

Relazione per
“J.A.R.G”
Just Another Rogue-like Game

Fabbri Lucia matr. 0000873799
Frattini Francesca matr. 0000882269



Indice

1	Analisi dei Requisiti	3
1.1	Definizione delle Specifiche in Linguaggio Naturale	3
1.2	Estrazione dei Concetti Fondamentali	4
1.3	Rilevamento delle ambiguità	5
2	Progettazione Concettuale	6
2.1	Sviluppo dell'Ambito "Utente - Personaggio"	8
2.2	Sviluppo dell'Ambito "Difficoltà di Gioco - Livello"	9
2.3	Sviluppo dell'Ambito "Mondo di Gioco"	9
2.4	Sviluppo dell'Ambito "Personaggio - Interazione Oggetto"	10
2.5	Sviluppo dell'Ambito "Nemici"	11
2.6	Schema concettuale finale	12
3	Progettazione Logica	13
3.1	Stima del Volume dei Dati	13
3.2	Descrizione delle Operazioni Principali e Stima della loro Frequenza	14
3.3	Schemi di Navigazione e Stima degli Accessi	15
3.4	Raffinamento dello Schema Concettuale	25
3.4.1	Eliminazione di Gerarchie	25
3.4.2	Scelta delle Chiavi Primarie	26
3.5	Analisi delle ridondanze	26
3.6	Traduzione di Entità e Associazioni in relazioni	26
3.7	Schema Relazionale finale	29
3.8	Traduzione delle Operazioni in Query SQL	30
4	Progettazione dell'applicazione	33
4.1	Descrizione architettura dell'applicazione realizzata	33

Introduzione

Il nostro progetto consiste nella realizzazione di un database che funga da supporto all'applicazione Java realizzata in OOP di nome "J.A.R.G - Just Another Rogue-like Game". Il gioco consiste nell'esplorazione di un dungeon formato da quattro livelli da parte di un personaggio.

Il principale compito del database creato è quello di mantenere le informazioni riguardanti il personaggio, la difficoltà di gioco, i nemici, la stanza in cui si trova ed i vari ostacoli ed oggetti ottenibili all'interno di essa.

Sulla base della posizione corrente all'interno del mondo di gioco, il personaggio potrà:

- visualizzare le sue statistiche di gioco (vita, attacco e difesa), i livelli e le stanze completate;
- conoscere gli oggetti in suo possesso all'interno del proprio inventario;
- conoscere quanti nemici sono presenti all'interno della stanza e quali sono le loro caratteristiche ed abilità;
- conoscere quali e quanti sono gli oggetti da poter raccogliere e aggiungere all'inventario all'interno della stanza;
- sapere quali porte può aprire con le chiavi in suo possesso e quelle presenti in una data stanza;
- visualizzare gli oggetti ottenibili dai nemici dopo averli sconfitti nella stanza corrente.

In questo modo il giocatore potrà avere una visione completa dell'evoluzione del suo stato durante l'esplorazione e, in base ad essa, saprà come muoversi all'interno del gioco.

Dopo l'accesso al database tramite l'apposita interfaccia, a seconda della scelta effettuata, sarà possibile visualizzare le statistiche relative al proprio account e le informazioni riguardanti l'ambiente di gioco.

Per avere un esempio pratico, si potrebbe pensare che il giocatore debba poter interrogare il database per ottenere informazioni riguardanti le sue statistiche e la stanza corrente, le caratteristiche dei nemici da affrontare e i vari oggetti e ostacoli che lo circondano.

Il sistema informativo sviluppato offre un'interfaccia dinamica pur senza la presenza di un lato amministrativo in quanto è possibile scegliere il grado di difficoltà tra facile, medio oppure difficile.

Capitolo 1

Analisi dei Requisiti

1.1 Definizione delle Specifiche in Linguaggio Naturale

La seguente descrizione illustra in linguaggio naturale quali siano i requisiti da soddisfare per la realizzazione del nostro sistema informativo.

Prendendo spunto dal progetto svolto per l'esame di Programmazione ad Oggetti (a.a. 2019/20) abbiamo deciso di creare un sistema informatico di supporto all'applicazione Java J.A.R.G (Just Another Rogue like Game) per la gestione delle informazioni relative al giocatore e all'ambiente di gioco.

Sarà possibile accedere al database solamente dal lato utente e visualizzare tutte le statistiche di gioco.

Per quanto riguarda le caratteristiche dell'applicazione, non è necessario un accesso dal lato amministrativo in quanto la creazione e le impostazioni del gioco vengono gestite in un contesto esterno.

Il sistema deve poter memorizzare i dati dell'utente, quali indirizzo mail e relativa password, e successivamente consentire l'accesso ed il ripristino delle condizioni di salvataggio precedenti.

Ad ogni utente è possibile associare un solo personaggio. Dopo la creazione di quest'ultimo, l'utente dovrà scegliere una tra le tre difficoltà di gioco: Facile, Medio e Difficile. In base alla scelta eseguita verrà calibrata la distribuzione dei nemici e degli oggetti all'interno delle stanze.

All'avvio del gioco, il personaggio si trova all'interno di una delle stanze appartenenti al primo tra i quattro livelli disponibili e potrà quindi iniziare la sua esplorazione.

Trovandosi all'interno di una determinata stanza, prima di poter accedere a quelle contigue, il giocatore deve trovare la giusta chiave per aprire la porta corrispondente in quanto tutte le porte sono inizialmente chiuse.

Per facilitare il gioco, la corrispondenza tra porta che deve essere sbloccata e chiave si basa sulla composizione dello stesso materiale e quest'ultime sono distribuite in maniera casuale nell'ambiente di gioco.

All'interno di ogni stanza, in aggiunta ai nemici da sconfiggere, è possibile trovare varie tipologie di oggetti come monete collezionabili e potenziamenti ma sono presenti anche ostacoli che impediranno il libero cammino del personaggio. Per quanto riguarda le monete, se ne può trovare solamente una per ciascun livello di gioco ed una volta raccolte

vengono automaticamente aggiunte all'inventario. I potenziamenti invece servono ad incrementare le statistiche di difesa e di attacco del giocatore.

Durante la sua esplorazione, così come i nemici ed i boss finali, il personaggio ha la possibilità di acquisire diverse abilità, ciascuna delle quali ha un intervallo di azione limitato, presenta specifiche caratteristiche e necessita di un certo tempo per ricaricarsi.

Dopo aver sconfitto tutti i nemici e raccolto le chiavi all'interno di essa, ogni stanza verrà contrassegnata come "completata".

Una volta raggiunta l'ultima stanza del livello corrente, si deve affrontare la sfida contro il boss: un nemico con statistiche più alte rispetto alla media.

Una volta sconfitto, il personaggio potrà apprendere la sua abilità speciale non ottenibile diversamente ed insieme ad essa si procurerà la chiave per sbloccare la stanza del livello successivo.

Dopo aver fronteggiato e sconfitto anche il boss dell'ultimo livello del gioco, verrà visualizzata la schermata di vittoria.

1.2 Estrazione dei Concetti Fondamentali

Per realizzare uno schema significativo del progetto, ci occupiamo ora di individuare le espressioni e le parole chiave all'interno del contesto appena descritto. Abbiamo adottato la scrittura in grassetto per metterle in rilievo.

Prendendo spunto dal progetto svolto per l'esame di Programmazione ad Oggetti (a.a. 2019/20) abbiamo deciso di creare un sistema informatico di supporto all'applicazione Java J.A.R.G (Just Another Rogue like Game) per la gestione delle informazioni relative al giocatore e all'ambiente di gioco.

Sarà possibile accedere al database solamente dal lato utente e visualizzare tutte le statistiche di gioco.

Per quanto riguarda le caratteristiche dell'applicazione, non è necessario un accesso dal lato amministrativo in quanto la creazione e le impostazioni del gioco vengono gestite in un contesto esterno.

*Il sistema deve poter memorizzare i dati dell'**utente**, quali indirizzo mail e relativa password, e successivamente consentire l'accesso ed il ripristino delle condizioni di salvataggio precedenti.*

*Ad ogni utente è possibile associare un solo **personaggio**. Dopo la creazione di quest'ultimo, l'utente dovrà scegliere una tra le tre **difficoltà** di gioco: Facile, Medio e Difficile. In base alla scelta eseguita verrà calibrata la distribuzione dei nemici e degli oggetti all'interno delle stanze.*

*All'avvio del gioco, il personaggio si trova all'interno di una delle **stanze** appartenenti al primo tra i quattro livelli disponibili e potrà quindi iniziare la sua esplorazione.*

*Trovandosi all'interno di una determinata stanza, prima di poter accedere a quelle contigue, il giocatore deve trovare la giusta **chiave** per aprire la **porta** corrispondente in quanto tutte le porte sono inizialmente chiuse.*

Per facilitare il gioco, la corrispondenza tra porta che deve essere sbloccata e chiave si basa sulla composizione dello stesso materiale e quest'ultime sono distribuite in maniera casuale nell'ambiente di gioco.

*All'interno di ogni stanza, in aggiunta ai normali **nemici** da sconfiggere, è possibile trovare varie tipologie di **oggetti** come **monete** collezionabili e **potenziamenti** ma sono*

presenti anche **ostacoli** che impediranno il libero cammino del personaggio. Per quanto riguarda le monete, se ne può trovare solamente una per ciascun livello di gioco ed una volta raccolte vengono automaticamente aggiunte all'**inventario**. I potenziamenti invece servono ad incrementare le statistiche di **difesa** e di **attacco** del giocatore.

