



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, Prof. Federico Mari

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

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

Progetto 20060411 (P.20060411)

# Travel to the Moon

Versione 2016-05-14

# 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>7</b>
<b>A.1 Raffinamento dei Requisiti</b>	<b>9</b>
A.1.1 Testo . . . . .	9
A.1.2 Soluzione . . . . .	10
<b>A.2 Diagramma ER</b>	<b>13</b>
A.2.1 Testo . . . . .	13
A.2.2 Soluzione . . . . .	14
<b>A.3 Estensione del Diagramma ER</b>	<b>19</b>
A.3.1 Testo . . . . .	19
A.3.2 Soluzione . . . . .	20
<b>A.4 Estensione del Diagramma ER</b>	<b>27</b>
A.4.1 Testo . . . . .	27
A.4.2 Soluzione . . . . .	28
<b>A.5 Identificazione delle Entità e Definizione Informale di Vincoli Esterni</b>	<b>35</b>
A.5.1 Testo . . . . .	35
A.5.2 Soluzione . . . . .	36

<b>A.6</b>	<b>Diagramma UML degli Use-Case</b>	<b>45</b>
A.6.1	Testo . . . . .	45
A.6.2	Soluzione . . . . .	46
<b>A.7</b>	<b>Specifiche Concettuali Informali degli Use-Case</b>	<b>47</b>
A.7.1	Testo . . . . .	47
A.7.2	Soluzione . . . . .	48

Questo materiale è concesso a  
Filippo Borsi  
esclusivamente per uso personale.  
È vietata la sua diffusione in qualsiasi forma.

# 1

## Introduzione

Si vuole sviluppare un sistema per la gestione di una agenzia turistica che organizza crociere.

Questo materiale è concesso a  
Filippo Boni  
esclusivamente per uso personale.  
È vietata la sua diffusione in qualsiasi forma.



Questo materiale è concesso a  
Filippo Borsi  
esclusivamente per uso personale.  
È vietata la sua diffusione in qualsiasi forma.

## 2

## Specifica dei Requisiti

I dati di interesse per il sistema sono le crociere offerte dall'agenzia con le relative prenotazioni e le destinazioni in catalogo.

Il sistema deve essere in grado di rappresentare le crociere offerte dall'agenzia, con codice, date di inizio e fine, e la nave utilizzata. Delle navi, che hanno un nome (ad es. "LoveBoat"), interessa il grado di comfort, espresso in un numero di stelle che può variare da 3 a 5, e il numero massimo di passeggeri che possono ospitare.

Ciascuna crociera consta di un itinerario caratterizzato da un nome (ad es. "Panorami d'Oriente") il quale prevede una sequenza –ordinata– di destinazioni. Di queste interessa il nome e il continente in cui si trovano. Gli itinerari fissano, oltre che l'ordine delle destinazioni da visitare, anche la relativa data ed ora di arrivo e di partenza. Dato che, in generale, un itinerario può essere previsto da più di una crociera, le date di arrivo e partenza relative ad una destinazione vengono espresse come differenze rispetto la data di inizio della crociera stessa (ad es., l'itinerario "Panorami d'Oriente" prevede di raggiungere la destinazione  $x$  alle 16:00 del quinto giorno di crociera, e di ripartire alle 12:00 del giorno successivo, il sesto).

Inoltre, le destinazioni sono caratterizzate da un insieme di posti da vedere durante eventuali escursioni organizzate. Questi ultimi sono caratterizzati dal nome, dalla descrizione, e dalla fascia oraria consigliata per le visite. Il sistema deve permettere di risalire ai posti da vedere in ogni singola destinazione.

L'agenzia classifica le crociere in *crociere di luna di miele* e *crociere per famiglia* (di queste ultime interessa conoscere se sono adatte o meno ai bambini), e le destinazioni in *romantiche* e *divertenti*. Si noti che possono esistere destinazioni che sono sia romantiche che divertenti. Per venire incontro alle nuove tendenze delle giovani coppie, le crociere di luna di miele vengono ulteriormente classificate in *tradizionali* e *alternative*: sono definite tradizionali quelle che prevedono un numero di destinazioni romantiche maggiore o uguale al numero di destinazioni divertenti, alternative le altre.

Infine, il sistema deve anche permettere di gestire le prenotazioni di crociere effettuate



dai clienti. In particolare, dei clienti interessa nome, cognome, età ed indirizzo, mentre delle prenotazioni interessa l'istante di prenotazione, la crociera ed il numero di posti prenotati.

Le funzionalità richieste al sistema sono le seguenti:

1. Dato un cliente che desidera prenotare un certo numero di posti per una crociera  $c$ , il personale dell'Ufficio Prenotazioni deve poter effettuare la relativa prenotazione. La richiesta di prenotazione deve essere rifiutata nel caso il numero di posti disponibili, all'istante corrente, per la crociera  $c$  non sia sufficiente.
2. Dato un insieme di clienti, l'Ufficio Marketing deve poter calcolare l'età media di quelli che hanno prenotato almeno una crociera che prevede una destinazione esotica (ovvero che si trova in un continente diverso dall'Europa).
3. Dato un insieme di destinazioni, l'Ufficio Marketing deve poter calcolare la percentuale di quelle *gettonate*. Una destinazione si dice gettonata se è stata raggiunta da almeno dieci crociere di luna di miele, oppure da almeno quindici crociere per famiglie nel corso degli ultimi due anni.

Questo materiale è messo a disposizione di  
Filippo Bonaccorsi  
esclusivamente per uso personale.  
È vietata la sua diffusione in qualsiasi forma.

# Parte A

## Analisi Concettuale

Questo materiale è concesso a  
Filippo Borsi  
esclusivamente per uso personale.  
È vietata la sua diffusione in qualsiasi forma.



Questo materiale è concesso a  
Filippo Borsi  
esclusivamente per uso personale.  
È vietata la sua diffusione in qualsiasi forma.

# A.1

## Raffinamento dei Requisiti

### A.1.1 Testo

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

Questo materiale è concesso a  
Filippo Borsi  
esclusivamente per uso personale.  
È vietata la sua diffusione in qualsiasi forma.

## A.1.2 Soluzione

1. Delle crociere offerte interessa:
  - 1.1. codice
  - 1.2. data di inizio
  - 1.3. data di fine
  - 1.4. nave utilizzata (Req. 2.)
  - 1.5. itinerario (Req. 3.).
  - 1.6. Le crociere si classificano in *crociere di luna di miele* e *crociere per famiglia*:
    - 1.6.1. Le crociere di luna di miele si classificano in:
      - 1.6.1.1. tradizionali, se il loro itinerario (Req. 1.5.) prevede un numero di destinazioni romantiche (Req. 4.4.) maggiore o uguale al numero di destinazioni divertenti (Req. 4.5.)
      - 1.6.1.2. alternative, se il loro itinerario (Req. 1.5.) prevede un numero di destinazioni romantiche (Req. 4.4.) strettamente minore al numero di destinazioni divertenti (Req. 4.5.).
    - 1.6.2. Delle crociere per famiglia interessa sapere:
      - 1.6.2.1. se sono adatte ai bambini.
2. Delle navi utilizzate per le crociere interessa:
  - 2.1. nome
  - 2.2. grado di comfort (intero tra 3 e 5)
  - 2.3. numero massimo di passeggeri.
3. Degli itinerari delle crociere interessa:
  - 3.1. nome
  - 3.2. sequenza destinazioni. Per ogni destinazione interessa:
    - 3.2.1. numero d'ordine nell'itinerario
    - 3.2.2. data di arrivo, in termini di numero di giorni dall'inizio della crociera
    - 3.2.3. ora di arrivo
    - 3.2.4. data di partenza, in termini di numero di giorni dall'inizio della crociera
    - 3.2.5. ora di partenza.
4. Delle destinazioni interessa:
  - 4.1. nome
  - 4.2. continente
  - 4.3. posti da vedere

