



UNIVERSITÀ DEGLI STUDI DI
MILANO

PROGETTO DI BASI DI DATI

DOCUMENTAZIONE TECNICA

Marco Cesana
909159

Elia Togni
909923

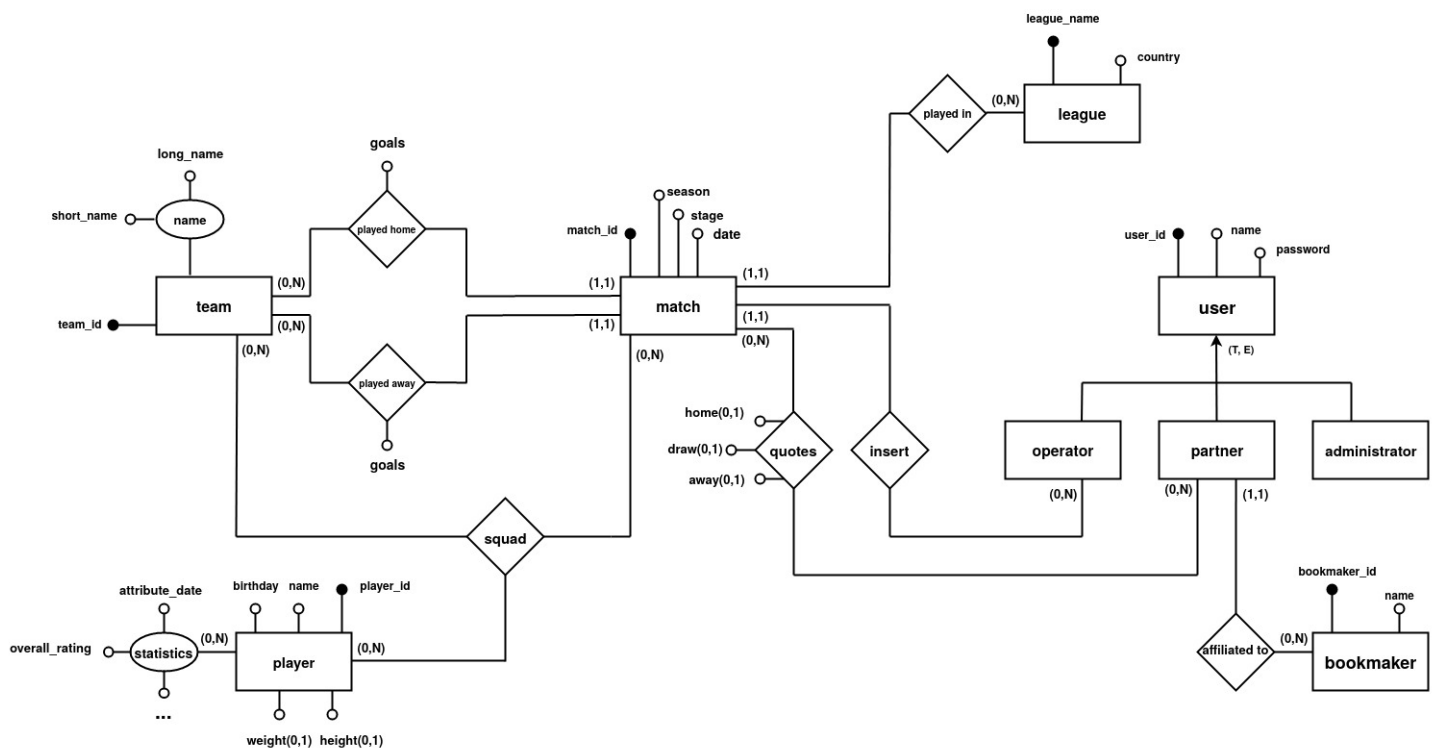
Indice

1. Tecnologie usate.	3
2. Schema ER	3
3. Schema logico	4
4. Vincoli	6
5. Scelte di progettazione del database	6
5.1 Importazione dati	6
5.2 Player	6
5.3 Statistics	6
5.4 Match	7
5.5 Squad	7
5.6 Quotes	8
5.7 Classifiche	8
5.8 Utente	9
5.9 Controllo dei permessi	9
5.10 Funzioni	9
5.11 Trigger	10
6. Scelte di progettazione dell'applicazione web . .	11
6.1 Autenticazione degli utenti	11
6.2 Form di registrazione	12
6.3 Form di inserimento dati	12

