

Documentazione database
Compagnie di navigazione

15/12/2023



UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II

Alessio Nesi
Simone Sommella
Marco Guido Scotto Di Uccio

Contents

1	Progettazione concettuale	3
1.1	Analisi dei requisiti	3
1.2	Schema Concettuale	4
1.3	Dizionario Entità	5
1.4	Dizionario Associazioni	7
2	Ristrutturazione del Modello Concettuale	8
2.1	Analisi delle Ridondanze	8
2.2	Eliminazione degli attributi multivalore	8
2.3	Eliminazione degli attributi composti	8
2.4	Analisi delle generalizzazioni	8
2.5	Partizionamento/Accorpamento di entità e associazioni	8
2.6	Identificazione Chiavi Primarie	9
2.7	Schema ristrutturato UML	10
2.8	Schema ristrutturato ER	10
3	Traduzione al modello logico	11
3.1	Mapping associazioni	11
3.1.1	Associazioni 1-N	11
3.1.2	Associazioni N-N	12
3.2	Modello logico	12
3.3	Dizionario dei vincoli	13
3.4	Descrizione di trigger e procedure	14
3.4.1	Procedure	14
3.4.2	Trigger	14

1 Progettazione concettuale

1.1 Analisi dei requisiti

*Il sistema si basa sulla conoscenza delle corse offerte dalle **compagnie di navigazione**. Ogni corsa è offerta da una specifica compagnia di navigazione, che indica il tipo di **natante** utilizzato. Tra i tipi di natante si distinguono **traghetti** (che trasportano persone e automezzi), gli **aliscafi** e le **motonavi** (che trasportano entrambe solo passeggeri). Ogni corsa ha cadenza giornaliera, un orario di partenza e un orario di arrivo ma può essere operata solo in alcuni giorni della settimana e solo in alcuni specifici periodi dell'anno*

La base dati si occupa della gestione delle corse offerte da più compagnie di navigazione.

Le compagnie di navigazione hanno a disposizione vari tipi di natanti:

- **Traghetto**, capace di trasportare autoveicoli e persone;
- **Aliscafo**, capace di trasportare solo persone;
- **Motonave**, capace di trasportare solo persone.

Ognuno di questi si occupa di una o più corse a cadenza giornaliera.

*Ogni **corsa** ha diversi prezzi: un prezzo per il biglietto intero, uno per il biglietto ridotto. Inoltre può esserci un sovrapprezzo per la prenotazione e uno per i bagagli. Ogni corsa è caratterizzata da un **porto** di arrivo e uno di partenza. Il sistema può essere utilizzato da compagnia e dai passeggeri.*

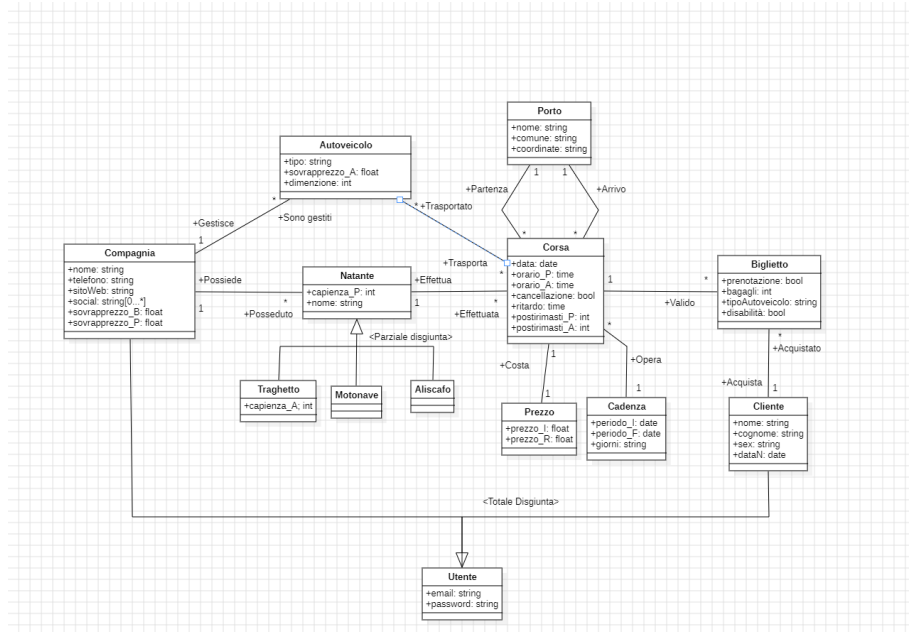
Individuiamo che ogni corsa ha prezzi diversi che andiamo a raccogliere nell'entità "Prezzi".