- 4.4. se sono *romantiche*
- 4.5. se sono *divertenti*
- 4.6. se sono *gettonate*, ovvero sono state raggiunte da almeno dieci crociere di luna di miele oppure da almeno quindici crociere per famiglie nel corso degli ultimi due anni
- 4.7. se sono *esotiche*, ovvero si trovano in un continente diverso dall'Europa.
5. Dei posti da vedere interessa:
  - 5.1. nome
  - 5.2. descrizione
  - 5.3. fascia oraria consigliata per le visite.
6. Dei clienti interessa:
  - 6.1. nome
  - 6.2. cognome
  - 6.3. data di nascita
  - 6.4. indirizzo.
7. Di ogni prenotazione effettuata da un cliente per una crociera interessa:
  - 7.1. cliente che ha effettuato la prenotazione
  - 7.2. istante di prenotazione
  - 7.3. crociera prenotata
  - 7.4. numero di posti prenotati.
8. Le funzionalità richieste al sistema sono:
  - 8.1. Dato un cliente che desidera prenotare un certo numero di posti per una crociera  $c$ , il personale dell'Ufficio Prenotazioni deve poter effettuare la relativa prenotazione. La richiesta di prenotazione deve essere rifiutata nel caso il numero di posti disponibili, all'istante corrente, per la crociera  $c$  non sia sufficiente.
  - 8.2. Dato un insieme di clienti, l'Ufficio Marketing deve poter calcolare l'età media di quelli che hanno prenotato almeno una crociera che prevede una destinazione esotica (Req. 4.7.).
  - 8.3. Dato un insieme di destinazioni, l'Ufficio Marketing deve poter calcolare la percentuale di quelle gettonate (Req. 4.6.).



Questo materiale è concesso a  
Filippo Borsi  
esclusivamente per uso personale.  
È vietata la sua diffusione in qualsiasi forma.

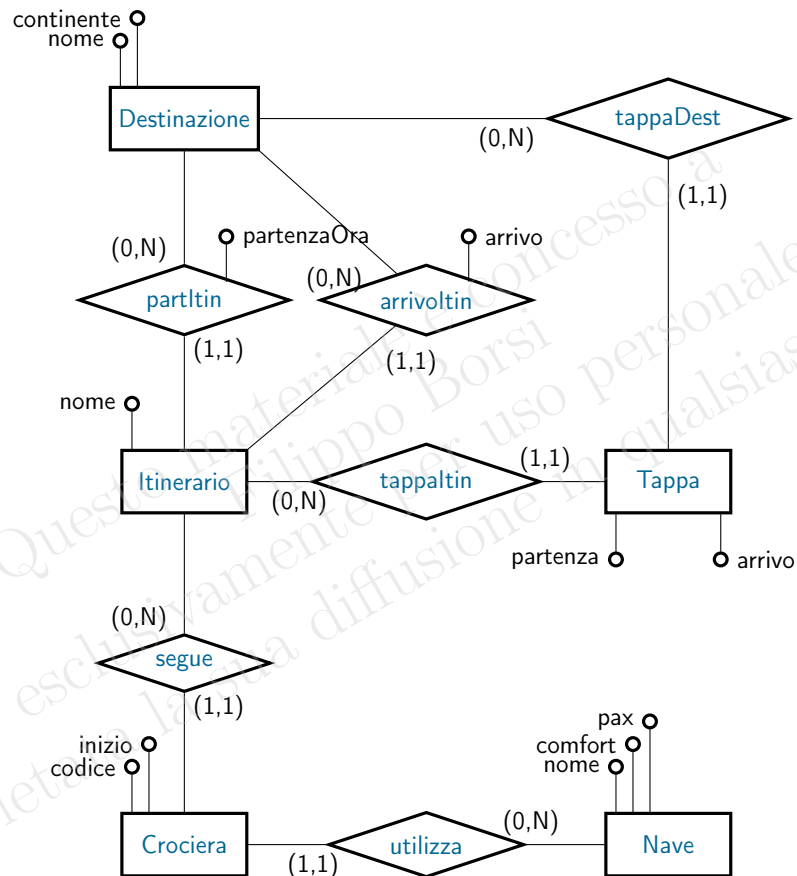
# A.2

## 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., Req. 1.2., Req. 1.3., Req. 1.4., Req. 1.5., Req. 2., Req. 3., Req. 4.1., Req. 4.2.

## A.2.2 Soluzione



## Specifiche dei Dati

### Entità **Crociera**

Ogni istanza di questa entità rappresenta una crociera (Req. 1.)

attributo	dominio	molteplicità	descrizione
codice	stringa		Il codice della crociera
inizio	data		La data di inizio della crociera

### Entità **Nave**

Ogni istanza di questa entità rappresenta una nave (Req. 2.)

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome della nave
comfort	[3,5]		Il grado di comfort della nave
pax	intero > 0		Il numero massimo di passeggeri per la nave

### Entità **Itinerario**

Ogni istanza di questa entità rappresenta un'itinerario (Req. 3.)

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome dell'itinerario

### Entità **Destinazione**

Ogni istanza di questa entità rappresenta una destinazione toccata da **Itinerari** (Req. 4.)

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome della destinazione
continente	{E, AS, AMN, AMS, AF, O}		Il continente della destinazione (nell'ordine: Europa, Asia, America del Nord, America del Sud, Africa, Oceania)



**Entità Tappa**

Ogni istanza di questa entità rappresenta una **Tappa** toccata da un **Itinerario** (Req. 3.2.), escluse quelle di partenza e arrivo dell'**Itinerario**

attributo	dominio	molteplicità	descrizione
arrivo	<b>DeltaDataOra</b>		La data (relativa alla data di partenza della crociera) e l'ora di arrivo alla <b>Destinazione</b>
partenza	<b>DeltaDataOra</b>		La data (relativa alla data di partenza della crociera) e l'ora di ripartenza dalla <b>Destinazione</b>

**Relationship utilizza**

Ogni istanza di questa relationship lega una **Crociera** alla **Nave** utilizzata (Req. 1.4.)  
Attributi: Nessuno

**Relationship segue**

Ogni istanza di questa relationship lega una **Crociera** al suo **Itinerario** (Req. 1.5.)  
Attributi: Nessuno

**Relationship tappaItin**

Ogni istanza di questa relationship lega un **Itinerario** ad una sua **Tappa** (Req. 3.2.)  
Attributi: Nessuno

**Relationship tappaDest**

Ogni istanza di questa relationship lega una **Tappa** di un **Itinerario** alla relativa **Destinazione** (Req. 3.2.)  
Attributi: Nessuno

**Relationship partItin**

Ogni istanza di questa relationship lega un **Itinerario** alla relativa **Destinazione** di partenza (Req. 3.2.)

attributo	dominio	molteplicità	descrizione
partenzaOra	ora		L'ora di partenza della <b>Crociera</b>

**Relationship [arrivoltin](#)**

Ogni istanza di questa relationship lega un [Itinerario](#) alla relativa [Destinazione](#) di arrivo (Req. [3.2.](#))

attributo	dominio	molteplicità	descrizione
arrivo	<a href="#">DeltaDataOra</a>		La data (in termini di giorni dalla data di partenza dell'itinerario) e l'ora di arrivo della <a href="#">Crociera</a> alla <a href="#">Tappa</a>

**Dominio [DeltaDataOra](#)**

Il dominio è un record composto dai seguenti campi:

- giorni: intero  $> 0$
- ora: ora

Questo materiale è concesso a  
Filippo Borsi  
esclusivamente per uso personale.  
È vietata la sua diffusione in qualsiasi forma.



