



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

DIPARTIMENTO DI MATEMATICA

LAUREA TRIENNALE IN INFORMATICA

PROGETTO DI BASI DI DATI

---

# Basi di Dati di un Sistema Informativo per la Gestione di Crociere Organizzate

---

Ghiraldin Mirco, Stevanin Michele

# 1 Abstract

---

In questo elaborato viene presentato lo sviluppo di una base di dati relazionale pensata per **gestire le attività delle compagnie di crociere**. L'obiettivo è costruire un sistema informativo in grado di organizzare in modo chiaro ed efficiente tutte le informazioni legate alla gestione delle navi, del personale e delle attività a bordo. Il progetto copre l'intero ciclo di vita di una crociera: si parte dall'identificazione delle compagnie organizzatrici (tramite partita IVA), per arrivare alla gestione delle singole crociere, ciascuna con il proprio itinerario, i porti di partenza e arrivo, le date, le tappe intermedie e altri dettagli tecnici. A bordo interagiscono diverse figure: passeggeri, membri dell'equipaggio e animatori, rappresentati nel modello tramite un sistema di generalizzazioni e specializzazioni che permette di descrivere con flessibilità i diversi ruoli. Il modello affronta anche aspetti legati alle prestazioni del sistema, come ad esempio l'analisi della ridondanza relativa al numero di passeggeri prenotati. Il progetto è stato sviluppato seguendo un approccio modulare, ispirato alle metodologie viste durante il corso, con l'obiettivo di garantire integrità, coerenza e adattabilità in contesti reali di gestione delle crociere.

## 2 Analisi dei Requisiti

---

### Compagnie Marittime

Ogni compagnia è identificata da una partita IVA univoca e include le seguenti informazioni:

- **Partita IVA** (PK)
- **Nome**
- **Sede**
- **Recapito**

Le compagnie possiedono una o più crociere.

### Crociere

Ogni crociera è identificata dal codice IMO e registra:

- **Codice IMO** (PK)
- **Nome della nave**
- **Porto di partenza**
- **Porto di arrivo**
- **Data e ora di partenza**
- **Durata** (in giorni)
- **Numero minimo di membri dell'equipaggio**
- **Numero massimo di passeggeri**
- **Tipologia** (es. mediterranea, fluviale, transatlantica)

Ogni crociera:

- È di proprietà di una **compagnia**
- Parte da un **porto** e arriva in un altro e può prevedere **tappe** intermedie
- Ha a bordo **ospiti** e **membri dell'equipaggio**

- Include **eventi** gestiti da **animatori**

## Porti

Ogni porto è identificato dalla città in cui si trova:

- **Nome della città** (PK)
- **Numero massimo di navi**

Una crociera può prevedere soste in più porti (tappe).

## Tappe

Le tappe rappresentano le fermate della crociera nei porti e includono:

- **Data e ora di arrivo**
- **Data e ora di partenza**

Ogni tappa è collegata a un porto e a una specifica crociera.

## Persone

Tutti gli individui a bordo (sia personale che ospiti) sono entità del tipo "Persona", identificata tramite:

- **Codice Fiscale** (PK)
- **Nome**
- **Cognome**
- **Sesso**

Una persona può essere un ospite o un membro dell'equipaggio.

## Ospiti (Passeggeri)

Gli ospiti sono persone che hanno prenotato una crociera:

- **Codice Fiscale** (PK, FK da Persona)
- **Costo della crociera**
- **Crociera** (IMO) (FK)

## Equipaggio

L'equipaggio rappresenta il personale operativo a bordo. Ogni membro è associato a:

- **Codice Fiscale** (PK, FK da Persona)
- **ID equipaggio**
- **Stipendio**
- **Anni di servizio**
- **Lingue parlate**

Un membro dell'equipaggio può essere specializzato in animatore.

## Animatori

Gli animatori sono una specializzazione dell'equipaggio e dispongono di una o più abilità:

- **Codice Fiscale** (PK, FK da Equipaggio)
- **Abilità specifiche** (es. ballo, canto, giochi)

## Eventi

Durante le crociere vengono organizzati eventi, ciascuno identificato da:

- **Nome dell'evento** (PK)
- **Tipo di evento** (PK)
- **Età consigliata**
- **Numero minimo di animatori**
- **Numero consigliato di partecipanti**

Ogni evento è gestito da uno o più animatori e può essere frequentato da più ospiti.

