

# Gestionale servizi del diporto

Progetto di basi di dati

Alessandro Massarenti, Filippo Zonta Rocha

a.a. 2021/2022

## 0.1 Abstract

Moltissime imbarcazioni ogni giorno navigano i mari di tutto il mondo, e dopo giorni, a volte settimane, di navigazione è importante per gli equipaggi di questi scafi da diporto riposare in un posto sicuro, accogliente ed efficiente.

Per rispondere a questa necessità numerose località di mare, turistiche e non, si sono dotate di piccoli porticcioli in luoghi strategici, mirati a proteggere e sistemare le imbarcazioni, a semplificare le pratiche burocratiche e a connettere le persone al mare.

Questi porticcioli si chiamano **marina**. I marina devono soddisfare molti requisiti e hanno molte operazioni che possono essere migliorate e coadiuvate da un sistema informativo, molte operazioni sono infatti ripetitive, lavorano su grosse quantità di dati o sono richieste ad orari anomali. Un numero significativo di imbarcazioni arriva nei marina agli orari più disparati, e non sempre è possibile avere tutto il personale disponibile per gestire le richieste.

## 1 Analisi dei requisiti

### 1.1 Descrizione testuale

Si vuole realizzare una base di dati per l'applicazione *Gestionale servizi del diporto* che dovrà gestire tutti i dati relativi ad un marina e le **imbarcazioni** che vi sosterranno, i **servizi** offerti e i dati relativi ai **clienti**.

Le funzioni desiderate prevedono: il calcolo delle **fatture** dei clienti; la gestione di **soste** e **prenotazioni** con recupero dei posti disponibili; la presentazione dei **servizi** aperti ai clienti.

Il marina in esame ha circa 250 **posti barca**, numero che cambia solo in caso di ristrutturazioni particolari o durante ampliamenti del marina. Il traffico di **imbarcazioni**, **persone** ed informazioni è quindi commisurato a questo numero di **posti barca**.

Alcuni dei **clienti** del marina sono *occasional*, ad esempio persone che effettuano crociere e navigano di porto in porto; Altri clienti sono *abituali* e tengono ormeggiate le loro imbarcazioni nel marina per molti mesi all'anno se non addirittura a tempo indeterminato. Dei primi si vuole sapere quante volte hanno ormeggiato nel marina, dei secondi si vuole tenere nota del loro sconto personale applicato alle fatture. Spesso un *cliente occasionale* diventa un *cliente abituale* e quando questo accade gli viene assegnato uno sconto deciso per persona.

Dei **moli**<sup>1</sup> si vuole tenere traccia della loro occupazione nel tempo per fini statistici<sup>2</sup>. Per ogni **molo**, identificato da un numero positivo, è importante memorizzare le dimensioni (Molto importanti, poiché solo le imbarcazioni più piccole di queste dimensioni potranno ormeggiare in questo **moli**), il prezzo giornaliero, gli allacciamenti<sup>3</sup> e lo stato di occupazione.

Gli **allacciamenti** sono identificati dal nome e hanno un'unità di misura ed un costo unitario.

Relativamente agli **allacciamenti** ne vengono anche registrati i **consumi** da parte di ogni cliente, come procedura interna del marina viene fatta una lettura del contatore dall'inizio del mese<sup>4</sup> alla fine del mese. Se l'imbarcazione arriva e/o parte in momenti differenti dall'inizio del mese la lettura viene fatta in quelle due occasioni.

Questi **consumi**, assieme alle **soste** effettuate, vengono poi fatturati al cliente e viene registrato se la **fattura** è stata pagata oppure no. La **fattura** è identificata univocamente dal cliente destinatario e la data di scadenza della stessa.

<sup>1</sup> Sinonimo di posto barca

<sup>2</sup> Ad esempio il **moli** preferito da una determinata persona o il **moli** più utilizzato

<sup>3</sup> Si intendono allacciamenti idrici, elettrici, di aria compressa

<sup>4</sup> Chiaramente solo se c'è una barca in sosta

Da quando l'applicativo verrà messo in funzione si inizieranno a memorizzare le **soste**.

Le **imbarcazioni** che sosterranno nel marina sosterranno nel tempo in più moli differenti, e chiaramente i **moli** accoglieranno nel tempo imbarcazioni differenti, ma non contemporaneamente. Un **molo** può infatti accogliere solo un' **imbarcazione** alla volta.

Di ogni **imbarcazione** vengono registrati codice internazionale<sup>5</sup> univoco, la bandiera battuta, il nome del capitano, le dimensioni<sup>6</sup>, il numero di posti letto e se è presente anche il nome. Queste informazioni vengono memorizzate assieme ai dati dei **clienti**, ovvero i proprietari delle suddette **imbarcazioni** per onorare gli obblighi di registrazione doganale.

I **clienti** sono coloro che possiedono un' **imbarcazione** ormeggiata all'interno del marina e di loro ne viene salvato il nome, il cognome, il codice fiscale, gli eventuali contatti, la residenza, la cittadinanza, la data di nascita.

Ogni **cliente** può quindi prenotare un **molo** del marina preventivamente. Ogni **prenotazione** è caratterizzata da una data prevista di arrivo, una data prevista di partenza, il **molo** interessato, il cliente prenotante e un'informazione per capire se la prenotazione è stata trasformata in sosta oppure no.

Per controllare se un **molo** è disponibile bisogna quindi controllare se il **molo** è al momento occupato e inoltre se esistono già delle **prenotazioni** per le date in cui si prevede che l'**imbarcazione** sosti. Se la **sosta** prevista è a tempo indeterminato ma una prenotazione parte più tardi, l'imbarcazione può essere spostata a sostare in un altro **moli** non prenotato.