Questo materiale è concesso a  
Filippo Borsi  
esclusivamente per uso personale.  
È vietata la sua diffusione in qualsiasi forma.

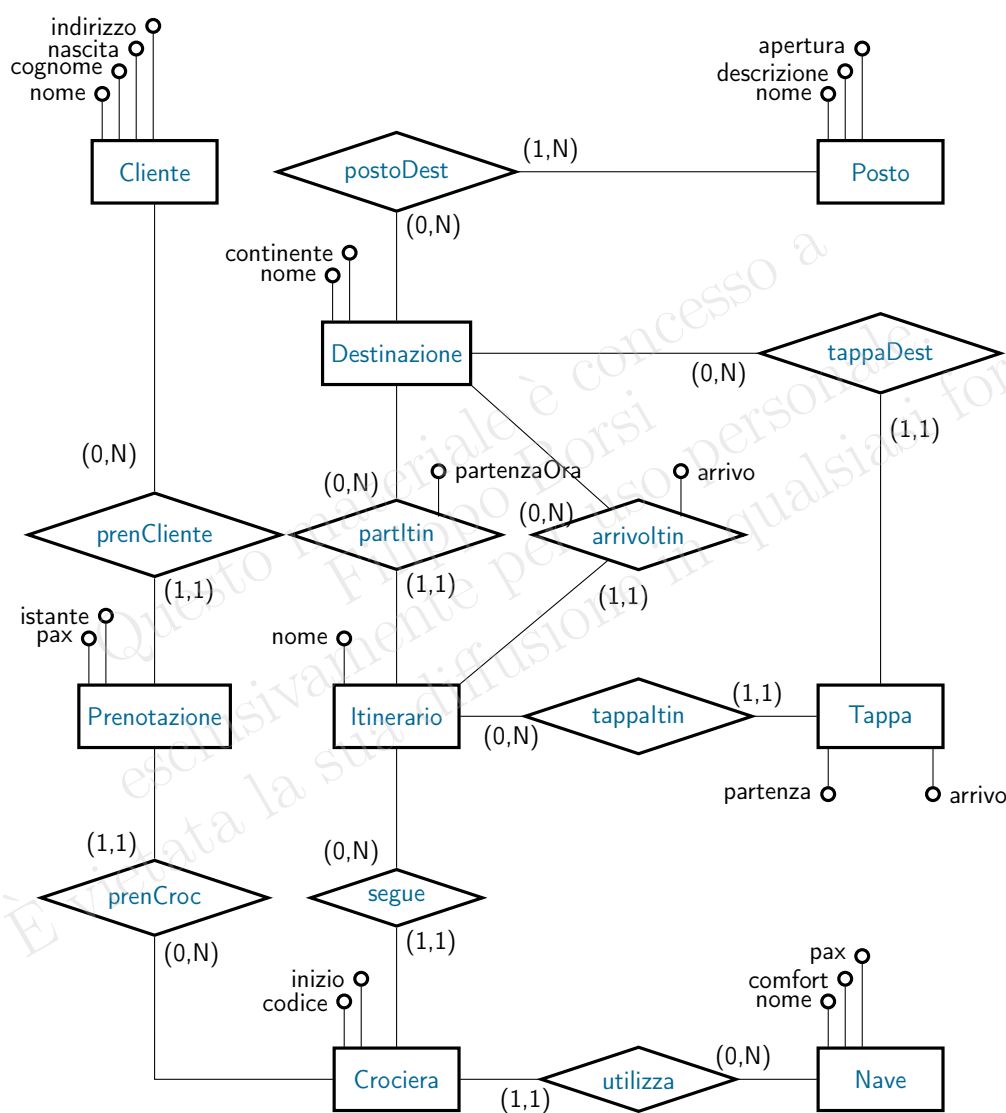
# A.3

## 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. 4.3., Req. 5., Req. 6., Req. 7.

## A.3.2 Soluzione



## Specifiche dei Dati

### Entità **Crociera**

Ogni istanza di questa entità rappresenta una crociera (Req. 1.)

attributo	dominio	molteplicità	descrizione
codice	stringa		Il codice della crociera
inizio	data		La data di inizio della crociera

### Entità **Nave**

Ogni istanza di questa entità rappresenta una nave (Req. 2.)

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome della nave
comfort	[3,5]		Il grado di comfort della nave
pax	intero > 0		Il numero massimo di passeggeri per la nave

### Entità **Itinerario**

Ogni istanza di questa entità rappresenta un'itinerario (Req. 3.)

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome dell'itinerario

### Entità **Destinazione**

Ogni istanza di questa entità rappresenta una destinazione toccata da **Itinerari** (Req. 4.)

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome della destinazione
continente	{E, AS, AMN, AMS, AF, O}		Il continente della destinazione (nell'ordine: Europa, Asia, America del Nord, America del Sud, Africa, Oceania)

### Entità **Tappa**

Ogni istanza di questa entità rappresenta una **Tappa** toccata da un **Itinerario** (Req. 3.2.), escluse quelle di partenza e arrivo dell'**Itinerario**

attributo	dominio	molteplicità	descrizione
arrivo	<b>DeltaDataOra</b>		La data (relativa alla data di partenza della crociera) e l'ora di arrivo alla <b>Destinazione</b>
partenza	<b>DeltaDataOra</b>		La data (relativa alla data di partenza della crociera) e l'ora di ripartenza dalla <b>Destinazione</b>

### Entità **Posto**

Ogni istanza di questa entità rappresenta un posto da vedere (Req. 5.)

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome del posto da vedere
descrizione	stringa		La descrizione del posto da vedere
apertura	<b>FasciaOraria</b>		La fascia oraria di apertura del posto da vedere

### Entità **Cliente**

Ogni istanza di questa entità rappresenta un cliente (Req. 6.)

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome del cliente
cognome	stringa		Il cognome del cliente
nascita	data		La data di nascita del cliente
indirizzo	<b>Indirizzo</b>		L'indirizzo del cliente

**Entità Prenotazione**

Ogni istanza di questa entità rappresenta una prenotazione di una **Crociera** da parte di un **Cliente** (Req. 7.)

attributo	dominio	molteplicità	descrizione
pax	intero > 0		Il numero di posti della prenotazione
istante	dataora		L'istante nel quale è stata effettuata la prenotazione

**Relationship utilizza**

Ogni istanza di questa relationship lega una **Crociera** alla **Nave** utilizzata (Req. 1.4.)

Attributi: Nessuno

**Relationship segue**

Ogni istanza di questa relationship lega una **Crociera** al suo **Itinerario** (Req. 1.5.)

Attributi: Nessuno

**Relationship tappaItin**

Ogni istanza di questa relationship lega un **Itinerario** ad una sua **Tappa** (Req. 3.2.)

Attributi: Nessuno

**Relationship tappaDest**

Ogni istanza di questa relationship lega una **Tappa** di un **Itinerario** alla relativa **Destinazione** (Req. 3.2.)

Attributi: Nessuno

**Relationship partItin**

Ogni istanza di questa relationship lega un **Itinerario** alla relativa **Destinazione** di partenza (Req. 3.2.)

attributo	dominio	molteplicità	descrizione
partenzaOra	ora		L'ora di partenza della <b>Crociera</b>



**Relationship [arrivoltin](#)**

Ogni istanza di questa relationship lega un [Itinerario](#) alla relativa [Destinazione](#) di arrivo (Req. 3.2.)

attributo	dominio	molteplicità	descrizione
arrivo	<a href="#">DeltaDataOra</a>		La data (in termini di giorni dalla data di partenza dell'itinerario) e l'ora di arrivo della <a href="#">Crociera</a> alla <a href="#">Tappa</a>

**Relationship [postoDest](#)**

Ogni istanza di questa relationship lega una [Destinazione](#) ad un [Posto da vedere](#) (Req. 4.3.)

