



Università degli Studi di Salerno



Dipartimento di Ingegneria dell'Informazione ed Elettrica e
Matematica Applicata

Corso di Laurea in Ingegneria Informatica

Basi di Dati 2021/2022

Canale A-H

Project Work

Traccia N. 4 - Sport Tech: eventi e azioni in match sportivi

Gruppo n. **14 – AH**

WP	Cognome e Nome	Matricola	e-mail	Responsabil e
1	Di Maio Marco	0612704713	m.dimaio39@studenti.unisa.it	
2	Cuomo Ferdinando	0612704742	f.cuomo35@studenti.unisa.it	X
3	Ciancio Vittorio	0612704825	v.ciancio2@studenti.unisa.it	
4	Ambrosone Alessandro	0612704917	a.ambrosone8@studenti.unisa.it	

Anno accademico 2021-2022

Sommario

1. Descrizione della realtà di interesse	3
1.1. Analisi della realtà di interesse	4
2. Analisi delle specifiche	6
2.1. Glossario dei termini	6
2.2. Strutturazione dei requisiti in frasi	6
2.2.1. Frasi di carattere generale	6
2.2.2. Frasi relative a giocatore	6
2.2.3. Frasi relative a partita	6
2.2.4. Frasi relative a evento	6
2.2.5. Frasi relative ad azione	7
2.2.6. Frasi relative a competizione	7
2.2.7. Frasi relative a stadio	7
2.3. Identificazione delle operazioni principali	7
3. Progettazione Concettuale	8
3.1. Schema Concettuale	8
3.1.1. Note sullo schema E-R	8
3.2. Design Pattern	9
3.2.1. Pattern Evoluzione di concetto	9
3.2.2. Pattern Part of	9
3.3. Dizionario dei Dati	10
3.4. Regole Aziendali	12
4. Progettazione Logica	13
4.2. Ristrutturazione Schema Concettuale	13
4.2.2. Analisi delle Prestazioni	13
4.2.2.1. Tavola dei volumi	13
4.2.2.2. Tavola delle operazioni	13
4.3. Analisi delle ridondanze	14
4.3.2. Analisi della ridondanza: Numero azioni totali	14
4.3.2.1. Valutazione della ridondanza	15
4.4. Eliminazione delle generalizzazioni	15
4.4.2. Generalizzazione <i>Partita</i>	15
4.4.3. Generalizzazione <i>Evento</i>	15
4.4.4. Generalizzazione <i>Azione</i>	15
4.5. Partizionamento/Accorpamento Entità e Associazioni	15
4.6. Scelta degli identificatori principali	15
4.7. Schema ristrutturato finale	16
4.8. Schema logico	16

4.9.	Documentazione dello schema logico	17
5.	Normalizzazione	18
6.	Script Creazione e Popolamento Database	19
7.	Query SQL	20
7.2.	Query con operatore di aggregazione e join: Giocatori che hanno fatto più di dieci passaggi nelle azioni concluse	20
7.3.	Query nidificata complessa: Statistica passaggi zone	20
7.4.	Query insiemistica: Azioni gol	21
7.5.	Eventuali Altre query	21
7.5.2.	Numero dribbling	21
7.5.3.	Evoluzione gioco	21
7.5.4.	Stampa eventi	21
8.	Viste	22
8.2.	Vista gol stagione	22
8.3.	Vista gol competizione	22
8.3.2.	Query con Vista: Capocannoniere competizione	22
9.	Trigger	23
9.2.	Trigger inizializzazione	23
9.2.2.	Trigger1: <i>checkstadio</i>	23
9.2.3.	Trigger2: <i>check_gol</i>	23
9.2.4.	Trigger3: <i>check_posticipazione</i>	24
9.2.5.	Trigger4: <i>check_numeventi</i>	24
9.2.6.	Trigger5: <i>check_eventi</i>	25
9.3.	Trigger per vincoli aziendali	26
9.3.2.	Trigger1: <i>Atleast7</i>	26
9.3.3.	Trigger2: <i>check_presenza</i>	26
9.3.4.	Trigger3: <i>check_delete_partecipazione</i>	27
9.3.5.	Trigger4: <i>check_tipo_evento</i>	27
9.3.6.	Trigger5: <i>check_azione</i>	28

1. Descrizione della realtà di interesse

Titolo: Sport Tech: eventi e azioni in match sportivi

Con il termine Sport Tech si fa genericamente riferimento all'intersezione tra lo sport e la tecnologia, dall'innovazione nel campo delle performance degli atleti (per il monitoraggio, la preparazione, etc.), alle nuove modalità di fan engagement, al supporto per il management e l'organizzazione.

In questo ambito, con il termine sport analytics si fa riferimento all'acquisizione, elaborazione e analisi di informazioni relative, tra le altre, alle singole azioni e giocate durante i match, come ad esempio durante le partite di calcio, al fine di supportare gli sportivi, i preparatori, gli allenatori, le società, nell'analisi delle partite e delle prestazioni dei giocatori, ottenendo informazioni di grande interesse per migliorare le performance.

In tale contesto, è di interesse realizzare un database per la memorizzazione di azioni e giocate relative ad uno sport (e.g., calcio, basket, tennis, etc.) a supporto di applicazioni di sport analytics.

In particolare, è di interesse memorizzare almeno:

- Le informazioni sul match: luogo, stadio/struttura, squadre o giocatori coinvolti, e altre informazioni rilevanti, come la competizione (e.g., Champions League, Roland Garros, etc.), l'esito del match, la durata, etc. Se presenti, sono interessanti informazioni dettagliate sulle squadre.
- Informazioni sull'evento che si intende tracciare: tipologia di evento (e.g., nel calcio: passaggio, tiro, contrasto; nel tennis: battuta, fallo, volee, dritto, rovescio, etc.), giocatori coinvolti, minuto in cui è avvenuto, posizioni dei giocatori se rilevanti, etc.
- Per i giocatori, sono di interesse almeno le informazioni anagrafiche, i dati fisici, il numero di maglia se presente, i dati biometrici se disponibili, etc.
- Può essere di interesse memorizzare le informazioni su eventuali dispositivi, sensori e tecnologie utilizzate per effettuare i rilevamenti: telecamere, sensori wearable, etc., con i relativi dati tecnici.
- Può essere di interesse tracciare le sequenze di eventi registrati durante la partita, eventuali eventi correlati, etc.
- È di interesse rilevare informazioni contestuali circa l'evento: orario, informazioni meteorologiche, etc.
- È di interesse riportare eventuali statistiche e indici calcolati sulla base degli eventi atomici registrati per il match.

Esempi di possibili variazioni: è possibile specializzare il database rispetto al tipo di sport considerato, che influisce molto sugli eventi da memorizzare e le tecnologie impiegate (e.g., calcio, tennis, basket, automobilismo, motociclismo, ciclismo, sport acquatici, etc.). Inoltre, fermo restando la necessità di rappresentare alcune informazioni di base quali il match, i giocatori, e gli eventi, è possibile focalizzare l'attenzione maggiormente su dettagli relativi all'evento e alle tecnologie impiegate, oppure sulle statistiche e report da realizzare.

1.1. Analisi della realtà di interesse

L'obiettivo del progetto è di realizzare un database per memorizzare le informazioni relative alle azioni di una squadra di calcio che avvengono durante il match.

Si considera l'analisi di una singola squadra di calcio composta da venticinque giocatori per una singola stagione. Per ciascun giocatore vengono definiti: il numero del tesserino, il nome, il cognome, il numero di maglia, l'altezza, il peso, la data di nascita, la nazionalità e il piede forte.

