

Relazione progetto

Basi di Dati

VGDB

Carica il progetto:

Bano Mattia (mbano)

matricola 1097541

Cabrera Bernabe Christian (ccabrera)

matricola 1102459

Indice

1. Abstract	3
2. Descrizione dei requisiti	5
3. Progettazione concettuale	9
3.1 Descrizione testuale delle Classi	9
3.1.1 Videogiochi	9
3.1.2 Publisher	10
3.1.3 Developer	10
3.1.4 Direttore	11
3.1.5 Piattaforma	11
3.1.6 Utente	12
3.2 Lista delle associazioni	13
3.2.1 Videogioco - Piattaforma	13
3.2.2 Videogioco - Publisher	13
3.2.3 Videogioco - Developer	13
3.2.4 Videogioco - Direttore	14
3.2.5 Direttore - Developer	14
3.2.6 Console - Publisher	14
3.2.7 Videogioco - Utente	15
4. Progettazione logica	17
4.1 Ristrutturazione ER	17
4.1.1 Eliminazione delle gerarchie	17

4.2 Descrizione testuale dello schema relazionale.....	19
5. Implementazione schema logico	21
5.1 Creazione tabelle	21
5.2 Popolamento tabelle	23
6. Trigger, funzioni, procedure e query.....	31
6.1 Trigger.....	31
6.1.1 Aggiorna voto utente.....	31
6.1.2 Controlla utente	32
6.2 Funzioni.....	33
6.2.1 Voto medio utente	33
6.2.2 N° Giochi per console	34
6.2.3 Livello utente	35
6.3 Procedure.....	36
6.3.1 Modifica voto critica	36
6.4 Query	37
6.4.1 Utenti maggiorenni con almeno 2 voti.....	37
6.4.2 Giochi senza voto utenti.....	38
6.4.3 Videogiochi recenti console.....	38
6.4.4 Developer/Publisher nati prima degli anni 90	39
6.4.5 Giochi Action con votazione alta.....	40
6.4.6 Utenti non votanti o votanti solo giochi PS4	42
6.4.7 N° Giochi console	43

1. Abstract

I videogiochi sono nati negli anni 50 in ambito scientifico. Negli anni 70 nacque la prima console chiamata Magnavox Odyssey, ma i videogiochi divennero fenomeno di massa solo grazie ad Atari. Da quel momento in poi sono stati sviluppati milioni di videogiochi per decine di console diverse.

Per questo motivo si sente la necessità di dare uno strumento di ricerca all'utente.

Il nostro progetto vuole rappresentare una base di dati riguardante i videogiochi.

La base di dati dovrà contenere informazioni quali il nome del videogioco, il genere, la piattaforma o le piattaforme in cui esso è presente, se è single player o multiplayer, la casa editrice, il gruppo developer che lo ha sviluppato e il loro direttore che potrà essere anche direttore dello sviluppo di un videogioco.

Gli utenti potranno così ricercare i videogiochi in base al genere, in quale piattaforma è presente, ecc..

Inoltre, gli utenti potranno anche dare un voto ai videogiochi, rispecchiando la propria opinione.

Questa applicazione potrebbe così essere utilizzata su un sito web ed aiutare gli utenti che visitano la pagina a scoprire nuovi videogiochi a loro sconosciuti e guidarli ad un eventuale acquisto.

2. Descrizione dei requisiti

Si vuole realizzare una base di dati che contenga informazioni riguardanti un videogioco, tra cui publisher, developer, piattaforma e direttore developer/videogioco.

In più si vuole dare la possibilità di salvare informazioni riguardanti utenti che possano votare i videogiochi.

Di un videogioco si vogliono avere le seguenti informazioni:

- titolo
- su quale piattaforma funziona
- in che anno (o la data precisa) è uscito
- genere
- se esistono espansioni e quante sono
- se vi è la possibilità di giocare in multiplayer
- gruppo developer
- publisher
- chi è il direttore dello sviluppo del videogioco

Ogni publisher può pubblicare da uno a più videogiochi ma un videogioco può essere pubblicato da un solo publisher, dei publisher ci interessa:

- nome
- sede principale
- anno di fondazione

I videogiochi funzionano in una piattaforma che può essere una piattaforma digitale su pc o una console. Di una console sono necessarie le seguenti informazioni:

- nome
- anno di uscita
- developer che ha sviluppato la console

Di una piattaforma digitale su pc sono necessarie le seguenti informazioni:

- nome

Un gruppo di developer può realizzare da uno a più videogiochi, ma un videogioco può essere realizzato da un solo gruppo. Inoltre ogni gruppo developer ha un direttore. Di ogni gruppo developer si vuole sapere:

- nome
- anno di nascita
- sede principale
- direttore developer

Ogni direttore può essere a capo di un solo gruppo di developer, può aver seguito più videogiochi e ogni videogioco ha un singolo direttore. Di ogni direttore ci interessa:

- nome
- cognome
- gruppo di developer di cui è direttore

Un utente può votare da zero a più videogiochi, e può dare un solo voto per ogni videogioco. Due utenti non possono avere lo stesso nickname. Di un utente sono necessari:

- nome
- cognome
- nickname
- password
- data di nascita

3. Progettazione Concettuale

3.1 Descrizione testuale delle Classi

3.1.1 Videogiochi

La classe Videogiochi contiene tutte le informazioni principali riguardanti i singoli videogiochi. I vantaggi legati a questa classe è la facilità di accesso ad informazioni secondarie da parte dell'utente interessato.

Attributi

- *Id: int <<PK>>* - identificativo videogioco
- *Titolo: string* - nome del videogioco
- *Genere: string* - genere videogioco
- *Multiplayer: bool* - presenza di multiplayer
- *N° DLC: int* - quantità di espansioni presenti
- *Voto Critica: int* - voto ricevuto dai critici
- *Voto Medio Utenti: int* - voto medio ricevuto da parte degli utenti
- *NomeDeveloper: string* - nome developer
- *NomePublisher: string* - nome publisher
- *CodDirettore: int* - identificativo direttore

3.1.2 Publisher

La classe Publisher raccoglie le informazioni riguardanti le varie case che pubblicano videogiochi.

Attributi

- *Nome: string <<PK>>* - nome Publisher
- *Sede Principale: string* - nome sede principale
- *Anno fondazione: data* - data di nascita della casa produttrice
- *Attivo: bool* - dichiara se è ancora in attività o meno

3.1.3 Developer

La classe Developer contiene le informazioni principali dei vari gruppi di developer di videogiochi.

Attributi

- *Nome: string <<PK>>* - nome gruppo developer
- *Sede Principale: string* - nome sede principale
- *CodDirettore: string* - codice direttore gruppo developer
- *Anno fondazione: date* - anno di fondazione del gruppo developer
- *Attivo: bool* - dichiara se è ancora in attività o meno

3.1.4 Direttore

La classe Direttore contiene le informazioni riguardanti il direttore di un gruppo di developer o di un videogioco. Risulta utile per verificare a quanti progetti partecipa ogni direttore.

Attributi

- *Id: int <<PK>>* - identificatore
- *Nome: string* - nome direttore
- *Cognome: string* - cognome direttore

3.1.5 Piattaforma

La classe Piattaforma contiene le informazioni delle varie piattaforme in cui è presente un videogioco, siano esse console o piattaforme digitali su pc.

Attributi

- *CodPiattaforma: int <<PK>>* - identificatore

Sono definite le seguenti sottoclassi di piattaforma:

1. Console: identifica le diverse console

Attributi

- *Nome: string* - nome della console
- *Tipo: string* - indica se la console è portatile o da casa
- *AnnoNascita: data* - anno di nascita della console

- *NomePublisher: string* - indica chi ha pubblicato la console

2. **PiattaformaDigitale:** identifica le piattaforme digitali su pc in cui vengono pubblicati i videogiochi

Attributi

- *Nome: string* - nome della piattaforma digitale

3.1.6 Utente

La classe Utente contiene le informazioni riguardanti gli utenti registrati che possono votare i videogiochi.

