



SAPIENZA  
UNIVERSITÀ DI ROMA

## Basi di Dati, Modulo 2

Sapienza Università di Roma

Facoltà di Ing. dell'Informazione, Informatica e Statistica

Laurea in Informatica

Prof. Toni Mancini

<http://tmancini.di.uniroma1.it>

Progetto 20220222 (P.20220222)

# Amazon

Versione 2022-02-22

# Indice

<b>Indice</b>	<b>1</b>
<b>1 Introduzione</b>	<b>3</b>
<b>2 Specifica dei Requisiti</b>	<b>5</b>
<b>A Analisi Concettuale</b>	<b>9</b>
<b>A.1 Raffinamento dei requisiti</b>	<b>11</b>
A.1.1 Testo . . . . .	11
A.1.2 Soluzione . . . . .	12
<b>A.2 Diagramma ER</b>	<b>17</b>
A.2.1 Testo . . . . .	17
A.2.2 Soluzione . . . . .	18
<b>A.3 Estensione del diagramma ER</b>	<b>21</b>
A.3.1 Testo . . . . .	21
A.3.2 Soluzione . . . . .	22
<b>A.4 Estensione del diagramma ER</b>	<b>29</b>
A.4.1 Testo . . . . .	29
A.4.2 Soluzione . . . . .	30
<b>A.5 Estensione del diagramma ER</b>	<b>31</b>
A.5.1 Testo . . . . .	31
A.5.2 Soluzione . . . . .	32

<b>A.6</b>	<b>Estensione del diagramma ER</b>	<b>33</b>
A.6.1	Testo . . . . .	33
A.6.2	Soluzione . . . . .	34
<b>A.7</b>	<b>Estensione del diagramma ER</b>	<b>35</b>
A.7.1	Testo . . . . .	35
A.7.2	Soluzione . . . . .	36
<b>A.8</b>	<b>Diagramma UML degli use case e specifiche concettuali degli use case</b>	<b>37</b>
A.8.1	Testo . . . . .	37

## 1

---

## Introduzione

Si vuole progettare e realizzare *AmaZZon*, un piattaforma online di e-commerce che vuole offrirsi come *marketplace* comune di offerte di venditori indipendenti.

In particolare, si desidera che il sistema consenta ai negozi registrati nella piattaforma di pubblicare *offerte* di vendita per prodotti di varie tipologie, che possono essere acquistati dagli utenti privati registrati.



## 2

## Specifica dei Requisiti

La piattaforma *AmaZZon* è utilizzata sia da negozi (che pubblicano offerte di loro prodotti) che dai clienti (che acquistano tali prodotti).

Di ogni negozio registrato, il sistema deve mantenere informazioni circa il nome, l'indirizzo della sede legale, almeno un indirizzo email e opzionalmente un numero di telefono per la richiesta di informazioni e assistenza post-vendita.

I negozi registrati usano la piattaforma per pubblicare offerte di (ovvero per mettere in vendita) prodotti di vario genere.

Gli articoli disponibili su *AmaZZon* possono essere offerti da più negozi. Le diverse offerte per uno stesso articolo possono differenziarsi nelle condizioni applicate. In particolare, all'atto dell'inserimento di un articolo in vendita, il negozio deve dichiarare nome e descrizione dell'articolo, marca, numero del modello, categoria, un insieme di tag (stringhe), prezzo e istante di inizio e, eventualmente, di fine dell'offerta di vendita. Si noti *AmaZZon* non è a conoscenza delle informazioni di quanti pezzi di un certo articolo possono essere vendute da un certo negozio. In caso un negozio dovesse terminare la disponibilità di magazzino di un articolo che ha posto in vendita, sarà sua cura eliminare la relativa offerta su *AmaZZon*.

Il seguente esempio potrebbe aiutare a chiarire alcuni dei requisiti precedenti. Il negozio 'New Tech s.r.l.' con sede legale in 'via A. Turing 1, Milano' mette in vendita il nuovo modello di televisore al plasma 'Pony PN123 plus' al prezzo di 1199.00 Eur. L'offerta è valida dal 23 ottobre al 15 novembre 2022 (salvo disponibilità). *AmaZZon* infatti non ha informazioni sulle disponibilità di magazzino del negozio: sarà cura di quest'ultimo eliminare l'offerta dal marketplace in caso di tutto esaurito.

*AmaZZon* permette ai negozi di definire le spese di spedizione di un articolo in base al paese di destinazione e al numero di pezzi acquistati. In particolare, il negozio deve associare ad ogni articolo le spese di spedizione per l'acquisto di un singolo pezzo in base al paese di destinazione.