Attributi: Nessuno

**Relationship [prenCliente](#)**

Ogni istanza di questa relationship lega una [Prenotazione](#) al relativo [Cliente](#) (Req. 7.1.)

Attributi: Nessuno

**Relationship [prenCroc](#)**

Ogni istanza di questa relationship lega una [Prenotazione](#) alla relativa [Crociera](#) (Req. 7.3.)

Attributi: Nessuno

**Dominio [DeltaDataOra](#)**

Il dominio è un record composto dai seguenti campi:

- giorni: intero  $> 0$
- ora: ora

**Dominio [FasciaOraria](#)**

Il dominio è un record composto dai seguenti campi:

- da: ora
- a: ora

**Dominio [Indirizzo](#)**

Il dominio è un record composto dai seguenti campi:



- via: stringa
- civico: intero  $> 0$  (0,1)
- CAP: stringa di 5 cifre numeriche
- città: stringa
- nazione: stringa

Questo materiale è concesso a  
Filippo Borsi  
esclusivamente per uso personale.  
È vietata la sua diffusione in qualsiasi forma.



Questo materiale è concesso a  
Filippo Borsi  
esclusivamente per uso personale.  
È vietata la sua diffusione in qualsiasi forma.

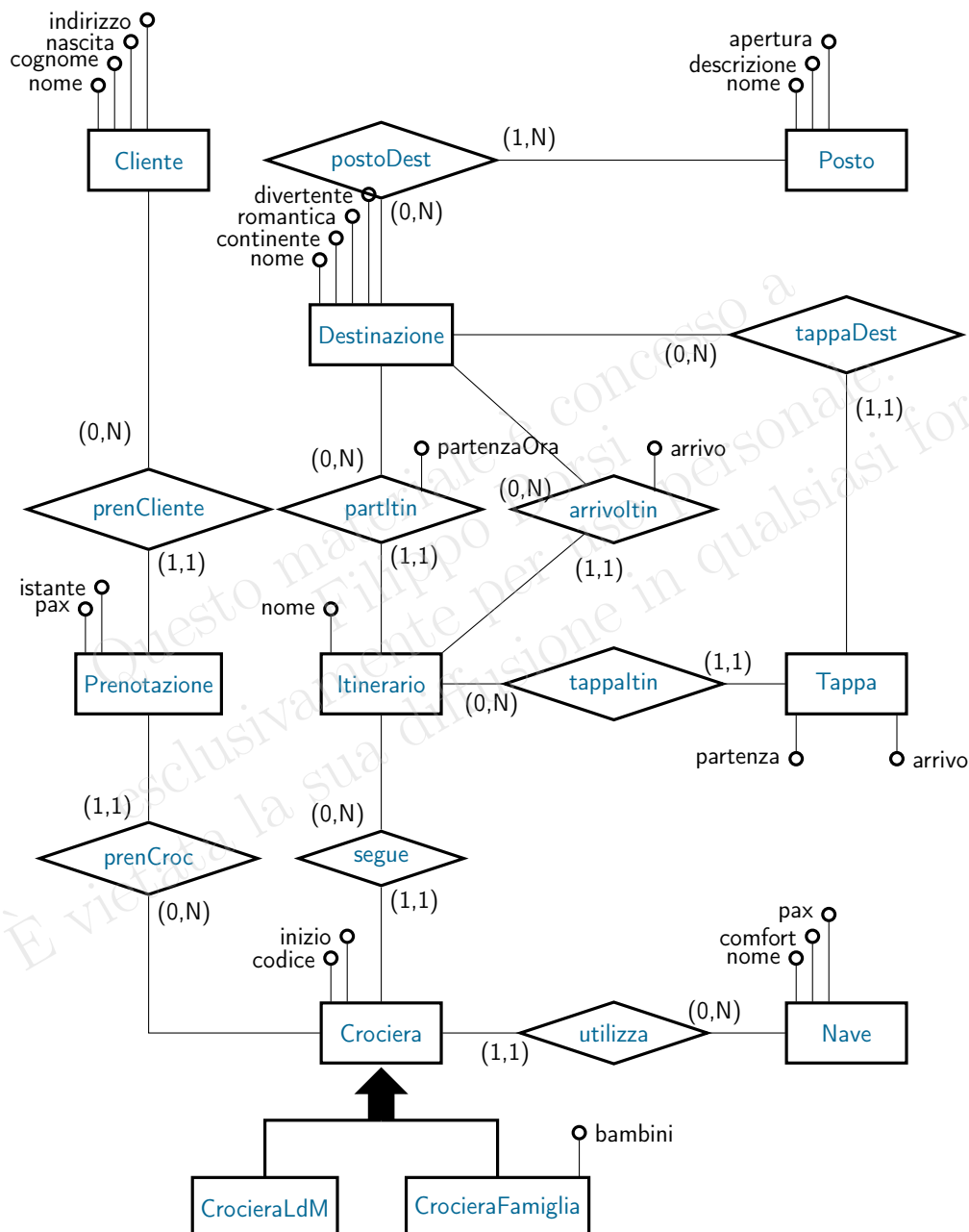
# A.4

## 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. 1.6. (con l'esclusione di Req. 1.6.1.), Req. 4.4., Req. 4.5.

## A.4.2 Soluzione



## Specifiche dei Dati

### Entità **Crociera**

Ogni istanza di questa entità rappresenta una crociera (Req. 1.)

attributo	dominio	molteplicità	descrizione
codice	stringa		Il codice della crociera
inizio	data		La data di inizio della crociera

### Entità **Nave**

Ogni istanza di questa entità rappresenta una nave (Req. 2.)

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome della nave
comfort	[3,5]		Il grado di comfort della nave
pax	intero > 0		Il numero massimo di passeggeri per la nave

### Entità **Itinerario**

Ogni istanza di questa entità rappresenta un'itinerario (Req. 3.)

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome dell'itinerario

### Entità **Destinazione**

Ogni istanza di questa entità rappresenta una destinazione toccata da **Itinerari** (Req. 4.)

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome della destinazione
continente	{E, AS, AMN, AMS, AF, O}		Il continente della destinazione (nell'ordine: Europa, Asia, America del Nord, America del Sud, Africa, Oceania)
romantica	booleano		true se la destinazione è classificata come romantica, false altrimenti
divertente	booleano		true se la destinazione è classificata come divertente, false altrimenti

### Entità **Tappa**

Ogni istanza di questa entità rappresenta una **Tappa** toccata da un **Itinerario** (Req. 3.2.), escluse quelle di partenza e arrivo dell'**Itinerario**

attributo	dominio	molteplicità	descrizione
arrivo	<b>DeltaDataOra</b>		La data (relativa alla data di partenza della crociera) e l'ora di arrivo alla <b>Destinazione</b>
partenza	<b>DeltaDataOra</b>		La data (relativa alla data di partenza della crociera) e l'ora di ripartenza dalla <b>Destinazione</b>

### Entità **Posto**

Ogni istanza di questa entità rappresenta un posto da vedere (Req. 5.)

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome del posto da vedere
descrizione	stringa		La descrizione del posto da vedere
apertura	<b>FasciaOraria</b>		La fascia oraria di apertura del posto da vedere

### Entità **Cliente**

Ogni istanza di questa entità rappresenta un cliente (Req. 6.)

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome del cliente
cognome	stringa		Il cognome del cliente
nascita	data		La data di nascita del cliente
indirizzo	<b>Indirizzo</b>		L'indirizzo del cliente

### Entità **Prenotazione**

Ogni istanza di questa entità rappresenta una prenotazione di una **Crociera** da parte di un **Cliente** (Req. 7.)

attributo	dominio	molteplicità	descrizione
pax	intero > 0		Il numero di posti della prenotazione
istante	dataora		L'istante nel quale è stata effettuata la prenotazione

### Entità **CrocieraLdM**

Ogni istanza di questa entità rappresenta una crociera di luna di miele (Req. 1.6.1.)