Durante la sua esplorazione, così come i nemici ed i **boss finali**, il personaggio ha la possibilità di acquisire diverse **abilità**, ciascuna delle quali ha un intervallo di azione limitato, presenta specifiche caratteristiche e necessita di un certo tempo per ricaricarsi. Dopo aver sconfitto tutti i nemici e raccolto le chiavi all'interno di essa, ogni stanza verrà contrassegnata come "completata".

Una volta raggiunta l'ultima stanza del livello corrente, si deve affrontare la sfida contro il boss: un nemico con statistiche più alte rispetto alla media.

Una volta sconfitto, il personaggio potrà apprendere la sua abilità speciale non ottenibile diversamente ed insieme ad essa si procurerà la chiave per sbloccare la stanza del livello successivo.

Dopo aver fronteggiato e sconfitto anche il boss dell'ultimo livello del gioco, verrà visualizzata la schermata di vittoria.

1.3 Rilevamento delle ambiguità

Nonostante la descrizione del sistema informativo appena fornita sia abbastanza chiara ed esaurente, ci accorgiamo della presenza di termini e concetti che possono essere fuorivianti. Procediamo quindi con ulteriori chiarimenti:

- con "utente" si indica la persona che fisicamente può accedere al database ed interrogarlo per ottenere informazioni
- con "personaggio" si indica invece l'entità che viene rappresentata all'interno del mondo di gioco, gestita all'esterno dal volere dell'utente
- con "normale" e "boss" si intende la stessa entità nemica, differenziata solo da attributi specifici

Relativamente a quanto appena detto, nel prossimo paragrafo della trattazione ci occuperemo del passaggio da linguaggio naturale alla realizzazione dello schema concettuale per il nostro database.

Capitolo 2

Progettazione Concettuale

Lo sviluppo dello schema Entity-Relationship avverrà ora per fasi: a partire dall'analisi dei requisiti, analizzeremo il problema cercando distinguere e suddividere in sottoparti tutti i vari aspetti che lo riguardano. Successivamente cercheremo di far interagire tra di loro i vari ambiti e li integreremo mano a mano in modo coerente per formare uno schema finale completo.

In particolare, ci concentreremo sull'analisi dei seguenti ambiti:

- Utente - Personaggio
- Difficoltà di Gioco - Livello
- Mondo di Gioco
- Personaggio - Interazione Oggetto
- Nemici

Nella versione finale dello schema concettuale possiamo distinguere le seguenti entità ed associazioni, ognuna delle quali seguita da una descrizione sintetica:

NOME	TIPO	DESCRIZIONE
ABILITA'	R	lega il nemico con l'abilità specifica che può avere
ABILITA'	E	rappresenta l'oggetto che il personaggio può momentaneamente acquisire per aumentare le sue statistiche
BOSS	E	rappresenta l'entità speciale contro la quale si deve scontrare il personaggio alla fine di ogni livello
CHIAVE	E	rappresenta l'oggetto con il quale è possibile aprire una porta
COMUNE	E	rappresenta l'entità nemica standard contro la quale si deve scontrare il personaggio

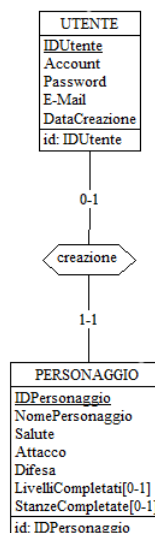
COMPOSIZIONE	R	lega il livello con le stanze di cui è composto
CONDUCE	R	lega la porta tramite la quale si può accedere ad una determinata stanza
CONTIENE	R	lega la stanza con tutti i vari oggetti che essa contiene
CREAZIONE	R	lega l'utente con il personaggio che verrà creato ed utilizzato per rappresentarlo nel mondo di gioco
DIFFICOLTA' GIOCO	E	rappresenta la difficoltà con la quale si decide di affrontare il gioco (facile, medio o difficile)
HA	R	lega i nemici con la stanza in cui si trovano
INVENTARIO	R	lega il personaggio con l'oggetto di cui dispone
LIVELLO	E	rappresenta il livello in cui si trova il giocatore
MONETA COLLEZIONABILE	E	rappresenta l'oggetto singolarmente collezionabile ad ogni livello del gioco
NEMICO	E	rappresenta l'entità contro la quale si deve scontrare il personaggio
OGGETTO	E	rappresenta la generica entità che si può trovare all'interno di una stanza
OSTACOLO	E	rappresenta l'oggetto con il quale si può avere una collisione e che non può essere raccolto dal giocatore
PERSONAGGIO	E	rappresenta l'entità che si muove all'interno del mondo di gioco
PORTA	E	rappresenta l'oggetto tramite il quale ci sposta di stanza in stanza
POTENZIAMENTO	E	rappresenta l'oggetto che è possibile raccogliere ed utilizzare per aumentare le statistiche del personaggio
PU_ATTACCO	E	rappresenta i punti che verranno aggiunti alla potenza di attacco del personaggio
PU_DIFESA	E	rappresenta i punti che verranno aggiunti alla potenza di difesa del personaggio
PUO' APPRENDERE	R	lega il personaggio con l'abilità che può apprendere

PUO' RILASCIARE	R	lega i nemici con l'oggetto che esso può rilasciare nel mondo di gioco una volta sconfitto
SBLOCCA	R	lega la chiave con la porta che può aprire, entrambe composte dallo stesso materiale
SCEGLIE	R	lega l'utente con la difficoltà di livello che può selezionare
SI BASA SU	R	lega i vari livelli di gioco calibrati sulla difficoltà di gioco scelta
SI MUOVE ATTRAVERSO	R	lega il personaggio con la stanza in cui si trova
SI TROVA	R	lega l'ostacolo con la stanza in cui si trova
STANZA	E	rappresenta il luogo in cui si muove il giocatore
UTENTE	E	rappresenta la persona che fisicamente interagisce, interroga e visualizza il database

2.1 Sviluppo dell'Ambito "Utente - Personaggio"

La modellazione di questo aspetto del database non presenta particolari difficoltà: sarà infatti necessaria l'entità Utente, legata all'entità Personaggio per mezzo della relazione Creazione.

Ad un account è possibile associare un solo personaggio e a sua volta ogni personaggio è relativo ad un singolo account utente.



Ambito Utente - Personaggio

2.2 Sviluppo dell'Ambito "Difficoltà di Gioco - Livello"

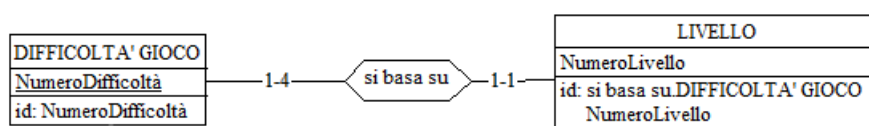
In questa sezione tratteremo la relazione tra l'entità Difficoltà Gioco ed i relativi Livelli esplorabili.

Al termine della creazione del personaggio, prima di poter iniziare l'esplorazione, sarà necessario selezionare la difficoltà di gioco tra Facile, Normale e Difficile.

La difficoltà determinerà non solo il numero di stanze di cui è composto ogni livello ma anche la quantità di nemici e le proprie statistiche, gli ostacoli da evitare e gli oggetti che possono essere raccolti relativi ad ogni stanza.

Ovviamente, quanto più complicata sarà l'esplorazione tante più stanze e nemici si incontreranno, ma d'altro canto anche i potenziamenti e le abilità disponibili aumenteranno.

A prescindere da quale difficoltà venga scelta, i livelli di gioco esplorabili sono sempre 4.



Ambito Difficoltà di Gioco - Livello

2.3 Sviluppo dell'Ambito "Mondo di Gioco"

All'interno di questo ambito discuteremo di come è organizzato il mondo di gioco, nel quale il personaggio può muoversi liberamente.

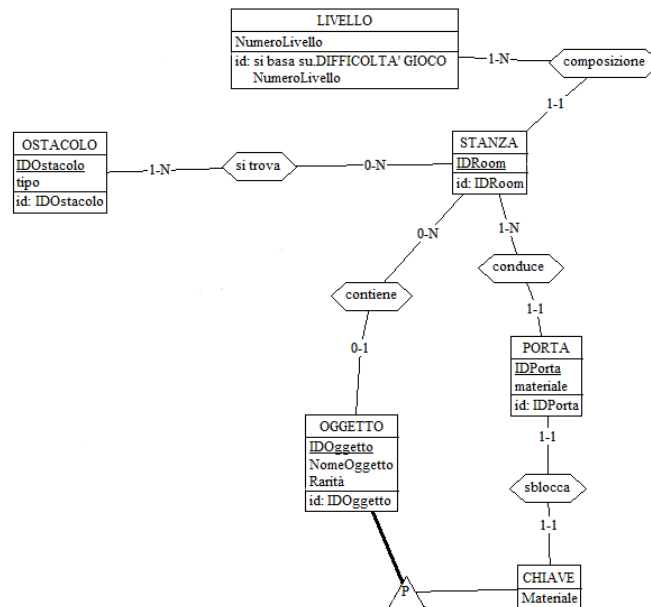
Guardando attentamente la schema sottostante, possiamo notare una sorta di gerarchia e sequenzialità che ci permette di fornire una descrizione precisa.

In linea generale, il mondo è formato dall'entità livello legata attraverso la relazione Composizione a Stanze, un numero variabile di entità che compongono il livello in base alla difficoltà scelta.

Successivamente, all'interno di ogni stanza, possiamo trovare una serie di altre entità che interagiscono direttamente con il personaggio: tramite la relazione Si trova abbiamo Ostacolo, ovvero l'entità con la quale si può avere una collisione e che impedisce il proseguimento del cammino in quella direzione oppure possiamo trovare la generica entità Oggetto legata a stanza tramite la relazione Contiene, formata a sua volta da una gerarchia di entità che illustreremo in un secondo momento.