Il negozio può inoltre definire definire spese di spedizione ridotte in caso di acquisto di

più articoli identici. Ad esempio, il negozio di cui sopra definisce le seguenti spese di spedizione (per un certo paese) per il 'Pony PN123 plus': acquisto di 2-4/5-9/10-99/100+ articoli: Eur 4.99/3.99/2.99/gratis. Le spese di spedizione per 7 pezzi equivarranno quindi a  $7 \times \text{Eur } 3.99$ .

Dato che più negozi possono mettere in vendita lo stesso articolo (a condizioni differenti), *AmaZZon* mantiene una sua vetrina di tutti gli oggetti disponibili, permettendo quindi all'utente di: (i) navigare nella vetrina di tutti gli oggetti in vendita; (ii) consultare, per ognuno di essi, la scheda tecnica con codice identificativo univoco dell'articolo (una stringa secondo un certo standard globale), nome, marca, numero di modello, categoria, tag; (iii) accedere a tutte le offerte riguardanti l'articolo da parte dei diversi negozi (accedendo quindi al prezzo ed ai costi di spedizione).

Come detto, tutti gli articoli in vendita su *AmaZZon* sono presenti in una vetrina unica. Dunque, all'atto della pubblicazione, da parte di un negozio, di un'offerta per un certo articolo, il sistema deve permettere di creare una scheda tecnica per quest'ultimo solo se non esiste già. Altrimenti, la nuova offerta dell'articolo consiste (oltre che al riferimento all'articolo nella vetrina comune) solo nel prezzo e nei costi di spedizione.

Per evitare l'inserimento di un articolo nella vetrina comune con dati errati (ad es., errata categoria), un negozio che, all'atto dell'inserimento di una offerta per un articolo già esistente in vetrina, dovesse notare un errore nei dati della scheda tecnica dell'oggetto, può segnalare l'errore ed eventuali correzioni allo staff di *AmaZZon*.

*AmaZZon* permette agli utenti di acquistare dei *buoni regalo*, i quali possono essere spesi sulla piattaforma per l'acquisto di altri prodotti. Di ogni buono regalo interessa conoscere l'utente che l'ha acquistato, l'importo e il periodo di validità. Al momento del pagamento, l'utente può scegliere se utilizzare o meno ognuno dei suoi buoni regalo attivi. In caso i buoni scelti non dovessero essere sufficienti, il cliente potrà effettuare il pagamento residuo con una delle carte di credito registrate (di cui il sistema deve memorizzare: nome del titolare, numero –una stringa di 16 cifre–, e data di scadenza), se non scaduta.

Gli utenti privati possono registrarsi al sistema. Gli utenti registrati (dei quali interessa il nome utente e l'istante di registrazione) possono visualizzare gli oggetti in vendita e le relative offerte (cosa per la quale non è necessaria la registrazione).

Gli utenti privati registrati possono effettuare acquisti sulla piattaforma, includendo un qualsiasi numero di articoli e di pezzi per articolo. Il sistema deve permettere all'utente di ottenere l'importo totale dell'acquisto, considerando il costo di ogni articolo e le spese di spedizione verso il suo paese (scegliendo ovviamente la tariffa di spedizione più conveniente tra quelle offerte dal venditore applicabili al numero di articoli acquistati).

Inoltre, gli utenti privati registrati possono salvare delle liste dei desideri (*wish list*), contenenti oggetti che desiderano acquistare. Di ogni *wish list* il sistema mantiene (oltre che l'utente registrato proprietario), il nome e l'insieme di oggetti che contiene.

Periodicamente *AmaZZon* effettua una verifica del prezzo più basso disponibile per ogni oggetto presente nelle *wish list* di ogni utente e, se rileva un abbassamento di prezzo, notifica la cosa all'utente.

Le wish list possono essere pubbliche o private. Mentre le wish list private sono visibili solo all'utente che le ha create, le wish list pubbliche sono consultabili anche agli utenti *amici* del proprietario, che quindi possono avere ispirazione circa regali particolarmente graditi.

L'amicizia tra utenti di *AmaZZon* viene instaurata con una richiesta dell'uno e l'accettazione dell'altro. L'amicizia è un concetto simmetrico, quindi in ogni coppia di utenti amici, ogni utente può consultare le wish list pubbliche dell'altro.