Attributi

- *Nickname: string* - Nickname scelto dall'utente
- *E-mail: string <<PK>>* - mail dell'utente
- *Nome: string* - nome dell'utente
- *Cognome: string* - cognome dell'utente
- *Password: string* - password decisa dall'utente
- *Data di nascita: data* - data di nascita dell'utente

3.2 Lista delle associazioni

3.2.1 Videogioco - Piattaforma: “funziona”

Molteplicità N:M Ogni videogioco può funzionare in una o più piattaforme. In ogni console può funzionare uno o più videogiochi.

Associazione: *totale* da entrambe le parti, poiché una piattaforma per esistere deve far funzionare almeno un videogioco e un videogioco deve funzionare almeno in una console.

Attributi

- *Anno pubblicazione : data* - anno di pubblicazione di un videogioco in una data piattaforma

3.2.2 Videogioco - Publisher: “pubblica”

Moleplicità 1:N Ogni videogioco viene pubblicato da un solo publisher. Ogni Publisher può pubblicare da uno a più videogiochi.

Associazione: *totale* da entrambe le parti, in quanto un videogioco deve essere necessariamente associato ad un publisher, mentre ogni publisher per essere definito tale deve aver pubblicato almeno un videogioco.

3.2.3 Videogioco - Developer: “sviluppa”

Molteplicità 1:N Ogni videogioco è sviluppato da un unico developer. Ogni developer sviluppa uno o più videogiochi.

Associazione: *totale* da entrambe le parti, in quanto un videogioco deve essere necessariamente associato ad un gruppo developer, mentre ogni

gruppo developer per essere definito tale deve aver sviluppato almeno un videogioco.

3.2.4 Videogioco - Direttore: “partecipa”

Molteplicità 1:N Ogni videogioco viene diretto da un unico direttore. Ogni direttore partecipa alla progettazione di uno o più videogiochi.

Associazione: *totale* da entrambe le parti, in quanto un videogioco deve avere necessariamente un direttore, mentre ogni direttore deve aver diretto almeno un videogioco.

3.2.5 Direttore - Developer: “dirige”

Molteplicità 1:1 Ogni direttore dirige un unico gruppo developer. Ogni gruppo è diretto da un unico direttore.

Associazione: *parziale* da direttore a developer, in quanto un direttore potrebbe essere stato licenziato o aver cambiato gruppo developer, *totale* da developer a direttore, in quanto un gruppo developer deve essere diretto da un direttore.

3.2.6 Console - Publisher: “produce”

Molteplicità 1:N Ogni console è prodotta da un unico publisher. Ogni publisher può aver prodotto una o più console.

Associazione: *totale* da console a publisher, in quanto una console deve essere prodotta da un publisher, *parziale* da publisher a console in quanto un publisher può non aver prodotto una console.

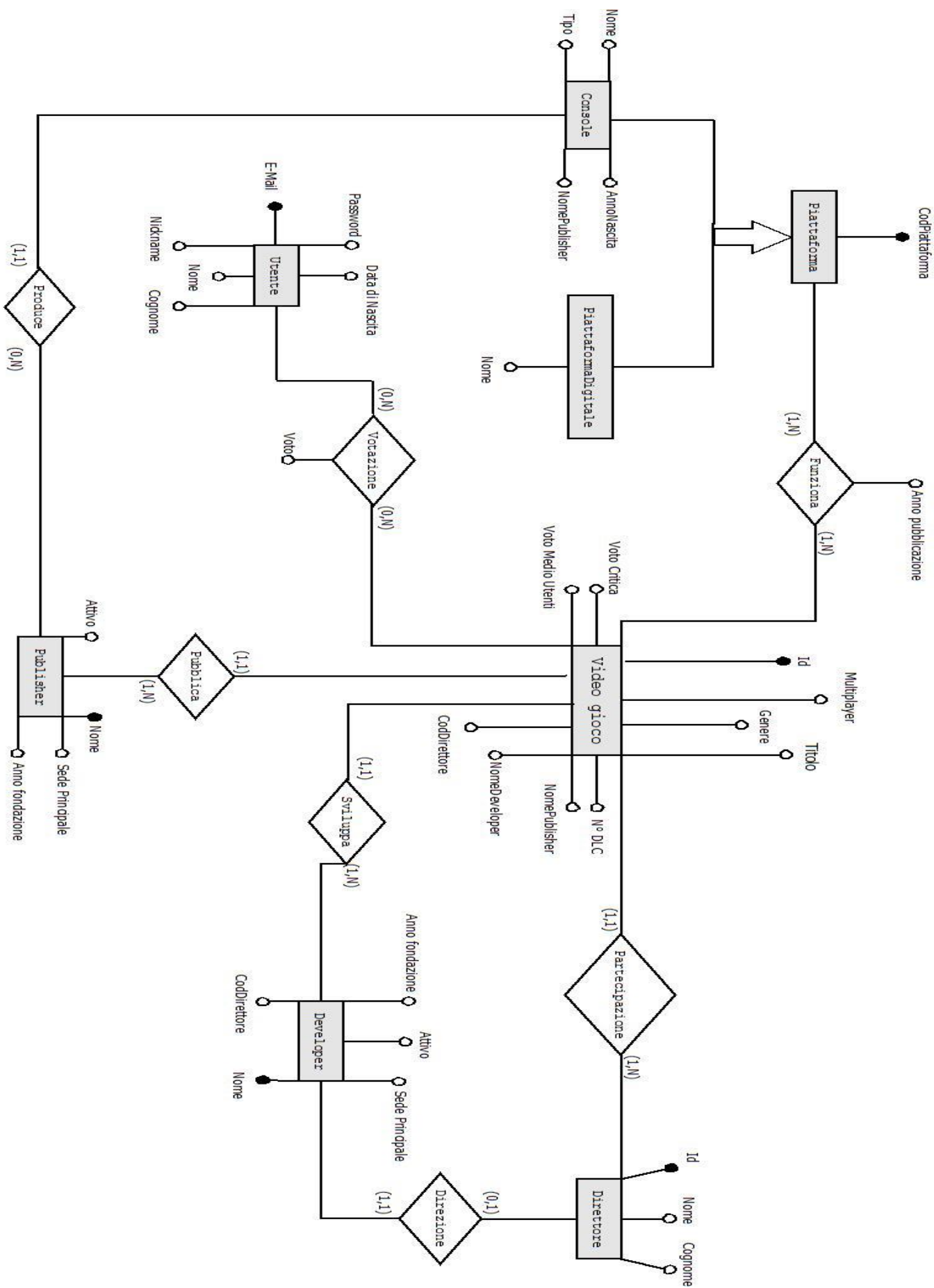
3.2.7 Videogioco - Utente: “votazione”

Molteplicità N:M Ogni videogioco può essere votato da più utenti. Ogni utente può votare più di un videogioco.

Associazione: *parziale* da entrambe le parti, in quanto un videogioco può non essere stato votato da nessun utente e un utente può non aver votato nessun videogioco.