Il marina offre poi una serie di **servizi** utili ai naviganti<sup>7</sup>. Ogni **servizio** è caratterizzato da un nome unico e gli **orari** in cui è aperto nei vari giorni di una settimana.

Ogni servizio è amministrato da un **addetto** del quale ci interessa la data di inizio contratto e la fine. Se la fine non è segnata allora lavorerà lì a tempo indeterminato. Ogni **addetto** è definito univocamente dalla data di inizio contratto e il servizio che gestisce. Per questioni legate al loro contratto di lavoro ci interessa di loro anche il codice fiscale e la data di nascita, oltre al loro nome, cognome e ai loro contatti.

Gli **addetti** inoltre spesso sono anche **clienti**. Una **persona** può essere quindi o un **addetto** o un **cliente** o entrambi.

---

<sup>5</sup>MMSI

<sup>6</sup>Definite come: Pescaggio, Larghezza e lunghezza fuoritutto con acronimo inglese LOA

<sup>7</sup>Ad esempio lavanderie a gettoni, ristoranti, cantieri

## 1.2 Glossario dei termini

Entità	Descrizione	Sinonimi	collegamenti
Imbarcazione	Il mezzo di trasporto ormeggiato nei moli		Cliente, Sosta
Molo	Il luogo in cui le imbarcazioni sostano	Posto barca, ormeggio	Sosta, Prenotazione, Allacciamento
Servizio	Attività di varia utilità rivolte ai clienti del marina		Periodo apertura, addetto
Addetto	Persona che gestisce un servizio	Responsabile	Servizio, Persona
Cliente	Persona che possiede un'imbarcazione ormeggiata nel marina	Armatore	Entità figlia di Persona, Imbarcazione, Consumo, Prenotazione
Cliente occasionale	Persone che viaggiano salturiamente		Entità figlia di Cliente
Cliente abituale	Persone che utilizzano stabilmente un marina		Entità figlia di Cliente
Persona	Tutte le persone che frequentano un marina		Padre di Cliente e Addetto
Prenotazione	Tutte le prenotazioni dei clienti		Molo, cliente
Sosta	Possibilità data ai clienti delle imbarcazioni di un marina		Imbarcazione, Molo
Allacciamento	Un servizio consumabile tipo acqua o elettricità disponibile ad un molo	Fornitura	Molo, Consumo
Periodo di apertura	Orari in cui un servizio è aperto durante la settimana		Servizio
Consumo	Utilizzo misurato di un allacciamento presente ai moli		Allacciamento, Cliente, Fattura
Fattura	Ricevuta sui consumi e le soste effettuate da parte dei clienti	Ricevuta, scontrino	Cliente, Consumo, Sosta

## 1.3 Operazioni tipiche

Contando che al marina appartengono circa 250 **moli**, in media le operazioni effettuate nel marina con annessa frequenza sono le seguenti.

Operazione	Frequenza
1. Stampa della fattura di pagamento di un cliente	40 volte al giorno
2. Generazione conteggi fatture clienti	30-40 volte al giorno più picco mensile a fine mese <sup>8</sup>
3. Controllo dei posti disponibili per una certa imbarcazione (Controllo delle dimensioni)	40 volte al giorno
4. Cambio dell'addetto ad un servizio	1-2 volte ogni due anni
5. Prenotazioni di posti barca	30-50 volte al giorno
6. Arrivo di un'imbarcazione nel marina	30-40 volte al giorno
7. Consultazione delle aperture di un servizio	20-40 volte al giorno

<sup>8</sup>Vengono calcolate le fatture mensili di tutti i clienti che stanno oltre il mese

## 2 Progettazione concettuale

### 2.1 Analisi delle entità

Imbarcazione			
Codice internazionale	CHAR(9)	Identifica univocamente un'imbarcazione nel mondo	Chiave
Nome	VARCHAR	Il nome dell'imbarcazione	
QtPostiLetto	INTEGER	La quantità di posti letto dell'imbarcazione	
Nome del capitano	VARCHAR	Il nome del capitano	
Bandiera	VARCHAR	Nome dello stato di cui batte bandiera l'imbarcazione	
Dimensioni		Attributo composto: Pescaggio, Larghezza, LOA	
Pescaggio	DECIMAL	La misura del punto più in profondità dell'imbarcazione	
Larghezza	DECIMAL	La misura della larghezza al baglio massimo dell'imbarcazione	
LOA	DECIMAL	La misura della lunghezza fuoritutto dell'imbarcazione	

Molo			
Numero molo	INTEGER	Identifica univocamente un molo	Chiave
Stato di occupazione	BOOLEAN	Lo stato di occupazione del molo	
Dimensioni		Attributo composto: Profondità minima, Larghezza, Lunghezza	
Profondità minima	DECIMAL	La profondità misurata alla più bassa marea sizigiale	
Larghezza	DECIMAL	La misura della larghezza del molo	
Lunghezza	DECIMAL	La misura della lunghezza del molo	

Servizio			
Nome	VARCHAR	Il nome del servizio	Chiave

Addetto			
Servizio	VARCHAR	È il servizio in cui lavora	Chiave
DataInizioContratto	DATE	È la data in cui il contratto è iniziato	
DataFineContratto	DATE	È la data in cui il contratto è finito o finirà.	

