

# **PROGETTO DI BASI DI DATI**

ANNO ACCADEMICO 2017/2018

Giacomo Barzon Matricola: 1143164

Francesco De Filippis Matricola: 1143408

## Abstract

L'azienda Blozzard Entertainment necessita una base di dati per gestire le informazioni del suo nuovo progetto che consiste in un MMORPG (Gioco di ruolo online).

Ogni utente deve avere un proprio profilo contenente le sue informazioni personali necessario per la registrazione al gioco.

Ciascun utente potrà possedere uno o più personaggi all'interno del gioco. Ciascun personaggio sarà descritto con delle caratteristiche proprie, avrà un suo inventario in cui potrà gestire i suoi oggetti personali.

Oltre ai personaggi associati a ciascun utente, ci potranno essere altri tipi di entità anch'esse personaggi gestite dall'intelligenza artificiale, per esempio nemici o neutrali. Alcuni di questi potranno condividere tutte le caratteristiche dei personaggi associati agli utenti.

Si vuole inoltre tenere traccia delle missioni che gli utenti possono svolgere con i loro personaggi, alcune delle quali assegnate dall'intelligenza artificiale.

Sarà necessario tenere traccia del luogo in cui verranno svolte le missioni e dove si potranno trovare le entità della IA.

Ogni personaggio avrà la possibilità di avere una creatura domabile con cui spostarsi all'interno del mondo di gioco.

## Analisi dei requisiti

Si vuole realizzare una base di dati relativa ad un videogioco online che gestisca le informazioni riguardanti gli utenti, i personaggi ad essi associati e il modo in cui interagiscono con il mondo di gioco. In particolare si vuole conoscere:

Ciascun utente avrà un proprio profilo necessario per accedere al gioco, ed ogni profilo deve memorizzare le informazioni personali del giocatore:

- Indirizzo e-mail che lo identifica univocamente
- Nickname
- Nome
- Cognome
- Sesso
- Data di nascita

Ad ogni profilo possono essere associati uno o più personaggi i quali avranno:

- Un numero progressivo
- Un inventario il quale può contenere
  - Oggetti (armi, armature, consumabili, ecc.)
  - Oggetti attualmente equipaggiati
- Progresso(punti esperienza attuali, punti spendibili, ecc.)
- Classe di cui si conosce
  - Nome
  - Descrizione

- Caratteristiche le quali sono composte di
  - Vita
  - Forza
  - Destrezza
  - Altre...
- Le missioni completate
- Denaro disponibile
- Creature domate

Nel gioco saranno presenti entità controllate dall'intelligenza artificiale i quali sono anch'essi personaggi e in quanto tali potranno condividere le caratteristiche principali dei giocatori, in particolare ci saranno due tipi di IA, ovvero quella neutrale e quella nemica. Dell'IA nemica si vuole sapere se è domabile e che oggetti può lasciare al giocatore una volta sconfitta, di quella neutrale si vuole sapere quali oggetti vende e le missioni che assegna, e di entrambi i tipi si vuole sapere il luogo in cui si trovano.

Si vuole inoltre tenere traccia degli oggetti di gioco di diversa tipologia ad esempio armi, armature, oggetti consumabili e molti altri, e di questi si vuole tenere traccia delle seguenti informazioni:

- Id che lo identifica univocamente
- Nome
- Descrizione
- Peso
- Prezzo

Esistono varie tipologie di oggetti; ogni classe di oggetti avrà delle caratteristiche aggiuntive proprie:

- Armature
  - Difesa
  - Tipo
- Armi
  - Danno
  - Velocità attacco
  - Tipo
- Consumabili
  - Effetto

Essendo il gioco ricco di luoghi esplorabili è necessario conoscere i principali luoghi della mappa in cui sono ubicate le missioni che sono descritti con:

- Codice che lo identifica univocamente
- Nome
- Descrizione

All'interno del gioco è possibile svolgere delle missioni delle quali si vuole tener traccia di:

- Id che la identifica univocamente
- Nome della missione
- IA Neutrale (eventuale) che l'ha assegnata
- Livello minimo
- Ubicazione
- Premio missione (oggetti)

# Progettazione Concettuale

## Descrizione delle classi

La classe Profilo si occupa di gestire tutte le informazioni riguardo agli utenti che si registrano in modo da poter essere identificati all'interno del videogioco (solitamente il servizio viene erogato tramite un client che gestisce i giochi presenti).

