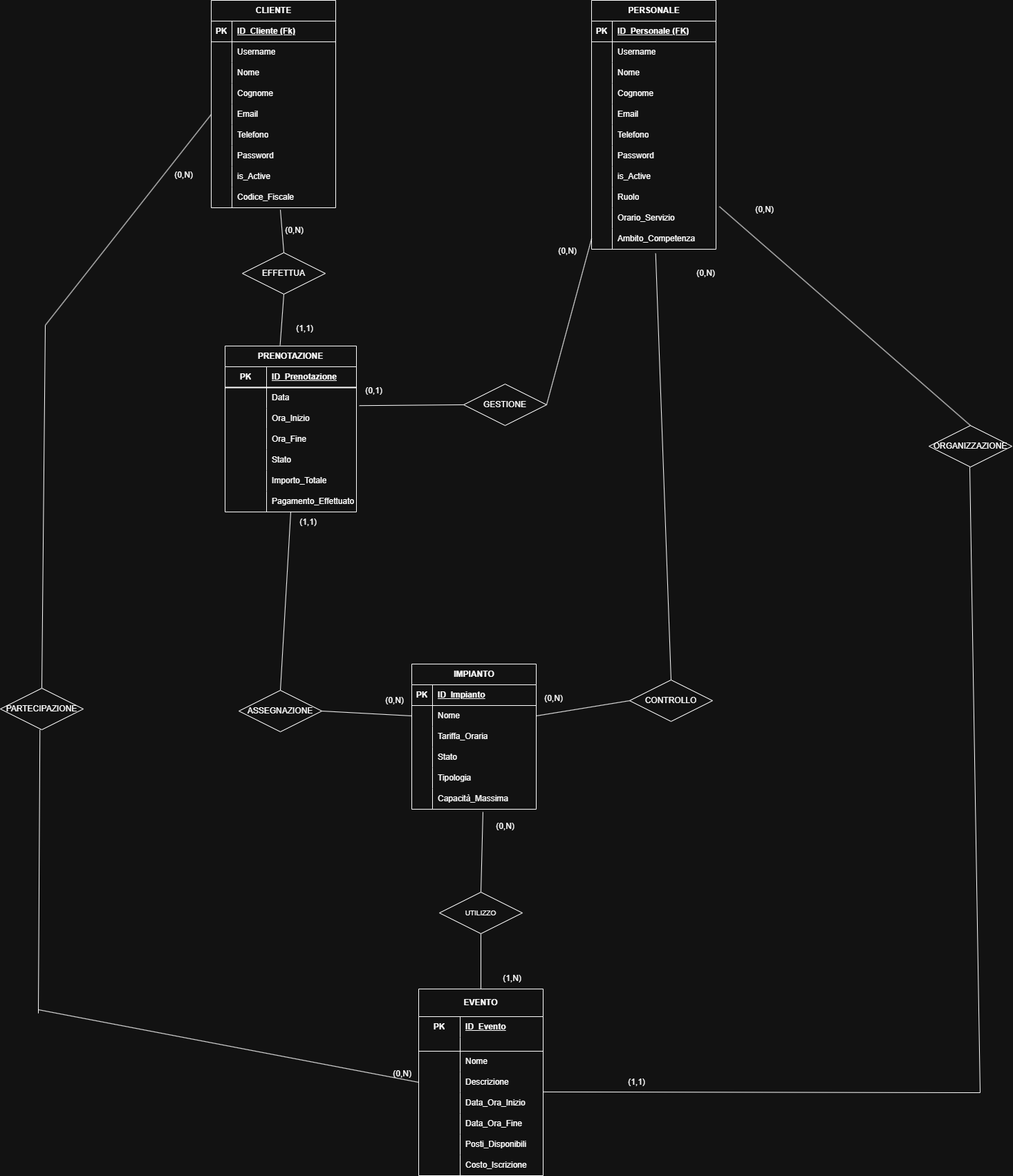
La generalizzazione è stata pensata come:

* **Totale**: ogni utente deve essere o un cliente o un membro del personale.
* **Disgiunta**: un utente può appartenere solo a uno dei due sottotipi.

Questa scelta iniziale rispecchia l’idea che esistano **attributi e comportamenti comuni** a clienti e personale (es. nome, email, credenziali di accesso), ma anche campi distintivi (es. ruolo, codice fiscale, ecc.).

## Diagramma ER (con specializzazione risolta)

* La generalizzazione inizialmente prevista tra Utente, Cliente e Personale è stata risolta accorpando direttamente gli attributi e le funzionalità all'interno delle due classi figlie. La decisione è motivata dalla natura **totale e disgiunta** della generalizzazione (un utente è obbligatoriamente un cliente o un membro del personale, e mai entrambi).
* Cliente e Personale hanno sì alcuni attributi simili (username, nome, email, password) ma anche molti attributi specifici (es. codice\_fiscale, ambito\_competenza, ruolo), se gli attributi condivisi sono pochi o non giustificano l'astrazione, meglio tenere classi separate.
* Se si fosse mantenuta una tabella Utente con chiavi esterne opzionali a Cliente e Personale, si avrebbe l’obbligatorietà nel garantire che **ogni utente appartenga a una sola sottoclasse**.  
  Inoltre, si sarebbero dovuto gestire **vincoli complessi** per assicurasi che utente\_id non esista contemporaneamente in entrambe le tabelle.  
  Accorpare direttamente nei figli **elimina** questa complessità.



