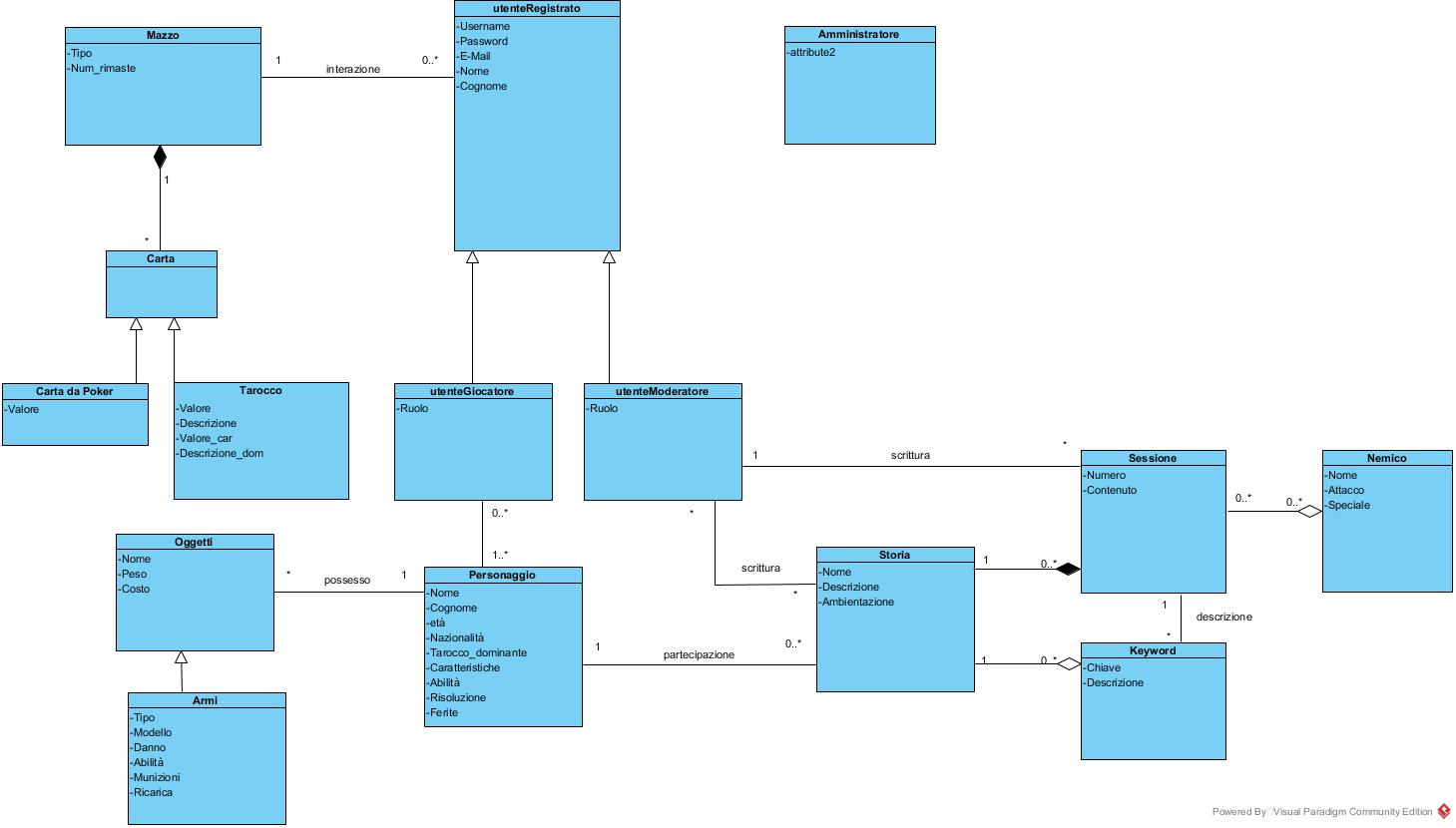
3.4 GESTIONE DEI DATI PERSISTENTI

La memorizzazione dei dati è stata gestita con l’uso di un DataBase di tipo relazionale, in quanto, oltre allo spazio di archiviazione richiesto, i database consentono di ottenere un veloce tempo di risposta (query di ricerca), garantendo una gestione multiutente (multiutente) ed una gestione tramite DBMS.  
Questa scelta è stata fortemente pensata data la sicurezza offerta da un DBMS di ultima generazione, insieme ad una maggiore affidabilità, c’è la garanzia di coerenza, facilità di gestione, nonché dalla velocità di accesso e trasmissione dei dati.  
Riportiamo in seguito lo schema generale e successivamente le singole tabelle con i relativi campi ed una breve descrizione. Infine, prima di questi vi è il class diagram generale riproposto nel RAD con una breve descrizione delle corrispondenze che hanno portato alla creazione del database.