Per spostarsi di stanza in stanza, il personaggio può fare uso dell'entità Porta la quale viene abilitata dalla corrispondente entità Chiave tramite la relazione Sblocca ed entrambe devono essere composte dallo stesso materiale.

Tramite una porta si può accedere ad una determinata stanza grazie alla relazione Conduce ed una stanza può a sua volta contenere più porte per avere riferimento ad altre.



Ambito Mondo di Gioco

2.4 Sviluppo dell'Ambito "Personaggio - Interazione Oggetto"

In quest'ambito si cerca di modellare la caratterizzazione del personaggio man mano che ci si addentra nel gioco.

Individuiamo quindi le entità Personaggio ed Oggetto, quest'ultima formata da una gerarchia di sottoentità: Potenziamento ed Abilità relative allo sviluppo del personaggio principale, Chiave necessaria per il proseguimento nell'esplorazione del mondo di gioco ed infine Moneta Collezionabile.

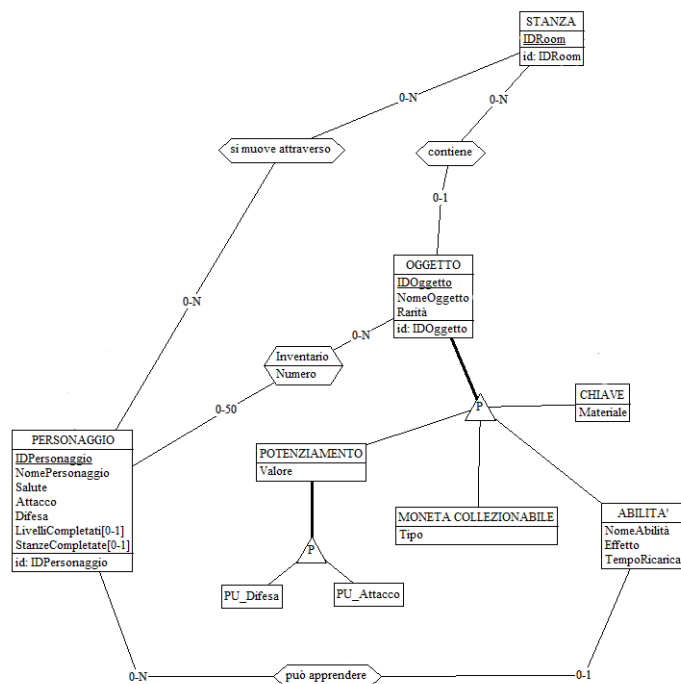
Per quanto riguarda le abilità, possono essere guadagnate dal personaggio una volta che quest'ultimo ha sconfitto un nemico oppure si possono incontrare casualmente esplorando le varie stanze del dungeon.

Le abilità che vengono rilasciate dai nemici dopo una vittoria sono uniche e non possono essere scovate durante la normale esplorazione. Questo implica che ogni livello offre uno specifico set di abilità ottenibili ed il personaggio potrà impararne solo una alla volta durante il gioco.

Così come le abilità, anche i potenziamenti si possono trovare durante l'esplorazione oppure si possono ottenere con un po' di fortuna sconfiggendo i nemici comuni. Di conseguenza, quanto più alta sarà la difficoltà ed il livello in cui ci si trova, tanto più saranno frequenti le possibilità di ottenere potenziamenti dai nemici.

Sia le chiavi che le monete possono essere incontrate casualmente nelle varie stanze. Mentre le monete sono puramente collezionabili senza una specifica utilità, le chiavi vengono impiegate per sbloccare le porte e muoversi nel dungeon. La chiave speciale per passare da un livello al successivo viene rilasciata solamente dal boss finale dopo la sua sconfitta. Per quanto riguarda l'inventario, esso consiste in 50 slot oggetto che possono contenere anche più copie degli stessi oggetti e per ciascuno di essi viene registrata ed aggiornata in ogni momento la quantità disponibile.

Per modellare tutti gli aspetti appena elencati faremo uso delle relazioni Inventario e Numero.



Ambito Personaggio - Interazione Oggetto

2.5 Sviluppo dell'Ambito "Nemici"

Per quanto riguarda l'ambito nemici, ci occuperemo della loro modellazione andando ad analizzare la loro interazione con l'ambiente di gioco in maniera generale.

I nemici vengono disposti in maniera casuale all'interno di ogni stanza del gioco e vengono suddivisi in due sottoentità: Comune sono tutti quei nemici con abilità e caratteristiche normali mentre Boss sono quei nemici che si incontrano solamente a fine di ogni livello e presentano abilità e caratteristiche speciali.

Ciascun nemico, indipendentemente dalla sua natura, può quindi avere determinate abilità ma il loro utilizzo è limitato come si evince dalla relazione Abilità e Chance di Utilizzo. Inoltre, attraverso la relazione Può rilasciare e Percentuale ottenimento che lega ogni nemico con un oggetto, possiamo affermare come la presenza di quest'ultimo all'interno di una stanza sia proporzionale allo specifico nemico che lo possiede.

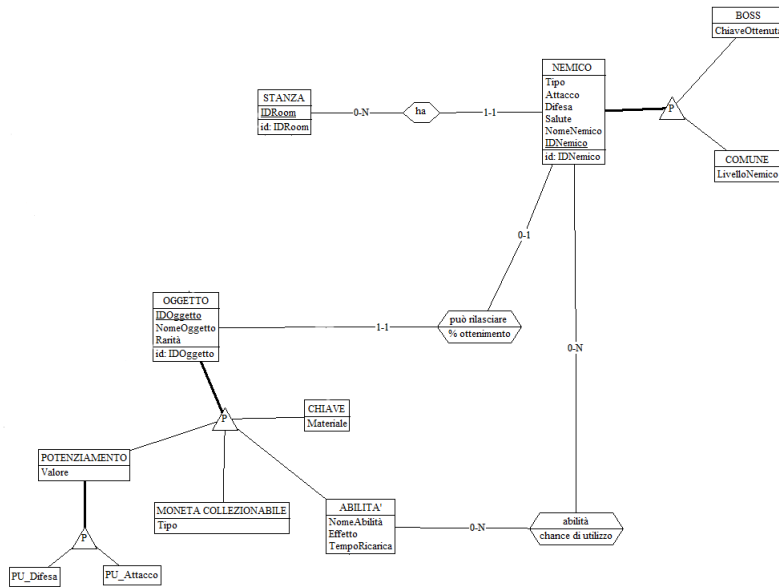
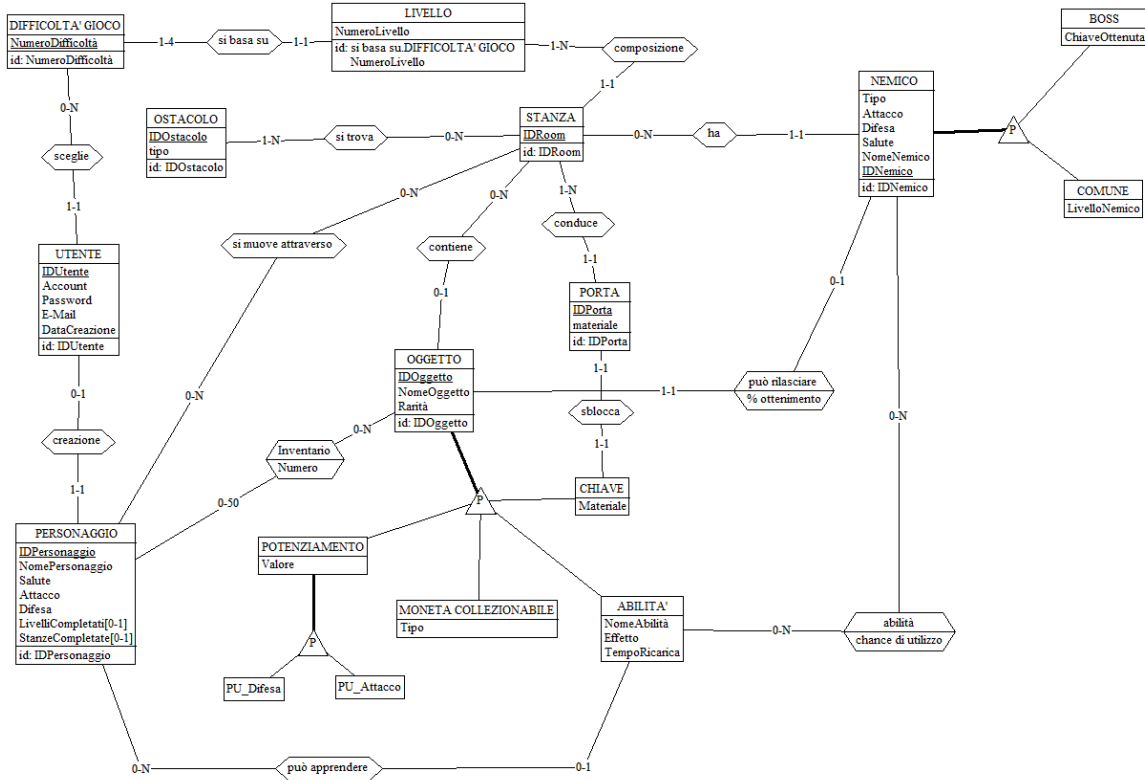


Figura 2.1: Ambito Nemici

2.6 Schema concettuale finale

La seguente immagine illustra come tutti gli ambiti descritti precedentemente siano stati uniti per andare a formare lo schema Entity-Relationship finale del nostro database.