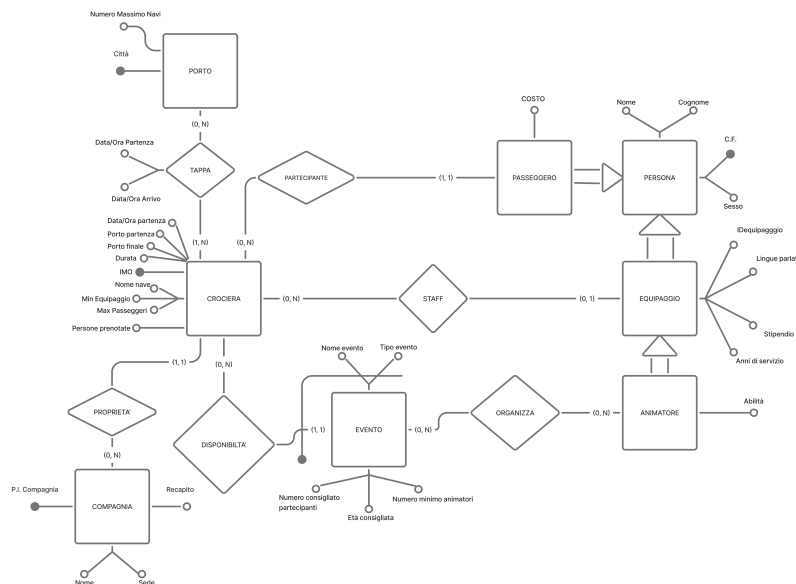


Grafico1

### 3 Progettazione Concettuale

La Figura 1 riporta il diagramma Entità–Relazione (E–R) che riassume i requisiti descritti nella sezione 2. Nel modello la progettazione concettuale si è basata su una visione gerarchica del personale di bordo, dove la generalizzazione rappresenta il fatto che tutti i membri dell’equipaggio e gli ospiti condividono un insieme comune di attributi base, ma possono essere specializzati in ruoli specifici. Ogni crociera è organizzata da una compagnia marittima e ha una propria identità definita dal codice IMO. Le crociere sono associate a uno o più porti attraverso una relazione Tappa, che rappresenta gli scali effettuati durante il viaggio. Ogni tappa è caratterizzata da una data e ora di arrivo e partenza, ed è sempre riferita a un porto e a una crociera specifici. Le persone imbarcate sono suddivise in Equipaggio mediante una generalizzazione parziale. Il personale di bordo può ricoprire ruoli specializzati: ad esempio gli Animatori, sottoinsieme dell’equipaggio, sono incaricati dell’organizzazione e conduzione degli eventi. Gli Eventi sono attività previste a bordo della crociera. Ogni evento può essere gestito da uno o più animatori (tramite la relazione Gestione). Gli eventi sono identificati in modo univoco dalla combinazione tra nome e tipologia e presentano attributi come età consigliata e limiti sulla partecipazione. La generalizzazione tra Equipaggio e la specializzazione Animatore è modellata come parziale: non tutti i membri dell’equipaggio sono necessariamente animatori. Tabella 1 riassume le entità e relazioni individuate nella progettazione concettuale, riportando per ciascuna gli attributi rilevanti e l’identificatore scelto. Per le entità derivate da generalizzazione viene anche specificato il tipo di specializzazione utilizzato. Il presente schema E-R non permette di rappresentare direttamente il seguente vincolo: se una persona  $pe$  è animatore in due crociere  $cr'$  e  $cr''$ , allora  $cr'$  e  $cr''$  appartengono alla stessa Classe:

$$(cr', pe) \in Animatore \wedge (cr'', pe) \in Animatore \Rightarrow (cr'.Classe = cr''.Classe)$$

### 4 Progettazione Logica

In questa sezione viene illustrato il processo di “traduzione” dello schema concettuale in uno schema logico, con l’obiettivo di rappresentare i dati in modo preciso ed efficiente. Il primo passo consiste nell’analizzare le eventuali ridondanze nel modello, al fine di ottimizzare la struttura complessiva. Successivamente, si procede con l’eliminazione delle due generalizzazioni. Infine, viene presentato il diagramma ristrutturato, con una descrizione delle modifiche apportate.