1. Tecnologie usate

- Lappstack-7.3.10-0
- Apache 2.4.41
- PostgreSQL 11.5
- PHP 7.3.11
- CSS
- JAVASCRIPT
- HTML

2. Schema ER



3. Schema logico

team (team_id, long_name, short_name)

player (player_id, name, birthday, height*, weight*)

statistics (player, attribute_date, overall_rating, potential, preferred_foot, attacking_work_rate*, defensive_work_rate*, crossing, finishing, heading_accuracy, short_passing, volleys*, dribbling, curve*, free_kick_accuracy, long_passing, ball_control, acceleration, sprint_speed, agility*, reactions, balance*, shot_power, jumping*, stamina, strength, long_shots, aggression, interceptions, positioning, vision*, penalties, marking, standing_tackle, sliding_tackle*, gk_diving, gk_handling, gk_kicking, gk_positioning, gk_reflexes)

squad (match, player, team)

league (league_name, country)

match (match_id, season, stage, date, home_team, away_team, league, operator, home_goals, away_goals)

quotes (match, partner, home, draw, away)

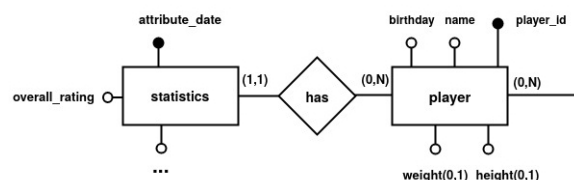
administrator (user_id, name, password)

partner (user_id, name, password, bookmaker)

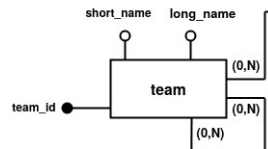
operator (user_id, name, password)

bookmaker (bookmaker_id, name)

Nel diagramma ER, *statistics* era un attributo multivalore di *player*. Abbiamo deciso di ristrutturarlo come una nuova entità con chiave esterna *player_id*.



Nell'entità *team* abbiamo un altro attributo multivalore, *name*, composto da *long_name* e *short_name*. Lo ristrutturiamo come due attributi separati.

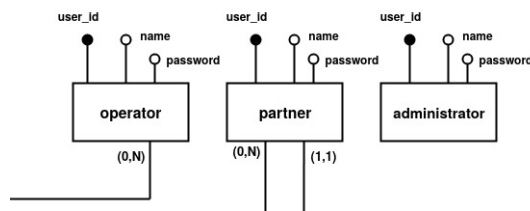


Tra l'entità *user* e le entità *administrator*, *operator* e *partner* esiste una gerarchia di specializzazione. Per ristrutturare la gerarchia ci siamo trovati di fronte a tre differenti scelte implementative.

Tenere sia l'entità padre che le entità figlie ci è sembrata fin da subito una scelta non adatta.

Mantenere solamente l'entità padre avrebbe comportato l'utilizzare un attributo *type* per definire quale sottoutente stiamo trattando. Avrebbe inoltre aggiunto l'attributo *bookmaker*, NULL nel caso di *administrator* e *operator*. Questo attributo è direttamente dipendente da *tipo* e non da *user_id*, quindi renderebbe lo schema ER non in terza forma normale.

Per questo ci è sembrata una soluzione "poco elegante". Abbiamo quindi optato per separare le tre entità figlie senza mantenere *user*, aggiungendo ad ognuna di loro gli attributi dell'entità padre.



4. Vincoli

bookmaker: **UNIQUE**(name)
administrator: **UNIQUE**(name)
partner: **UNIQUE**(name)
operator: **UNIQUE**(name)
match: **UNIQUE**(league, season, stage, home_team, date)
 UNIQUE(league, season, stage, away_team, date)
 CHECK(home_team <> away_team)
 CHECK(home_goals >= 0)
 CHECK(away_goals >= 0)
 CHECK(stage > 0)

5 Scelte di progettazione del database

5.1 Importazione dati

Le nostre scelte riguardanti l'import dei dati sono state quasi totalmente condizionate dalla struttura dei CSV. In alcuni casi, degli attributi con dominio **INTEGER** contenevano **VARCHAR** o erano **NULL**. In altri casi, quando ci aspettavamo che degli attributi contenessero stringhe predefinite, abbiamo trovato stringhe prive di significato. In ognuno di questi casi abbiamo deciso di ignorare il contenuto delle tuple e di inserire **NULL** al loro posto.

