

Progetto Basi Dati: Agenzia Viaggi

Autori:

Roberto Piscopo

Andrea Pio Villano

Punto I: Descrizione delle realtà d'interesse, raccolta delle specifiche e prima analisi.

Si vuole progettare una base di dati per la gestione dei voli di un'agenzia viaggi.

Alla base del database sono i clienti, identificati dal loro codice fiscale. Di questi si vuole anche memorizzare il loro nome, il cognome e l'età.

Ogni viaggiatore acquista biglietti, di cui memorizziamo il loro codice identificativo, insieme al tipo di bagaglio, il prezzo e il numero del corrispondente posto in aereo. Ogni biglietto può essere di prima classe oppure economy; dei primi è necessario memorizzare i tipi di benefici dei quali il viaggiatore può usufruire, mentre dei biglietti economy salviamo il buono sconto usato per l'acquisto di quel biglietto.

I biglietti sono relativi ai voli, dei quali è necessario tener traccia di diversi fattori: numero del volo (che lo identifica univocamente), durata del volo, data di partenza, ora di partenza, numero dei passeggeri, aeroporto di partenza e aeroporto di destinazione.

Un volo viene gestito da una compagnia aerea, di cui vogliamo sapere l'ID identificativo e il nome. Ciascuna compagnia ha degli impiegati che vogliamo considerare, i quali possono essere di due tipi: piloti o assistenti di volo. Dei piloti memorizziamo il numero di voli eseguiti, degli assistenti di volo le lingue di competenza. Per tutti gli impiegati vanno anche specificati: ID che li identificano, l'ammontare dello stipendio, il numero di anni di lavoro e l'eventuale/i numero/i di telefono.

Infine, ogni compagnia aerea ha affiliati degli aeroporti, di cui salviamo il nome, la città in cui si trovano e l'ID dell'aeroporto, che lo identifica.

Il database in fase di progettazione riguarda la gestione di una piattaforma per il monitoraggio di una frazione dell'agenzia viaggi, con il fine di rendere possibile l'analisi dell'andamento nell'ambito dei voli commerciali e le relazioni che ci sono con le varie compagnie aeree di quest'ultima.

Mentre, nella sua totalità, un'agenzia viaggi svolge la funzione di intermediario tra il cliente, con le sue necessità e richieste, e i vari enti di trasporto passeggeri, alloggio, ristorazione etc., il partizionamento della stessa permette uno studio più approfondito di strategie di marketing e analisi del rapporto domanda/richiesta. Questo è il motivo della scelta di trattare solo l'ambito dei voli.

Nella realtà che stiamo rappresentando, ogni viaggiatore viene registrato automaticamente nella piattaforma al momento dell'acquisto di un biglietto tramite i servizi della nostra agenzia. La distinzione del biglietto in economy e prima classe permette, in maniera indiretta, un'approssimativa disponibilità economica del cliente. I benefits di cui il cliente decide di usufruire possono essere usati da altre partizioni dell'agenzia per proporre servizi e offerte più mirati.

La piattaforma memorizza tutti i voli effettuati dai clienti. In questo modo, si possono controllare le tappe più gettonate, anche in relazione al periodo dell'anno, oppure se un determinato cliente visita una tappa abitualmente.

Per una scelta di progetto, anche i voli sono registrati solamente al momento della vendita del primo biglietto, in maniera automatica.

Nella nostra piattaforma vengono registrate anche tutte le compagnie che si occupano della creazione e gestione dei voli, in modo da monitorare quali compagnie favorire nella scelta di un volo in base a fattori di rapporto qualità/prezzo, numero di voli messi a disposizione e su quali percorsi.

Con l'obiettivo di fornire al cliente il volo più adatto, risulta logico analizzare anche le caratteristiche dei dipendenti di ciascuna compagnia aerea. In particolare, i fattori principali che influenzano la scelta sono quelli relativi agli assistenti di volo per il cliente, mentre quelli generali, come gli anni di lavoro e lo stipendio, interessano la scelta da parte dell'agenzia.

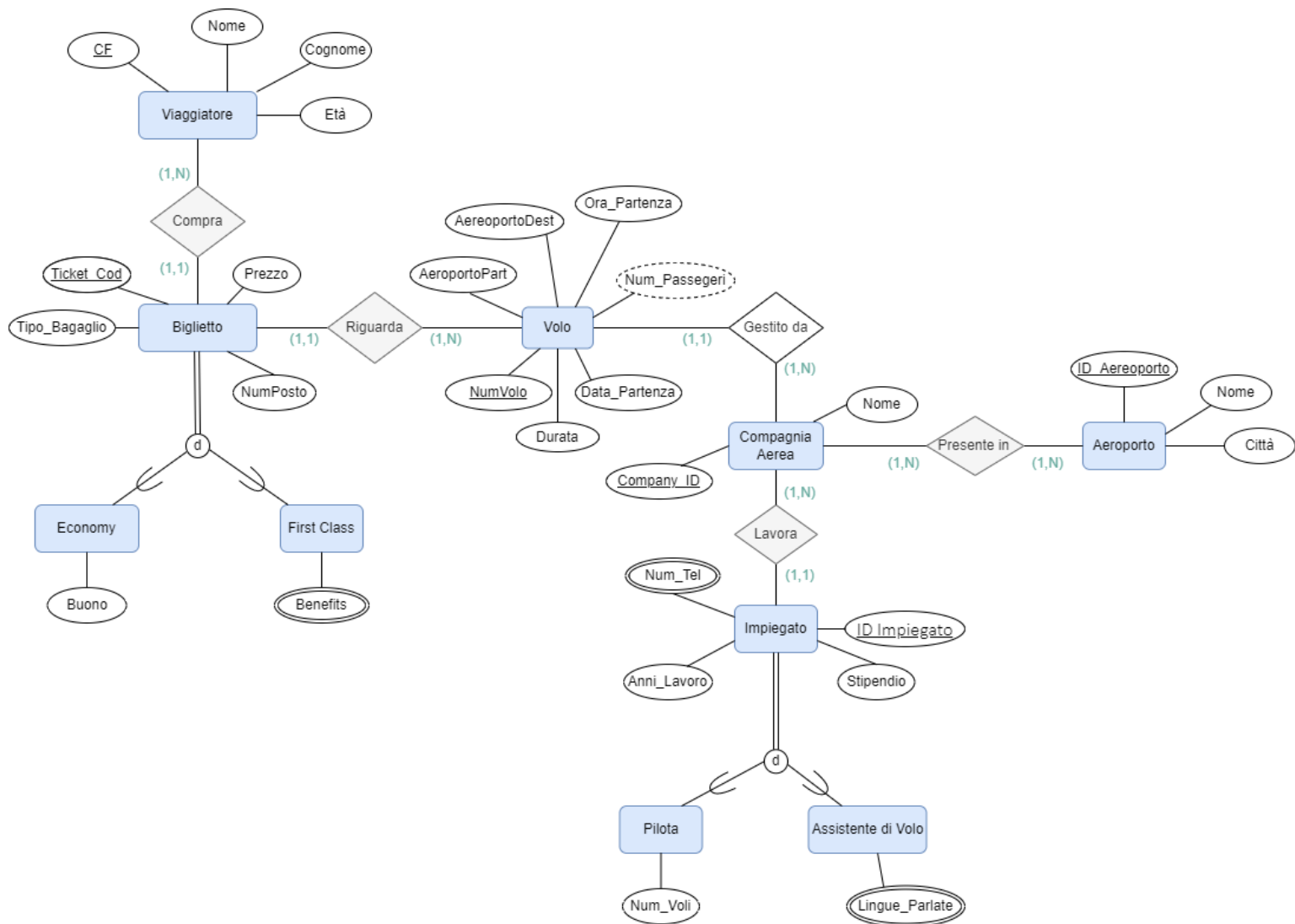
Il nostro database conserva al suo interno anche gli aeroporti con i quali lavorano le compagnie. Tramite la memorizzazione di questo dato è possibile per l'agenzia rilevare quali viaggi sono disponibili per un cliente.