Entità	Descrizione	Attributi	Identificatore
Crociera	Mezzo di trasporto marittimo	Imo, Nome nave, Porto partenza, Porto finale, Data/ora partenza, Durata, Min equipaggio, Max passeggeri, Persone prenotate	Imo
Porto	Località in cui la crociera può attraccare	Città, Numero massimo navi	Città
Persona	Persone partecipanti alle crociere	Codice fiscale (CF), Nome, Cognome, Sesso	CF

Entità	Descrizione	Attributi	Identificatore
Passeggero	Passeggeri ospiti della crociera	Codice fiscale (CF), Costo, Imo crociera	CF
Equipaggio	Persone membri dell'equipaggio	Codice fiscale (CF), Id equipaggio, Lingue parlate, Stipendio, Anni di servizio, Imo crociera	CF
Animatore	Membri dell'equipaggio che si occupano degli eventi	Codice fiscale (CF), Abilità	CF
Compagnia	Società che possiedono le navi da crociera	Partita IVA (P.I. compagnia), Nome, Sede, Recapito	P.I. compagnia
Evento	Eventi ricreativi svolti a bordo delle crociere	Nome evento, Tipo evento, Età consigliata, Numero minimo animatori, Numero consigliato partecipanti, Imo crociera	Nome evento, Tipo evento, Imo crociera

Tabella 1 (Entità)

Relazione	Descrizione	Componenti	Attributi
Tappa	Tappe portuali fatte da una nave	Porto, Crociera	Data/ora arrivo, Data/ora partenza
Partecipante	Assegnazione di una persona (ospite) a una crociera	Crociera, Passeggero (ospite)	–
Staff	Membri dell'equipaggio assegnati a una crociera	Crociera, Equipaggio	–
Proprietà	Navi da crociera possedute da una compagnia	Crociera, Compagnia	–
Disponibilità	Eventi programmati per una crociera	Crociera, Evento	–
Organizza	Assegnazione di uno o più animatori a un evento	Evento, Animatore	–

Tabella 2 (Relazioni)

## 4.1 Analisi delle ridondanze

L'attributo `Persone_Prenotate` in `CROCIERA`, che memorizza il numero di persone prenotate in quella crociera presenta una ridondanza. Questo valore può essere infatti ottenuto contando il numero di passeggeri attivi per quella crociera tramite la relazione `PARTECIPANTE`. Questo attributo viene modificato ogni volta che si aggiunge una nuova persona alla crociera (circa 400 persone nuove al giorno tra tutte le crociere) e viene visualizzato ogni ora del giorno per monitorare il numero di posti rimanenti. Questo si riassume nelle seguenti due operazioni:

- Operazione 1 (400 al giorno): memorizza una nuova prenotazione in relativa crociera.
- Operazione 2 (24 al giorno): visualizza il numero di prenotazioni attuali in una crociera.

Assumendo i seguenti volumi nella base di dati:

Concetto	Costrutto	Volume
CROCIERA	E	30
PARTECIPANTE	R	150000
PASSEGGERO	E	150000

La seguente analisi serve per stabilire se sia utile o meno tenere l'attributo ridondante `Persone_Prenotate` in `CROCIERA`.

### CON RIDONDANZA

Analizziamo prima il costo totale con ridondanza.

- Operazione 1:

Concetto	Costrutto	Accessi	Tipo	Ripetizioni
PASSEGGERO	E	1	S	× 400
PARTECIPANTE	R	1	S	× 400
CROCIERA	E	1	L	× 400
CROCIERA	E	1	S	× 400

- Operazione 2:

Concetto	Costrutto	Accessi	Tipo	Ripetizioni
Crociera	E	1	L	× 24

Assumendo costo doppio per gli accessi in scrittura:

$$\text{Costo Totale} = 400 * 3 * 2 + 400 + 24 = 2824$$

SENZA RIDONDANZA

Analizziamo il costo totale senza ridondanza.

- Operazione 1:

Concetto	Costrutto	Accessi	Tipo	Ripetizioni
PASSEGGERO	E	1	S	× 400
PARTECIPANTE	R	1	S	× 400

- Operazione 2 (con circa  $150000/30 = 5000$  passeggeri al giorno):

Concetto	Costrutto	Accessi	Tipo	Ripetizioni
CROCIERA	E	1	L	× 24
PARTECIPANTE	R	5000	L	× 24

Assumendo costo doppio per gli accessi in scrittura:

Costo Totale =  $400 * 2 * 2 + 5001 * 24 = 121624$

L’analisi suggerisce quindi di tenere l’attributo ridondante, ottimizzando così il numero di accessi.