Attributi

- *Voto: int* - voto dell'utente in centesimi

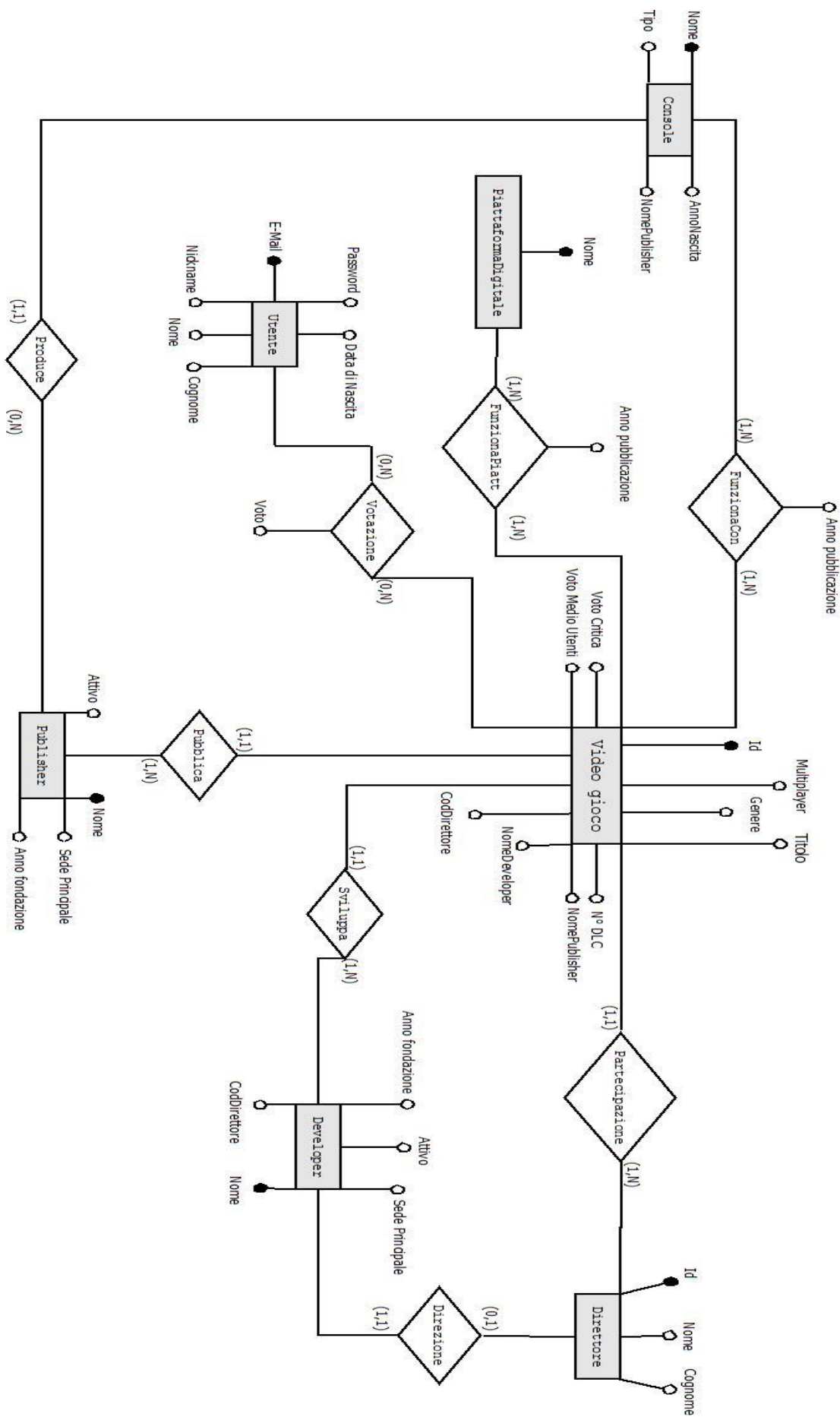


4. Progettazione logica

4.1 Ristrutturazione ER

4.1.1 Eliminazione delle gerarchie

Si è deciso di eliminare la gerarchia accorpendo il genitore della generalizzazione nelle figlie, questa scelta è dovuta al fatto che le due relazioni (console, piattaformaDigitale) vengono accedute separatamente visto che la maggior parte dei videogiocatori utilizzano una console o una piattaformaDigitale su pc per utilizzare i videogiochi.



4.2 Descrizione testuale dello schema relazionale

- Direttore (Id: integer, Nome: string, Cognome: string)
 - PK (Id)
- Publisher (Nome: string, SedePrincipale: string, AnnoFondazione: date, Attivo: boolean)
 - PK (Nome)
- Developer (Nome: string, SedePrincipale: string, CodDirettore: integer, AnnoFondazione: date, Attivo: boolean)
 - PK (Nome)
 - CodDirettore FK Direttore (Id)
- Videogioco (Id: integer, Titolo: string, Genere: string, N° DLC: integer, Multiplayer: boolean, NomePublisher: string, NomeDeveloper: string, CodDirettore: integer, Voto Critica: integer, Voto Medio Utenti: integer)
 - PK (Id)
 - NomePublisher FK Publisher (Nome)
 - CodDirettore FK Direttore (Id)
 - NomeDeveloper FK Developer (Nome)
- Utente (Email: string, Nickname: string, Password: string, Nome: string, Cognome: string, DataNascita: date)
 - PK (Email)
- Votazione (MailUtente: string, IdGioco: integer, Voto: integer)
 - PK (MailUtente, IdGioco)

- MailUtente FK Utente (Email)
- IdGioco FK Videogioco (Id)
- Console (Nome: string, Tipo: enum{"portable", "home"}, AnnoNascita: date, NomePublisher: string)
 - PK (Nome)
 - NomePublisher FK Publisher (Nome)
- PiattaformaDigitale (Nome: string)
 - PK (Nome)
- FunzionaCon (NomeConsole: string, CodGioco: integer, AnnoPubblicazione: date)
 - PK (NomeConsole, CodGioco)
 - NomeConsole FK Console (Nome)
 - CodGioco FK Videogioco (Id)
- FunzionaPiattDig (NomePiattDig: string, CodGioco: integer, AnnoPubblicazione: date)
 - PK (NomePiattDig, CodGioco)
 - NomePiattDig FK PiattaformaDigitale (Nome)
 - CodGioco FK Videogioco (Id)

5. Implementazione schema logico

5.1 Creazione tabelle

```
1  -- Crea la tabella Direttore
2
3  CREATE TABLE Direttore (
4      Id          INT          PRIMARY KEY,
5      Nome        VARCHAR(20)  NOT NULL,
6      Cognome     VARCHAR(20)  NOT NULL
7  );
8
9
10 -- Crea la tabella Publisher
11
12 CREATE TABLE Publisher(
13     Nome          VARCHAR(20) PRIMARY KEY,
14     SedePrincipale VARCHAR(20) NOT NULL,
15     AnnoFondazione DATE,
16     Attivo        BOOLEAN
17 );
18
19
20 -- Crea la tabella Developer
21
22 CREATE TABLE Developer(
23     Nome          VARCHAR(20) PRIMARY KEY,
24     SedePrincipale VARCHAR(20) NOT NULL,
25     CodDirettore  INT,
26     AnnoFondazione DATE DEFAULT NULL,
27     Attivo        BOOLEAN,
28     FOREIGN KEY (CodDirettore) REFERENCES Direttore(Id)
29 );
30
31
32 -- Crea la tabella Videogioco
33
34 CREATE TABLE Videogioco(
35     Id            INT PRIMARY KEY,
36     Titolo        VARCHAR(20) NOT NULL,
37     Genere        VARCHAR(20),
38     N_DLC         INT DEFAULT 0,
39     Multiplayer   BOOLEAN NOT NULL,
40     NomePublisher VARCHAR(20),
41     NomeDeveloper VARCHAR(20),
42     CodDirettore  INT,
```

```

43     VotoCritica          INT DEFAULT 0,
44     VotoMedioUtenti     INT DEFAULT 0,
45     FOREIGN KEY (NomePublisher) REFERENCES Publisher(Nome),
46     FOREIGN KEY (CodDirettore) REFERENCES Direttore(Id),
47     FOREIGN KEY (NomeDeveloper) REFERENCES Developer(Nome)
48 );
49
50
51 -- Crea la tabella Utenti
52
53 CREATE TABLE Utente(
54     Email                VARCHAR(20) PRIMARY KEY,
55     NickName             VARCHAR(20) NOT NULL,
56     Password             VARCHAR(20) NOT NULL,
57     Nome                 VARCHAR(20) NOT NULL,
58     Cognome              VARCHAR(20) NOT NULL,
59     DataNascita          DATE
60 );
61
62
63 -- Crea la tabella Votazione
64
65 CREATE TABLE Votazione(
66     MailUtente           VARCHAR(20),
67     IdGioco              INT,
68     Voto                 INT DEFAULT 0,
69     PRIMARY KEY (MailUtente, IdGioco),
70     FOREIGN KEY (MailUtente) REFERENCES Utente(Email) ON DELETE CASCADE,
71     FOREIGN KEY (IdGioco) REFERENCES Videogioco(Id) ON DELETE CASCADE
72 );
73
74
75 -- Crea la tabella Console
76
77 CREATE TABLE Console(
78     Nome                 VARCHAR(20) PRIMARY KEY,
79     Tipo                 ENUM('Portable', 'Home') NOT NULL,
80     AnnoNascita          DATE NOT NULL,
81     NomePublisher        VARCHAR(20),
82     FOREIGN KEY (NomePublisher) REFERENCES Publisher(Nome)
83 );
84
85
86 -- Crea la tabella PiattaformaDigitale
87
88 CREATE TABLE PiattaformaDigitale(
89     Nome                 VARCHAR(20) PRIMARY KEY
90 );
91
92

