PROGETTO BASI DI DATI

MARTINELLO GIACOMO - 1097397
MASEVSKI MARTIN - 1123062

progetto consegnato da mmsevsk
caricato su mmasevsk-PR

Abstract

Il mercato dei videogiochi è da sempre in forte crescita sin dai primi anni della loro introduzione. Con l'avanzare degli anni e delle tecnologie vengono pubblicati sempre più videogiochi che migliorando in grafica e complessità di anno in anno necessitano di una maggiore capacità di memoria per la loro distribuzione. Per le persone che vogliono possedere una copia della maggior parte dei videogiochi pubblicati fino ad ora diverrebbe impossibile trovare lo spazio fisico per raccogliere tutti i CD/DVD. Per questo motivo è nato HavenOfGamer o più semplicemente HOG. HOG è una piattaforma che permette a qualsiasi utente registrato di acquistare videogiochi attraverso un sistema di distribuzione digitale: una volta effettuato l'acquisto l'utente non riceverà una copia fisica, ma gli verrà assegnata una chiave di attivazione con la quale potrà attivare il prodotto acquistato che gli permetterà di scaricare il gioco direttamente sul suo hard disk. Ogni prodotto acquistato viene salvato nella base di dati ed associato in modo univoco all'account dell'utente che ha attivato quel prodotto. HOG permette agli utenti di tenersi in contatto tra di loro ed eventualmente giocare insieme tramite la lista amici. In HOG ogni utente può creare anche una lista dei desideri dove può aggiungere tutti i giochi ai quali è interessato ed eventualmente acquistarli in un secondo momento.

Analisi dei requisiti

Si vuole realizzare una base di dati che contenga e gestisca le informazioni relative ad una piattaforma di vendita e distribuzione online di videogiochi. In particolare ci si vuole concentrare sulle relazioni degli utenti con i giochi in loro possesso e le relative statistiche associate ad essi.

Per ogni gioco si vogliono conoscere le seguenti informazioni:

- IdGioco: codice univoco per identificare ogni gioco;
- Nome: nome del relativo videogioco;
- DataRilascio: data in cui il gioco è stato rilasciato;
- Prezzo: prezzo del gioco;
- Pegi: sistema di classificazione di videogiochi europeo suddiviso in fasce d'età in base ai contenuti del gioco (3, 7, 12, 16, 18 anni);
- Lingua: rappresenta le lingue disponibili per il gioco (id che fa riferimento ad un attributo della tabella lingua);
- Requisiti: lista di requisiti hardware per il funzionamento corretto del gioco;
- Valutazione: media dei voti assegnati dagli utenti che possiedono il gioco (da 1 a 5);
- Categoria: attributo usato per raggruppare più giochi con aspetti e caratteristiche in comune;
- CasaProduttrice: id della casa che ha sviluppato il gioco;
- Achievement: tutti gli obiettivi sbloccabili di un certo gioco;
- Tipo: contradistinto da 0, 1 o 2 rappresenta la tipologia di gioco (Gioco normale, DLC, DEMO);

Per ogni utente iscritto invece vogliamo sapere:

- Username: identifica univocamente l'utente e viene usato da quest'ultimo per identificarsi nella piattaforma;
- Password: usata per l'autenticazione;
- Nome: nome dell'utente;
- Cognome: cognome dell'utente;
- Email: email dell'utente usata durante l'iscrizione;
- Età: età dell'utente;