I sovrapprezzi non dipendendo dalla corsa li inseriamo nell'entità "Compagnia".

Per tenere traccia dei porti di partenza e di arrivo creiamo l'entità "Porto".

Utilizziamo una generalizzazione in modo da poter far utilizzare il sistema sia dalle compagnie sia dai passeggeri.

1.2 Schema Concettuale



1.3 Dizionario Entità

Entità	Descrizione	Attributi
Compagnia	La compagnia di navigazione che si occupa delle corse.	nome: Nome compagnia telefono: Contatto telefonico della compagnia sitoWeb: URL del sito della compagnia social: Elenco dei social della compagnia sovrapprezzo_B: Sovrapprezzo per il bagagli sovrapprezzo_P: Sovrapprezzo per la prenotazione
Autoveicolo	Autoveicoli che sono trasportati dai traghetti.	tipo: Il tipo di autoveicolo sovrapprezzo_A: Sovrapprezzo dell'autoveicolo dimensione: Dimensione dell'Autoveicolo
Natante	I natanti posseduti dalle compagnie.	capienza: Numero massimo di persone trasportabili nome: Nome del Natante
Traghetto	Tipologia di natante in grado di trasportare anche autoveicoli.	capienzaA: Numero massimo di autoveicoli trasportabili
Aliscafo	Tipologia di natante in grado di trasportare solo persone.	
Motonave	Tipologia di natante in grado di trasportare solo persone.	
Porto	Luogo da cui partono o arrivano i natanti.	nome: Nome del porto comune: Nome del comune in cui si trova il porto coordinate: Coordinate del Porto

Corsa	Tratta percorsa per un periodo dai natanti.	data: Data in cui si svolge la corsa orario_P: Orario di partenza orario_A: Orario di arrivo cancellazione: Indica se la corsa è stata o meno cancellata ritardo: Eventuali minuti di ritardo postirimasti_P: Posti rimasti per i passeggeri postirimasti_A: Posti rimasti per gli autoveicoli
Prezzo	Costo della corsa in euro.	prezzo_I: Prezzo del biglietto intero prezzo_R: Prezzo del biglietto ridotto
Cadenza	Periodo in cui i natanti effettuano una corsa.	periodo_I: Inizio del periodo in cui viene effettuata la corsa periodo_F: Fine del periodo in cui viene effettuata la corsa
Biglietto	Biglietto acquistato da un cliente per una corsa.	prenotazione: Indica se il biglietto è stato prenotato bagagli: Numero di bagagli tipoAutoveicolo: Tipo dell'autoveicolo portato dal cliente disabilita: Indica se il cliente è o meno disabile
Cliente	Fruitore del sistema.	nome: Nome del cliente cognome: Cognome del cliente sex: sesso del cliente dataN: Data di nascita del cliente
Utente	Cliente o compagnia che accedono al sistema.	email: email di accesso password: password di accesso

1.4 Dizionario Associazioni

Associazione	Descrizione	Attributi
Gestisce	Associazione tra Compagnia e Autoveicolo Uno a Molti	
Possiede	Associazione tra Compagnia e Natante Uno a Molti	
Trasportato	Associazione tra Autoveicolo e Corsa Molti a Molti	
Effetua	Associazione tra Natante e Corsa Uno a Molti	
Partenza	Associazione tra Corsa e Porto Molti a Uno	
Arrivo	Associazione tra Porto e Corsa Uno a Molti	
Valido	Associazione tra Corsa e Biglietto Uno a Molti	
Acquistato	Associazione tra Biglietto e Cliente Molti a Uno	
Opera	Associazione tra Corsa e Cadenza Molti a Uno	
Costa	Associazione tra Corsa e Prezzo Uno a Uno	

2 Ristrutturazione del Modello Concettuale

2.1 Analisi delle Ridondanze

Abbiamo scelto di tenere gli attributi *PostirimastiP* e *PostirimastiA* in *Corsa*, anche se era possibile calcolarli sottraendo alla capienza del *Natante* i *Biglietti* comprati per quella *Corsa*. Abbiamo quindi deciso questa modalità perché il calcolo sarebbe stato troppo dispendioso rispetto all'aggiornare i due attributi.