```

```

93  -- Crea la tabella FunzionaCon
94
95  CREATE TABLE FunzionaCon(
96      NomeConsole      VARCHAR(20),
97      CodGioco          INT,
98      AnnoPubblicazione DATE DEFAULT NULL,
99      PRIMARY KEY (NomeConsole, CodGioco),
100     FOREIGN KEY (NomeConsole) REFERENCES Console(Nome),
101     FOREIGN KEY (CodGioco) REFERENCES Videogioco(Id) ON DELETE CASCADE
102 );
103
104
105  -- Crea la tabella FunzionaPiattDig
106
107  CREATE TABLE FunzionaPiattDig(
108      NomePiattDig      VARCHAR(20),
109      CodGioco          INT,
110      AnnoPubblicazione DATE,
111      PRIMARY KEY (NomePiattDig, CodGioco),
112      FOREIGN KEY (NomePiattDig) REFERENCES PiattaformaDigitale(Nome),
113      FOREIGN KEY (CodGioco) REFERENCES Videogioco(Id) ON DELETE CASCADE
114 );

```

5.2 Popolamento tabelle

```

1  -- Popolamento tabella Direttore
2
3  INSERT INTO Direttore (Id, Nome, Cognome)
4  VALUES (1,'Guillume','De Fondaumiere'),
5          (2,'Guillemot','Yves'),
6          (3,'Todd','Howard'),
7          (4,'Yoshimi','Yasuda'),
8          (5,'Jenova','Chen'),
9          (6,'Sam','Houser'),
10         (7,'Gabe','Newell'),
11         (8,'Kevin','Bruner'),
12         (9,'Tatsumi','Kimishima'),
13         (10,'Kenzo','Tsujimoto'),
14         (11,'David','Gardner'),
15         (12,'Alex','Garden'),
16         (13,'Matt','Webster'),
17         (14,'Hiroshi','Matsuyama'),
18         (15,'David','Adams'),
19         (16,'Bruce','Straley'),
20         (17,'Michael','Morhaime'),
21         (18,'Masayoshi','Tanimura'),
22         (19,'Satoshi','Oshita'),
23         (20,'Casey','Hudson'),

```



```

24      (21,'Christina', 'Hernandez'),
25      (22,'Fred', 'Dixon'),
26      (23,'Jean', 'Reid'),
27      (24,'Wayne', 'Murphy'),
28      (25,'Alan', 'Cruz'),
29      (26,'Norma', 'Hicks'),
30      (27,'Kevin', 'Harrison'),
31      (28,'Nicole', 'Day'),
32      (29,'Harold', 'Fernandez'),
33      (30,'Edward', 'Banks'),
34      (31,'Thomas', 'Tucker'),
35      (32,'Steven', 'Robinson'),
36      (33,'Kelly', 'Nichols'),
37      (34,'Larry', 'Perry'),
38      (35,'Raymond', 'Phillips'),
39      (36,'Dennis', 'Andrews'),
40      (37,'Ryan', 'Harold');
41
42
43  -- Popolamento tabella Publisher
44
45  INSERT INTO Publisher (Nome, SedePrincipale, AnnoFondazione, Attivo)
46  VALUES ('3DO', 'Redwood City', '1991-04-25', 0),
47          ('Atari', 'Lione', '1972-06-27', 0),
48          ('Bandai Namco', 'Tokyo', '1955-06-01', 1),
49          ('Blizzard ENT', 'Irvine', '1991-02-08', 1),
50          ('EA', 'California', '1982-06-28', 1),
51          ('Nintendo', 'Kyoto', '1889-09-23', 1),
52          ('SEGA', 'Tokyo', '1960-06-03', 1),
53          ('Sony', 'Tokyo', '1993-11-16', 1),
54          ('Telltale', 'California', '2004-06-13', 1),
55          ('THQ', 'California', '1991-05-09', 0),
56          ('Ubisoft', 'Rennes', '1986-03-01', 1),
57          ('Activision', 'Santa Monica', '1979-10-01', 1);
58
59
60  -- Popolamento tabella Developer
61
62  INSERT INTO Developer (Nome, SedePrincipale, CodDirettore, AnnoFondazione, Attivo)
63  VALUES ('Atari', 'Lione', 11, '1972-06-27', 0),
64          ('Bandai Namco', 'Tokyo', 19, '1955-06-01', 1),
65          ('Bethesda', 'Maryland', 3, '2002-01-01', 1),
66          ('Bioware', 'Edmonton', 20, '1995-02-01', 1),
67          ('Blizzard ENT', 'Irvine', 17, '1991-02-08', 1),
68          ('Capcom', 'Osaka', 10, '1979-05-30', 1),
69          ('Criterion', 'Guildford', 13, '1993-01-01', 1),
70          ('CyberConnect2', 'Fukuoka', 14, '1996-02-16', 1),
71          ('From Software', 'Tokyo', 4, '1986-11-01', 1),
72          ('Hal Laboratory', 'Tokyo', 18, '1980-02-24', 1),
73          ('Naughty Dog', 'Santa Monica', 16, '1989-09-09', 1),

