

**Descrizione:** Un'azienda di albi e almanacchi sportivi vuole gestire e pubblicare con un'applicazione web la storia di più anni di un campionato di calcio con le squadre, i giocatori e le classifiche di un campionato sportivo.

**Link all'Elaborato su GitHub:**

<https://github.com/strambinienea/AlmanaccoCalcistico>

## PARTE DI INFORMATICA

### 1.0 Analisi

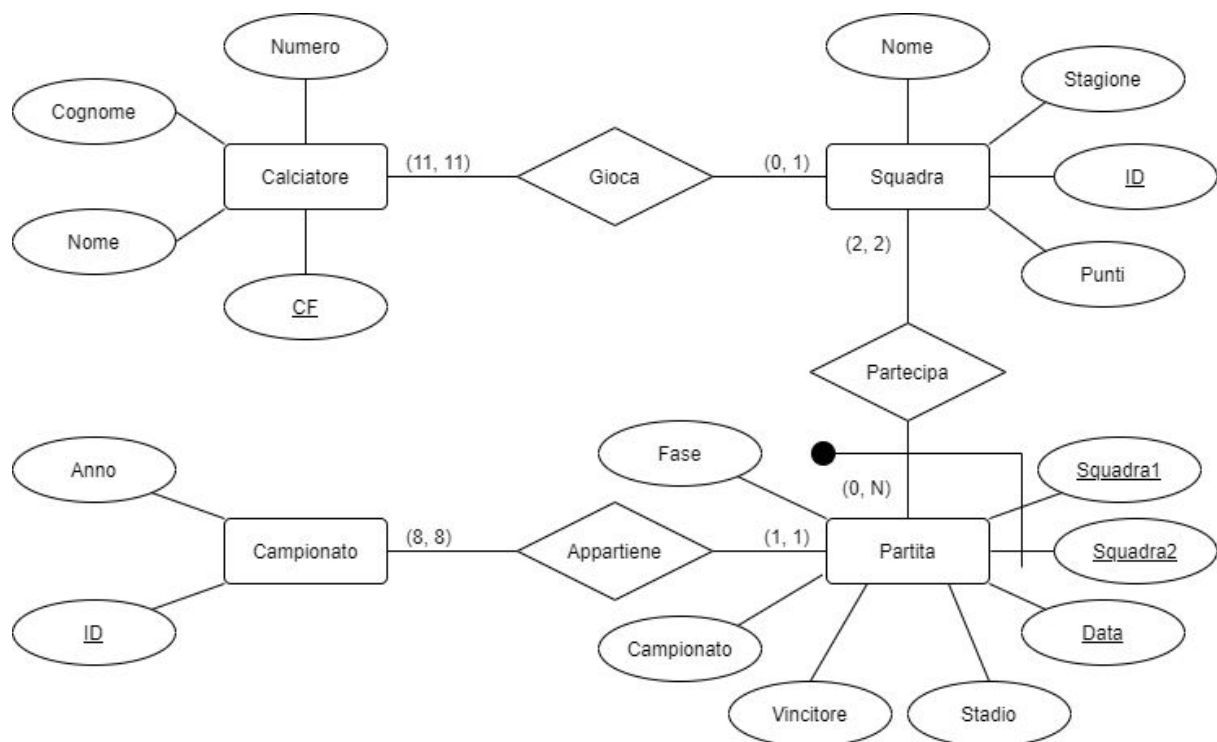
In base alla descrizione del progetto la soluzione scelta è una semplice **applicazione web**, sviluppata in **Python** tramite l'utilizzo della libreria **Flask**. Seguendo la consegna il sito sarà accessibile da **qualsiasi dispositivo**, senza la necessità di un login.