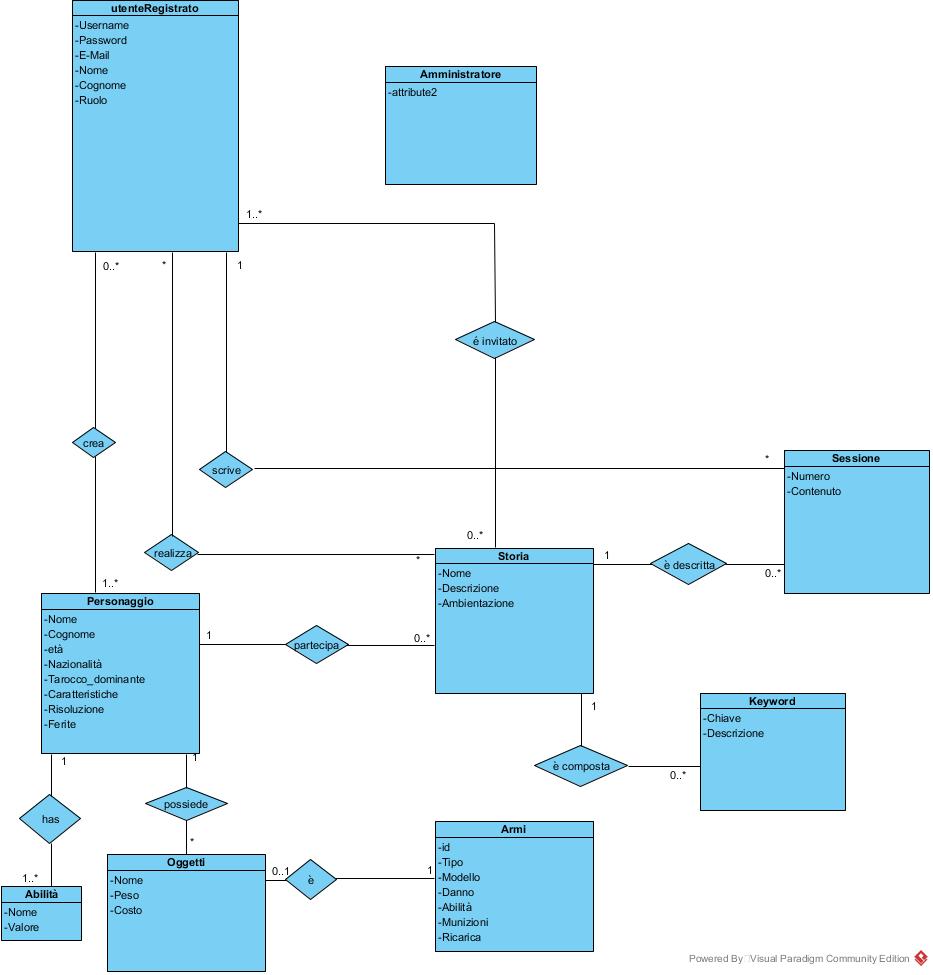


* La classe utente, che può rappresentare o un utenteGiocatore o un utenteModeratore, è rappresentato dalla tabella utenteRegistrato, dove il ruolo ne chiarisce la tipologia. Nella tabella sono. memorizzare le informazioni dell’utente relative a username, password, e-mail, nome e cognome; queste sono memorizzate nella stessa tabella utenteRegistrato associato ad ogni utenteGiocatore o utenteModeratore.
* Una Storia è scritta da un utenteModeratore la quale è composta da diverse Sessioni e da alcune keyword. Essa è composta da un nome, da una descrizione e dall’ambientazione in cui viene svolta la Storia. Le informazioni relative ad una Storia sono mantenute all’interno della tabella Storia.
* Un utenteModeratore scrive anche le Sessioni, le quali sono identificate da un numero progressivo e dal contenuto che fornisce la descrizione di tutto quello che succede all’interno della sessione. Ad ogni sessione può far parte un certo numero di Nemici e ad ogni sessione può essere collegata uno o più keyword. Queste informazioni sono contenute all’interno della tabella Sessioni.
* Un Nemico che prende parte ad una Sessione è identificato da un nome, ha un certo valore di attacco, una descrizione e un potere speciale. Tutte queste informazioni sono all’interno della tabella Nemico.
* Un utenteGiocatore per poter partecipare ad una Storia deve necessariamente creare un Personaggio, abbreviato PG. Il PG è identificato dalle sue Generalità, ha un Tarocco dominante, delle caratteristiche, delle abilità, una risoluzione e quando partecipa alla Sessione, può subire delle ferite durante uno scontro. Dunque, queste informazioni sono mantenute all’interno della tabella Personaggio.