Attributi: Nessuno

### Entità **CrocieraFamiglia**

Ogni istanza di questa entità rappresenta una crociera per famiglia (Req. 1.6.2.)

attributo	dominio	molteplicità	descrizione
bambini	booleano		true se la crociera è adatta ai bambini, false altrimenti

### Relationship **utilizza**

Ogni istanza di questa relationship lega una **Crociera** alla **Nave** utilizzata (Req. 1.4.)

Attributi: Nessuno

### Relationship **segue**

Ogni istanza di questa relationship lega una **Crociera** al suo **Itinerario** (Req. 1.5.)

Attributi: Nessuno



**Relationship `tappaltin`**

Ogni istanza di questa relationship lega un `Itinerario` ad una sua `Tappa` (Req. 3.2.)

Attributi: Nessuno

**Relationship `tappaDest`**

Ogni istanza di questa relationship lega una `Tappa` di un `Itinerario` alla relativa `Destinazione` (Req. 3.2.)

Attributi: Nessuno

**Relationship `partItin`**

Ogni istanza di questa relationship lega un `Itinerario` alla relativa `Destinazione` di partenza (Req. 3.2.)

attributo	dominio	molteplicità	descrizione
partenzaOra	ora		L'ora di partenza della <code>Crociera</code>

**Relationship `arrivoltin`**

Ogni istanza di questa relationship lega un `Itinerario` alla relativa `Destinazione` di arrivo (Req. 3.2.)

attributo	dominio	molteplicità	descrizione
arrivo	<code>DeltaDataOra</code>		La data (in termini di giorni dalla data di partenza dell'itinerario) e l'ora di arrivo della <code>Crociera</code> alla <code>Tappa</code>

**Relationship `postoDest`**

Ogni istanza di questa relationship lega una `Destinazione` ad un `Posto da vedere` (Req. 4.3.)

Attributi: Nessuno

**Relationship `prenCliente`**

Ogni istanza di questa relationship lega una `Prenotazione` al relativo `Cliente` (Req. 7.1.)

Attributi: Nessuno

**Relationship `prenCroc`**

Ogni istanza di questa relationship lega una `Prenotazione` alla relativa `Crociera` (Req. 7.3.)

Attributi: Nessuno

**Dominio DeltaDataOra**

Il dominio è un record composto dai seguenti campi:

- giorni: intero  $> 0$
- ora: ora

**Dominio FasciaOraria**

Il dominio è un record composto dai seguenti campi:

- da: ora
- a: ora

**Dominio Indirizzo**

Il dominio è un record composto dai seguenti campi:

- via: stringa
- civico: intero  $> 0$  (0,1)
- CAP: stringa di 5 cifre numeriche
- città: stringa
- nazione: stringa



Questo materiale è concesso a  
Filippo Borsi  
esclusivamente per uso personale.  
È vietata la sua diffusione in qualsiasi forma.

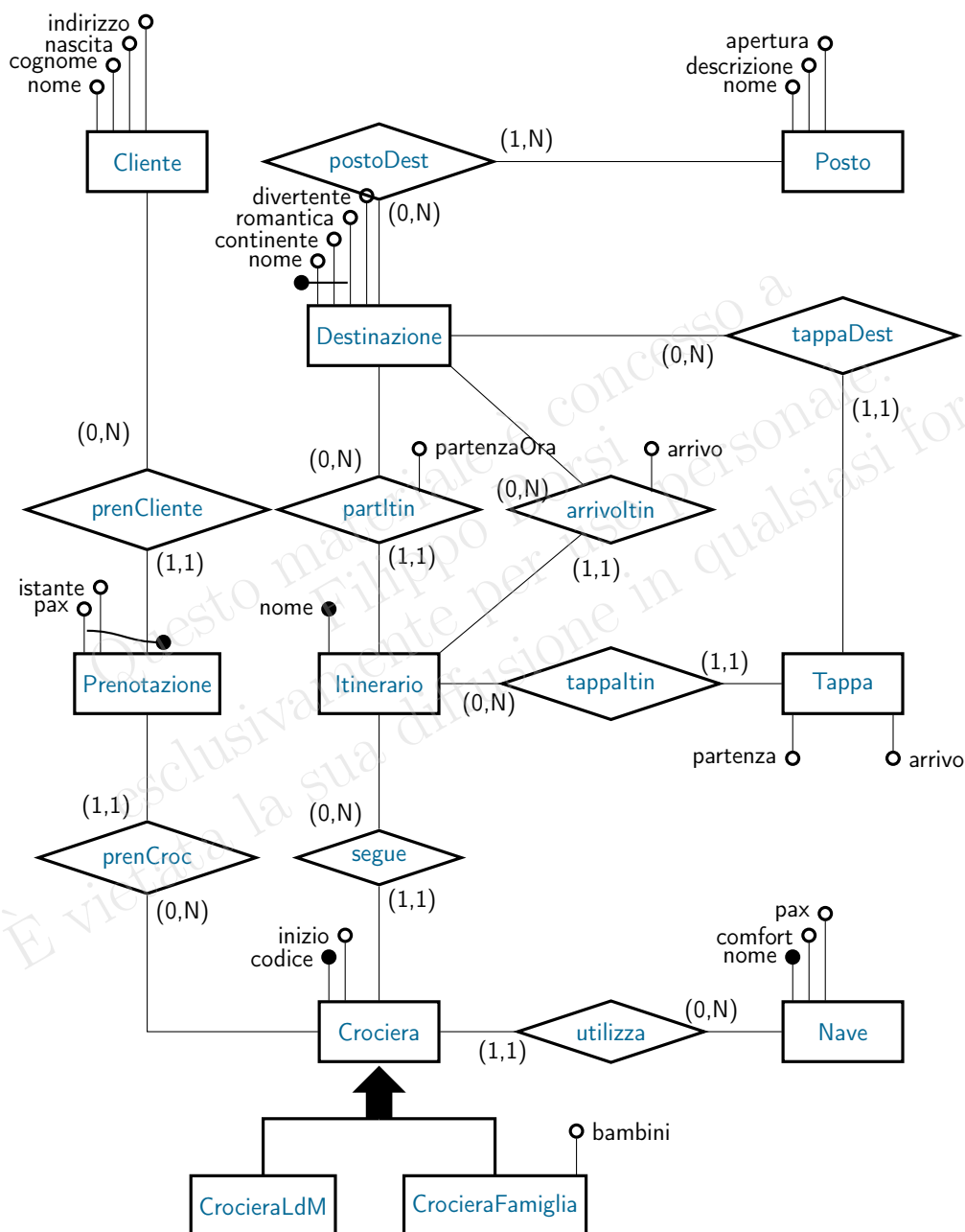
# A.5

## Identificazione delle Entità e Definizione Informale di Vincoli Esterni

### 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 con eventuali vincoli di identificazione delle entità che provengono dai requisiti e la definizione informale di eventuali ulteriori vincoli esterni.

## A.5.2 Soluzione



## Specifiche dei Dati

### Entità **Crociera**

Ogni istanza di questa entità rappresenta una crociera (Req. 1.)

attributo	dominio	molteplicità	descrizione
codice	stringa		Il codice della crociera
inizio	data		La data di inizio della crociera

### Vincoli:

**[V.Crociera.posti]** Il numero di posti prenotati per ogni crociera non può mai eccedere il numero di posti disponibili nella nave utilizzata.