4.2 Eliminazioni delle Generalizzazioni

Le generalizzazioni descritte in Sezione 3 vengono eliminate attraverso una ristrutturazione dello schema concettuale, con l’obiettivo di semplificare la successiva implementazione del modello relazionale e ridurre la presenza di valori nulli. Le modifiche vengono applicate come segue:

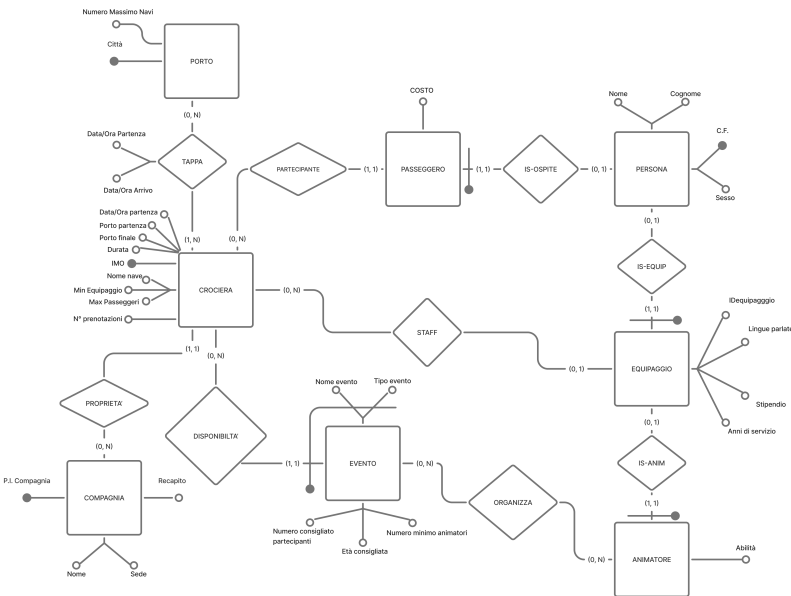


Grafico2



**PERSONA.** La generalizzazione parziale PERSONA viene sostituita con le relazioni **IS-OSPITE** e **IS-EQUIP** (vedi Figura 2), che collega alcuni individui alla relativa specializzazione: EQUIPAGGIO o PASSEGGERO. Tale scelta consente di evitare la presenza di valori nulli che si verificherebbero mantenendo un'unica entità PERSONA con tutti gli attributi specifici delle due categorie (ad esempio, Stipendio, Anni\_Di\_Servizio, Lingue\_Parlare per EQUIPAGGIO, Costo per PASSEGGERO). Separando le informazioni tramite relazioni specializzate, si garantisce che ciascuna entità contenga soltanto gli attributi rilevanti per il proprio ruolo. In linea con la metodologia adottata a lezione, l'identificatore di EQUIPAGGIO e di PASSEGGERO coincide con quello della rispettiva PERSONA. Poiché la generalizzazione è parziale da entrambe le parti (non tutte le persone sono membri dell'equipaggio né passeggeri), l'eliminazione dell'entità padre PERSONA non è corretta. Essa viene mantenuta per rappresentare tutte le informazioni comuni (come Nome, Cognome, Data\_Nascita), mentre le informazioni specifiche sono distribuite nelle entità figlie.

**EQUIPAGGIO.** Analogamente, la generalizzazione parziale EQUIPAGGIO viene sostituita con **IS-ANIM** (vedi Figura 2). Anche in questo caso, la ristrutturazione consente di evitare valori nulli, in quanto non tutti i membri dell'equipaggio svolgono il ruolo di Animatore. Essendo la generalizzazione parziale, l'eliminazione dell'entità padre EQUIPAGGIO risulterebbe nuovamente scorretta.

Il diagramma E-R ristrutturato, riportato in **Figura 2** riflette tali modifiche rispetto alla versione originale presentata in **Figura 1**.

## 4.3 Schema Relazionale

Lo schema ristrutturato in Figura 2 contiene solamente costrutti mappabili in corrispettivi dello schema relazionale, detto anche schema logico. Lo schema logico è rappresentato a seguire, dove l'asterisco dopo il nome degli attributi indica quelli che ammettono valori nulli.