```

```

74      ('Nintendo', 'Kyoto', 9, '1889-09-23', 1),
75      ('Quantic Dream', 'Paris', 1, '1997-03-05', 1),
76      ('Relic ENT', 'Vancouver', 12, '1997-05-01', 1),
77      ('Rockstar Games', 'New York', 6, '1998-12-01', 1),
78      ('Telltale', 'California', 8, '2004-06-13', 1),
79      ('Thatgamecompany', 'Los Angeles', 5, '2006-05-15', 1),
80      ('Ubisoft', 'Rennes', 2, '1986-03-01', 1),
81      ('Valve', 'Washington', 7, '1996-08-24', 1),
82      ('Vigil Games', 'Texas', 15, '2013-01-23', 0),
83      ('Bungie Studios', 'Washington', 37, '1991-05-01', 1);
84
85
86  -- Popolamento tabella Videogioco
87
88  INSERT INTO Videogioco (Id, Titolo, Genere, N_DLC, Multiplayer, NomePublisher,
89  NomeDeveloper, CodDirettore, VotoCritica)
90  VALUES (1, 'Mass Effect', 'Action', 1, 0, 'EA', 'Bioware', 20, 89),
91          (11, 'Burnout Paradise', 'Racing', 2, 1, 'EA', 'Criterion', 21, 87),
92          (21, 'Warhammer 40000', 'Tactical', 4, 1, 'THQ', 'Relic ENT', 22, 85),
93          (31, 'Darksiders 2', 'Hack & slash', 2, 0, 'THQ', 'Vigil Games', 23, 75),
94          (41, 'Company of Heroes 2', 'FPS', 3, 1, 'SEGA', 'Relic ENT', 24, 84),
95          (51, 'Walking Dead', 'Adventure', 5, 0, 'Telltale', 'Telltale', 25, 92),
96          (61, 'Bloodborne', 'Action', 1, 1, 'Sony', 'From Software', 26, 92),
97          (71, 'The last of Us', 'Adventure', 1, 1, 'Sony', 'Naughty Dog', 27, 95),
98          (81, 'Overwatch', 'FPS', 0, 1, 'Blizzard ENT', 'Blizzard ENT', 28, 92),
99          (91, 'Zelda', 'Adventure', 0, 0, 'Nintendo', 'Nintendo', 29, 95),
100         (101, 'Super Smash Bros', 'Fighting', 10, 1, 'Nintendo', 'Hal Laboratory', 30, 92),
101         (111, 'Dark Souls 3', 'Action', 0, 1, 'Bandai Namco', 'From Software', 31, 89),
102         (121, 'Naruto UNS3', 'Fighting', 0, 0, 'Bandai Namco', 'Cybercoonnct2', 32, 73),
103         (131, 'Tekken 5', 'Arcade', 0, 1, 'Bandai Namco', 'Bandai Namco', 33, 88),
104         (141, 'Rayman 2', 'Adventure', 0, 0, 'Ubisoft', 'Ubisoft', 34, 90),
105         (151, 'Pacman', 'Action', 0, 0, 'Atari', 'Atari', 35, 71),
106         (161, 'Skate or Die', 'Skateboarding', 0, 0, 'EA', 'Atari', 36, 50),
107         (171, 'Destiny', 'FPS', 3, 1, 'Activision', 'Bungie Studios', 37, 85);
108
109
110  -- Popolamento tabella Utente
111
112  INSERT INTO Utente (Email, Nickname, Password, Nome, Cognome, DataNascita)
113  VALUES ('agilbertm@princeton.', 'agilbertm', 'CN0flx5Dw', 'Angela', 'Gilbert',
114          '1996-01-04'),
115          ('amorris1@earthlink.n', 'amorris1', 'Z3dpyHz8V', 'Andrew', 'Morris',
116          '1993-07-16'),
117          ('apattersonx@fda.gov', 'apattersonx', 'Zb2l3woCI2', 'Alice', 'Patterson',
118          '1995-07-28'),
119          ('bbarnes4@virginia.ed', 'bbarnes4', 'vBzG9us4V', 'Barbara', 'Barnes',
120          '1996-10-17'),
121          ('bmarshallu@uol.com.b', 'bmarshallu', 'QWqWvXKH2Uw', 'Bruce', 'Marshall',
122          '1985-08-31'),
123          ('bmorrisonz@marriott.', 'bmorrisonz', 'Ufn9lQ44Pk', 'Barbara', 'Morrison',

```

124 '1985-09-09'),
 125 ('cford2@bloglines.com', 'cford2', 'XI5a6j17v', 'Carlos', 'Ford', '1985-12-11'),
 126 ('cmccoyo@indiatimes.c', 'cmccoyo', '13nJpxcff', 'Carl', 'Mccoy', '1992-09-23'),
 127 ('cstewart5@constantco', 'cstewart5', 'Qt4hMObof5', 'Carol', 'Stewart',
 128 '1995-05-30'),
 129 ('dbrooksy@youtube.com', 'dbrooksy', '7uuEA2ne2', 'Donald', 'Brooks',
 130 '1994-12-02'),
 131 ('dramirez3@arizona.ed', 'dramirez3', 'HKvn26Mcq3', 'Doris', 'Ramirez',
 132 '1999-01-19'),
 133 ('drileye@topsy.com', 'drileye', '4TylFU6xR', 'Diana', 'Riley', '1999-01-06'),
 134 ('dwoodsv@joomla.org', 'dwoodsv', 'jVrMol7NW5Im', 'Donald', 'Woods',
 135 '1991-09-10'),
 136 ('ehuntp@nyu.edu', 'ehuntp', 'JT7jPSBpnSH', 'Eric', 'Hunt', '1993-11-10'),
 137 ('fdavison@nbcnews.com', 'fdavison', 'vh8A6Q0m8', 'Fred', 'Davis', '1995-04-11'),
 138 ('flawson9@myspace.com', 'flawson9', 'nhLGO7YRtyl', 'Fred', 'Lawson',
 139 '1997-01-05'),
 140 ('fsimsb@bbc.co.uk', 'fsimsb', 'Zt5uPC2n5e32', 'Fred', 'Sims', '1986-12-19'),
 141 ('gweavert@istockphoto', 'gweavert', 'n2b0q9ycW', 'Gloria', 'Weaver',
 142 '1988-04-09'),
 143 ('iparkera@indiatimes.', 'iparkera', 'QjDYbhPm44', 'Irene', 'Parker',
 144 '1987-11-18'),
 145 ('jcampbell10@house.go', 'jcampbell10', 'r5SA0Q2Y', 'Janet', 'Campbell',
 146 '1986-12-25'),
 147 ('jgibsonc@tuttocitta.', 'jgibsonc', 'fT7d9omynKpl', 'Juan', 'Gibson',
 148 '1988-12-23'),
 149 ('jhenryi@seesaa.net', 'jhenryi', 'WJ3hRaEj', 'Julie', 'Henry', '1999-09-24'),
 150 ('jperezj@skype.com', 'jperezj', 'sTLoco5voZA', 'Jimmy', 'Perez', '1998-09-03'),
 151 ('jreynoldsl@cyberchim', 'jreynoldsl', 'nLpUpmqv66em', 'Janet', 'Reynolds',
 152 '1998-02-12'),
 153 ('kweaverg@dot.gov', 'kweaverg', 'Mg9ZxAoZV', 'Katherine', 'Weaver',
 154 '2000-10-25'),
 155 ('kweavers@pagesperso-', 'kweavers', 'EiSjLTR5hvDP', 'Kevin', 'Weaver',
 156 '1988-09-06'),
 157 ('lcunninghamk@tinypic', 'lcunninghamk', '3eyXB4cF', 'Lawrence',
 158 'Cunningham', '1993-02-10'),
 159 ('lfosterw@weebly.com', 'lfosterw', 'wH0LimT8M', 'Lisa', 'Foster', '1990-01-09'),
 160 ('loliver13@webmd.com', 'loliver13', 'KXhm6RbBQrwF', 'Louise', 'Oliver',
 161 '1988-10-11'),
 162 ('lromero15@ox.ac.uk', 'lromero15', 'nR6YkSzApVi0', 'Lawrence', 'Romero',
 163 '1999-01-12'),
 164 ('mcarrollr@symantec.c', 'mcarrollr', 'Sbl1M0KKEa', 'Martin', 'Carroll',
 165 '1992-05-08'),
 166 ('mhunt12@gravatar.com', 'mhunt12', 'VmF4ANfv0U', 'Martin', 'Hunt',
 167 '1999-09-06'),
 168 ('rhicks11@example.com', 'rhicks11', 'WQB5uKeX', 'Rose', 'Hicks',
 169 '1992-05-22'),
 170 ('rryanh@usa.gov', 'rryanh', 'Ngy1hujx', 'Robert', 'Ryan', '1997-09-28'),
 171 ('scookd@washington.ed', 'scookd', 'tyj8KDVyJM', 'Sarah', 'Cook',
 172 '2000-09-04'),
 173

```

174      ('shenry7@businessweek', 'shenry7', 'ObTC98fes', 'Sara', 'Henry',
175      '1997-09-14'),
176      ('smartinez14@mozilla.', 'smartinez14', 'KAgrI9jz', 'Sharon', 'Martinez',
177      '1992-02-08'),
178      ('staylor16@squidoo.co', 'staylor16', 'Kp68wBELR', 'Steven', 'Taylor',
179      '1999-09-26'),
180      ('swagner8@bloglines.c', 'swagner8', '5kIaVK0E7vQ', 'Samuel', 'Wagner',
181      '1991-08-10'),
182      ('tortizf@shinystat.co', 'tortizf', 'QLiOk1Cg3', 'Thomas', 'Ortiz', '1987-05-25'),
183      ('wreynolds0@weibo.com', 'wreynolds0', 'CzS21wrgY', 'Walter', 'Reynolds',
184      '1992-12-02'),
185      ('wrobertson6@g.co', 'wrobertson6', '70ys1RdzzEx', 'Wayne', 'Robertson',
186      '1999-02-13'),
187      ('wromeroq@wikia.com', 'wromeroq', 'bMSN5lLg3', 'Wayne', 'Romero',
188      '1994-05-23');
189
190
191  -- Popolamento tabella Votazione
192
193  INSERT INTO Votazione (MailUtente, IdGioco, Voto)
194  VALUES ('agilbertm@princeton.', 1, 92),
195          ('agilbertm@princeton.', 21, 84),
196          ('agilbertm@princeton.', 31, 77),
197          ('agilbertm@princeton.', 41, 79),
198          ('agilbertm@princeton.', 51, 90),
199          ('agilbertm@princeton.', 61, 89),
200          ('agilbertm@princeton.', 71, 88),
201          ('agilbertm@princeton.', 91, 94),
202          ('agilbertm@princeton.', 121, 70),
203          ('agilbertm@princeton.', 141, 87),
204          ('agilbertm@princeton.', 161, 24),
205          ('amorris1@earthlink.n', 1, 40),
206          ('amorris1@earthlink.n', 21, 81),
207          ('apattersonx@fda.gov', 31, 77),
208          ('apattersonx@fda.gov', 41, 85),
209          ('apattersonx@fda.gov', 51, 89),
210          ('bbarnes4@virginia.ed', 71, 86),
211          ('bbarnes4@virginia.ed', 91, 84),
212          ('cford2@bloglines.com', 1, 87),
213          ('cford2@bloglines.com', 11, 92),
214          ('cford2@bloglines.com', 21, 86),
215          ('cford2@bloglines.com', 31, 70),
216          ('cford2@bloglines.com', 41, 90),
217          ('cford2@bloglines.com', 51, 95),
218          ('cford2@bloglines.com', 61, 98),
219          ('cford2@bloglines.com', 71, 99),
220          ('cford2@bloglines.com', 81, 91),
221          ('cford2@bloglines.com', 91, 96),
222          ('cford2@bloglines.com', 101, 86),
223          ('cford2@bloglines.com', 111, 82),

