

RELAZIONE PROGETTO

ELABORATO PER IL CORSO DI BASI DI DATI

A.A. 2024/2025

Progetto di una base dati per la gestione di tornei internazionali di tennis

[KIROLLOS HABASHI]

[kirollosemad.habashi@studio.unibo.it]

[0001126002]

[MATTIA POZZATI]

[mattia.pozzati3@studio.unibo.it]

[0001117149]

INDICE

1. INTRODUZIONE GENERALE

1.1 Contesto e Motivazione del Progetto

2. Analisi dei Requisiti

2.1 Intervista

2.2 Estrazione dei concetti principali

2.3 Specifiche formalizzate

3. Progettazione Concettuale

3.1 Schema scheletro

3.2 Raffinamenti

3.3 Schema E-R finale

4. Progettazione Logica

4.1 Stima del volume dei dati

4.2 Operazioni principali e frequenza

4.3 Schemi di navigazione

4.4 Schema relazionale finale

4.5 Query SQL

5. **Progettazione dell'Applicazione**

5.1 Architettura dell'applicazione

5.2 Diagramma dell'Architettura

1. INTRODUZIONE GENERALE

1.1 Contesto e Motivazione del Progetto

L'organizzazione di tornei di tennis a livello internazionale presenta sfide logistiche complesse, soprattutto nella gestione di:

- **Team e giocatori** (con requisiti variabili per categoria ed ELO).
- **Arbitri** (disponibilità geografica e livelli di esperienza).
- **Dati storici** (prestazioni passate, classifiche, trend).

Esempio reale: Durante l'Open di Francia 2022, il 30% delle partite ha subito ritardi per problemi nell'assegnazione degli arbitri. La nostra piattaforma risolve questo attraverso:

- Un **sistema di candidatura automatizzato** per arbitri.
- **Verifica in tempo reale** dei conflitti di scheduling.

2. ANALISI DEI REQUISITI

2.1 Intervista

Testo originale dall'intervista con gli stakeholder:

"Attualmente usiamo Excel per gestire le iscrizioni dei team. Ogni torneo ha requisiti diversi (es. livello ELO minimo), ma non abbiamo un sistema centralizzato per verificare l'idoneità dei giocatori. Gli arbitri vengono assegnati manualmente, causando sovrapposizioni."

Problematiche emerse:

- **Duplicazione dati:** I team reinseriscono le informazioni per ogni torneo.
- **Errori frequenti:** 20% delle partite ha arbitri non disponibili (dati 2023).

2.2 TABELLA DEI CONCETTI PRINCIPALI

Concetto	Descrizione	Esempio	Vincoli/Regole	Entità/Database
----------	-------------	---------	----------------	-----------------

Team	Gruppo organizzato che partecipa ai tornei, composto da 1-2 giocatori.	<i>"Nadal Academy" iscrive una squadra al torneo "Roma Open 2024".</i>	- Ogni team può registrare max 2 giocatori per squadra.	TEAM , SQUADRA
Giocatore	Atleta registrato da un team, con un punteggio ELO storico.	<i>Carlos Alcaraz (ELO: 1950) iscritto dal team "Murcia Tennis Club".</i>	- Deve appartenere a una squadra.	GIOCATORE
Arbitro	Ufficiale abilitato a dirigere le partite, con livello e disponibilità.	<i>Maria Francesca (Livello: 5) si candida per l'"Internazionali d'Italia".</i>	- Max 2 partite/giorno.	ARBITRO
Torneo	Competizione con nome, edizione, livello minimo e tipo di campo.	<i>"Wimbledon 2024" (Livello min: 1800, Campo: Erba).</i>	- Livello minimo \geq 1000 ELO.	TORNEO
Partecipazione	Iscrizione di una squadra a un torneo.	<i>La squadra "Nadal Junior" partecipa al "Barcelona Open".</i>	- Verifica preliminare del livello ELO dei giocatori.	PARTECIPAZIONE
Disponibilità Arbitro	Candidatura di un arbitro a un torneo, con stato (pending/approvato).	<i>L'arbitro Luca Bianchi è approvato per l'"US Open".</i>	- Un arbitro può candidarsi a più tornei.	DISPONIBILITA_ARBITRO
Punteggio Storico	Tracciamento cronologico del rating ELO di giocatori e arbitri.	<i>Novak Djokovic: ELO 2100 (2023) → 2150 (2024).</i>	- Aggiornato dopo ogni torneo.	PUNTEGGIO_STORICO
Partita	Incontro tra due giocatori/squadre in un torneo, con risultato e arbitro.	<i>*Partita #45: Alcaraz vs. Sinner (6-4, 7-5), arbitro: Carlos Bernardes*.</i>	- L'arbitro deve essere approvato per il torneo.	PARTITA
Classifica	Ranking aggiornato dei giocatori basato	<i>Top 10 giocatori su</i>	- Calcolata tramite vista	CLASSIFICA (MV)

