

Elaborato per il corso di Basi di dati A.A. 2022/2023

Base di dati a supporto di un gioco FPS

Antonio Emanuele Pepe
0000921056
antonioemanuele.pepe@studio.unibo.it

Alex Giamperoli
0000923932
alex.giamperoli@studio.unibo.it

Sommario

ANALISI DEI REQUISITI	3
INTERVISTA	3
ESTRAZIONE DEI CONCETTI PRINCIPALI	4
PRINCIPALI AZIONI RICHIESTE	5
PROGETTAZIONE CONCETTUALE.....	6
SCHEMA SCHELETRO	6
SCHEMA CONCETTUALE FINALE.....	10
PROGETTAZIONE LOGICA.....	12
STIMA DEL VOLUME DEI DATI	12
OPERAZIONI PRINCIPALI E STIMA DELLA LORO FREQUENZA	13
SCHEMI DI NAVIGAZIONE E TABELLE DEGLI ACCESSI.....	13
RAFFINAMENTO DELLO SCHEMA	18
ANALISI DELLE RIDONDANZE	19
TRADUZIONE DI ENTITÀ E ASSOCIAZIONI IN RELAZIONI.....	20
SCHEMA RELAZIONALE FINALE.....	22
TRADUZIONE DELLE OPERAZIONI IN QUERY SQL.....	24
PROGETTAZIONE DELL'APPLICAZIONE.....	28
DESCRIZIONE DELL'APPLICAZIONE REALIZZATA	28

Analisi dei requisiti

Si vuole realizzare un applicativo web per la visualizzazione dello storico delle partite di un gioco del genere FPS, giocate da un utente. Tale applicativo si appoggia su di un database in cui verranno memorizzati i dati necessari, per esempio i giocatori che hanno preso parte all'incontro, i round vinti da ogni squadra, ecc.

Intervista

Si vuole tracciare lo svolgimento delle partite effettuate da un utente, registrato alla piattaforma tramite uno username e una password. Un utente deve essere in grado di eliminarsi dalla base di dati qualora lo richieda. In questo caso il suo username non potrà più essere riutilizzato da un nuovo iscritto.

Ad una partita prendono parte due squadre, una offensiva e una difensiva, che hanno l'obiettivo di vincere un certo numero di round (tredici). Al termine del dodicesimo round i ruoli delle due squadre vengono invertiti. Una partita può terminare prematuramente, in questo caso vince la squadra che ha totalizzato il maggior numero di vittorie.

Le squadre sono composte da cinque giocatori. I giocatori, prima dell'inizio effettivo della partita, dovranno scegliere un agente ciascuno. Gli agenti sono identificati da un nome e posseggono dei punti salute.

All'inizio di un round è prevista una fase di acquisto durante la quale sarà possibile ottenere delle armi. Le armi disponibili hanno un nome univoco e sono distinte in due tipologie: armi primarie e armi secondarie. È possibile acquistare una sola arma per tipologia.

Un round, di cui si vuole salvare la sua durata e l'ordine sequenziale, può essere vinto da una squadra a seconda del ruolo che ricopre. La squadra attaccante vince nel caso in cui riesca a posizionare la bomba in un sito e difenderla fino alla sua esplosione; la squadra difendente vince se riesce a evitare il piazzamento della bomba da parte degli attaccanti o, nel caso questo accada, ad effettuarne il disinnescamento. Nel caso in cui una squadra elimini completamente quella avversaria, si assicura la vittoria del round.

Si vogliono memorizzare le uccisioni avvenute durante un round, nello specifico salvandone l'istante di gioco, chi l'ha effettuata, chi l'ha subita e l'arma coinvolta.

La partita ha luogo in una mappa, caratterizzata da un nome e un numero di siti. Il sito è l'unico luogo della mappa in cui è possibile posizionare la bomba. Un sito è generalmente identificato da una lettera, non necessariamente univoca tra le diverse mappe. Si vuole salvare il sito in cui una bomba viene collocata.

Degli utenti che sono stati eliminati, si vuole comunque tenere traccia della loro partecipazione nelle partite in cui hanno giocato, in modo da rendere più chiara l'interpretazione delle informazioni agli utenti ancora iscritti.

Estrazione dei concetti principali

Termine	Descrizione
Utente/Giocatore	Colui che si iscrive alla piattaforma e visualizza i risultati delle proprie partite.
Partita	È l'incontro tra le due squadre, svolto su più round.
Round	È il momento durante il quale le due squadre si scontrano, conseguendo un esito.
Agente	Personaggio scelto da ogni giocatore.
Arma	Oggetto con cui è possibile effettuare uccisioni.
Uccisione	Sconfitta di un giocatore in un round.
Mappa	Luogo in cui si svolge una partita.

A seguito di quanto espresso nell'intervista, si possono estrapolare i vari concetti, generalizzandoli con le seguenti asserzioni:

Il database deve essere in grado di gestire un numero indefinito di **utenti**, ciascuno con il suo username e password. Non possono esistere più utenti che si registrano con il medesimo username.

Un utente può partecipare a una **partita** dopo aver selezionato un agente tra quelli disponibili. Di ogni partita, identificata mediante un codice univoco, si vuole tener traccia dell'esito e della mappa in cui viene giocata.

Ogni **agente** ha un nome che lo identifica e dei punti salute.

Un **round** ha una durata massima entro la quale le due squadre devono provare a ottenere la vittoria. La partita viene vinta dalla prima squadra che ottiene tredici vittorie.

All'inizio del round i giocatori procedono all'acquisto delle armi. Ogni **arma** si identifica attraverso un nome e ha una tipologia. Sono ammesse solo due tipologie (primaria e secondaria).

Durante il round, i giocatori possono effettuare delle **uccisioni**. Si vuole tener traccia dell'istante in cui questa avviene, dei due giocatori coinvolti e dell'arma utilizzata.

Ogni giocatore interagisce con la bomba, innescandola o disinnescandola, compatibilmente con il ruolo della squadra a cui appartiene.

Le **mappe** si identificano tramite un nome e si vuole conoscere quali siti questa possa ospitare. Si richiede, inoltre, di tener traccia del sito in cui una bomba viene posizionata durante un round.

Principali azioni richieste

Di seguito sono elencate le funzionalità che l'applicativo e la base di dati devono essere in grado di soddisfare.

A livello più generale, si ha:

1. Iscrizione di un utente
2. Eliminazione di un utente
3. Visualizzazione dell'esito delle partite giocate e loro durata
4. Visualizzazione della percentuale di vittorie in un certo lasso di tempo
5. Top 3 delle armi più utilizzate con rispettivo numero di uccisioni effettuate
6. Top 3 delle mappe più giocate in un certo lasso di tempo

Per quanto riguarda una singola partita:

7. Salvataggio al dettaglio di un round (esito, partecipanti, uccisioni, ...)
8. Conteggio dei round vinti e persi
9. Visualizzazione al dettaglio di un round (esito, partecipanti, uccisioni, ...)
10. Storico delle armi possedute da un giocatore
11. Dettaglio delle eliminazioni avvenute
12. Conteggio delle eliminazioni per ogni arma acquistata da un giocatore

Progettazione concettuale

Schema scheletro

Al seguito di un'analisi del dominio del problema, si è arrivati a produrre il seguente schema scheletro per quanto riguarda una **partita**:

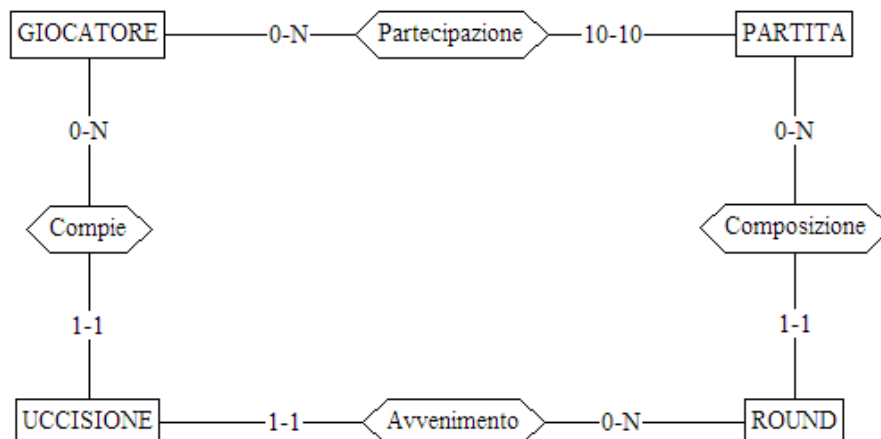


Figura 1 - Schema scheletro per le principali entità di una partita

L'entità **giocatore** racchiude in sé due concetti distinti: il primo risulta essere l'utente che decide di iscriversi alla piattaforma con delle credenziali; il secondo rappresenta l'utente che partecipa a una specifica partita dopo aver scelto un agente. Per questo motivo si decide di modellare questa situazione introducendo le entità **utente** e **agente**.