### 2. Progettazione Logica

Derivazione dal modello E-R e descrizione dei vincoli.

### 2.1 Entità principali

**CLIENTE(**

id INTEGER PRIMARY KEY,

username VARCHAR(15) UNIQUE NOT NULL,

nome VARCHAR(20) NOT NULL,

cognome VARCHAR(20) NOT NULL,

email VARCHAR(255) UNIQUE NOT NULL,

telefono VARCHAR(15),

password VARCHAR(128) NOT NULL,

is\_active BOOLEAN DEFAULT TRUE,

codice\_fiscale VARCHAR(16),

data\_registrazione DATETIME NOT NULL,

ultimo\_accesso DATETIME

)

**PERSONALE (**

id INTEGER PRIMARY KEY,

username VARCHAR(150) UNIQUE NOT NULL,

nome VARCHAR(30) NOT NULL,

cognome VARCHAR(30) NOT NULL,

email VARCHAR(255) UNIQUE NOT NULL,

telefono VARCHAR(15),

password VARCHAR(128) NOT NULL,

is\_active BOOLEAN DEFAULT TRUE,

ruolo VARCHAR(30) NOT NULL,

orario\_servizio VARCHAR(100) NOT NULL,

ambito\_competenza VARCHAR(200) NOT NULL,

data\_assunzione DATETIME NOT NULL,

ultimo\_accesso DATETIME

)

**IMPIANTO(**

id INTEGER PRIMARY KEY,

nome VARCHAR(100) NOT NULL,

tipologia VARCHAR(20) NOT NULL,

capacita\_massima INTEGER NOT NULL,

stato VARCHAR(20) DEFAULT 'disponibile',

tariffa\_oraria DECIMAL(6,2) NOT NULL

)

**EVENTO(**

id INTEGER PRIMARY KEY,

organizzatore\_id INTEGER,

nome VARCHAR(200) NOT NULL,

descrizione TEXT NOT NULL,

tipo VARCHAR(20) DEFAULT 'NULL',

data\_ora\_inizio DATETIME NOT NULL,

data\_ora\_fine DATETIME NOT NULL,

posti\_disponibili INTEGER NOT NULL,

costo\_iscrizione DECIMAL(6,2) DEFAULT 0.00,

data\_creazione DATETIME NOT NULL,

data\_modifica DATETIME NOT NULL,

FOREIGN KEY (organizzatore\_id) REFERENCES PERSONALE(id) ON DELETE SET NULL

)

**PRENOTAZIONE(**

id INTEGER PRIMARY KEY,

cliente\_id INTEGER NOT NULL,

impianto\_id INTEGER NOT NULL,

data DATE NOT NULL,

ora\_inizio TIME NOT NULL,

ora\_fine TIME NOT NULL,

stato VARCHAR(15) DEFAULT 'attesa',

importo\_totale DECIMAL(8,2) NOT NULL,

pagamento\_effettuato BOOLEAN DEFAULT FALSE,

gestita\_da\_id INTEGER,

data\_creazione DATETIME NOT NULL,

data\_modifica DATETIME NOT NULL,

FOREIGN KEY (cliente\_id) REFERENCES CLIENTE(id) ON DELETE CASCADE,

FOREIGN KEY (impianto\_id) REFERENCES IMPIANTO(id) ON DELETE CASCADE,

FOREIGN KEY (gestita\_da\_id) REFERENCES PERSONALE(id) ON DELETE SET NULL,

UNIQUE (impianto\_id, data, ora\_inizio)

)

**PARTECIPAZIONE(**

id INTEGER PRIMARY KEY,

cliente\_id INTEGER NOT NULL,

evento\_id INTEGER NOT NULL,

data\_iscrizione DATETIME NOT NULL,

presente BOOLEAN DEFAULT FALSE,

FOREIGN KEY (cliente\_id) REFERENCES CLIENTE(id) ON DELETE CASCADE,

FOREIGN KEY (evento\_id) REFERENCES EVENTO(id) ON DELETE CASCADE,

UNIQUE (cliente\_id, evento\_id)

)

**UTILIZZO\_IMPIANTO\_EVENTO(**

id INTEGER PRIMARY KEY,

evento\_id INTEGER NOT NULL,

impianto\_id INTEGER NOT NULL,

FOREIGN KEY (evento\_id) REFERENCES EVENTO(id) ON DELETE CASCADE,

FOREIGN KEY (impianto\_id) REFERENCES IMPIANTO(id) ON DELETE CASCADE,

UNIQUE (evento\_id, impianto\_id)

)