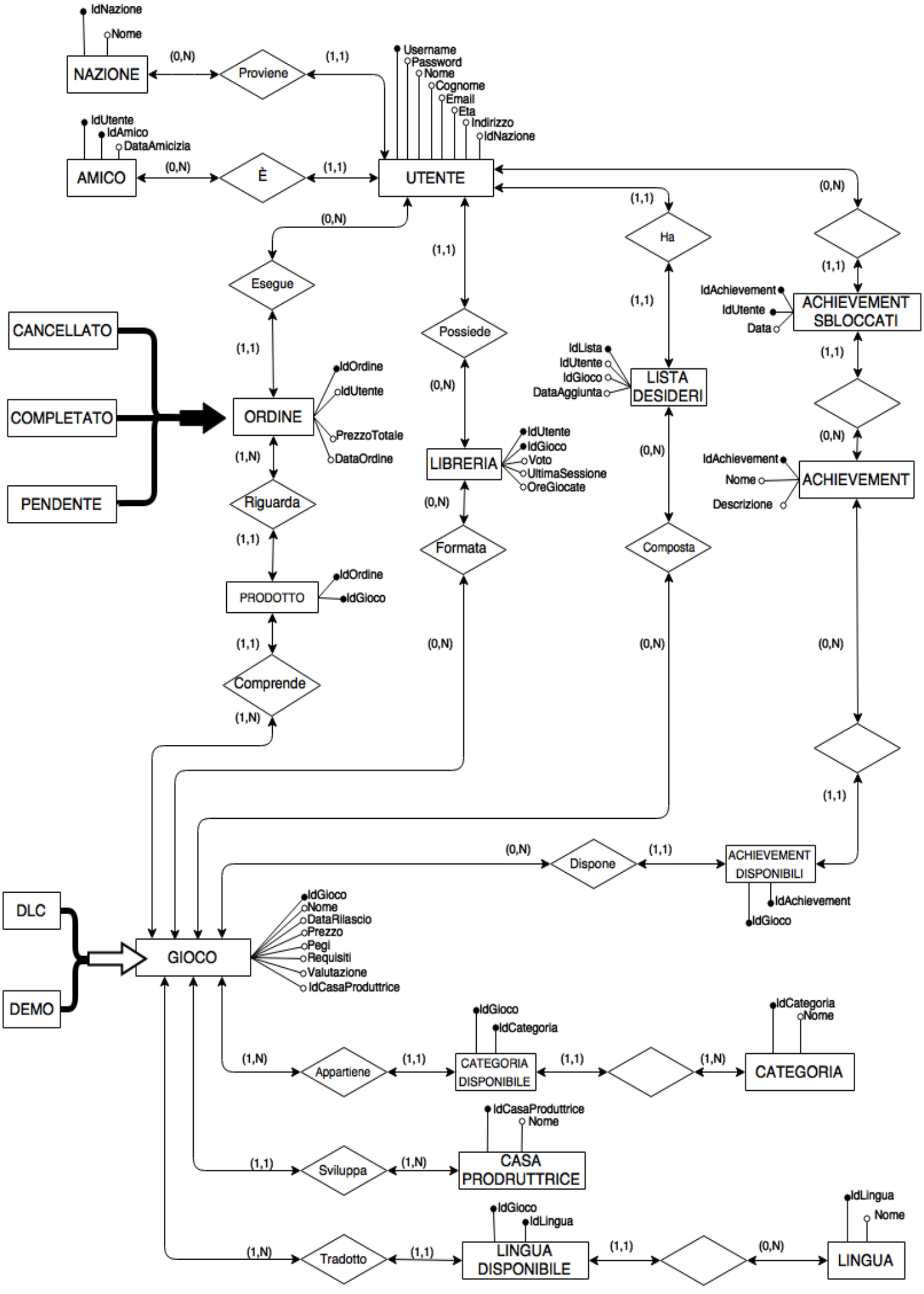
- Indirizzo: indirizzo di fatturazione usato durante l'acquisto composto da: Via, numero civico, città; (abbiamo scelto di mantenere l'indirizzo in un'unica stringa, si sarebbero potute creare singole tabelle per ogni informazione concentrandoci troppo su una parte del database che non è così importante in questo contesto)

Inoltre per ogni utente si vuole conoscere anche la nazionalità e per ognuno di loro si memorizzerà nella base di dati una lista amici (qual'ora l'utente ne avesse all'interno della piattaforma) e una lista dei desideri contenente i giochi ritenuti interessanti dall'utente ma non ancora in suo posseso. In più per ogni utente si avrà la lista di tutti gli achievement da lui sbloccati.