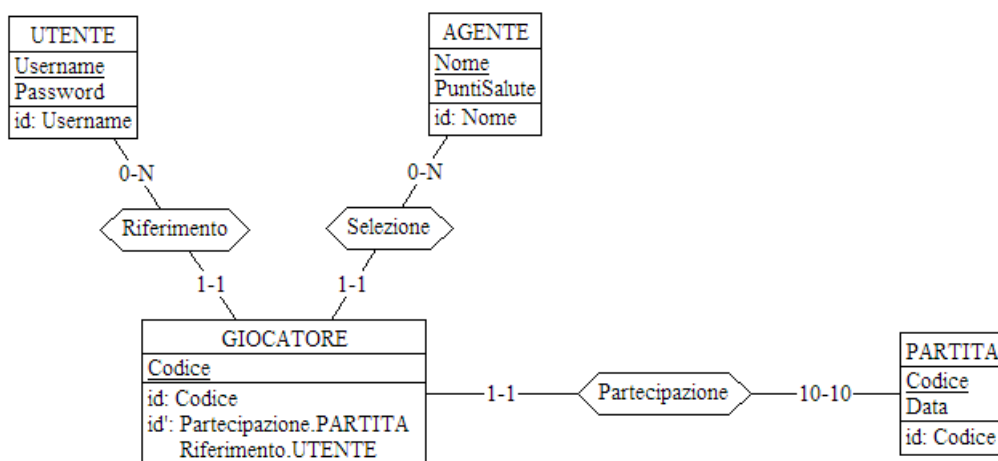


Figura 2 - Raffinamento giocatore

Con il seguente raffinamento è possibile tener traccia di tutti gli utenti registrati attraverso uno username identificativo e una password. Un giocatore viene così identificato dal suo utente e dalla partita a cui prende parte. Per semplificare la gestione successiva e rendere più chiaro lo schema, si è deciso di introdurre come ulteriore identificatore un codice univoco.

Questo raffinamento permette, inoltre, di associare l'agente scelto in una partita. Un **agente** è identificato dal suo nome e possiede un certo numero di punti salute.

È importante notare che a ogni partita prendono parte un totale di dieci giocatori, divisi in due squadre. Al fine di semplificare la gestione e meglio tener traccia di questa suddivisione, è necessario introdurre un attributo per la relazione *partecipazione*.

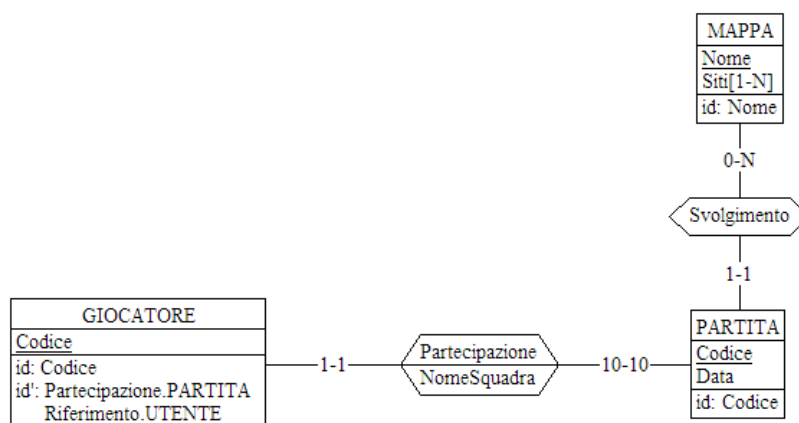


Figura 3 - Introduzione dell'attributo "nome squadra"

L'attributo *nome squadra* ha lo scopo di stabilire in quale delle due squadre faccia parte un determinato giocatore. Tale attributo può assumere qualunque valore, purché ogni partita veda dieci relazioni *partecipazione* con esattamente due valori distinti di questo attributo. Questi controlli verranno eseguiti a livello implementativo al fine di garantire una coerenza con la base di dati.

Infine, come da specifiche, viene introdotta l'entità **mappa** che ha come identificativo il suo nome e tiene traccia di quanti (quali) siti vi sono al suo interno.

Come già enunciato nell'analisi dei requisiti, ogni partita si compone di un certo numero di round, tanti quanti sono necessari per la vittoria di una squadra. Si è reso opportuno introdurre un'entità **round** che lo rappresenti e che renda possibile le successive implementazioni delle specifiche.

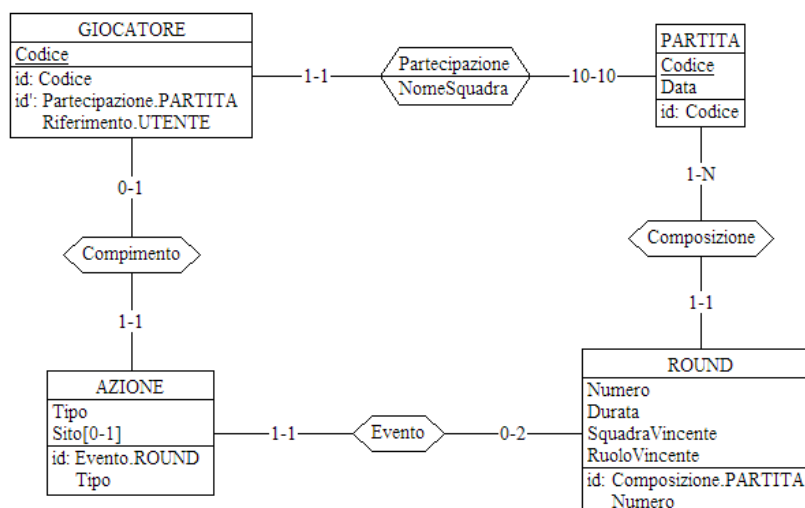


Figura 4 - Introduzione dell'entità round

Un round è unico all'interno di una partita e quindi è naturale che venga identificato dal numero sequenziale dello stesso e dalla sua partita. La durata del round, di cui se ne tiene traccia, verrà impiegata in modo da soddisfare le funzionalità richieste e verificare le condizioni di vittoria.

Gli attributi *squadra vincente* e *ruolo vincente* si rendono necessari rispettivamente per memorizzare la vittoria di una squadra e per tener traccia se la vittoria sia stata ottenuta da una squadra attaccante piuttosto che difendente. Per questo motivo, entrambi gli attributi avranno un dominio ben specifico su cui esisteranno.

Le specifiche ci suggeriscono che è necessario salvare anche gli avvenimenti inerenti alla bomba. Alla luce di ciò si è reso necessario introdurre l'entità **azione**. Questa entità permette di registrare se si tratti di un innesco o di un disinnesco e nel caso della prima ipotesi, anche del sito in cui questo avviene.

Un'azione è identificata dal round in cui avviene e dal suo tipo, in modo tale da ottenere solo azioni diverse all'interno dello stesso round.

Ogni giocatore, per permettere lo svolgimento del gioco, ha la possibilità di compiere uccisioni all'interno di un round.