**GESTIONE\_IMPIANTO(**

id INTEGER PRIMARY KEY,

personale\_id INTEGER NOT NULL,

impianto\_id INTEGER NOT NULL,

data\_assegnazione DATE NOT NULL,

note TEXT,

FOREIGN KEY (personale\_id) REFERENCES PERSONALE(id) ON DELETE CASCADE,

FOREIGN KEY (impianto\_id) REFERENCES IMPIANTO(id) ON DELETE CASCADE,

UNIQUE (personale\_id, impianto\_id)

)

### 2.2 Vincoli principali

* Il sistema implementa un controllo rigoroso degli accessi dove ogni utente deve essere autenticato tramite sessione per accedere alle funzionalità protette.   
  Il sistema distingue tra clienti e personale, garantendo che ogni categoria possa accedere solo alle funzionalità appropriate al proprio ruolo.
* I clienti possono visualizzare, modificare o annullare esclusivamente le proprie prenotazioni e iscrizioni agli eventi.
* Per le iscrizioni agli eventi, il sistema verifica che un cliente non possa iscriversi più volte allo stesso evento.
* Una prenotazione annullata non può essere ulteriormente modificata.
* Le iscrizioni agli eventi si chiudono automaticamente dopo l'inizio dell'evento.
* I clienti non possono cancellare iscrizioni a eventi già iniziati.
* Solo i manager e i responsabili prenotazioni possono modificare o annullare prenotazioni, mentre altri ruoli hanno accessi più limitati.
* Per gli eventi, controlla che ci siano ancora posti disponibili prima di permettere nuove iscrizioni.
* Prima di confermare una prenotazione, verifica che l'impianto non sia già occupato nell'orario richiesto, considerando sia le prenotazioni esistenti che lo stato dell'impianto.
* Solo l'organizzatore può eliminare i propri eventi.
* Un impianto non può essere utilizzato contemporaneamente per un evento e per prenotazioni individuali.

### 2.2.1 Vincoli di Dominio

**CLIENTE**

* username: massimo 15 caratteri, univoco, obbligatorio
* codice\_fiscale: formato alfanumerico di 16 caratteri (regex: ^[A-Z0-9]{16}$)
* email: formato email valido, univoco

**PERSONALE**

* ruolo: deve essere uno dei valori: 'responsabile\_prenotazioni', 'manutenzione', 'istruttore', 'manager', 'receptionist'
* username: massimo 150 caratteri, univoco
* email: formato email valido, univoco

**IMPIANTO**

* tipologia: deve essere uno dei valori: 'calcio', 'tennis', 'padel', 'basket', 'pallavolo'
* stato: deve essere uno dei valori: 'disponibile', 'prenotato', 'manutenzione', 'fuori\_servizio'
* capacita\_massima: numero intero positivo
* tariffa\_oraria: valore decimale positivo

**EVENTO**

* tipo: deve essere uno dei valori: 'calcio', 'tennis', 'padel', 'basket', 'pallavolo'
* posti\_disponibili: numero intero positivo
* costo\_iscrizione: valore decimale non negativo
* data\_ora\_fine > data\_ora\_inizio

**PRENOTAZIONE**

* stato: deve essere uno dei valori: 'attesa', 'annullata'
* ora\_fine > ora\_inizio. L'ora di fine deve essere successiva all'ora di inizio
* importo\_totale: valore decimale positivo
* data >= data corrente (per nuove prenotazioni)

### 2.2.3 Vincoli di Integrità Referenziale

1. **PRENOTAZIONE.cliente\_id** → **CLIENTE.id** (CASCADE su DELETE)
2. **PRENOTAZIONE.impianto\_id** → **IMPIANTO.id** (CASCADE su DELETE)
3. **PRENOTAZIONE.gestita\_da\_id** → **PERSONALE.id** (SET NULL su DELETE)
4. **EVENTO.organizzatore\_id** → **PERSONALE.id** (SET NULL su DELETE)
5. **PARTECIPAZIONE.cliente\_id** → **CLIENTE.id** (CASCADE su DELETE)
6. **PARTECIPAZIONE.evento\_id** → **EVENTO.id** (CASCADE su DELETE)
7. **UTILIZZO\_IMPIANTO\_EVENTO.evento\_id** → **EVENTO.id** (CASCADE su DELETE)
8. **UTILIZZO\_IMPIANTO\_EVENTO.impianto\_id** → **IMPIANTO.id** (CASCADE su DELETE)
9. **GESTIONE\_IMPIANTO.personale\_id** → **PERSONALE.id** (CASCADE su DELETE)
10. **GESTIONE\_IMPIANTO.impianto\_id** → **IMPIANTO.id** (CASCADE su DELETE)

### 2.3.4 Vincoli di Unicità

1. **CLIENTE**: username e email devono essere univoci
2. **PERSONALE**: username e email devono essere univoci
3. **PRENOTAZIONE**: combinazione (impianto\_id, data, ora\_inizio) deve essere univoca
4. **PARTECIPAZIONE**: combinazione (cliente\_id, evento\_id) deve essere univoca
5. **UTILIZZO\_IMPIANTO\_EVENTO**: combinazione (evento\_id, impianto\_id) deve essere univoca
6. **GESTIONE\_IMPIANTO**: combinazione (personale\_id, impianto\_id) deve essere univoca