Nella libreria invece vengono memorizzati per ogni giocatore tutti i giochi da lui posseduti ed anche
- UltimaSesione: data nella quale l'utente ha giocato a quel relativo gioco;
- OreGiocate: totale delle ore accumulate dall'utente su un certo gioco;

Come ultima cosa nel database verranno memorizzate le informazione di tutti gli acquisti eseguiti dagli utenti che sono:
- IdOrdine: codice univoco per ogni prodotto;
- IdUtente: username dell'utente che effettua l'acquisto;
- PrezzoTotale: somma dei prezzi dei vari articoli;
- DataOrdine: giorno in cui viene effettuato l'acquisto;
- Stato: stato dell'acquisto che può essere: cancellato se l'utente decide di non volerlo più eseguire durante la transazione, completato quando il pagamento viene effettuato con successo e pendente quando il pagamento deve ancora avvenire.

SCHEMA E-R


IdNazione
Nome
NAZIONE
(0,N)
Proviene
(1,1)
Username
Password
Nome
Cognome
Email
Eta
Indirizzo
IdNazione
IdUtente
IdAmico
DataAmicizia
AMICO
(0,N)
È
(1,1)
UTENTE
(0,N)
(1,1)
Esegue
(1,1)
Possiede
(0,N)
Ha
(1,1)
(0,N)
(1,1)
IdAchievement
IdUtente
Data
ACHIEVEMENT SBLOCCATI
(1,1)
(0,N)
IdAchievement
Nome
Descrizione
ACHIEVEMENT
CANCELLATO
COMPLETATO
PENDENTE
ORDINE
IdOrdine
IdUtente
PrezzoTotale
DataOrdine
(1,N)
Riguarda
(1,1)
PRODOTTO
IdOrdine
IdGioco
(1,1)
Comprende
(1,N)
IdLista
IdUtente
IdGioco
DataAggiunta
LISTA DESIDERI
(0,N)
LIBRERIA
IdUtente
IdGioco
Voto
UltimaSessione
OreGiocate
(0,N)
Formata
(0,N)
(0,N)
Composta
(0,N)
(0,N)
(1,1)
(0,N)
Dispone
(1,1)
ACHIEVEMENT DISPONIBILI
IdAchievement
IdGioco
DLC
DEMO
GIOCO
IdGioco
Nome
DataRilascio
Prezzo
Pegi
Requisiti
Valutazione
IdCasaProduttrice
(1,N)
Appartiene
(1,1)
IdGioco
IdCategoria
CATEGORIA DISPONIBILE
(1,1)
(1,N)
IdCategoria
Nome
CATEGORIA
(1,1)
Sviluppa
(1,N)
IdCasaProduttrice
Nome
CASA PRODRUTTRICE
(1,N)
Tradotto
(1,1)
IdGioco
IdLingua
LINGUA DISPONIBILE
(1,1)
(0,N)
IdLingua
Nome
LINGUA

Entità e Attributi

- **Utente**
  - Username: VARCHAR
  - Password: VARCHAR
  - Nome: VARCHAR
  - Cognome: VARCHAR
  - Email: VARCHAR
  - Età: INT
  - Indirizzo: VARCHAR
  - IdNazione: INT
- **Gioco**
  - IdGioco: INT
  - Nome: VARCHAR
  - DataRilascio: VARCHAR
  - Prezzo: DOUBLE
  - Pegi: VARCHAR
  - Requisiti: VARCHAR
  - Valutazione: DOUBLE
  - IdCasaProduttrice: INT
  - Tipo: INT
- **CasaProduttrice**
  - IdCasaProduttrice: INT
  - Nome: VARCHAR
- **Achievement**
  - IdAchievement: INT
  - Nome: VARCHAR
  - Descrizione: VARCHAR
- **AchievementSbloccati**
  - IdAchievement: INT
  - IdUtente: VARCHAR
  - Data: DATE
- **AchievementDisponibili**
  - IdAchievement: INT
  - IdGioco: INT
- **Libreria**
  - IdUtente: VARCHAR
  - IdGioco: INT
  - UltimaSessione: DATE
  - OreGiocate: INT
- **ListaDesideri**
  - IdLista: INT
  - IdUtente: VARCHAR
  - IdGioco: INT
  - DataAggiunta: DATE
- **Categoria**
  - IdCategoria: INT
  - Nome: VARCHAR
  - Descrizione: VARCHAR
- **CategoriaDisponibile**
  - IdGioco: INT
  - IdCategoria: INT
- **Lingua**
  - IdLingua: INT
  - Nome: VARCHAR
- **LinguaDisponibile**
  - IdGioco: INT
  - IdLingua: INT
- **Amico**
  - IdUtente: VARCHAR
  - IdAmico: VARCHAR
  - DataAmicizia: DATE
- **Nazione**
  - IdNazione: INT
  - Nome: VARCHAR
- **Ordine**
  - IdOrdine: INT
  - IdUtente: VARCHAR
  - PrezzoTotale: DOUBLE
  - DataOrdine: DATE
  - StatoOrdine: VARCHAR
- **Prodotto**
  - IdOrdine: INT
  - IdGioco: INT