Schema E-R completo

Capitolo 3

Progettazione Logica

3.1 Stima del Volume dei Dati

La seguente tabella riporta il volume medio stimato dei dati che compongono le varie entità ed associazioni evidenziato nel nostro schema concettuale.

Abbiamo stimato i valori considerando la scelta di un livello di difficoltà medio, inserendo i dati relativi all'intero database con riferimento generale e non al singolo livello.

Considerando le operazioni che devono essere supportate dal nostro database insieme ai valori riportati in tabella, rimodelleremo poi il nostro schema per migliorarne l'efficienza e l'aderenza al modello logico.

Nel nostro caso, il modello logico adottato è quello relazionale.

NOME	TIPO	VOLUME DI DATI
ABILITA'	R	80
ABILITA'	E	58
BOSS	E	12
CHIAVE	E	60
COMUNE	E	160
COMPOSIZIONE	R	1000
CONDUCE	R	60
CONTIENE	R	180
CREAZIONE	R	20
DIFFICOLTA' GIOCO	E	3
HA	R	172
INVENTARIO	R	1000
LIVELLO	E	12
MONETA COLLEZIONABILE	E	12

NEMICO	E	172
OGGETTO	E	180
OSTACOLO	E	300
PERSONAGGIO	E	20
PORTA	E	60
POTENZIAMENTO	E	50
PU_ATTACCO	E	25
PU_DIFESA	E	25
PUO' APPRENDERE	R	1160
PUO' RILASCIARE	R	172
SBLOCCA	R	60
SCEGLIE	R	60
SI BASA SU	R	12
SI MUOVE ATTRAVERSO	R	1000
SI TROVA	R	300
STANZA	E	50
UTENTE	E	20

3.2 Descrizione delle Operazioni Principali e Stima della loro Frequenza

È importante rendersi conto ed analizzare attentamente tutte le operazioni sui dati che verranno eseguite maggiormente dal nostro database così come comprendere le varie funzionalità che dovrà supportare in modo tale da agevolarne lo sviluppo futuro.

Nei paragrafi successivi ci occuperemo quindi di esporre questi concetti, analizzando le operazioni elementari previste per il corretto funzionamento del nostro database e per l'estrazione e manipolazione dei dati contenuti al suo interno.

Nella tabella seguente distinguiamo le principali operazioni:

FUNZIONALITÀ		FREQUENZA
NUMERO	DESCRIZIONE	
1	Visualizzare l'inventario del personaggio e le sue statistiche correnti	100/Giorno

2	Visualizzare in quali stanze si può apprendere una determinata abilità	20/Giorno
3	Nemici presenti in una determinata stanza e le loro statistiche	400/Giorno
4	Abilità ottenibili e relative statistiche dei boss presenti nei 4 livelli della difficoltà scelta	20/Giorno
5	Visualizzare quante porte si possono aprire con le chiavi disponibili in inventario e quelle in una data stanza	1100/Giorno
6	Visualizzare stanza in cui si trova il giocatore, numero di stanze da lui completate e difficoltà selezionata	20/Giorno
7	Visualizzare oggetti con rarità massima in un dato livello	80/Giorno
8	Oggetti ottenibili dai nemici comuni dopo averli sconfitti nella stanza corrente	400/Giorno

3.3 Schemi di Navigazione e Stima degli Accessi

Avendo stabilito il volume dei dati ed associato a ciascuna operazione la propria frequenza di esecuzione, determiniamo ora lo schema di navigazione per le operazioni richieste al nostro database.

Utilizzando i dati riportati nella tabella delle frequenze e confrontandoli con quelli ricavati dalla tabella dei volumi dei dati, potremo poi stimare il volume medio degli accessi al nostro database.

Ricordiamo che il peso degli accessi in scrittura è considerato il doppio rispetto al peso degli accessi in lettura.

Visualizzare l'inventario del personaggio e le sue statistiche correnti

Quest'operazione è utile al giocatore per poter visualizzare gli oggetti a sua disposizione nell'inventario e, dopo aver consultato le sue statistiche, poter organizzare la sua esplorazione all'interno di una stanza.

Dall'utente si seleziona il proprio personaggio, successivamente si scansionano e si raggruppano gli oggetti dell'inventario e vengono elencate le statistiche del personaggio stesso.

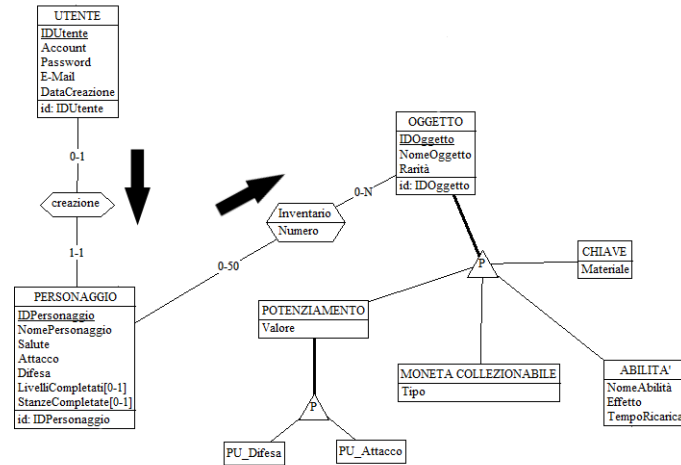
ENTITÀ COINVOLTE:

- Utente
- Personaggio

- Oggetto

ASSOCIAZIONI COINVOLTE:

- Creazione
- Inventario



CONCETTO	COSTRUTTO	ACCESSI	TIPO (L/S)
Utente	E	1.00	L
Personaggio	E	1.00	L
Oggetto	E	50.00	L
Creazione	R	1.00	L
Inventario	R	50.00	L

Visualizzare le stanze in cui si può apprendere una determinata abilità

Questa opzione può rivelarsi utile nel caso in cui si voglia far apprendere al personaggio una determinata abilità.

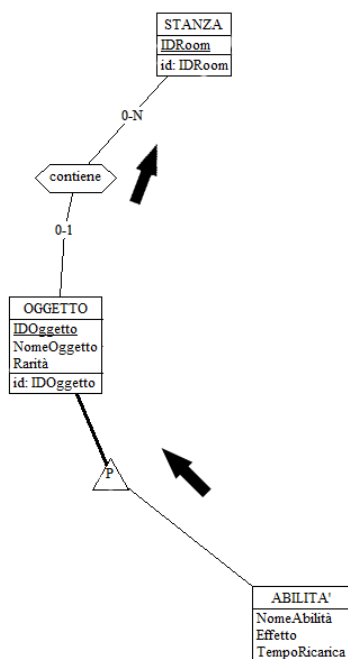
Si inizia scegliendo quale abilità cercare, dopodichè vengono raggruppate le stanze collegate all'abilità scelta.

ENTITÀ COINVOLTE:

- Stanza
- Oggetto

ASSOCIAZIONI COINVOLTE:

- Contiene



CONCETTO	COSTRUTTO	ACCESSI	TIPO (L/S)
Stanza	E	50.00	L
Abilità	E	1.00	L
Contiene	R	50.00	L

Nemici presenti in una determinata stanza e le loro statistiche

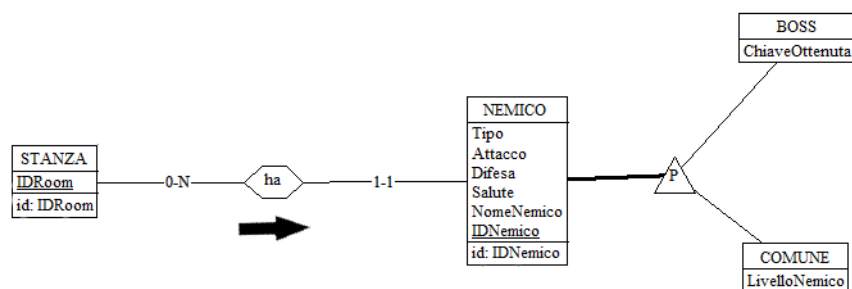
La seguente operazione può essere utile per vedere quali e quanti nemici sono rimasti da sconfiggere e aumentare le probabilità di batterli, studiando le loro statistiche di gioco. Per prima cosa viene selezionata una stanza ed in base alla scelta si va ad analizzare la presenza di entità nemiche, visualizzando le relative caratteristiche.

ENTITÀ COINVOLTE:

- Stanza
- Nemico

ASSOCIAZIONI COINVOLTE:

- Ha



CONCETTO	COSTRUTTO	ACCESSI	TIPO (L/S)
Stanza	E	1.00	L
Nemico	E	3.00	L
Ha	R	3.00	L

Abilità ottenibili e relative statistiche dei boss presenti nei 4 livelli della difficoltà scelta

La seguente operazione risulta essere utile quando si vuole avere una visuale completa su quali abilità speciali possano essere ottenute dai vari boss in base alla difficoltà di gioco scelta.

Una volta scelto l'utente, è possibile visualizzare quale difficoltà di gioco è collegata al suo account ed i vari livelli.