```

```

224      ('cford2@bloglines.com', 121, 69),
225      ('cford2@bloglines.com', 131, 89),
226      ('cford2@bloglines.com', 141, 91),
227      ('cford2@bloglines.com', 151, 65),
228      ('cford2@bloglines.com', 161, 35),
229      ('cmccoyo@indiatimes.c', 1, 93),
230      ('fdavisn@nbcnews.com', 61, 87),
231      ('jcampbell10@house.go', 91, 89),
232      ('jcampbell10@house.go', 101, 90),
233      ('jgibsonc@tuttocitta', 151, 70),
234      ('jgibsonc@tuttocitta', 161, 66),
235      ('jperezj@skype.com', 1, 95),
236      ('jperezj@skype.com', 101, 94),
237      ('jperezj@skype.com', 151, 98),
238      ('jperezj@skype.com', 161, 44),
239      ('jreynoldsl@cyberchim', 21, 92),
240      ('jreynoldsl@cyberchim', 41, 78),
241      ('lfosterw@weebly.com', 1, 92),
242      ('loliver13@webmd.com', 1, 93),
243      ('loliver13@webmd.com', 41, 80),
244      ('loliver13@webmd.com', 71, 95),
245      ('loliver13@webmd.com', 81, 90),
246      ('lromero15@ox.ac.uk', 1, 90),
247      ('lromero15@ox.ac.uk', 31, 75),
248      ('lromero15@ox.ac.uk', 61, 79),
249      ('lromero15@ox.ac.uk', 91, 96),
250      ('mcarrollr@symantec.c', 111, 86),
251      ('mhunt12@gravatar.com', 121, 88),
252      ('mhunt12@gravatar.com', 131, 79),
253      ('mhunt12@gravatar.com', 141, 90),
254      ('rryanah@usa.gov', 1, 89),
255      ('rryanah@usa.gov', 11, 86),
256      ('rryanah@usa.gov', 21, 80),
257      ('rryanah@usa.gov', 61, 86),
258      ('rryanah@usa.gov', 91, 93),
259      ('rryanah@usa.gov', 131, 85),
260      ('rryanah@usa.gov', 161, 62),
261      ('swagner8@bloglines.c', 11, 90),
262      ('swagner8@bloglines.c', 31, 72),
263      ('swagner8@bloglines.c', 91, 85),
264      ('wrobertson6@g.co', 1, 87),
265      ('wrobertson6@g.co', 11, 89),
266      ('wromeroq@wikia.com', 61, 87);
267
268
269  -- Popolamento tabella Console
270
271  INSERT INTO Console (Nome, Tipo, AnnoNascita, NomePublisher)
272  VALUES ('Atari 2600','Home','1977-09-11','Atari'),
273          ('GameCube','Home','2001-09-14','Nintendo'),

```

```

274      ('Xbox360','Home','2005-11-22','Microsoft'),
275      ('XboxOne','Home','2013-11-22','Microsoft'),
276      ('PS3','Home','2006-11-11','Sony'),
277      ('PS4','Home','2013-11-15','Sony'),
278      ('PSP','Portable','2004-12-12','Sony'),
279      ('3DS','Portable','2011-02-23','Nintendo'),
280      ('Wii U','Home','2012-11-18','Nintendo');
281
282
283  -- Popolamento tabella PiattaformaDigitale
284
285  INSERT INTO PiattaformaDigitale (Nome)
286  VALUES ('Steam'),
287          ('Origin'),
288          ('BattleNet');
289
290
291  -- Popolamento tabella FunzionaCon
292
293  INSERT INTO FunzionaCon (NomeConsole, CodGioco, AnnoPubblicazione)
294  VALUES ('3DS', 51, '2014-12-11'),
295          ('3DS', 91, '2015-08-17'),
296          ('3DS', 101, '2015-09-04'),
297          ('Atari 2600', 141, '1980-04-12'),
298          ('Atari 2600', 151, '1979-04-12'),
299          ('Atari 2600', 161, '1980-04-12'),
300          ('GameCube', 91, '2005-02-24'),
301          ('GameCube', 141, '2005-04-12'),
302          ('PS3', 1, '2008-02-08'),
303          ('PS3', 11, '2008-01-03'),
304          ('PS3', 31, '2012-08-14'),
305          ('PS3', 51, '2014-12-11'),
306          ('PS3', 71, '2013-06-14'),
307          ('PS3', 131, '2005-03-11'),
308          ('PS3', 141, '2007-01-12'),
309          ('PS4', 31, '2015-10-30'),
310          ('PS4', 51, '2014-12-11'),
311          ('PS4', 61, '2016-05-17'),
312          ('PS4', 71, '2016-06-01'),
313          ('PS4', 81, '2016-03-24'),
314          ('PS4', 111, '2016-04-12'),
315          ('PS4', 141, '2016-04-12'),
316          ('PSP', 11, '2008-01-03'),
317          ('PSP', 31, '2014-10-30'),
318          ('PSP', 51, '2014-12-11'),
319          ('PSP', 121, '2013-04-12'),
320          ('Wii U', 31, '2013-08-14'),
321          ('Wii U', 51, '2014-12-11'),
322          ('Wii U', 91, '2015-12-02'),
323          ('Wii U', 101, '2015-09-04'),