Al fine di produrre una **soluzione snella** sono state apportate varie **semplificazioni**, tra cui:

- il numero di squadre in un torneo è di 8
- il sito è uno storico, ovvero si inseriscono tornei già svolti
- una squadra è composta da solo 11 giocatori, senza riserve

## 2.0 Diagramma E/R e Schema Logico

Il diagramma E/R del database è il seguente:



Lo schema logico del database è il seguente:

- **Calciatore** (**CF**, Nome, Cognome, Numero)
- **Squadra** (**ID**, Nome, Stagione, Punti)
- **CalciatoreSquadra** (**CF**, **IDSquadra**)
- **Campionato** (**ID**, Stagione)
- **Partita** (**IDSquadra1**, **IDSquadra2**, **Data**, Stadio, Vincitore, Campionato, Fase)

### 1.3 Implementazione del Database

Come tecnologie per lo sviluppo del database ho fatto affidamento a **PostgreSQL** e **PGAdmin**, visto le mie passate esperienze con entrambe. Per la comunicazione tra database e sito ho invece utilizzato una libreria di Python, chiamata **psycopg2**.

Di seguito un estratto dal file **.ddl** per la creazione del database:

```
CREATE TYPE fase AS ENUM ('Quarti di Finale', 'Semifinale', 'Finale');
CREATE TABLE public.Partita (
    "IDSquadra1" INT NOT NULL,
    "IDSquadra2" INT NOT NULL,
    "Data" DATE NOT NULL,
    "Stadio" VARCHAR(50) NOT NULL,
    "Vincitore" INT NOT NULL,
    "Campionato" INT NOT NULL,
    "Fase" fase NOT NULL,

    PRIMARY KEY ("IDSquadra1", "IDSquadra2", "Data"),

    FOREIGN KEY ("IDSquadra1") REFERENCES public.Squadra ("ID")
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY ("IDSquadra2") REFERENCES public.Squadra ("ID")
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY ("Vincitore") REFERENCES public.Squadra ("ID")
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY ("Campionato") REFERENCES public.Campionato ("ID")
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

Per l'inserimento dei dati ho invece creato un breve script in **Python** che mi permettesse di inserire velocemente i dati, generandoli **casualmente** pur mantenendo un certo livello di **veridicità**; per fare ciò ho sfruttato una libreria Python chiamata **Faker**, che permette la creazione di nomi e cognomi reali in modo casuale. Di seguito un breve estratto del codice:

```
# INIZIO INSERT CALCIATORI

sql = 'INSERT INTO public.Calciatore ("CF", "Nome", "Cognome", "Numero")
VALUES (%s, %s, %s, %s)'
calciatori = []

n = 0
while n < 100:
    nomecognome = faker.name_male().split(' ')
    if len(nomecognome) == 2:
        calciatore = {
            "CF": fake_cf(),
            "Nome": nomecognome[0],
            "Cognome": nomecognome[1]
        }
        cursor.execute(sql, (calciatore["CF"], calciatore["Nome"],
                             calciatore["Cognome"],
                             str(random.randint(10, 99))))
        n += 1

database.commit()

# FINE INSERT CALCIATORI
```

## 1.4 Query Significantive

Come query significative ho scelto le due query su cui si basa maggiormente il sito, la prima consente di mostrare la classifica di un dato campionato.

```
-- QUERY N. 1 --  
-- MOSTRA LA CLASSIFICA DI UN DETERMINATO CAMPIONATO  
PREPARE getClassifica (int) AS  
SELECT  
    S."Nome",  
    S."Stagione",  
    S."Punti"  
FROM  
    public.Squadra AS S,  
    public.Campionato AS C  
WHERE  
    C."Stagione" = S."Stagione"  
    AND C."ID" = $1  
ORDER BY  
    S."Punti" DESC;  
  
EXECUTE getClassifica (1);
```

La seconda ritorna invece tutte le partite giocate in un campionato, associando le varie squadre tra di loro.

```
-- QUERY N. 2 --  
-- MOSTRA TUTTE LE PARTITE DI UN DETERMINATO CAMPIONATO  
--  
PREPARE getPartite (int) AS  
SELECT  
    S1."Nome" AS Squadra1,  
    S2."Nome" AS Squadra2,  
    P."Stadio",  
    P."Fase",  
    S3."Nome" AS SquadraVincente,  
    P."Data"  
FROM  
    Squadra AS S1,  
    Squadra AS S2,  
    Squadra AS S3,  
    Partita AS P  
WHERE  
    P."IDSquadra1" = S1."ID"  
    AND P."IDSquadra2" = S2."ID"  
    AND P."Vincitore" = S3."ID"  
    AND P."Campionato" = $1  
ORDER BY  
    P."Data" ASC;  
  