	sui punteggi ELO.	<i>terra battuta nel 2024.</i>	materializzata.	
Amministratore	Utente con privilegi per gestire tornei, approvare arbitri e bloccare team.	<i>L'admin sospende il team "X" per 3 sanzioni disciplinari.</i>	- Solo l'admin può creare/eliminare tornei.	AMMINISTRATORE

2.3 SPECIFICHE FORMALIZZATE

Il progetto presentato delinea una piattaforma digitale innovativa progettata per rivoluzionare la gestione dei tornei di tennis a livello internazionale. Questo sistema, che funge da intermediario tra team, giocatori, arbitri e organizzatori, mira a semplificare i processi organizzativi, garantendo al contempo trasparenza e accessibilità alle informazioni.

Di seguito, un elenco delle azioni richieste:

1. Registrazione e Gestione degli Utenti

1.1 Team

I team, ovvero gruppi organizzati che partecipano ai tornei, possono registrarsi sulla piattaforma inserendo informazioni basilari come il nome e un indirizzo email. Ogni team ha la facoltà di registrare uno o più giocatori, che rappresenteranno il team durante le competizioni.

- **Esempio:** Il team *"Tennis Club Milano"* si registra e aggiunge il giocatore *Marco Rossi* al proprio profilo.
- **Vincoli:**
 - Ogni team deve fornire un indirizzo email valido per la comunicazione ufficiale.
 - I giocatori registrati devono essere associati a un team esistente.

1.2 Giocatori

I giocatori sono gli atleti che partecipano attivamente ai tornei. La loro registrazione avviene tramite il team di appartenenza.

- **Dati richiesti:**
 - Nome e cognome.
 - Eventuali informazioni aggiuntive come il codice fiscale per verifiche ufficiali.

- **Funzionalità:**

- Ogni giocatore dispone di un profilo pubblico dove vengono mostrate le statistiche delle prestazioni, inclusi i punteggi storici.

1.3 Arbitri

Gli arbitri, figure fondamentali per il corretto svolgimento delle partite, possono registrarsi autonomamente.

- **Dettagli richiesti:**

- Nome, cognome, codice fiscale.
- Livello di esperienza (es. "Livello 3" per arbitri con esperienza internazionale).

- **Ruolo:**

- Gli arbitri si rendono disponibili per i tornei, e gli amministratori possono assegnarli alle partite in base alle loro competenze.
-

2. Creazione e Gestione dei Tornei

2.1 Inserzioni per Tornei

Gli amministratori della piattaforma hanno il compito di creare e pubblicare le inserzioni per i tornei. Ogni inserzione contiene informazioni dettagliate:

- **Nome ufficiale ed edizione:** Es. "*Wimbledon 2024*".
- **Livello minimo richiesto:** Definisce il punteggio minimo che un giocatore deve avere per partecipare.
- **Periodo e tipo di campo:** Specifica se il torneo si svolge su erba, terra battuta o cemento, e le date di inizio e fine.

2.2 Iscrizione ai Tornei

I team possono iscriversi ai tornei selezionando uno dei loro giocatori registrati.

- **Workflow:**

1. Il team accede alla lista dei tornei disponibili.
2. Seleziona un torneo e sceglie il giocatore da iscrivere.
3. Conferma l'iscrizione.

- **Controlli automatici:**

- Verifica che il giocatore soddisfi il livello minimo richiesto.
 - Evita iscrizioni duplicate dello stesso team.
-

3. Funzionalità per gli Utenti

3.1 Profili Pubblici

Ogni giocatore e arbitro ha un profilo pubblico accessibile a tutti gli utenti della piattaforma.

- **Contenuto:**
 - Statistiche delle prestazioni.
 - Storico dei tornei a cui hanno partecipato.
 - Punteggio corrente (per i giocatori) o livello di esperienza (per gli arbitri).

