PROGETTO BASI DI DATI 2016/2017

“APP STORE”

Gianmarco Pettenuzzo
Matricola 1097856

Francesco Battistella
Matricola 1102842

1. Abstract

Con la nascita degli smartphone e il loro continuo sviluppo, si è reso necessario associare a questi diverse funzionalità.
L'App Store è un servizio che permette ai possessori di smartphone di acquistare e scaricare applicazioni al fine di migliorare e diversificare la propria esperienza con i suddetti dispositivi.
Tramite l'App Store è possibile scaricare:

- Funzionalità;
- Giochi;
- Film;
- Libri;
- Musica;

Lo scopo di questo progetto è la creazione di un database che organizza le raccolte divise per categorie e ordinate secondo un sistema di recensioni.
Gli utenti, registrati attraverso la propria e-mail, possono scaricare e/o comprare contenuti dall'App Store e assegnare un voto da 0 a 5 su ciò che hanno scaricato.

2. Analisi dei Requisiti

Si vuole progettare un database che contenga e organizzi i servizi e i contenuti offerti da un **App Store**.
L'entità **App** sarà la principale. Essa si può suddividere in base al Genere in:

- **Giochi,** suddivisi in Categorie (per esempio Puzzle, Arcade ecc.) e di cui vogliamo il nome dello Sviluppatore;
- **Funzionalità,** suddivisi in Categorie (per esempio, Social, Istruzione ecc.) e di cui vogliamo il nome dello Sviluppatore;
- **Film,** suddivisi in Categorie (per esempio Thriller, Animazione ecc.) e di cui vogliamo tenere traccia del Cast e della Produzione;
- **Libri,** suddivisi in Categorie (per esempio Fantasy, Avventura ecc.) e di cui vogliamo conoscere lo Scrittore e la Casa produttrice;
- **Musica,** suddivisa in Categoria (per esempio Rock, Pop ecc.) e di cui vogliamo conoscere l'Artista.

Per ogni App ci interessa poi sapere il Nome, una Descrizione (comprendente di immagini, video, una descrizione testuale, lo spazio occupato e la durata di film e musica), la Data di rilascio, e il Prezzo poiché ci saranno contenuti a **Pagamento** e contenuti **Gratuiti**.

Le App a pagamento genereranno una **Fattura** emessa dall'**Azienda fondatrice** dell'App Store di cui ci basta sapere il Nome. Di essa ci interessa:

- il Numero identificativo;
- la Data di emissione;
- il Mittente, che sarà la stessa Azienda fondatrice.

Ciascuna fattura necessita di queste informazioni per essere identificata univocamente ed evitare conflitti del tipo:

- Assenza del fornitore;
- Assenza della data fattura;
- Assenza del numero fattura.

L'**Utente** potrà pagare la Fattura relativa all'acquisto di servizi, giochi o contenuti attraverso la **Carta di credito,** di cui ci interessa:

- il Numero della carta unico e non duplicabile;
- la Data di scadenza;
- l'Intestatario, che potrà anche essere diverso dallo stesso Utente;
- il Codice di sicurezza.

Per poter scaricare le App il cliente dovrà creare un **Account** dove saranno salvate le seguenti informazioni:

- E-Mail, che sarà univoca;
- Nome e cognome dell'utente;
- Data di nascita dell'utente;
- Password con la quale accedere al proprio Account e confermare gli acquisti effettuati.

Ogni Utente, ovviamente, possiede uno **Smartphone**, con il quale accedere all'App Store, di cui ci interessa solo il Modello.
Tramite l'Account, inoltre, l'Utente potrà:

- rilasciare **Valutazioni** alle App scaricate tramite un Voto (da 0 a 5) e una Recensione scritta facoltativa;
- gestire **Lista Download** che contiene le App scaricate in una precisa data, che chiameremo DataDownload.

3. Progettazione Concettuale

3.1 Entità, Attributi e Gerarchie

- **App:** Si tratta dell'entità principale del database, in quanto la gestione dell'App Store è incentrata su di essa.
  Attributi:
  - CodiceApp: int (identificatore);
  - NomeApp: string;
  - Descrizione: string;
  - DataRilascio: date;
  - Genere: string (attributo multi valore).

  Gerarchia – Entità figlie

  In base al genere:

  - **Giochi:**
    - Categoria: string;
    - Sviluppatore: string.
  - **Funzionalità:**
    - Categoria: string;
    - Sviluppatore: string.
  - **Film:**
    - Categoria: string;
    - Cast: string;
    - Regista: string.

- **Musica:**
  - Categoria: string;
  - Artista: string.
- **Libri:**
  - Categoria: string;
  - Autore: string;
  - CasaEditrice: string.

In base al prezzo:

- **Gratuiti;**
- **A pagamento.**

Rappresentano due generalizzazioni totali di un'App.