Glossario dei termini principali

Termine	Descrizione
Viaggiatore	Una persona che ha viaggiato o che possiede un biglietto
Biglietto	Documento che autorizza il passeggero a volare su una determinata tratta.
First Class	Documento che autorizza il passeggero ad accedere ad una serie di benefits.
Economy	Documento che permette al viaggiatore di volare e risparmiare.
Compagnia Aerea	Impresa che gestisce i voli per passeggeri o merci.
Impiegato	Una persona che lavora per una compagnia.
Pilota	Una persona che lavora per la compagnia, abilitata e autorizzata a guidare un aereo.
Assistente di Volo	L'insieme di personale della compagnia specializzato nell'assistenza ai passeggeri e nel controllo delle carte di imbarco.
Volo	Spostamento da una zona/Paese ad un altro, attraverso un aereo.
Aeroporto	Infrastruttura attrezzata per il decollo e l'atterraggio di aerei

Punto II: Progettazione Concettuale

Schema EER



Dizionario delle entità

Legenda: sotto-entità, attributo-multivalore, attributo ridondante.

Entità	Descrizione	Attributi	Identificatore
Viaggiatore	Una persona che ha viaggiato o che possiede un biglietto	CF, Nome, Cognome, Età	CF
Biglietto	Documento che autorizza il passeggero a volare su una determinata tratta.	Ticket_Cod, Prezzo, Tipo Bagaglio, Numero Posto	Ticket_Cod
Economy	Documento che permette al viaggiatore di volare e risparmiare.	Buono	-
First Class	Documento che autorizza il passeggero ad accedere ad una serie di benefits.	Benefits	-
Volo	Spostamento da una zona/Paese ad un altro, attraverso un aereo.	Num Volo, AeroportoDest, AereoportoPart, Ora Partenza, Data Partenza, Durata, Num Passegeri,	Num Volo
Compagnia Aerea	Impresa che gestisce i voli per passeggeri o merci.	Company ID, Nome	Company ID
Aeroporto	Infrastruttura attrezzata per il decollo e l'atterraggio di aerei	ID Aeroporto, Nome, Città	ID Aeroporto
Impiegato	Una persona che lavora per una compagnia.	ID Impiegato, Anni lavoro, Stipendio, Num telefono	ID Impiegato
Pilota	Una persona che lavora per la compagnia, abilitata e autorizzata a guidare un aereo.	Numero Voli	-
Assistente Volo	L'insieme di personale della compagnia specializzato nell'assistenza ai passeggeri e nel controllo delle carte di	Lingue Parlate	-

	imbarco.		
--	----------	--	--

Dizionario delle relazioni

Relazione	Descrizione	Entità coinvolte	Attributi
Compra	Un viaggiatore compra i biglietti.	Viaggiatore: (1, N) Biglietto: (1, 1)	-
Riguarda	Il Biglietto riguarda uno ed un solo Volo	Biglietto: (1, 1) Volo: (1, N)	-
Gestito da	Un Volo è gestito da una compagnia	Volo: (1, 1) Compagnia Aerea: (1, N)	-
Presente in	La compagnia opera in uno o più aeroporti	Compagnia Aerea: (1, N) Aeroporto: (1, N)	-
Lavora	Un Impiegato lavora in una singola Compagnia	Impiegato: (1, 1) Compagnia: (1, N)	-

Vincoli non esprimibili nello schema

- Il *prezzo* del biglietto dev'essere un intero compreso fra 0 e 2000.
- Il *Ticket code* del biglietto deve essere un valore alfanumerico
- Il *codice fiscale* è un valore alfanumerico di 16 caratteri.
- Le *date* sono del tipo *aaaa-mm-gg*.
- La *durata* deve essere del tipo *hh:mm*.
- Il *Buono* deve essere uno scelto tra: "Family", "Student", "U12"
- I *benefits* possono essere uno o più tra: "Streaming", "Menu_Premium", "Poltrona"
- Il *tipo bagaglio* può essere uno dei tre valori: S, M, L.

Punto III: Definizione delle procedure per la gestione della base di dati

Tavola dei volumi

Definiamo di seguito la tavola dei volumi della base di dati.