- **Crociera**(IMO, Nome\_Nave, Min\_Equipaggio, Max\_Passeggeri, Num\_Prenotazioni, Porto\_Partenza\*, Porto\_Finale\*, Data\_Ora\_Partenza\*, Durata\*, PI\_Compagnia)
  - Crociera.PI\_Compagnia -> Compagnia.PI
  - Crociera.Porto\_Partenza -> Porto.Città
  - Crociera.Porto\_Finale -> Porto.Città
- **Porto**(Città, Numero\_Massimo\_Navi\*)
- **Tappa**(IMO, Città, Data\_Ora\_Partenza, Data\_Ora\_Arrivo)
  - Tappa.IMO -> Crociera.IMO
  - Tappa.Città -> Porto.Città
- **Compagnia**(PI, Nome, Sede, Recapito\_Telefonico\*)
- **Persona**(CF, Nome, Cognome, Sesso)
- **Ospite**(CF, Costo\*, IMO\_Crociera)
  - Ospite.CF -> Persona.CF
  - IMO\_Crociera -> Crociera.IMO
- **Equipaggio**(CF, IDequipaggio, Lingue\_Parlare\*, Stipendio\*, Anni\_Servizio\*, IMO\_Crociera\*)
  - Equipaggio.CF -> Persona.CF
  - IMO\_Crociera -> Crociera.IMO
- **Animatore**(CF, Abilità\*)
  - Animatore.CF -> Equipaggio.CF
- **Evento**(Nome, Tipologia, IMO\_Crociera, Num\_Cons\_Partecipanti\*, Num\_Min\_Animatori\*, Età\_Consigliata\*)
  - Evento.IMO\_Crociera -> Crociera.IMO

- **ORGANIZZA**(CF Animatore, Nome Evento, Tipologia Evento, IMO Crociera Evento)
  - Organizza.CF\_Animatore -> Animatore.CF
  - Organizza.Nome\_Evento -> Evento.Nome
  - Organizza.Tipologia\_evento -> Evento.Tipologia
  - Organizza.IMO\_Crociera\_Evento -> Evento.IMO\_Crociera

## 5 Implementazione in PostgreSQL e Definizione delle Query

Il file Crociere.sql contiene il codice SQL necessario per la creazione e il popolamento delle tabelle del database. Questo file include inoltre una serie di query per l'estrazione dei dati e un indice creato specificamente per migliorare le prestazioni di una di queste interrogazioni.

### 5.1 Definizione delle Query

Di seguito vengono presentate e descritte le query con i relativi output generati e viene motivato l'utilizzo dell'indice proposto.

**Query 1** Trovare le crociere che toccano più di 3 porti diversi e indicarne la città di partenza, di arrivo e il numero di tappe.

```
SELECT C.IMO, C.Nome_Nave, C.Porto_Partenza, C.Porto_Finale, COUNT(DISTINCT
T.Città) AS Numero_Tappe
FROM Crociera C
JOIN Tappa T ON C.IMO = T.IMO
GROUP BY C.IMO, C.Nome_Nave, C.Porto_Partenza, C.Porto_Finale
HAVING COUNT(DISTINCT T.Città) > 3
ORDER BY Numero_Tappe DESC;
```

Estratto dell'output:

**QUERY 2** Trovare per ogni città (porto), quante crociere partono da lì e la media del numero di prenotazioni.

```
SELECT p.Città, COUNT(c.IMO) AS Numero_Crociere, AVG(c.Num_Prenotazioni) AS
Media_Prenotazioni
FROM Porto p
JOIN Crociera c ON p.Città = c.Porto_Partenza
GROUP BY p.Città
ORDER BY Numero_Crociere DESC;
```

Estratto dell'output:

**Query 3** Trovare le crociere che hanno una media del costo dei biglietti superiore a 500 euro

```
SELECT O.IMO_Crociera, C.Nome_Nave, AVG(O.Costo) AS Media_Costo
FROM Ospite O
JOIN Crociera C ON O.IMO_Crociera = C.IMO
GROUP BY O.IMO_Crociera, C.Nome_Nave
HAVING AVG(O.Costo) > 500
ORDER BY Media_Costo DESC;
```

Estratto dell'output:

**Query 4** Visualizzare gli animatori con più eventi organizzati, mostrando solo quelli con più di 2 eventi

```
SELECT A.CF, P.Nome, P.Cognome, COUNT(*) AS Num_Eventi
FROM Animatore A
JOIN Persona P ON A.CF = P.CF
JOIN ORGANIZZA O ON A.CF = O.CF_Animatore
GROUP BY A.CF, P.Nome, P.Cognome
HAVING COUNT(*) > 2
ORDER BY Num_Eventi DESC;
```