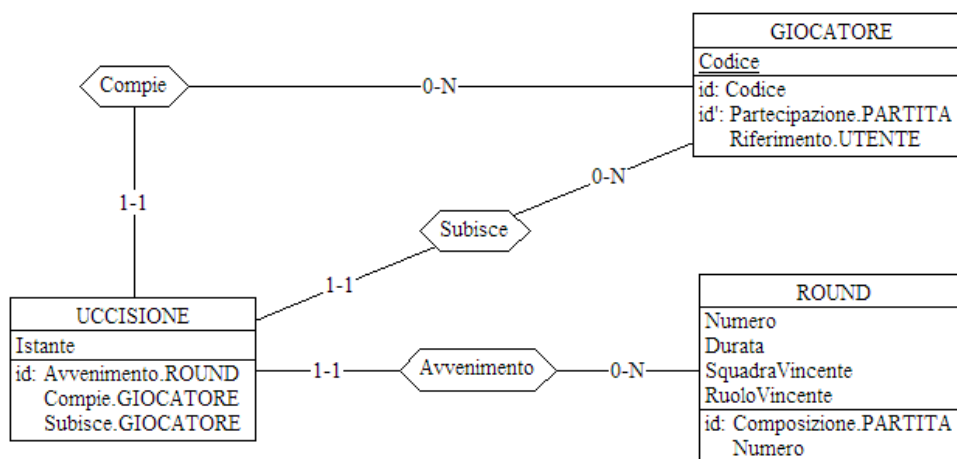


Figura 5 - Entità uccisione

Questa modellazione permette di conservare uno storico delle uccisioni subite e/o effettuate da ogni giocatore. L'attributo *istante* ci permette di risalire al momento preciso in cui questa è avvenuta. Ogni uccisione si identifica dalla seguente terna: giocatore che la compie, giocatore che la subisce e il round in cui avviene.

Le specifiche indicano di voler tenere traccia dell'arma che ha causato un'uccisione e per questo si rende opportuno introdurre l'entità **arma**.

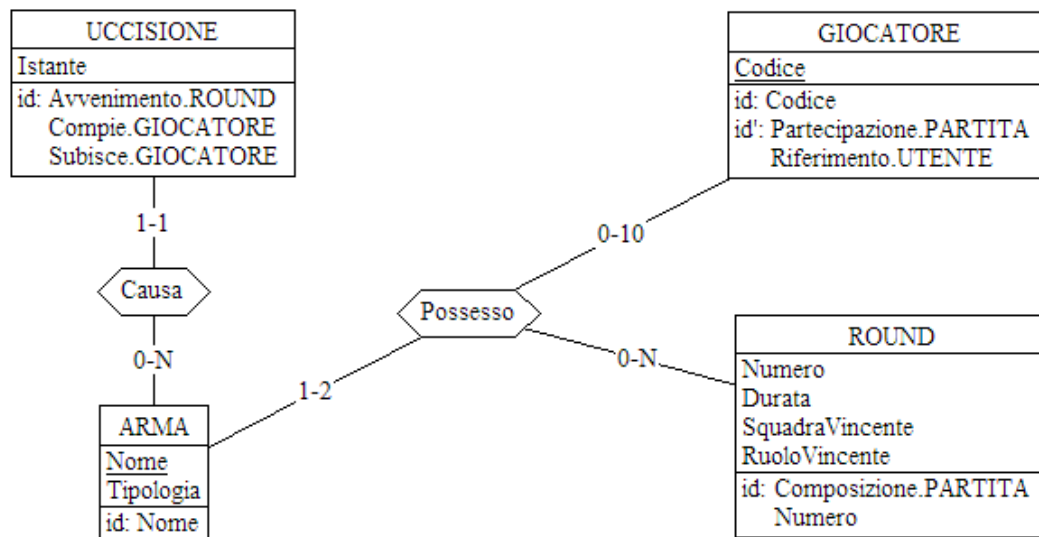


Figura 6 - Entità arma

Questo scheletro vuole mostrare le associazioni che l'entità arma ha nei confronti di round, uccisione e giocatore.

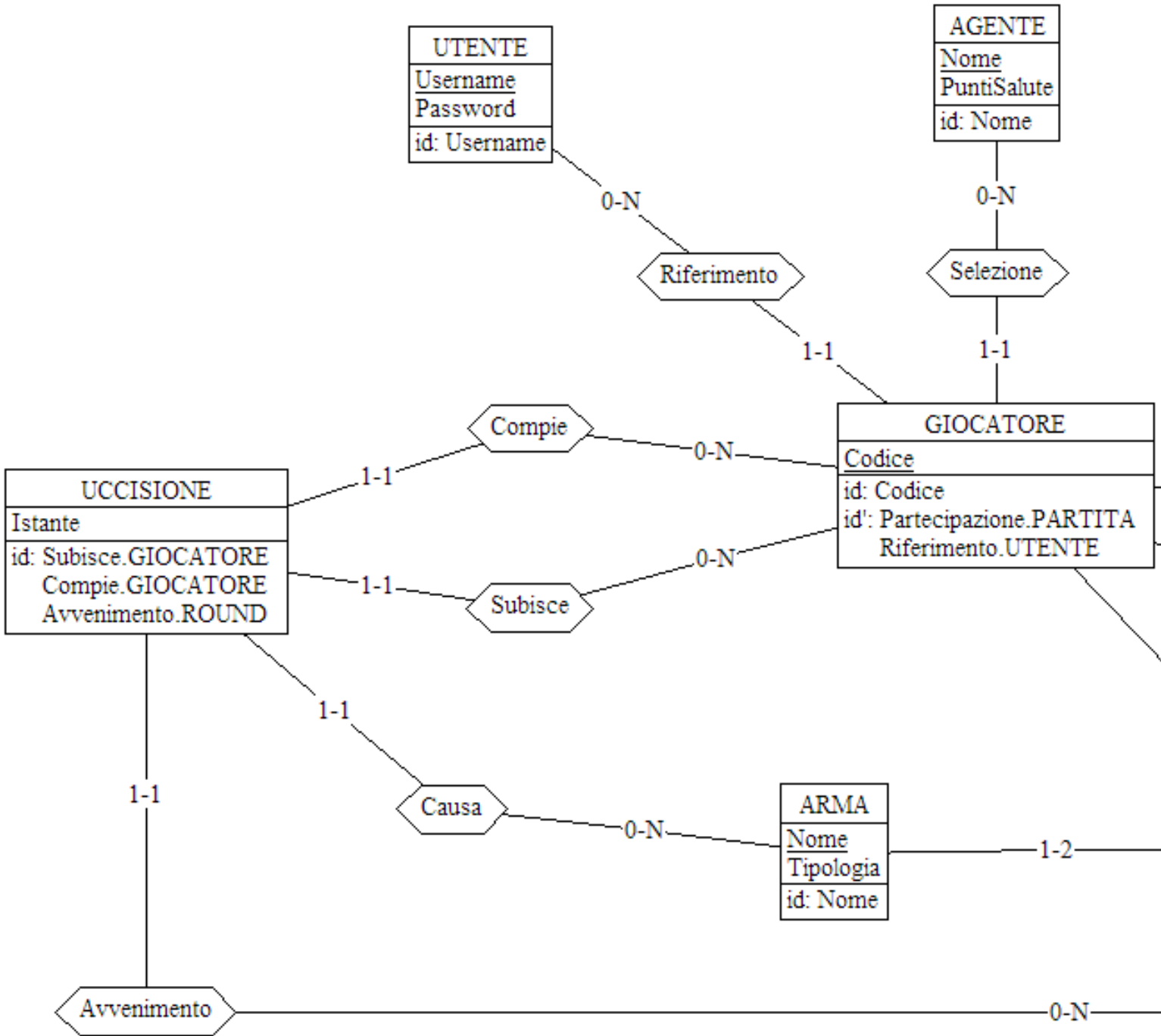
Un'arma viene identificata dal suo nome e può essere di due tipologie distinte. Si renderà opportuno imporre un dominio sull'attributo tipologia.