Concetto	Tipo	Carico Applicativo
Viaggiatore	E	40
Biglietto	E	200
Economy	E	150
First Class	E	50
Volo	E	100
Compagnia Aerea	E	10
Aeroporto	E	110
Impiegato	E	120
Pilota	E	40
Assistente Volo	E	80
Compra	R	200
Riguarda	R	200
Gestito da	R	100
Presente in	R	120
Lavora	R	120

Tavola delle operazioni

Definiamo di seguito la tavola delle operazioni per la gestione dei dati memorizzati nella base di dati.

Operazione	Tipo	Frequenza
Inserire un viaggiatore	I	3/m
Inserire un aeroporto	I	1/aa
Inserire un impiegato	I	2/aa
Dato Nome e Cognome di un viaggiatore, stampare i relativi voli	I	2/giorno
<i>Stampare numero voli di durata inferiore alle due ore e relativo ID impiegato</i>	I	1/mese
Dato un numero volo, stampa la città in cui si trova l'aeroporto di partenza.	I	20/mese
Stampare quanto ha speso un viaggiatore in totale	I	2/mm
Stampare numero passeggeri di ogni volo	B	2/mese
Stampare il resoconto biglietti	B	1/mese

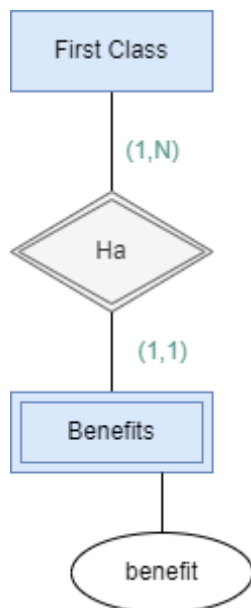
Stampare i privilegi riguardo un dato biglietto	I	10/m
Controllo validità buono U12	I	10/m
Stampare tabella impiegati	B	2/mese
Rimozione impiegato in base agli anni di lavoro	I	2/anno
Aumentare stipendio di un impiegato se conosce due o più lingue	I	1/anno
Inserire un volo	I	2/mese
Modifica costo biglietto in base al posto	I	1/mese

Punto IV: Progettazione Logica

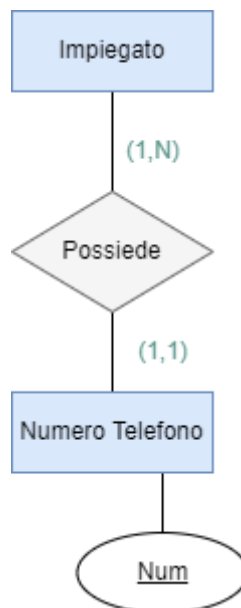
Eliminazione attributi multivalore

Nello schema inizialmente elaborato sono presenti 3 attributi multivalore, Benefits nell'entità First Class, Numero Telefono nell'entità Impiegato e Lingue Parlate nell'entità Assistente di volo. In fase di progettazione logica, questo tipo di attributo va rimosso o ristrutturato del tutto, legandolo alla sua ex-entità tramite una relazione, e trasformandolo a sua volta in entità.

1)



2)



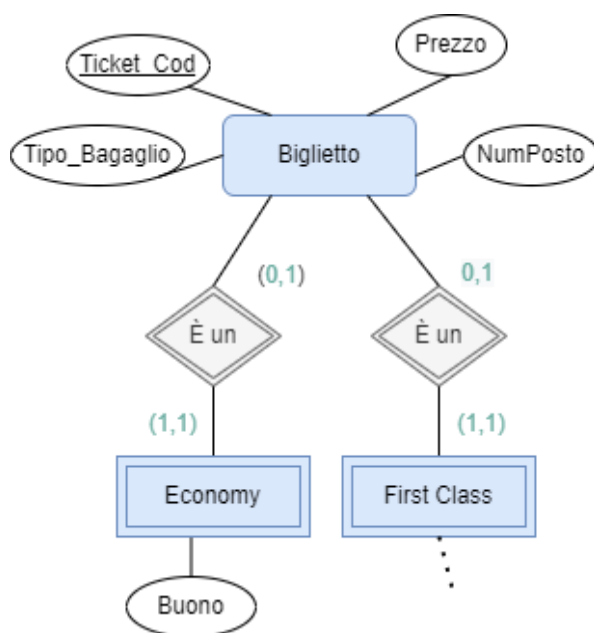
3) Lingue Parlate->

Vedi Punto Eliminazione
Gerarchie

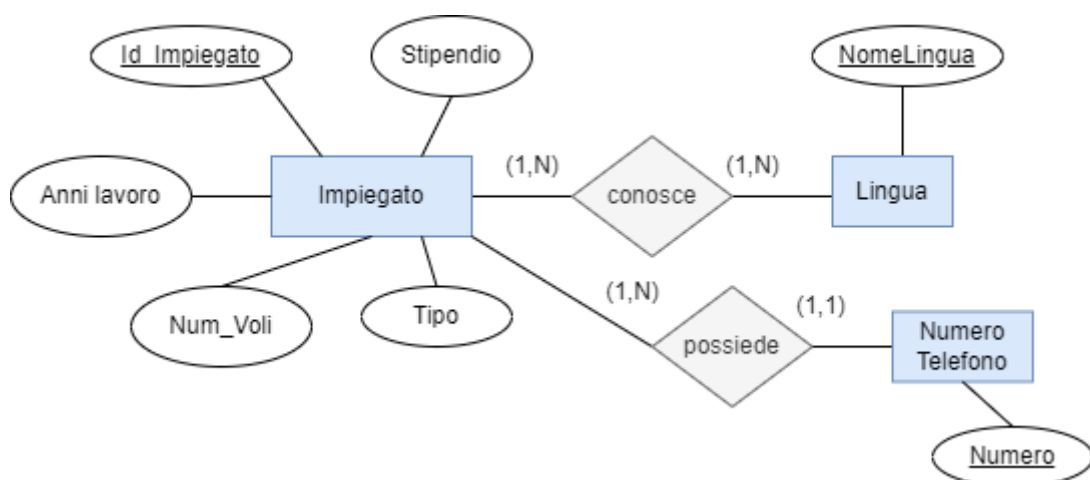
Eliminazione delle gerarchie

Nell'ambito dell'accorpamento dei concetti, dell'eliminazione di specializzazioni e generalizzazioni per semplificare il database e lo schema proposto, si è deciso di **effettuare la creazione di due nuove relazioni che rendendo le entità figlie due nuove entità deboli**, per il primo punto. Mentre si è deciso di **accorpare le entità figlie al padre** nel secondo punto, con l'aggiunta dell'attributo 'TIPO'.