Infine, il sistema deve permettere:

- al management di calcolare, dato un periodo di tempo, il paese dal quale sono provenuti più acquisti nel periodo dato
- visualizzare tutti gli oggetti in vendita di una data categoria e contenente un dato insieme di tag (mostrando, per ognuno, il prezzo più basso, comprensivo del costo di spedizione)
- visualizzare gli articoli attualmente più moda, che sono gli articoli il cui volume di vendite è salito percentualmente di più nell'ultimo mese, rispetto alla media mensile dei due mesi precedenti.





# Parte A

## Analisi Concettuale



# A.1

---

## Raffinamento dei requisiti

### A.1.1 Testo

Raffinare la specifica dei requisiti eliminando inconsistenze, omissioni o ridondanze producendo un elenco numerato di requisiti il meno ambiguo possibile.

## A.1.2 Soluzione

1. Requisiti sui negozi:
  - 1.1. nome (stringa)
  - 1.2. indirizzo sede legale
  - 1.3. indirizzi email (uno o più)
  - 1.4. telefono (zero o uno)
  - 1.5. offerte ([Req. 3.](#)).
2. Requisiti sugli articoli:
  - 2.1. Scheda tecnica, in termini di:
    - 2.1.1. codice identificativo (una stringa secondo un certo standard globale)
    - 2.1.2. nome
    - 2.1.3. descrizione
    - 2.1.4. marca
    - 2.1.5. modello
    - 2.1.6. categoria ([Req. 5.](#))
    - 2.1.7. insieme (anche vuoto) di tag ([Req. 6.](#)).
  - 2.2. offerte ([Req. 3.](#)).
3. Requisiti sulle offerte:
  - 3.1. pacchetto in offerta, ovvero:
    - 3.1.1. insieme di articoli ([Req. 2.](#)), ognuno con il numero di pezzi
  - 3.2. negozio ([Req. 1.](#)) che mette in vendita l'offerta
  - 3.3. prezzo di un singolo pacchetto dell'offerta
  - 3.4. periodo di validità offerta (possibilmente aperto a destra), entro il quale può essere acquistata
  - 3.5. spese di spedizione: un insieme di istanze, ognuna che definisce:
    - 3.5.1. paese di destinazione
    - 3.5.2. numero di pacchetti minimo e massimo
    - 3.5.3. importo ( $\geq 0$ )
    - 3.5.4. Non esistono istanze di spese di spedizione relative alla stessa offerta per lo stesso paese e ad un intervallo di numero di pacchetti con valori in comune.
4. Requisiti sugli utenti privati:
  - 4.1. i buoni regalo acquistati, per ognuno:

- 4.1.1. tipologia di buono acquistato ([Req. 7.](#))
    - 4.1.2. importo residuo
    - 4.1.3. istante di scadenza del buono ([Req. 7.3.](#))
  - 4.2. le carte di credito dichiarate, per ognuna:
    - 4.2.1. numero (una stringa di 16 cifre)
    - 4.2.2. data di scadenza
  - 4.3. nome utente
  - 4.4. istante di registrazione
  - 4.5. indirizzo email
  - 4.6. ordini effettuati, per ognuno dei quali:
    - 4.6.1. istante dell'ordine
    - 4.6.2. offerta acquistata ([Req. 3.](#))
    - 4.6.3. numero di pacchetti dell'offerta acquistati ([Req. 3.1.](#))
    - 4.6.4. indirizzo di consegna
    - 4.6.5. modalità di pagamento, ovvero:
      - 4.6.5.1. un sottoinsieme dei buoni (in suo possesso e attivi, [Req. 4.1.](#)), ognuno l'importo da usare ( $\leq$  dell'importo residuo del buono)
      - 4.6.5.2. un sottoinsieme delle carte di credito associate ([Req. 4.1.1.](#)), ognuna l'importo da usare.
  - 4.7. le wish list, di ognuna delle quali interessa:
    - 4.7.1. nome
    - 4.7.2. tipologia (pubblica o privata)
    - 4.7.3. gli articoli ([Req. 2.](#)) contenuti
  - 4.8. gli utenti privati amici (concetto simmetrico)
  - 4.9. le richieste ricevute di amicizia pendenti, di ognuna delle quali interessa l'utente privato richiedente
5. Requisiti sulle categorie degli articoli:
- 5.1. nome
  - 5.2. la eventuale super-categoria
  - 5.3. l'insieme delle categorie formano una foresta
6. Requisiti sui tag degli articoli:
- 6.1. nome
7. Requisiti sulle tipologie di buoni regalo in vendita: per ogni tipologia di buono regalo interessa