Esempio:

Profilo di Giocatore

Nome: Maria Bianchi

Punteggio: 1950

Tornei recenti: Roma Open 2023, Milano Challenger 2024

3.2 Classifiche e Statistiche Aggregate

La piattaforma offre strumenti per visualizzare dati aggregati, utili per analisi e confronti:

- **Classifica generale:** Top 10 giocatori in base al punteggio.
 - **Classifica per tipo di campo:** Top 10 giocatori su erba, terra battuta, o cemento.
 - **Torneo con più spettatori:** Una sezione dedicata ai tornei più seguiti, utile per sponsor e organizzatori.
-

4. Gestione Amministrativa

4.1 Assegnazione Arbitri

Gli amministratori possono selezionare i migliori arbitri disponibili per un torneo.

- **Criteri di selezione:**
 - Livello di esperienza.

- Disponibilità nel periodo del torneo.
- **Processo:**
 1. L'amministratore consulta la lista degli arbitri disponibili.
 2. Assegna gli arbitri alle partite in base alle loro specializzazioni.

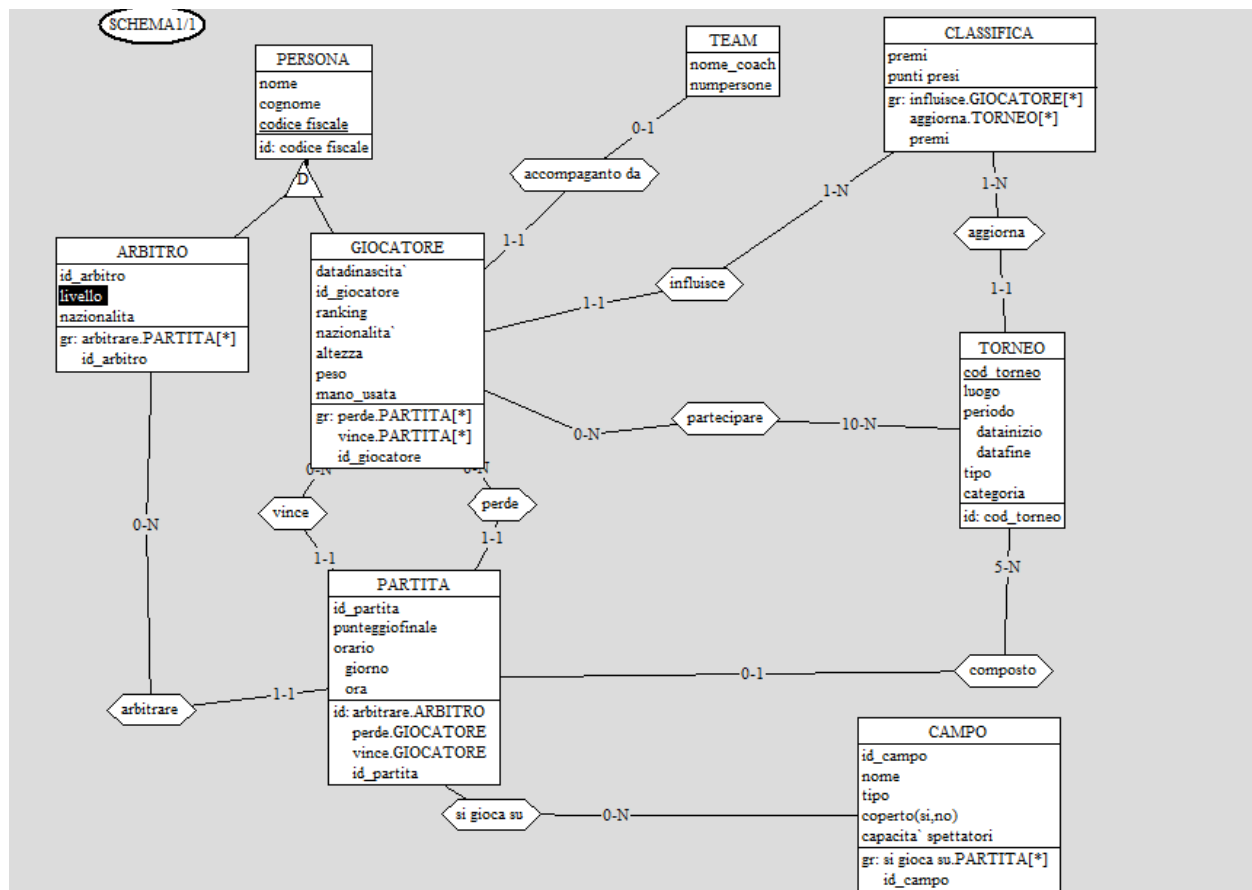
4.2 Sanzioni e Blocchi

Per mantenere un ambiente competitivo corretto, la piattaforma prevede meccanismi di sanzione:

- **Blocco dei team:** Dopo 3 interventi disciplinari, il profilo di un team viene bloccato temporaneamente.
- **Sblocco:** Solo l'amministratore può ripristinare l'accesso a un team bloccato.