```

```

324      ('Wii U', 141, '2016-04-12'),
325      ('Xbox360', 1, '2007-05-01'),
326      ('Xbox360', 11, '2008-01-03'),
327      ('Xbox360', 31, '2012-08-14'),
328      ('Xbox360', 51, '2014-12-11'),
329      ('Xbox360', 141, '2007-01-12'),
330      ('XboxOne', 31, '2015-10-30'),
331      ('XboxOne', 51, '2014-12-11'),
332      ('XboxOne', 81, '2016-03-24'),
333      ('XboxOne', 111, '2016-04-12'),
334      ('XboxOne', 121, '2016-06-12'),
335      ('Xboxone', 141, '2016-04-12');
336
337
338  -- Popolamento tabella FunzionaPiattDig
339
340  INSERT INTO FunzionaPiattDig (NomePiattDig, CodGioco, AnnoPubblicazione)
341  VALUES ('BattleNet', 81, '2016-03-24'),
342          ('BattleNet', 141, '2014-04-12'),
343          ('Orgin', 141, '2014-04-12'),
344          ('Origin', 1, '2008-05-28'),
345          ('Origin', 11, '2009-02-05'),
346          ('Steam', 11, '2009-02-05'),
347          ('Steam', 21, '2009-02-18'),
348          ('Steam', 31, '2012-08-14'),
349          ('Steam', 41, '2013-06-25'),
350          ('Steam', 51, '2014-12-11'),
351          ('Steam', 111, '2016-04-12'),
352          ('Steam', 141, '2014-04-12');

```

6. Trigger, funzioni, procedure e query

6.1 Trigger

6.1.1 Aggiorna voto utente

```
1  DROP TRIGGER IF EXISTS update_voto_medio;
2
3  DELIMITER |
4
5  CREATE TRIGGER update_voto_medio
6  AFTER INSERT ON Votazione
7  FOR EACH ROW
8  BEGIN
9      UPDATE Videogioco
10     SET  VotoMedioUtenti = ( SELECT AVG(Voto)
11                             FROM Votazione
12                             WHERE Votazione.IdGioco = NEW.IdGioco )
13     WHERE Id = NEW.IdGioco;
14 END |
15
16 DELIMITER ;
```

Il trigger è necessario per aggiornare la votazione media degli utenti ogni qual volta venga inserito un nuovo voto. Serve a mantenere una consistenza tra le votazioni e il voto medio.

```
1  DROP TRIGGER IF EXISTS restore_voto_medio;
2
3  DELIMITER |
4
5  CREATE TRIGGER restore_voto_medio
6  AFTER DELETE ON votazione
7  FOR EACH ROW
8  BEGIN
9      UPDATE Videogioco
10     SET  VotoMedioUtenti = ( SELECT AVG ( Voto )
11                             FROM Votazione
12                             WHERE Votazione.IdGioco = OLD.IdGioco )
13     WHERE Id = OLD.IdGioco;
14 END |
15
```


16 DELIMITER ;

A differenza del primo, questo trigger aggiorna il voto medio quando un voto viene eliminato, è necessario in 2 casi:

- se viene eliminato un utente, e di conseguenza tutti i suoi voti
- se un utente decide di eliminare il suo voto

6.1.2 Controllo utente

```
1  DROP TRIGGER IF EXISTS check_Utenti;
2
3  DELIMITER |
4
5  CREATE TRIGGER check_Utenti
6  BEFORE INSERT ON utente
7  FOR EACH ROW
8  BEGIN
9      DECLARE Num INT;
10     DECLARE Find INT;
11
12     SELECT COUNT(*) INTO Num
13     FROM utente;
14
15     SELECT COUNT(*) INTO Find
16     FROM utente
17     WHERE NickName = NEW.NickName;
18
19     IF(Num >= 500 OR Find = 1)
20     THEN INSERT INTO utente SELECT * FROM utente LIMIT 1;
21     END IF;
22
23     END|
24
25     DELIMITER ;
```

Il seguente trigger effettua un controllo nel momento in cui viene inserito un nuovo utente. Ipotizzando che ci siano problemi a salvare più di 500 utenti è stato inserito un errore se la capienza è stata raggiunta. Oltretutto se il nickname che si cerca di inserire è già stato utilizzato si genera un errore.

6.2 Funzioni

6.2.1 Voto Medio Utente

```
1  DROP FUNCTION IF EXISTS Voto_Medio_utente;
2
3  DELIMITER |
4
5  CREATE FUNCTION Voto_Medio_utente( NickName VARCHAR(20)) RETURNS INT
6
7  BEGIN
8
9      DECLARE votoM INT;
10
11      SELECT  AVG(Voto) INTO votoM
12      FROM    Utente U JOIN Votazione V ON U.Email = V.MailUtente
13      WHERE   U.NickName = NickName;
14
15      IF votoM IS NULL
16      THEN SET votoM = 0;
17      END IF;
18
19      RETURN votoM;
20
21  END |
22
23  DELIMITER ;
```

La seguente funzione prende come input il nickname di un utente e restituisce il suo voto medio. Può essere utile per capire se un utente è severo o meno nel valutare i videogiochi.

6.2.2 N° Giochi per console

```
1  DROP FUNCTION IF EXISTS AmmontareGiochi;
2
3  DELIMITER |
4
5  CREATE FUNCTION AmmontareGiochi( CNome VARCHAR(16)) RETURNS INT
6
7  BEGIN
8      DECLARE Totale INT;
9      SELECT  COUNT( CodGioco ) INTO Totale
10     FROM    funzionacon f
11     WHERE   f.NomeConsole = CNome;
12
13     RETURN Totale;
14
15     END |
16
17     DELIMITER ;
```

Funzione che prende in input il nome di una console e restituisce il numero di videogiochi presenti in quella console. Utile per creare query più semplici.

6.2.3 Livello Utente

```
1  DROP FUNCTION IF EXISTS UserLevel;
2
3  DELIMITER |
4
5  CREATE FUNCTION UserLevel (Mail VARCHAR(20)) RETURNS varchar(255)
6  BEGIN
7
8  DECLARE lvl varchar(255);
9  DECLARE votoU INT;
10
11
12  SELECT COUNT(*) INTO votoU
13  FROM  votazione
14  WHERE votazione.MailUtente=Mail;
15
16
17  IF (votoU > 20) THEN
18      SET lvl = 'Elite';
19  ELSEIF (votoU <= 20 AND votoU >= 10) THEN
20      SET lvl = 'Veteran';
21  ELSEIF(votoU <10 AND votoU >= 5) THEN
22      SET lvl = 'Regular User';
23  ELSEIF(votoU<5) THEN
24      SET lvl = 'Novice User';
25  END IF;
26
27  RETURN (lvl);
28
29  END|
30
31  DELIMITER ;
```

La seguente funzione prende in input la mail di un utente e restituisce il “livello” dell’utente stesso. È stata aggiunta in quanto sembra ragionevole pensare che in una successiva implementazione venga aggiunto un sistema a livelli basato sul numero di votazioni effettuate dagli utenti.

6.3 Procedure

6.3.1 Modifica voto critica

```
1  DROP PROCEDURE IF EXISTS ModificaVotoCritica;
2
3  DELIMITER |
4
5  CREATE PROCEDURE ModificaVotoCritica(idGioco INT, NuovoVoto INT)
6  BEGIN
7      DECLARE VecchioVoto INT;
8      SELECT VotoCritica INTO VecchioVoto
9      FROM videogioco
10     WHERE Id = idGioco;
11
12     IF NuovoVoto > VecchioVoto*0.9
13     THEN UPDATE videogioco
14          SET VotoCritica = NuovoVoto
15          WHERE Id = idGioco;
16     END IF;
17 END |
18
19 DELIMITER ;
```

La seguente procedura serve a modificare il voto della critica, chiaramente può servire se il voto dopo una più attenta analisi del gioco vuole essere modificato. Se il voto è troppo discordante dal precedente non viene modificato alcunchè.