- 7.1. periodo nel quale i buoni di quella tipologia possono essere acquistati
  - 7.2. l'importo minimo e massimo dei buoni acquistabili
  - 7.3. la durata di scadenza del buono (entro la quale il buono può essere usato per acquistare su AmaZZon)
  - 7.4. gli acquisti di buoni regalo sono equiparati agli acquisti di articoli, ma sono venduti direttamente da AmaZZon (e non dai negozi affiliati al sistema)
8. Il sistema deve offrire le seguenti funzionalità:
- 8.1. Agli utenti non identificati:
    - 8.1.1. registrarsi al sistema come utenti privati, indicando i propri dati ([Req. 4.](#))
    - 8.1.2. registrarsi al sistema come negozi, indicando i dati richiesti ([Req. 1.](#))
    - 8.1.3. Ottenere i dati della scheda tecnica di un articolo
    - 8.1.4. Dato un articolo 'a', un numero di pezzi 'n', ed un paese 'p', ottenere le offerte attualmente attive per pacchetti contenenti l'articolo 'a', che permettono l'acquisto di un numero di pacchetti che, globalmente, includano 'n' pezzi di 'a', e che ammettano la spedizione verso il paese 'p'.
    - 8.1.5. Data un'offerta, un numero di pacchetti ed un paese, restituire il prezzo finale di un acquisto del numero di pacchetti dato dell'offerta con spedizione verso il paese richiesto.
  - 8.2. Un utente privato (registrato) deve poter:
    - 8.2.1. Acquistare buoni regalo ([Req. 7.](#))
    - 8.2.2. Acquistare una o più offerte, indicando:
      - 8.2.2.1. l'insieme delle offerte da acquistare, ognuna con il numero di pacchetti
      - 8.2.2.2. il paese di destinazione della spedizione
      - 8.2.2.3. un insieme di buoni (in suo possesso e attivi), ognuno con l'importo da usare, che deve essere minore o uguale del credito residuo del buono in quel momento
      - 8.2.2.4. un insieme di carte di credito, tra quelle associate all'utente attive in quel momento
    - 8.2.3. Gestire le proprie wish list ([Req. 4.7.](#))
      - 8.2.3.1. creare una nuova wish list indicandone il nome e la tipologia (pubblica o privata)
      - 8.2.3.2. aggiungere un articolo ad una propria wish list
      - 8.2.3.3. rimuovere un articolo da una propria wish list
      - 8.2.3.4. modificare la tipologia di una propria wish list
      - 8.2.3.5. eliminare una propria wish list

- 8.2.3.6. ricevere notifiche circa l'abbassamento di prezzo degli articoli nelle proprie wish list
- 8.2.4. Gestire i propri rapporti di amicizia, ovvero:
  - 8.2.4.1. Effettuare una richiesta di amicizia verso un altro utente
  - 8.2.4.2. Accettare o rifiutare una richiesta di amicizia pendente da un altro utente
  - 8.2.4.3. Eliminare un rapporto di amicizia verso un altro utente
- 8.2.5. Consultare le wish list pubbliche di un utente amico ([Req. 4.8.](#))
- 8.3. I negozi devono poter:
  - 8.3.1. Inserire un nuovo articolo (non già presente in vetrina) indicando tutti i dati della scheda tecnica ([Req. 2.](#))
  - 8.3.2. Inserire una nuova offerta per un articolo in vetrina, con tutti i dati associati ([Req. 3.](#))
  - 8.3.3. Inviare una richiesta di correzione dei dati della scheda tecnica di un articolo in vetrina ([Req. 2.](#))
- 8.4. Il personale di AmaZZon competente per la gestione della vetrina deve poter:
  - 8.4.1. Aggiornare i dati della scheda tecnica di un articolo in vetrina ([Req. 2.](#))
  - 8.4.2. Evadere una richiesta effettuata come da [Req. 8.3.3.](#) inviando una risposta testuale.
  - 8.4.3. Creare una nuova categoria, indicando nome e, eventualmente, la super-categoria; rinominare una categoria; eliminare una categoria che non ha articoli
  - 8.4.4. Creare, rinominare o un elininare un tag
- 8.5. Il management di AmaZZon deve poter:
  - 8.5.1. calcolare, dato un periodo di tempo, il paese dal quale sono provenuti più acquisti nel periodo dato
- 8.6. Gli utenti privati registrati e Gli utenti non identificati devono poter:
  - 8.6.1. ottenere, dati: - una categoria 'c' - un insieme di tag 'T' - un numero di pezzi 'n' - un paese 'p' tutte le offerte attive di pacchetti che contengono un articolo 'a' di categoria 'c' che ha, tra i suoi tag, tutti quelli in 'T'; le offerte restituire sono quelle che hanno il prezzo minimo per la spedizione nel paese 'p' di un numero di pacchetti che contiene 'n' pezzi dell'articolo 'a'.
  - 8.6.2. restituire l'insieme degli articoli più di moda, ovvero il cui volume di vendite è salito percentualmente di più nell'ultimo mese, rispetto alla media mensile dei due mesi precedenti.