- **Profilo**

- Nickname: string
- Email: string
- Nome: string
- Cognome: string
- Sesso: char
- Data di Nascita: date

Personaggio Utente ha il compito di associare ad ogni utente uno o più personaggi da poter utilizzare a scelta durante una partita. Ogni utente infatti può gestire i suoi personaggi facendoli crescere con i punti esperienza.

- **Personaggio Utente**

- Numero: auto-increment <PK>
- Esperienza corrente: int
- Esperienza prossimo livello: int
- Livello: int
- Punti spendibili: int

Classe tiene traccia delle informazioni riguardanti ai tipi di personaggi che ci possono essere nel gioco, ogni personaggio utente avrà una classe associata.

- **Classe**

- Id: auto-increment <PK>
- Nome
- Descrizione

IA è essenzialmente un personaggio ma è gestito dall'intelligenza artificiale, quindi ha delle caratteristiche che lo specializzano.

- **IA**

- Id: auto-increment <PK>
- Nome: string
- Livello: int

Ne esistono di due tipi:

- **IA neutrale**

- Nome
- Livello

- **IA nemica**

- Nome
- Livello
- Domabile: bool

Caratteristiche è una classe che si occupa di tenere traccia dei punti assegnati a ciascuna proprietà del personaggio, tenendo quindi traccia del suo progresso.

- **Caratteristiche**

- Id: auto-increment <PK>
- Forza: int
- Destrezza: int
- Vita: int

Oggetto è una classe che contiene le informazioni condivise da tutte le tipologie di oggetti.

- **Oggetto**

- Id: auto-increment <PK>
- Nome: string
- Descrizione: string
- Peso: double
- Prezzo: double

Arma è un sottotipo di oggetto

- **Arma**

- Tipo: string
- Velocità di attacco: double
- Danno: double

Armatura è un sottotipo di oggetto

- **Armatura**

- Tipo: enum {Testa, Petto, Gambe, Piedi, Mani}
- Difesa: double

Consumabile è un sottotipo di oggetto

- **Consumabile**

- Effetto: string
- Tipo enum {Vita, Forza, Destrezza, Resistenza}

Missione è una classe che si occupa di memorizzare ciascuna missione descrivendo le caratteristiche essenziali.

- **Missione**

- Id: auto-increment
- Titolo: string
- Descrizione: string
- Livello necessario: int

Luogo è una classe che gestisce i luoghi presenti sulla mappa di gioco, in particolare l'ubicazione di missioni dell'intelligenza artificiale.