3. PROGETTAZIONE CONCETTUALE

3.1 Schema Scheletro



Prima versione con problemi critici:

- Entità **ARBITRO** non collegata a **TORNEO**.
- Nessun tracciamento storico dei punteggi.

3.2 Raffinamenti

Il progetto iniziale presentava uno schema concettuale essenziale, ma durante l'analisi sono emerse diverse criticità che hanno richiesto l'introduzione di nuove entità per garantire una gestione completa e accurata dei tornei. Di seguito, un'analisi dettagliata delle entità mancanti nella prima versione e del loro ruolo nello schema finale.

1. Entità "Fase" (Mancante nello Schema Iniziale)

Problema Rilevato

Nella prima versione dello schema, il torneo era rappresentato come un'entità monolitica, senza suddivisione in fasi (es. qualificazioni, ottavi, semifinali). Questo approccio limitava la capacità di:

- **Pianificare partite in modo strutturato.**
- **Tracciare l'avanzamento dei giocatori** nel torneo.
- **Assegnare risorse** (arbitri, campi) in base alla fase.

Soluzione Adottata

È stata introdotta l'entità **FASE** con i seguenti attributi:

- **id_fase** : Identificativo univoco.
- **nome** (es. "Qualificazioni", "Finale").
- **data_inizio** e **data_fine** .
- **id_torneo** (chiave esterna verso **TORNEO**).

Relazioni Aggiunte:

- **TORNEO** → **FASE** **(1-N)**: Ogni torneo può avere più fasi.
- **FASE** → **PARTITA** **(1-N)**: Ogni fase contiene più partite.

Vantaggi:

- Maggiore flessibilità nella schedulazione delle partite.
- Possibilità di assegnare arbitri specifici per fasi critiche (es. finali).

2. Entità "Punteggio Storico" (Mancante nello Schema Iniziale)

Problema Rilevato

La prima versione registrava solo il punteggio corrente di giocatori e arbitri, senza tracciare l'evoluzione storica. Questo impediva:

- **Analisi delle prestazioni** nel tempo.
- **Calcolo di trend** (es. miglioramento su superfici specifiche).

Soluzione Adottata

Creazione dell'entità **PUNTEGGIO_STORICO** :

- **id_giocatore** / **id_arbitro** : Chiavi esterne.
- **data** : Data del punteggio.
- **valore** : Punteggio ELO o rating arbitrale.

- `tipo_superficie` (opzionale): Per classifiche specifiche.

Vantaggi:

- Supporto a reportistica avanzata per allenatori e team.
 - Possibilità di implementare algoritmi di predizione (es. crescita ELO).
-

3. Entità "Disponibilità Arbitro" (Mancante nello Schema Iniziale)

Problema Rilevato

Nella bozza iniziale, gli arbitri erano assegnati direttamente ai tornei senza gestire:

- **Candidature volontarie.**
- **Stati di approvazione** (pending/approvato/rifiutato).
- **Conflitti di scheduling** (es. arbitro impegnato in due tornei sovrapposti).