Cliente		
Cittadinanza	VARCHAR	Lo stato di cui il cliente è cittadino
Residenza	VARCHAR	La città in cui il cliente è residente

Cliente occasionale		
Quantità di soste	INTEGER	Il numero di soste che il cliente ha effettuato presso la struttura

Cliente abituale		
Sconto personale	DECIMAL	Lo sconto percentuale applicato alle sue fatture

Persona			
CF	CHAR(16)	Identifica univocamente una persona	Chiave
Nome	VARCHAR	Il nome della persona	
Cognome	VARCHAR	Il cognome della persona	
DataNascita	DATE	La data di nascita della persona	
Contatto	VARCHAR	I contatti della persona(multi-valore)	

Prenotazione			
CF	VARCHAR	Il codice fiscale del cliente prenotante	Chiave
Molo	INTEGER	Il molo prenotante	
Previsione arrivo	DATE	La data in cui si prevede il cliente arrivo	
Previsione partenza	DATE	La data in cui si prevede il cliente parta	
Sosta	SOSTA	La sosta in cui si è trasformata la presente prenotazione	

Sosta			
Imbarcazione	VARCHAR	Il codice internazionale dell'imbarcazione	Chiave
Molo	INTEGER	Il molo della sosta	
Data arrivo	TIMESTAMP	Data in cui l'imbarcazione ha ormeggiato	
Data partenza	TIMESTAMP	Data in cui l'imbarcazione ha sciolto gli ormeggi	

Allacciamento			
Nome	VARCHAR	Il nome che identifica la tipologia di allacciamento	Chiave
Prezzo unitario	DECIMAL	Il prezzo unitario dell'allacciamento	
Unità di misura	VARCHAR	L'unità di misura utilizzata per conteggiarne il consumo	

Periodo di apertura			
Giorno di apertura	CHAR(3)	Giorno a cui si riferisce l'orario di apertura	Chiave
Orario di apertura	TIME	Orario a cui il servizio apre nel giorno	
Orario di chiusura	TIME	Orario a cui il servizio chiude nel giorno	

Consumo			
Cliente	VARCHAR	Cliente consumante	Chiave
Allacciamento	VARCHAR	il tipo di allacciamento consumato	
Data inizio lettura	TIMESTAMP	Momento dell'inizio del conteggio	
Data fine lettura	TIMESTAMP	Momento della fine del conteggio	
Quantità consumata	DECIMAL	La quantità consumata nel periodo	

Fattura			
Cliente	VARCHAR	Cliente ricevente	Chiave
Data scadenza	DATE	Identifica la scadenza della fattura	
Pagato	TIMESTAMP	Identifica quando è stata pagata e lo stato del pagamento della fattura	

## 2.2 Generalizzazioni

- Persona è generalizzazione parziale esclusiva di Addetto;
- Persona è generalizzazione parziale esclusiva di Cliente;
- Cliente è generalizzazione totale esclusiva di cliente abituale e cliente occasionale;

## 2.3 Analisi delle relazioni e delle cardinalità

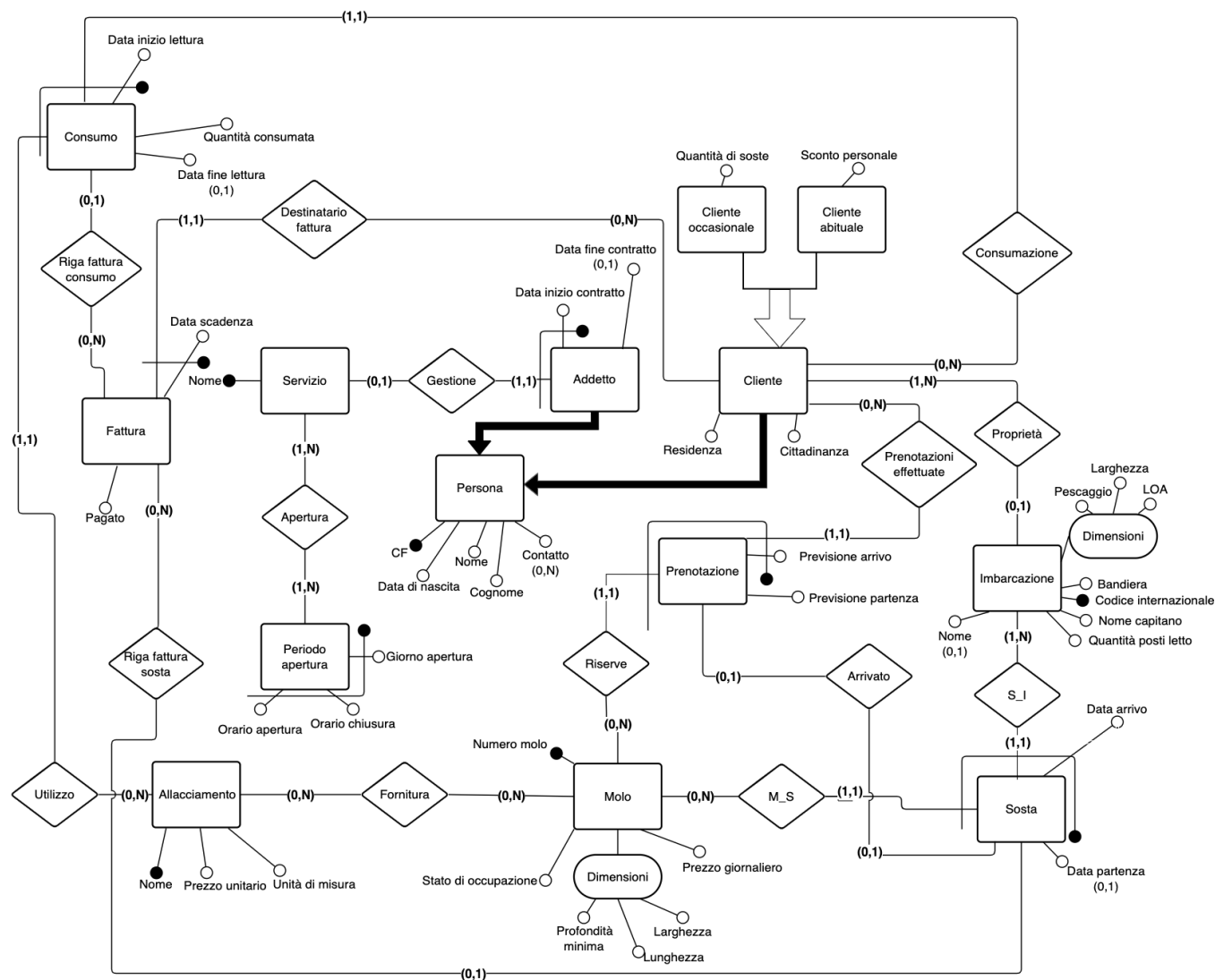
- Molo - Allacciamento: **Fornitura**
  - Un Molo possiede più allacciamenti o nessuno (0,N);
  - Un allacciamento può essere posseduto da più moli o da nessuno(0,N);
- Consumo - Allacciamento: **Utilizzo**
  - Un consumo utilizza uno e un solo allacciamento(1,1);
  - Un allacciamento può essere utilizzato da 0 a molti consumi(0,N);
- Consumo - Fattura: **Riga fattura consumo**
  - Un consumo è riga di una o nessuna fattura(0,1);
  - Una fattura ha da 0 a molte righe relative ai consumi(0,N);
- Sosta - Fattura: **Riga fattura sosta**
  - Una sosta è riga di una o nessuna fattura(0,1);
  - Una fattura ha da 0 a molte righe relative alle soste (0,N);