Per ogni istanza *cr* dell'entità **Crociera**, siano:

- *postiOccupati* il numero totale di posti prenotati per la crociera *cr*, ovvero la somma dei valori dell'attributo *pax* di tutte le istanze *z* dell'entità **Prenotazione** tali per cui (*z*, *cr*) è un'istanza della relationship **prenCrociera**
- *totPosti* il numero totale di posti disponibili sulla nave utilizzata dalla crociera *cr*, ovvero il valore dell'attributo *pax* dell'istanza *n* dell'entità **Nave** per la quale (*cr*, *n*) è un'istanza della relationship **utilizza**.

Deve essere  $\text{totPosti} - \text{postiOccupati} \geq 0$ .

### Entità **Nave**

Ogni istanza di questa entità rappresenta una nave (Req. 2.)

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome della nave
comfort	[3,5]		Il grado di comfort della nave
pax	intero > 0		Il numero massimo di passeggeri per la nave

### Vincoli:

**[V.Nave.crociere.disj]** Le crociere che utilizzano la stessa nave devono avere periodi tutti disgiunti.

Per ogni coppia di istanze diverse *c* e *c'* dell'entità **Crociera** che utilizzano la stessa Nave *n* (ovvero tali per cui sia (*c*,*n*) che (*c'*,*n*) sono istanze della relationship **utilizza**), siano:

- *c\_in* e *c'\_in* i valori dell'attributo *inizio* per, rispettivamente, *c* e *c'*



- $itin\_c$  e  $itin\_c'$  le istanze dell'entità Itinerario per cui si ha che  $(c, itin\_c)$  e  $(c', itin\_c')$  sono istanze della relationship segue
- $dest\_arrivo\_c$  e  $dest\_arrivo\_c'$  le istanze dell'entità Destinazione per cui si ha che  $(dest\_arrivo\_c, itin\_c)$  e  $(dest\_arrivo\_c', itin\_c')$  sono istanze della relationship arrivoltin
- $arrivo\_c$  e  $arrivo\_c'$  i valori (di dominio DeltaDataOra) dell'attributo arrivo, rispettivamente, delle istanze  $(dest\_arrivo\_c, itin\_c)$  e  $(dest\_arrivo\_c', itin\_c')$  della relationship arrivoltin
- $giorno\_arrivo\_c$  e  $giorno\_arrivo\_c'$  i valori del campo giorni, rispettivamente, delle istanze  $arrivo\_c$  e  $arrivo\_c'$ .

Deve essere:

$$c\_in > c\_in + giorno\_arrivo\_c \text{ giorni}$$

oppure

$$c\_in > c'\_in + giorno\_arrivo\_c' \text{ giorni.}$$

### Entità **Itinerario**

Ogni istanza di questa entità rappresenta un'itinerario (Req. 3.)

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome dell'itinerario

### Entità **Destinazione**

Ogni istanza di questa entità rappresenta una destinazione toccata da **Itinerari** (Req. 4.)

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome della destinazione
continente	{E, AS, AMN, AMS, AF, O}		Il continente della destinazione (nell'ordine: Europa, Asia, America del Nord, America del Sud, Africa, Oceania)
romantica	booleano		true se la destinazione è classificata come romantica, false altrimenti
divertente	booleano		true se la destinazione è classificata come divertente, false altrimenti

**Entità Tappa**

Ogni istanza di questa entità rappresenta una **Tappa** toccata da un **Itinerario** (Req. 3.2.), escluse quelle di partenza e arrivo dell'**Itinerario**

attributo	dominio	molteplicità	descrizione
arrivo	<b>DeltaDataOra</b>		La data (relativa alla data di partenza della crociera) e l'ora di arrivo alla <b>Destinazione</b>
partenza	<b>DeltaDataOra</b>		La data (relativa alla data di partenza della crociera) e l'ora di ripartenza dalla <b>Destinazione</b>

**Vincoli:**

**[V.Tappa.date]** La data di partenza da ogni tappa segue la relativa data di arrivo.

Per ogni istanza  $t$  dell'entità Tappa avente  $a$  e  $p$  come valori, rispettivamente, dei suoi attributi arrivo e partenza, deve essere:  $a < p$ .

**[V.Tappa.succPart]** Le istanze dell'entità Tappa con giorno di arrivo pari ad 1 devono avere valore per l'ora di arrivo successivo all'orario di partenza del relativo itinerario. In altri termini, eventuali tappe di un'itinerario previste nel giorno di viaggio 1 devono essere previste in orari successivi a quello di inizio dell'itinerario.

Per ogni istanza  $t$  dell'entità Tappa, siano:

- $i$  l'itinerario relativo a  $t$ , ovvero l'istanza di Itinerario tale che  $(t, i)$  è un'istanza della relationship tappaltin
- $p$  la destinazione di partenza di  $i$ , ovvero l'istanza dell'entità Destinazione tale che  $(p, i)$  è una istanza della relationship partltin
- $o_p$  l'ora di partenza dell'itinerario dalla destinazione di partenza  $p$ , ovvero il valore dell'attributo partenzaOra dell'istanza  $(p, i)$  della relationship partltin
- $a_t$  il momento di arrivo previsto alla tappa  $t$ , ovvero il valore dell'attributo arrivo (di dominio DeltaDataOra) dell'istanza  $t$
- $g_t$  e  $o_t$  i valori, rispettivamente, dei campi giorni e ora di  $a_t$ .

Se il campo  $g_t$  è pari ad 1, allora deve essere:  $o_p < o_t$ .

**[V.Tappa.primaDiArrivo]** Le istanze dell'entità Tappa devono avere un valore per l'attributo partenza precedente al valore dell'attributo arrivo della destinazione finale del relativo itinerario.

Per ogni istanza  $t$  dell'entità Tappa, siano:



- $i$  l'itinerario relativo a  $t$ , ovvero l'istanza di `Itinerario` tale che  $(t, i)$  è un'istanza della relationship `tappatin`
- `dest_arr` la destinazione di arrivo di  $i$ , ovvero l'istanza dell'entità `Destinazione` tale che  $(dest\_arr, i)$  è una istanza della relationship `arrivoltin`
- `arr` l'istante di arrivo dell'itinerario  $i$  alla destinazione di arrivo `dest_arr`, ovvero il valore (di dominio `DeltaDataOra`) dell'attributo `arrivo` dell'istanza  $(dest\_arr, i)$  della relationship `arrivoltin`
- `t_part` il momento di partenza previsto dalla tappa  $t$ , ovvero il valore dell'attributo `partenza` (di dominio `DeltaDataOra`) dell'istanza  $t$

Deve essere:  $t\_part < arr$ .

### Entità **Posto**

Ogni istanza di questa entità rappresenta un posto da vedere (Req. 5.)

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome del posto da vedere
descrizione	stringa		La descrizione del posto da vedere
apertura	<a href="#">FasciaOraria</a>		La fascia oraria di apertura del posto da vedere

### Entità **Cliente**

Ogni istanza di questa entità rappresenta un cliente (Req. 6.)

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome del cliente
cognome	stringa		Il cognome del cliente
nascita	data		La data di nascita del cliente
indirizzo	<a href="#">Indirizzo</a>		L'indirizzo del cliente

### Entità **Prenotazione**

Ogni istanza di questa entità rappresenta una prenotazione di una [Crociera](#) da parte di un [Cliente](#) (Req. 7.)

attributo	dominio	molteplicità	descrizione
pax	intero $> 0$		Il numero di posti della prenotazione
istante	dataora		L'istante nel quale è stata effettuata la prenotazione

**Vincoli:**

**[V.Prenotazione.istante]** Le prenotazioni devono essere effettuate prima del giorno di inizio della relativa crociera.

Per ogni istanza  $p$  di entità Prenotazione, siano:

- $i$  il valore dell'attributo istante di  $p$
- $cr$  l'istanza dell'entità Crociera per la quale  $(p, cr)$  è un'istanza della relationship  $prenCroc$
- $cr\_i$  il valore dell'attributo inizio di  $cr$ .