Relativamente alla partita, rappresentiamo: la giornata relativa alla competizione a cui si sta partecipando, la data in cui questa viene disputata, i minuti totali giocati (compresi supplementari ed eventuali recuperi), il numero di azioni totali effettuate, la squadra di casa e la squadra ospite, con il risultato finale della partita e in caso di eventuali posticipazioni la data di posticipazione.

Ad ogni partita devono partecipare un minimo di 7 giocatori e un massimo di 16 giocatori (11 titolari più 5 sostituzioni). Un giocatore titolare non sostituito avrà come minuto di ingresso 0 e come minuto di uscita la durata della partita totale.

In questo progetto l'attenzione sarà focalizzata sulla rappresentazione delle azioni che avvengono durante la partita; specificheremo il numero dell'azione, l'esito dell'azione (Fermata, Conclusa).

- **Azione fermata:** il giocatore che ha perso il possesso palla, come è stata fermata e la zona.

- **Azione conclusa:** esito dell'azione (gol/no gol).

Un'azione si definisce conclusa quando l'ultimo evento è un tiro; invece, fermata, nel caso è avvenuto un intervento della squadra avversaria, il passaggio non è stato realizzato oppure c'è stato un fuorigioco.

Per gli eventi è di interesse rappresentare il passaggio, il tiro e la conduzione palla.

- **Passaggio:** il giocatore che effettua il passaggio, la zona di campo da cui è partito ed è arrivato ed il tipo di passaggio (Rimessa, Angolo, Corto, Lungo e Cross).

- **Tiro:** il giocatore che effettua il tiro, l'esito del tiro.

- **Conduzione palla:** il giocatore che effettua la conduzione palla, la zona di campo da cui è partita, la zona in cui termina e il numero di dribbling effettuati.

Non inseriamo il giocatore ricevente del passaggio poiché è il giocatore che comincerà l'evento successivo. Le rimesse laterali e i calci d'angolo le inseriamo nel 'tipo' del passaggio. Le conduzioni palla che analizziamo sono solo quelle in cui il calciatore cambia zona.

Per descrivere l'azione divideremo il campo da calcio in ventuno zone.

Viene diviso in quattro sezioni verticali e in cinque orizzontali.

I diversi settori formatisi, prendendo come punto di partenza l'area di rigore della squadra presa in esame, si chiameranno:

- A(1,2,3,4)
- B(1,2,3,4)
- C(1,2,3,4)
- D(1,2,3,4)
- E(1,2,3,4)
- F

L'immagine descrive la posizione delle zone sopra citate.

Non vengono rappresentati eventuali calci piazzati che terminano di prima battuta in porta, questo poiché il nostro database si focalizza sulle azioni di movimento e non su calci piazzati.



Tutti i dati raccolti per rappresentare le azioni descritte sopra verranno raggruppati in partite delle diverse competizioni a cui la squadra partecipa, includendo per ogni partita i giocatori titolari, le avvenute sostituzioni, il ruolo occupato da ciascun giocatore durante la partita e lo stadio in cui quest'ultima viene disputata.

Per la competizione rappresentiamo il nome della competizione.

Per lo stadio rappresentiamo il nome della struttura, la nazione, la città in cui è situato e la capienza massima che può contenere.

2. Analisi delle specifiche

Workpackage	Task	Responsabile
WP0	Analisi delle specifiche	Intero Gruppo

2.1. Glossario dei termini

	Termine	Descrizione	Sinonimi	Collegamenti
1	Giocatore	L'atleta è un giocatore tesserato della squadra di calcio.	Atleta	Evento, Partita
2	Azione	Sequenza di eventi.	-	Evento, Partita
3	Partita	Incontro tra due squadre di calcio.	Match	Giocatore, Stadio, Competizione, Azione
4	Evento	Rappresenta il passaggio, il tiro o la conduzione palla, avvenuto durante la partita.	Tiro, Passaggio, Conduzione palla	Giocatore, Azione
5	Competizione	Partite di campionato o di un determinato torneo a cui la squadra partecipa.	-	Partita
6	Stadio	Struttura in cui viene disputata la partita di calcio.	-	Partita

Tabella 1. Regole di derivazione

2.2. Strutturazione dei requisiti in frasi

2.2.1. Frasi di carattere generale

Si vuole realizzare un database per memorizzare le informazioni relative alle azioni di una squadra di calcio che avvengono durante il match di una stagione.

2.2.2. Frasi relative a giocatore

Per ciascun giocatore vengono definiti: il numero del tesserino, il nome, il cognome, il numero di maglia, l'altezza, il peso, la data di nascita, la nazionalità e il piede forte.

2.2.3. Frasi relative a partita

Relativamente alla partita, rappresentiamo: la giornata relativa alla competizione a cui si sta partecipando, la data in cui questa viene disputata, i minuti totali giocati (compresi supplementari ed eventuali recuperi), il numero di azioni totali effettuate, la squadra di casa e la squadra ospite, con il risultato finale della partita e in caso di eventuali posticipazioni la data di posticipazione.

Ad ogni partita devono partecipare un minimo di 7 giocatori e un massimo di 16 giocatori (11 titolari più 5 sostituzioni). Un giocatore titolare non sostituito avrà come minuto di ingresso 0 e come minuto di uscita la durata della partita totale.

2.2.4. Frasi relative a evento

Per gli eventi è di interesse rappresentare il passaggio, il tiro e la conduzione palla.

- *Passaggio: il giocatore che effettua il passaggio, la zona di campo da cui è partito ed è arrivato il passaggio ed il tipo di passaggio (Rimessa, Angolo, Corto, Lungo e Cross).*
- *Tiro: il giocatore che effettua il tiro, l'esito del tiro.*
- *Conduzione palla: il giocatore che effettua la conduzione palla, la zona di campo da cui è partita, la zona in cui termina e il numero di dribbling effettuati.*

Non inseriamo il giocatore ricevente del passaggio poiché è il giocatore che comincerà l'evento successivo.

Le rimesse laterali e i calci d'angolo le inseriamo nel 'tipo' del passaggio. Le conduzioni palla che analizziamo sono solo quelle in cui il calciatore cambia zona.

2.2.5. Frasi relative ad azione

Per l'azione rappresentiamo il numero dell'azione, l'esito dell'azione (Fermata, Conclusa).

- *Azione fermata: il giocatore che ha perso il possesso palla e come è stata fermata e la zona*
- *Azione conclusa: esito dell'azione (gol/no gol).*

Un'azione si definisce conclusa quando l'ultimo evento è un tiro; invece, fermata, nel caso è avvenuto un intervento della squadra avversaria, il passaggio non è stato realizzato oppure c'è stato un fuorigioco.

Per descrivere l'azione divideremo il campo da calcio in ventuno zone.

Non vengono rappresentati eventuali calci piazzati che terminano di prima battuta in porta, questo poiché il nostro database si focalizza sulle azioni di movimento e non su calci piazzati.

2.2.6. Frasi relative a competizione

Per la competizione rappresentiamo il nome della competizione.

2.2.7. Frasi relative a stadio

Per lo stadio rappresentiamo il nome della struttura, la nazione e la città in cui è situata e la capienza massima che può contenere.