Relazioni

Utente - Nazione:
- Ogni utente vive in una sola nazione
- Una nazione può ospitare 0 o più utenti

Utente - Amico:
- Ogni utente può avere 0 o più amici

Utente - Libreria
- Ogni utente possiede una libreria
- Ogni libreria appartiene ad un solo utente

Libreria - Gioco:
- Ogni libreria può contenere più giochi
- Più giochi possono appartenere ad una libreria

Utente - Lista Desideri:
- Ogni utente ha una sola lista desideri
- Una lista desidere può appartenere ad un solo utente

Lista Desideri - Gioco:
- Una lista desideri può avere zero o più giochi al suo interno
- Un gioco può essere in nessuna o più liste desideri

Utente - Achievement Sbloccati:
- Un utente può avere sbloccato zero o più achievement
- Un achievement sbloccato appartiene ad un solo utente

Achievement Sbloccati - Achievement:
- Un achievement sbloccato appartiene ad un achievement
- Un achievement può avere zero o piu achievement sbloccati

Achievement Disponibile - Achievement:
- Un achievement disponibile appartiene ad un achievement
- Un achievement può avere zero o più achievement disponibili

Achievement Disponibile - Gioco:
- Un achievement disponibile appartiene ad un gioco
- Un gioco può avere zero o più achievement disponibili

Utente - Ordine:
- Un utente può eseguire zero o più ordini
- Un ordine può essere eseguito da un solo utente

Ordine - Prodotto:
- Un ordine contiene almeno 1 prodotto
- Un certo prodotto appartiene ad un solo ordine

Prodotto - Gioco:
- Un gioco può essere in piu prodotti
- Un prodotto può essere solo 1 gioco

Gioco - Categoria Disponibile:
- Un gioco può appartenere a più di una categoria disponibile
- Una categoria disponibile è associata ad un solo gioco

Categoria Disponibile - Categoria:
- Una categoria disponibile appartiene ad una categoria
- Una categoria può appartenere a più categorie disponibili

Gioco - Casa Produttrice:

- Un gioco può essere sviluppato da una sola casa produttrice
- Una casa produttrice può sviluppare più di un gioco

Gioco - Lingua Disponibili:

- Un gioco può avere più lingue disponibili
- Una lingua disponibile appartiene ad un solo gioco

Lingua Disponibili - Lingua:

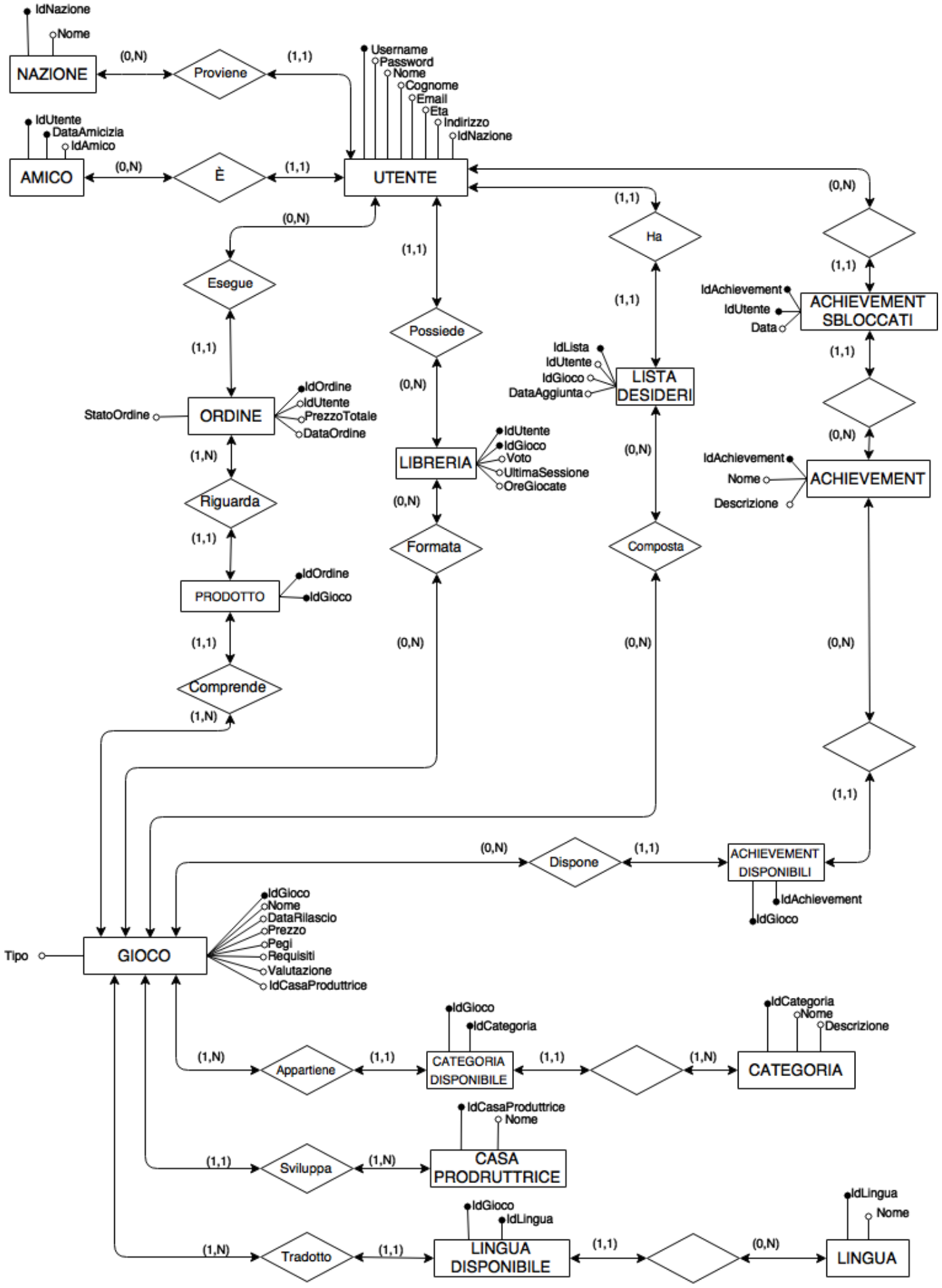
- Una lingua disponibile è collegata con una solo lingua
- Una lingua può appartenere a più lingue disponibili

SCHEMA RELAZIONALE (**Chiave Primaria**, Chiave Esterna)

- **UTENTE** (**Username**, Password, Nome, Cognome, Email, Età, Indirizzo, IdNazione)
- **GIOCO** (**IdGioco**, Nome, DataRilascio, Prezzo, Pegi, Requisiti, Valutazione, Tipo, IdCasaProduttrice)
- **CASAPRODUTTRICE** (**IdCasaProduttrice**, Nome)
- **ACHIEVEMENT** (**IdAchievement**, Nome, Descrizione)
- **ACHIEVEMENTSBLOCCATI** (**IdAchievement**, **IdUtente**, Data)
- **ACHIEVEMENTDISPONIBILI** (**IdAchievement**, **IdGioco**)
- **LIBRERIA** (**IdUtente**, **IdGioco**, Voto, UltimaSessioneData, OreGiocate)
- **LISTADESIDERI** (**IdLista**, IdUtente, IdGioco, DataAggiunta)
- **CATEGORIA** (**IdCategoria**, Nome, Descrizione)
- **CATEGORIADISPONIBILE** (**IdGioco**, **IdCategoria**)
- **LINGUA** (**IdLingua**, Nome)
- **LINGUADISPONIBILE** (**IdGioco**, **IdLingua**)
- **AMICO** (**IdUtente**, **IdAmico**, DataAmicizia)
- **NAZIONE** (**IdNazione**, Nome,)
- **ORDINE** (**IdOrdine**, IdUtente, PrezzoTotale, DataOrdine, StatoOrdine)
- **PRODOTTO** (**IdOrdine**, **IdGioco**)