- Fattura - Cliente: **Destinatario fattura**
  - Una fattura è destinata ad uno e un solo cliente(0,1);
  - Un cliente può avere da 0 a molte fatture destinategli(0,N);
- Addetto - Servizio: **Gestione**
  - Un addetto gestisce uno e un solo servizio (1,1);
  - Un servizio è gestito da un addetto o da nessuno (0,1);
- Servizio - Periodo apertura: **Apertura**
  - Un servizio ha da uno a molte aperture(1,N);
  - Periodo di apertura definisce aperte da uno a molti servizi (1,N);
- Consumo - Cliente: **Consumazione**
  - Un Cliente effettua da 0 a molte consumazioni di consumazione(0,N);
  - Un consumo è consumazione di uno e un solo cliente (1,1);
- Prenotazione - Molo: **Riserve**
  - Una prenotazione Riserva uno e un solo molo (1,1);
  - Un molo può essere riservato da più prenotazioni o da nessuna (0,N);
- Cliente - Imbarcazione: **Proprietà**
  - Un cliente può possedere più imbarcazioni(Almeno una) (1,N);
  - Un imbarcazione è posseduta da una e una sola una persona oppure da nessuno<sup>9</sup>(0,1);
- Cliente - Prenotazione: **Prenotazioni effettuate**
  - Un cliente può aver effettuato da 0 a molte prenotazioni(0,N);
  - Una prenotazione può essere stata effettuata al massimo da un cliente (1,1);
- Molo - Sosta: **M\_S**
  - Un molo può accogliere da 0 a molte soste(0,N);
  - Una sosta è relativa a uno e un solo molo(1,1);
- Sosta - Imbarcazione: **S\_I**
  - Una sosta è relativa ad una e una sola imbarcazione(1,1);
  - Un imbarcazione può sostare da 0 a molte volte (0,N);
- Prenotazione - Sosta: **Arrivato**
  - Una prenotazione Arriva in 0 o 1 molo(0,1);
  - Un sosta ha in arrivo da 0 a una prenotazione (0,N);

---

<sup>9</sup>Barca abbandonata

## 2.4 Schema ER concettuale



### 2.4.1 Vincoli non rappresentabili nello schema Relazionale

Entità / Relazione	Vincolo
Molo	Lo stato di occupazione deve essere coerente con la sosta
Periodo di apertura	orario di chiusura deve essere postumo all'orario di apertura
Fattura	deve essere presente almeno una linea di consumo o una linea di sosta
Prenotazione	La previsione della partenza deve essere postuma alla previsione dell'arrivo
Prenotazione	Un molo non può essere prenotato nello stesso momento da due persone differenti e non può essere prenotato se c'è già una barca in sosta
Sosta	La data di partenza deve essere postuma alla data di arrivo
Sosta	Un molo non può essere utilizzato se è già occupato o è prenotato e un'imbarcazione non può occupare due moli contemporaneamente
Fattura	La data di pagamento di una fattura può essere solo postuma alla sua emissione
Consumo	L'inizio della lettura di un consumo è chiaramente precedente alla fine della stessa lettura

### 3 Progettazione logica

#### 3.1 Analisi delle ridondanze

Nello schema concettuale possiamo individuare stato di occupazione che è attributo di molo e deducibile dalla relazione sosta, la quale presenta l'attributo data di arrivo e data di partenza.

Le operazioni coinvolte sono la numero 3 e la numero 6 che presentano una frequenza di 40 volte al giorno e 30-40 volte ogni giorno.

Si procede alla valutazione del costo totale in termini di accessi nel caso di ridondanza e di assenza di essa.

Operazione	Frequenza
3. Controllo dei posti disponibili per una certa imbarcazione (Controllo delle dimensioni)	40 volte al giorno
6. Arrivo di un'imbarcazione nel marina	30-40 volte al giorno

Ci riferiremo per i nostri calcoli a dei volumi plausibili nella vita della base di dati. I moli non variano particolarmente di numero, le imbarcazioni vengono registrate al loro primo accesso e le soste aumentano ogni giorno. Mediamente un'imbarcazione sosta 2,7 volte in un molo.

Concetto	Tipo	Volume
Molo	ENTITÀ	250
Imbarcazione	ENTITÀ	5000
Sosta	ENTITÀ	13500
M_S	RELAZIONE	13500

#### operazione 3

Con ridondanza				Senza ridondanza			
Concetto	Costrutto	Accessi	Tipo	Concetto	Costrutto	Accessi	Tipo
Molo	E	250	L	Sosta	E	13500	L
				M_S	R	13500	L
				Molo	E	250	L

#### Con ridondanza

- Totale accessi (Solo in lettura):  $250 \xrightarrow{*40} 10000$

#### Senza ridondanza

- Totale accessi (Solo in lettura):  $250 + 13500 + 13500 = 27250 \xrightarrow{*40} 1090000$  Giornalieri

**operazione 6** Assumo che l'imbarcazione sia già presente perché l'esserci o non esserci non cambia il calcolo della ridondanza.