- **Azienda Fondatrice**: contiene le generalità dell'Azienda creatrice dell'App Store. Essa, è titolare di un unico App Store.
  Attributi:
  - CodiceAzienda: int (identificatore);
  - NomeAppStore: string
  - Nome: string.

- **Fattura:** modella le Fatture d'acquisto emesse in caso di App a pagamento, emesse dall'Azienda fondatrice che vengono registrate all'interno del database.
  Attributi:
  - NumeroFattura: int (identificatore);
  - NomeFornitore: string (identificatore esterno);
  - DataFattura: date;
  - CartaPagamento (identificatore esterno).

- **Carta di credito:** contiene le Carte di credito con le quali sono stati effettuati pagamenti delle Fatture.
  Attributi:
  - NumeroCarta: int (identificatore);
  - DataScadenzaCarta: date;
  - Intestatario: string (identificatore esterno);
  - CodiceSicurezza: int.

- **Utente:** contiene le generalità della persona fisica che acquista/scarica le App.
  Attributi:
  - CodiceFiscale (Identificatore);
  - NomeCognome (attributo composto):
    - Nome (string);
    - Cognome (string);
  - DataNascita: date.

- **Smartphone:** contiene le generalità del dispositivo con cui l'Utente effettua download nell'App Store.
  Attributi:
  - Modello: string (identificatore);
  - Utilizzatore: string (identificatore esterno).

- **Account:** modella le generalità dell'account attraverso cui l'Utente scarica le App e con cui è registrato all'App Store.
  Attributi:
  - E-mail: string (identificatore esterno);
  - Utente: string (identificatore esterno);
  - Password: string.

- **Valutazioni**: contiene le valutazioni rilasciate dall'utente tramite l'account.
  Attributi:
  - Voto: int (identificatore);
  - Recensione: string (identificatore);
  - Account: string (identificatore esterno);
  - CodiceApp: int (identificatore).

- **Lista Download:** contiene le App scaricate dall'Account.
  Attributi:
  - Account: string (identificatore / identificatore esterno);
  - AppScaricata: string (identificatore / identificatore esterno);
  - DataDownload: date (identificatore).

3.2 Relazioni e Cardinalità

- **Azienda Fondatrice – Fattura: Emette**
  - (0, N): 0 o più Fatture possono essere emesse da un'Azienda;
  - (1, 1): Ogni Fattura ha un solo mittente.

- **Fattura – Carta di Credito: Pagata**
  - (1, 1): Ogni Fattura viene pagata da una sola Carta di credito;
  - (0, N): Una Carta di credito può pagare 0 o più Fatture.

- **App a Pagamento – Fattura: Genera**
  - (1, 1): Un acquisto genera una Fattura;
  - (1, 1): Una Fattura è generata da un'unica App.

- **App – Account: Scaricate tramite**
  - (0, N): Un'App può essere scaricata da 0 o più Account;
  - (0, N): Un Account può scaricare 0 o più App.

- **Utente – Carta di Credito: Possiede**
  - (1, 1): Un Utente possiede un'unica Carta di credito;
  - (1, 1): Ogni Carta di credito ha un solo Intestatario.

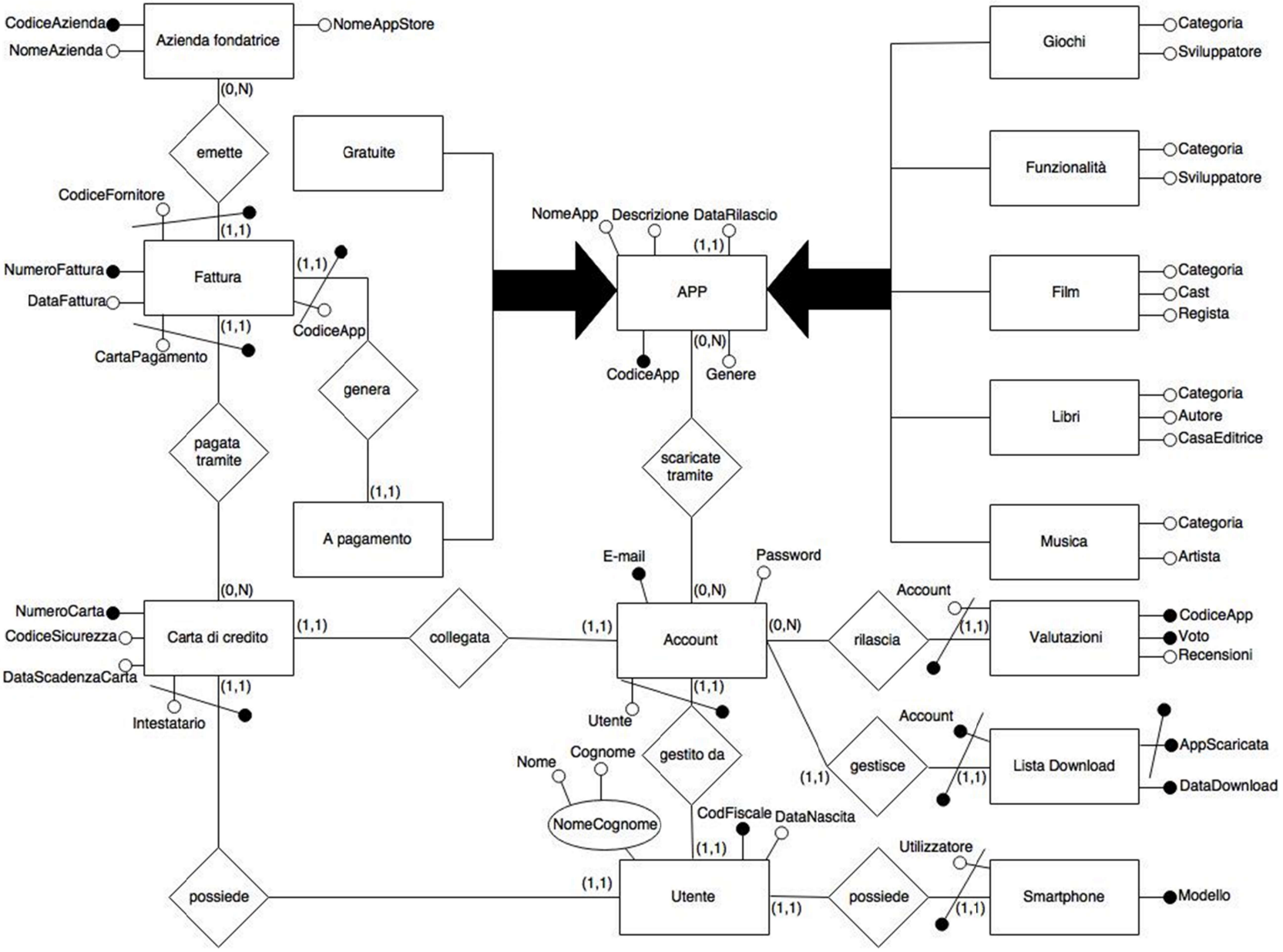
- **Utente – Smartphone: Possiede**
  - (1, 1): Ogni Utente possiede uno Smartphone;
  - (1, 1): Ogni Smartphone ha un unico proprietario.

- **Account – Utente: Gestito da**
  - (1, 1): Un Utente può gestire un solo Account;
  - (1, 1): Ogni Account è gestito da un unico Utente.

- **Carta di Credito – Account: Collegata**
  - (1, 1): Una Carta di credito è collegata ad un solo Account;
  - (1, 1): Ad ogni Account è collegata un'unica carta di credito.

- **Account – Valutazioni: Rilascia**
  - (0, N): Ogni Account può rilasciare da 0 a più Valutazioni;
  - (1, 1): Ogni Valutazione è personale e viene rilasciata da un unico Account.

- **Account – Lista Download: Gestisce**
  - (1, 1): Un Account ha solo una Lista Download;
  - (1, 1): Una Lista Download appartiene ad un solo Account.


CodiceAzienda
NomeAzienda
Azienda fondatrice
NomeAppStore
(0,N)
emette
Gratuite
CodiceFornitore
(1,1)
NumeroFattura
DataFattura
Fattura
(1,1)
CodiceApp
(1,1)
CartaPagamento
genera
pagata tramite
(1,1)
A pagamento
NomeApp
Descrizione
DataRilascio
(1,1)
APP
CodiceApp
(0,N)
Genere
scaricate tramite
Giochi
Categoria
Sviluppatore
Funzionalità
Categoria
Sviluppatore
Film
Categoria
Cast
Regista
Libri
Categoria
Autore
CasaEditrice
Musica
Categoria
Artista
E-mail
Password
(0,N)
NumeroCarta
CodiceSicurezza
DataScadenzaCarta
Carta di credito
(1,1)
collegata
(1,1)
Account
(0,N)
rilascia
Account
(1,1)
Valutazioni
CodiceApp
Voto
Recensioni
(1,1)
Intestatario
(1,1)
Utente
gestito da
Nome
Cognome
NomeCognome
Account
gestisce
(1,1)
(1,1)
Lista Download
AppScaricata
DataDownload
CodFiscale
DataNascita
(1,1)
possiede
(1,1)
Utente
(1,1)
possiede
Utilizzatore
(1,1)
Smartphone
Modello

4. Progettazione Logica

4.1 Eliminazione delle generalizzazioni

Si è deciso di mantenere l'entità App accorpando le entità figlie nell'entità genitore con l'introduzione di un attributo composto **Genere** e un altro attributo **Prezzo,** per preservare la distinzione tra le occorrenze di tale entità che le generalizzazioni rappresentavano.
All'occorrenza le entità figlie possono assumere il valore NULL quando il costo è 0 e quindi l'App è gratuita.

4.2 Eliminazione Attributo composto

L'attributo NomeCognome dell'entità Utente viene eliminato e ciascuno dei suoi attributi viene inserito nell'entità principale.

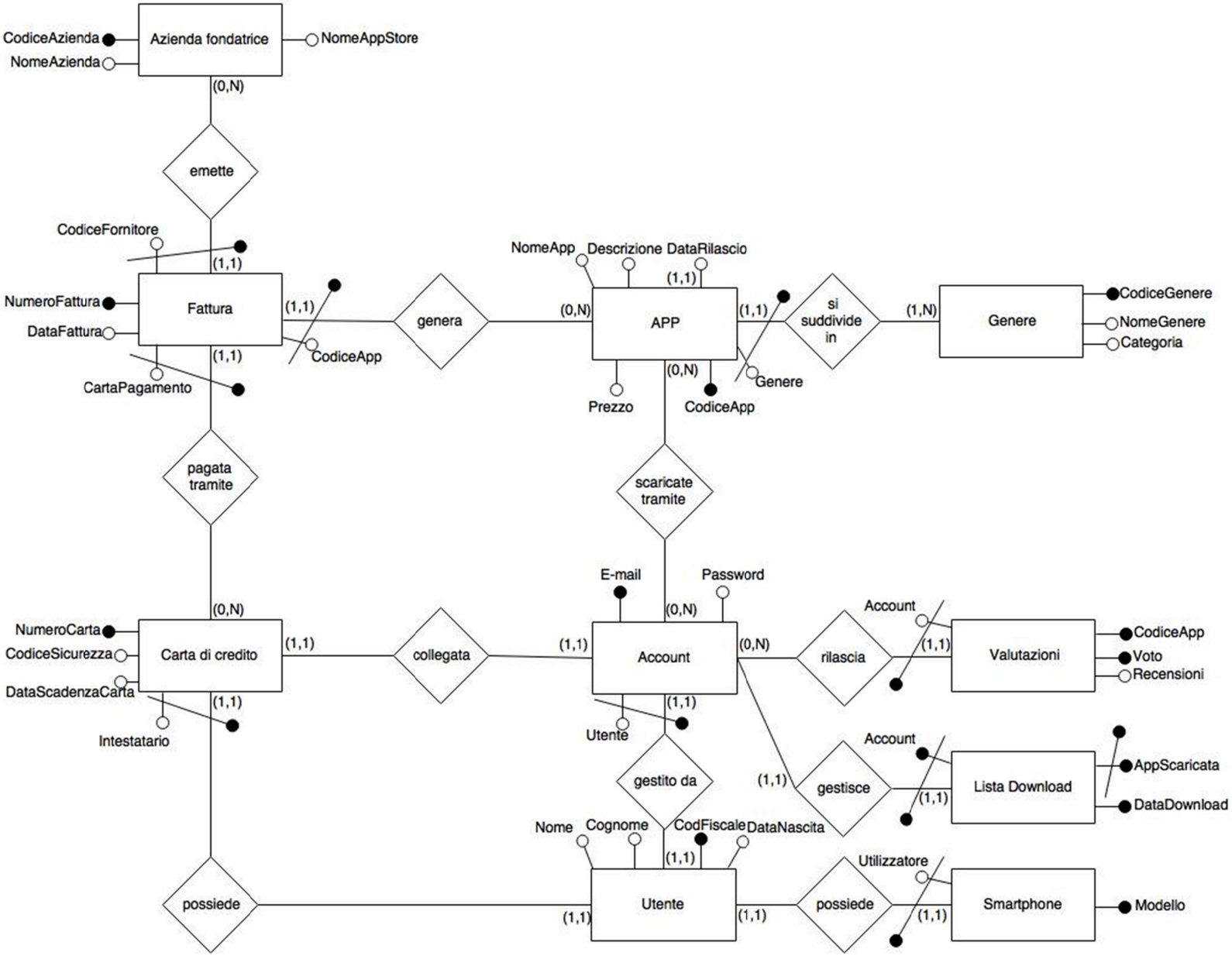
4.3 Eliminazione Attributi Multi valore

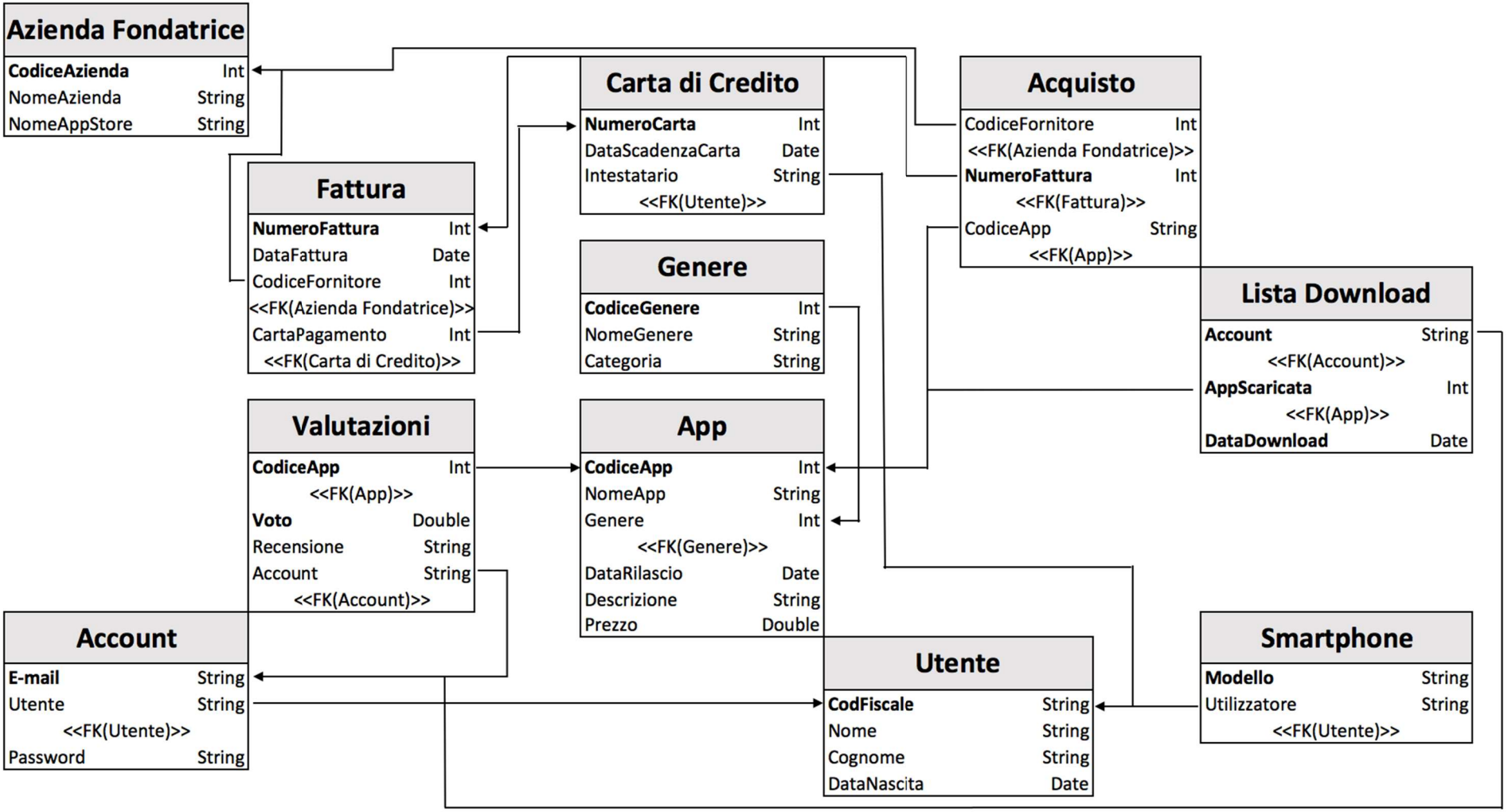
Viene partizionato l'entità App avente l'attributo multi valore Genere in due entità: un'entità App (con lo stesso nome e gli stessi attributi eccetto l'attributo multi valore), e un'entità Genere con attributi:

- CodiceGenere;
- NomeGenere;
- Categoria.

4.4 Analisi delle Ridondanze

Si crea una sorta di ridondanza tra Utente – Account dovuta dall'attributo CodiceFiscale – Utente.
Essa è risolvibile unendo l'entità Utente all'entità Account.
Il nostro progetto prevede una netta distinzione tra Utente (la persona fisica, possessore di una Carta di credito e di uno Smartphone, titolare di Account) e Account ("modulo" di registrazione per mezzo del quale l'utente può accedere all'App Store e scaricare quello che offre).
Infatti, le entità Smartphone e CartadiCredito posseggono una chiave esterna di "CodFiscale" di Utente, mentre ListaDownload e Valutazione si legano a "Email" di Account.
Pertanto, preferiamo continuare a distinguere le suddette entità anche all'interno del database.


Azienda fondatrice
CodiceAzienda
NomeAzienda
NomeAppStore
(0,N)
emette
CodiceFornitore
(1,1)
Fattura
NumeroFattura
DataFattura
CartaPagamento
(1,1)
(1,1)
CodiceApp
genera
NomeApp
Descrizione
DataRilascio
(1,1)
(0,N)
APP
Prezzo
(0,N)
CodiceApp
(1,1)
Genere
si suddivide in
(1,N)
Genere
CodiceGenere
NomeGenere
Categoria
pagata tramite
scaricate tramite
(0,N)
NumeroCarta
CodiceSicurezza
DataScadenzaCarta
Intestatario
Carta di credito
(1,1)
(1,1)
collegata
E-mail
Password
(0,N)
(1,1)
Account
(1,1)
Utente
(0,N)
rilascia
Account
(1,1)
Valutazioni
CodiceApp
Voto
Recensioni
gestito da
(1,1)
gestisce
Account
(1,1)
Lista Download
AppScaricata
DataDownload
Nome
Cognome
CodFiscale
DataNascita
(1,1)
Utente
(1,1)
possiede
possiede
Utilizzatore
(1,1)
Smartphone
Modello


Azienda Fondatrice
CodiceAzienda Int
NomeAzienda String
NomeAppStore String
Carta di Credito
NumeroCarta Int
DataScadenzaCarta Date
Intestatario String
<<FK(Utente)>>
Acquisto
CodiceFornitore Int
<<FK(Azienda Fondatrice)>>
NumeroFattura Int
<<FK(Fattura)>>
CodiceApp String
<<FK(App)>>
Fattura
NumeroFattura Int
DataFattura Date
CodiceFornitore Int
<<FK(Azienda Fondatrice)>>
CartaPagamento Int
<<FK(Carta di Credito)>>
Genere
CodiceGenere Int
NomeGenere String
Categoria String
Lista Download
Account String
<<FK(Account)>>
AppScaricata Int
<<FK(App)>>
DataDownload Date
Valutazioni
CodiceApp Int
<<FK(App)>>
Voto Double
Recensione String
Account String
<<FK(Account)>>
App
CodiceApp Int
NomeApp String
Genere Int
<<FK(Genere)>>
DataRilascio Date
Descrizione String
Prezzo Double
Account
E-mail String
Utente String
<<FK(Utente)>>
Password String
Utente
CodFiscale String
Nome String
Cognome String
DataNascita Date
Smartphone
Modello String
Utilizzatore String
<<FK(Utente)>>

4.5 Chiavi Esterne

- **CodiceApp:**
  È una chiave esterna tra le entità “Valutazioni” → “App”;
  È una chiave esterna tra le entità “Lista Download” → “App”;
  È una chiave esterna tra le entità “Acquisto” → “App”.
- **Genere:**
  È una chiave esterna tra le entità “App” → ”Genere”.
- **Utente:**
  È una chiave esterna tra le entità “Smartphone” → “Utente”;
  È una chiave esterna tra le entità “Account” → “Utente”;
  È una chiave esterna tra le entità “Carta di Credito” → “Utente”.
- **Codice Azienda:**
  È una chiave esterna tra le entità “Fattura” → “Azienda Fondatrice”;
  È una chiave esterna tra le entità “Acquisto” → “Azienda Fondatrice”.
- **NumeroFattura:**
  È una chiave esterna tra le entità “Acquisto” → “Fattura”.
- **Account:**
  È una chiave esterna tra le entità “Lista Download” → “Account”;
  È una chiave esterna tra le entità “Valutazioni” → “Account”.
- **Numero Carta:**
  È una chiave esterna tra le entità “Fattura” → “Carta di Credito”.

4.6 Schema Relazionale

- **Entità:**
  AZIENDA FONDATRICE: (**CodiceAzienda,** NomeAppStore, NomeAzienda);
  APP: (**CodiceApp,** NomeApp, Genere, DataRilascio, Descrizione, Prezzo);
  GENERE: (**CodiceGenere,** NomeGenere, Categoria);
  FATTURA: (**NumeroFattura,** DataFattura, CodiceFornitore, CartaPagamento);
  CARTA DI CREDITO: (**NumeroCarta,** DataScadenzaCarta, Intestatario, CodiceSicurezza);
  UTENTE: (**CodFiscale,** NomeCognome, DataNascita);
  SMARTPHONE: (**Modello**, Utilizzatore);
  ACCOUNT: (**E-mail,** Utente, Password);
  VALUTAZIONI: (**CodiceApp, Voto,** Recensione**,** Account);
  LISTA DOWNLOAD: (**Account**, **AppScaricata, DataDownload**).

- **Associazioni molti a molti**
  ACQUISTO: (**NumeroFattura**, CodiceFornitore, CodiceApp).

- **Associazioni uno a uno:**
  - **Associata**
    Carta di credito – Utente;
  - **Collegata**
    Carta di credito – Account;
  - **Possiede**
    Utente – Smartphone;
  - **Gestisce**
    Utente – Account,
    Account – Lista Download.

- **Associazioni uno a molti**
  - **Genera:** (0, N) App – Fattura (1, 1);
  - **Emette:** (0, N) Azienda Fondatrice – Fattura (1, 1);
  - **Pagata:** (1, 1) Fattura – Carta di Credito (0, N);
  - **Suddivide:** (1, 1) App – Genere (1, N);
  - **Scaricate:** (0, N) App – Utente (0, N);
  - **Rilascia:** (0, N) Account – Valutazioni (1, 1).

Complessivamente:
App (**CodiceApp,** NomeApp, DataRilascio, Genere, Descrizione, Prezzo);
AziendaFondatrice (**CodiceAzienda,** NomeAppStore, NomeAzienda);
Genere (**CodiceGenere**, NomeGenere, Categoria);
Fattura (**NumeroFattura**, CodiceFornitore, DataFattura, CartaPagamento);
CartadiCredito (**NumeroCarta**, Intestatario, DataScadenzaCarta, CodiceSicurezza);
Utente (**CodFiscale,** NomeCognome, DataNascita);
Account (**E-Mail**, Utente, Password);
Smartphone (**Utilizzatore**, Modello);
Valutazione (**Account**, **CodiceApp**, **Voto**, Recensione);
ListaDownload (**Account**, **AppScaricata**, **DataDownload**);
Acquisto (**NumeroFattura**, CodiceFornitore, CodiceApp).

5. Query, Funzioni e Trigger

5.1 Query

1) Vogliamo sapere il nome e il cognome degli utenti che hanno acquistato il libro “Semplice è bello”. Visto che, il suddetto libro, ha un costo, vogliamo sapere anche la data d’acquisto, il prezzo e ordinare i risultati per data.

```
SELECT NomeCognome, DataFattura as DataAcquisto, Prezzo
FROM ((app JOIN acquisto on app.CodiceApp = acquisto.CodiceApp)
JOIN fattura on acquisto.NumeroFattura = fattura.NumeroFattura
JOIN cartadicredito on fattura.CartaPagamento = cartadicredito.NumeroCarta
JOIN utente on cartadicredito.Intestatario = utente.CodFiscale)
WHERE app.NomeApp = 'Semplice è bello'
ORDER BY DataAcquisto ASC;
```

NomeCognome	DataAcquisto	Prezzo
Stefano Nordio	2017-05-22	13.9
Francesco Battistella	2017-07-17	13.9
Matteo Depascale	2017-12-05	13.9

2) Il NomeCognome degli utenti che hanno nella loro lista download l’applicazione “Eurosport”.

```
DROP VIEW if exists Eurosport;
CREATE VIEW Eurosport as
SELECT * from App
WHERE CodiceApp=25;

SELECT NomeCognome
FROM ((Utente JOIN Account on Account.Utente = Utente.CodFiscale)
JOIN ListaDownload on ListaDownload.Account = Account.EMail)
WHERE CodiceApp1 in (SELECT CodiceApp FROM Eurosport)
OR CodiceApp2 in (SELECT CodiceApp FROM Eurosport)
OR CodiceApp3 in (SELECT CodiceApp FROM Eurosport)
OR CodiceApp4 in (SELECT CodiceApp FROM Eurosport)
OR CodiceApp5 in (SELECT CodiceApp FROM Eurosport)
OR CodiceApp6 in (SELECT CodiceApp FROM Eurosport);
```

NomeCognome
Gianmarco Pettenuzzo
Francesco Battistella
Francesco Pecile

3) Selezionare tutti gli acquisti di “Giochi” fatti nel 2016 con il numero di fattura e il numero di carta di credito.

```
SELECT DataFattura, Fattura.NumeroFattura, CartaPagamento
FROM ((Genere JOIN App on App.Genere = Genere.CodGenere)
JOIN Acquisto on App.CodiceApp = Acquisto.CodiceApp
JOIN Fattura on Acquisto.NumeroFattura = Fattura.NumeroFattura)
WHERE Genere.NomeGenere='Giochi'
AND DataFattura < '2017-01-01'
AND DataFattura > '2015-31-12';
```

DataFattura	NumeroFattura	CartaPagamento
2016-01-20	9966	5698745213654526

4) Vogliamo un elenco con tutte le valutazioni rilasciate all’applicazione “Facebook” ordinate in ordine alfabetico dall’email dell’utente, il suo NomeCognome e il cellulare in possesso.
Ove possibile, vogliamo anche leggere la recensione.

```
SELECT EMail, NomeCognome, Modello, Voto, Recensione
FROM ((Utente JOIN Account on Account.Utente = Utente.CodFiscale)
JOIN Smartphone on Smartphone.Utilizzatore = Utente.CodFiscale
JOIN Valutazione on Valutazione.Account = Account.EMail)
WHERE Valutazione.CodiceApp=1
ORDER BY EMail;
```

EMail	NomeCognome	Modello	Voto	Recensione
baranto@icloud.com	Barbara Antonello	Iphone 7	4	molto meglio dopo l ultimo aggiornamento
baranto@icloud.com	Barbara Antonello	Iphone 7	2	da sistemare
frapec@gmail.com	Francesco Pecile	Huawei P10 Lite	3	NULL
marcmasi@icloud.com	Marco Masiero	Iphone 4C	3	qualche bug da aggiustare
marrosso@outlook.com	Mario Rossi	HTC One	5	NULL
mattedepasca@gmail.com	Matteo Depascale	Nexus 5X	4	molto carino
nicociste@outlook.com	Nicola Cisternino	Nokia Lumia 950	2	NULL

5) Query che ci restituisce il numero di acquisti delle app a pagamento, il nome dell'app e il relativo costo.
Vogliamo anche ordinare il risultato in modo che il primo risultato abbia il numero di acquisti maggiore.

```
SELECT CodiceApp, NomeApp, COUNT(App.CodiceApp) as NumeroAcquisti, Prezzo
FROM (App JOIN Acquisto on App.CodiceApp = Acquisto.CodApp)
GROUP BY CodiceApp
ORDER BY NumeroAcquisti DESC;
```

CodiceApp	NomeApp	NumeroAcquisti	Prezzo
20	Semplice è bello	3	13.9
19	Tutto Sherlock Holmes	2	4.99
11	Hello	2	1.3
7	Annabelle	2	3.99
23	Minions	1	13.33
15	Puzzle Bubble	1	5.5
8	Evolve	1	8.5
31	La signora in giallo - omicidio sul ghiaccio	1	6.99
24	I Puffi	1	5.99
16	Netflix	1	10
9	Kind of Blue	1	7.99
6	Fast&Furious 8	1	13.99
29	Ghost in the Shell	1	9.99
21	Come istruire un robot	1	18
18	Il Signore Degli Anelli	1	13.89
13	Neo Monster	1	1.3
30	Saw IV	1	5.99

6) Top 5 App più scaricate.
N.B. La query utilizza la funzione NDownl definita dall'utente

```
SELECT CodiceApp, NomeApp, NomeGenere, Categoria, NDownl(CodiceApp) as
Top5Download
FROM App JOIN Genere on App.Genere = Genere.CodGenere
GROUP BY CodiceApp
ORDER BY Top5Download DESC LIMIT 5;
```

CodiceApp	NomeApp	NomeGenere	Categoria	Top5Download
1	Facebook	Funzionalità	Social Network	6
4	TuttoMercatoWeb	Funzionalità	Sport	3
14	Candy Crush Jelly Saga	Giochi	Puzzle	3
2	Clash of Clans	Giochi	Strategia	3
20	Semplice è bello	Libri	Tecnologia	3

5.2 Funzioni

1) Funzione che conta il numero di download di una certa App, basandoci sulla ListaDownload.

```
DELIMITER $$
CREATE FUNCTION `NDownl` (CodiceApp int)
RETURNS int

BEGIN
DECLARE App int;

SELECT count(CodiceApp)
INTO App
FROM ListaDownload
WHERE AppScaricata = CodiceApp;

RETURN (App);
END$$
```

2) Funzione che restituisce la media aritmetica dei Voti rilasciati ad un App tramite la tabella Valutazioni.

```
DELIMITER $$
CREATE FUNCTION `MediaVoti` (CodiceApp int)
RETURNS float
BEGIN
DECLARE Media float;

SELECT AVG(Voto)
        INTO Media
        FROM Valutazione
        WHERE Valutazione.CodiceApp = CodiceApp;
RETURN Media;
END$$
```

5.3 Trigger

1) Vogliamo un trigger che, prima di inserire una nuova Recensione, controlli che l'Account recensore abbia effettivamente scaricato l'App da recensire.

```
DELIMITER $$
CREATE TRIGGER ControllaValutazione
BEFORE INSERT ON `Valutazione`
FOR EACH ROW
BEGIN
DECLARE Account varchar (45);
DECLARE App int;
DECLARE Voto int;
DECLARE Recensione varchar (100);
IF (ListaDownload.AppScaricata = App AND ListaDownload.Account=Account)
        THEN SET NEW.Valutazione = CONCAT(Account, App, Voto, Recensione);
END IF;
END $$
```

2) Trigger che, dopo la registrazione di un nuovo Utente, crea un record nelle tabelle Account e Smartphone.

```
DELIMITER $$
CREATE TRIGGER Aggiornamento
AFTER INSERT ON `Utente`
FOR EACH ROW
BEGIN
DECLARE Mail varchar(45);
DECLARE Pass varchar(10);
DECLARE Model varchar(20);
INSERT INTO Account (Utente, EMail, Password) values (NEW.CodFiscale, Mail, Pass);
INSERT INTO Smartphone (Utilizzatore, Modello) values (NEW.CodFiscale, Model);
END $$
```