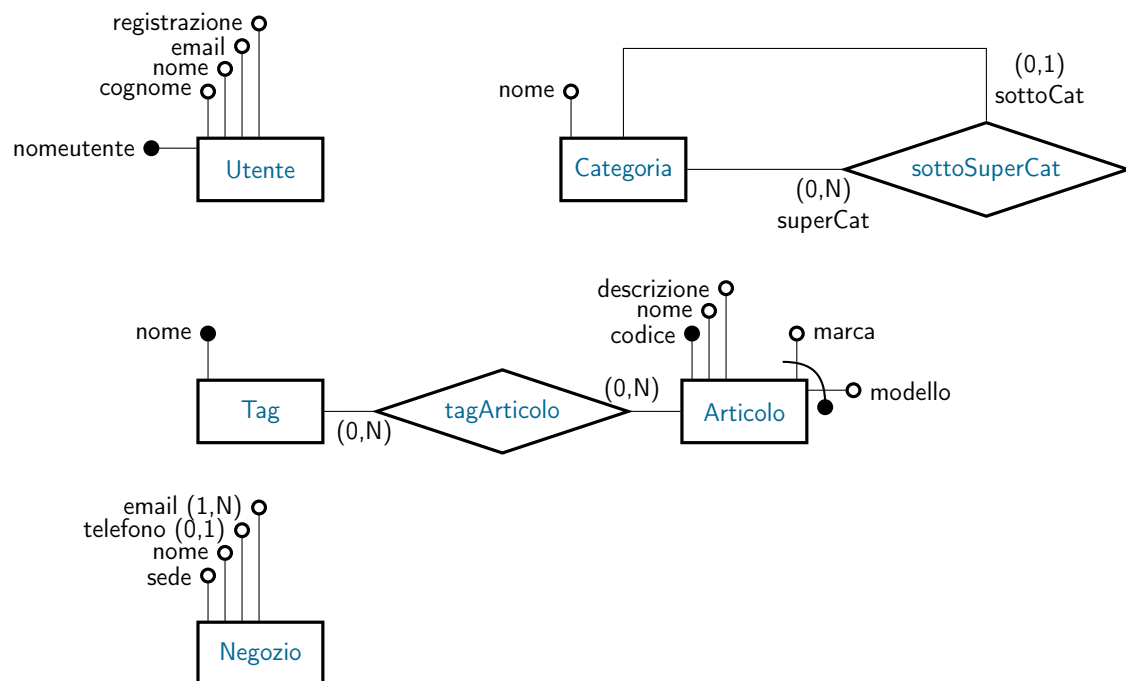
# A.2

## Diagramma ER

### A.2.1 Testo

Proseguire la fase di Analisi Concettuale dei requisiti a partire dall'output della fase di raffinamento effettuata al passo A.1. In particolare, produrre il diagramma ER concettuale per l'applicazione ed il dizionario dei dati per modellare i seguenti requisiti: Req. 1.1–1.4, 2.1, 4.3–4.5, 5, 6.

## A.2.2 Soluzione



## Specifiche dei Dati

### Entità **Negozio**

Ogni istanza di questa entità rappresenta un negozio ([Req. 1.](#))

attributo	dominio	molteplicità	descrizione
sede	Indirizzo		L'indirizzo della sede del negozio
nome	stringa		Il nome del negozio
telefono	stringa	(0,1)	Il numero di telefono del negozio
email	stringa	(1,N)	L'indirizzo email del negozio

### Entità **Utente**

Ogni istanza di questa entità rappresenta un utente ([Req. 4.](#))

attributo	dominio	molteplicità	descrizione
cognome	stringa		Il cognome dell'utente
nome	stringa		Il nome dell'utente
nomeutente	stringa		Il nome utente dell'utente
email	stringa		L'indirizzo email dell'utente
registrazione	data		L'istante di registrazione dell'utente