EXECUTE getPartite (1);
```

## 1.5 Applicazione Web

L'applicazione web è sviluppata in 3 semplici pagine, che permettono di vedere i **campionati**, i vari **gironi**, la **classifica** ed un **dettaglio di ogni partita**. Per semplicità riporto la cattura delle due pagine principali.

### Campionato Anno: 2000/2001

#### Fasi

Squadra Ospitante	Squadra Ospite	Fase	Vincitore
Milan	Lazio	Quarti di Finale	Milan
Inter	Roma	Quarti di Finale	Inter
Torino	Hellas Verona	Quarti di Finale	Hellas Verona
Juventus	Chievo	Quarti di Finale	Juventus
Hellas Verona	Inter	Semifinale	Milan
Juventus	Milan	Semifinale	Juventus
Hellas Verona	Juventus	Finale	Hellas Verona

*Rappresenta le fasi di un campionato*

Classifica

Squadra	Punteggio
Hellas Verona	9
Juventus	6
Inter	3
Milan	3
Torino	0
Chievo	0
Roma	0
Lazio	0

Rappresenta la classifica finale di un campionato

Torino			Hellas Verona		
Nome	Cognome	Numero	Nome	Cognome	Numero
Ferdinando	Solimena	31	Gianluigi	Proietti	41
Enzio	Sommaruga	76	Alfredo	Ioppi	53
Hugo	Canevascini	16	Gioacchino	Barese	72
Michelangelo	Duodo	47	Orazio	Moneta-Zaguri	75
Azeglio	Petrocelli	45	Tiziano	Correr-Taliercio	75
Cipriano	Fuseli	18	Uberto	Duse-Micca	25
Gianpaolo	Castiglione	56	Temistocle	Tutino	86
Umberto	Raimondi	35	Lodovico	Duodo	49
Ennio	Pulci	75	Augusto	Zanzi	54
Giosuè	Udinesi	30	Ugolino	Barbarigo	66
Luciano	Bertoni	81	Elladio	Iannuzzi	16

Data	Vincitore	Fase	Stadio
2000-02-05	Hellas Verona	Quarti di Finale	Juventus Stadium

Rappresenta in dettaglio una partita, con i giocatori delle squadre



# SISTEMI E RETI

**Richiesta:** L'azienda vuole offrire ai propri clienti la possibilità di acquistare su Internet riviste e libri sportivi nonché gadget delle varie squadre. Esporre una possibile soluzione tecnologica, giustificandola, che consenta all'azienda di offrire ai clienti la possibilità di effettuare gli acquisti online in sicurezza. Illustrare in particolare gli aspetti di sicurezza delle comunicazioni a base web.

## Sviluppo della Soluzione

Per fornire un servizio di E-Commerce l'azienda deve ovviamente possedere un **server** ed un'**infrastruttura di rete adeguata**. Ciò significa averne uno di proprietà o affittarne uno da un data center. La sicurezza nelle comunicazione web è fondamentale, soprattutto se vista nel contesto dell'E-Commerce, dove vengono scambiati molti **dati sensibili** tra il Client ed il Server. Per assicurare la comunicazione si devono usare tecnologie come **SSL/TLS**.

**SSL e TLS** sono protocolli utilizzati per garantire una trasmissione di dati sicura nel World Wide Web, garantendo **integrità, segretezza ed autenticità**. Questo, in combinazione con l'HTTP, consente di ottenere il nuovo protocollo HTTPS, il quale invia le informazioni sotto forma di pacchetti criptati, prevenendo eventuali manomissioni da parte di terzi. Inoltre, visto che la maggior parte dei browser moderni supportano questi protocolli, permettono un'implementazione rapida ed efficace allo stesso tempo, senza doversi preoccupare di compatibilità dal lato Client.

In conclusione visto il continuo scambio di **dati personali**, quali generalità o informazioni su conti bancari, si dovrà sviluppare un'applicazione **compatibile con il sopracitato protocollo**, e una struttura di rete adeguata, così da consentire un **trasferimento sicuro** dei dati dei clienti.