Operazioni principali
Cerco l'imbarcazione
Cerco i moli disponibili
Confronto le dimensioni dell'imbarcazione con i moli disponibili
Scrivo la sosta

Con ridondanza				Senza ridondanza			
Concetto	Costrutto	Accessi	Tipo	Concetto	Costrutto	Accessi	Tipo
Imbarcazione	E	1	L	Imbarcazione	E	1	L
Molo	E	250	L	Sosta	E	13500	L
Sosta	E	1	S	M_S	R	13500	L
Molo	E	1	S	Molo	E	250	L
				Sosta	E	1	S



### Con ridondanza

- Totale scritture:  $1 + 1 = 2 \xrightarrow{*2} 4$
- Totale letture:  $250 + 1 = 251$
- Totale accessi:  $4 + 251 = 255 \xrightarrow{*40} 102000$  Giornalieri

### Senza ridondanza

- Totale scritture:  $1 \xrightarrow{*2} 2$
- Totale letture:  $1 + 13500 + 13500 + 250 = 27251$
- Totale accessi:  $2 + 27251 = 27253 \xrightarrow{*40} 1090120$  Giornalieri

**In conclusione** Nell'**operazione 2** e nell'**operazione 6** risulta meno costosa in termini di accessi l'operazione con ridondanza. Notando un grosso divario tra le operazioni con ridondanza e le operazioni senza ridondanza, qui si sceglie di mantenere la ridondanza diminuendo il numero di accessi necessari.

#### 3.1.1 altre ridondanze

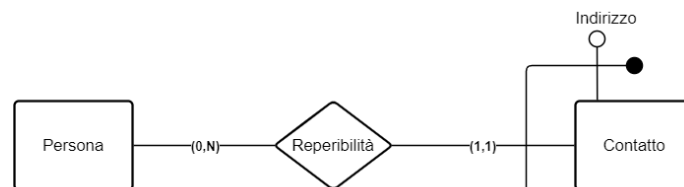
Un'altra ridondanza è Quantità di soste in **Cliente occasionale** è deducibile da Cliente possiede imbarcazione che sosta in un molo andando a vedere quante soste hanno effettuato le sue imbarcazioni;

In seguito all'analisi sugli accessi abbiamo stimato che risulti essere preferibile mantenere la ridondanza relativamente ad eventuali operazioni di lettura e scrittura di questi valori, pertanto si è deciso di mantenere la ridondanza.

### 3.2 Eliminazione degli attributi multi-valore

Contatto in Persona è multi-valore.

Abbiamo quindi reificato contatto in una relazione binaria come segue.



### 3.3 Eliminazione delle generalizzazioni

#### Cliente

Si accorpano Cliente abituale e cliente occasionale a Cliente. Dato che la quantità di soste è un attributo utile per entrambe le tipologie di cliente. Lo sconto personale sarà invece impostato ad un valore che indica la percentuale di sconto se il cliente è abituale e a NULL se il cliente è occasionale.

#### Persona

Nelle generalizzazioni di persona, per evitare inutili valori NULL in un eventuale accorpamento **Persona** viene mantenuta come entità. Questa entità conterrà tutti i dati relativi alla persona e avrà due entità deboli collegate ovvero **Addetto** e **Cliente**.

Persona avrà da 0 a 1 **Cliente** e da 0 a 1 **Addetto**. Chiaramente ogni **Cliente** e **Addetto** avrà una e una sola persona legata. In **Addetto** e **Cliente** posizioneremo i dati relativi a queste due entità come da schema concettuale.

### 3.4 Modifiche, aggiunte e chiarimenti alle chiavi

Tutte le chiavi primarie sono definite utilizzando le chiavi della progettazione concettuale, tranne nei seguenti casi.

**Imbarcazione** Ad **Imbarcazione** oltre alla sua chiave primaria *Codice internazionale* viene aggiunto un id autoincrementante utilizzato come chiave<sup>10</sup>. Questo è stato scelto per poter utilizzare il constraint relativo alla sovrapposizione delle soste. Infatti esso funziona solo con attributi interi e non con Varchar<sup>11</sup>.

**Cliente** A **Cliente** oltre alla sua chiave primaria *Codice fiscale* viene aggiunto un id autoincrementante utilizzato come chiave<sup>12</sup>. Questo è stato scelto per poter utilizzare il constraint relativo alla sovrapposizione delle prenotazioni. Infatti esso funziona solo con attributi interi e non con Varchar.

**Addetto** Addetto concettualmente eredita la chiave primaria di **Persona** ma come da progettazione concettuale mantiene la chiave su *Servizio gestito*, *Data inizio contratto*. Come chiave primaria si sceglie quindi di utilizzare il *codice fiscale* derivante da **persona** perché più comodo e leggero.

**Periodo di apertura** Per ridurre lo spazio utilizzato nella tabella che rappresenta la relazione *Apertura*, a **Periodo di apertura** abbiamo aggiunto una chiave primaria tramite id autoincrementante.

**Sosta** Per semplificare il lavoro con le prenotazioni trasformate in **Sosta** a sosta viene aggiunto un id autoincrementante come chiave. Questo id verrà utilizzato da **Prenotazione** per definire se la prenotazione è stata trasformata in sosta.