6.4 Query

6.4.1 Utenti maggiorenni con almeno 2 voti

```
1  SELECT Cognome, Nome, DataNascita, TIMESTAMPDIFF ( YEAR, DataNascita,
2                                     CURDATE() ) AS Eta
3  FROM  utente
4  WHERE ( TIMESTAMPDIFF ( YEAR, DataNascita, CURDATE())) > 17 AND Email IN
5        ( SELECT v.MailUtente
6          FROM  votazione v JOIN votazione v2 ON v.MailUtente = v2.MailUtente
7          WHERE v.MailUtente = v2.MailUtente AND v.IdGioco <> v2.IdGioco )
8  ORDER BY Eta;
```

La seguente query mostra gli utenti maggiorenni (in base alla data corrente) che abbiano votato almeno 2 videogiochi e li ordina per età. Vengono utilizzate la funzione `TIMESTAMPDIFF` e `CURDATE`. La prima calcola la differenza tra due date, la seconda restituisce la data odierna.

OUTPUT

Cognome	Nome	DataNascita	Eta
Ryan	Robert	1997-09-28	18
Reynolds	Janet	1998-02-12	18
Barnes	Barbara	1996-10-17	19
Gilbert	Angela	1996-01-04	20
Patterson	Alice	1995-07-28	20
Morris	Andrew	1993-07-16	22
Wagner	Samuel	1991-08-10	24
Oliver	Louise	1988-10-11	27
Campbell	Janet	1986-12-25	29
Ford	Carlos	1985-12-11	30

6.4.2 Giochi senza voto utenti

```
1  SELECT  v.Titolo, v.VotoCritica
2  FROM    videogioco v
3  WHERE   v.Id NOT IN
4          ( SELECT IdGioco
5            FROM votazione );
```

La seguente query restituisce i titoli dei videogiochi, il corrispondente voto della critica, senza nessun voto da parte degli utenti.

OUTPUT

Titolo	VotoCritica
Destiny	85

6.4.3 Videogiochi recenti console

```
1  CREATE VIEW ConsoleLastGen AS
2  SELECT Nome
3  FROM console
4  WHERE Nome IN ( 'Wii U', 'PS4', 'XboxOne' );
5
6  SELECT DISTINCT VG.Titolo, FC.NomeConsole, FC.AnnoPubblicazione
7  FROM      Videogioco VG JOIN funzionacon FC ON VG.Id=FC.CodGioco
8  WHERE     VG.Id NOT IN
9            ( SELECT FP.CodGioco
10              FROM funzionapiattdig FP )
11          AND FC.NomeConsole IN ( SELECT *
12                                FROM ConsoleLastGen)
13          AND YEAR ( FC.AnnoPubblicazione ) = YEAR ( CURDATE() );
```

La query restituisce i videogiochi usciti nell'ultimo anno nelle console di ultima generazione, in questa query si possono aggiungere altre condizioni per trovare i videogiochi di un solo genere o di una sola console.

Per selezionare le console di ultima generazione è stata creata una view che restituisce le console di ultima generazione.

OUTPUT ConsoleLastGen

Nome
PS4
Wii U
XboxOne

OUTPUT

Titolo	NomeConsole	AnnoPubblicazione
Bloodborne	PS4	2016-05-17
The last of Us	PS4	2016-06-01
Naruto UNS3	XboxOne	2016-06-12

6.4.4 Developer/publisher nati prima degli anni 90

```
1  SELECT  Nome, AnnoFondazione
2  FROM    publisher
3  WHERE   Nome IN
4           ( SELECT Nome
5             FROM   developer )
6          AND YEAR ( AnnoFondazione ) < 1990;
```

Query che restituisce i Developer, nati prima degli anni 90, che sono anche Publisher.

OUTPUT

Nome	AnnoFondazione
Atari	1972-06-27
Bandai Namco	1955-06-01
Nintendo	1889-09-23
Ubisoft	1986-03-01

6.4.5 Giochi Action con votazione alta

```
1  CREATE VIEW Votanti AS
2  SELECT      v.Titolo, COUNT ( v1.MailUtente ) AS N_Vot
3  FROM        videogioco v JOIN votazione v1 ON ( v.Id = v1.IdGioco )
4  GROUP BY    v1.IdGioco;
5
6
7  SELECT      v.Titolo, v.VotoMedioUtenti
8  FROM        Videogioco v JOIN Developer d ON v.NomeDeveloper = d.Nome
9             JOIN Votanti v1 ON v.Titolo = v1.Titolo
10 WHERE       v.Genere = 'Action'
11             AND v.Multiplayer = 1
12             AND v.VotoMedioUtenti > 85
13             AND v1.N_Vot > 5;
```

Restituisce i videogiochi di genere Action che hanno il multiplayer, votati da almeno 5 utenti e con una votazione media superiore a 85. È stato necessario creare un'altra view che restituisse il numero di voti per ogni videogioco.

OUTPUT VOTANTI

Titolo	N_Vot
Mass Effect	10
Burnout Paradise	4
Warhammer 40000	5
Darksiders 2	5
Company of Heroes 2	5
Walking Dead	3
Bloodborne	6
The last of Us	4
Overwatch	2
Zelda	7
Super Smash Bros	3
Dark Souls 3	2
Naruto UNS3	3
Tekken 5	3

Rayman 2	3
Pacman	3
Skate or Die	5

OUTPUT QUERY

Titolo	VotoMedioUtenti
Bloodborne	88

6.4.6 Utenti non votanti o votanti solo giochi PS4

```
1  SELECT  Email, Nickname
2  FROM    utente
3  WHERE   Email NOT IN
4          ( SELECT DISTINCT MailUtente
5            FROM             votazione )
6  UNION
7  SELECT DISTINCT  u.Email, u.Nickname
8  FROM             utente u JOIN votazione v ON u.Email = v.MailUtente
9                  JOIN videogioco v1 ON v.IdGioco = v1.Id
10                 JOIN funzionacon f ON v1.Id = f.CodGioco
11 WHERE           f.NomeConsole = 'PS4'
12 ORDER BY       Nickname;
```

La seguente query restituisce le mail e i nickname degli utenti che non hanno votato videogiochi o che hanno votato videogiochi solo della PS4. Per realizzarla si è usato un operatore insiemistico, UNION. La prima parte restituisce gli utenti che non hanno mai votato, la seconda gli utenti che hanno votato solo giochi della PS4.

OUTPUT

Email	Nickname
agilbertm@princeton.	agilbertm
apattersonx@fda.gov	apattersonx
bbarnes4@virginia.ed	bbarnes4
bmarshallu@uol.com.b	bmarshallu
bmorrisonz@marriott.	bmorrisonz
cford2@bloglines.com	cford2
cstewart5@constantco	cstewart5
dbrooksy@youtube.com	dbrooksy
dramirez3@arizona.ed	dramirez3
drileye@topsy.com	drileye
dwoodsv@joomla.org	dwoodsv
ehuntp@nyu.edu	ehuntp
fdavisn@abcnews.com	fdavisn
flawson9@myspace.com	flawson9

fsimsb@bbc.co.uk	fsimsb
gweavert@istockphoto	gweavert
iparkera@indiatimes.	iparkera
kgibsonc@tuttocitta.	kgibsonc
jhenryi@seesaa.net	jhenryi
kweaverg@dot.gov	kweaverg
kweavers@pagesperso-	kweavers
lcunninghamk@tinypic	lcunninghamk
loliver13@webmd.com	loliver13
lromero15@ox.ac.uk	lromero15
mcarrollr@symantec.c	mcarrollr

6.4.7 N° Giochi Console

```

1  SELECT    NomeConsole, COUNT( CodGioco ) AS N_Giochi
2  FROM      funzionacon
3  GROUP BY  NomeConsole;
```

La seguente query restituisce i nomi delle console e il numero di videogiochi presenti in esse.

OUTPUT

NomeConsole	N_Giochi
3DS	3
Atari 2600	3
GameCube	2
PS3	7
PS4	7
PSP	4
Wii U	5
Xbox360	5
XboxOne	6