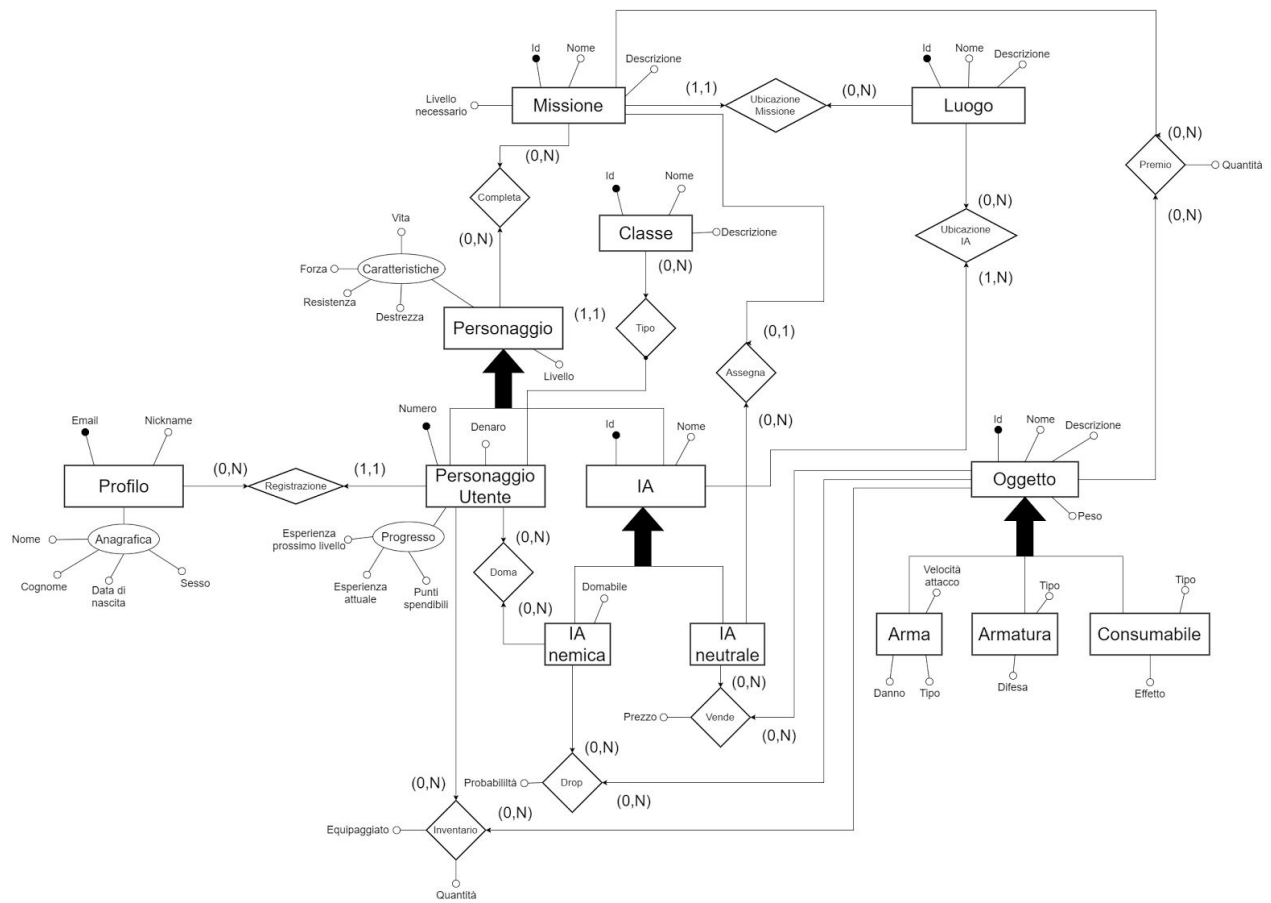
- **Luogo**

- Id: auto-increment
- Nome: string
- Descrizione: string

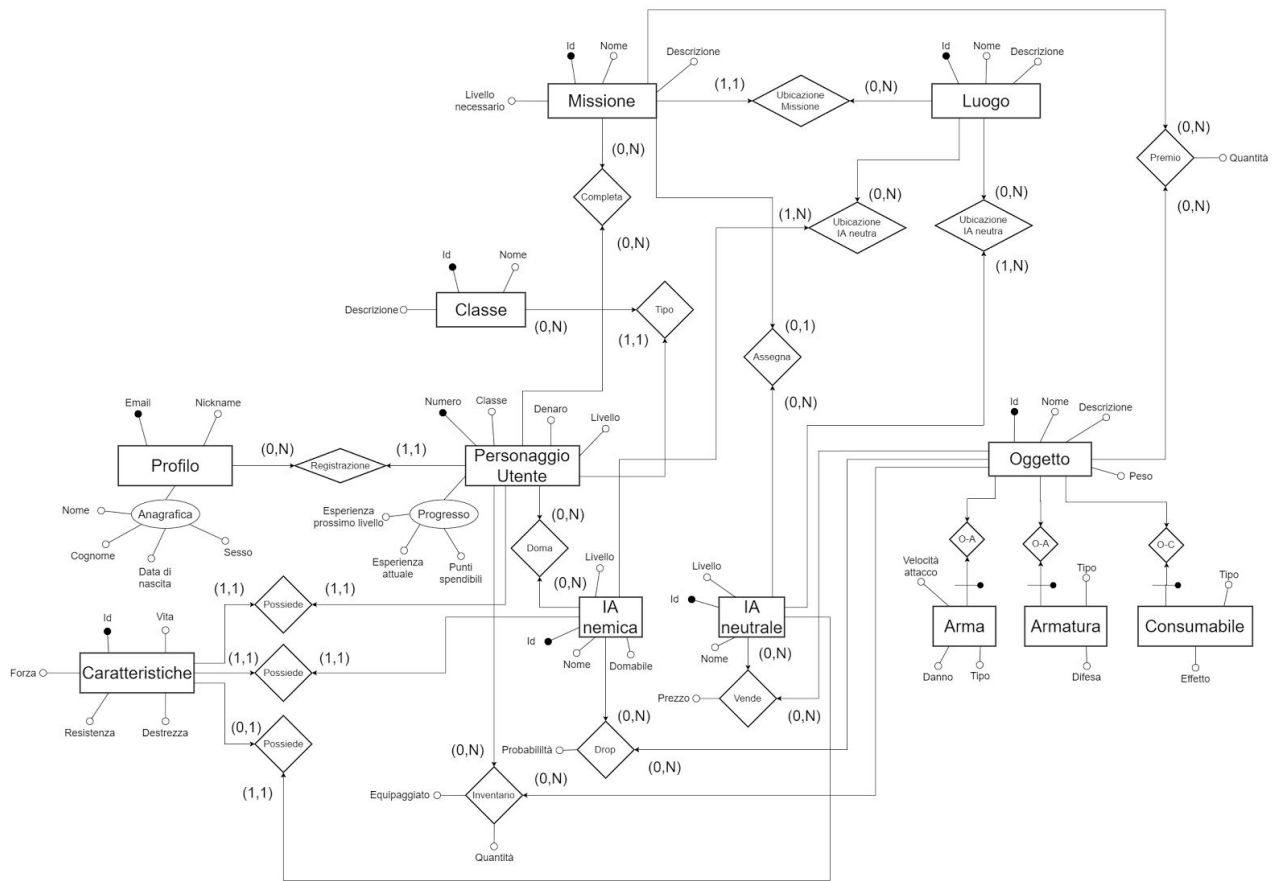
## Associazioni

- **Profilo – Personaggio Utente (Registrazione) N:1**
  - Ogni profilo può avere uno o più personaggi utente
  - Un personaggio utente deve appartenere ad un solo profilo
- **Personaggio Utente – Oggetto (Inventario) N:N**
  - Un personaggio utente possono appartenere uno o più oggetti
  - Un oggetto può essere posseduto da uno o più personaggi utente
- **Personaggio Utente – Classe(Tipo) 1:N**
  - Ciascuna classe può essere un tipo per uno o più personaggi utente
  - Un personaggio utente ha una sola classe
- **Personaggio Utente – Caratteristiche(Tipo) 1:1**
  - Ciascun personaggio utente possiede un solo set di caratteristiche
  - Un set di caratteristiche appartiene ad un solo personaggio utente
- **Personaggio Utente – IA nemica (Doma) N:N**
  - Un personaggio utente può domare uno o più IA nemiche
  - Una IA nemica può essere domata da uno o più personaggi
- **Personaggio Utente – Missione (Svolge) N:N**
  - Ogni personaggio utente può svolgere una o più missioni
  - Ogni missione può essere svolta da uno o più personaggi utente
- **IA nemica – Oggetto (Drop) N:N**