Soluzione Adottata

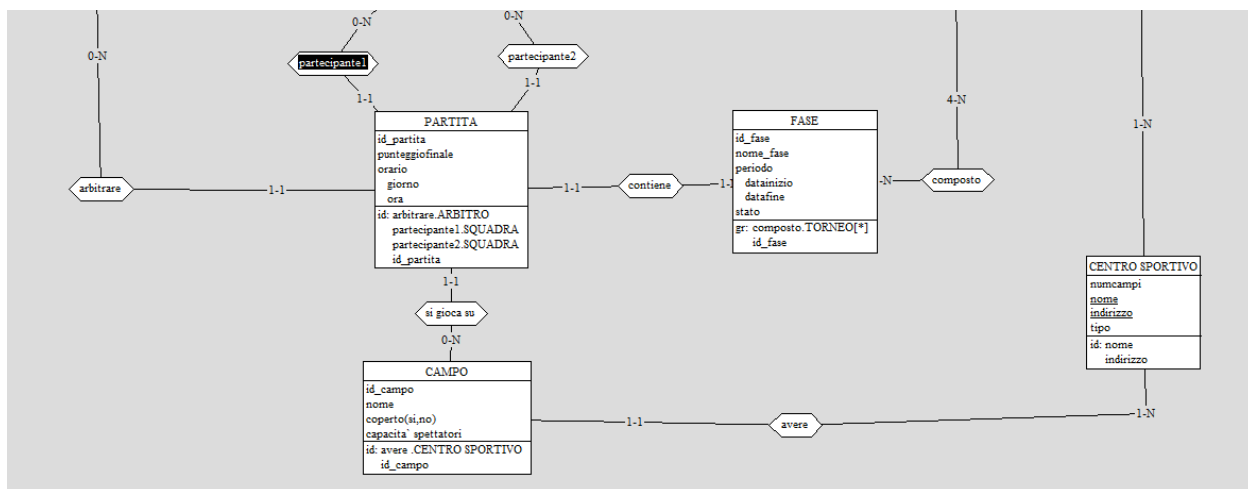
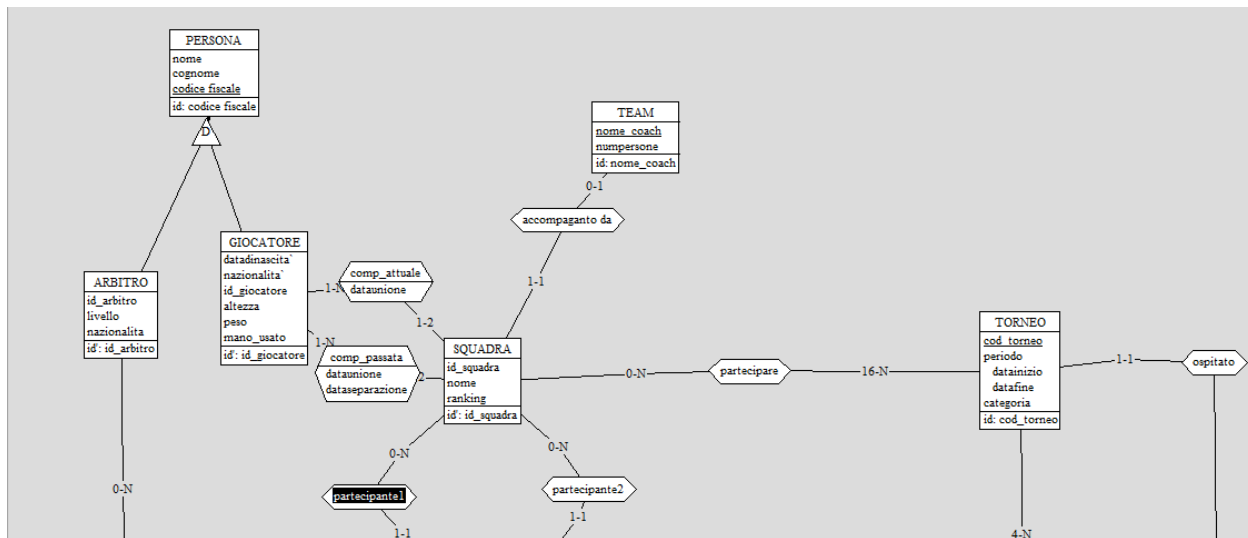
Introduzione dell'entità `DISPONIBILITA_ARBITRO` :

- `id_arbitro` e `id_torneo` : Chiavi esterne composite.
- `stato` : Enum (`pending` , `approvato` , `rifiutato`).
- `data_richiesta` : Tracciamento audit.

Flusso di Lavoro:

1. L'arbitro si candida a un torneo → record con stato `pending` .
2. L'amministratore approva → stato diventa `approvato` .
3. Il sistema verifica i conflitti di日程 prima dell'assegnazione.

3.3 Schema E-R Finale



Nuove entità introdotte:

- **DISPONIBILITA_ARBITRO**: Stato (pending/approvato/rifiutato).
- **PUNTEGGIO_STORICO**: Data, punteggio, tipo superficie.

Scelta controversa:

- **Normalizzazione di tipo_campo**:
 - *Opzione 1*: Attributo in **TORNEO** (semplice ma poco flessibile).
 - *Opzione 2*: Tabella separata (più complessa ma estensibile).
 - **Decisione**: Opzione 1, con **ENUM** per controllare i valori

4. PROGETTAZIONE LOGICA

4.1 Stima del Volume

Concetto	Costrutto	Volume (Record/Anno)
Team	E (Entità)	500
Giocatore	E (Entità)	1,500
Torneo	E (Entità)	100
Partecipazione	R (Relazione)	2,000
Arbitro	E (Entità)	200
Disponibilità Arbitro	R (Relazione)	1,000
Fase	E (Entità)	300
Partita	E (Entità)	5,000
Punteggio Storico	E (Entità)	10,000
Classifica	MV (Vista Materializzata)	30 (aggiornamenti)