1)



2)



Analisi delle ridondanze

Il dato ridondante nella realtà d'interesse descritta è il **Numero di passeggeri in un volo** (*Num_passegeri*) dell'entità Volo. Infatti, il numero di passeggeri è derivabile dalle relazioni che legano Volo e Biglietto, e poi Biglietto e Viaggiatore. Sarebbe possibile attraverso un'interrogazione più o meno articolata ricavare il valore senza che questa permanga come attributo all'interno dell'entità Volo. Supponendo che l'attributo abbia un peso di **2 byte**, essendo un normale intero, e considerato che il volume dell'entità volo è uguale a **100**, il dato andrebbe ad occupare uno spazio totale di **200 Byte**. Per decidere se mantenere o meno il dato ridondante vanno calcolate, per le operazioni che lo coinvolgono, la differenza nel numero di accessi con e senza quest'ultimo.

Tavola degli accessi

Si valutino dunque gli accessi con e senza l'attributo ridondante dell'entità volo. In particolare, osserviamo i dati relativi alle operazioni...

Operazione "Inserire un viaggiatore"

Calcolo con ridondanza				Calcolo senza ridondanza			
Concetto	Costrutto	Accessi	Tipo	Concetto	Costrutto	Accessi	Tipo
Viaggiatore	E	1	S	Viaggiatore	E	1	S
Compra	R	5	S	Compra	R	5	S
Biglietto	E	5	S	//Biglietto	E	5	S
Riguarda	R	5	<u>S</u>				
Volo	E	5	L				
Volo	E	5	S				
TOT: $21S + 5L = 42 + 5 = 47 * 3 = 141$ al mese				TOT: $11S = 22 * 3 = 66$ al mese			

Operazione “Stampare voli compreso numero Passeggeri”

Calcolo con ridondanza				Calcolo senza ridondanza			
Concetto	Costrutto	Accessi	Tipo	Concetto	Costrutto	Accessi	Tipo
Volo	E	100	L	Volo	E	100	L
				Riguarda	R	2 x 100	L
				Biglietto	E	2 x 100	L
				Compra	R	2 x 100	L
Tot= 100L= 100*2 = 200 al mese				Tot= 100+200+200+200=700*2=1400 al mese			

Operazione “Inserire un volo”

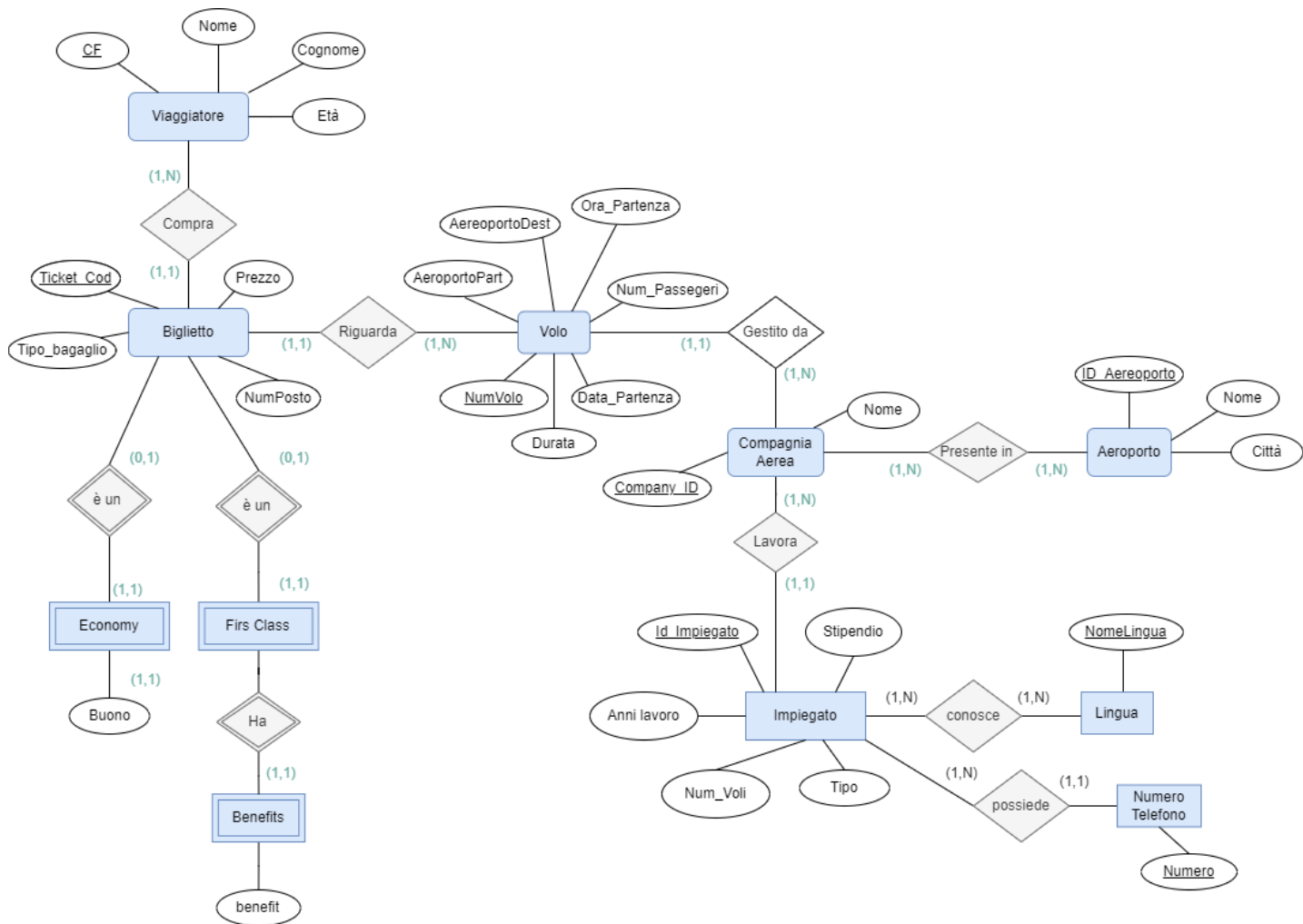
Calcolo con ridondanza				Calcolo senza ridondanza			
Concetto	Costrutto	Accessi	Tipo	Concetto	Costrutto	Accessi	Tipo
Volo	E	1	S	Volo	E	1	S
Riguarda	R	2	S	Riguarda	R	2	S
Biglietto	E	2	L				
Compra	R	2	L				
Tot 3S+4L=6+4=10*2=20 al mese				Tot=3S=6*2=12 al mese			