  - Una IA nemica lascia uno o più oggetti con uno specifico fattore di probabilità
  - Un oggetto può essere lasciato da una o più IA nemiche
- **IANemica – Caratteristiche(Tipo) 1:1**
  - Ciascuna IA nemica possiede un solo set di caratteristiche
  - Un set di caratteristiche appartiene ad una sola IA Nemica
- **IA neutrale – Oggetto (Vende) N:N**
  - Una IA neutrale può vendere uno o più oggetti ad un determinato prezzo
  - Un oggetto può essere venduto da una o più IA neutrali
- **IANeutrale – Caratteristiche(Tipo) 1:1**
  - Ciascuna IA neutrale possiede un solo set di caratteristiche
  - Un set di caratteristiche può appartenere ad una sola IA neutrale
- **IANeutrale – Missione (Assegna) N:1**
  - Una IA neutrale può assegnare una o più missioni
  - Una missione è assegnata da una sola IA neutrale
- **Missione – Luogo (Ubicazione) 1:N**
  - Una missione è ubicata in un solo luogo
  - In un luogo possono essere ubicate una o più missioni
- **Missione – Oggetto (Premio) N:N**
  - Una missione può assegnare uno o più oggetti come premio
  - Un oggetto può essere assegnato come premio ad una o più missioni
- **Luogo – IA (Ubicazione IA) N:N**
  - In un luogo possono trovarsi uno o più IA
  - Ogni IA può essere ubicato in uno o più luoghi