### Entità **Articolo**

Ogni istanza di questa entità rappresenta un articolo ([Req. 2.](#))

attributo	dominio	molteplicità	descrizione
codice	stringa		Il codice dell'articolo
nome	stringa		Il nome dell'articolo
descrizione	stringa		La descrizione dell'articolo
marca	stringa		La marca dell'articolo
modello	stringa		Il modello dell'articolo

**Entità Tag**

Ogni istanza di questa entità rappresenta un tag ([Req. 6.](#))

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome del tag

**Entità Categoria**

Ogni istanza di questa entità rappresenta una categoria ([Req. 5.](#))

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome della categoria

**Vincoli:**

**[V.Categoria.foresta]** Non esistono due categorie c1 e c2 con lo stesso nome e che hanno la stessa supercategoria oppure non hanno alcuna supercategoria.

**Relationship tagArticolo**

Ogni istanza di questa relationship lega un [Articolo](#) ad un [Tag](#) ([Req. 2.](#), [Req. 6.](#))

Attributi: Nessuno

**Relationship sottoSuperCat**

Ogni istanza di questa relationship lega un [Articolo](#) ad un [Tag](#) ([Req. 2.](#), [Req. 6.](#))

Attributi: Nessuno

**Dominio Indirizzo**

Il dominio è un record composto dai seguenti campi:

- via: stringa
- civico: stringa secondo il pattern: [1-9][0-9]\*[a-z]?