Tot operazione con ridondanza: 361

Tot operazione senza ridondanza: 1478

Dopo aver effettuato l’analisi si evince che conviene mantenere l’attributo numero Passeggeri nell’entità Volo.

Schema EER ristrutturato



Schema relazionale

Si procede con il Mapping:

Viaggiatore (CF, Nome, Cognome, Età).

Biglietto (Ticket_Cod, Prezzo, Tipo Bagaglio, NumPosto, Viaggiatore.CF↑, Volo.NumVolo↑).

Economy (Biglietto.Ticket_Cod ↑, Buono).

First Class (Biglietto.Ticket_Cod ↑).

Benefits (FirstClass.Biglietto.Ticket_Cod ↑, Benefit).

Volo (NumVolo, AeroportoPart, AeroportoDest, Ora_Partenza, Data_Partenza, Durata,

Compagnia.Id_Company↑, Num_Passeggeri).

Compagnia (Id_Company, Nome).

Aeroporto (Id_Aeroporto, Nome, Città).

Impiegato (Cod_Emp, Stipendio, Anni Lavoro, Compagnia.Id_Company↑, Tipo, NumVol).

Lingue (Nome).

CONOSCE (Impiegato.Cod_Emp↑, Lingue.Nome↑).

Numero Telefono (Num, Impiegato.Cod_emp↑).

PRESENTE IN (Compagnia.Id_Company↑, Id_Aeroporto↑).

Normalizzazione

Il database Scuola si presenta già normalizzato.

- È in **prima forma normale** perché, una volta ristrutturato, lo schema EER non presenta né attributi composti, né attributi multivalore, rendendo ogni campo fondamentalmente atomico.
- È in **seconda forma normale** perché, oltre ad essere in prima forma normale, quando è presente una chiave primaria composta da più attributi (ad esempio nell'entità Classe), tutte le dipendenze funzionali che la riguardano sono piene e non parziali.
- È in **terza forma normale** perché, oltre ad essere in seconda forma normale e in prima forma normale, in tutte le tabelle non sono presenti dipendenze transitive fra attributi non chiave e la chiave primaria.

Punto V: Realizzazione della Base di Dati con MySQL

Di seguito sono elencate le istruzioni MySQL per la creazione della base di dati:

```
CREATE SCHEMA IF NOT EXISTS AgenziaViaggi ;
USE AgenziaViaggi ;

CREATE TABLE aeroporto
(
    id_aeroporto INT NOT NULL,
    nome VARCHAR(30) NOT NULL,
    città VARCHAR(20) NOT NULL,
    PRIMARY KEY (id_aeroporto),
);

CREATE TABLE viaggiatore (
    cf VARCHAR(20) NOT NULL,
    nome VARCHAR(20) NOT NULL,
    cognome VARCHAR(20) NOT NULL,
    età INT NULL,
    PRIMARY KEY (cf),
);

CREATE TABLE biglietto (
    ticket_cod VARCHAR(5) NOT NULL,
    tipo_bagaglio ENUM(S, M, L) NOT NULL DEFAULT S,
    prezzo DECIMAL(2) NOT NULL,
    numPosto VARCHAR(4) NOT NULL,
    viaggiatore_cf VARCHAR(20) NOT NULL,
    volo_numVolo INT NOT NULL,
    PRIMARY KEY (ticket_cod),
    FOREIGN KEY (viaggiatore_cf) REFERENCES viaggiatore (cf) ON UPDATE cascade ON DELETE ,
    FOREIGN KEY (volo_numVolo) REFERENCES volo (numVolo) ON DELETE cascadeON UPDATE cascade
);

CREATE TABLE compagnia_aerea (
    id_compagnia INT NOT NULL,
    nome VARCHAR(30) NOT NULL,
    PRIMARY KEY (id_compagnia),
);

CREATE TABLE impiegato (
    id_impiegato INT NOT NULL,
    Anni lavoro INT NOT NULL,
    Stipendio INT NOT NULL,
    Num voli INT NOT NULL,
    Tipo ENUM("Pilota", "Assistente") NOT NULL,
    compagnia INT NOT NULL,
    PRIMARY KEY (id_impiegato),
    FOREIGN KEY (compagnia) REFERENCES compagnia_aerea (id_compagnia)ON UPDATE cascade ON DELETE cascade
);

CREATE TABLE lingua (
    Nome Lingua INT NOT NULL,
    PRIMARY KEY (Nome Lingua),
);
```

```

CREATE TABLE conosce (
    impiegato_id_impiegato INT NOT NULL,
    lingua_Nome Lingua INT NOT NULL,
    PRIMARY KEY (impiegato_id_impiegato, lingua_Nome Lingua),
    FOREIGN KEY (impiegato_id_impiegato) REFERENCES impiegato (id_impiegato) ON UPDATE cascade ON DELETE cascade ,
    FOREIGN KEY (lingua_Nome Lingua) REFERENCES lingua (Nome Lingua) ON UPDATE cascade ON DELETE cascade,
);

CREATE TABLE numero_tel (
    numero INT NOT NULL,
    impiegato INT NOT NULL,
    PRIMARY KEY (numero),
    FOREIGN KEY (impiegato) REFERENCES impiegato (id_impiegato) ON UPDATE cascade ON DELETE cascade,
);

CREATE TABLE Economy (
    buono ENUM("Family", "Student", "U12") NOT NULL,
    biglietto_ticket_cod VARCHAR(5) NOT NULL,
    PRIMARY KEY (biglietto_ticket_cod),
    FOREIGN KEY (biglietto_ticket_cod) REFERENCES biglietto (ticket_cod) ON UPDATE cascade ON DELETE cascade,
);

CREATE TABLE First Class (
    biglietto_ticket_cod VARCHAR(5) NOT NULL,
    PRIMARY KEY (biglietto_ticket_cod),
    FOREIGN KEY (biglietto_ticket_cod) REFERENCES biglietto (ticket_cod) ON UPDATE cascade ON DELETE cascade,
);

CREATE TABLE benefits (
    benefit VARCHAR(20) NOT NULL,
    First Class_biglietto_ticket_cod VARCHAR(5) NOT NULL,
    PRIMARY KEY (First Class_biglietto_ticket_cod),
    FOREIGN KEY (First Class_biglietto_ticket_cod)
    REFERENCES First Class (biglietto_ticket_cod) ON UPDATE cascade ON DELETE cascade,
);