2.3. Identificazione delle operazioni principali

Operazione 1: Inserire i dati di una partita (2 volte a settimana)

Operazione 2: Inserire un nuovo giocatore o rimuoverne uno attuale (2 volte a stagione)

Operazione 3: Stampa i dati di una partita (incluso il numero di azioni totali) (2 volte a settimana)

Operazione 4: Inserire i dati degli eventi di un'azione (140 volta a settimana)

Operazione 5: Realizzare una statistica delle zone di campo in cui ci sono state più azioni fermate (1 volta a settimana)

Operazione 6: Inserire i dati di un'azione che costituisce la partita (140 volte a settimana)

3. Progettazione Concettuale

Workpackage	Task	Responsabile
WP1	Progettazione Concettuale	Di Maio Marco

3.1. Schema Concettuale

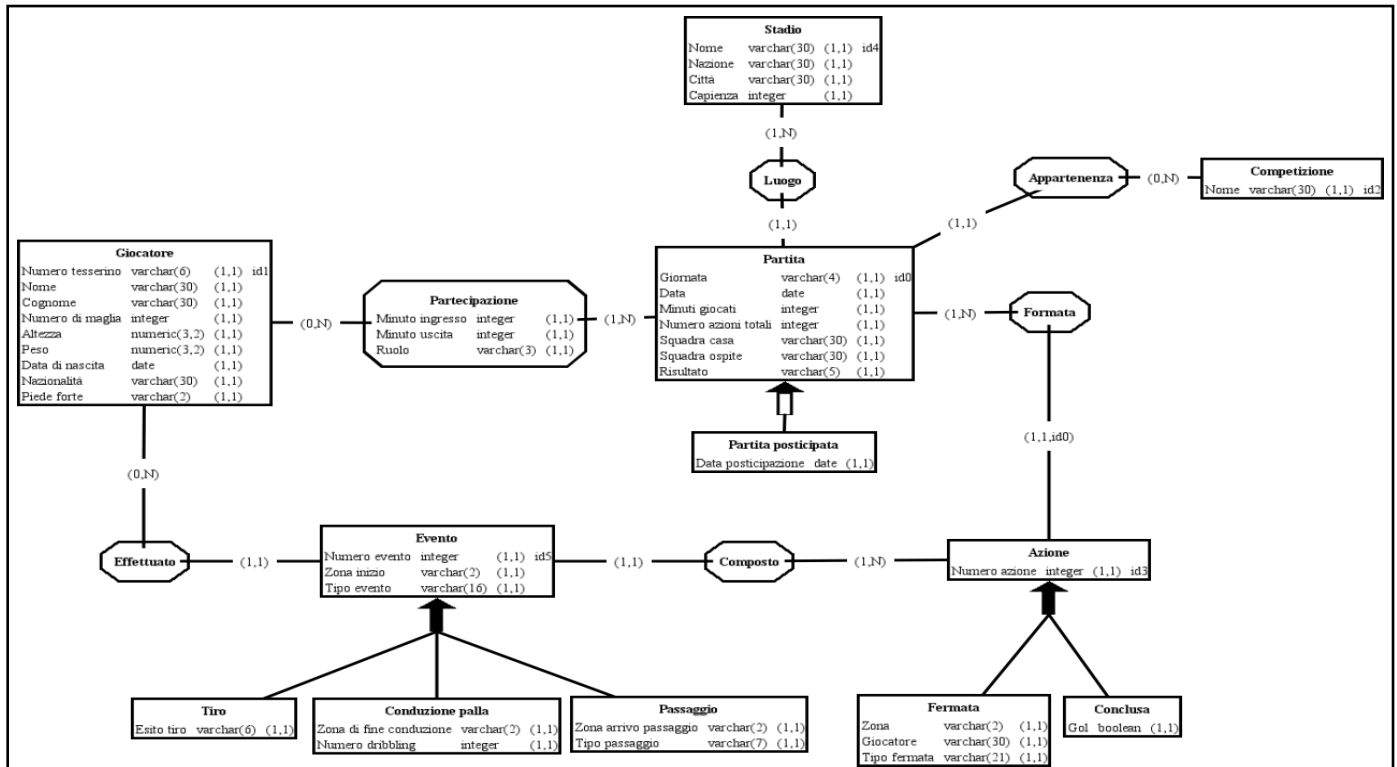


Figura 1. Schema E-R.

3.1.1. Note sullo schema E-R

La strategia utilizzata per la progettazione concettuale è Top – Down. Lo schema è stato prodotto mediante raffinamenti successivi a partire da uno che rappresentava le specifiche con un livello di dettaglio minimo.

3.2. Design Pattern

I design pattern applicati all'interno dello schema concettuale sono: Pattern Evoluzione di concetto e Pattern Part of.

3.2.1. Pattern Evoluzione di concetto

Questo pattern è stato utilizzato per riportare lo stato di inizio di una partita. In particolare, si vuole rappresentare l'eventuale posticipazione della partita in una prossima data rispetto a quella indicata da calendario.

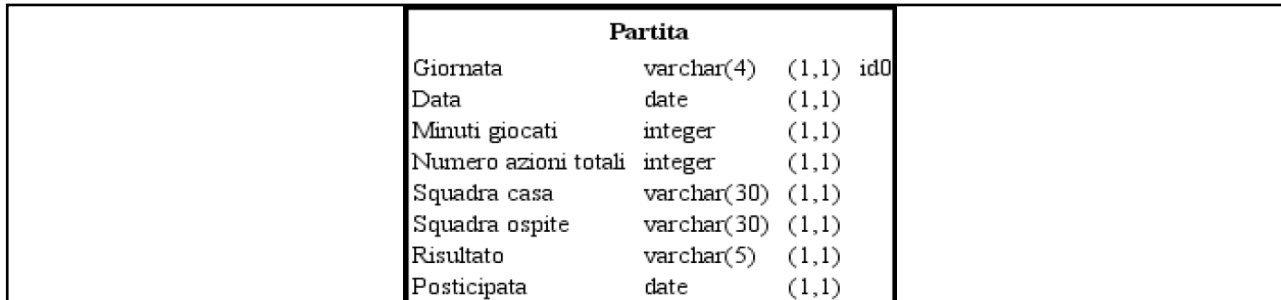


Figura 2. Schema precedente all'applicazione del Pattern EVOLUZIONE DI CONCETTO.

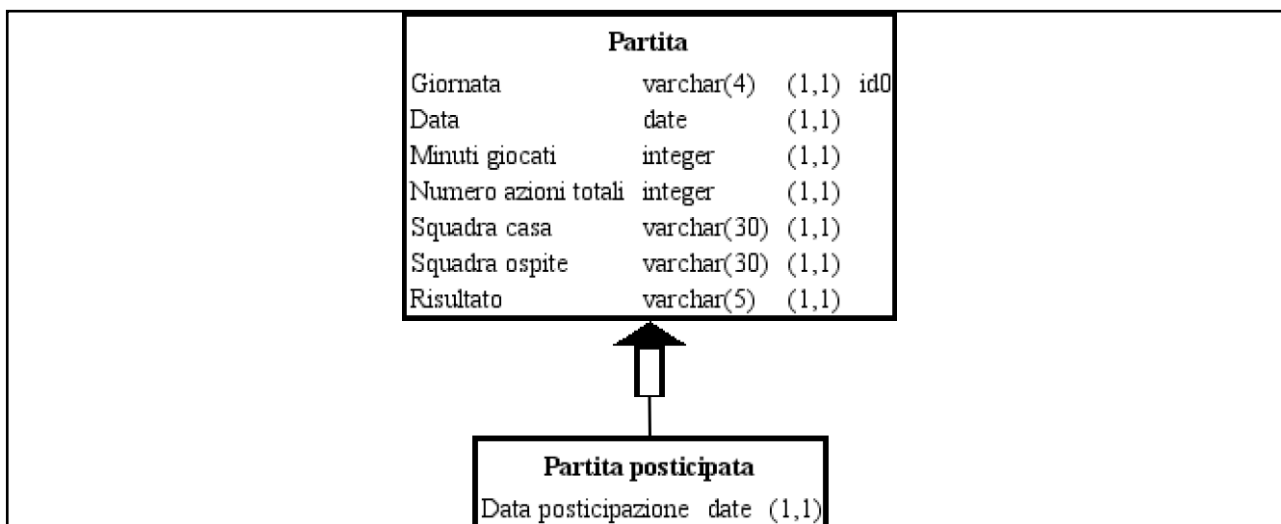


Figura 3. Schema successiva all'applicazione del Pattern EVOLUZIONE DI CONCETTO.