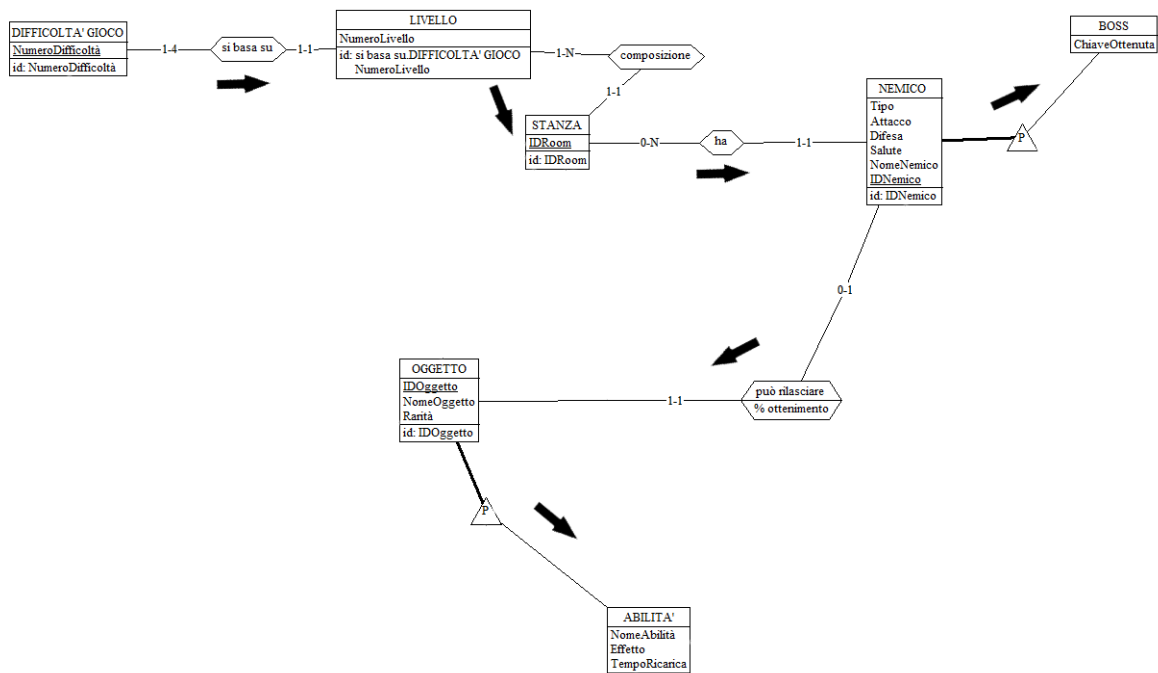
Successivamente, a partire da ciascun livello, è possibile risalire ai relativi boss finali e visualizzarne le statistiche.

ENTITÀ COINVOLTE:

- Difficoltà Gioco
- Livello
- Stanza
- Nemico
- Boss
- Oggetto
- Abilità

ASSOCIAZIONI COINVOLTE:

- Si basa su
- Composizione
- Ha
- Può rilasciare



CONCETTO	COSTRUTTO	ACCESSI	TIPO (L/S)
Difficoltà Gioco	E	1.00	L
Livello	E	4.00	L
Stanza	E	50.00	L
Nemico	E	172.00	L
Boss	E	4.00	L
Oggetto	E	4.00	L
Abilità	E	4.00	L
Si basa su	R	4.00	L
Composizione	R	50.00	L
Ha	E	172.00	L
Può rilasciare	R	4.00	L

Visualizzare quante porte si possono aprire con le chiavi disponibili in inventario e quelle presenti in una data stanza

L'operazione appena descritta è molto utile per riuscire a tenere traccia e sfruttare tutte le chiavi raccolte in modo tale da esplorare il mondo di gioco nella sua interezza, stanza dopo stanza.

A partire dal personaggio, vengono selezionate e confrontate le chiavi presenti nel suo inventario con le porte che possono essere aperte nella stanza corrente ma la visualizzazione avviene solo su corrispondenza di materiale.

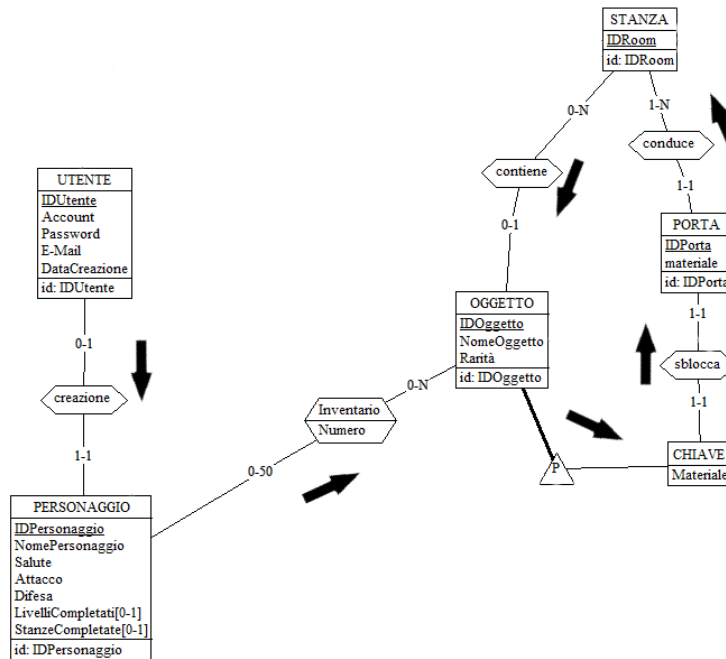
Considerando invece le chiavi contenute e che possono essere raccolte nella stanza corrente, vengono confrontate e visualizzate solo quelle con corrispondenza di materiale.

ENTITÀ COINVOLTE:

- Utente
- Personaggio
- Oggetto
- Porta
- Chiave
- Stanza

ASSOCIAZIONI COINVOLTE:

- Creazione
- Inventario
- Sblocca
- Contiene
- Conduce



CONCETTO	COSTRUTTO	ACCESSI	TIPO (L/S)
Utente	E	1.00	L
Personaggio	E	1.00	L
Oggetto	E	150.00	L
Porta	E	150.00	L
Chiave	E	150.00	L
Stanza	E	1.00	L
Porta	E	150.00	L
Inventario	E	50.00	L
Creazione	R	1.00	L
Sblocca	R	150.00	L
Contiene	R	3.00	L
Si muove attraverso	R	50.00	L

Visualizzare stanza in cui si trova il giocatore, numero di stanze completate e difficoltà selezionata

Questa operazione può essere utile per visualizzare in quale stanza il personaggio si trovi al momento rispetto all'intero mondo di gioco esplorabile e quale difficoltà di gioco sia stata scelta.

In base all'utente selezionato viene visualizzata la difficoltà di gioco relativa al suo account.

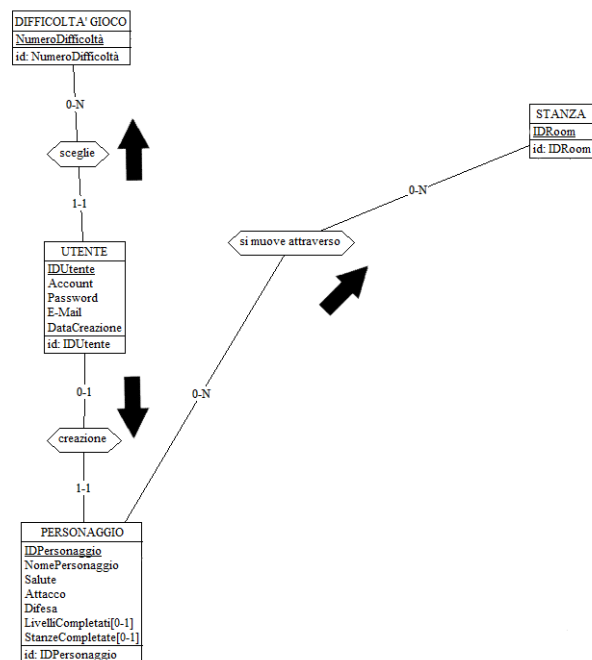
Sulla base del personaggio che corrisponde all'utente, è possibile risalire alla stanza in cui si trova al momento.

ENTITÀ COINVOLTE:

- Utente
- Difficoltà Gioco
- Personaggio
- Stanza

ASSOCIAZIONI COINVOLTE:

- Sceglie
- Si muove attraverso
- Creazione



CONCETTO	COSTRUTTO	ACCESSI	TIPO (L/S)
Utente	E	2.00	L
Difficoltà Gioco	E	1.00	L
Personaggio	E	1.00	L
Stanza	E	50.00	L
Sceglie	R	1.00	L
Si muove attraverso	R	50.00	L
Creazione	R	1.00	L

Visualizzare oggetti con rarità massima in un dato livello

La seguente opzione risulta utile quando si vogliono trovare tutti gli oggetti all'interno di un certo livello con il massimo grado di rarità, poichè essi hanno una minor probabilità di essere raccolti.

A partire dall'utente, si accede alla difficoltà di gioco scelta all'inizio, poi si raggiunge uno dei quattro livelli.

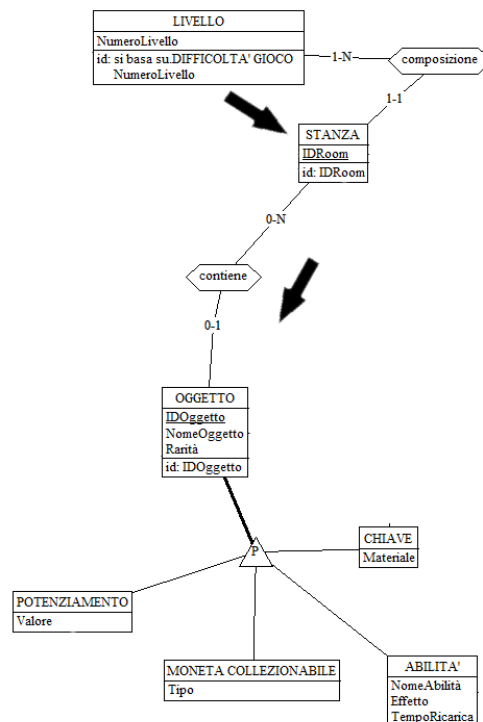
Sucessivamente, vengono analizzate le stanze di un dato livello e vengono raggruppati tutti gli oggetti con il massimo grado di rarità ottenibili tramite l'esplorazione.