4.2 Descrizione delle operazioni principali e stima della loro frequenza

Operazione	Frequenza
Registrazione nuovo Team	50/giorno
Registrazione nuovo Giocatore	100/giorno
Creazione nuovo Torneo	5/giorno
Iscrizione Team a Torneo	200/giorno
Candidatura Arbitro a Torneo	150/giorno
Approvazione Candidatura Arbitro	100/giorno
Inserimento Risultato Partita	300/giorno
Aggiornamento Punteggio Giocatore	500/giorno
Generazione Classifica	10/ora
Visualizzazione Profilo Giocatore	1.000/giorno
Blocco Team (per sanzioni)	5/giorno
Sblocco Team	2/giorno
Prenotazione Campo di Gioco	50/giorno
Invio Notifiche Sistema	2.000/giorno

4.3 Schemi di Navigazione e Tabelle degli Accessi

Questa sezione descrive i percorsi di navigazione principali e stima gli accessi alle tabelle per le operazioni più frequenti.

- (Peso accessi: Lettura = 1, Scrittura = 2)*

1. Operazioni Principali e Schemi di Navigazione

Operazione	Percorso di Navigazione	Tabelle Coinvolte
Registrazione Team	Team → Verifica Email → Inserimento DB	TEAM (Scrittura)
Iscrizione Torneo	Team → Seleziona Torneo → Verifica Requisiti Giocatori → Conferma	TEAM , GIOCATORE , TORNEO , PARTECIPAZIONE (Lettura + Scrittura)
Assegnazione Arbitro	Admin → Visualizza Candidature → Verifica Disponibilità → Approva	ARBITRO , DISPONIBILITA_ARBITRO , TORNEO (Lettura + Scrittura)
Inserimento Risultato	Arbitro → Seleziona Partita → Inserisci Punteggio → Conferma	PARTITA , GIOCATORE , PUNTEGGIO_STORICO (Lettura + Scrittura)
Generazione Classifica	Sistema → Calcola Punteggi → Aggiorna Vista	GIOCATORE , PUNTEGGIO_STORICO , CLASSIFICA (Lettura + Scrittura)

2. Tabella degli Accessi

Operazione	Tabelle	Tipo Accesso	Accessi	Totale (Pesato)
Registrazione Team	TEAM	Scrittura	1	2
Iscrizione Torneo	TEAM	Lettura	1	1
	GIOCATORE	Lettura	2	2
	TORNEO	Lettura	1	1
	PARTECIPAZIONE	Scrittura	1	2
Assegnazione Arbitro	ARBITRO	Lettura	1	1
	DISPONIBILITA_ARBITRO	Lettura	3	3

	TORNEO	Lettura	1	1
	DISPONIBILITA_ARBITRO	Scrittura	1	2
Inserimento Risultato	PARTITA	Scrittura	1	2
	GIOCATORE	Lettura	2	2
	PUNTEGGIO_STORICO	Scrittura	2	4
Generazione Classifica	GIOCATORE	Lettura	10	10
	PUNTEGGIO_STORICO	Lettura	50	50
	CLASSIFICA	Scrittura	1	2

3. Dettaglio Accessi per Operazione

3.1 Iscrizione Torneo

1. Lettura TEAM → Verifica esistenza team (1 accesso).
2. Lettura GIOCATORE → Controllo requisiti ELO (2 accessi, 1 per giocatore).
3. Lettura TORNEO → Verifica date e livello (1 accesso).
4. Scrittura PARTECIPAZIONE → Conferma iscrizione (1 accesso).

Totale pesato: 1 (TEAM) + 2 (GIOCATORE) + 1 (TORNEO) + 2 (PARTECIPAZIONE) = **6**.

3.2 Inserimento Risultato Partita

1. Scrittura PARTITA → Salva risultato (2 accessi).
2. Lettura GIOCATORE → Recupera ELO attuale (2 accessi).
3. Scrittura PUNTEGGIO_STORICO → Aggiorna punteggi (4 accessi, 2 per giocatore).

Totale pesato: 2 (PARTITA) + 2 (GIOCATORE) + 4 (PUNTEGGIO_STORICO) = **8**.