* Ogni PG è in possesso di un numero imprecisato di Oggetti, definiti da nome, che li identifica, il peso e il costo. Gli oggetti possono essere delle armi, definite dal tipo, modello, dal danno che fanno, dall’abilità, dal numero di munizioni e dalla ricarica.
* Un utenteRegistrato, sia come nella figura di utenteModeratore che di utenteGiocatore, può interagire con dei Mazzi, composti da Carte, le quali possono essere Tarocchi o Carte da Poker, il Mazzo è descritto dal tipo e dal numero di carte rimaste; le Carte da Poker sono caratterizzate dal loro valore, mentre i Tarocchi sono definiti dalla descrizione, dal valore delle carte e dalla descrizione dominate. Tutte queste informazioni sono contenute nelle tabelle: Mazzo, Carta, Carta da Poker e Tarocco.

3.5 Schema Entity-Relationship 

Questo è lo schema iniziale dello schema ER, non ancora ristrutturato. La sezione di Mazzo e le corrispondenti relazioni: “ha” e “possiede” e le entità “Carta da Poker” e “Tarocco”, essendo delle entità statiche che non cambiano nel tempo, abbiamo deciso di memorizzarle in un file da utilizzare poi al momento della loro istanziazione; anche l’entità Nemico, essendo statica, come la sezione Mazzi, sarà memorizzate in un file. Detto questo, riportetemo di seguito lo schema ER con i dovuti cambiamenti.

In fasi di ristrutturazione abbiamo risolto tutte le generalizzazioni, accorpando le classi “utenteModeratore” e “utenteGiocatore” nella tabella “utenteRegistrato” la quale si differenzierà tramite l’attributo ruolo, che è un enumeratore che può essere “utenteModeratore” o “utenteGiocatore”. Inoltre, abbiamo deciso di rendere l’attributo della classe “Personaggio” ‘attributo’, una tabella, poiché abbiamo bisogno di tenere traccia delle abilità del Personaggio in maniera consistente ed efficace. Di seguito l’ER creato con Visual Paradigm versione definitiva.