```

```

CREATE TABLE volo (
    numVolo INT NOT NULL,
    aeroporto_partenza VARCHAR(30) NOT NULL,
    aereoporto_destinazione VARCHAR(30) NOT NULL,
    ora_partenza TIME NOT NULL,
    data_partenza DATE NOT NULL,
    numPassegeri INT NOT NULL,
    durata TIME NOT NULL,
    id_compagnia INT NOT NULL,
    PRIMARY KEY (numVolo),
    FOREIGN KEY (id_compagnia) REFERENCES compagnia_aerea (id_compagnia) ON UPDATE NO ACTION ON DELETE NO ACTION,
);

```

Punto VI: Operazioni SQL

OPERAZIONE I) *Inserimento di un viaggiatore*

```
INSERT INTO viaggiatore(cf,nome,cognome,età)
VALUES
(
    ?,?,?,?
);
```

```
("PTFRBH95T42F832K", "Giovanni", "Giorgio",35),
("BDMFPM38D50F976N","Miriam","Camaleonte",21),
("RMNPMQ71E23E660V", "Emilia","Caglio",11),
("FVRMQT39A58F392Q", "Anna","Bolla",30),
("LNSXNP34M48Z332X", "Alessio","Spadafora",19),
("FAAPWQ47L46B099I", "Alberto","Tatiana",61);
```

OPERAZIONE II) *Stampare numero voli di durata inferiore alle due ore e relativo ID impiegato*

```
SELECT COUNT(*)
    AS 'NumVoli sotto le 2 ore', id_impiegato,Tipo
FROM volo AS V, impiegato AS I
WHERE V.id_compagnia = I.compagnia AND Tipo='Pilota' AND V.durata<'02:00'
GROUP BY id_impiegato;
```

OPERAZIONE III) *Rimuovere impiegati in base agli anni di lavoro*

```
DELETE FROM impiegato
    WHERE AnniLavoro > ?;
```

OPERAZIONE IV) *Modifica prezzo di un biglietto in base al posto*

```
UPDATE biglietto AS b
    SET b.prezzo = ?
    WHERE b.numPosto = ?;
```

OPERAZIONE V) *Stampare il nome di un dato aeroporto*

```
SELECT nome
    FROM aeroporto
    WHERE nome id_aeroporto = ?;
```


OPERAZIONE VI) *Stampare tabella impiegato*

```
SELECT *  
FROM volo INNER JOIN  
(  
  SELECT id_impiegato, Tipo, compagnia  
    FROM impiegato  
   WHERE id_impiegato=123  
)  
AS tbl ON volo.id_compagnia=tbl.compagnia;
```

OPERAZIONE VII) *Stampare privilegi dato un ticket cod.*

```
SELECT benefit  
FROM benefits  
WHERE ticket_cod=?;
```

OPERAZIONE VIII) *Dato nome e cognome di un viaggiatore, stampare i relativi voli*

```
SELECT *  
✓ FROM volo INNER JOIN  
  (  
    SELECT volo_numVolo, cognome  
    FROM viaggiatore v  
    INNER JOIN biglietto b ON v.cf=b.viaggiatore_cf  
✓ WHERE nome=? && cognome=?  
  )  
  AS tbl ON volo.numVolo=tbl.volo_numVolo;
```

```
SELECT v.numVolo, vi.cognome  
  FROM volo AS v, viaggiatore AS vi, biglietto AS b  
 WHERE vi.cf = b.viaggiatore_cf AND b.numVolo=v.numVolo  
    AND vi.nome = ?  
    AND vi.cognome = ?;
```

OPERAZIONE IX) Stampare numero passeggeri di ogni volo

```
SELECT numPassegeri,numVolo
FROM volo;
```

OPERAZIONE X) Stampare il totale speso da un viaggiatore

```
Select SUM (prezzo) AS 'totale speso'
FROM biglietto b INNER JOIN
(
  SELECT cf
  FROM viaggiatore
  WHERE nome=? && cognome=?
)
AS tbl ON b.viaggiatore_cf=tbl.cf;
```

```
SELECT SUM(prezzo) AS 'totale speso', V.cf
      FROM viaggiatore AS V, biglietto AS b
      WHERE V.cf=B.viaggiatore_cf AND V.cf = ?;
```

OPERAZIONE XI) Controllo validità buono U12

```
SELECT nome, cognome, età
FROM viaggiatore v
INNER JOIN cf c
ON v.cf=c.viaggiatore_cf;
CREATE VIEW cf
AS select viaggiatore_cf
FROM biglietto
INNER JOIN (
  SELECT biglietto_ticket_cod
  FROM economy
  WHERE buono='U12'
)
AS tbl ON biglietto.ticket_cod=tbl.biglietto_ticket_cod;
```

```
SELECT V.nome, V.cognome, V.età
      FROM viaggiatore AS V, biglietto AS B, economy AS E,
      WHERE E.buono='U12' AND B.ticket_cod=E.biglietto_ticket_cod AND B.viaggiatore_cf=V.cf
```