ENTITÀ COINVOLTE:

- Livello
- Stanza
- Oggetto

ASSOCIAZIONI COINVOLTE:

- Composizione
- Contiene



CONCETTO	COSTRUTTO	ACCESSI	TIPO (L/S)
Livello	E	5.00	L
Stanza	E	12.00	L

Oggetto	E	45.00	L
Composizione	R	12.00	L
Contiene	R	45.00	L

Oggetti ottenibili dai nemici comuni dopo averli sconfitti nella stanza corrente

Questa operazione serve per visualizzare quali oggetti i nemici comuni presenti nella stanza corrente possono rilasciare nell'ambiente di gioco una volta sconfitti e quindi permette di decidere chi affrontare in base alle preferenze.

Per prima cosa, viene analizzata e visualizzata in quale stanza il personaggio si trovi al momento della richiesta.

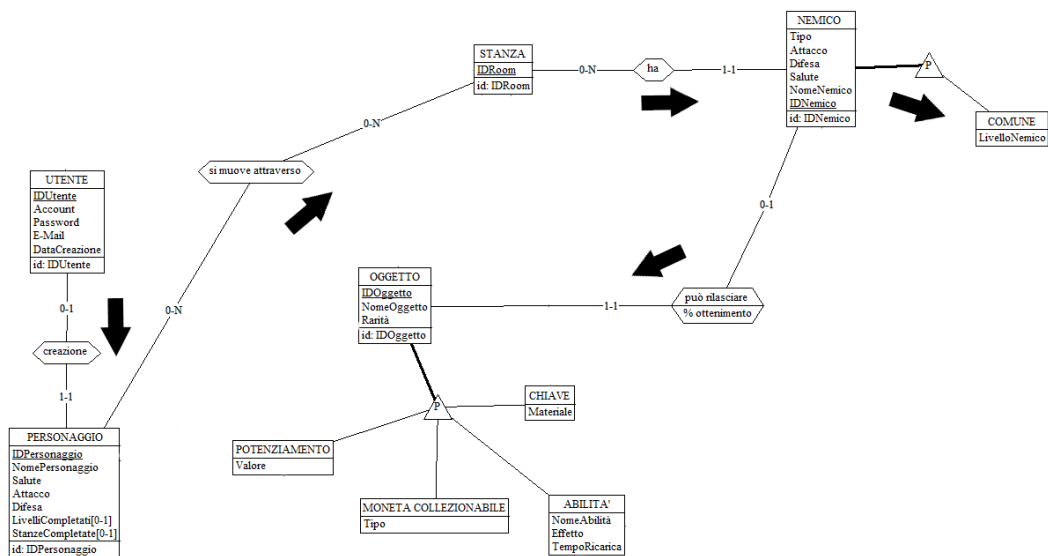
Successivamente, vengono analizzati tutti i nemici comuni che si trovano all'interno di essa e per ciascuno viene visualizzato l'oggetto che potrà rilasciare dopo la sua sconfitta.

ENTITÀ COINVOLTE:

- Utente
- Personaggio
- Stanza
- Oggetto
- Nemico
- Comune

ASSOCIAZIONI COINVOLTE:

- Creazione
- Si muove attraverso
- Può rilasciare
- Ha



CONCETTO	COSTRUTTO	ACCESSI	TIPO (L/S)
Stanza	E	51.00	L
Oggetto	E	108.00	L
Utente	E	1.00	L
Personaggio	E	1.00	L
Nemico	E	172.00	L
Si muove attraverso	R	50.00	L
Creazione	R	1.00	L
Ha	R	172.00	L
Può rilasciare	R	160.00	L

3.4 Raffinamento dello Schema Concettuale

In questa sezione ci occuperemo di effettuare tutti i raffinamenti e tutte le modifiche necessarie per trasformare e tradurre il nostro schema Entity-Relationship nello schema logico relazionale corretto.

In particolare, ci occuperemo dell'eliminazione delle gerarchie di generalizzazione e della scelta delle chiavi primarie, tralasciando invece l'eliminazione di identificatori esterni e di attributi composti in quanto non compaiono all'interno del nostro schema concettuale di partenza.

3.4.1 Eliminazione di Gerarchie

Nel nostro schema Entity-Relationship possiamo individuare varie gerarchie e per ciascuna di esse effettueremo una ristrutturazione:

- la gerarchia dell'entità Potenzimento è di tipo totale ed esclusiva. Essendo composta solamente da due sottoentità Pu_Attacco e Pu_Difesa, abbiamo deciso di procedere con il collasso verso l'alto in quanto i sottotipi non presentano ulteriori associazioni: è necessario inserire un riferimento al tipo di potenziamento

che potrà essere scelto nell'entità padre, ma non avendo attributi non ci saranno ulteriori modifiche da fare.

- la gerarchia dell'entità Oggetto è di tipo totale ed esclusiva.
In questo caso, abbiamo deciso di procedere con il collasso verso l'alto, operazione possibile dato il tipo di copertura. Questa scelta è stata dettata dalla presenza di più associazioni legate all'entità Oggetto poichè se avessimo scelto il collasso verso il basso, avremmo dovuto replicare per ogni sottotipo di oggetto le quattro associazioni coinvolte.
- la gerarchia dell'entità Nemico è di tipo totale ed esclusiva.
Abbiamo scelto di procedere con il collasso verso l'alto in quanto sono presenti solamente due entità figlie Nemico e Boss, le quali presentano molti campi comuni e non svolgono azioni specifiche.
È quindi necessario inserire un riferimento al tipo che potrà essere scelto nell'entità padre ed aggiungere i due attributi rendendoli opzionali.

3.4.2 Scelta delle Chiavi Primarie

Per quanto riguarda la scelta della chiave primaria, essa risulta un'operazione molto semplice, in quanto per ogni entità abbiamo deciso che venga identificata tramite un campo di tipo numerico vista la natura del sistema informativo e quindi non potranno presentarsi problemi di uguaglianza di chiavi.

3.5 Analisi delle ridondanze

All'interno dell'entità nemico, notiamo la presenza dell'attributo "LivelloNemico" riferito al nemico comune che ci indica il livello nel quale possiamo trovarlo.

Questa informazione la possiamo ottenere con un metodo alternativo: a partire dall'entità Nemico utilizziamo la relazione Ha per accedere alla stanza in cui si trova e poi dalla stanza tramite la relazione Composizione risaliamo all'entità Livello da noi ricercata.

Osservando gli schemi di navigazione delle operazioni e le relative tabelle degli accessi, si evince che l'attributo da noi considerato ridondante assume in realtà un ruolo superfluo nell'effettuazione delle operazioni richieste al database.

Per questo motivo, abbiamo deciso di mantenere tale attributo considerando anche lo spreco trascurabile di memoria per conservarlo.

3.6 Traduzione di Entità e Associazioni in relazioni

Traduzione delle Entità

La traduzione delle entità presenti nel nostro schema concettuale risulta essere un processo nel complesso semplice.

Il campo in grassetto indica l'entità che prendiamo in considerazione mentre il campo sottolineato evidenzia la relativa chiave primaria.

- * **UTENTE** (IDUtente, Account, Password, E-Mail, DataCreazione)
AK: Account, Password

- * **DIFFICOLTÀ GIOCO** (NumeroDifficoltà)
- * **LIVELLO** (NumeroLivello)
- * **STANZA** (IDRoom)
- * **OSTACOLO** (IDOstacolo, Tipo)
- * **PORTA** (IDPorta, Materiale)
- * **OGGETTO** (IDOggetto, TipoOggetto, NomeOggetto, Rarità, Valore*, TipoPotenziamento*, Effetto*, Materiale*, TempoRicarica*, TipoMoneta*)
- * **PERSONAGGIO** (IDPersonaggio, NomePersonaggio, Salute, Attacco, Difesa, LivelliCompletati*, StanzeCompletate*)
AK: NomePersonaggio
- * **NEMICO** (IDNemico, TipoNemico, Attacco, Difesa, Salute, NomeNemico, LivelloNemico*, ChiaveOttenuta*)

Traduzione delle Associazioni

La traduzione delle associazioni presenti nel nostro schema concettuale risulta essere invece un processo più complicato.

Nel nostro caso, si tratta di tradurre solamente associazioni binarie di diversi tipi, ma in base alla scelta che verrà effettuata è possibile che si andranno a modificare anche le entità descritte precedentemente introducendo nuovi attributi o creando nuove entità.

Infatti, anche se le associazioni di tipo molti a molti seguiranno la traduzione standard, le informazioni riguardanti le associazioni binarie uno a molti potranno essere racchiuse in una delle relazioni già codificate per le entità legate dall'associazione, senza dover introdurre una nuova relazione. Anche in questo caso, come per le entità, il campo in grassetto indicherà l'associazione e le relative entità che lega mentre i campi sottolineati evidenziano le chiavi primarie.