3.2.2. Pattern Part of

Questo pattern è stato utilizzato per rappresentare che un evento è parte dell'entità azione, essa è composta da eventi di vario tipo descritti precedentemente.

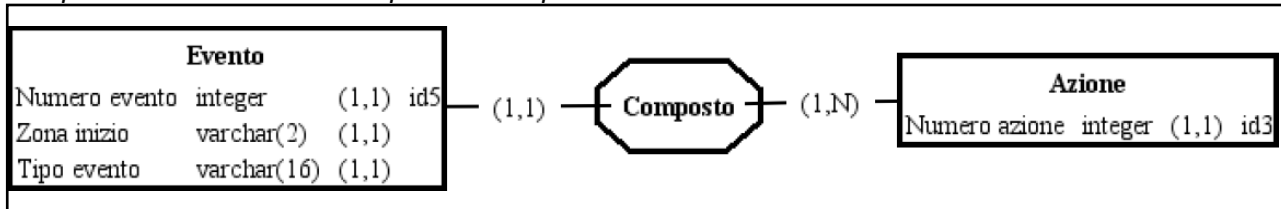


Figura 4. Schema all'applicazione del Pattern COMPOSIZIONE.

3.3. Dizionario dei Dati

Entità	Descrizione	Attributi	Identificatore
Giocatore	L'atleta è un giocatore tesserato della squadra di calcio.	<ul style="list-style-type: none"> • Numero tesserino • Nome • Cognome • Numero di maglia • Altezza • Peso • Data di nascita • Nazionalità • Piede forte 	Numero tesserino
Partita	Incontro tra due squadre di calcio.	<ul style="list-style-type: none"> • Giornata • Data • Minuti giocati • Numero azioni totali • Squadra casa • Squadra ospite • Risultato 	Giornata
Competizione	Partite di campionato o di un determinato torneo a cui la squadra partecipa.	<ul style="list-style-type: none"> • Nome 	Nome
Stadio	Struttura in cui viene disputata la partita di calcio.	<ul style="list-style-type: none"> • Nome • Nazione • Città • Capienza 	Nome
Partita posticipata	Data in cui viene posticipata la partita. Figlia dell'entità 'Partita'.	<ul style="list-style-type: none"> • Data posticipazione 	'Giornata' della tabella 'Partita'
Evento	Rappresenta il passaggio, il tiro o la conduzione palla, avvenuto durante la partita.	<ul style="list-style-type: none"> • Numero evento • Zona inizio • Tipo evento 	Numero evento
Tiro	Tipologia di evento. Figlia dell'entità 'Evento'.	<ul style="list-style-type: none"> • Esito tiro 	'Numero evento' della tabella 'Evento'
Conduzione palla	Tipologia di evento. Figlia dell'entità 'Evento'.	<ul style="list-style-type: none"> • Zona di fine conduzione • Numero dribbling 	'Numero evento' della tabella 'Evento'

Passaggio	Tipologia di evento. Figlia dell'entità 'Evento'.	<ul style="list-style-type: none"> • Zona arrivo passaggio • Tipo passaggio 	'Numero evento' della tabella 'Evento'
Azione	Sequenza di eventi.	<ul style="list-style-type: none"> • Numero azione 	Numero azione
Fermata	Azione che non termina con un tiro. Figlia dell'entità 'Azione'.	<ul style="list-style-type: none"> • Zona; Giocatore • Tipo fermata 	'Numero azione' della tabella 'Azione'
Conclusa	Azione che termina con un tiro. Figlia dell'entità 'Azione'.	<ul style="list-style-type: none"> • Gol 	'Numero azione' della tabella 'Azione'

Tabella 2. Dizionario dei dati – Entità

Relazioni	Descrizione	Entità Coinvolte	Attributi
Partecipazioni	Associa ciascun giocatore ad una determinata partita.	<ul style="list-style-type: none"> • Giocatore (0,N) • Partita (1,N) 	<ul style="list-style-type: none"> • Minuto Ingresso • Minuto uscita • Ruolo
Appartenenza	Associa ciascuna partita alla competizione di cui fa parte.	<ul style="list-style-type: none"> • Partita (1,1) • Competizione (0,N) 	-
Luogo	Associa ciascuna partita allo stadio in cui si disputa.	<ul style="list-style-type: none"> • Partita (1,1) • Stadio (1,N) 	-
Formata	Associa ciascuna partita alle azioni da cui è formata.	<ul style="list-style-type: none"> • Partita (1,N) • Azione (1,1) 	-
Composto	Associa a ciascun'azione gli eventi che la compongono.	<ul style="list-style-type: none"> • Evento (1,1) • Azione (1,N) 	-
Effettuato	Associa a ciascun'giocatore il relativo evento effettuato.	<ul style="list-style-type: none"> • Giocatore (0,N) • Evento (1,1) 	-

Tabella 3. Dizionario dei dati - Relazioni

Workpackage	Task	Responsabile
WP4	Regole Aziendali	Alessandro Ambrosone

3.4. Regole Aziendali

Regole di Vincolo
<p>(RV1) Ad una partita devono partecipare un minimo di 7 giocatori e un massimo di 11 più eventuali sostituzioni (5).</p> <p>(RV2) Un'azione si definisce conclusa quando l'ultimo evento è un tiro.</p> <p>(RV3) Una conduzione palla deve avere gli attributi 'zonainizio' e 'zonafineconduzione' diversi.</p> <p>(RV4) Se l'azione è conclusa non deve presentare gli attributi 'giocatorefermata' e 'zona'.</p> <p>(RV5) Azione fermata deve presentare gli attributi 'giocatorefermata' e 'zona'.</p> <p>(RV6) Gli attributi 'giocatorefermata' e 'giocatore', delle tabelle 'azione' ed 'evento', devono far riferimento ad un giocatore che partecipa alla partita.</p> <p>(RV7) Se 'tipoevento' ha come valore 'passaggio', dovranno essere valorizzati anche 'zonafine' e 'tipopassaggio'.</p> <p>(RV8) Se 'tipoevento' ha come valore 'conduzione palla', dovranno essere valorizzati anche 'zonafine' e 'numerodribling'.</p> <p>(RV9) Se 'tipoevento' ha come valore 'tiro', dovranno essere valorizzati anche 'esitotiro'.</p>

Tabella 4. Regole di vincolo

Regole di derivazione
<p>(RD1) Numero azioni totali è ottenibile contando il numero di occorrenze di una partita all'interno della tabella azione.</p>

Tabella 5. Regole di derivazione

4. Progettazione Logica

Workpackage	Task	Responsabile
WP2	Progettazione Logica	Cuomo Ferdinando

4.2. Ristrutturazione Schema Concettuale

4.2.2. Analisi delle Prestazioni

4.2.2.1. Tavola dei volumi

Concetto	Tipo	Volume
Giocatore	E	30
Partita	E	70
Stadio	E	40
Competizione	E	4
Azione	E	5000
Evento	E	51000
Partita posticipata	E	10
Tiro	E	1000
Conduzione palla	E	20000
Passaggio	E	30000
Fermata	E	3000
Conclusa	E	2000
Partecipazione	R	1000
Luogo	R	70
Appartenenza	R	70
Formata	R	5000
Composto	R	51000
Effettuata	R	51000

Tabella 6. Tavola dei volumi

4.2.2.2. Tavola delle operazioni

Operazione	Tipo	Frequenza
Operazione 1: Inserire dati partita	I	2 volte a settimana
Operazione 2: Aggiornamento rosa	I	2 volte a stagione
Operazione 3: Stampa dati partita	I	2 volte a settimana
Operazione 4: Inserire eventi di un'azione	I	140 volte a settimana
Operazione 5: Statistica palle perse	B	1 volta a settimana
Operazione 6: Inserire azione di una partita	I	140 volta a settimana

Tabella 7. Tavola delle operazioni

4.3. Analisi delle ridondanze

- **Ridondanza:** Numero azioni totali in PARTITA.
Il numero di azioni totali di una partita, si ottiene contando il numero di occorrenze dell'entità AZIONE in cui è presente l'attributo relativo a quella partita.
TIPO: Attributo Derivabile da conteggio di occorrenze.