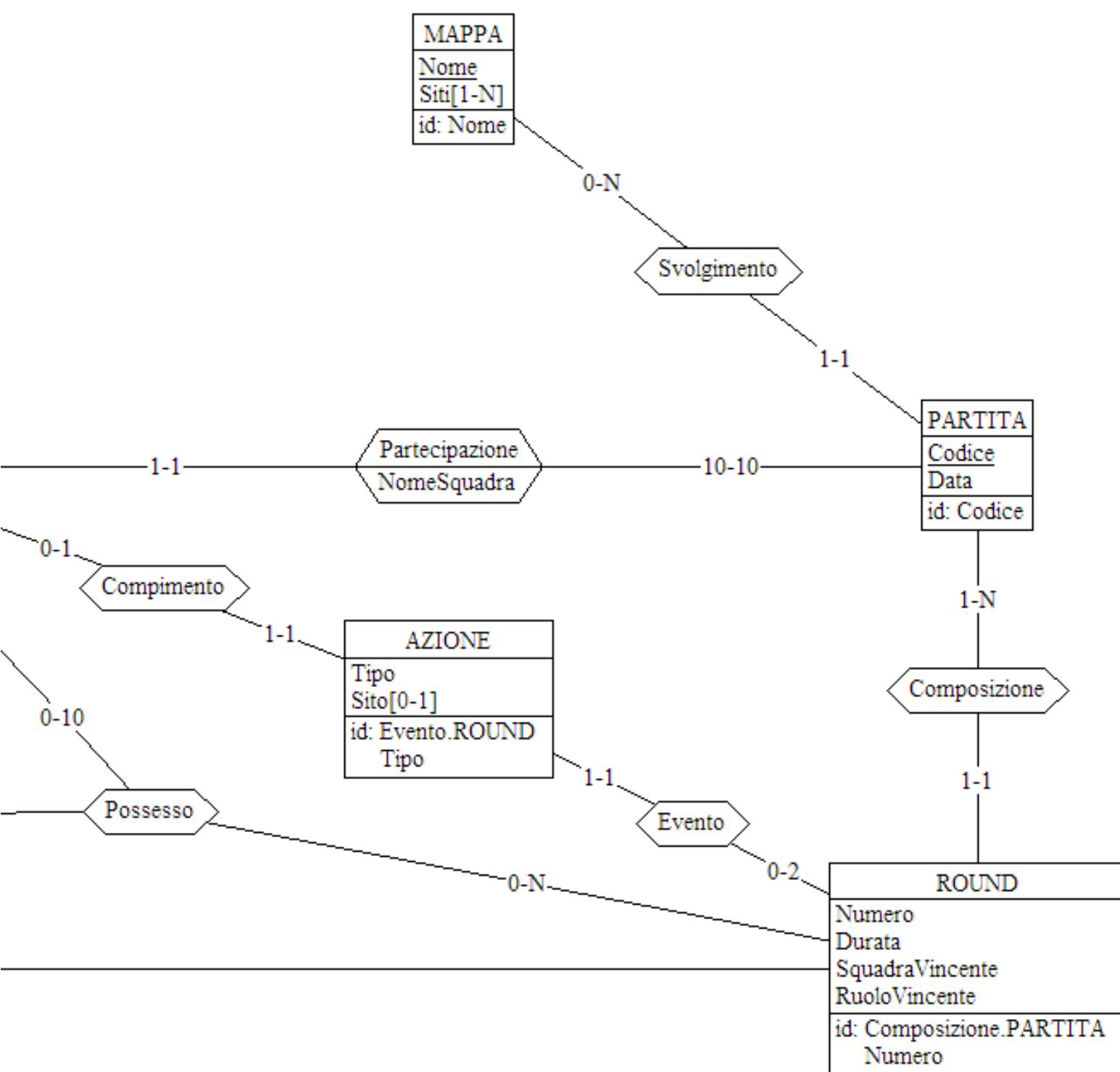
Ogni uccisione è causata da un'arma e la relazione che sussiste ci permette, appunto, di risalire a chi l'ha effettuata.

Una funzionalità richiesta è quella di sapere quali sono state le armi acquistate da un giocatore durante i vari round di una partita. Con questa modellazione, un giocatore, all'interno di un round, può possedere al massimo due armi. È richiesta una verifica esterna affinché siano di tipologie differenti.

Si noti il fatto che, in questo caso, l'associazione *possesso* è effettivamente un'associazione ternaria dal momento che le cardinalità massime sono tutte diverse da 1.

Schema concettuale finale





Progettazione logica

Stima del volume dei dati

Concetto	Costrutto	Volume
Utente	E	10.000
Agente	E	20
Mappa	E	10
Uccisione	E	6.000.000
Giocatore	E	1.000.000
Partita	E	100.000
Arma	E	20
Azione	E	3.000.000
Round	E	2.000.000
Riferimento	R	1.000.000
Selezione	R	1.000.000
Svolgimento	R	100.000
Compie	R	6.000.000
Subisce	R	6.000.000
Partecipazione	R	1.000.000
Compimento	R	3.000.000
Causa	R	6.000.000
Possesso	R	20.000.000
Evento	R	3.000.000
Composizione	R	2.000.000
Avvenimento	R	6.000.000

Operazioni principali e stima della loro frequenza

In riferimento a quanto già elencato per le operazioni principali da svolgere, viene riportata di seguito una tabella che ne indica la loro frequenza:

N°	Operazione	Frequenza
1	Iscrizione di un utente	2 al giorno
2	Eliminazione di un utente	1 al mese
3	Visualizzazione dell'esito delle partite giocate e loro durata	100 al giorno
4	Visualizzazione della percentuale di vittorie in un certo lasso di tempo	10 al giorno
5	Top 3 delle armi più utilizzate con rispettivo numero di uccisioni effettuate	10 a settimana
6	Top 3 delle mappe più giocate in un certo lasso di tempo	10 a settimana
7	Salvataggio al dettaglio di un round	500 al giorno
8	Conteggio dei round vinti e persi	100 al giorno
9	Visualizzazione al dettaglio di un round	500 al giorno
10	Storico delle armi possedute da un giocatore	10 al giorno
11	Visualizzazione degli esiti degli scontri diretti	100 al giorno
12	Conteggio delle eliminazioni per ogni arma acquistata da un giocatore	20 al giorno

Schemi di navigazione e tabelle degli accessi

Si riportano in seguito le tabelle relative agli accessi di ciascuna operazione espressa in precedenza.

Operazione 1 – Iscrizione di un utente

Concetto	Costrutto	Accessi	Tipo
Utente	E	1	S
Totale: 1S → 4 al giorno			

Operazione 2 – Eliminazione di un utente

Concetto	Costrutto	Accessi	Tipo
Utente	E	1	S
Totale: 1S → 2 al mese			

Operazione 3 – Esito e durata delle partite

Per ricavare l'esito delle partite è necessario stabilire di quale squadra un giocatore faccia parte. Sulla base di ciò è possibile determinare l'esito di un round. La durata totale di una partita si ricava come somma dei singoli round che la compongono.

Concetto	Costrutto	Accessi	Tipo
Giocatore	E	100	L
Partecipazione	R	100	L
Partita	E	100	L
Composizione	R	2.000	L
Round	E	2.000	L
Totale: 4.300L → 430.000 al giorno			

Operazione 4 – Percentuale delle vittorie

Questa operazione ha bisogno di accedere agli stessi dati dell'operazione 3. La query che ricaverà l'informazione necessaria risulterà essere differente.

Concetto	Costrutto	Accessi	Tipo
Giocatore	E	100	L
Partecipazione	R	100	L
Partita	E	100	L
Composizione	R	2.000	L
Round	E	2.000	L
Totale: 4.300L → 43.000 al giorno			

Operazione 5 – Top 3 delle armi

Si controllano tutti i possessi di un utente per stabilire una classifica delle armi più utilizzate. Successivamente si ricavano le rispettive uccisioni.