- **SI TROVA** (lega Ostacolo e Stanza)
In una stanza sono presenti più ostacoli, allo stesso modo gli ostacoli possono essere disponibili per più di una stanza.
L'associazione è quindi di tipo molti a molti.
Di conseguenza, viene utilizzata la traduzione standard con tre relazioni: le entità Ostacolo e Stanza rimangono invariate e viene introdotta la nuova entità *si trova* che ha come identificatore le chiavi importate dalle due entità partecipanti.
Si trova (Ostacolo, Stanza)
FK:Ostacolo REFERENCES Ostacolo
FK:Stanza REFERENCES Stanza
- **ABILITÀ** (lega Nemico e Oggetto)
La situazione è analoga a quella precedente, con l'unica differenza dell'introduzione del campo *chance di utilizzo* proprio della nuova entità.
Abilità (IDNemico, IDOggetto, Chance di utilizzo)
FK:Nemico REFERENCES Nemico
FK:Oggetto REFERENCES Oggetto

- **INVENTARIO** (lega Personaggio e Oggetto)
Situazione analoga alle precedenti.
Inventario (IDPersonaggio, IDOggetto, Numero)
FK: IDPersonaggio REFERENCES Personaggio
- **SI MUOVE ATTRAVERSO** (lega Personaggio e Stanza)
In una stanza possono trovarsi più personaggi, in quanto ogni utente gioca individualmente. Allo stesso modo, ogni personaggio può muoversi attraverso più stanze.
Si muove attraverso (IDPersonaggio, IDRoom)
FK: IDPersonaggio REFERENCES Personaggio
FK: IDRoom REFERENCES Stanza
- **SCEGLIE** (lega Utente e Difficoltà Gioco)
Dal momento che un utente seleziona una ed una sola difficoltà di gioco ma ad ogni difficoltà di gioco si possono associare più utenti, ci troviamo davanti ad una relazione 1:n. Per tale motivo l'ID della difficoltà gioco diventa un attributo della relazione utente. La relazione utente viene perciò modificata.
Utenti (IDUtente, Account, Password, E-Mail, DataCreazione, NumeroDifficoltà)
FK: IDUtente REFERENCES Utente
- **POSSONO APPRENDERE** (lega Personaggio e Oggetto)
Un personaggio può apprendere più abilità, ma una singola abilità può essere imparata da un solo personaggio alla volta.
Se l'oggetto non è di tipo Abilità, il campo sarà contrassegnato con NULL.
Oggetti (IDOggetto, IDPersonaggio, NomeOggetto, Rarità, TipoOggetto, TipoPotenziamento*, Valore*, Effetto*, Materiale*, TempoRicarica*, TipoMoneta*)
FK: IDPersonaggio REFERENCES Personaggio
- **COMPOSIZIONE** (lega Stanza e Livello)
Situazione analoga alle precedenti.
Stanza (IDStanza, NumeroLivello, IDPersonaggio)
FK: NumeroLivello REFERENCES Livello
- **SI BASA SU** (lega Difficoltà Gioco e Livello)
La creazione di un livello si basa sulla difficoltà di gioco selezionata mentre una difficoltà di gioco contiene a prescindere dalla scelta sempre 4 livelli.
Livello (NumeroLivello, NumeroDifficoltà)
FK: NumeroDifficoltà REFERENCES Difficoltà
- **HA** (lega Stanza e Nemico)
Situazione analoga alle precedenti.
Nemico (IDNemico, IDRoom, TipoNemico, Attacco, Difesa, Salute, NomeNemico, ChiaveOttenuta*, LivelloNemico*)
FK: IDRoom REFERENCES Stanza
- **CONDUCE** (lega Stanza e Porta)
Una porta conduce ad una sola stanza, mentre si può accedere ad una stanza attraverso più porte.

Porta(IDPorta, IDRoom, Materiale)
FK: IDRoom REFERENCES Stanza

- **CONTIENE** (lega Stanza e Oggetto)
Situazione analoga alle precedenti.

Oggetti (IDOggetto, IDRoom*, IDPersonaggio, NomeOggetto, Rarità, TipoOggetto, TipoPotenziamento*, Valore*, Effetto*, Materiale*, TempoRicarica*, TipoMoneta*)

FK: IDRoom REFERENCES Stanza

- **SBLOCCA** (lega Porta e Oggetto)
Situazione analoga alle precedenti.

Questa relazione ha valenza solo quando il tipo di oggetto è Chiave, altrimenti il valore è NULL.

Porta (IDPorta, IDChiave, Materiale)
FK: IDChiave REFERENCES Oggetto

- **PUÓ RILASCIARE** (lega Oggetto e Nemico)

Essendo le due entità legate da una relazione 1:1, nell'entità Oggetto viene aggiunto il riferimento al nemico.

La seguente relazione è valida solo per i Potenzamenti e le Abilità, negli altri casi il campo ha valore NULL.

L'attributo dell'associazione *%ottenimento* viene incorporato nell'entità oggetto.

Oggetto (IDOggetto, NemicocheLoRilascia*, IDRoom*, IDPersonaggio, NomeOggetto, Rarità, TipoOggetto, TipoPotenziamento*, Valore*, Effetto*, Materiale*, TempoRicarica*, TipoMoneta*, %Otttenimento*)

FK: NemicoCheLoRilascia REFERENCES Nemico

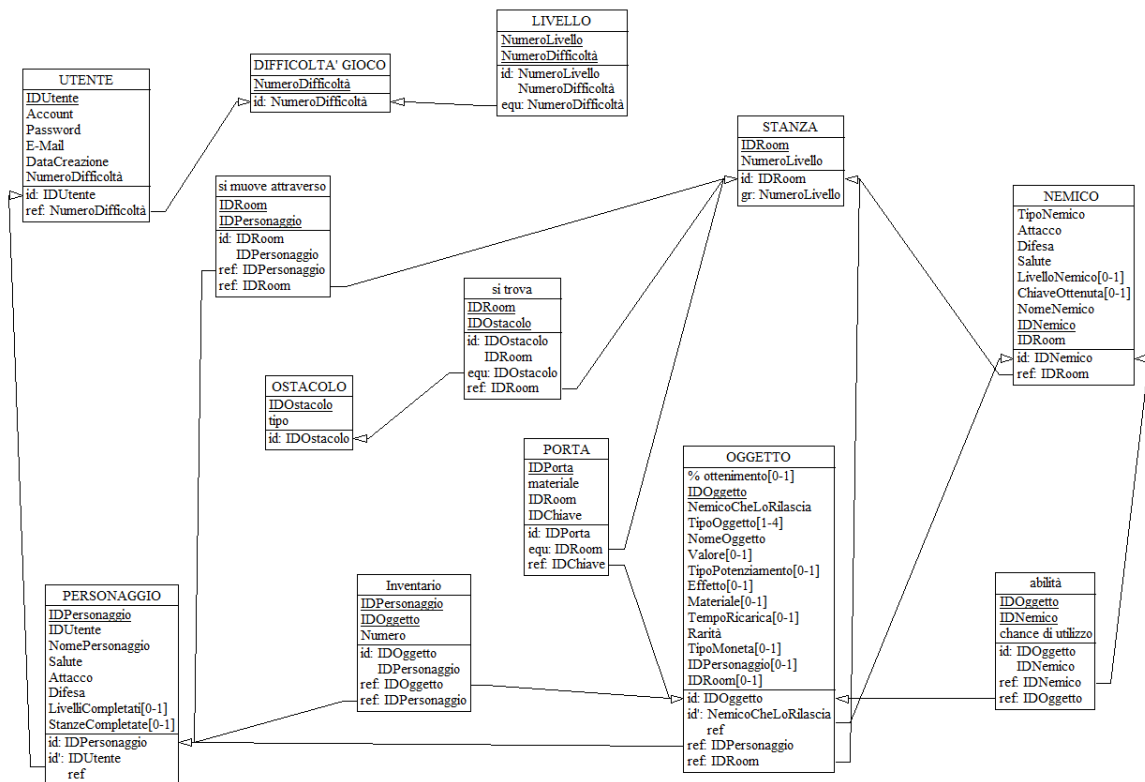
- **CREAZIONE** (lega Utente e Personaggio)
Situazione analoga al caso precedente.

Personaggio (IDPersonaggio, IDUtente, NomePersonaggio, Attacco, Difesa, Salute, LivelliCompletati*, StanzeCompletate*)

FK:IDUtente REFERENCES Utente

3.7 Schema Relazionale finale

A seguito di tutti i raffinamenti e delle traduzioni eseguite, riportiamo l'immagine dello schema relazionale finale.



Schema logico

3.8 Traduzione delle Operazioni in Query SQL

All'interno di questa sezione, tradurremo le operazioni precedentemente considerate ed interrogheremo il nostro database tramite la formulazioni di query in linguaggio SQL, in modo da reperire il corretto risultato.

In grassetto sono evidenziate le parole chiave del linguaggio SQL mentre il simbolo '?' verrà sostituito con l'identificatore delle specifiche entità che si vorranno prendere in esame.

- **Visualizzare l'inventario del personaggio e le sue statistiche correnti**

```

SELECT o.*, p.Attacco, p.Difesa, p.Salute, p.LivelliCompletati, p.StanzeCompletate
FROM personaggio p, inventario i, oggetto o
WHERE o.IDOggetto = i.IDOggetto
AND o.IDPersonaggio = p.IDPersonaggio
AND p.IDPersonaggio = ?;
    
```

- **Visualizzare le stanze in cui si può apprendere una determinata abilità**

```

SELECT s.IDRoom
FROM stanza s, oggetto o
WHERE o.TipoOggetto = 'Abilità'
AND o.IDRoom = s.IDRoom
AND o.NomeOggetto = Abilità?;
    
```

- **Nemici presenti in una determinata stanza e le loro statistiche**

```
SELECT n.IDNemico, n.TipoNemico, n.NomeNemico, n.Attacco, n.Difesa, n.Salute,
n.LivelloNemico
FROM nemico n, stanza s
WHERE s.IDRoom = n.IDRoom
AND s.IDRoom = ?
GROUP BY n.IDNemico;
```