4.3.2. Analisi della ridondanza: Numero azioni totali

- **Operazione 1: Inserire dati partita**
Con Ridondanza

Concetto	Costrutto	Accessi	Tipo
Partita	E	1	S

Senza Ridondanza

Concetto	Costrutto	Accessi	Tipo
Partita	E	1	S

- **Operazione 3: Stampa dati partita**
Con Ridondanza

Concetto	Costrutto	Accessi	Tipo
Partita	E	1	L

Senza Ridondanza

Concetto	Costrutto	Accessi	Tipo
Partita	E	1	L
Formata	R	70	L

- **Operazione 6: Inserire azione di una partita**
Con Ridondanza

Concetto	Costrutto	Accessi	Tipo
Azione	E	1	S
Formata	R	1	S
Partita	E	1	L
Partita	E	1	S

Senza Ridondanza

Concetto	Costrutto	Accessi	Tipo
Azione	E	1	S
Formata	R	1	S

4.3.2.1. Valutazione della ridondanza

Dopo aver analizzato le operazioni che coinvolgono la ridondanza si osserva che, con il carico considerato:

- In presenza di ridondanza il costo delle operazioni è di circa 986 accessi settimanali
- L'occupazione di memoria è di circa 4*70(280) byte
- In assenza di ridondanza il costo delle operazioni è di 706 accessi settimanali

Pertanto, si decide di rimuovere la ridondanza in quanto riduce il numero di accessi.

4.4. Eliminazione delle generalizzazioni

4.4.2. Generalizzazione Partita

Si applica un accorpamento delle figlie della generalizzazione nel genitore. L'entità Partita posticipata viene eliminata e aggiunta all'entità genitore Partita. A tale entità viene aggiunto un ulteriore attributo 'Posticipazione' (attributo booleano) che serve a distinguere se una partita è stata posticipata oppure no. Questo accorpamento è stato utilizzato poiché l'operazione che coinvolge l'entità Partita non fa molta distinzione tra le occorrenze e gli attributi di Partita e Partita posticipata. In questo caso si avrà un numero minore di accessi in quanto le diverse occorrenze sono state accorpate nell'unica entità Evento.

4.4.3. Generalizzazione Evento

Si applica un accorpamento delle figlie della generalizzazione nel genitore. Le entità Tiro, Conduzione palla e Passaggio vengono eliminate e aggiunte all'entità genitore Evento. A tale entità viene aggiunto un ulteriore attributo 'Tipo evento' che serve a distinguere il "tipo" di un'occorrenza di Evento. Questo accorpamento è stato utilizzato poiché l'operazione che coinvolge l'entità Evento non fa molta distinzione tra le occorrenze e gli attributi di Evento, Tiro, Conduzione palla e Passaggio. In questo caso si avrà un numero minore di accessi in quanto le diverse occorrenze sono state accorpate nell'unica entità Evento.

4.4.4. Generalizzazione Azione

Si applica un accorpamento delle figlie della generalizzazione nel genitore. Le entità Fermata e Conclusa vengono eliminate e aggiunte all'entità genitore Azione. A tale entità viene aggiunto un ulteriore attributo 'Tipo azione' che serve a distinguere il "tipo" di un'occorrenza di Azione. Questo accorpamento è stato utilizzato poiché l'operazione che coinvolge l'entità Azione non fa molta distinzione tra le occorrenze e gli attributi di Azione, Fermata e Conclusa. In questo caso si avrà un numero minore di accessi in quanto le diverse occorrenze sono state accorpate nell'unica entità Azione.

4.5. Partizionamento/Accorpamento Entità e Associazioni

Indicare le eventuali partizioni e accorpamento di entità o associazioni effettuate, motivando le scelte.

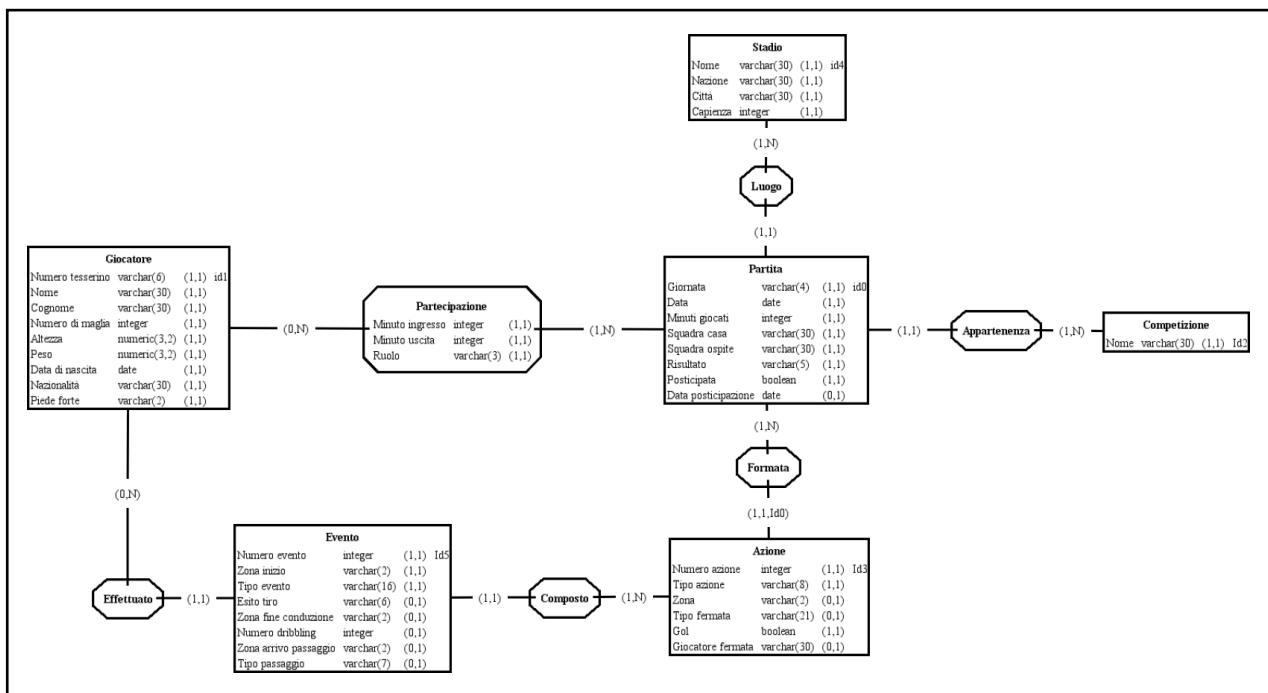
Inserire qui anche l'eventuale eliminazione di attributi multivalore.