Deve essere  $i < cr\_i$ .

**Entità [CrocieraLdM](#)**

Ogni istanza di questa entità rappresenta una crociera di luna di miele (Req. [1.6.1.](#))

Attributi: Nessuno

**Entità [CrocieraFamiglia](#)**

Ogni istanza di questa entità rappresenta una crociera per famiglia (Req. [1.6.2.](#))

attributo	dominio	molteplicità	descrizione
bambini	booleano		true se la crociera è adatta ai bambini, false altrimenti

**Relationship [utilizza](#)**

Ogni istanza di questa relationship lega una [Crociera](#) alla [Nave](#) utilizzata (Req. [1.4.](#))

Attributi: Nessuno

**Relationship [segue](#)**

Ogni istanza di questa relationship lega una [Crociera](#) al suo [Itinerario](#) (Req. [1.5.](#))

Attributi: Nessuno

**Relationship [tappaltin](#)**

Ogni istanza di questa relationship lega un [Itinerario](#) ad una sua [Tappa](#) (Req. [3.2.](#))

Attributi: Nessuno

**Vincoli:**

**[V.tappaltin.soste.disj]** I periodi di sosta di ogni itinerario nelle diverse tappe sono tutti disgiunti.

Per ogni coppia di istanze diverse  $t$  e  $t'$  dell'entità Tappa relative allo stesso Itinerario  $i$  (ovvero tali per cui sia  $(t,i)$  che  $(t',i)$  sono istanze della relationship tappaltin), siano:

- $t\_arr$  e  $t\_part$  i valori degli attributi, rispettivamente, arrivo e partenza di  $t$
- $t'\_arr$  e  $t'\_part$  i valori degli attributi, rispettivamente, arrivo e partenza di  $t'$ .

Deve essere:  $t'\_arr > t\_part$  oppure  $t\_arr > t'\_part$ .

#### Relationship **tappaDest**

Ogni istanza di questa relationship lega una Tappa di un Itinerario alla relativa Destinazione (Req. 3.2.)

Attributi: Nessuno

#### Relationship **partItin**

Ogni istanza di questa relationship lega un Itinerario alla relativa Destinazione di partenza (Req. 3.2.)

attributo	dominio	molteplicità	descrizione
partenzaOra	ora		L'ora di partenza della Crociera

#### Relationship **arrivoltin**

Ogni istanza di questa relationship lega un Itinerario alla relativa Destinazione di arrivo (Req. 3.2.)

attributo	dominio	molteplicità	descrizione
arrivo	DeltaDataOra		La data (in termini di giorni dalla data di partenza dell'itinerario) e l'ora di arrivo della Crociera alla Tappa

#### Relationship **postoDest**

Ogni istanza di questa relationship lega una Destinazione ad un Posto da vedere (Req. 4.3.)

Attributi: Nessuno

#### Relationship **prenCliente**

Ogni istanza di questa relationship lega una Prenotazione al relativo Cliente (Req. 7.1.)

Attributi: Nessuno

**Relationship [prenCroc](#)**

Ogni istanza di questa relationship lega una [Prenotazione](#) alla relativa [Crociera](#) (Req. [7.3.](#))

Attributi: Nessuno

**Dominio [DeltaDataOra](#)**

Il dominio è un record composto dai seguenti campi:

- giorni: intero  $> 0$
- ora: ora

**Dominio [FasciaOraria](#)**

Il dominio è un record composto dai seguenti campi:

- da: ora
- a: ora

**Dominio [Indirizzo](#)**

Il dominio è un record composto dai seguenti campi:

- via: stringa
- civico: intero  $> 0$  (0,1)
- CAP: stringa di 5 cifre numeriche
- città: stringa
- nazione: stringa



Questo materiale è concesso a  
Filippo Borsi  
esclusivamente per uso personale.  
È vietata la sua diffusione in qualsiasi forma.

# A.6

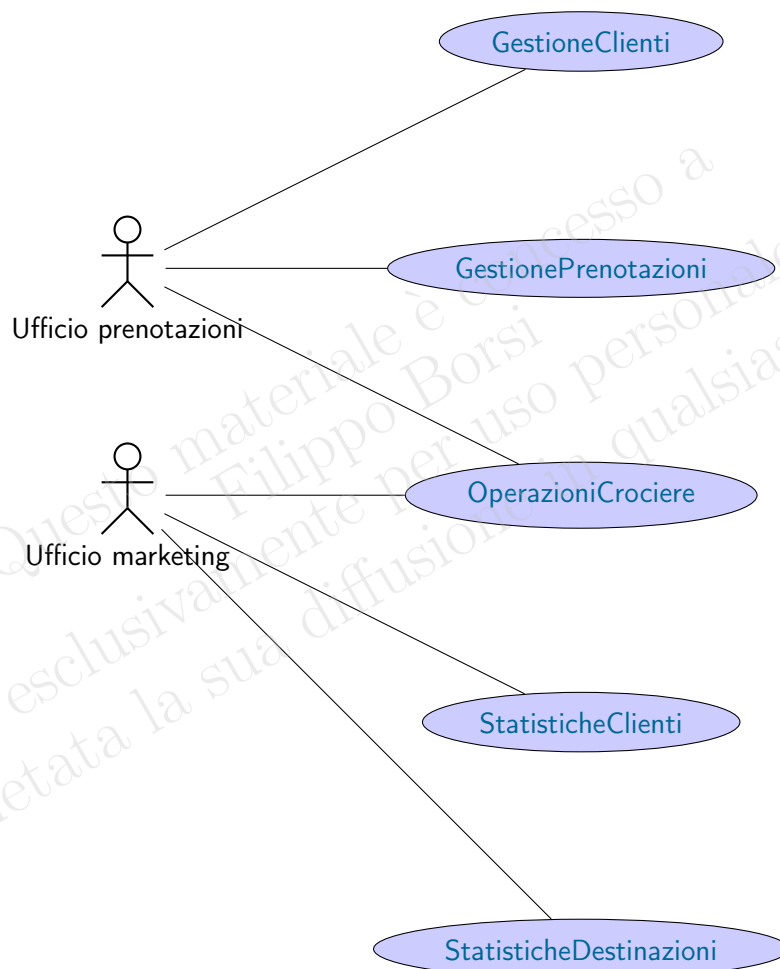
## Diagramma UML degli Use-Case

### A.6.1 Testo

Proseguire la fase di Analisi Concettuale dei requisiti producendo un diagramma UML degli use-case.

Questo materiale è concesso a  
Filippo Borsi  
esclusivamente per uso personale.  
È vietata la sua diffusione in qualsiasi forma.

## A.6.2 Soluzione



# A.7

## Specifiche Concettuali Informali degli Use-Case

### A.7.1 Testo

Proseguire la fase di Analisi Concettuale dei requisiti producendo le specifiche concettuali informali degli use-case.

Questo materiale è concesso a  
Filippo Borsi  
esclusivamente per uso personale.  
È vietata la sua diffusione in qualsiasi forma.



## A.7.2 Soluzione

### Specifica Use-Case **GestionePrenotazioni**

- **prenota**(cl : Cliente, cr : Crociera, posti: intero  $> 0$ ) : Prenotazione (Req. 8.1.)

**precondizioni:** La data di inizio della crociera cr deve essere successiva all'istante corrente.

Inoltre, la crociera cr deve avere almeno posti liberi, ovvero: GestionePrenotazioni.postiDisponibili(cr)  $\geq$  posti.

**postcondizioni:** Il livello estensionale dei dati viene modificato creando le seguenti nuove istanze:

- $\alpha$  dell'entità Prenotazione, con valore posti per l'attributo pax e valore per l'attributo istante pari all'istante corrente
- ( $\alpha$ , cl) della relationship prenCliente
- ( $\alpha$ , cr) della relationship prenCrociera

Viene restituita l'istanza  $\alpha$ .