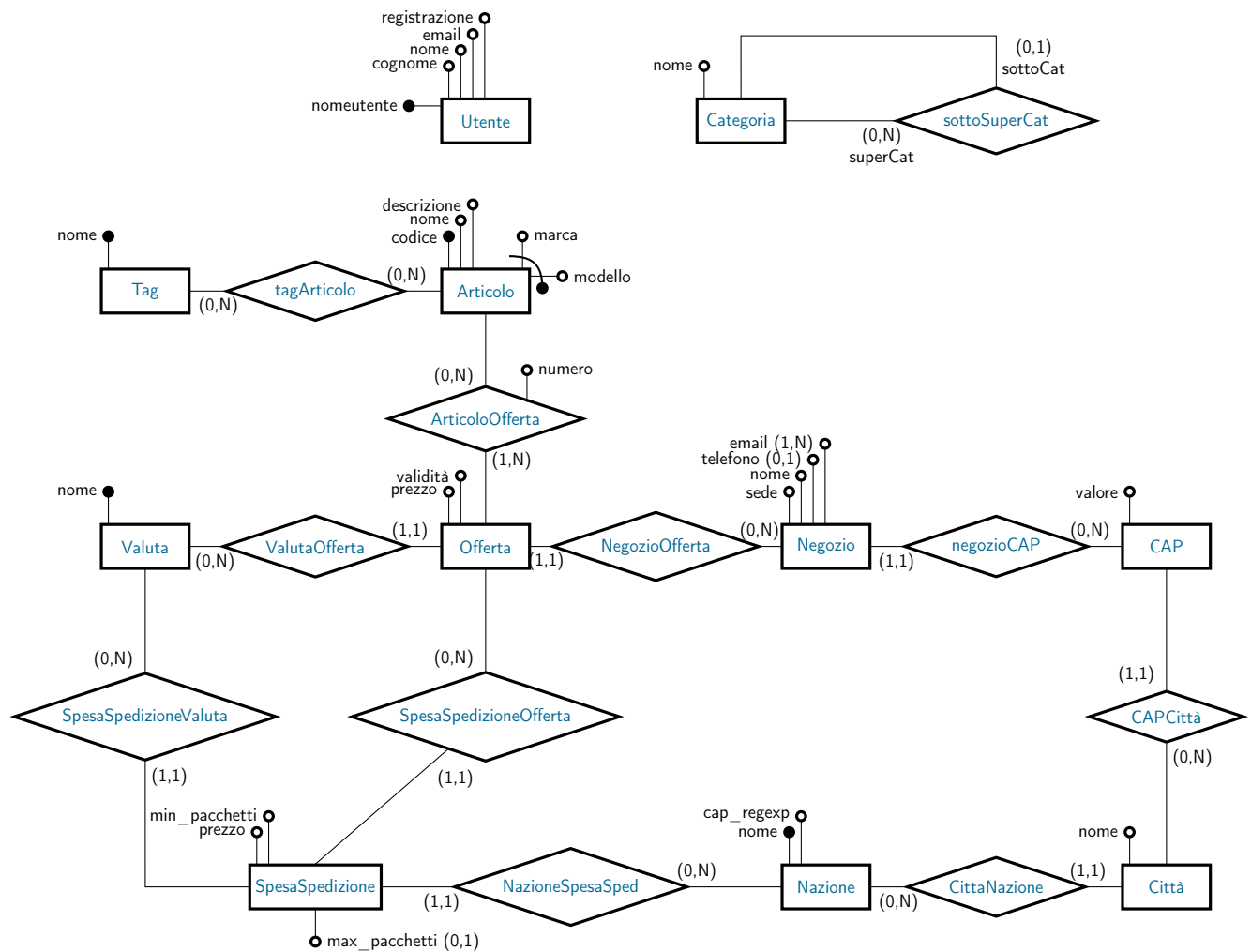
# A.3

## Estensione del diagramma ER

### A.3.1 Testo

Proseguire la fase di Analisi Concettuale dei requisiti estendendo il diagramma ER concettuale per l'applicazione ed il dizionario dei dati per modellare anche i seguenti requisiti: Req. 1.5, 2.2, 3.

## A.3.2 Soluzione



## Specifiche dei Dati

### Entità **Negozio**

Ogni istanza di questa entità rappresenta un negozio ([Req. 1.](#))

attributo	dominio	molteplicità	descrizione
sede	Indirizzo		L'indirizzo della sede del negozio
nome	stringa		Il nome del negozio
telefono	stringa	(0,1)	Il numero di telefono del negozio
email	stringa	(1,N)	L'indirizzo email del negozio

### Entità **Utente**

Ogni istanza di questa entità rappresenta un utente ([Req. 4.](#))

attributo	dominio	molteplicità	descrizione
cognome	stringa		Il cognome dell'utente
nome	stringa		Il nome dell'utente
nomeutente	stringa		Il nome utente dell'utente
email	stringa		L'indirizzo email dell'utente
registrazione	data		L'istante di registrazione dell'utente

### Entità **Articolo**

Ogni istanza di questa entità rappresenta un articolo ([Req. 2.](#))

attributo	dominio	molteplicità	descrizione
codice	stringa		Il codice dell'articolo
nome	stringa		Il nome dell'articolo
descrizione	stringa		La descrizione dell'articolo
marca	stringa		La marca dell'articolo
modello	stringa		Il modello dell'articolo



### Entità Tag

Ogni istanza di questa entità rappresenta un tag ([Req. 6.](#))

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome del tag

### Entità Categoria

Ogni istanza di questa entità rappresenta una categoria ([Req. 5.](#))

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome della categoria

#### Vincoli:

**[V.Categoria.foresta]** Non esistono due categorie c1 e c2 con lo stesso nome e che hanno la stessa supercategoria oppure non hanno alcuna supercategoria.

### Entità CAP

Ogni istanza di questa entità rappresenta un CAP ([Req. 1.2.](#))

attributo	dominio	molteplicità	descrizione
valore	stringa		Il CAP della sede del negozio

#### Vincoli:

**[V.CAP.no.duplicati.stessa.nazione]** Non esistono due istanze di [CAP](#) con lo stesso valore ed associate a [Città](#) della stessa [Nazione](#).

### Entità Città

Ogni istanza di questa entità rappresenta una città ([Req. 1.2.](#))

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome della città sede del negozio

**Entità [Nazione](#)**

Ogni istanza di questa entità rappresenta una nazione ([Req. 1.2.](#))

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome della nazione della sede del negozio
cap_regexp	stringa		Il CAP della nazione della sede del negozio

**Entità [SpesaSpedizione](#)**

Ogni istanza di questa entità rappresenta una spesa di spedizione ([Req. 3.5.](#))

attributo	dominio	molteplicità	descrizione
prezzo	stringa		Il prezzo della spedizione
min_pacchetti	stringa		Il numero minimo di pacchetti della spedizione
max_pacchetti	stringa	(0,1)	Il numero massimo di pacchetti della spedizione

**Vincoli:**

**[V.SpesaSpedizione.intervallo.pacchetti.legale]** Per ogni istanza di [SpesaSpedizione](#), il valore dell'attributo '[max](#)' (se definito) deve essere  $\geq$  del valore dell'attributo '[min](#)'.

**[V.SpesaSpedizione.no.conflitti]** Non esistono due istanze di [SpesaSpedizione](#) relative alla stessa [Offerta](#) per la stessa [Nazione](#) e ad un intervallo di numero di pacchetti con valori in comune.