4.6. Scelta degli identificatori principali

- *Relativamente all'entità Giocatore si è scelto di lasciare l'identificatore principale Numero Tesserino in quanto rispecchia già i requisiti.*
- *Relativamente all'entità Partita si è scelto di lasciare l'identificatore principale Giornata in quanto rispecchia già i requisiti.*

- Relativamente all'entità **Stadio** si è scelto di lasciare l'identificatore principale **Nome** in quanto rispecchia già i requisiti.
- Relativamente all'entità **Competizione** si è scelto di lasciare l'identificatore principale **Nome** in quanto rispecchia già i requisiti.
- Relativamente all'entità **Evento** si è scelto di lasciare l'identificatore principale **Numero Evento** in quanto rispecchia già i requisiti.
- Relativamente all'entità **Azione** si è scelto di lasciare l'identificatore principale **Numero Azione** in quanto rispecchia già i requisiti.

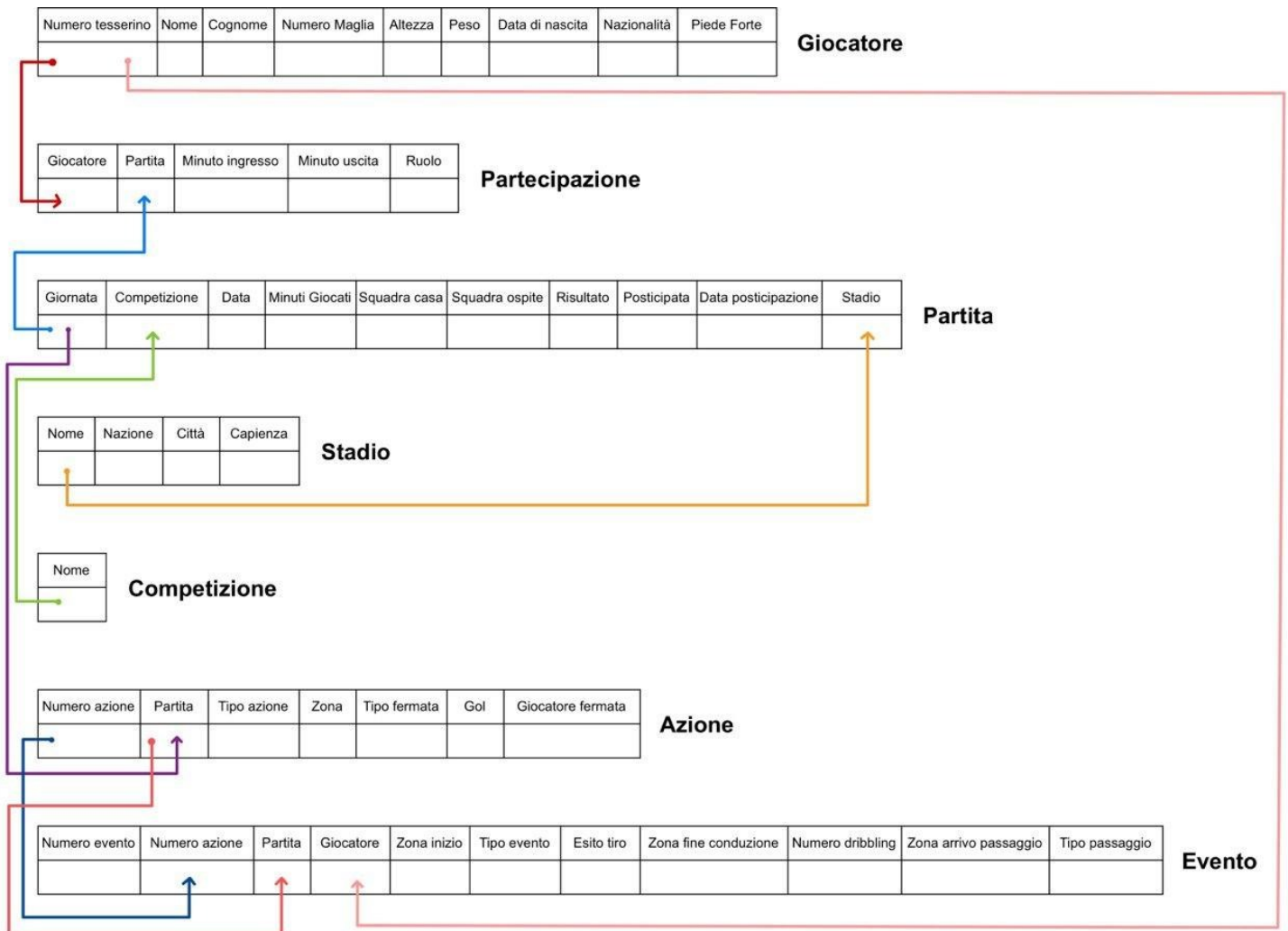
4.7. Schema ristrutturato finale



4.8. Schema logico

- **Giocatore** (Numero Tesserino, Nome, Cognome, Numero Maglia, Altezza, Peso, Data di Nascita, Nazionalità, Piede Forte)
- **Partecipazione** (Giocatore, Partita, Minuto Ingresso, Minuto Uscita, Ruolo)
- **Partita** (Giornata, Competizione, Data, Minuti Giocati, Squadra Casa, Squadra Ospite, Risultato, Posticipata, Data Posticipazione*, Stadio)
- **Stadio** (Nome, Nazione, Città, Capienza)
- **Competizione** (Nome)
- **Azione** (Numero Azione, Partita, Tipo Azione, Zona*, Tipo Fermata*, Gol, Giocatore Fermata*)
- **Evento** (Numero Evento, Numero Azione, Partita, Giocatore, Zona Inizio, Tipo Evento, Esito Tiro*, Zona Fine Conduzione*, Numero Dribbling*, Zona Arrivo Passaggio*, Tipo Passaggio*)

4.9. Documentazione dello schema logico



5. Normalizzazione

Workpackage	Task	Responsabile
WP3	Normalizzazione	Ciancio Vittorio

PRIMA FORMA NORMALE (1FN)

Una tabella è in Prima Forma Normale (1FN) se presenta domini di attributi atomici, cioè valori indivisibili, quindi informazioni elementari, se ogni tupla è diversa dalle altre e il loro ordine è irrilevante; pertanto, quando è in linea con i requisiti del modello relazionale.

FORMA NORMALE DI BOYCE AND CODD (BCNF)

Una relazione è in BCNF, quando rispetta le caratteristiche fondamentali del modello relazionale, quindi quando si trova in 1FN, e in essa ogni DETERMINANTE delle dipendenze funzionali è una chiave candidata, cioè ogni attributo dal quale dipendono altri attributi può svolgere la funzione di chiave.

- **Giocatore** (Numero Tesserino, Nome, Cognome, Numero Maglia, Altezza, Peso, Data di Nascita, Nazionalità, Piede Forte)
La relazione è in BCNF
- **Partecipazione** (Giocatore, Partita, Minuto Ingresso, Minuto Uscita, Ruolo)
La relazione è in BCNF
- **Partita** (Giornata, Competizione, Data, Minuti Giocati, Squadra Casa, Squadra Ospite, Risultato, Posticipata, Data Posticipazione*, Stadio)
La relazione è in BCNF
- **Stadio** (Nome, Nazione, Città, Capienza)
La relazione è in BCNF
- **Competizione** (Nome)
La relazione è in BCNF
- **Azione** (Numero Azione, Partita, Tipo Azione, Zona*, Tipo Fermata*, Gol, Giocatore Fermata*)
La relazione è in BCNF
- **Evento** (Numero Evento, Numero Azione, Partita, Giocatore, Zona Inizio, Tipo Evento, Esito Tiro*, Zona Fine Conduzione*, Numero Dribbling*, Zona Arrivo Passaggio*, Tipo Passaggio*)
La relazione è in BCNF

6. Script Creazione e Popolamento Database

Workpackage	Task	Responsabile
WP2	SQL: Script creazione e popolamento	Cuomo Ferdinando

Per la creazione ed il popolamento del database si fa riferimento allo script allegato alla relazione, poiché contenente troppe righe.