5.2 Player

Tutti gli attributi presenti in *player* sono contenuti in match.csv. Ogni giocatore possiede già un id associato alla propria persona quindi ci è sembrata la scelta più corretta come primary key. Nella tabella *player* gli attributi *height* e *weight* non hanno vincolo di **NOT NULL** poiché dal CSV si può notare che per alcuni giocatori non sono presenti tali informazioni.

5.3 Statistics

La stessa situazione della tabella *player* si presenta in *statistics*: dei numerosi attributi riferiti ad un singolo giocatore (40 attributi), alcuni possono essere **NULL**; Nelle specifiche viene indicato di basarsi

esclusivamente sui dati dei CSV, quindi solamente gli attributi che non contenevano valori in nessuna delle righe del file sono stati privati di vincolo NULL. Per quanto riguarda gli attributi *attacking_work_rate* e *defensive_work_rate* la nostra scelta è stata quella di creare un tipo *rating* contenente solo le stringhe consentite nei due attributi (*high*, *medium* e *low*). Allo stesso modo abbiamo deciso di creare un tipo *pfoot* per l'attributo *preferred_foot* (*right*, *left*). Se il tipo di dati inserito non rispetta il dominio, viene settato a NULL. La chiave primaria è formata dalla coppia (*player_id*, *attribute_date*) in quanto su un giocatore viene effettuata al più una valutazione al giorno. Viene così impedito di avere più misurazioni di un giocatore *player* nella medesima data.

5.4 Match

In *match* per evitare che una squadra partecipi a più match nella stessa giornata della stessa season utilizziamo due vincoli UNIQUE tra gli attributi (*league*, *season*, *stage*, *home_team*, *date*) e (*league*, *season*, *stage*, *away_team*, *date*).

Questi campi non sono stati resi chiave primaria, come si potrebbe pensare ad un primo acchito poiché la primary key della tabella *match*, *match_id* è già presente nel CSV corrispettivo e perchè, essendo di piccole dimensioni, è di più facile utilizzo nelle tabelle e nel lato PHP.

5.5 Squad

La tabella *squad* deriva dalla relazione ternaria tra *match*, *team* e *player*. Tiene traccia della formazione di un team in un determinato match. È possibile che ad un match corrispondano meno di undici tuple, cioè il numero di titolari in una partita, poichè nel csv non sono sempre presenti gli id di alcuni dei giocatori in corrispondenza della partita osservata.

In questo punto ci siamo trovati di fronte a due scelte implementative:

- La prima consisteva nel utilizzare come chiave primaria solo la coppia (*player*, *match*) per evitare che un giocatore giocasse un match per entrambe le squadre. Questa scelta però stona con lo schema ER in quanto la tabella *squad* è sostanzialmente una relazione ternaria tra *match*, *player* e *team*.
- La seconda scelta invece comportava l'utilizzo di tutte e tre le chiavi esterne come chiave primaria per poi forzare un vincolo

UNIQUE tra *player* e *match*. La nostra scelta è ricaduta su questa implementazione in quanto più coerente con lo schema ER.

Nell'importazione dei dati questa scelta ci ha permesso di trovare una tupla dove un giocatore giocava per entrambe le squadre in una partita.

5.6 Quotes

Per ogni match, un bookmaker deve avere inserito tre quote oppure nessuna. Per questo motivo effettuiamo un controllo e se troviamo meno di tre quote corrispondenti ad un match per un determinato bookmaker, allora non inseriamo la riga. Accettiamo solamente tre caselle contenenti quote oppure tre caselle contenenti NULL.

5.7 Classifiche

Per la classifica dei campionati utilizziamo una vista materializzata composta dagli attributi (*league*, *season*, *team_id*, *team_long_name*, *points*). Questa vista è ottenuta tramite diverse WITH: la prima conta il numero di vittorie di ogni squadra in casa; la seconda conta invece il numero di vittorie di ogni squadra fuori casa; con la terza e la quarta WITH contiamo il numero totale di pareggi tra le partite giocate in casa e quelle giocate fuori casa. Poi sommiamo il numero di vittorie totali moltiplicato per 3 al numero totale di pareggi ed otteniamo i punti ottenuti in una determinata season da una squadra. Poichè una squadra non ha vinto nessuna partita in trasferta in una determinata stagione, al numero di partite vinte e pareggiate, sommiamo anche zero per farla comparire nella tabella.

Da un form facciamo selezionare la league e la season desiderate e, tramite una query sulla vista, facciamo visualizzare la relativa classifica. Per ogni posizione della classifica viene mostrato il nome della squadra e il punteggio.