2.2 Eliminazione degli attributi multivalore

L'attributo *social* dell'entità *Compagnia* è un attributo multivalore, abbiamo quindi deciso di trattarlo come attributo singolo di tipo stringa che contiene tutti i *social* della *Compagnia*.

L'attributo *giorni* dell'entità **Cadenza** è un attributo multivalore, abbiamo quindi deciso di trattarlo come attributo singolo di tipo stringa che contiene tutti i giorni della settimana in cui avvengono le partenze.

2.3 Eliminazione degli attributi composti

Non abbiamo riscontrato attributi Composti.

2.4 Analisi delle generalizzazioni

Procediamo all'analisi e all'eliminazione delle generalizzazioni:

1. Accorpamento della generalizzazione disgiunta parziale; in quanto tale abbiamo scelto di accorpare le entità figlie **Traghetto**, **Motonave** e **Aliscafo** all'interno dell'entità padre **Natante** aggiungendo nel padre l'attributo *tipo*.
2. Accorpamento della generalizzazione disgiunta totale; in quanto tale abbiamo deciso di accorpare l'entità padre **Utente**, nelle entità figlie **Compagnia** e **Cliente**; inserendo gli attributi di *Utente* in entrambe le entità.

2.5 Partizionamento/Accorpamento di entità e associazioni

Abbiamo scelto di accorpare l'entità **Prezzo** all'interno dell'entità **Corsa** essendo una relazione 1-1.

2.6 Identificazione Chiavi Primarie

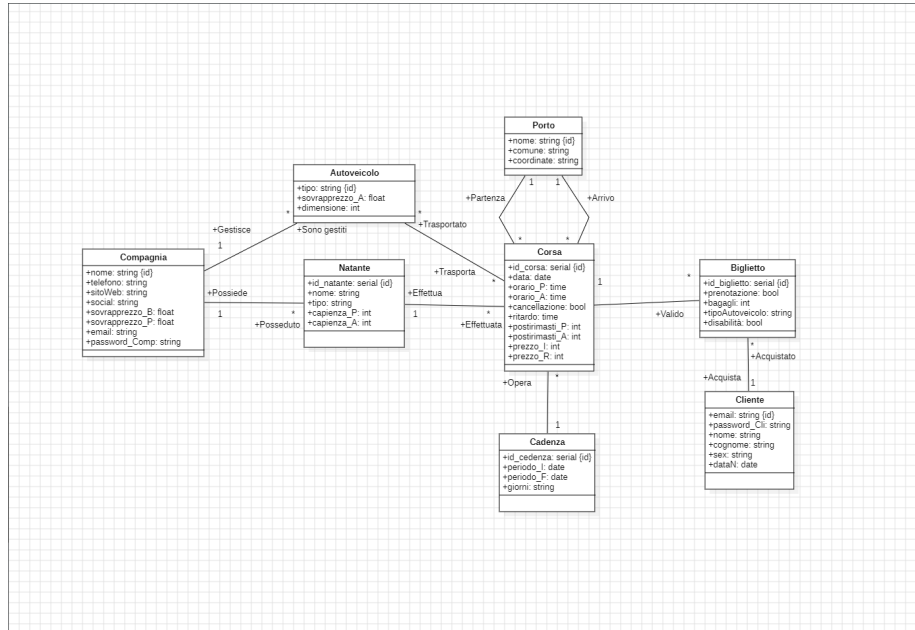
Abbiamo identificato le seguenti chiavi primarie:

- Nell'entità **Compagnia** abbiamo l'attributo *nome* di tipo stringa;
- Nell'entità **Autoveicolo** abbiamo l'attributo *tipo* di tipo stringa;
- Nell'entità **Porto** abbiamo l'attributo *nome* di tipo stringa;
- Nell'entità **Cliente** abbiamo l'attributo *email* di tipo stringa;