NB: Nelle tabelle AchievementDisponibili, CategoriaDisponibile, LinguaDisponibile e Prodotto gli attributi che sono chiave primaria sono anche chiavi esterne.

SCHEMA E-R NORMALIZZATO



QUERY

1)Visualizzare la libreria di ogni utente:

```
SELECT lib.IdUtente, g.Nome as Gioco, lib.OreGiocate
FROM Libreria lib, Gioco g
WHERE g.IdGioco=lib.IdGioco;
```

IdUtente	Gioco	OruGiocate
Inte	Mass Effect 3	15
Inte	Enslaved: Odyssey to the West	5
Inte	Mercenaries 2: World in Flames	68
Inte	Too Human	170
Inte	Bayonetta	126
Miba	Call of Duty: World at War	185
Miba	Mass Effect 3	29
Miba	Grand Theft Auto IV	120
Miba	The Orange Box	160
Oba	No More Heroes	184
Oba	BioShock 2	80
Oba	Too Human	92
Omm	Guitar Hero III: Legends of Rock	42
Omm	Silent Hill: Homecoming	155
Omm	Transformers: War For Cybertron	64
Placero	Guitar Hero III: Legends of Rock	73
Placero	Final Fantasy X	28
Placero	The Elder Scrolls V: Skyrim	74
Placero	Shadow Complex	25
Skinti	Indigo Prophecy	33
Skinti	Bulletstorm	50

2)Visualizzare le ore totali giocate dagli utenti:

```
SELECT lib.IdUtente, sum(lib.Oregiocate) as OreGiocateTotali
FROM Libreria lib
GROUP BY lib.IdUtente;
```

IdUtente	OreGiocateTotali
Inte	384
Miba	494
Oba	356
Omm	261
Placero	200
Skinti	83

3)Visualizzare gli Achievement Sbloccati per ogni utente e ordinarli per nome:

```
SELECT asb.IdUtente, a.Nome, a.Descrizione, asb.Data
FROM AchievementSbloccati asb, Achievement a
WHERE asb.IdAchievement=a.IdAchievement
ORDER BY asb.IdUtente;
```

IdUtente	Nome	Descrizione	Data
Inte	Efficient	A glittering gem is not enough	2016-01-03
Miba	Obvious	Please wait outside of the house	2017-01-01
Miba	Efficient	A glittering gem is not enough	2016-01-01
Miba	Tall	Malls are great places to shop I can find everything I need under one roof	2016-01-02
Oba	Visible	The old apple revels in its authority	2016-01-05
Oba	Efficient	A glittering gem is not enough	2016-01-04
Omm	Accurate	The quick brown fox jumps over the lazy dog	2015-03-25
Omm	Impressive	Check back tomorrow I will see if the book has arrived	2016-07-04
Omm	Critical	She did her best to help him	2015-10-05
Omm	Efficient	A glittering gem is not enough	2017-10-17
Omm	Powerful	Hurry	2015-05-01
Placero	Informal	Sometimes it is better to just walk away from things and go back to them later when you're in a better frame of mind	2017-09-27
Placero	Sexual	Wow, does that work	2016-08-16
Placero	Visible	The old apple revels in its authority	2017-07-07
Placero	Several	Yeah, I think its a good environment for learning English	2016-04-28
Skinti	Sexual	Wow, does that work	2016-09-29
Skinti	Powerful	Hurry	2017-08-09
Skinti	Acceptable	I love eating toasted cheese and tuna sandwiches	2016-12-18
Skinti	Critical	She did her best to help him	2015-09-17