7. Query SQL

Workpackage	Task	Responsabile
WP3	SQL: Query	Ciancio Vittorio

- Giocatori che hanno fatto più di dieci passaggi nelle azioni concluse
- Statistica passaggi zone
- Azioni gol

7.2. Query con operatore di aggregazione e join: Giocatori che hanno fatto più di dieci passaggi nelle azioni concluse

Stampa i giocatori che hanno effettuato più di dieci passaggi nelle azioni concluse.

```

1 select cognome, count(distinct numeroevento) as Numeropassaggi
2 from Evento e
3     join azione az on(e.numeroazione = az.numeroazione)
4     and az.tipoazione = 'Conclusa' and e.tipoevento = 'Passaggio'
5     join giocatore g1 on(g1.numerotesserino = e.giocatore)
6 group by g1.cognome
7 having count(distinct numeroevento)>10

```

7.3. Query nidificata complessa: Statistica passaggi zone

Stampa il numero di passaggi ricevuti ed effettuati nelle singole zone del campo.

```

1 select passaggi_ricevuti.zona, numero_passaggi_ricevuti, numero_passaggi_effettuati
2 from (select zonaarrivopassaggio as zona, count(*) as numero_passaggi_ricevuti
3 from evento
4 where tipoevento='Passaggio' and (zonaarrivopassaggio <> 'F' or zonaarrivopassaggio is not null)
5 group by zonaarrivopassaggio
6 order by numero_passaggi_ricevuti desc) as passaggi_ricevuti
7 join (select zonainizio as zona, count(*) as numero_passaggi_effettuati
8 from evento
9 where tipoevento='Passaggio' and zonainizio <> 'F'
10 group by zonainizio
11 order by numero_passaggi_effettuati desc) as passaggi_effettuati on passaggi_ricevuti.zona =passaggi_effettuati.zona
12 order by passaggi_ricevuti.zona

```

7.4. Query insiemistica: Azioni gol

Stampa le azioni che hanno portato al gol per ogni competizione.

```

1  select *
2  from azione
3  where gol = true and partita in (
4      select giornata
5      from partita
6      where giornata like 'SA%'
7  )
8
9  union
10 select *
11 from azione
12 where gol = true and partita in (
13     select giornata
14     from partita
15     where giornata like 'EL%'
16 )
17
18 union
19 select *
20 from azione
21 where gol = true and partita in (
22     select giornata
23     from partita
24     where giornata like 'CL%'
25 )
26
27 union
28 select *
29 from azione
30 where gol = true and partita in (
31     select giornata
32     from partita
33     where giornata like 'CI%'
34 )
35 order by partita

```

7.5. Eventuali Altre query

7.5.2. Numero dribbling

Stampa le azioni che hanno portato al gol per ogni competizione.

```

1  select cognome,sum(umerodribbling) as dribblingtot
2  from evento join giocatore on evento.giocatore = giocatore.numerotesserino
3  where tipoevento = 'Conduzione palla' and umerodribbling > 0
4  group by cognome
5  order by dribblingtot desc

```

7.5.3. Evoluzione gioco

Stampa il numero di verticalizzazioni e il numero di passaggi verso l'esterno per ogni partita.

```

1  select verticalizzazioni.giornata,numero_verticalizzazioni,numero_passaggi_verso_esterno
2  from(select giornata,count(*) as numero_verticalizzazioni
3  from partita join evento on partita.giornata = evento.partita
4  where tipoevento = 'Passaggio' and (zonaarrivopassaggio = 'E2' or zonaarrivopassaggio = 'E3' or zonaarrivopassaggio = 'D2' or zonaarrivopassaggio = 'D3') and zonainizio <>'F'
5  group by giornata) as verticalizzazioni
6  join
7  (select giornata,count(*) as numero_passaggi_verso_esterno
8  from partita join evento on partita.giornata = evento.partita
9  where tipoevento = 'Passaggio' and (zonaarrivopassaggio like '%1' or zonaarrivopassaggio like '%4') and zonainizio <>'F'
10 group by giornata) as esterno on verticalizzazioni.giornata = esterno.giornata
11 order by giornata asc

```

7.5.4. Stampa eventi

Stampa tutti gli eventi di una determinata partita sostituendo al numero tesserino il cognome del giocatore (sostituendo a evento.partita=partita, partita con la partita di interesse).

```

1  select cognome,evento.*
2  from evento join giocatore on evento.giocatore = giocatore.numerotesserino
3  where evento.numeroazione = numeroazione and evento.partita = partita

```

8. Viste

Workpackage	Task	Responsabile
WP4	Viste	Ambrosone Alessandro

8.2. Vista gol stagione

Stampa la somma di tutti i gol segnati da tutti i giocatori in tutte le competizioni.

```

1 drop view if exists golstagione;
2 create view golstagione as
3 select cognome,count(distinct numeroevento) as gol
4 from evento join giocatore on evento.giocatore = giocatore.numerotesserino
5 where evento.esitotiro = 'Gol'
6 group by cognome
7 order by gol desc;
```

8.3. Vista gol competizione

Stampa i gol segnati da ogni giocatore in Champion League, Europa League, Coppa Italia, Serie A.

```

1 drop view if exists golCL;
2 create view golCL as
3 select cognome,competizione,count(distinct numeroevento) as gol
4 from evento join partita on evento.partita = partita.giornata and evento.esitotiro = 'Gol' and partita.competizione = 'UEFA Champions League'
5 join giocatore on evento.giocatore = giocatore.numerotesserino
6 group by cognome,competizione;

1 drop view if exists golEL;
2 create view golEL as
3 select cognome,competizione,count(distinct numeroevento) as gol
4 from evento join partita on evento.partita = partita.giornata and evento.esitotiro = 'Gol' and partita.competizione = 'UEFA Europa League'
5 join giocatore on evento.giocatore = giocatore.numerotesserino
6 group by cognome,competizione;

1 drop view if exists golcoppaitalia;
2 create view golcoppaitalia as
3 select cognome,competizione,count(distinct numeroevento) as gol
4 from evento join partita on evento.partita = partita.giornata and evento.esitotiro = 'Gol' and partita.competizione = 'Coppa Italia'
5 join giocatore on evento.giocatore = giocatore.numerotesserino
6 group by cognome,competizione;

1 drop view if exists golseriea;
2 create view golseriea as
3 select cognome,competizione,count(distinct numeroevento) as gol
4 from evento join partita on evento.partita = partita.giornata and evento.esitotiro = 'Gol' and partita.competizione = 'Serie A'
5 join giocatore on evento.giocatore = giocatore.numerotesserino
6 group by cognome,competizione;
```

8.3.2. Query con Vista: Capocannoniere competizione

Stampa il giocatore che ha realizzato più gol in ogni competizione.

```

1 select *
2 from golel
3 where gol=(select max(gol)
4             from golel)
5 union
6 select *
7 from golseriea
8 where gol=(select max(gol)
9             from golseriea)
10 union
11 select *
12 from golcoppaitalia
13 where gol=(select max(gol)
14             from golcoppaitalia)
15 union
16 select *
17 from golcl
18 where gol=(select max(gol)
19             from golcl)
20 order by gol desc
```

9. Trigger

9.2. Trigger inizializzazione

Workpackage	Task	Responsabile
WP1	Trigger inizializzazione/popoloamento database	Di Maio Marco