OPERAZIONE XII) *Aumentare Stipendio di un dato impiegato se conosce due o più lingue*

```
UPDATE impiegato AS I
  SET I.stipendio = (I.stipendio + 200)
  WHERE I.id_impiegato = ?
  AND EXISTS(
    SELECT C.id_Impiegato
    FROM conosce AS C
    WHERE C.id_impiegato = I.id_impiegato
    GROUP BY C.id_impiegato
    HAVING COUNT(*)>=2);
```

OPERAZIONE XIII) *Dato il numero di un volo stampare la città in cui si trova l'aeroporto di partenza*

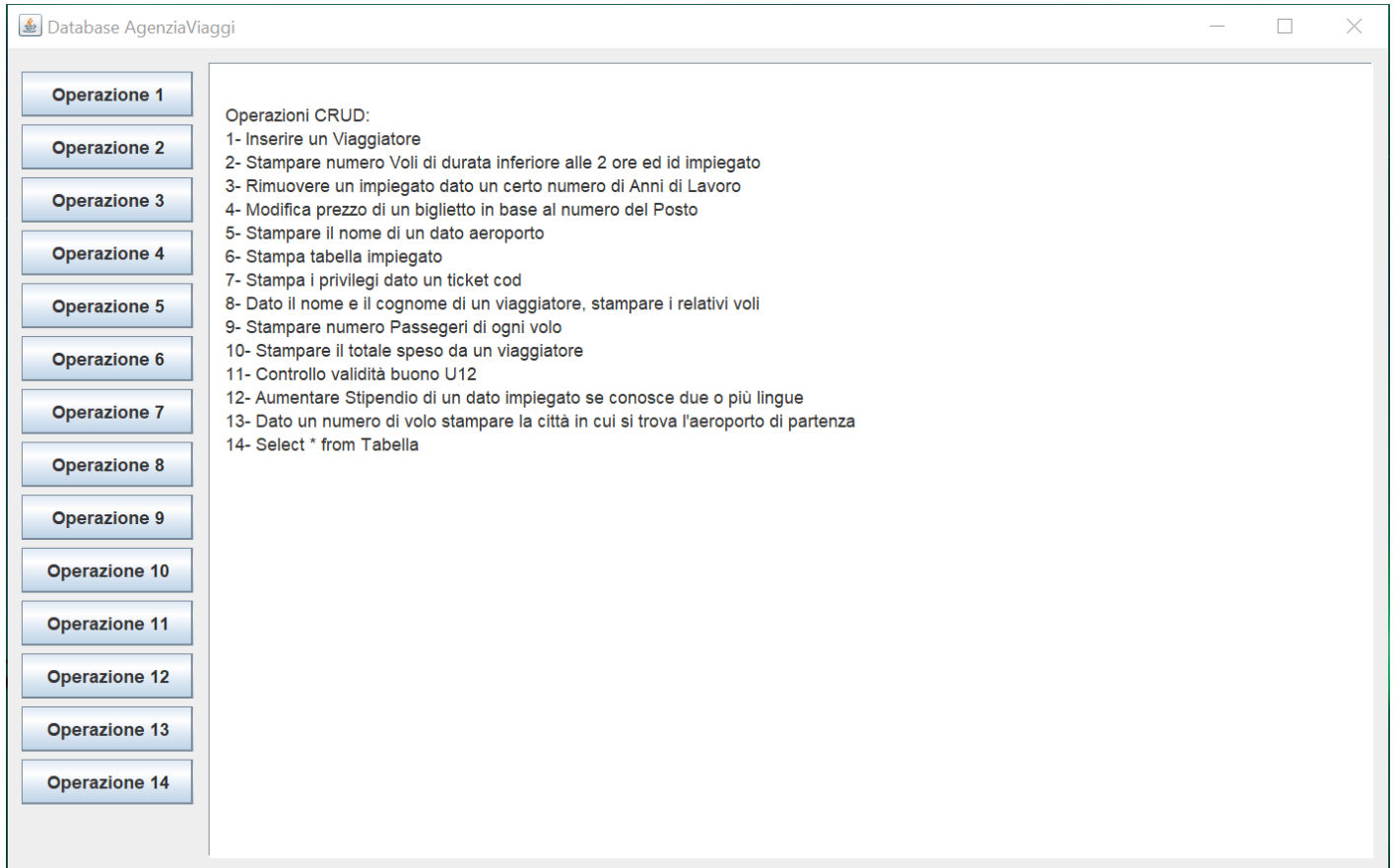
```
SELECT v.aeroporto_partenza,a.città
FROM volo AS v, aeroporto AS a
WHERE v.numVolo=? AND v.aeroporto_partenza=a.id_aeroporto;
```