|  |  |
| --- | --- |
| UtenteRegistrato | |
| **Campo** | **Vincoli** |
| Username | Lunghezza massima: 15 caratteri, unique  Chiave primaria |
| Password | Lunghezza massima: 15 caratteri, not null |
| E-Mail | Lunghezza massima: 20 caratteri, not null |
| Nome | Not null |
| Cognome | not null |
| Ruolo | Enumeratore: utenteModeratore/utenteGiocatore |

|  |  |
| --- | --- |
| Personaggio | |
| **Campo** | **Vincoli** |
| Nome | Lunghezza massima: 30 caratteri,  Chiave primaria |
| Cognome | Lunghezza massima: 30 caratteri,  chiave primaria |
| Età | Not null |
| Nazionalità | Not null |
| Tarocco\_dominante | Lunghezza massima: 15, not null |
| Caratteristiche | Intero compreso tra 0 e 10, not null |
| Abilità | not null |
| Risoluzione | not null |
| Ferite | not null |

|  |  |
| --- | --- |
| Storia | |
| **Campo** | **Vincoli** |
| Titolo | Lunghezza massima: 50 caratteri, unique  Chiave primaria |
| Descrizione | Lunghezza massima: 500 caratteri, not null |
| Ambientazione | Enumeratore: Terre perdute/Quarto Reich/Soviet/Sanctum Imperum , not null |
| Nome (chiave primaria di Personaggio) | Lunghezza massima: 30 caratteri,  Chiave esterna |
| Cognome (chiave primaria di Personaggio) | Lunghezza massima: 30 caratteri,  Chiave esterna |

|  |  |
| --- | --- |
| Sessione | |
| **Campo** | **Vincoli** |
| Numero | Intero, unique, Chiave primaria |
| Contenuto | not null |
| Username (chiave primaria di utenteRegistrato) | not null, chiave esterna |
| Titolo (chiave primaria di Storia) | Lunghezza massima: 50 caratteri, unique  Chiave primaria |

|  |  |
| --- | --- |
| Oggetti | |
| **Campo** | **Vincoli** |
| Nom | Lunghezza massima: 30 caratteri, unique  Chiave primaria |
| Peso | Lunghezza: numero decimale, not null |
| Costo | Lunghezza: numero decimale, not null |
| Nome (chiave primaria di Personaggio) | Lunghezza massima: 30 caratteri,  Chiave esterna |
| Cognome (chiave primaria di Personaggio) | Lunghezza massima: 30 caratteri,  Chiave esterna |

|  |  |
| --- | --- |
| Armi | |
| **Campo** | **Vincoli** |
| Id | Intero, unique, Chiave primaria |
| Tipo | Enumeratore:Pistola/Mitra/Fucile, not null |
| Modello | Lunghezza massima: 15 caratteri, not null, Chiave primaria |
| Danno | Intero compreso tra 0 e 2, not null |
| Abilità | Intero compreso tra 0 e 10, not null |
| Munizione | Lunghezza massima: 10 caratteri, not null |
| Ricarica | Intero compreso tra 0 e 3, not null |

|  |  |
| --- | --- |
| Abilità | |
| **Campo** | **Vincoli** |
| Identificativo | Lunghezza massima: 15 caratteri, unique  Chiave primaria |
| Valore | Intero |
| Nome (chiave primaria di Personaggio) | Lunghezza massima: 30 caratteri,  Chiave esterna |
| Cognome (chiave primaria di Personaggio) | Lunghezza massima: 30 caratteri,  Chiave esterna |

|  |  |
| --- | --- |
| Realizza (relazione tra utenteRegistrato e Storia) | |
| **Campo** | **Vincoli** |
| Username (chiave primaria di utenteRegistrato) | Lunghezza massima: 15 caratteri, unique  Chiave esterna |
| Titolo (chiave primaria di Storia) | Lunghezza massima: 50 caratteri, unique  Chiave esterna |

|  |  |
| --- | --- |
| Crea (relazione tra utenteRegistrato e Personaggio) | |
| **Campo** | **Vincoli** |
| Username (chiave primaria di utenteRegistrato) | Not null, chiave esterna |
| Numero | Intero, unique, Chiave esterna |

|  |  |
| --- | --- |
| È invitato (relazione tra utenteRegistrato e Storia) | |
| **Campo** | **Vincoli** |
| Username (chiave primaria di utenteRegistrato) | Not null, chiave esterna |
| Titolo (chiave primaria di Storia) | Lunghezza massima: 50 caratteri, unique  Chiave esterna |