4. Stima Accessi Totali Giornalieri

Operazione	Frequenza	Accessi Totali (Pesati)
Iscrizione Torneo	200/giorno	$200 \times 6 = 1.200$
Inserimento Risultato	300/giorno	$300 \times 8 = 2.400$
Generazione Classifica	10/ora (240/giorno)	$240 \times 62 = 14.880$
Totale		~18.500 accessi/giorno

4.4 Schema relazionale finale

arbitro

(idArbitro, nome, nazionalità, livello)

campo

(idcampo, coperto, capienza, nomecentrospor, indirizzocentospor)

centro_sportiv

(numcampi, nome, indirizzo, tipo)

composto

(idFase: fase, codTorneo: torneo)

comp_attuale

(idgiocatore: giocatore, nomecoach: team, dataUnione)

comp_passata

(idgiocatore: giocatore, nomecoach: team, dataUnione, dataSeparazione)

fase

(idFase, nomeFase, dataInizio, dataFine, stato, codTorneo: torneo)

giocatore

(idgiocatore, nome, dataNascita, nazionalità, altezza, peso, mano_usata)

partecipazione

(idsquadra: squadra, codTorneo: torneo)

partita

(idpartita, idsquadrap: squadra, idsquadras: squadra, idrase: fase, idarbitro: arbitro, codTorneo: torneo, punteggiofinale, ora, giorno)

squadra

(idsquadra, nomecoach: team, nomesquadra, ranking)

team

(nomecoach, numpersone)

torneo

(codTorneo, nomeTorneo, dataInizio, dataFine, categoria)

4.5 Query SQL

- - 1. TABELLA ARBITRO: Registra gli arbitri con i loro dati anagrafici e livello
CREATE TABLE arbitro (
idArbitro INT(11) NOT NULL,
nome TEXT NOT NULL,
nazionalita TEXT NOT NULL,
livello TEXT NOT NULL,
PRIMARY KEY (idArbitro)
ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
- - 2. TABELLA CENTRO SPORTIVO: Contiene informazioni sulle strutture sportive
CREATE TABLE centro_sportiv (
numcampi INT(11) NOT NULL,
nome TEXT NOT NULL,
indirizzo TEXT NOT NULL,
tipo TEXT NOT NULL)
ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
- - 3. TABELLA CAMPO: Dettagli dei singoli campi da gioco
CREATE TABLE campo (
idcampo INT(11) NOT NULL,
coperto TINYINT(1) NOT NULL,
capienza INT(11) NOT NULL,
nomecentrospor TEXT NOT NULL,
indirizzocentospo TEXT NOT NULL,
PRIMARY KEY (idcampo))
ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
- - 4. TABELLA TORNEO: Gestisce i tornei organizzati
CREATE TABLE torneo (
codTorneo INT(11) NOT NULL,
nomeTorneo TEXT NOT NULL,
dataInizio DATE NOT NULL,
dataFine DATE NOT NULL,
categoria TEXT NOT NULL,
PRIMARY KEY (codTorneo))
ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
- - 5. TABELLA FASE: Le diverse fasi di un torneo (gironi, eliminatorie, ecc.)
CREATE TABLE fase (
idFase INT(11) NOT NULL,


```

nomeFase TEXT NOT NULL,
dataInizio DATE NOT NULL,
dataFine DATE NOT NULL,
stato TEXT NOT NULL,
codTorneo INT(11) NOT NULL,
PRIMARY KEY (idFase),
FOREIGN KEY (codTorneo) REFERENCES torneo(codTorneo))
ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

```

- - 6. TABELLA TEAM: Registra gli staff tecnici (allenatori e preparatori)
CREATE TABLE team (
nomecoach TEXT NOT NULL,
numpersone INT(11) NOT NULL,
PRIMARY KEY (nomecoach))
ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
- - 7. TABELLA SQUADRA: Contiene tutte le squadre partecipanti
CREATE TABLE squadra (
idsquadra INT(11) NOT NULL,
nomecoach TEXT NOT NULL,
nomesquadra TEXT NOT NULL,
ranking INT(11) NOT NULL,
PRIMARY KEY (idsquadra),
FOREIGN KEY (nomecoach) REFERENCES team(nomecoach))
ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
- - 8. TABELLA GIOCATORE: Anagrafica completa dei giocatori
CREATE TABLE giocatore (
idgiocatore TEXT NOT NULL,
nome TEXT NOT NULL,
datanascita DATE NOT NULL,
nazionalita TEXT NOT NULL,
altezza INT(11) NOT NULL,
peso INT(11) NOT NULL,
mano_usata TEXT NOT NULL,
PRIMARY KEY (idgiocatore))
ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
- - 9. TABELLA COMPOSTO: Associazione tra fasi e tornei (relazione molti-a-molti)
CREATE TABLE composto (
idFase INT(11) NOT NULL,
codTorneo INT(11) NOT NULL,