- **postiDisponibili**(cr : Crociera) : intero

**precondizioni:** nessuna

**postcondizioni:** La funzione non modifica il livello estensionale dei dati.

Siano:

- **postiOccupati** il numero totale di posti prenotati per la crociera cr, ovvero la somma dei valori dell'attributo pax di tutte le istanze z dell'entità Prenotazione tali per cui (z, cr) è un'istanza della relationship prenCrociera
- **totPosti** il numero totale di posti disponibili sulla nave utilizzata dalla crociera cr, ovvero il valore dell'attributo pax dell'istanza n dell'entità Nave per la quale (cr, n) è un'istanza della relationship utilizza.

Viene restituito **totPosti** – **postiOccupati**.

## Specifica Use-Case **GestioneClienti**

- nuovo(no : stringa, c : stringa, na: data, i: Indirizzo) : Cliente

**precondizioni:** nessuna

**postcondizioni:** Il livello estensionale dei dati viene modificato creando la seguente nuova istanza:

- $\alpha$  dell'entità Cliente, con valori no, c, na ed i rispettivamente per gli attributi nome, cognome, nascita, indirizzo.

Viene restituita l'istanza  $\alpha$ .

- cerca(no : stringa (0,1), co : stringa (0,1), na: data (0,1)) : Cliente (0,N)

**precondizioni:** nessuna

**postcondizioni:** La funzione non modifica il livello estensionale dei dati.

Viene restituito l'insieme delle istanze cl dell'entità Cliente che soddisfano tutti i seguenti criteri:

- se no è definito, allora il valore dell'attributo nome di cl deve essere uguale a no
- se c è definito, allora il valore dell'attributo cognome di cl deve essere uguale a no
- se na è definito, allora il valore dell'attributo nascita di cl deve essere uguale a na.

## Specifica Use-Case **OperazioniCrociera**

- **tipoCrociera**( $c : \text{CrocieraLdM}$ ) : {tradizionale, alternativa} (Req. 1.6.1.)

**precondizioni:** nessuna

**postcondizioni:** La funzione non modifica il livello estensionale dei dati.

Sia  $i$  l'itinerario della crociera  $c$ , ovvero l'istanza dell'entità Itinerario tale per cui  $(c, i)$  è istanza della relationship segue.

Siano:

- $D_r$  l'insieme delle destinazioni romantiche della crociera  $c$ , ovvero l'insieme delle istanze  $d$  dell'entità Destinazione tali che:
  - \* esiste un'istanza  $t$  dell'entità Tappa tale che  $(t, i)$  è istanza della relationship tappaIn e  $(t, d)$  è istanza della relationship tappaDest
  - \* il valore dell'attributo romantica di  $d$  è true
- $D_d$  l'insieme delle destinazioni divertenti della crociera  $c$ , ovvero l'insieme delle istanze  $d$  dell'entità Destinazione tali che:
  - \* esiste un'istanza  $t$  dell'entità Tappa tale che  $(t, i)$  è istanza della relationship tappaIn e  $(t, d)$  è istanza della relationship tappaDest
  - \* il valore dell'attributo divertente di  $d$  è true.

Viene restituito il valore 'tradizionale' se  $|D_r| \geq |D_d|$  e 'alternativa' altrimenti.

## Specifica Use-Case **StatisticheClienti**

- **etaMediaEsotica(C : Cliente (0,N)) : reale  $\geq 0$  (Req. 8.2.)**

**precondizioni:** Sia  $C_e$  il sottoinsieme dei clienti in  $C$  che hanno prenotato almeno una crociera il cui itinerario tocca una destinazione esotica, ovvero l'insieme contenente tutti e soli gli elementi  $c \in C$  che soddisfano tutti i seguenti criteri:

- esiste una istanza  $p$  di entità Prenotazione tale per cui  $(p,c)$  è istanza della relationship  $\text{prenCliente}$
- esiste una istanza  $cr$  di entità Crociera tale per cui  $(p,cr)$  è istanza della relationship  $\text{prenCroc}$
- esiste una istanza  $i$  di entità Itinerario tale per cui  $(cr,i)$  è istanza della relationship  $\text{segue}$
- esiste una istanza  $t$  di entità Tappa tale per cui  $(t,i)$  è istanza della relationship  $\text{tappatin}$
- esiste una istanza  $d$  di entità Destinazione tale per cui  $(t,d)$  è istanza della relationship  $\text{tappaDest}$
- $\text{StatisticheDestinazioni.esotica}(d)$ .

Deve essere  $|C_e| \geq 1$ .

**postcondizioni:** La funzione non modifica il livello estensionale dei dati.

Viene restituito:

$$\text{result} = \frac{\sum_{c \in C_e} \text{StatisticheClienti.eta}(c)}{|C_e|}.$$

- **eta(c : Cliente) : intero  $\geq 0$**

**precondizioni:** nessuna

**postcondizioni:** La funzione non modifica il livello estensionale dei dati.

Sia  $na$  il valore (di dominio data) dell'attributo nascita di  $c$  e sia  $diff$  la differenza (di dominio data) tra la data corrente e il valore  $na$ .

Viene restituito il valore del campo anno di  $diff$ .

## Specifica Use-Case **Statistiche Destinazioni**

- **esotica(d : Destinazione) : booleano**

**precondizioni:** nessuna

**postcondizioni:** La funzione non modifica il livello estensionale dei dati.

Sia cont il valore dell'attributo continente di d.

Viene restituito 'true' se cont  $\neq$  'E' e 'false' altrimenti.

- **gettonata(d : Destinazione) : booleano**

**precondizioni:** nessuna

**postcondizioni:** La funzione non modifica il livello estensionale dei dati.

Siano L e F l'insieme, rispettivamente, delle crociere di luna di miele e per famiglia degli ultimi due anni che toccano la destinazione d:

- L è l'insieme delle istanze ldm dell'entità CrocieraLdM che soddisfano tutti i seguenti criteri:
  - \* ldm ha un valore in per l'attributo inizio (di dominio data) il cui campo anno ha un valore a tale per cui la differenza tra il valore dell'anno corrente e a è al più 2
  - \* esiste una istanza it di entità Itinerario tale per cui (ldm, it) è una istanza della relationship segue
  - \* esiste una istanza t di entità Tappa tale per cui (it, t) è una istanza della relationship tappaltin
  - \* (t, d) è una istanza della relationship tappaDest.
- F è l'insieme delle istanze fam dell'entità CrocieraFamiglia che soddisfano tutti i seguenti criteri:
  - \* fam ha un valore in per l'attributo inizio (di dominio data) il cui campo anno ha un valore a tale per cui la differenza tra il valore dell'anno corrente e a è al più 2
  - \* esiste una istanza it di entità Itinerario tale per cui (fam, it) è una istanza della relationship segue
  - \* esiste una istanza t di entità Tappa tale per cui (it, t) è una istanza della relationship tappaltin
  - \* (t, d) è una istanza della relationship tappaDest.

Viene restituito 'true' se  $|L| \geq 10$  oppure se  $|F| \geq 15$ , e 'false' altrimenti.

- **percentualeGettonate(D: Destinazione (1,N)):** reale in [0,1] (Req. 8.3.)

**precondizioni:** nessuna



**postcondizioni:** La funzione non modifica il livello estensionale dei dati.

Sia  $D'$  l'insieme composto da tutte e sole le destinazioni  $d \in D$  gettonate, ovvero per cui vale: `StatisticheDestinazioni.gettonata(d) = true`.

Viene restituito  $\frac{|D'|}{|D|}$ .

Questo materiale è concesso a  
Filippo Borsi  
esclusivamente per uso personale.  
È vietata la sua diffusione in qualsiasi forma.