Concetto	Costrutto	Accessi	Tipo
Giocatore	E	100	L
Possesso	R	2.000	L
Arma	E	2.000	L
Causa	R	2.000	L
Uccisione	E	2.000	L
Compie	R	2.000	L
Totale: 10.100L → 101.000 a settimana			

Operazione 6 – Top 3 delle mappe

Similmente si procede come per l'operazione precedente.

Concetto	Costrutto	Accessi	Tipo
Giocatore	E	100	L
Partecipazione	R	100	L
Partita	E	100	L
Svolgimento	R	100	L
Mappa	E	100	L
Totale: 500L → 5.000 a settimana			

Operazione 7 – Salvataggio di un round

Il salvataggio di un round è un'operazione complessa che richiede di unire differenti informazioni: le azioni avvenute, le armi acquistate dai giocatori, le uccisioni effettuate.

Per avere una stima più semplice e chiara, verrà fatta un'analisi per ciascuna di queste componenti.

Si darà per assunto che esista già un round. Nella realtà, questo aspetto verrà gestito a livello implementativo.

Operazione 7.1 – Salvataggio di un'azione

Concetto	Costrutto	Accessi	Tipo
Round	E	1	L
Evento	R	1	S
Azione	E	1	S
Compimento	R	1	S
Giocatore	E	1	L
Totale: 2L+3S → 6.000 al giorno			

Operazione 7.2 – Acquisto di un'arma

Concetto	Costrutto	Accessi	Tipo
Round	E	1	L
Possesso	R	1	S
Arma	E	1	L
Giocatore	E	1	L
Totale: 3L+1S → 25.000 al giorno			

Operazione 7.3 – Salvataggio di un'uccisione

Concetto	Costrutto	Accessi	Tipo
Round	E	1	L
Avvenimento	R	1	S
Uccisione	E	1	S
Compie	R	1	S
Subisce	R	1	S
Giocatore	E	2	L
Causa	R	1	S
Arma	E	1	L
Totale: 4L+5S → 21.000 al giorno			

Come si può notare, l'operazione di salvataggio di un round una delle più complesse. Al fine di stimare quante scritture sono necessarie, si fa riferimento alla tabella dei volumi. Tale operazione prevede un costo totale di 52.000 accessi al giorno.

Operazione 8 – Conteggio dei round vinti e persi

Concetto	Costrutto	Accessi	Tipo
Round	E	20	L
Totale: 20L → 2.000 al giorno			

Operazione 9 – Dettaglio di un round

Di un round si vogliono sapere tutte le uccisioni effettuate, con quale arma e le interazioni avute con la bomba.

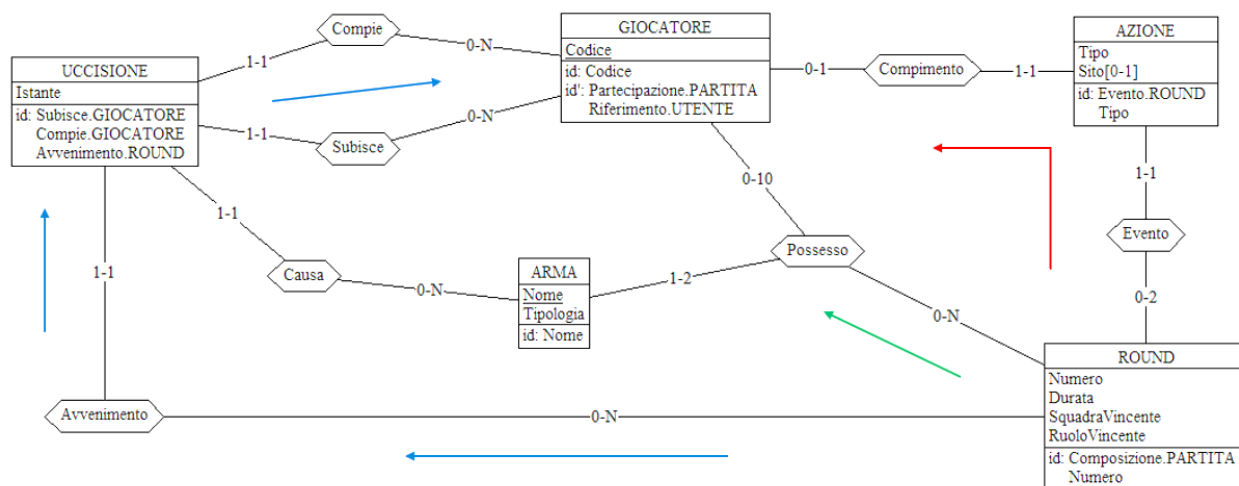


Figura 7 - Mappa di navigazione operazione 9

Concetto	Costrutto	Accessi	Tipo
Round	E	1	L
Evento	R	1,5	L
Azione	E	1,5	L
Compimento	R	1,5	L
Giocatore	E	1,5 + 10 + 6 (17,5)	L
Possesso	R	10	L
Arma	E	10 + 3 (13)	L
Causa	R	3	L
Uccisione	E	3	L
Compie	R	3	L
Subisce	R	3	L
Avvenimento	R	3	L
Totale: 61L → 30.500 al giorno			

Per quanto riguarda le entità *giocatore* e *arma*, i loro accessi sono calcolati in relazione alle uccisioni e agli acquisti effettuati durante un round.

Operazione 10 – Storico delle armi possedute da un giocatore

Concetto	Costrutto	Accessi	Tipo
Possesso	R	20	L
Arma	E	20	L
Totale: 40L → 400 al giorno			

Operazione 11 – Visualizzazione degli scontri diretti

Con questa operazione si vuole ottenere una lista di tutti gli scontri avvenuti tra un giocatore e i suoi avversari.

Concetto	Costrutto	Accessi	Tipo
Giocatore	E	12	L
Compie	R	12	L
Subisce	R	12	L
Uccisione	E	12	L
Totale: 48L → 4.800 al giorno			

In questo caso si considera che un giocatore effettua in media sei uccisioni e viene ucciso mediamente lo stesso numero di volte.

Operazione 12 – Conteggio eliminazioni per ogni arma

Concetto	Costrutto	Accessi	Tipo
Compie	R	6	L
Uccisione	E	6	L
Causa	R	6	L
Arma	E	6	L
Possesso	R	20	L
Totale: 44L → 880 al giorno			

Raffinamento dello schema

Eliminazione degli attributi composti e multipli

Nello schema non sono presenti attributi composti. Situazione diversa è data dall'entità **mappa** che contiene l'attributo multiplo *Siti*. Dal momento che ogni mappa presenta un numero variabile di siti di cui non è necessario differenziarli in modo particolare, si procede a salvare direttamente il numero di siti di una mappa. Sarà poi gestita a livello implementativo la visualizzazione degli stessi.

Eliminazione delle gerarchie

Nello schema non è presente alcuna gerarchia.

Scelta delle chiavi primarie

Per la maggior parte delle entità dello schema è già presente una chiave primaria che le identifichi. Tuttavia, l'entità **giocatore** presenta due chiavi candidate. Si decide, quindi, di identificarla attraverso l'attributo Codice.

Eliminazione degli identificatori esterni

Dallo schema prodotto sono eliminate le seguenti relazioni, apportando le relative modifiche:

- *Riferimento*, importando Username in **Giocatore** (UsernameUtente)
- *Selezione*, importando Nome in **Giocatore** (NomeAgente)
- *Svolgimento*, importando Nome in **Partita** (NomeMappa)
- *Partecipazione*, importando Codice in **Giocatore** (CodicePartita) e allo stesso modo l'attributo della relazione (NomeSquadra).
- *Compimento*, importando Codice in **Azione** (CodiceGiocatore)
- *Composizione*, importando Codice in **Round** (CodicePartita)
- *Evento*, importando CodicePartita e Numero in **Azione** (CodicePartitaRound, NumeroRound)
- *Compie*, importando Codice in **Uccisione** (CodiceGiocatoreC)
- *Subisce*, importando Codice in **Uccisione** (CodiceGiocatoreS)

- *Avvenimento*, importando CodicePartita e Numero in **Uccisione** (CodicePartitaRound, NumeroRound)
- *Causa*, importando Nome in **Uccisione** (NomeArma)
- *Possesso*, reificata importando Codice da **Giocatore** (CodiceGiocatore); CodicePartita e Numero da **Round** (CodicePartitaRound, NumeroRound); Nome da **Arma** (NomeArma)

Analisi delle ridondanze

Considerato l'elevato numero di richieste dell'operazione 3, si prende in considerazione la possibilità di aggiungere degli attributi ridondanti in **Partita**. Questi attributi sono: DurataTotale, SquadraVincente, RoundTotali e RoundVinti. Quest'ultimo attributo farà riferimento al risultato ottenuto dalla squadra vincente. Viene mostrato di seguito il risultato di questa possibile modifica.