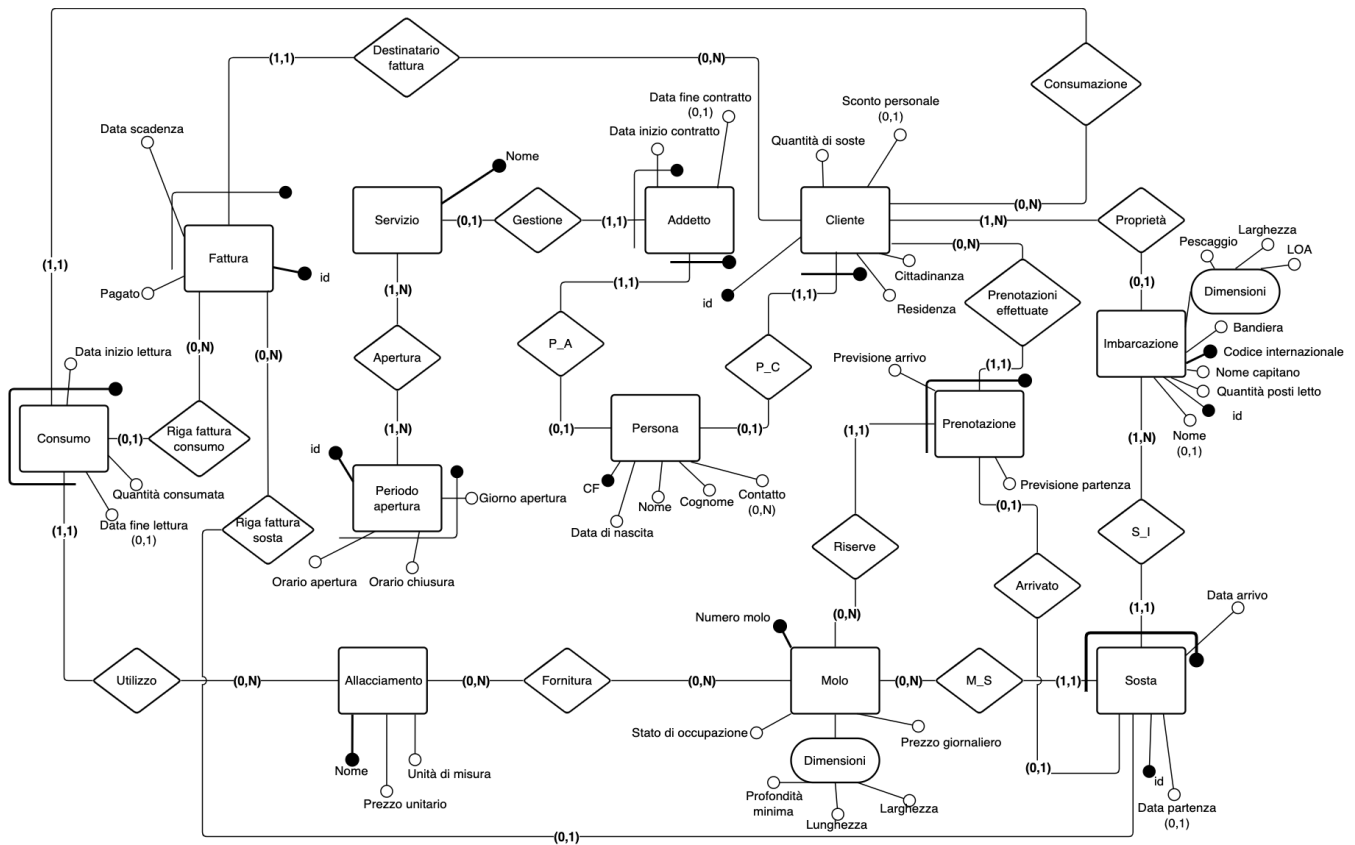
---

<sup>10</sup>Chiave unica

<sup>11</sup>A meno di installare il pacchetto aggiuntivo di Postgres che lo permetta ma per lo scopo di questo progetto abbiamo preferito evitare per questioni di compatibilità.

<sup>12</sup>Chiave unica

### 3.5 Schema concettuale ristrutturato - Schema logico



### 3.6 Descrizione schema relazionale

Per questione di compatibilità con il DBMS alcuni nomi di attributi entità e relazioni sono stati normalizzati, utilizzando il camelCase, togliendo gli accenti, accorciando i nomi molto lunghi e con altre piccole accortezze.

La **chiave primaria** è la prima delle chiavi indicate dalla **sottolineatura**, ovvero la chiave indicata in **grassetto**.

*Imbarcazione*(**MMSI**, id, cliente, bandiera, nomeCapitano, nPostiLetto, nome, pescaggio, larghezza, LOA);

*Molo* (id, occupato, profonditaMinima, larghezza, lunghezza, prezzoGiorno);

*Servizio* (**nome**);

*Addetto* (**persona**, servizio, inizioContratto, fineContratto);

*Cliente* (**persona**, id, cittadinanza, residenza, quantitaSoste, scontoPersonale);

*Persona* (**CF**, dataNascita, nome, cognome);

*Prenotazione* (**cliente**, **prevArrivo**, **molo**, prevPartenza, sosta);

*Sosta* (**imbarcazione**, **molo**, **arrivo**, id, partenza, fattura);

*Allacciamento* (**nome**, prezzoUnitario, unitaMisura);

*Fornitura* (**allacciamento**, **molo**);

*PeriodoApertura* (id, giorno, apertura, chiusura);

*AperturaServizio* (**servizio**, **periodoApertura**);

*Consumo*(**cliente**, **allacciamento**, **inizio**, fine, quantita, fattura);

*Fattura*(id, cliente, scadenza, pagato);