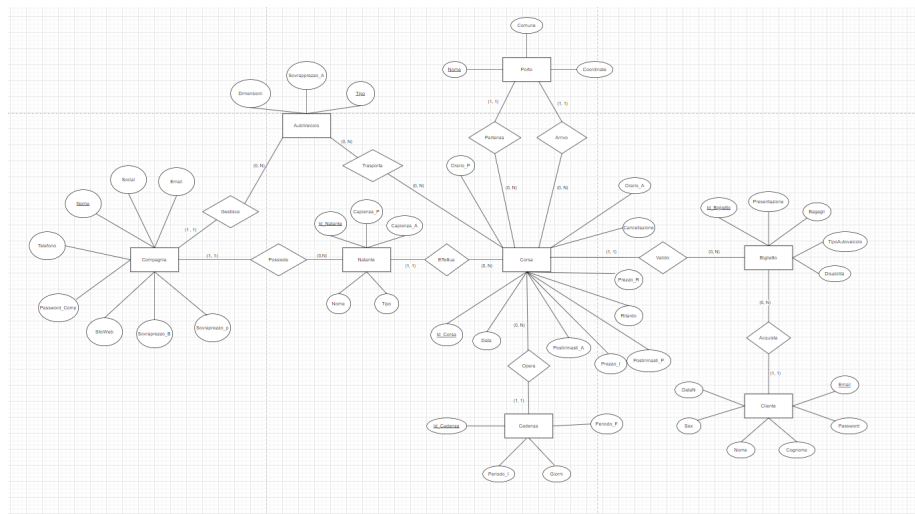
Abbiamo invece inserito il seguenti codici come chiavi primarie:

- Nell'entità **Natante** abbiamo inserito come attributo il codice *id_Natante* che si autoincrementa;
- Nell'entità **Corsa** abbiamo inserito come attributo il codice *id_Corsa* che si autoincrementa;
- Nell'entità **Biglietto** abbiamo inserito come attributo il codice *id_Biglietto* che si autoincrementa;
- Nell'entità **Cadenza** abbiamo inserito come attributo il codice *id_Cadenza* che si autoincrementa;

2.7 Schema ristrutturato UML



2.8 Schema ristrutturato ER



3 Traduzione al modello logico

3.1 Mapping associazioni

3.1.1 Associazioni 1-N

- *Compagnia-possiede-Natante* inserimento chiave di Compagnia in Natante come chiave esterna;
- *Natante-effettua-Corsa* inserimento chiave di Natante in Corsa come chiave esterna;
- *Corsa-valida-Biglietto* inserimento chiave di Corsa in Biglietto come chiave esterna;
- *Porto-partenza-Corsa* inserimento chiave di Porto in Corsa come chiave esterna;
- *Porto-arrivo-Corsa* inserimento chiave di Porto in Corsa come chiave esterna;
- *Cliente-acquista-Biglietto* inserimento chiave di Cliente in Biglietto come chiave esterna;
- *Cadenza-Opera-Corsa* inserimento chiave di Cadenza in Corsa come chiave esterna;
- *Compagnia-gestisce-Autoveicolo* inserimento chiave di Compagnia in Autoveicolo come chiave esterna;

3.1.2 Associazioni N-N

Per questa relazione inseriamo le chiavi delle due relazioni come chiavi esterne:

Associazione	Relazione	Associazione
Autoveicolo	Trasporta	Corsa

3.2 Modello logico

Durante la traduzione abbiamo identificato come chiave primaria comporta nella classe Autoveicolo gli attributi tipo e la foreign key nome_compagnia.

Gli attributi sottolineati sono chiavi primarie, Gli attributi con asterisco sono chiavi esterne:

Compagnia (nome, telefono, email, sitoWeb, social, passwordComp, sovrapprezzo_B, sovrapprezzo_P)

Natante (id_Natante, capienzaP, capienzaA, nome, tipo, compagnia*)

Natante.compagnia → Compagnia.nome

Autoveicolo (tipo, sovrapprezzo_A, dimensioni, nome_compagnia*)

Autoveicolo.nome – compagnia → Compagnia.nome

Porto (nome, comune, coordinate)

Cadenza (id_Cadenza, giorno, periodo_I, periodo_F)

Corsa (id_Corsa, data, orario_P, orario_A, cancellazione, ritardo, prezzo_I, prezzo_R, postirimasti_P, postirimasti_A, id_natante*, id_cadenza*, partenza*, arrivo*)

Corsa.id – natante → Natante.id – Natante

Corsa.id – cadenza → Cadenza.id – cadenza

Corsa.partenza → Porto.nome

Corsa.arrivo → Porto.nome

Biglietto (id_biglietto, prenotazione, bagagli, tipoAutoveicolo, disabilità, id_corsa*, email*)

Biglietto.id_corsa → Corsa.id_corsa

Biglietto.email → Cliente.email

Cliente (email, password, nome, cognome, sex, dataN)

Trasporta (tipo*, nome_Compagnia*, id_corsa*)