Inoltre, è presente un trigger che, ad ogni inserimento o modifica nelle tabelle necessarie a creare la vista (*squad*, *statistics*, *player* e *match*), aggiorna la vista materializzata.

Per la classifica dei bookmaker invece creiamo una vista materializzata sulla base della WITH *tab1*. Nella WITH creiamo un attributo *result* tale che, se il numero di goal della squadra di casa è maggiore del numero di goal della squadra in trasferta, conterrà 'H'.

Se il numero di goal delle due squadre è uguale, conterrà '*D*', altrimenti conterrà '*A*'. Poi *result* sarà affiancato a *match_id*. La vista seleziona quindi i due attributi della *WITH*, il bookmaker associato al match e la quota più bassa riferita al match, scritta nella forma '*H*', '*A*' o '*D*'. La query corrispondente raggruppa le tuple secondo *bookmaker* e conta il numero di *result* quando il suo contenuto è uguale al contenuto di *lower_quote*.

5.8 Utente

Nel database è presente una tabella per ognuno dei tre tipi di utenti: *operator*, *partner* e *administrator*. Un vincolo *UNIQUE(name)* impedisce che due utenti dello stesso tipo condividano il nome. Il campo *password* contiene un hash della password in modo da non salvare in chiaro dati sensibili dell'utente.

5.9 Controllo dei permessi

Il controllo dei permessi viene effettuato tramite una serie di *GRANT*. Il database è strutturato in modo da avere quattro tipi di ruoli:

- *Guest*: corrisponde all'utente non loggato nel sito. Nelle pagine delle classifiche e in alcuni tratti della pagina dei match si è loggati come *guest*. Questo garantisce un maggiore livello di protezione in quanto si tratta del ruolo con meno permessi (solo di login in un altro ruolo).
- *Administrator*: corrisponde all'utente *administrator*. Quando si accede ad una pagina riservata ad un admin, far parte di questo ruolo permette di avere i privilegi corrispondenti.
- *Operator*: corrisponde all'utente *operator*. Quando si accede ad una pagina riservata ad un operatore, far parte di questo ruolo permette di avere i privilegi corrispondenti.
- *Partner*: corrisponde all'utente *partner*. Quando si accede ad una pagina riservata ad un *partner*, far parte di questo ruolo permette di avere i privilegi corrispondenti.

Inoltre l'utilizzo delle sessioni consente di evitare ad un utente malintenzionato di accedere ad un'area a lui non consentita attraverso l'URL di tale pagina. Infatti, se l'utente non possiede i permessi per accedere a quella pagina, viene rimandato a *index.php*.

5.10 Funzioni

La funzione *refresh_bookmaker_mview* viene chiamata dai trigger ad ogni inserimento o modifica delle tabelle che formano la materialized view della classifica dei bookmaker. La funzione refresha semplicemente la vista.

```
CREATE OR REPLACE FUNCTION refresh_bookmaker_mview() RETURNS
trigger AS $BODY$
BEGIN
    REFRESH MATERIALIZED VIEW soccerscheme.bookmaker_leaderboards;
    RETURN NEW;
END $BODY$ LANGUAGE plpgsql;
```

Allo stesso modo la funzione *refresh_leaderboard_view* refresha la vista materializzata riguardante la classifica dei campionati

```
CREATE OR REPLACE FUNCTION refresh_leaderboard_mview() RETURNS
trigger AS $BODY$
BEGIN
    REFRESH MATERIALIZED VIEW soccerscheme.complete_leaderboard;
    RETURN NEW;
END $BODY$ LANGUAGE plpgsql;
```

Utilizziamo inoltre delle funzioni per inserire, cancellare o modificare delle tuple:

- *insert_table*: Sono nove funzioni che consentono di inserire elementi nella corrispettiva tabella solamente al ruolo proprietario della funzione;
- *delete_table_name*: Sono sei funzioni che consentono di eliminare elementi dalla corrispettiva tabella solamente al ruolo proprietario della funzione;
- *update_table_name*: Come i due tipi di funzione precedenti, sono cinque funzioni che consentono l'update di elementi dalla corrispettiva tabella solamente al ruolo proprietario della funzione.

Infine utilizziamo due funzioni, una per trovare il miglior giocatore in casa e una per trovare il miglior giocatore fuori casa.