## 3.7 Vincoli di integrità referenziali

**Imbarcazione**.cliente -> *Cliente.persona*  
**Addetto**.persona -> *Persona.CF*  
**Cliente**.persona -> *Persona.CF*  
**Prenotazione**.cliente -> *Cliente.id*  
**Prenotazione**.molo -> *Molo.id*  
**Prenotazione**.sosta -> *Sosta.id*  
**Sosta**.imbarcazione -> *Imbarcazione.id*  
**Sosta**.molo -> *Molo.id*  
**Sosta**.fattura -> *Fattura.id*  
**Fornitura**.allacciamento -> *Allacciamento.nome*  
**Fornitura**.molo -> *Molo.id*  
**AperturaServizio**.servizio -> *Servizio.nome*  
**AperturaServizio**.periodoApertura -> *PeriodoApertura.id*  
**Consumo**.cliente -> *Cliente.persona*  
**Consumo**.allacciamento -> *Allacciamento.nome*  
**Consumo**.fattura -> *Fattura.id*  
**Fattura**.cliente -> *Cliente.persona*

## 3.8 Check e constraint

### 3.8.1 Vincoli non gestiti

Non è stato purtroppo possibile mantenere fede a tutti i vincoli posti nella progettazione concettuale, e relativamente ad essi viene lasciato al programmatore che utilizzerà la base dati l'onere di mantenerla consistente.

**Fattura** Non viene gestito il vincolo relativo a fattura, per via di una dipendenza circolare tra la generazione della fattura e la creazione delle sue righe. Si considera quindi la fattura con zero linee come "fattura da riempire".

**Molo** Per gestire la consistenza di **Molo** si possono utilizzare dei trigger che la controllino ad ogni variazione dei dati, si è deciso però di lasciare l'onere al programmatore. In caso di riscontrate inconsistenze sarà comunque sempre possibile fare affidamento sulle soste le quali saranno sempre consistenti.

**Sosta** Un altro vincolo purtroppo non gestito è la palese impossibilità di inserire un'imbarcazione in un molo più piccolo di essa. Questo vincolo non è gestibile, se non con soluzioni esageratamente complesse. Si può però comunque notare che questo vincolo sarà rispettato fisicamente nella realtà.

**Prenotazione** In prenotazione, non sarà possibile controllare nelle soste o nei moli se al momento della prenotazione sono già occupati. Questo vincolo è però in buona parte rispettato dato che nelle soste si inseriscono solo date attuali o postume e non date relative al futuro, mentre in Prenotazioni si inseriscono date future. Si nota quindi come i concetti affrontati dalle due entità non collidano spesso. Si lascia in ogni caso al programmatore l'onere di mantenere il vincolo.

### 3.8.2 Vincoli gestiti

**PeriodoApertura** In **PeriodoApertura** è attivato un check che controlla che l'orario di chiusura sia sempre postumo all'orario di apertura.

**Prenotazione e Sosta** In **Prenotazione** ed in **Sosta** viene eseguito un check simile a quello di **PeriodoApertura**. In queste due entità viene quindi controllato che il momento della partenza sia sempre postumo al momento dell'arrivo, questo anche quando la partenza è non definita. Per fare questo si è quindi deciso di settare la partenza con valore di default ad "infinito" in sosta.

In queste due entità vengono inoltre fatti due controlli relativi alla sovrapposizione di soste e prenotazioni.

Se un **Molo** è già occupato in un determinato periodo esso non sarà occupabile, e se un **Imbarcazione** si trova già in un **Molo** non potrà sdoppiarsi ed essere sostante in più moli contemporaneamente.

Come precedentemente accennato per implementare questo constraint sono state fatte delle modifiche alle chiavi di imbarcazione e di cliente, aggiungendo una chiave intera in modo da poter utilizzare i GIST<sup>13</sup> nella versione standard di Postgres<sup>14</sup>.

**fattura e consumo** In fattura e consumo viene implementato un semplice check per controllare che le date di inizio, di fine, di emissione e di pagamento siano coerenti relativamente alle loro entità

## 4 Query

### 4.1 Consumi del marina nel mese indicato

```
1 select allacciamento ,
2       sum(quantita) as quantita_consumata ,
3       a.unitam misura , a.prezzounitario ,
4       sum(quantita) * a.prezzounitario as totale_bollette
5 from consumo
6 join allacciamento a on consumo.allacciamento = a.nome
7 where inizio between '05-01-2020' and '05-31-2025'
8 group by allacciamento , unitam misura , prezzounitario ;
```

Ritorna i consumi del marina nel mese indicato, dando un'idea delle bollette che dovrà pagare.

allacciamento	quantita_consumata	unitam misura	prezzounitario	totale_bollette
Acqua	294	l	0.02	5.88
Gas	220	Mc	0.32	70.4
Elettricità	393	KW/h	0.21	82.53

### 4.2 Clienti con più di 2 di prenotazioni che iniziano nel quinquennio 2020->2025

```
1 select count(cliente) as conteggio , p.nome , p.cognome , c.persona
2 from prenotazione
3 join cliente c on prenotazione.cliente = c.id
4 join persona p on c.persona = p.cf
5 where preavviso between '01-01-2020' and '12-31-2025'
6 group by c.persona , p.nome , p.cognome
7 having count(cliente) >= 2
8 order by conteggio desc ;
```

Ritorna i clienti che hanno fatto almeno due prenotazioni nel quinquennio presente. Questa è una tipica ricerca fatta per scopi di marketing, volendo comprendere meglio il ritorno delle persone nel marina.