4)Visualizzare l'ordine dell'utente "Inte" nel periodo 2010/01/07 e 2017/01/07:

```
SELECT g.Nome as Prodotto, g.Prezzo, o.DataOrdine
FROM Gioco g, Prodotto p , Ordine o
WHERE o.IdOrdine=p.IdOrdine
AND p.IdGioco=g.IdGioco
AND o.DataOrdine BETWEEN '2010-01-07' AND '2017/01/07'
AND o.IdUtente='Inte';
```

Prodotto	Prezzo	DataOrdine
Mercenaries 2: World in Flames	19.81	2011-07-07
Bayonetta	8.15	2011-07-07
Mass Effect 3	34.74	2010-01-07
Enslaved: Odyssey to the West	7.19	2010-01-07
Too Human	34.34	2010-01-07

5)Visualizzare i giochi più giocati dagli amici di Dynante:

```
DROP VIEW IF EXISTS GiochiAmici;

CREATE VIEW GiochiAmici AS
SELECT a.IdAmico as Username, g.Nome as NomeGioco, l.OreGiocate
FROM Amico a, Libreria l, Gioco g
WHERE a.IdUtente='Dynante' and a.IdAmico=l.IdUtente and l.IdGioco=g.IdGioco;
```

```
SELECT Username, NomeGioco, OreGiocate
FROM GiochiAmici
GROUP BY Username
HAVING max(OreGiocate);
```

```
+----------+----------------------------+------------+
| Username | NomeGioco                  | OreGiocate |
+----------+----------------------------+------------+
| Miba     | Call of Duty: World at War |        185 |
| Oba      | No More Heroes             |        184 |
+----------+----------------------------+------------+
```

6)Visualizzare il numero di Achievement disponibili per ogni gioco compresi i giochi senza achievement:

```
SELECT g.IdGioco AS Id, g.Nome, COUNT(a.IdAchievement) AS 'Numero Achievement'
FROM Gioco g LEFT JOIN AchievementDisponibili a ON g.IdGioco=a.IdGioco
GROUP BY g.IdGioco;
```

```
+-----+-----------------------------------------------+--------------------+
| Id  | Nome                                          | Numero Achievement |
+-----+-----------------------------------------------+--------------------+
|   1 | World of Warcraft                             |                  2 |
|   2 | Psychonauts                                   |                  1 |
|   3 | 50 Cent: Blood on the Sand                    |                  2 |
|   4 | Medal of Honor                                |                  0 |
|   5 | Rock Band                                     |                  1 |
|   6 | Resistance 2                                  |                  0 |
|   7 | Grand Theft Auto: San Andreas                 |                  1 |
|   8 | Halo Wars                                     |                  1 |
|   9 | Brothers in Arms: Hells Highway               |                  2 |
|  10 | The Darkness                                  |                  2 |
                                    .
                                    .
                                    .
|  90 | BioShock                                      |                  2 |
|  91 | Final Fantasy XIII                            |                  1 |
|  92 | Ghostbusters: The Video Game                  |                  0 |
|  93 | Too Human                                     |                  1 |
|  94 | Robert Ludlums The Bourne Conspiracy          |                  1 |
|  95 | Army of Two: The 40th Day                     |                  1 |
|  96 | Demons Souls                                  |                  1 |
|  97 | Vanquish                                      |                  1 |
|  98 | Bayonetta                                     |                  3 |
|  99 | Bulletstorm                                   |                  1 |
| 100 | Super Street Fighter IV                       |                  0 |
+-----+-----------------------------------------------+--------------------+
```

7)Visualizza le prime dieci lingue utilizzate piu' frequentemente nei Giochi:

```
SELECT B.Nome, count(A.IdLingua) as Frequenza
FROM LinguaDisponibile A, Lingua B
WHERE A.IdLingua=B.IdLingua group by A.IdLingua
ORDER BY Frequenza DESC
LIMIT 10;
```