5.11 Trigger

I trigger riferiti alla classifica dei bookmaker non fanno altro che

chiamare la procedura *refresh_bookmaker_mview* ogni volta che una delle tabelle *quotes*, *match* o *bookmaker* vengono modificate.

```
CREATE TRIGGER refresh_bookmaker_mview_trigger AFTER INSERT OR
UPDATE OR DELETE
  ON soccerscheme.quotes
  FOR EACH ROW
    EXECUTE PROCEDURE refresh_bookmaker_mview();
```

```
CREATE TRIGGER refresh_bookmaker_mview_trigger1 AFTER INSERT OR
UPDATE OR DELETE
  ON soccerscheme.match
  FOR EACH ROW
    EXECUTE PROCEDURE refresh_bookmaker_mview();
```

```
CREATE TRIGGER refresh_bookmaker_mview_trigger2 AFTER INSERT OR
UPDATE OR DELETE
  ON soccerscheme.bookmaker
  FOR EACH ROW
    EXECUTE PROCEDURE refresh_bookmaker_mview();
```

```
CREATE TRIGGER refresh_bookmaker_mview_trigger2 AFTER INSERT OR
UPDATE OR DELETE
  ON soccerscheme.partner
  FOR EACH ROW
    EXECUTE PROCEDURE refresh_bookmaker_mview();
```

I trigger riferiti alla classifica dei campionati si differenziano solo per le modifiche alle tabelle per le quali vengono chiamati(*match*, *team*).

```
CREATE TRIGGER refresh_leaderboard_mview_trigger AFTER INSERT OR
UPDATE OR DELETE
  ON soccerscheme.match
  FOR EACH ROW
    EXECUTE PROCEDURE refresh_leaderboard_mview();
```

```
CREATE TRIGGER refresh_leaderboard_mview_trigger AFTER INSERT OR
UPDATE OR DELETE
  ON soccerscheme.team
  FOR EACH ROW
    EXECUTE PROCEDURE refresh_leaderboard_mview();
```

6 Scelte di progettazione dell'applicazione web

6.1 Autenticazione degli utenti

L'autenticazione degli utenti avviene attraverso una semplice sessione, dove vengono salvati le informazioni base dell'utente come l'id ed il nome.

Nel database non viene salvata la password degli utenti ma solamente un salted hash creato dalle funzioni standard PHP `password_hash`. In questo modo, anche in caso di intrusioni nel server, le password degli utenti sono al sicuro.

Accedendo come *administrator*, dalla navbar si può raggiungere l'*admin area*, dove è possibile inserir, cancellare, effettuare modifiche ai dati o importare i CSV.

Accedendo invece come *operator*, dalla navbar sarà possibile raggiungere l'*operator area*, dove l'operatore può inserire nuovi match. Infine, accedendo come *partner*, dalla navbar si può raggiungere l'area *matches-partner*. In questa area un partner può vedere i dati dei match e inserire le quote relative ad un match per conto della società di scommesse a cui l'utente è collegato.

6.2 Form di registrazione

Selezionando dalla navbar *login*, ci spostiamo nella schermata di login e registrazione.

Nella form di sinistra, inserendo *username* e *password* e selezionando la tipologia di utente, se l'utente è presente nel database, sarà possibile accedere. Altrimenti verrà stampato un messaggio d'errore. Nella form di destra è invece possibile registrarsi nel database. Nel caso la tipologia di utente sia *partner*, è possibile inserire anche il nome del bookmaker associato. Altrimenti la casella corrispondente viene bloccata da una funzione Javascript.

6.3 Form di inserimento dati

Un utente di tipo *administrator*, accedendo nell'area a lui dedicata, si trova di fronte a svariate form di inserimento dati. Oltre all'inserimento di CSV, In quest'area l'utente è in grado di creare, modificare o eliminare qualsiasi dato riguardante un match, team o player o riguardante un bookmaker fatta eccezione per le quote.

Un utente di tipo *operator*, nell'area a lui dedicata, può scegliere due team tra quelli presenti nel database e creare un nuovo match. Può inoltre associare giocatori esistenti ad un determinato match, inserendo il nome del giocatore.

Un utente di tipo *partner* può, infine, scegliere un match tra quelli inseriti nel database. Da una form può scegliere se inserire, modificare o cancellare le quote del match relative alla società di scommesse alla quale è collegato. Nel caso esistano già le quote, l'insert non andrà a buon fine. Nel caso di cancellazione di quote, le caselle dove le quote vanno inserite vengono bloccate da una funzione Javascript.