<sup>13</sup>Generalized Search Tree, ovvero un albero di ricerca molto simile al B-Tree

<sup>14</sup>DBMS scelto per il progetto presente

conteggio	nome	cognome	persona
4	Mariapia	Marino	MRNMRP81S70G224W
3	Ortensia	Bernardi	BRNRNS61D64G224W
2	Edipo	Battaglia	BTTDPE87L24G224C
2	Eufemia	Fiore	FRIFME76P50G224A

### 4.3 Conteggio soste di ogni imbarcazione

```

1 select count(imbarcazione) as qt_soste, imbarcazione.nome, mmsi, p.nome as nome_proprietario, p.cognome as cognome_proprietario
2 from sosta
3 join imbarcazione on sosta.imbarcazione = imbarcazione.id
4 join cliente c on imbarcazione.cliente = c.persona
5 join persona p on c.persona = p.cf
6 group by sosta.imbarcazione, imbarcazione.nome, mmsi, p.nome, p.cognome
7 order by qt_soste desc;

```

Ritorna quante volte ogni imbarcazione ha sostato nel marina per stilare una classifica.

qt_soste	nome	mmsi	nome_proprietario	cognome_proprietario
3	ALA	5217555	Eufemia	Fiore
1	NORDLICHT	8745943	Ortensia	Bernardi
1	ERIDANO	7945144	Fiorenzo	Parisi
1	ESPALMADOR JET	9809631	Fiorenzo	Parisi
1	FOX INSPECTOR	9921207	Teseo	Amato
1	MAZZORBO	9212553	Eufemia	Fiore
1	LA TRINITE	9831581	Silvano	Moretti
1	BROCELTANDE	9099391	Costanzo	Russo

Il risultato contiene tante righe quante le imbarcazioni, è stato quindi troncato per limitare il consumo di spazio.

### 4.4 Controllo dei posti disponibili per una certa imbarcazione

```

1 select distinct molo.id as id_molo, molo.prezzogiorno,
2 molo.profonditaminima, molo.larghezza, molo.lunghezza, molo.occupato
3 from molo, imbarcazione
4 where occupato=false
5 and molo.larghezza > imbarcazione.larghezza
6 and molo.profonditaminima > imbarcazione.pescaggio
7 and molo.lunghezza > imbarcazione.loa
8 and imbarcazione.mmsi = '8836340'
9 order by prezzogiorno
10 limit 5;

```

Relativamente ad un'imbarcazione già inserita ci dice quali sono i 5 migliori moli in cui può sostare.

id_molo	prezzogiorno	profonditamina	larghezza	lunghezza	occupato
6	30	10	6	14	no
15	30	10	6	14	no
16	30	10	6	14	no
17	30	10	6	14	no
18	30	10	6	14	no

## 4.5 Stampa fattura di un cliente

```

1 with intestazione_fattura as(
2     select fattura.*, p.nome, p.cognome from fattura
3     join cliente c on c.persona = fattura.cliente
4     join persona p on c.persona = p.cf
5     where cliente = 'GLLGNN81A54G224W'),
6
7 spese as(
8     select sosta.fattura ,
9           i.cliente ,
10          'sosta' as tipo ,
11          ROUND((tstzrange_subdiff(sosta.partenza , sosta.arrivo)
12                /86400.0 * m.prezzogiorno)::numeric,2) as prezzo
13     from sosta
14     join imbarcazione i on i.id = sosta.imbarcazione
15     join molo m on sosta.molo = m.id
16     where partenza != 'infinity'
17
18     union all
19     select consumo.fattura ,
20           consumo.cliente ,
21          'consumo' as tipo ,
22          ROUND((consumo.quantita * a2.prezzounitario)::numeric,2) as prezzo
23     from consumo
24     join fornitura f on consumo.allacciamento = f.allacciamento
25     join allacciamento a2 on consumo.allacciamento = a2.nome
26 )
27
28 select distinct id, scadenza, pagato, nome, cognome,
29               intestazione_fattura.cliente, tipo, prezzo
30 from spese, intestazione_fattura
31 where spese.fattura = intestazione_fattura.id;
```

Stampa le fatture a carico di un cliente senza contare gli sconti dei clienti occasionali.

id	scadenza	pagato	nome	cognome	cliente	tipo	prezzo
5	2022-07-02		Giovanna	Galli	GLLGNN81A54G224W	consumo	0.60
5	2022-07-02		Giovanna	Galli	GLLGNN81A54G224W	consumo	9.45
5	2022-07-02		Giovanna	Galli	GLLGNN81A54G224W	sosta	84.90
5	2022-07-02		Giovanna	Galli	GLLGNN81A54G224W	sosta	85.44
5	2022-07-02		Giovanna	Galli	GLLGNN81A54G224W	sosta	289.91

## 5 Indicizzazione

```
1 CREATE INDEX idx_molo ON Molo USING btree(lunghezza , larghezza , profonditamina);
```

Viene creato un indice su molo usando le dimensioni per migliorarne la velocità d'accesso nel momento di ricerca di un molo di dimensioni adeguate.

Sono stati poi, come precedentemente descritto al punto 3.8.2 , utilizzati dei search tree per utilizzare al meglio i constraint relativi alle date ed orari di **prenotazioni** e **soste**.

## 6 Esecuzione programma C++

Verificare all'interno del sorgente cpp che i parametri di connessione al database corrispondano (sono quelli di default, quindi se si sono apportate modifiche sarà necessario cambiarli).

```
1 //Parametri database
2 #define PG_HOST "127.0.0.1" // Indirizzo Host
3 #define PG_USER "postgres" // Nome utente
4 #define PG_DB "progetto_amassare_fzontaro" //Nome del database
5 #define PG_PASS "*****" //Password
6 #define PG_PORT "5432"
7
```

Per prima cosa è necessario compilare il programma utilizzando il comando:

```
1 $ compilatore_cpp codice.cpp -L dependencies/lib -lpq -o codice
2
```

Per eseguirlo (su sistemi UNIX like), basta utilizzare il comando:

```
1 $ ./query
2
```

All'avvio del programma, verrà mostrata all'utente la lista delle query, potrà digitare un numero da 1 a 5 per eseguirle. Se la query selezionata utilizza dei parametri, verranno richiesti in input. Tutte le query sono contenute all'interno di un unico file cpp.