Nome	Frequenza
Bengali	7
Northern Min	7
Chewa	7
Gan Chinese	7
Russian	7
Japanese	6
Deccan	5
Serbo-Croatian	5
Dutch	5
Southern Min	5

TRIGGER

1)Trigger per il controllo di eventuali aggiornameti dei prezzi dei giochi

```
DROP TRIGGER IF EXISTS ControlloPrezzo;

DELIMITER $$
CREATE TRIGGER ControlloPrezzo
AFTER UPDATE ON Gioco
FOR EACH ROW
BEGIN
DECLARE num integer(100);
SELECT count(g.IdGioco) INTO num FROM Gioco g WHERE Prezzo='0.00';
IF num>0 THEN
SIGNAL SQLSTATE '02000' SET MESSAGE_TEXT = 'Warning: possibile errore nei prezzi';
END IF;
END;
$$
DELIMITER ;
```

2)Trigger per controllo di un aventuale cambio di password ad esempio password vuota:

```
esempio:[update Utente set Password='' where Username ='Bubbleware';]

DROP TRIGGER IF EXISTS upd_pass;

DELIMITER $$
CREATE TRIGGER upd_pass BEFORE UPDATE ON Utente
FOR EACH ROW
BEGIN
IF new.Password IS NULL OR new.Password = old.Password THEN
SIGNAL SQLSTATE '02000' SET MESSAGE_TEXT = 'Warning password non valida';
END IF;
$$
DELIMITER ;
```

FUNZIONI

1) Dato il nome di un utente calcolare quante ore ha giocato in media a tutti i suoi giochi:

```
DROP FUNCTION IF EXISTS MediaOreGiocate;

DELIMITER $$
CREATE FUNCTION MediaOreGiocate (user varchar(150))
RETURNS integer (100)
BEGIN
DECLARE numGiochi integer (50);
DECLARE somma integer (100);
DECLARE media integer (100);
select count(IdGioco) into numGiochi from Libreria where IdUtente=user;
select sum(OreGiocate) into somma from Libreria where IdUtente=user;
SET media=somma/numGiochi;
return media;
END;
$$
DELIMITER ;
```

2a)Dato il nome di un utente mostrare il valore in euro della sua libreria:

```
DROP FUNCTION IF EXISTS ValoreLibreria;

DELIMITER $$
CREATE FUNCTION ValoreLibreria(utente varchar(100))
RETURNS double(6,2)
BEGIN
DECLARE valore double(6,2);
select sum(g.Prezzo) into valore from Libreria l, Gioco g where
l.IdGioco=g.IdGioco and l.IdUtente=utente;
return valore;
END;
$$
DELIMITER ;
```

2b)Dato il nome di un utente mostrare il valore in euro della sua lista desideri:

```
DROP FUNCTION IF EXISTS ValoreListaDesideri;

DELIMITER $$

CREATE FUNCTION ValoreListaDesideri(utente varchar(100))
RETURNS double(6,2)
BEGIN
```

```
DECLARE valore double(6,2);
select sum(g.Prezzo) into valore from Gioco g,Libreria l where
g.IdGioco=l.IdGioco and l.IdUtente=utente;
return valore;
END;
$$
DELIMITER ;
```

3)Dato il nome di un gioco dire se è economico oppure costoso in base al prezzo medio degli altri giochi:

```
DROP FUNCTION IF EXISTS ValoreGioco;

DELIMITER $$
CREATE FUNCTION ValoreGioco(giocoCercato varchar(50))
RETURNS varchar(50)
BEGIN
DECLARE risultato varchar(50);
DECLARE n integer(10);
DECLARE media double(6,2);
select avg(Prezzo) into media from Gioco;
select Prezzo into n from Gioco where Nome=giocoCercato;
IF n IS NULL THEN SET risultato='Gioco non esistente';
ELSE
IF n<media-10 THEN SET risultato='Economico';END IF;
IF n>=media-10 AND n<=media+10 THEN SET risultato='Nella media';END IF;
IF n>media+10 THEN SET risultato='Costoso';END IF;

END IF;
return risultato;
END;
$$
DELIMITER ;
```