Operazione 3 – Esito e durata delle partite

La ridondanza in questo caso è utile in quanto non è più necessario effettuare accessi a **Round** e **Composizione**. Per cui:

Concetto	Costrutto	Accessi	Tipo
Giocatore	E	100	L
Partecipazione	R	100	L
Partita	E	100	L
Totale: 300L → 30.000 al giorno			

Operazione 4 – Percentuale delle vittorie

Similmente, si ha un accesso diretto alle informazioni necessarie per soddisfare questa operazione.

Concetto	Costrutto	Accessi	Tipo
Giocatore	E	100	L
Partecipazione	R	100	L
Partita	E	100	L
Totale: 300L → 3.000 al giorno			

Operazione 7 – Salvataggio di un round

Ogni volta viene salvato un round, è necessario aggiornare coerentemente gli attributi ridondanti appena introdotti. L'operazione si semplifica come segue:

Concetto	Costrutto	Accessi	Tipo
Partita	E	20	L
Partita	E	20	S
Composizione	R	20	S
Totale: 100L → 50.000 al giorno			

Le operazioni **7.1**, **7.2** e **7.3** rimangono invariate.

Operazione 8 – Conteggio dei round vinti e persi

Concetto	Costrutto	Accessi	Tipo
Partita	E	1	L
Totale: 1L → 100 al giorno			

Si può notare che a seguito dell'introduzione di questa ridondanza, le operazioni interessate verranno effettuate con 135.100 accessi al giorno.

Precedentemente, le stesse operazioni, producevano un totale di 527.000 accessi al giorno.

Alla luce di quanto appena detto, si decide di introdurre questa ridondanza.

Traduzione di entità e associazioni in relazioni

utenti(username, password*)

agenti(nome, puntiSalute)

giocatori(codice, usernameUtente: utenti, codicePartita: partite, nomeAgente: agenti, nomeSquadra)

UNIQUE(usernameUtente, codicePartita)

partite(codice, data, nomeMappa: mappe, durataTotale*, squadraVincente*, roundTotali*, roundVinti*)

mappe(nome, numeroSiti)

round(codicePartita: partite, numero, durata, squadraVincente, ruoloVincente)

azioni((codicePartitaRound, numeroRound): round, tipo, codiceGiocatore: giocatori, sito*)

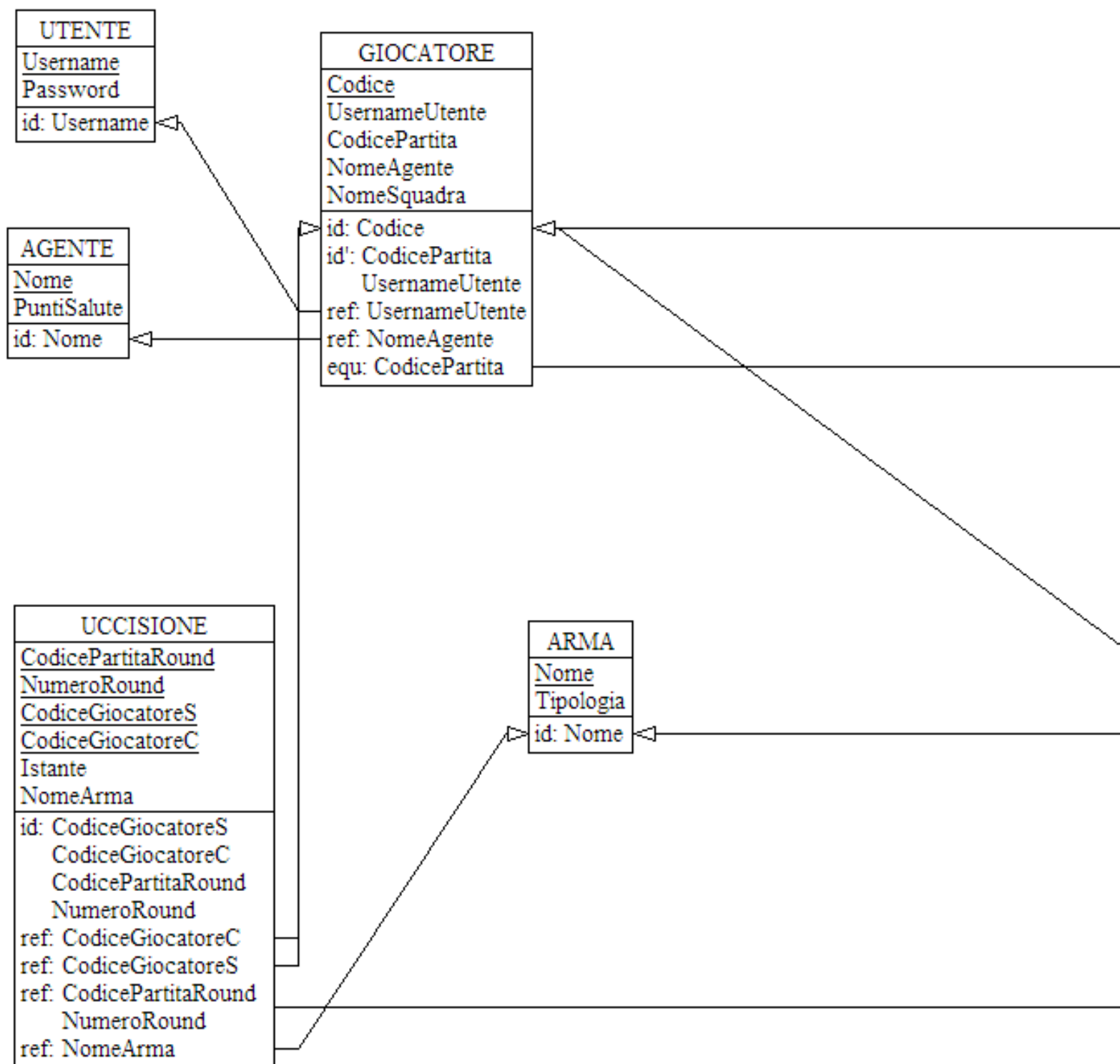
uccisioni((codicePartitaRound, numeroRound): round, (codiceGiocatoreS, codiceGiocatoreC): giocatori, istante, nomeArma: armi)