### Schema E-R

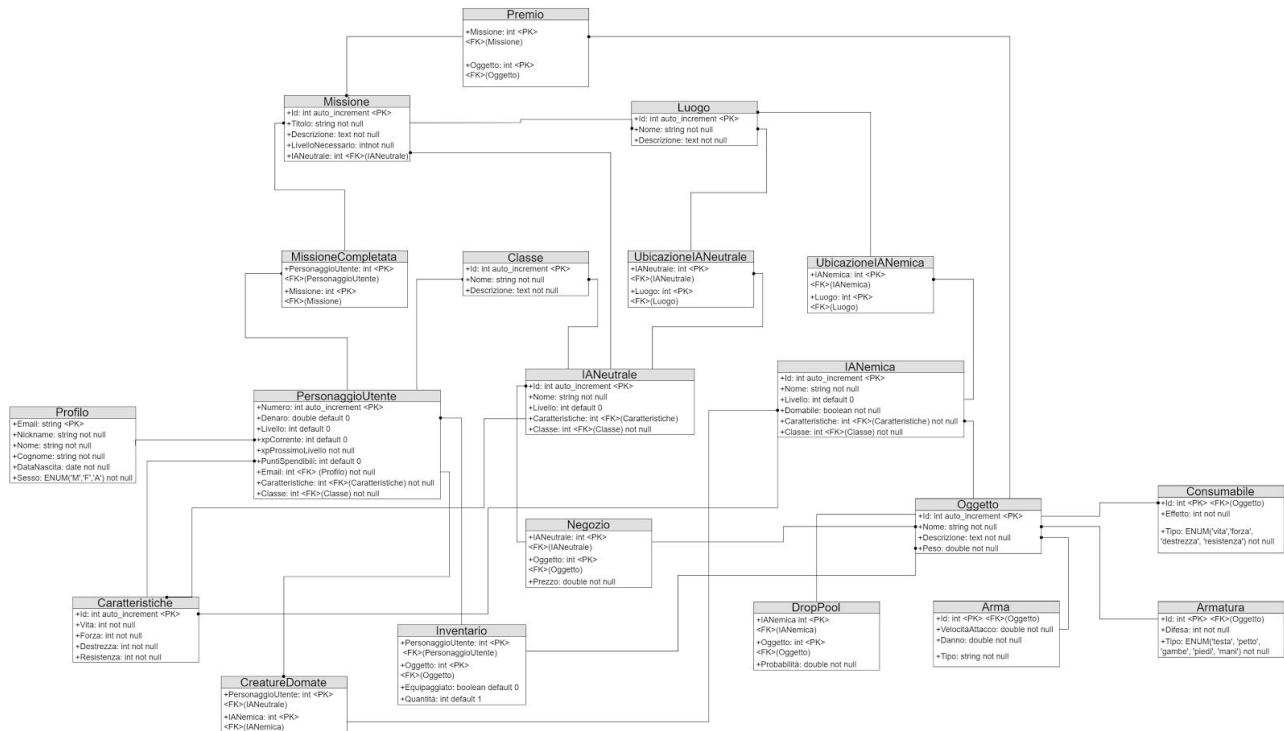


## Schema E-R ristrutturato





## Schema logico



## Schema relazionale

- **Profilo** (Email, Nickname, Nome, Cognome, DataNascita, Sesso)
  - PK (Email)
- **PersonaggioUtente** (Numero, <Email>, Denaro, Livello, xpCorrente, xpProssimoLivello, PuntiSpendibili, <Classe>, <Caratteristiche>)
  - PK (Numero)
  - Email FK(Profilo)
  - Caratteristiche FK(Caratteristiche)
- **Caratteristiche** (Id, Vita, Forza, Destrezza, Resistenza)
  - PK (Id)
- **Classe**(Id, Nome, Descrizione)
  - PK(Id)
- **IANemica** (Id, Nome, Livello, Domabile, <Caratteristiche>)
  - PK (Id)
  - Caratteristiche FK(Caratteristiche)
- **IANeutrale** (Id, Nome, Livello, <Caratteristiche>)
  - PK (Id)
  - Caratteristiche FK(Caratteristiche)
- **Oggetto** (Id, Nome, Descrizione, Peso)
  - PK (Id)
- **Arma** (<Id>, VelocitàAttacco, Danno, Tipo)
  - PK (Id)
  - Id FK(Oggetto)

- **Armatura** (<Id>, Difesa, Tipo)
  - PK (Id)
  - Id FK(Oggetto)
- **Consumabile** (<Id>, Effetto, Tipo)
  - PK (Id)
  - Id FK(Oggetto)
- **Inventario** (<PersonaggioUtente>, <Oggetto>, Equipaggiato, Quantità)
  - PK (PersonaggioUtente, Oggetto)
  - PersonaggioUtente FK(PersonaggioUtente)
  - Oggetto FK(Oggetto)
- **Negozi** (<IANeutrale>, <Oggetto>, Prezzo)
  - PK (IANeutrale, Oggetto)
  - IANeutrale FK(IANeutrale)
  - Oggetto FK(Oggetto)
- **DropPool** (<IANemica>, <Oggetto>, Probabilità)
  - PK (IANemica, Oggetto)
  - IANeutrale FK(IANemica)
  - Oggetto FK(Oggetto)
- **Missione** (<Id>, Titolo, Descrizione, LivelloNecessario, <Luogo>, <IANeutrale>)
  - PK (Id)
  - IANeutrale FK(IANeutrale)
- **Luogo** (<Id>, Nome, Descrizione)
  - PK(Id)