PRIMARY KEY (idFase, codTorneo),
 FOREIGN KEY (idFase) REFERENCES fase(idFase),
 FOREIGN KEY (codTorneo) REFERENCES torneo(codTorneo))
 ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

- - 10. TABELLA COMP_ATTUALE: Relazione corrente giocatore-squadra
 CREATE TABLE comp_attuale (
 idgiocatore TEXT NOT NULL,
 nomecoach TEXT NOT NULL,
 dataUnione DATE NOT NULL,
 PRIMARY KEY (idgiocatore),
 FOREIGN KEY (idgiocatore) REFERENCES giocatore(idgiocatore),
 FOREIGN KEY (nomecoach) REFERENCES team(nomecoach))
 ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
- - 11. TABELLA COMP_PASSATA: Storico delle squadre dei giocatori
 CREATE TABLE comp_passata (
 idgiocatore TEXT NOT NULL,
 nomecoach TEXT NOT NULL,
 dataUnione DATE NOT NULL,
 dataSeparazione DATE NOT NULL,
 FOREIGN KEY (idgiocatore) REFERENCES giocatore(idgiocatore),
 FOREIGN KEY (nomecoach) REFERENCES team(nomecoach))
 ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
- - 12. TABELLA PARTECIPAZIONE: Squadre iscritte a ciascun torneo
 CREATE TABLE partecipazione (
 idsquadra INT(11) NOT NULL,
 codTorneo INT(11) NOT NULL,
 PRIMARY KEY (idsquadra, codTorneo),
 FOREIGN KEY (idsquadra) REFERENCES squadra(idsquadra),
 FOREIGN KEY (codTorneo) REFERENCES torneo(codTorneo))
 ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
- - 13. TABELLA PARTITA: Registra tutti gli incontri disputati
 CREATE TABLE partita (
 idpartita INT(11) NOT NULL,
 idsquadrap INT(11) NOT NULL,
 idsquadras INT(11) NOT NULL,
 idrase INT(11) NOT NULL,
 idarbitro INT(11) NOT NULL,
 codTorneo INT(11) NOT NULL,

```
punteggiordine INT(11) NOT NULL,  
ora INT(11) NOT NULL,  
giorno TEXT NOT NULL,  
PRIMARY KEY (idpartita),  
FOREIGN KEY (idsquadrap) REFERENCES squadra(idsquadra),  
FOREIGN KEY (idsquadras) REFERENCES squadra(idsquadra),  
FOREIGN KEY (idrase) REFERENCES fase(idFase),  
FOREIGN KEY (idarbitro) REFERENCES arbitro(idArbitro),  
FOREIGN KEY (codTorneo) REFERENCES torneo(codTorneo))  
ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

5. Progettazione dell'Applicazione:

L'applicazione sviluppata è un sistema di gestione per tornei sportivi (es. tennis) che permette:

- **Organizzazione tornei** (creazione, fasi, registrazione team/giocatori)
- **Gestione partite** (pianificazione, risultati, arbitri)
- **Classifiche** (punteggi giocatori e arbitri)
- **Monitoraggio tornei** (stato, storico, statistiche)

L'API è realizzata con:

- **Backend:** FastAPI (Python) + SQLAlchemy (database relazionale)
- **Frontend:** React/Vite (interfaccia utente)
- **Database:** SQLite (per sviluppo)

5.1 Architettura dell'Applicazione

L'applicazione segue un'architettura **client-server** con:

Backend (FastAPI)

- **API RESTful** con endpoint per:
 - Autenticazione (`/api/login` , `/api/teams/register`)
 - Tornei (`/api/tournaments` , `/api/tournaments/{id}/matches`)
 - Partite (`/api/matches`)
 - Classifiche (`/api/players/rankings` , `/api/referees/rankings`)
- **Middleware CORS** per consentire richieste dal frontend

- **Modelli Pydantic** per validazione dati
- **SQLAlchemy ORM** per gestione database

Frontend (React/Vite)

- Interfacce per:
 - **Login/Registrazione**
 - **Dashboard Tornei**
 - **Gestione Partite**
 - **Visualizzazione Classifiche**
- Chiamate API tramite `fetch` o `axios`

Database

- Tabelle principali:
 - `Tournament` (tornei)
 - `Team` (squadre)
 - `Player` (giocatori)
 - `Match` (partite)
 - `Referee` (arbitri)
 - `Phase` (fasi del torneo)

5.2 Diagramma dell'Architettura:

