

PROGETTO BASE DI DATI 2016/2017

“FUMETTERIA”

Nicola Cisternino 1123075

Marco Masiero 1124621

1 Abstract

La crescente passione per i Manga che ha pervaso tutto il mondo a partire dagli anni '90 ha attratto anche una grande percentuale di lettori Italiani.

Le Fumetterie nascono proprio per soddisfare questa esigenza, proponendosi come collegamento tra oriente e occidente, fornendo ai lettori la possibilità di acquistare queste opere.

Il ruolo di una Fumetteria non si limita tuttavia alla vendita di manga, ma tra i prodotti che è possibile trovare vi sono:

- Action Figure, modellini raffiguranti personaggi di Anime/Manga famosi che catturano l'interesse dei collezionisti;
- Giochi di Ruolo, variante dei classici giochi da tavolo dove i giocatori si calano nel ruolo di uno o più personaggi immaginari;
- DVD/Anime, raccolta di episodi, OAV (Original Anime Video) e Movie riguardanti le serie manga famose in Giappone;
- Gadget a tema.

Il fine di questo progetto è la creazione di un database che organizzi la gestione interna di una Fumetteria.

Il database terrà traccia degli acquisti e delle relative vendite, con particolare attenzione alla disponibilità degli articoli. Oltre a queste operazioni elementari, sarà dotato di uno strumento di analisi delle vendite, finalizzato a gestire le esigenze della clientela: attraverso lo storico delle vendite sarà possibile stimare quali, tra le serie reperibili, siano le più richieste e, di conseguenza, adeguare i successivi acquisti alla domanda. Ai clienti dotati di tessera sarà permesso effettuare ordinazioni. Inoltre, il database è strutturato in modo tale che qualsiasi operazione di ricerca (aggiornamento/rimozione) verrà eseguita nel modo più efficiente possibile.

2 Analisi dei requisiti

Si vuole progettare una base di dati che contenga e organizzi le informazioni riguardanti un negozio di fumetti. In particolare si vuole gestire la compravendita di articoli, distinti attraverso un codice identificativo unico. Gli **articoli** appartengono a differenti categorie:

Manga e Fumetti sono i principali prodotti per cui nasce questa attività. Essi appartengono a serie differenti e possono avere più generi letterali. Per ogni Manga e Fumetto vengono mantenute informazioni relative all'autore dell'opera, il volume e la sua casa editrice.

Dvd e Anime sono altri prodotti derivanti dal successo dei fumetti e di essi si vuole registrare il titolo del film, della saga. In più viene fornita una descrizione della trama trattata.

Action Figure sono modellini interi o assemblabili, di cui si tiene traccia delle descrizioni generali e dell'età minima dell'acquirente.

Giochi di Ruolo, dei quali ci interessa mantenere una descrizione dell'articolo e dell'età minima dell'acquirente.

Gadget, di cui si vuole mantenere le informazioni sulle varie tipologie a cui appartengono (come magliette, tazze, portachiavi ecc...) e una breve descrizione delle loro caratteristiche.

Di tutti gli articoli presente in negozio, vengono registrati i prezzi di acquisto e di vendita, la data di pubblicazione, la disponibilità e le case editrici o produttrici. E' necessario avere informazioni relative alle **serie**, macrocategorie attraverso cui gli articoli vengono accomunati. Ogni articolo presente in negozio viene distribuito da un'azienda specializzata nel rifornire le fumetterie. Queste verranno indicate come **Fornitori** e distinte tramite un codice identificativo proprio. Altre informazioni di rilievo sono i dati anagrafici, quali partita IVA, ragione sociale, E-mail, numero di telefono e i dati relativi all'indirizzo della sede.

E' possibile ricavare i dati degli acquisti per mezzo delle fatture registrate. Le quantità disponibili a magazzino vengono incrementate all'atto di registrazione delle fatture ricevute.

I dati distintivi di una **fattura** si compongono quindi delle informazioni del fornitore, il numero della fattura stessa e la data di emissione.

Ciascuna fattura necessita di queste informazioni per essere identificata univocamente, in modo tale da non presentare ambiguità nell'eventualità di:

- assenza di codice fornitore, due o più fornitori distinti presentano una fattura nello stesso giorno e con lo stesso numero;
- assenza di data, lo stesso fornitore emette lo stesso numero di fattura a distanza di anni;
- assenza di numero fattura, un fornitore emette più fatture nello stesso giorno;

Sarà essenziale tenere conto della data di scadenza entro cui è possibile disporre del pagamento e il costo totale della fattura. E' importante che il negozio abbia la possibilità di concordare un metodo di pagamento differente per ciascuna fattura ricevuta da un fornitore. E' opportuno mantenere i dati dei vari **metodi di pagamento** disponibili. Per far ciò viene utilizzato un identificatore unico, una descrizione del pagamento, e il termine (totale di giorni entro cui saldare la fattura) associato al pagamento.

Le informazioni riguardanti i pagamenti delle fatture comprendono i dati sui **conti correnti** di cui dispone un fornitore. Questi sono identificati da IBAN, unico per ogni conto corrente e il nome del titolare del conto.

Il negozio offre la possibilità di sottoscrivere una **tessera** annuale rinnovabile per i clienti, la quale può essere di tre differenti gradi: Bronze, Silver, Gold. Ognuna di queste presenta un costo differente ed inoltre ha associato una specifica percentuale di sconto sul totale degli articoli acquistati. Una tessera viene riconosciuta univocamente all'interno della base di dati tramite un identificativo. Vengono mantenuti i dati relativi alla data di iscrizione e scadenza della tessera in modo tale da poter verificare in ogni momento gli utenti beneficiari delle agevolazioni derivanti dalla sottoscrizione. La politica del negozio offre la possibilità, ogni 10 acquisti effettuati in un anno, di cambiare il proprio tipo di tessera avanzando al tipo superiore. Le informazioni richieste al **cliente** per la sottoscrizione della tessera sono: nome, cognome, data di nascita e una E-mail per eventuali contatti. Ad ogni cliente viene assegnato un codice identificativo unico, poiché i dati anagrafici vengono mantenuti nel database anche dopo l'eventuale scadenza della tessera.

Viene offerta la possibilità di **ordine** dei vari prodotti ai soli clienti possessori di una tessera, così da dare l'opportunità di richiedere articoli al momento non disponibili o non ancora pubblicati. Le vendite sono registrate in **note di vendite**, che contengono tutti i prodotti disponibili richiesti dal cliente. Questi vanno a formare la nota completa, al netto dello sconto se il cliente è in possesso di una tessera. La nota di vendita si rivela particolarmente utile in quanto, attraverso un'attenta analisi, si può risalire ai prodotti più richiesti, i generi letterari più lette e le serie di tendenza, indirizzando i futuri approvvigionamenti per soddisfare sempre la domanda della clientela.

3 Progettazione Concettuale

3.1 Entità, Attributi e Gerarchie

- **Articolo:** modella un articolo generico registrato nel database. Si tratta dell'Entità principale del database, in quanto la gestione di acquisti e vendite del negozio è incentrata su di essi.

Attributi:

- CodiceArticolo: int (Identificatore)
- CodiceFornitore: int (Identificatore Esterno)
- Nome: string
- PrezzoAcquisto: Decimal
- PrezzoVendita: Decimal
- Disponibilità: int
- CodiceSerie: int (Identificatore Esterno)
- Genere: string (Attributo multivalore)
- DataPubblicazione: Date
- CasaEditrice: string

Gerarchia - Entità figlie

- **Manga/fumetto:**
 - Volume: int
 - Edizione: string
 - Autore: string
- **Action Figure:**
 - Descrizione: string
 - Età: int
- **Giochi:**
 - Descrizione: string
 - Età: int
- **Gadget:**
 - Descrizione: string
 - Sottocategoria: string
- **DVD/Anime:**
 - Descrizione: string

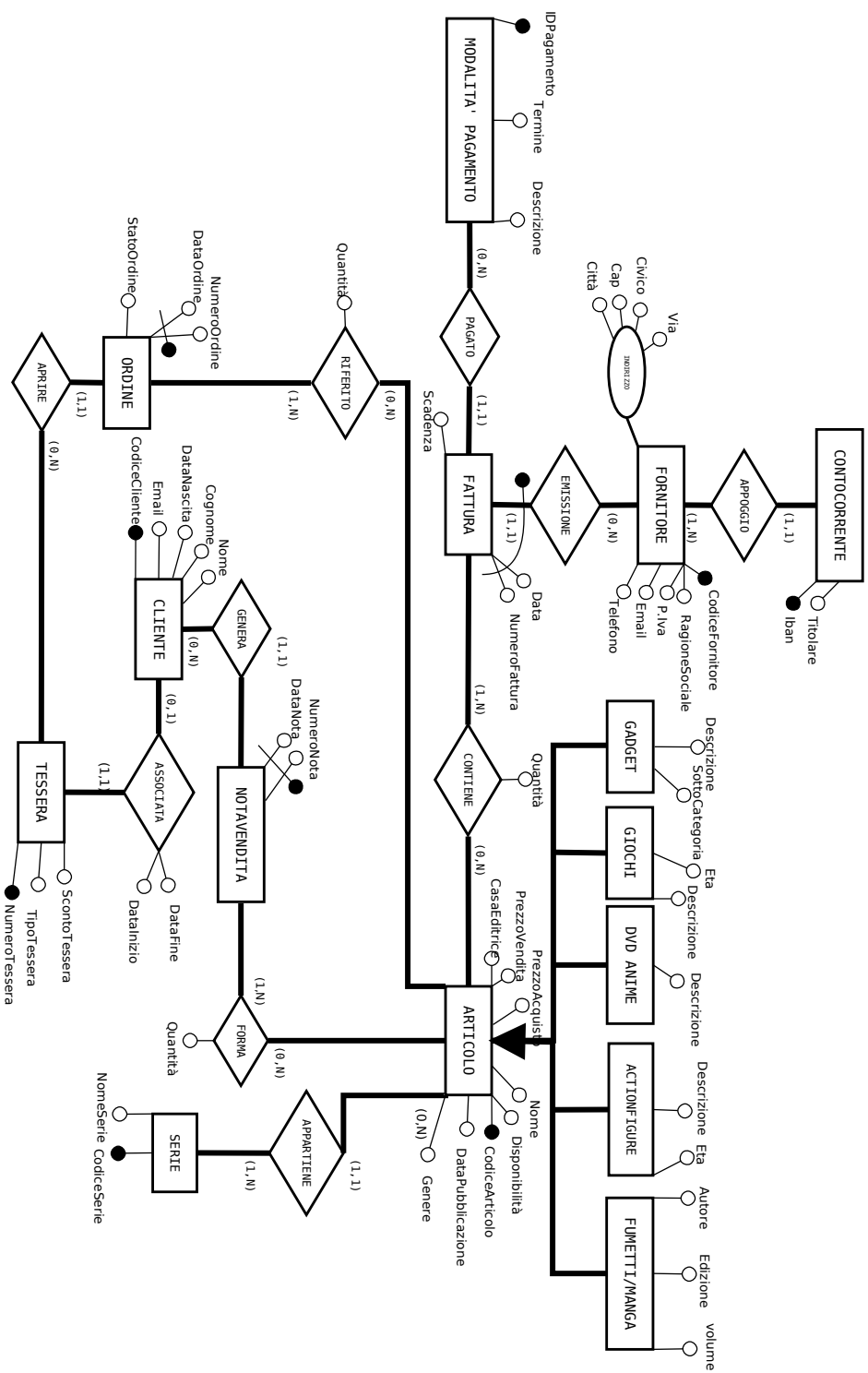
Rappresenta una generalizzazione totale di articolo.

- **Serie:** Modella una serie generica che raggruppa tutti gli articoli che appartengono ad una specifica serie animata o letteraria. Ogni oggetto venduto all'interno del negozio ha una serie a cui appartiene.
Attributi:
 - CodiceSerie: int (Identificatore)
 - NomeSerie: string
- **Fornitore:** modella le generalità di un Fornitore, il quale può essere un distributore di articoli particolari o in qualche caso la casa editrice stessa.
Attributi:
 - CodiceFornitore: int (Identificatore)
 - P.Iva: string
 - RagioneSociale: string
 - Telefono: string
 - Email: string
 - Indirizzo (Attributo Composto):
 - Via: string
 - Civico: string
 - Città: string
 - Cap: string
- **Fattura:** Modella le Fatture d'acquisto emesse dai Fornitori che vengono registrate all'interno del database.
Attributi:
 - NumeroFattura: int (Identificatore)
 - Data: date (Identificatore)
 - CodiceFornitore: int (Identificatore Esterno)
- **Modalità di Pagamento:** Contiene i vari metodi di pagamento con cui è permesso il pagamento delle fatture emesse dai Fornitori. "Termine" contiene i giorni entro cui deve essere pagata la fattura.
Attributi:
 - IDPagamento: int (Identificatore)
 - Descrizione: string
 - Termine: int
- **Conto Corrente:** Contiene i conti correnti relativi ai Fornitori.
Attributi:
 - Iban: string (Identificatore)
 - Titolare: String
 - CodiceFornitore (Identificatore Esterno)
- **Tessera:** Modella le informazioni sulle tessere registrate ancora valide.
Attributi:
 - NumeroTessera: int (Identificatore)
 - TipoTessera: string
 - DataInizio: date
 - DataFine: date
 - ScontoTessera: float
- **Cliente:** Modella un cliente generico che ha acconsentito all'utilizzo dei propri dati sensibili per la sottoscrizione di una tessera. Le informazioni dei clienti che non hanno mai sottoscritto una tessera non sono mantenute nel database.
Attributi:
 - CodiceCliente: int (Identificatore)
 - Nome: string
 - Cognome: string
 - DataNascita: date
 - Email: string
 - NumeroTessera: int (Identificatore Esterno)

- **Ordine:** Modella un ordine generico aperto dai clienti tesserati.
Attributi:
 - NumeroOrdine: int (Identificatore)
 - DataOrdine: date (Identificatore)
 - CodiceCliente: int (Identificatore Esterno)
 - StatoOrdine: bool
- **Nota Vendita:** Contiene informazioni relative alle note emesse per avvenuta vendita degli articoli presenti in negozio.
Attributi:
 - NumeroNota: int (Identificatore)
 - DataNota: date (Identificatore)
 - CodiceCliente: int (Identificatore Esterno)

3.2 Relazioni e Cardinalità:

- **Fornitore - Conto Corrente: Appoggio**
 - (1,N): Una Fornitore è titolare di almeno un conto corrente;
 - (1,1): Un conto corrente è legato ad un unico Fornitore;
- **Fornitore - Fattura: Emissione**
 - (0,N): Un Fornitore può emettere zero o più fatture;
 - (1,1): Ogni fattura registrata è relativa ad un unico fornitore;
- **Fattura - Modalità di Pagamento: Pagato**
 - (1,1): Una fattura è pagata solo con un determinato metodo di pagamento;
 - (0,N): Un metodo di pagamento può essere associato a zero o più fatture;
- **Fattura - Articolo: Contiene**
 - (1,N): Una fattura contiene uno o più articoli;
 - (0,N): Zero o più articoli possono essere presenti in una fattura;
- **Articolo - Nota vendita: Forma**
 - (0,N): Zero o più articolo può essere presente in una nota di vendita;
 - (1,N): Una nota di vendita deve essere costituita da almeno un articolo o più;
- **Tessera - Cliente: Associata**
 - (1,1): Una tessera, al massimo una, può essere associata ad un cliente;
 - (0,1): Una cliente può avere o non avere una tessera associata;
- **Tessera - Ordine: Aprire**
 - (0,N): Solo chi ha una tessera attiva può aprire o no più ordini;
 - (1,1): Un determinato ordine è aperto esclusivamente da cliente tesserato;
- **Ordine - Articolo: Riferito**
 - (1,N): Un ordine contiene almeno un articolo;
 - (0,N): Un articolo può far parte di un ordine oppure no;
- **Cliente - Nota Vendita: Genera**
 - (0,N): Un cliente genera zero o più note di vendita;
 - (1,1): Una nota di vendita è generata da un unico cliente;
- **Articolo - Serie: Appartiene**
 - (0,1): Un articolo può appartenere o meno ad una serie;
 - (1,N): Una serie può essere associata a uno o più articoli;



4 Progettazione Logica

4.1 Eliminazione delle generalizzazioni:

Si è deciso di mantenere l'entità Articolo accorpando le entità figlie nell'entità genitore con l'introduzione di un attributo apposito **Categoria**, per preservare la distinzione tra le occorrenze di tale entità che la generalizzazione originaria rappresentava. Gli attributi propri delle entità figlie non comuni per ogni occorrenza possono assumere il valore NULL. Si è preferito un minor numero di accessi a scapito di uno spreco di memoria per la presenza dei valori nulli.

4.2 Eliminazione Attributo Composto:

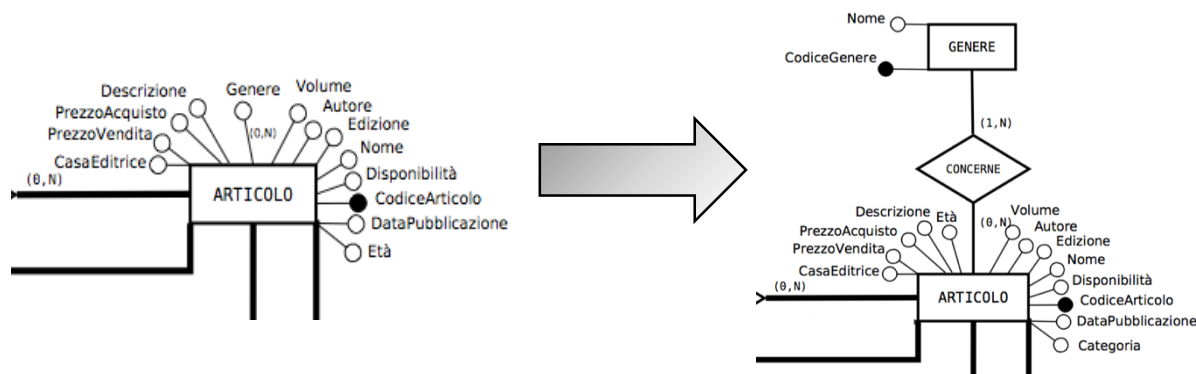
L'attributo composto "Indirizzo" appartenente all'entità Fornitore viene eliminato, ciascuno dei suoi attributi viene inserito nell'entità principale.

4.3 Eliminazione Attributi Multivalore:

Viene partizionato l'entità Articolo avente l'attributo multivalore Genere in due entità: un'entità con lo stesso nome e gli stessi attributi dell'entità originale eccetto l'attributo multivalore, e l'entità Genere con attributi:

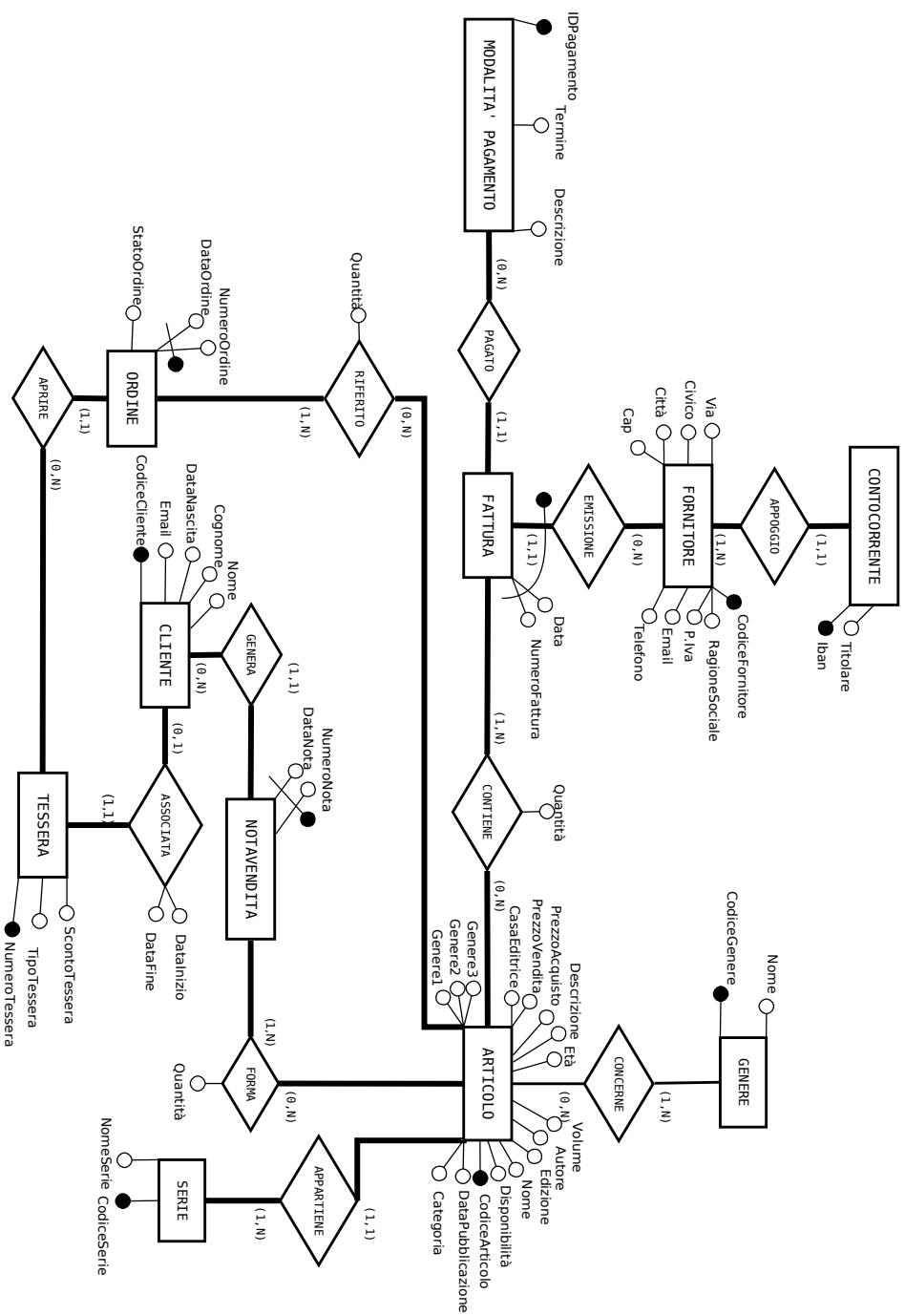
- **CodiceGenere**, distintivo per ogni tipologia di genere;
- **Nome**, nome della tipologia;

L'entità Articolo ora contiene la chiave esterna di Genere ed è legato mediante un'associazione molti a molti.

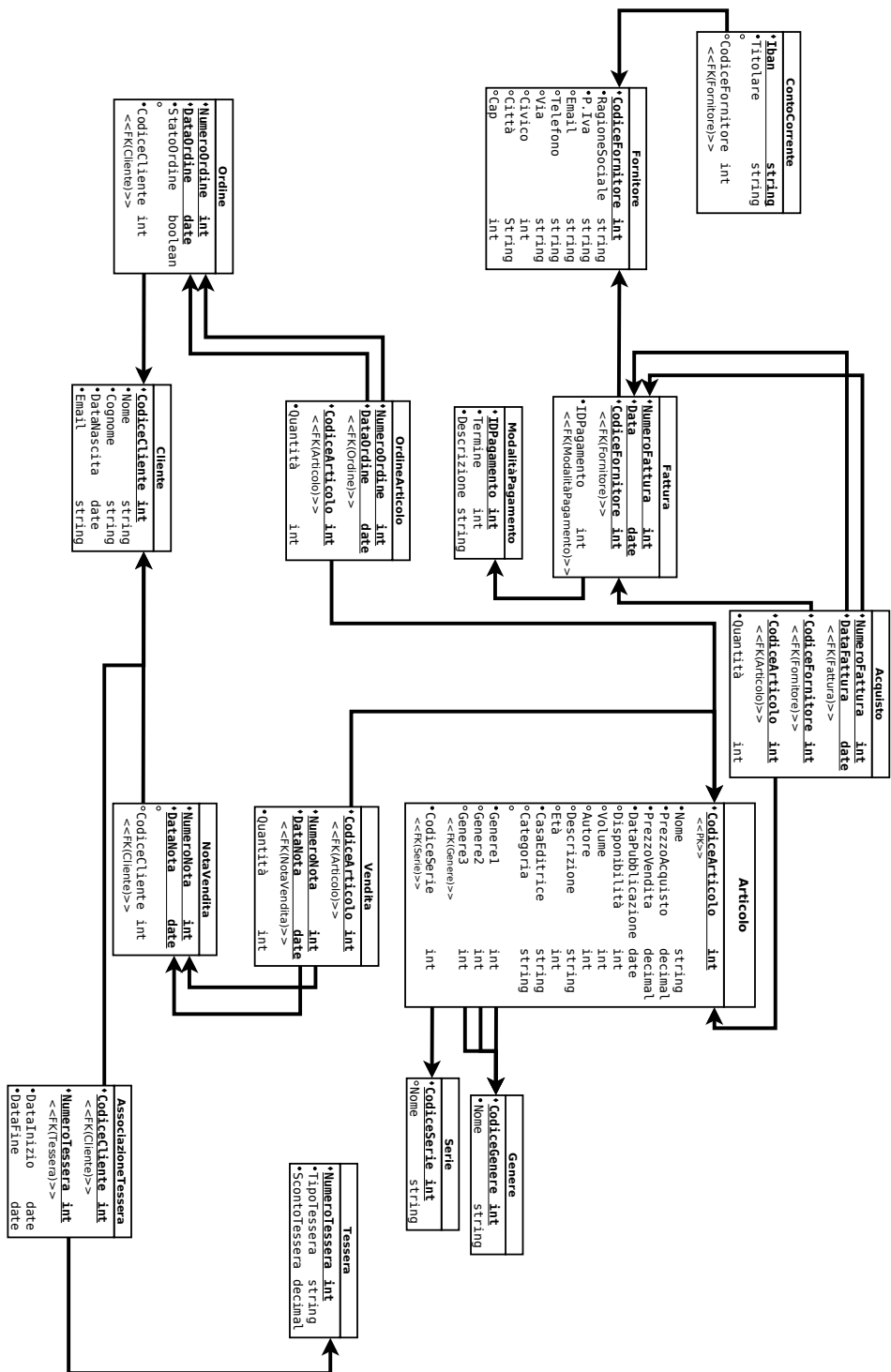


4.4 Analisi delle Ridondanze

- All'interno dell'entità Articolo è presente l'attributo **Disponibilità**, che risulta derivabile facendo una differenza tra le quantità acquistate (provenienti dall'entità Fattura) e le quantità vendute (ricavabili dall'entità NotaVendita). Considerando che l'operazione di "controllo disponibilità" viene effettuata in concomitanza della creazione di una riga nelle tabelle Fattura e NotaVendita, si ritiene sia di maggior utilità mantenere l'attributo Disponibilità, gestendo tale operazione attraverso un trigger.
- Si è optato di rimuovere l'attributo **Scadenza** dalla tabella Fattura perché ritenuto superfluo. E' possibile ottenerlo attraverso la data della Fattura d'interesse e il termine ultimo corrispondente alla modalità di pagamento associata.
- La tabella Articolo, in questa base di dati, risulta essere quella di maggiore importanza, non solo per la funzione di inventario, ma soprattutto per la mole di dati che essa contiene. E' opportuno dunque che tutte le operazioni su essa siano il più efficienti possibili e che non vi siano ridondanze non necessarie. A questo proposito l'attributo genere si comporta in modo problematico. Poiché un articolo (come ad esempio un manga) può essere inerente a più di un genere, sarebbe necessario che il suo record comparisse più di una volta, creando quindi una ridondanza inutile. A tal proposito si è deciso di optare per l'utilizzo di tre attributi genere, che rispecchiano i generi principali del prodotto. In questo modo un articolo con un massimo di generi comparirà una sola volta, mentre un articolo con meno di tre generi avrà posti a NULL gli attributi non necessari. Tale modifica ci consente anche di usufruire dell'attributo "Genere1" con la funzione di sottocategoria per gli articoli inerenti all'editoria.



4.4 Schema Logico



4.5 Chiavi Esterne

- **CodiceSerie:**
E' chiave esterna tra le entità "Articolo" → "Serie";
- **CodiceGenere:**
E' chiave esterna tra le entità "Articolo" → "Genere";
(nella tabella articolo si riferisce a Genere1, Genere2, Genere3)
- **CodiceFornitore:**
E' chiave esterna tra le entità "Fatture" → "Fornitore";
E' chiave esterna tra le entità "ContoCorrente" → "Fornitore";
E' chiave esterna tra le entità "Acquisto" → "Fornitore";
- **CodiceCliente:**
E' chiave esterna tra le entità "NotaVendita" → "Cliente";
E' chiave esterna tra le entità "Ordine" → "Cliente";
E' chiave esterna tra le entità "AssociazioneTessera" → "Cliente"
- **NumeroTessera:**
E' chiave esterna tra le entità "AssociazioneTessera" → "Tessera";
- **IDPagamento:**
E' chiave esterna tra le entità "Fattura" → "ModalitàPagamento";
- **NumeroFattura - DataFattura:**
E' chiave esterna tra le entità "Acquisto" → "Fattura";
- **NumeroNota - DataNota:**
E' chiave esterna tra le entità "Vendita" → "NotaVendita";
- **NumeroOrdine - DataOrdine:**
E' chiave esterna tra le entità "OrdineArticolo" → "Ordine";

4.6 Schema Relazionale

- **Entità:**
GENERE (CodiceGenere, Nome)
SERIE (CodiceSerie, Nome)
ARTICOLO (CodiceArticolo, Categoria, Nome, PrezzoAcquisto, PrezzoVendita, DataPubblicazione, Disponibilita, Volume, Autore, Descrizione, Età, CasaEditrice, CodiceSerie, Genere1, Genere2, Genere3)
FATTURA (NumeroFattura, DataFattura, CodiceFornitore, IDPagamento)
MODALITAPAGAMENTO (IDPagamento, Termine, Descrizione)
NOTAVENDITA (NumeroNota, DataNota, CodiceCliente)
CLIENTE (CodiceCliente, Nome, Cognome, DataNascita, Email)
TESSERA (NumeroTessera, TipoTessera, ScontoTessera)
ORDINE (NumeroOrdine, DataOrdine, StatoOrdine, CodiceCliente)
FORNITORE (CodiceFornitore, RagioneSociale, PIVA, Email, Telefono, Via, Civico, Città, Cap)
CONTOCORRENTE (Iban, Titolare, CodiceFornitore)
- **Associazioni molti a molti:**
ORDINEARTICOLO (Ordine, DataOrdine, Articolo, Quantita)
ACQUISTO (Fattura, DataFattura, Fornitore, Articolo, Quantita)
VENDITA (Articolo, NotaVendita, DataNota, Quantita)
CONCERNE (Articolo, Genere)
- **Associazioni uno a uno:**
Associata:
(0,1) Cliente - (1,1) Tessera
Dato che l'entità Cliente ha partecipazione opzionale all'associazione Associata si è deciso di optare per l'accorpamento degli attributi dell'associazione interessata nella relazione Tessera assieme alla chiave primaria di Cliente ammessa come identificatore esterno.
TESSERA (NumeroTessera, TipoTessera, ScontoTessera, Cliente, DataInizio, DataFine)
- **Associazioni uno a molti:**
Emissione: (1,1) Fattura - (0,N) Fornitore
Appoggio: (1,1) ContoCorrente - (1,N) Fornitore
Pagato: (0,N) ModalitàPagamento - (1,1) Fattura
Aprire: (1,1) Ordine - (0,N) Tessera
Genera: (1,1) NotaVendita - (0,N) Cliente

COMPLETO:

GENERE (CodiceGenere, Nome)

SERIE (CodiceSerie, Nome)

ARTICOLO (CodiceArticolo, Categoria, Nome, PrezzoAcquisto, PrezzoVendita, DataPubblicazione, Disponibilita, Volume, Autore, Descrizione, Eta, CasaEditrice, Serie, Genere1, Genere2, Genere3)

FATTURA (Fattura, DataFattura, Fornitore, IDPagamento)

MODALITAPAGAMENTO (IDPagamento, Termine, Descrizione)

NOTAVENDITA (NotaVendita, DataNota, Cliente)

CLIENTE (CodiceCliente, Nome, Cognome, DataNascita, Email)

TESSERA (NumeroTessera, TipoTessera, ScontoTessera, Cliente, DataInizio, DataFine)

ORDINE (NumeroOrdine, DataOrdine, StatoOrdine, Cliente)

FORNITORE (CodiceFornitore, RagioneSociale, PIVA, Email, Telefono, Via, Civico, Città, Cap)

CONTOCORRENTE (Iban, Titolare, Fornitore)

ORDINEARTICOLO (Ordine, DataOrdine, Articolo, Quantita)

ACQUISTO (Fattura, DataFattura, Fornitore, Articolo, Quantita)

VENDITA (Articolo, NotaVendita, DataNota, Quantita)

5 Codice SQL

QUERY 1: Per ogni cliente che ha acquistato almeno un articolo viene selezionato l'acquisto più dispendioso effettuato.

```
drop view if exists PrezzoNoteVendita;  
create view PrezzoNoteVendita as  
select NotaVendita, DataNota, sum(v.Quantita*PrezzoVendita) as NotaLordo  
from vendita v join articolo a on (v.Articolo=a.CodiceArticolo)  
group by NotaVendita, DataNota;
```

```
drop view if exists bestnotavendita;  
create view bestnotavendita as  
select nv.Cliente, (pv.NotaLordo) as Nota  
from notavendita nv join PrezzoNoteVendita pv on(pv.NotaVendita=nv.NotaVendita AND  
pv.DataNota=nv.DataNota)  
where pv.NotaLordo=(select max(pv2.NotaLordo) from PrezzoNoteVendita pv2 join notavendita nv2  
on(pv2.NotaVendita=nv2.NotaVendita AND pv2.DataNota=nv2.DataNota)  
where nv.Cliente=nv2.Cliente)  
order by nv.Cliente;
```

```
drop view if exists bestnotatesserati;  
create view bestnotatesserati as  
select bv2.Cliente, (bv2.Nota-(bv2.Nota*t.ScontoTessera))*1.22 as Nota  
from bestnotavendita bv2 join tessera t on(bv2.Cliente=t.Cliente)  
where CURDATE()<=DataFine  
order by bv2.Cliente;
```

```
drop view if exists bestnotanontesserati;  
create view bestnotanontesserati as  
select Cliente,(Nota*1.22) as Nota from bestnotavendita where Cliente not in(select Cliente from  
bestnotatesserati) order by Cliente;
```

```
select * from bestnotatesserati UNION select * from bestnotanontesserati order by Cliente;
```

+ Opzioni

Ciente	1	Nota
1	164.260800	
2	477.508000	
3	22.295500	
4	463.553640	
5	105.530000	
6	21.960000	
7	120.780000	
8	167.750000	
9	547.627500	
10	174.582000	
11	455.670000	
12	5.246000	
13	361.486000	
14	21.960000	
15	1185.774120	
16	18.300000	

Query 2: Prezzo totale di ogni nota di vendita compresa di sconto se il cliente possiede una tessera.

drop view if exists prezzonota;

create view prezzonota as select v.NotaVendita,v.DataNota,sum(v.Quantita*a.PrezzoVendita) as PrezzoTotaleNota

from articolo a join vendita v on CodiceArticolo=Articolo group by NotaVendita;

drop view if exists notetessera;

create view notetessera as select n.NotaVendita,n.DataNota,n.Ciente,p.PrezzoTotaleNota
from notavendita n join prezzonota p on (n.NotaVendita=p.NotaVendita and n.DataNota=p.DataNota)
where n.Ciente IN(select Ciente from tessera);

drop view if exists notetessera2;

create view notetessera2 as select n.NotaVendita,n.DataNota,(n.PrezzoTotaleNota-
(n.PrezzoTotaleNota*t.ScontoTessera))*1.22 as PrezzoTotaleNota from notetessera n join tessera t
on(n.Ciente=t.Ciente);

drop view if exists notenontessera;

create view notenontessera as select n.NotaVendita,n.DataNota,n.Ciente,p.PrezzoTotaleNota
from notavendita n join prezzonota p on (n.NotaVendita=p.NotaVendita and n.DataNota=p.DataNota)
where n.Ciente NOT IN(select Ciente from tessera) or n.Ciente is NULL;

drop view if exists notetotali;

create view notetotali as select NotaVendita,DataNota,PrezzoTotaleNota from notetessera2 UNION select
NotaVendita,DataNota,PrezzoTotaleNota from notetotali group by NotaVendita;

select NotaVendita,DataNota,PrezzoTotaleNota from notetotali order by NotaVendita;

NotaVendita	DataNota	PrezzoTotaleNota
1	2017-02-01	14.951100
2	2017-02-01	137.500000
3	2017-02-01	120.780000
4	2017-02-02	9.967400
5	2017-02-18	15.000000
6	2017-02-22	22.295500
7	2017-02-25	12.900000
8	2017-03-02	387.319500
9	2017-03-10	34.770000
10	2017-03-11	143.100000
11	2017-03-11	90.000000
12	2017-03-23	296.300000
13	2017-04-06	164.260800
14	2017-04-10	477.508000
15	2017-04-29	547.627500

16	2017-05-01	1185.774120
17	2017-05-02	4.300000
18	2017-05-15	86.500000
19	2017-05-24	825.000000
20	2017-06-01	18.000000
21	2017-06-01	18.000000
22	2017-06-01	19.764000
23	2017-06-10	463.553640
24	2017-06-10	4.983700
25	2017-06-11	4.983700
26	2017-06-12	4.983700
27	2017-06-13	6.374500
28	2017-06-14	6.374500
29	2017-06-15	6.374500
30	2017-06-16	6.374500

QUERY 3: Top 5 articoli più venduti nel mese precedente a quello corrente che non facciano parte della categoria giochi

```
select CodiceArticolo, Nome, Categoria, Genere1, Genere2, Genere3, Venduto(CodiceArticolo) as Top5Vendite
from articolo a join vendita v on (v.Articolo=a.CodiceArticolo)
where MONTH(v.DataNota)=MONTH(CURDATE())-1 AND YEAR(v.DataNota)=YEAR(CURDATE()) AND
a.Categoria<>'Giochi' group by CodiceArticolo order by Top5Vendite DESC LIMIT 5;
```

CodiceArticolo	Nome	Categoria	Genere1	Genere2	Genere3	Top5Vendite
35	Sanji Battle Version	ActionFigure	16	NULL	NULL	4
57	Naruto Movie 5	DVD	9	10	3	1
60	Naruto Movie 8	DVD	9	10	3	1
66	One Piece Movie 6	DVD	9	10	3	1
69	One Piece Movie 9	DVD	9	10	3	1

QUERY 4: Fornitori a cui rimangono ancora fatture, di cui non è ancora scaduto il termine di pagamento, dove compaiono articoli di categoria Gadget

drop view if exists scadenze;

create view scadenze as

```
select Fattura, DataFattura as Data, Fornitore as Fornitore1, Scadenza(IDPagamento, DataFattura) as
Scadenza
from fattura;
```

drop view if exists acquistifattureinscadenza;

create view acquistifattureinscadenza as

```
select *
from scadenze s join acquisto a on (s.Fattura=a.NumeroFattura and s.Data=a.DataFattura and
s.Fornitore1=a.Fornitore)
where s.Scadenza>=0;
```

drop view if exists articolifattureinscadenza;

create view articolifattureinscadenza as

```
select Fattura, DataFattura, Fornitore, Scadenza
from acquistifattureinscadenza afs join articolo a on afs.Articolo=a.CodiceArticolo
where a.Categoria='Gadget';
```

Select f.CodiceFornitore, f.RagioneSociale, afs.Scadenza

from articolifattureinscadenza afs join fornitore f on f.CodiceFornitore=afs.Fornitore group by afs.Scadenza;

+ Opzioni

CodiceFornitore	RagioneSociale	Scadenza
6	J-POP	13
6	J-POP	44

QUERY 5: Serie di maggior interesse degli ultimi tre mesi (compreso quello corrente come terzo mese) di cui fanno parte manga con i generi Avventura, Fantasy o entrambi

```
drop view if exists fantasy;
create view fantasy as
  select *
  from articolo
  where Genere1=4 or Genere2=4 or Genere3=4;
```

```
drop view if exists avventura;
create view avventura as
  select *
  from articolo
  where Genere1=9 or Genere2=9 or Genere3=9;
```

```
drop view if exists fantavventura;
create view fantavventura as
  select * from fantasy UNION select * from avventura;
```

```
select s.CodiceSerie,s.Nome,count(*) as VenditeSerie
from articolo a join serie s on(s.CodiceSerie=a.Serie) join vendita v on(a.CodiceArticolo=v.Articolo)
where a.Categoria='Manga' and a.CodiceArticolo IN (select f.CodiceArticolo from fantavventura f) and
MONTH(v.DataNota)>=MONTH(CURDATE())-2
group by s.CodiceSerie order by VenditeSerie DESC;
```

+ Opzioni

CodiceSerie	Nome	VenditeSerie
2	Naruto	7
1	One Piece	4

QUERY 6: Tutti i clienti tesserati a febbraio dell'anno corrente che hanno aperto almeno un'ordine coprensivo di un articolo della serie One Piece e senza un articolo della serie Naruto

```
drop view if exists seriediunordine;
create view seriediunordine as
  Select oa.Ordine, s.Nome
  from ordinearticolo oa join articolo a on (oa.Articolo=a.CodiceArticolo)
  join serie s on(a.Serie=s.CodiceSerie);
```

```
select c.CodiceCliente,c.Nome,c.Cognome
from cliente c join tessera t on(c.CodiceCliente=t.Cliente) join ordine o on(t.Cliente=o.Cliente) join
ordinearticolo oa on(o.NumeroOrdine=oa.Ordine AND o.DataOrdine=oa.DataOrdine) join
articolo a on(oa.Articolo=a.CodiceArticolo) join serie s on(a.Serie=s.CodiceSerie)
where t.DataInizio>'2017-01-31' AND t.DataInizio<'2017-03-01' AND s.Nome='One Piece' AND
o.NumeroOrdine not in(select Ordine from seriediunordine where Nome='Naruto' ) group by
c.CodiceCliente;
```

CodiceCliente	Nome	Cognome
1	Nicola	Cisternino

Funzione 1: Calcolo scadenza pagamento fattura

```
DELIMITER $$
CREATE FUNCTION `Scadenza`(IDPagamento INTEGER(10),DataFattura DATE)
RETURNS int(11)
BEGIN
DECLARE termine int(11);
DECLARE datacor DATE;
DECLARE giornitrascorsi int(11);
DECLARE Scadenza int(11);

SELECT
    m.Termine
INTO termine
FROM
    modalitapagamento as m
WHERE
    IDPagamento = m.IDPagamento;
SELECT CURDATE() INTO datacor;
SELECT DATEDIFF(datacor,DataFattura) INTO giornitrascorsi;
SELECT (termine-giornitrascorsi) INTO Scadenza;
RETURN Scadenza;
END$$
DELIMITER ;
```

Funzione 2: Calcolo quantità totale di un determinato articolo venduta nel mese precedente a quello corrente

```
DELIMITER $$
CREATE FUNCTION `Venduto`(CodiceArticolo INTEGER(10))
RETURNS int(11)
BEGIN
DECLARE maxvendite int(11);
SELECT
    sum(Quantita)
INTO maxvendite
FROM
    vendita as v
WHERE
    MONTH(v.DataNota)=MONTH(CURDATE())-1 AND YEAR(v.DataNota)=YEAR(CURDATE()) AND
    CodiceArticolo = v.Articolo;
RETURN maxvendite;
END$$
DELIMITER ;
```

Trigger 1: Aggiornamento disponibilità per una nuova fattura registrata.

```
DROP TRIGGER IF EXISTS dispacquisti;
```

```
DELIMITER $$
CREATE TRIGGER dispacquisti after INSERT ON `acquisto` for each row
begin
UPDATE articolo SET articolo.Disponibilita=articolo.Disponibilita+new.Quantita
WHERE new.Articolo=articolo.CodiceArticolo;
END$$
DELIMITER ;
```

TRIGGER2: Aggiornamento disponibilità per una nuova nota vendita emessa.

```
DROP TRIGGER IF EXISTS dispvendita;
```

```
DELIMITER $$
```

```
CREATE TRIGGER dispvendita after INSERT ON `vendita` for each row
begin
UPDATE articolo SET articolo.Disponibilita=articolo.Disponibilita-new.Quantita
WHERE new.Articolo=articolo.CodiceArticolo;
END$$
DELIMITER ;
```

Trigger 3: Avanzamento di grado tessera ogni 10 acquisti nell'anno corrente

```
DROP TRIGGER IF EXISTS fedelta;
```

```
DELIMITER $$
create trigger fedelta after insert on `notavendita` for each row
begin
declare numvendite int(10);
declare tesserato int(10);
declare tipo char(255);

select count(*) into tesserato from tessera t where new.Cliente=t.Cliente;

if(tesserato=1) then
select TipoTessera into tipo from tessera t where new.Cliente=t.Cliente;

select count(*) into numvendite
from notavendita n
where new.Cliente=n.Cliente AND YEAR(DataNota)=YEAR(CURDATE());

if(numvendite=20) then
update tessera set TipoTessera='GOLD' where new.Cliente=tessera.Cliente;
end if;
if(numvendite=10 and tipo<>'GOLD') then
update tessera set tessera.TipoTessera='SILVER' where new.Cliente=tessera.Cliente;
end if;

end if;
end$$
DELIMITER ;
```