- **UbicazioneIANeutrale** (<IANeutrale>, <Luogo>)
  - PK (IANeutrale, Luogo)
  - IANeutraleFK(IANeutrale)
  - Luogo FK(Luogo)
- **UbicazioneIANemica** (<IANemica>, <Luogo>)
  - PK (IANemica, Luogo)
  - IANeutraleFK(IANemica)
  - Luogo FK(Luogo)
- **MissioneCompletata** (<PersonaggioUtente>, <Missione>)
  - PK (PersonaggioUtente, Missione)
  - PersonaggioUtente FK(PersonaggioUtente)
  - Missione FK(Missione)
- **Premio** (<Missione>, <Oggetto>, Quantità)
  - PK (Missione, Oggetto)
  - Missione FK(Missione)
  - Oggetto FK(Oggetto)
- **CreatureDomate** (<PersonaggioUtente>, <Email>, <IANemica>)
  - PK (PersonaggioUtente, IANemica)
  - PersonaggioUtente FK(PersonaggioUtente)
  - IANemica FK(IANemica)

## Descrizione gerarchie

Ristrutturando lo schema E-R è stata rimossa la generalizzazione **Personaggio** e le entità **PersonaggioUtente** e **IA** hanno ereditato i campi Classe e Livello. Inoltre il campo multivalore Caratteristiche è stato trasformato in relazione e ciascun **PersonaggioUtente** e **IA** sono state collegate a queste nuova entità. In particolare togliendo la generalizzazione **IA**, e facendo ereditare di conseguenza i campi Id e Nome (oltre a quelli di personaggio) alle nuove entità **IANeutrale** e **IANemica**, le caratteristiche sono state collegate a queste due nuove entità.

La generalizzazione della classe **Oggetto** invece è stata rimossa introducendo così dei collegamenti con delle nuove entità per suddividere in modo più ordinato le varie classi di oggetti, rendendo lo schema più modulare e facile da estendere. Le nuove classe di suddivisione (**Arma**, **Armatura**, **Consumabile**) condividono la chiave primaria di **Oggetto**.

## Considerazioni e Vincoli non catturati dallo schema

Il campo **Tipo** di **Arma** non è enumerativo a differenza di **Armatura** e **Consumabile** in quanto in un secondo momento aggiungere nuovi tipi di arma necessiterebbe di una modifica a quel campo di tipo enum, cosa che non accade per le altre due entità già citate in quanto hanno un loro tipo che è già fissato a quello attualmente esistente.

In **Inventario** non ci possono essere oggetti con quantità pari a 0, in quanto non appena viene inserito un nuovo oggetto nell'inventario, questo avrà la quantità di default a 1.

Lo stesso vincolo si ripresenta in **Premio**.

## Implementazione

### Query (Viste)

1. **AlmenoDueMissioni**: Email e numero di tutti i personaggi che hanno completato almeno due missioni.

```
+-----+-----+
| email           | numero |
+-----+-----+
| sample1@lol.com |      1 |
| sample6@lol.com |      5 |
| sample8@lol.com |      3 |
+-----+-----+
3 rows in set (0.00 sec)
```

2. **IANeutNessunOggVenduto**: Tutti le IANeutrale che non vendono alcun oggetto.

```

+-----+
| id |
+-----+
| 1 |
| 4 |
+-----+
2 rows in set (0.00 sec)

```

3. **PersUtVitaMedia**: Tutti i personaggi utente che hanno la vita almeno pari alla media.

```

+-----+
| numero |
+-----+
| 1 |
| 6 |
| 8 |
| 4 |
| 5 |
+-----+
5 rows in set (0.00 sec)

```

4. **ArmaPremioElencare**: tutti i personaggi utente che hanno almeno un arma che può essere ottenuta come premio da una missione.

```

+-----+
| personaggioUtente |
+-----+
| 1 |
| 3 |
| 4 |
| 5 |
+-----+
4 rows in set (0.00 sec)

```

5. **TutteLeArmi**: Elencare gli utenti che possiedono tutte le armi del gioco.

```

+-----+
| personaggioUtente |
+-----+
| 1 |
| 3 |
+-----+
2 rows in set (0.00 sec)

```

6. **ProfiliTrePersUtente**: Elencare tutti i profili che hanno almeno 3 personaggi utente registrati tutti di classe diversa.

```

+-----+
| email |
+-----+
| sample4@lol.com |
+-----+
1 row in set (0.00 sec)

```

- a. **ProfiliTrePersUtenteBis**: Elencare tutti i profili che hanno almeno 3 personaggi utente registrati tutti di classe diversa.

```

+-----+
| email |
+-----+
| sample1@lol.com |
| sample4@lol.com |
+-----+
2 rows in set (0.00 sec)

```

7. **OggPiuPosseduto**: Oggetto più posseduto dai personaggi utente.

```

+-----+-----+
| nome | oggetto |
+-----+-----+
| spadaL | 1 |
| cura | 4 |
| attacco | 6 |
| stivali | 7 |
| giacca | 9 |
+-----+-----+
5 rows in set (0.00 sec)

```

8. **DenaroNegozioCoppie**: Coppie personaggio utente la neutrale tali che il personaggio utente abbia abbastanza denaro da comprare tutto il negozio della IA Neutrale.

personaggioUtente	iaNeutrale
1	3
2	3
2	5
2	2
4	3
4	5
4	2
6	3
7	3
7	5
7	2
8	3
9	3
9	5
9	2
10	3
10	5
10	2

18 rows in set (0.00 sec)

9. **IALuogoMissione**: IA Neutrali che hanno assegnato missioni in luoghi in cui esse sono ubicate.

iaNeutrale
2
5

2 rows in set (0.00 sec)

10. **ForzaPeso**: Tutti i personaggi Utente i quali hanno il parametro forza inferiore al peso di un'armatura che possiedono, si vuole sapere anche il nome dell'armatura.

numero	nome
3	gambali
3	giacca
4	giacca

3 rows in set (0.00 sec)

11. **MediaOggProfilo**: Media degli oggetti posseduti da ciascun personaggio utente per ciascun profilo (se un profilo non possiede oggetti con nessuno dei suoi personaggi verrà visualizzato NULL).

```

+-----+-----+
| email          | media |
+-----+-----+
| sample1@lol.com | 3.0000 |
| sample4@lol.com | NULL   |
| sample6@lol.com | 3.0000 |
| sample8@lol.com | 7.0000 |
| sample9@lol.com | 3.0000 |
+-----+-----+
5 rows in set (0.00 sec)

```

12. **CarattlANeutrale:** Caratteristiche delle IANeutrali se ci sono, altrimenti NULL.

```

+-----+-----+-----+-----+-----+
| IANeutrale | vita | forza | destrezza | resistenza |
+-----+-----+-----+-----+-----+
|          1 | NULL | NULL  | NULL      | NULL       |
|          3 | NULL | NULL  | NULL      | NULL       |
|          5 | NULL | NULL  | NULL      | NULL       |
|          2 | 15  | 10    | 20        | 21         |
|          4 | 18  | 12    | 24        | 19         |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

## Trigger

1. **CheckEquip:** Quando si ha equipaggiata un'arma del proprio inventario e si prova ad equipaggiarne una nuova il campo equipaggiato di quest'ultima verrà settato a false.
2. **MissionePremio:** Quando un personaggio completa una missione (insert in Missione completate) allora vengono aggiunti in inventario i rispettivi premi delle missioni.
3. **DropPoolCreatDomata:** Quando un personaggio utente doma una creatura, viene aggiunto il suo drop pool all'inventario del personaggio utente.
4. **Gift:** Se un profilo registra un personaggio utente nel giorno del suo compleanno, nell'inventario del nuovo personaggio utente verrà inserito un oggetto Consumabile casuale tra quelli presenti.
5. **MaxPersonaggiUtente:** Impedisce più di 5 personaggi utente per ciascun profilo.

## Funzioni

1. **NOggettiInventario:** Numero di oggetti non equipaggiati di un personaggio utente.

```

MariaDB [frdefili]> select NOggettiInventario(1);
+-----+
| NOggettiInventario(1) |
+-----+
|                      11 |
+-----+
1 row in set (0.00 sec)

```

2. **MissioniLuogo:** Numero di luoghi in cui almeno un personaggio utente ha effettuato missioni.



```
MariaDB [frdefili]> select MissioniLuogo();
+-----+
| MissioniLuogo() |
+-----+
|                5 |
+-----+
1 row in set (0.00 sec)
```

3. **DBStats**: Restituisce una stringa contenente lo “stato” del database (alcune informazioni).

```
MariaDB [frdefili]> select DBStats();
+-----+
| DBStats()
+-----+
| Profili totali: 11
Personaggi Utente totali: 10
Oggetti totali posseduti: 25
Missioni completate totali: 11
|
+-----+
1 row in set (0.00 sec)
```

4. **LuoghiMissioniUtenti**: Numero di personaggi che hanno completato almeno una missione in un dato luogo.

```
MariaDB [frdefili]> select LuoghiMissioniUtenti(2);
+-----+
| LuoghiMissioniUtenti(2) |
+-----+
|                2 |
+-----+
1 row in set (0.00 sec)
```

## Procedure

1. **IALuogo**: Tutte le IANemica e IANeutrale appartenenti ad un determinato luogo, indicando a quale tipo di IA appartengono.

```
MariaDB [frdefili]> call IALuogo(4);
+----+-----+-----+
| id | nome   | IANeutrale |
+----+-----+-----+
| 1  | neut1  | IANeutrale |
| 2  | neut2  | IANeutrale |
| 1  | enemy1 | IANemica   |
+----+-----+-----+
3 rows in set (0.00 sec)
```



2. **OggettoArma**: Inserisce un'arma in Oggetto e in Arma date tutte le informazioni (Nessun Output).
3. **OggettoArmatura**: Inserisce un'armatura in Oggetto e in Armatura date tutte le informazioni (Nessun Output).
4. **OggettoConsumabile**: Inserisce un consumabile in Oggetto e in Consumabile date tutte le informazioni (Nessun Output).
5. **CreaPersonaggioUtente**: Inserisce un nuovo personaggio utente in PersonaggioUtente date i valori delle caratteristiche da creare per quel nuovo personaggio utente, deve essere fornito inoltre l'indirizzo email del Profilo e la classe. La somma delle caratteristiche inoltre non deve superare 50 (Nessun Output).