**Entità [Offerta](#)**

Ogni istanza di questa entità rappresenta un'offerta ([Req. 3.](#))

attributo	dominio	molteplicità	descrizione
prezzo	reale $\geq 0$		Il prezzo dell'offerta
validità	Periodo		Il periodo di validità dell'offerta

**Entità Valuta**

Ogni istanza di questa entità rappresenta una valuta (Req. 3.5.)

attributo	dominio	molteplicità	descrizione
nome	stringa		La valuta utilizzata per le spese di spedizione

**Relationship tagArticolo**

Ogni istanza di questa relationship lega un Articolo ad un Tag (Req. 2., Req. 6.)

Attributi: Nessuno

**Relationship sottoSuperCat**

Ogni istanza di questa relationship lega un Articolo ad un Tag (Req. 2., Req. 6.)

Attributi: Nessuno

**Relationship negozioCAP**

Ogni istanza di questa relationship lega un Negozio ad un CAP (Req. 1., Req. 1.2.)

Attributi: Nessuno

**Relationship CAPCittà**

Ogni istanza di questa relationship lega un CAP ad una Città (Req. 1.2.)

Attributi: Nessuno

**Relationship CittàNazione**

Ogni istanza di questa relationship lega una Città ad una Nazione (Req. 1.2.)

Attributi: Nessuno

**Relationship NazioneSpesaSped**

Ogni istanza di questa relationship lega una Nazione ad una SpesaSpedizione (Req. 1.2., Req. 3.5.)

Attributi: Nessuno

**Relationship SpesaSpedizioneOfferta**

Ogni istanza di questa relationship lega una SpesaSpedizione ad una Offerta (Req. 3.5., Req. 3.)

Attributi: Nessuno

**Relationship SpesaSpedizioneValuta**

Ogni istanza di questa relationship lega una SpesaSpedizione ad una Valuta (Req. 3.5.)

Attributi: Nessuno

### Relationship **NegozioOfferta**

Ogni istanza di questa relationship lega un **Negozio** ad una **Offerta** (Req. 1., Req. 3.)

Attributi: Nessuno

### Relationship **ArticoloOfferta**

Ogni istanza di questa relationship lega un **Articolo** ad una **Offerta** (Req. 2., Req. 3.)

attributo	dominio	molteplicità	descrizione
numero	intero > 0		Il numero di articoli presenti nell'offerta

### Relationship **ValutaOfferta**

Ogni istanza di questa relationship lega una **Valuta** ad una **Offerta** (Req. 3.5., Req. 3.)

Attributi: Nessuno

### Dominio **Indirizzo**

Il dominio è un record composto dai seguenti campi:

- via: stringa
- civico: stringa secondo il pattern: `[1-9][0-9]*[a-z]?`
- cap: stringa

### Dominio **Periodo**

Il dominio è un record composto dai seguenti campi:

- da: dataora
- a: dataora (0,1)



# A.4

## Estensione del diagramma ER

### A.4.1 Testo

Proseguire la fase di Analisi Concettuale dei requisiti estendendo il diagramma ER concettuale per l'applicazione ed il dizionario dei dati per modellare anche i seguenti requisiti: Req. 4.1, 4.2, 7.

## A.4.2 Soluzione

# A.5

## Estensione del diagramma ER

### A.5.1 Testo

Proseguire la fase di Analisi Concettuale dei requisiti estendendo il diagramma ER concettuale per l'applicazione ed il dizionario dei dati per modellare anche i seguenti requisiti: Req. 4.6, 4.7.



## A.5.2 Soluzione

# A.6

## Estensione del diagramma ER

### A.6.1 Testo

Proseguire la fase di Analisi Concettuale dei requisiti estendendo il diagramma ER concettuale per l'applicazione ed il dizionario dei dati per modellare anche i seguenti requisiti: Req. 4.8, 4.9.

## A.6.2 Soluzione

# A.7

## Estensione del diagramma ER

### A.7.1 Testo

Proseguire la fase di Analisi Concettuale dei requisiti estendendo il diagramma ER concettuale per l'applicazione ed il dizionario dei dati per formalizzare i vincoli esterni isolati in precedenza usando la logica del primo ordine.

## A.7.2 Soluzione

# A.8

## Diagramma UML degli use case e specifiche concettuali degli use case

### A.8.1 Testo

Proseguire la fase di Analisi Concettuale dei requisiti definendo il diagramma UML degli use-case dell'applicazione e le relative specifiche concettuali.