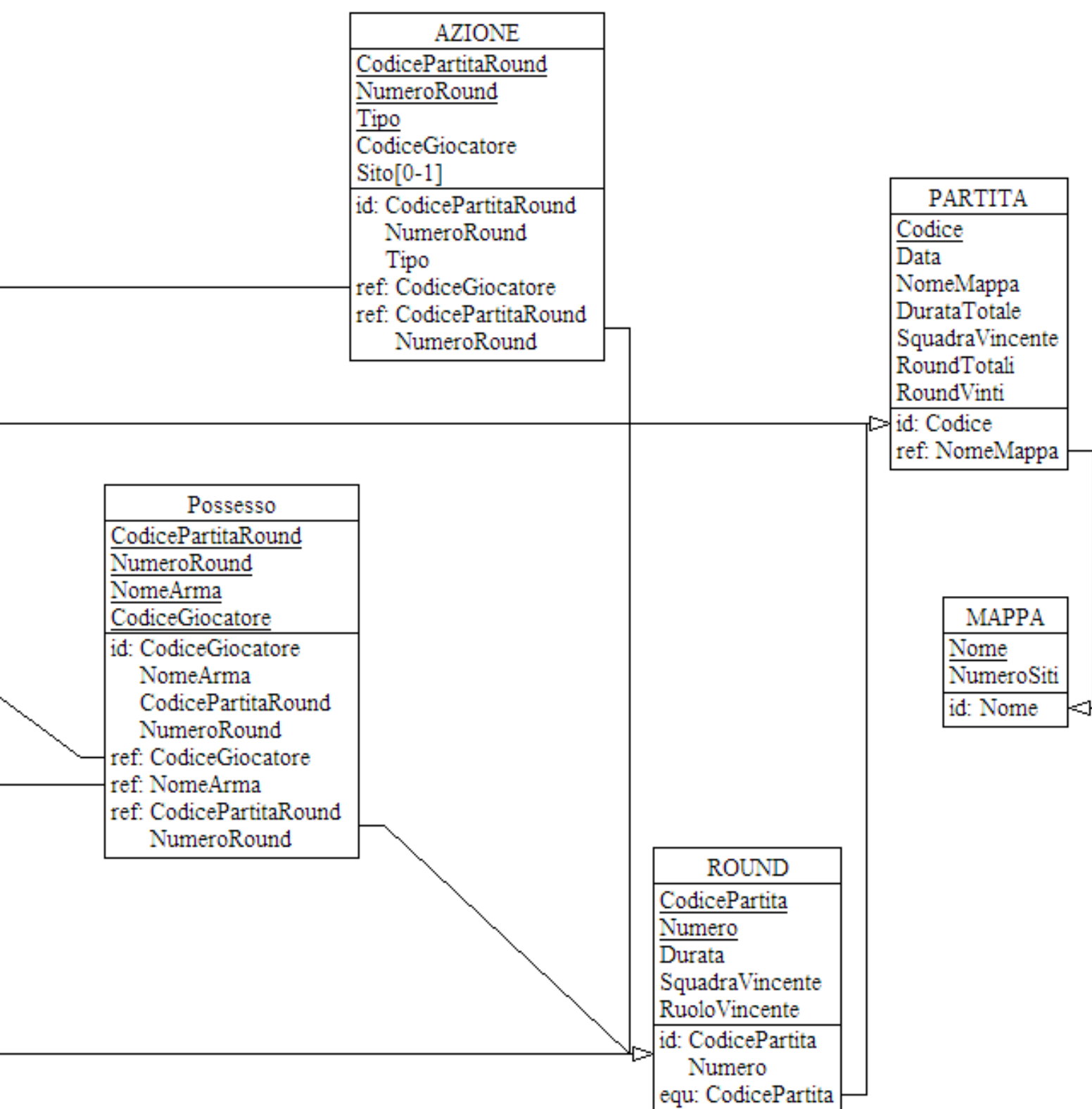
armi(nome, tipologia)

possessi((codicePartitaRound, numeroRound): round, nomeArma: armi, codiceGiocatore: giocatori)

Questa pagina è stata lasciata deliberatamente vuota. Segue.

Schema relazionale finale





Traduzione delle operazioni in query SQL

Creazioni delle viste

Per semplificare le query delle operazioni successive, si decide di creare delle viste di supporto.

```
1. CREATE VIEW partite_utente AS
   SELECT g.Codice AS CodiceG, g.UsernameUtente,
          g.NomeSquadra, g.CodicePartita, g.NomeAgente, p.*
   FROM partite p, giocatori g
   WHERE p.Codice = g.CodicePartita
```

Operazione 1 – Iscrizione di un utente

```
INSERT INTO utenti (username, password) VALUES (?, ?)
```

Operazione 2 – Eliminazione di un utente

Un utente quando richiede l'eliminazione non deve essere più in grado di accedere al portale. Invalidando il login si ottiene questo effetto mantenendo le informazioni richieste.

```
UPDATE utenti SET password = null WHERE username = ?
```

Operazione 3 – Esito e durata delle partite

Si vuole visualizzare l'elenco di tutte le partite giocate da un utente con il loro esito (vittoria o sconfitta) e la loro durata totale. Per questo scopo viene utilizzata la vista *partite_utente*.

```
SELECT Codice, Data, NomeMappa, DurataTotale,
   IF(SquadraVincente IS NULL, 'Pareggio',
      IF(SquadraVincente = NomeSquadra, 'Vittoria', 'Sconfitta')
  ) AS Esito
FROM partite_utente
WHERE Username = ?
```

Operazione 4 – Percentuale delle vittorie

Come per l'operazione 3, ci si appoggia alla vista *partite_utente*.

```
SELECT COUNT(*) * 100.0 / (SELECT COUNT(*) FROM partite_utente
   WHERE UsernameUtente = ? AND CAST(Data AS DATE)
   BETWEEN ? AND ?) AS Percentuale
FROM partite_utente
WHERE UsernameUtente = ?
   AND CAST(Data AS DATE) BETWEEN ? AND ?
   AND NomeSquadra = SquadraVincente
```


Operazione 5 – Top 3 delle armi

Si vuole stilare una classifica delle armi più utilizzate mostrando anche il numero delle uccisioni che hanno causato.

```
SELECT u.NomeArma, Utilizzi, COUNT(u.NomeArma) AS
    TotaleUccisioni
FROM uccisioni u, giocatori g
JOIN (
    SELECT p.NomeArma, COUNT(p.NomeArma) AS Utilizzi
    FROM giocatori g, possessi p
    WHERE g.UsernameUtente = ? AND g.Codice = p.CodiceGiocatore
    GROUP BY p.NomeArma
    ORDER BY Utilizzi DESC LIMIT 3
) AS top3
WHERE g.UsernameUtente = ?
    AND u.CodiceGiocatoreC = g.Codice
    AND u.NomeArma = top3.NomeArma
GROUP BY u.NomeArma
ORDER BY Utilizzi DESC
```

Operazione 6 – Top 3 delle mappe di un periodo

```
SELECT p.NomeMappa, COUNT(*) As Totale
FROM partite p, giocatori g
WHERE g.UsernameUtente = ?
    AND g.CodicePartita = p.Codice
    AND CAST(p.Data AS DATE) BETWEEN ? AND ?
GROUP BY p.NomeMappa
ORDER BY Totale DESC LIMIT 3
```

Operazione 7 – Salvataggio di un round

Come già indicato nelle tabelle degli accessi, questa operazione è possibile grazie all'esecuzione di tre operazioni distinte che corrispondono alle possibili fasi che avvengono durante un round. Di seguito sono mostrate le relative query SQL.

Operazione 7.0 – Creazione di un round

Ogni partita è sempre composta da almeno un round e quindi si procede a crearlo, contestualmente all'inizio di una partita. La stessa query verrà impiegata all'inizio di ogni round per favorire le operazioni successive.

```
INSERT INTO round (CodicePartita, Numero, Durata,
    SquadraVincente, RuoloVincente) VALUES (?, ?, ?, ?, ?)
```

Operazione 7.1 – Salvataggio di un'azione

```
INSERT INTO azioni (CodicePartitaRound, NumeroRound, Tipo,  
CodiceGiocatore, Sito) VALUES (?, ?, ?, ?, ?)
```

Operazione 7.2 – Acquisto di un'arma

```
INSERT INTO possessi (CodicePartitaRound, NumeroRound,  
NomeArma, CodiceGiocatore) VALUES (?, ?, ?, ?)
```

Operazione 7.3 – Salvataggio di un'uccisione

```
INSERT INTO uccisioni (CodicePartitaRound, NumeroRound,  
CodiceGiocatoreS, CodiceGiocatoreC, Istante, NomeArma)  
VALUES (?, ?, ?, ?, ?, ?)
```

Operazione 8 – Conteggio dei round vinti e persi

```
SELECT Codice, SquadraVincente, RoundVinti,  
RoundTotali - RoundVinti AS RoundPersi  
FROM partite  
WHERE Codice = ?
```

Operazione 9 – Dettaglio di un round

Come per l'operazione 7, è conveniente suddividere le varie informazioni che si vogliono ricavare attraverso delle query più semplici. Il livello applicativo si occuperà di presentare all'utente queste informazioni in modo coerente.