Estratto dell'output:

**Query 5** Trovare, per ogni crociera, la percentuale di occupazione rispetto alla capacità massima (Num\_Prenotazioni / Max\_Passeggeri)

```
SELECT IMO, Nome_Nave,
       ROUND((Num_Prenotazioni * 100.0) / Max_Passeggeri, 2) AS
Percentuale_Occupazione
FROM Crociera
WHERE Max_Passeggeri > 0
ORDER BY Percentuale_Occupazione DESC;
```

Estratto dell'output:

## 5.2 Creazione degli indici

Si suppone di voler ottimizzare la Query 2, per la quale occorre considerare:

1. Condizione di Join: p.Città = c.Porto\_Partenza
2. Group by sulla colonna p.Città e aggregazione COUNT(c.IMO), AVG(c.Num\_Prenotazioni)
3. Ordinamento sul numero di crociere in partenza (alias Numero\_Crociere)

Per il punto 1, è opportuno creare un indice sulla colonna Porto\_Partenza della tabella Crociera, poiché questa viene utilizzata in un join basato sull'uguaglianza con Porto(Città) e successivamente per il raggruppamento dei risultati. Inoltre, Porto\_Partenza non è una chiave primaria e non è automaticamente indicizzata dal sistema.

```
CREATE INDEX idx_porto_partenza ON Crociera (Porto_Partenza);
```

Questo indice consente:

1. Un accesso rapido alle tuple della tabella Crociera che partono da una certa città
2. Un'efficiente esecuzione del join con la tabella Porto usando l'equivalenza sulla città,
3. Una scansione ordinata utile all'ottimizzazione del GROUP BY, permettendo di aggregare le tuple con lo stesso valore di **Porto\_Partenza** con complessità inferiore.

Per quanto riguarda il punto 3, l'ordinamento viene effettuato su un attributo derivato (**Numero\_Crociere, alias di COUNT(c.IMO)**), quindi non può essere direttamente indicizzato. Tuttavia, un ordinamento efficace viene facilitato dalla pre-aggregazione ottimizzata tramite l'indice sul campo di raggruppamento (**Porto\_Partenza**).

Nota: la colonna Città in Porto è chiave primaria, quindi PostgreSQL crea automaticamente un indice B+ Tree su di essa. Non è necessario creare un ulteriore indice su Porto(Città) per il join.

## 6 Applicazione Software

Il file **Query.c** implementa un programma in linguaggio C che consente di connettersi a un database PostgreSQL contenente i dati relativi alla gestione delle crociere. Lo scopo principale del programma è eseguire e visualizzare i risultati di diverse query SQL predefinite, come descritto nella Sezione 5 del progetto. All'avvio, il programma presenta un'interfaccia testuale interattiva che mostra un menu numerato con le interrogazioni disponibili. L'utente può selezionare la query desiderata digitando il numero corrispondente.

Le query proposte coprono diversi aspetti gestionali, tra cui:

1. Identificazione delle crociere che toccano più di tre porti diversi, con indicazione delle città di partenza, di arrivo e del numero di tappe.
2. Visualizzazione delle crociere in partenza da una città specifica fornita dall'utente.
3. Individuazione delle crociere con costo medio dei biglietti superiore a 500 euro.
4. Elenco degli animatori con un numero di eventi organizzati superiore a due.
5. Calcolo, per ogni crociera, della percentuale di occupazione rispetto alla capacità massima disponibile.

In particolare, la seconda query è stata resa parametrica: all'utente viene richiesto di inserire il nome della città di partenza, che viene quindi utilizzato per personalizzare la query SQL in fase di esecuzione.

Il codice gestisce anche l'inizializzazione del database, caricando uno script SQL che crea e popola le tabelle necessarie, e suddivide il file delle query in modo da consentire una facile selezione e gestione tramite menu. Le query disponibili non sono legate al codice, quindi possono essere aggiunte, modificate o eliminate dal file Crociere.sql e si modificherà quindi in automatico il menù a scelta.

Note: in alcuni computer si può compilare direttamente con:

```
- gcc -o query Query.c -lpq
```

In altri non riesce a linkare da solo il pacchetto di postgresql quindi serve compilare con il codice:

```
- gcc -o query Query.c -I/usr/include/postgresql -lpq
```

Membri del gruppo:

```
- **Ghiraldin Mirco** 2102505  
- **Stevanin Michele** 2101741
```