9.2.2. Trigger1: *checkstadio*

Controlla che per lo stadio inserito vi sia almeno una partita.

```

1  create or replace function atleastone() returns trigger as $$
2  begin
3  if(not exists (select * from partita where partita.stadio = new.nome) )
4  then raise exception 'Inserire una partita';
5  end if;
6  return new;
7  end $$
8  language plpgsql;
9
10 create constraint trigger checkstadio
11 after insert on stadio
12 deferrable initially deferred
13 for each row
14 execute procedure atleastone();

```

9.2.3. Trigger2: *check_gol*

Controlla che se l'azione ha l'attributo gol 'TRUE', controlla che l'attributo esito tiro in evento sia 'Gol'.

```

1  create or replace function controlla_gol() returns trigger as $$
2  declare numgol integer;
3  begin
4      select count(*) into numgol
5      from evento
6      where evento.numeroazione=new.numeroazione and evento.partita=new.partita and evento.esitotiro='Gol';
7  if(new.gol = true and numgol <> 1) then
8      raise exception 'Controlla bene il campo esitotiro degli eventi!';
9  end if;
10 return new;
11 END $$ LANGUAGE plpgsql;
12
13 create trigger check_gol
14 after insert on azione
15 for each row
16 execute procedure controlla_gol();

```


9.2.4. Trigger3: *check_posticipazione*

Controlla se la data di posticipazione sia successiva a quella stabilita da calendario.

```

1  create or replace function controlla_data() returns trigger as $$
2  declare
3  posticipata date;
4  data_partita date;
5  begin
6      if(new.posticipata = true) then
7          posticipata=new.dataposticipazione;
8          data_partita=new.data;
9          if(data_partita::timestamp > posticipata::timestamp) then
10             raise exception 'La data di postecipazione deve essere successiva alla data della partita!';
11         else
12             return new;
13         end if;
14     else
15         return new;
16     end if;
17 END $$ LANGUAGE plpgsql;
18
19 create trigger check_posticipazione
20 after insert on partita
21 for each row
22 execute procedure controlla_data()

```

9.2.5. Trigger4: *check_numeventi*

Controlla che l'azione sia composta da almeno un evento.

```

1  create or replace function controlla_numeventi() returns trigger as $$
2  declare num integer;
3  begin
4      select count(*) into num
5      from evento
6      where evento.numeroazione=new.numeroazione and evento.partita=new.partita;
7      if(num<1) then
8          raise exception 'Una azione deve essere composta da almeno un evento';
9      end if;
10     return new;
11 END $$ LANGUAGE plpgsql;
12
13 create constraint trigger check_numeventi
14 after insert on azione
15 deferrable initially deferred
16 for each row
17 execute procedure controlla_numeventi()

```

9.2.6. Trigger5: check_eventi

Controlla che all'eliminazione di un evento l'azione a cui appartiene sia composta da almeno un evento.

```
1  create or replace function controlla_eventi() returns trigger as $$
2  declare num integer;
3  begin
4      select count(*) into num
5      from evento
6      where evento.numeroazione=old.numeroazione and evento.partita=old.partita;
7  ▼  if(num<1) then
8          raise exception 'Una azione deve essere composta da almeno un evento';
9      end if;
10     return old;
11 END $$ LANGUAGE plpgsql;
12
13 create trigger check_eventi
14 after delete on evento
15 for each row
16 execute procedure controlla_eventi();|
```

9.3. Trigger per vincoli aziendali

Workpackage	Task	Responsabile
WP4	Trigger per vincoli aziendali	Ambrosone Alessandro

9.3.2. Trigger1: Atleast7

Controlla se in una partita partecipano un minimo di 7 giocatori ed un massimo di 16.

```

1 CREATE OR REPLACE FUNCTION AtLeastSEVEN() returns trigger as $$
2 declare num bigint;
3 begin
4     select count(*) into num
5     from partecipazione
6     where partecipazione.partita = new.giornata;
7     if num<7 then
8         RAISE EXCEPTION 'numero minimo di giocatori (7) non rispettato!';
9     end if;
10    if num>16 then
11        RAISE EXCEPTION 'numero massimo di giocatori (16) non rispettato!';
12    end if;
13
14    return new;
15 END $$ LANGUAGE plpgsql;
16
17 create trigger AtLeast7
18 after insert on partita
19 for each row
20 execute procedure AtLeastSEVEN();

```

9.3.3. Trigger2: check_presenza

Controlla se il giocatore che effettua un evento stia effettivamente partecipando alla partita.

```

1 drop function if exists controlla_presenza();
2 create or replace function controlla_presenza() returns trigger as $$
3 begin
4     if(not exists (select * from partecipazione join evento on partecipazione.giocatore = new.giocatore
5                     where partecipazione.partita = new.partita
6                     )
7     ) then
8         raise exception 'Il giocatore che ha effettuato questo evento non partecipa alla partita!';
9     end if;
10    return new;
11 end $$
12 language plpgsql;
13
14 create constraint trigger check_presenza
15 after insert on evento
16 deferrable initially deferred
17 for each row
18 execute procedure controlla_presenza();

```

9.3.4. Trigger3: check_delete_partecipazione

Controlla se nel caso venga eliminata un'occorrenza di partecipazione, se il numero minimo di partecipanti ad una partita è sette.

```

1 CREATE OR REPLACE FUNCTION elimina_presenza() returns trigger as $$
2 declare num bigint;
3 begin
4     select count(*) into num
5     from partecipazione
6     where partita = old.giornata;
7
8     if num<7 then
9         RAISE EXCEPTION 'numero minimo di giocatori (7) non rispettato!';
10    end if;
11    return old;
12 END $$ LANGUAGE plpgsql;
13
14 create trigger check_delete_partecipazione
15 after delete on partecipazione
16 for each row
17 execute procedure elimina_presenza();

```

9.3.5. Trigger4: check_tipo_evento

Controlla in base al tipo di evento gli attributi da inserire.

```

1 create or replace function check_evento() returns trigger as $$
2 begin
3     case new.tipoevento
4     when 'Passaggio' then
5         if(new.esitotiro is null and new.zonafinecondizione is null and new.numerodribbling is null) then
6             return new;
7         else
8             raise exception 'Controlla bene i campi';
9             end if;
10    when 'Tiro' then
11        if(new.zonaarrivopassaggio is null and new.zonafinecondizione is null and new.numerodribbling is null and new.tipopassaggio is null) then
12            return new;
13        else
14            raise exception 'Controlla bene i campi';
15            end if;
16    when 'Conduzione palla' then
17        if(new.esitotiro is null and new.zonaarrivopassaggio is null and new.tipopassaggio is null and new.numerodribbling is not null and new.zonainizio <> new.zonafinecondizione) then
18            return new;
19        else
20            raise exception 'Controlla bene i campi';
21            end if;
22    end case;
23 END

```

9.3.6. Trigger5: check_azione

Controlla se ad azione di tipo 'Conclusa', coincide l'ultimo evento della medesima azione con tipo evento 'Tiro'.

```

1  create or replace function controlla_azione() returns trigger as $$
2  declare myvar varchar(16);
3  begin
4      select tipoevento into myvar
5      from evento
6      where evento.numeroazione = new.numeroazione and numeroevento = (select max(numeroevento)
7                                                                           from evento
8                                                                           where evento.numeroazione = new.numeroazione
9                                                                           );
10     if (myvar <> 'Tiro' and new.tipoazione = 'Conclusa') then
11         raise exception 'ERRORE: l'ultimo evento di una azione conclusa deve essere un tiro';
12     end if;
13     return new;
14 END $$ LANGUAGE plpgsql;
15
16 create constraint trigger check_azione
17 after insert on azione
18 deferrable initially deferred
19 for each row
20 execute procedure controlla_azione();

```