- **Abilità ottenibili e relative statistiche dei boss presenti nei 4 livelli della difficoltà scelta**

```
SELECT n.NomeNemico, n.Attacco, n.Difesa, n.Salute, o.NomeOggetto as
NomeAbilità, o.Effetto as 'Effetto(si/no)', o.Rarità
FROM livello l, 'difficoltà gioco' d, nemico n, oggetto o, stanza s
WHERE u.IDUtente = ?
AND d.NumeroDifficoltà = u.NumeroDifficoltà
AND l.NumeroDifficoltà = d.NumeroDifficoltà
AND s.NumeroLivello = l.NumeroLivello
AND n.IDRoom = s.IDRoom
AND n.TipoNemico = 'boss'
AND o.NemicoCheLoRilascia = n.IDNemico
AND o.TipoOggetto = 'Abilità'
GROUP BY n.IDNemico;
```

- **Visualizzare quante porte si possono aprire con le chiavi disponibili in inventario e quelle presenti in una data stanza**

```
SELECT MAX (numeroPorteInventario) as numeroPorteInventario,
MAX (numeroPorteStanza) as numeroPorteStanza
FROM ( SELECT COUNT (distinct pt.IDPorta) as numeroPorteInventario, 0 as
numeroPorteStanza
FROM personaggio p, oggetto o, inventario i, stanza s, utente u, porta pt
WHERE p.IDUtente = ?
AND i.IDPersonaggio = p.IDPersonaggio
AND o.IDOggetto = i.IDOggetto
AND o.TipoOggetto = 'Chiave'
AND pt.IDChiave = o.IDOggetto
AND s.IDRoom
GROUP BY pt.IDChiave
UNION
SELECT 0 as numeroPorteInventario, COUNT(distinct pt.IDPorta) as
numeroPorteStanza
FROM porta pt, stanza s, oggetto o
WHERE s.IDRoom = ?
AND o.IDRoom = s.IDRoom
AND o.TipoOggetto = 'Chiave'
AND o.IDPersonaggio IS NULL
AND o.IDOggetto = pt.IDChiave
```


GROUP BY pt.IDChiave) as subQry;

- **Visualizzare stanza in cui si trova il giocatore, numero di stanze da lui completate e difficoltà selezionata**

```
SELECT u.NumeroDifficoltà, p.StanzeCompletate, s.IDRoom as
numeroStanzaAttuale, s.NumeroLivello as LivelloAttuale
FROM utente u, personaggio p, 'si muove attraverso' sma, stanza s
WHERE u.IDUtente = ?
AND u.IDUtente = p.IDUtente
AND sma.IDPersonaggio = p.IDPersonaggio
AND s.IDRoom = sma.IDRoom;
```

- **Visualizzare oggetti con rarità massima in un dato livello**

```
SELECT distinct o.IDOggetto, o.TipoOggetto, o.Rarità, o.IDRoom
FROM oggetto o, livello l, stanza s
WHERE l.NumeroLivello = ?
AND s.NumeroLivello = l.NumeroLivello
AND o.IDRoom = s.IDRoom
AND o.IDPersonaggio IS NULL
HAVING o.Rarità = (SELECT MAX(o.Rarità)
FROM oggetto o, livello l, stanza s
WHERE l.NumeroLivello = ?
AND s.NumeroLivello = l.NumeroLivello
AND o.IDRoom = s.IDRoom);
```

- **Oggetti ottenibili dai nemici comuni dopo averli sconfitti nella stanza corrente**

```
SELECT n.IDNemico, n.NomeNemico, o.TipoOggetto, o.NomeOggetto,
o.%% ottenimento
FROM oggetto o, nemico n, utente u, personaggio p, stanza s,
'si muove attraverso' sma
WHERE u.IDUtente = ?
AND u.IDUtente = p.IDUtente
AND sma.IDPersonaggio = p.IDPersonaggio
AND s.IDRoom = sma.IDRoom
AND n.IDRoom = s.IDRoom
AND n.TipoNemico = 'comune'
AND o.NemicoCheLoRilascia = n.IDNemico
AND o.IDPersonaggio IS NULL;
```

Capitolo 4

Progettazione dell'applicazione

4.1 Descrizione architettura dell'applicazione realizzata

Per la gestione del nostro database, abbiamo sviluppato un'applicazione realizzata in linguaggio Java in modo da rendere possibile l'esecuzione e la visualizzazione delle varie operazioni descritte nel corso della trattazione.

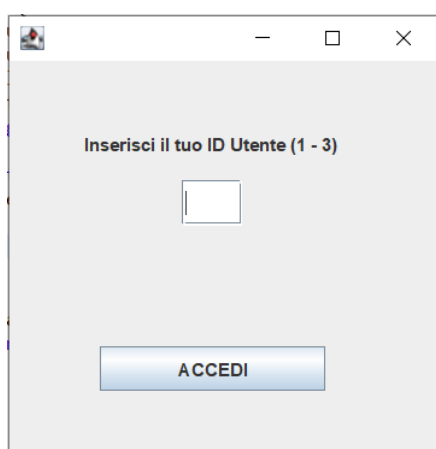
Abbiamo gestito l'approccio verso il nostro database tramite l'utilizzo di JDBC, inserendo come libreria esterna facilitativa Apache Commons DbUtils. Il database risiede in locale e utilizza MySQL come DBMS.

Per ogni tabella rappresentata all'interno del database è stata creata una rispettiva classe Java ed essa si trova nella nostra applicazione all'interno del package model.

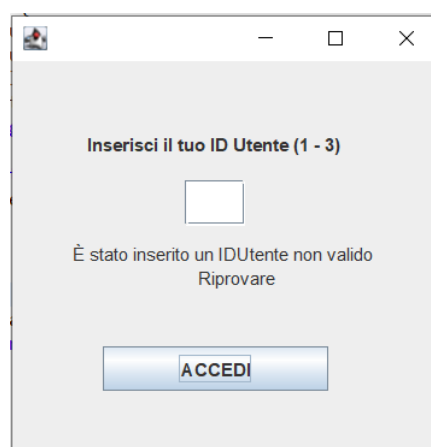
Per quanto riguarda l'accesso al database, ad ogni account vengono associati un IDUtente ed una password univoci: all'avvio dell'applicazione viene richiesto però solo l'inserimento del proprio IDUtente come conferma della corretta corrispondenza tra account e password.

Per aumentare ulteriormente la sicurezza dei propri dati, si potrebbe pensare di implementare un login vero e proprio con inserimento dell'account e relativa password, in quanto essi vengono memorizzati nel database.

Una volta effettuato l'accesso, si apre la schermata principale dell'applicazione dalla quale è possibile eseguire le operazioni descritte.



Schermata di accesso



Schermata di accesso con errore nell'inserimento dell'IDUtente

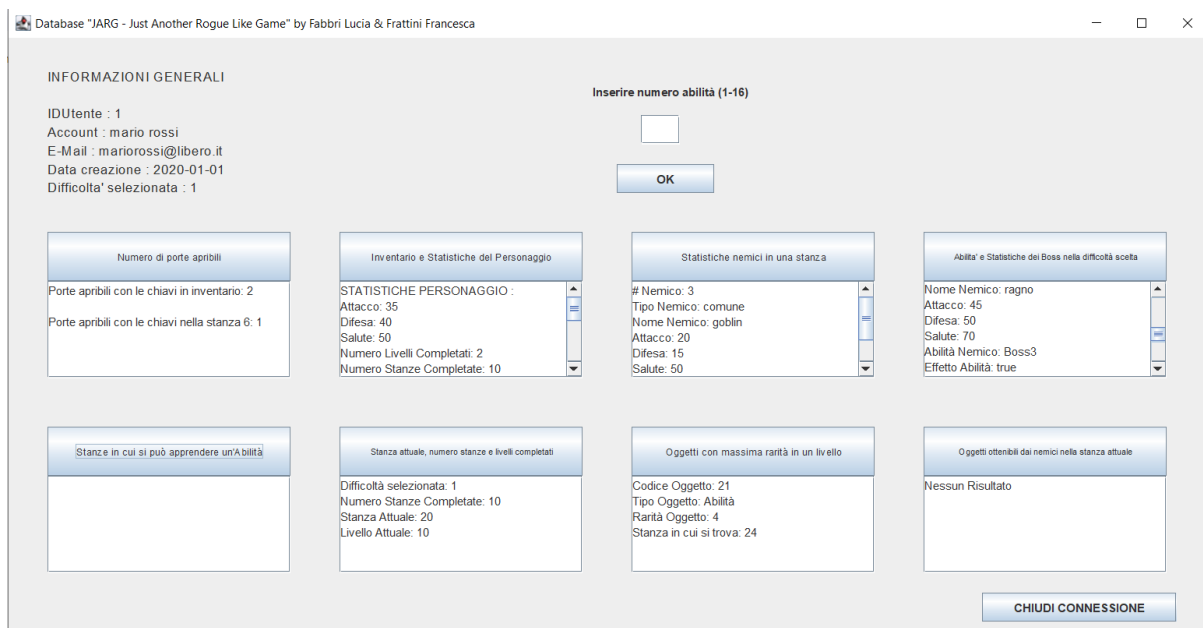
Nella schermata principale, in alto a sinistra, troviamo subito riassunte le informazioni generali riguardanti l'utente che ha effettuato l'accesso.

In corrispondenza di ogni query disponibile abbiamo inserito i relativi pulsanti, i quali, una volta premuti, renderanno visibile il risultato di ogni specifica operazione nell'area sottostante.

Nell'immagine riportata abbiamo appositamente visualizzato una query con risultato vuoto, che funge da esempio nel caso in cui nel database non siano state inserite informazioni per soddisfare la richiesta.

Nel caso in cui una query richieda l'inserimento di un campo da parte dell'utente, all'interno della finestra comparirà una specifica area per permettere l'inserimento dei dati da tastiera, come si può notare nell'immagine.

Al termine del lavoro, è possibile chiudere il database e l'applicazione attraverso l'apposito bottone situato in basso a destra.



Schermata principale dell'applicazione