Trasporta.tipo, Trasporta.nome_Compagnia → Autoveicolo.tipo, Autoveicolo.nome_Compagnia

Trasporta.id_corsa → Corsa.id_corsa

3.3 Dizionario dei vincoli

Nome del vincolo	Classe su cui agisce	Descrizione
SovrapprezzoValidoA	Autoveicolo	Impedisce che il sovrapprezzo sia negativo
dimensionevalida	Autoveicolo	Impedisce che la dimensione sia negativa o zero
SovrapprezzoValidoP	Compagnia	Impedisce che il sovrapprezzo per la prenotazione sia negativo
SovrapprezzoValidoB	Compagnia	Impedisce che il sovrapprezzo per i bagagli sia negativo
PeriodoValido	Cadenza	Impone il periodo di fine cadenza successivo a quello di inizio, ed entrambi successivi al giorno corrente
EmailValida	Compagnia	Impone l'inserimento di una mail realistica
SitoValido	Compagnia	Impone l'inserimento di un sito realistico
PostiValidiP	Natante	Impedisce l'inserimento di una capienza negativa
PostiValidiA	Natante	Impedisce l'inserimento di una capienza auto negativa
PortiDistinti	Corsa	Impedisce l'inserimento di un porto di destinazione uguale a quello di partenza
bagagliReali	Biglietto	Impedisce di inserire un numero negativo di bagagli
EmailValida	Clientre	Impone l'inserimento di una mail realistica
portiValidi	Corsa	Permette l'esistenza di una corsa senza porto di partenza o destinazione solo nel caso in cui quella corsa si astata annullata
NascitaValida	Cliente	Permette la registrazione solo di clienti maggiorenni e nati dopo il 1900

3.4 Descrizione di trigger e procedure

3.4.1 Procedure

Non sono state individuate procedure.

3.4.2 Trigger

Abbiamo individuato i seguenti Trigger:

AnnullaCorsa: Una volta aggiornato il campo cancellazioe in corsa, invia il rimborso a tutti gli utenti che hanno comprato il biglietto per quella corsa.

AggiornaPostoCli: Dopo l'acquisto di un biglietto da parte del Cliente aggiorna i posti rimasti della corsa;

AggiornaPostoA: Dopo l'acquisto di un biglietto con posto auto da parte del Cliente aggiorna i posti auto rimanenti nella corsa;

NatanteVenduto: Una volta eliminato un Natante le corse corrispondenti vengono dismesse;

ModificaPeriodoI: Ogni volta che viene aggiornato il periodo di inizio (Attributo PeriodoI) in Cadenza viene controllato se la data inserita è successiva o antecedente alla data sostituita:

Se successiva vengono eliminate le corse previste nel lasso di tempo tra la vecchia data e la nuova.

Se la data è antecedente vengono create nuove corse;

ModificaPeriodoF: Ogni volta che viene aggiornato il periodo di fine (Attributo PeriodoF) in Cadenza viene controllato se la data inserita è successiva o antecedente alla data sostituita:

Se successiva vengono create le corse previste nel lasso di tempo tra la vecchia data e la nuova.

Se la data è antecedente vengono eliminate le vecchie corse;

AnnullaBiglietto: Quando un biglietto viene eliminato vengono riaggiornati i posti rimanenti per le persone (Attributo Postirimasti_P) e eventualmente i posti rimanenti dei autoveicoli (Attributo Postirimasti_A);

ModificaGiorni: Quando viene modificato il campo Giorno in Cadenza questo trigger:

Elimina tutte le corse nelle date del giorno selezionato se l'Utente elimina un giorno;

Aggiunge tutte le corse nelle date del giorno selezionate se l'Utente aggiunge un giorno;

InserimentoCorse: Ogni volta che viene inserita una nuova Corsa con parametro data nullo questo trigger inserisce Corse con le stesse specifiche in tutte le date del periodo definito in Cadenza;

RiempiTrasporta: Ogni volta che viene creata una corsa riempie in maniera opportuna la tabella Trasporta;

RiempiTrasporta2: Ogni volta che viene creata una corsa riempie in maniera opportuna la tabella Trasporta;

EliminaPorto: Ogni volta che viene eliminato un Porto vengono cancellate tutte le Corse che partono o arrivano a esso;

InserisciPosti: Dopo l'inserimento di una corsa riempie automaticamente i posti rimanenti di clienti e auto con i dati presi dalla capienza del natante selezionato.