Operazione 9.0 – Dettaglio generale

```
SELECT Numero, Durata, IF(SquadraVincente = ?, 'Vittoria',  
'Sconfitta') AS Esito  
FROM round  
WHERE CodicePartita = ? AND Numero = ?
```

Operazione 9.1 – Elenco delle azioni

```
SELECT g.NomeAgente, g.NomeSquadra, a.Tipo, a.Sito  
FROM giocatori g, azioni a  
WHERE a.CodicePartitaRound = ? AND a.NumeroRound = ?  
AND a.CodiceGiocatore = g.Codice
```

Operazione 9.2 – Elenco delle uccisioni

```
SELECT u.Istante, CONCAT(gc.NomeAgente, gc.NomeSquadra) AS  
CausataDa, CONCAT(gs.NomeAgente, gs.NomeSquadra) AS SubitaDa,  
u.NomeArma FROM giocatori gc, giocatori gs, uccisioni u  
WHERE u.CodicePartitaRound = ? AND u.NumeroRound = ?  
AND u.CodiceGiocatoreC = gc.Codice  
AND u.CodiceGiocatoreS = gs.Codice  
ORDER BY u.Istante
```

Operazione 10 – Storico delle armi

Noto il giocatore e la partita di cui si vuole ricavare questa informazione, si procede con la seguente query:

```
SELECT p.NumeroRound, p.NomeArma, a.Tipologia
FROM possessi p, armi a
WHERE p.CodicePartitaRound = ?
      AND p.CodiceGiocatore = ?
      AND p.NomeArma = a.Nome
```

Operazione 11 – Visualizzazione degli scontri diretti

Similmente all'operazione 10 e rifacendosi all'operazione 9.2, si ottiene la seguente query:

```
SELECT u.NumeroRound, IF(gc.Codice = ?, gs.NomeAgente,
      gc.NomeAgente) AS Contro,
      IF(gc.Codice = ?, 'Vinto', 'Perso') AS Esito, u.NomeArma
FROM giocatori gc, giocatori gs, uccisioni u
WHERE u.CodicePartitaRound = ?
      AND (gc.Codice = ? OR gs.Codice = ?)
      AND u.CodiceGiocatoreC = gc.Codice
      AND u.CodiceGiocatoreS = gs.Codice
```

Operazione 12 – Conteggio delle eliminazioni per ogni arma

```
SELECT p.NomeArma, COUNT(u.NumeroRound) AS TotaleUccisioni
FROM possessi p LEFT JOIN uccisioni u
ON u.CodicePartitaRound = p.CodicePartitaRound
   AND u.NumeroRound = p.NumeroRound
   AND u.CodiceGiocatoreC = p.CodiceGiocatore
   AND u.NomeArma = p.NomeArma
WHERE p.CodicePartitaRound = ? AND p.CodiceGiocatore = ?
GROUP BY p.NomeArma
```

Progettazione dell'applicazione

Descrizione dell'applicazione realizzata

Si è deciso di realizzare un applicativo web che permetta di consultare i dati e generarne di nuovi. Tale applicativo è stato realizzato attraverso un web server PHP che risiede localmente, così come il database MySQL a cui si connette.

Il database deve avere il nome elaborato e deve esistere l'utente utente1 con password utente1! con i permessi di scrittura e lettura, necessari al fine di garantire una corretta operatività. I dati necessari al funzionamento sono presenti nello script di inizializzazione fornito.

Login

Username: Password:

[Clicca qui per iscriverti](#)

Torna [indietro](#) oppure [alla home](#)

Figura 8 - Schermata di login

Una volta eseguito il login all'utente è permesso di consultare le partite svolte o simulare una nuova partita. Alternativamente, può effettuare il logout oppure disiscriversi dalla piattaforma.

Storico delle partite giocate

Codice	Data	NomeAgente	NomeMappa	DurataTotale	Esito	Dettaglio
1	2023-07-26 15:48:09	Breach	Icebox	01:02:14	Vittoria	link
2	2023-07-26 15:48:11	Breach	Ascent	00:59:39	Sconfitta	link
3	2023-07-26 15:48:23	Viper	Icebox	00:12:22	Vittoria	link
4	2023-07-26 15:48:27	Yoru	Icebox	00:04:25	Pareggio	link
5	2023-07-26 15:48:29	Jett	Ascent	00:20:24	Vittoria	link
6	2023-07-26 15:48:31	Breach	Bind	00:56:33	Vittoria	link
7	2023-07-26 15:48:33	Astra	Fracture	00:45:07	Sconfitta	link
8	2023-07-26 15:48:37	Jett	Ascent	00:02:30	Sconfitta	link
9	2023-07-26 15:48:38	Killjoy	Ascent	00:58:35	Sconfitta	link
10	2023-07-26 15:48:43	Reyna	Ascent	01:00:33	Vittoria	link
11	2023-07-26 15:49:06	Breach	Fracture	00:08:37	Vittoria	link

Funzionalità aggiuntive

- [Operazione 4](#) - Percentuale vittorie
- [Operazione 5](#) - Top 3 armi più utilizzate
- [Operazione 6](#) - Top 3 mappe più giocate

Torna [indietro](#) oppure [alla home](#)

Figura 9 - Storico delle partite

Nella schermata dello storico delle partite possono essere effettuate le operazioni 3, 4, 5, 6 relative a tutte le partite giocate dall'utente. Le rimanenti operazioni sono possibili consultando il dettaglio di una partita.

Partita 3 - Esito: 3-2

Composizione squadre

UsernameUtente	NomeAgente	NomeSquadra
user11	Astra	Alpha
user8	Breach	Alpha
user7	Phoenix	Alpha
user13	Reyna	Alpha
user1	Viper	Alpha
user6	Astra	Beta
user9	Breach	Beta
user2	Phoenix	Beta
user12	Reyna	Beta
user3	Yoru	Beta

Elenco dei round

Numero	Durata	Ruolo	Esito	ArmaPrimaria	ArmaSecondaria	Dettaglio
1	00:02:19	Attacco	Vittoria	Vandal	Sheriff	link
2	00:02:58	Attacco	Sconfitta	Operator	Sheriff	link
3	00:03:00	Attacco	Sconfitta	Odin	Sheriff	link
4	00:02:23	Attacco	Vittoria	Odin	Sheriff	link
5	00:01:42	Attacco	Vittoria	Odin	Sheriff	link

Funzionalità aggiuntive

- [Operazione 11](#) - Scontri diretti
- [Operazione 12](#) - Conteggio eliminazioni per arma

Torna [indietro](#) oppure alla [home](#)

Figura 10 - Dettaglio di una partita

È possibile ricavare il dettaglio di ogni round in modo tale da sapere quali scontri sono avvenuti ed eventuali azioni compiute dai giocatori.

Partita 3 - Round 1

Uccisioni

Istante	CausataDa	SubitaDa	NomeArma
00:00:16	Viper (Alpha)	Breach (Beta)	Sheriff
00:00:36	Reyna (Beta)	Phoenix (Alpha)	Shorty
00:01:05	Reyna (Alpha)	Reyna (Beta)	Ghost
00:01:23	Astra (Beta)	Reyna (Alpha)	Odin
00:01:49	Yoru (Beta)	Astra (Alpha)	Vandal

Avvenimenti

NomeAgente	NomeSquadra	Tipo	Sito
Viper	Alpha	Innesco	A

Torna [indietro](#) oppure alla [home](#)

Figura 11 - Dettaglio di un round

L'applicazione permette di eseguire tutte le funzionalità richieste emerse nella fase di progettazione.

Al fine di rendere più fruibile e comprensibile la lettura dei dati, questi ultimi sono stati manipolati anche dal livello applicativo. Inoltre, sono state implementate funzionalità secondarie come la visualizzazione dei membri di una partita nelle relative squadre, insieme a tutte le funzioni necessarie per la simulazione casuale di una partita.