OPERAZIONE XIII) *Stampa una tabella*

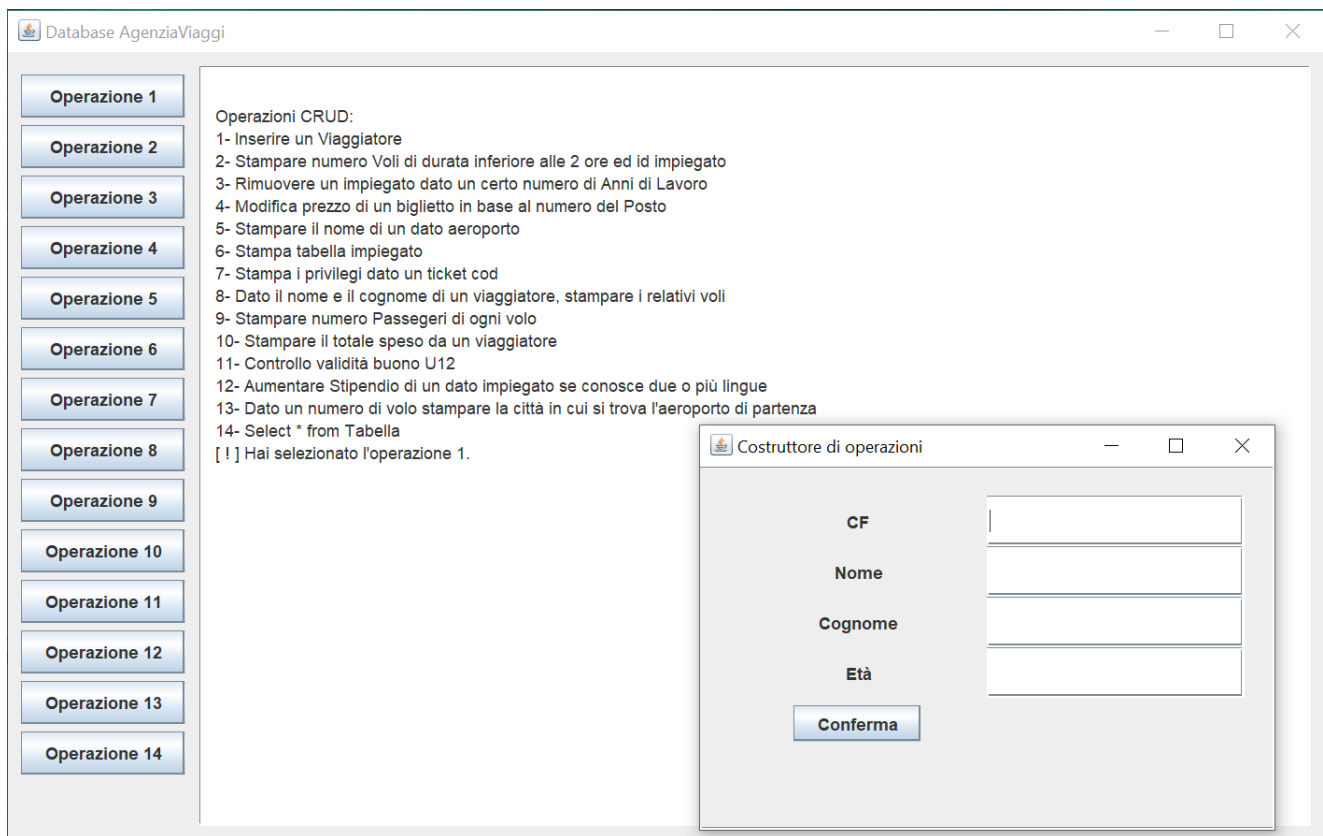
```
SELECT * FROM ?;
```

Punto VII: Applicazione Java e test

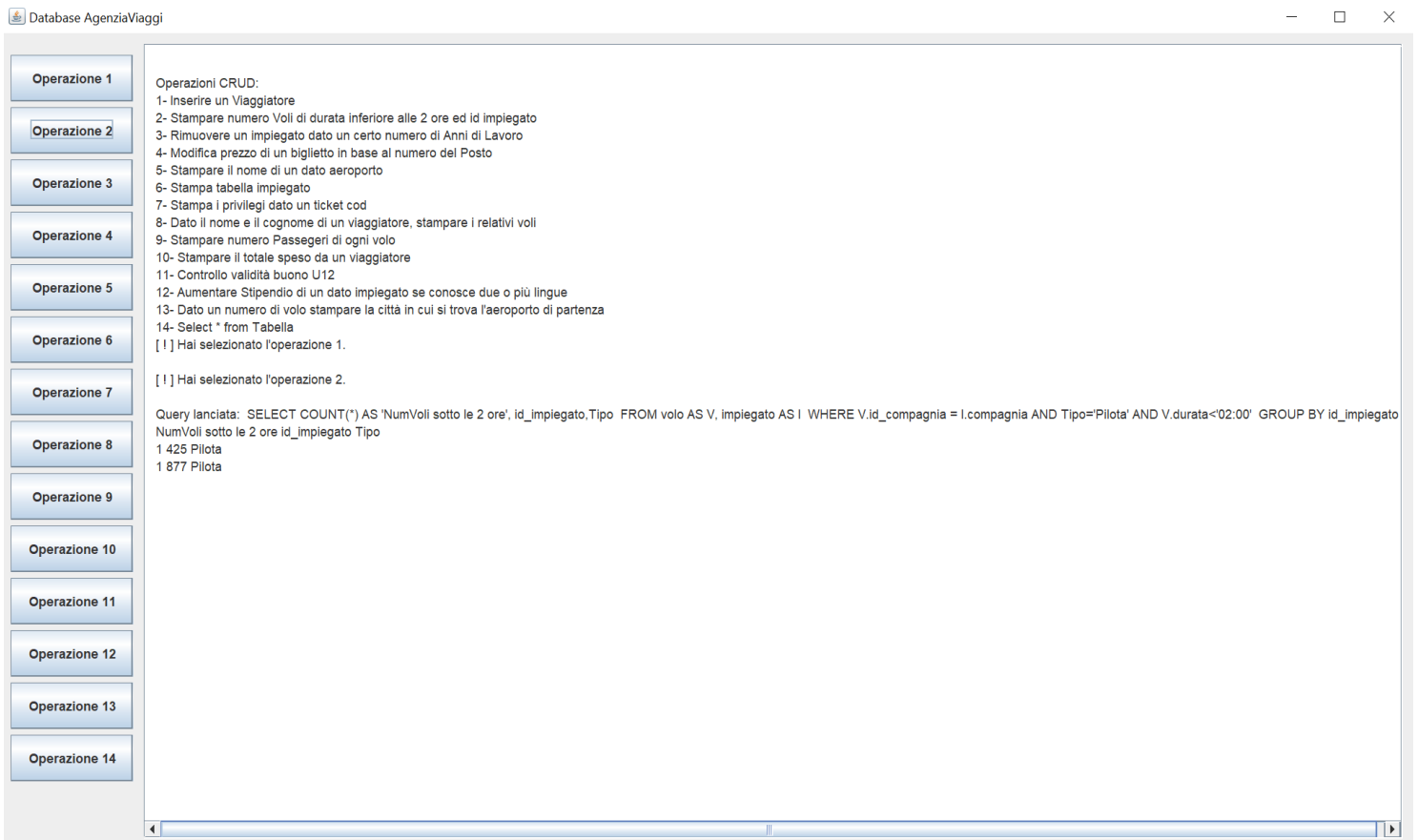
All'avvio dell'applicazione JAVA, attraverso il seguente frame è possibile svolgere le operazioni precedentemente citate, con il supporto di una interfaccia grafica, dove sono mostrate in maniera sintetica le operazioni eseguibili.



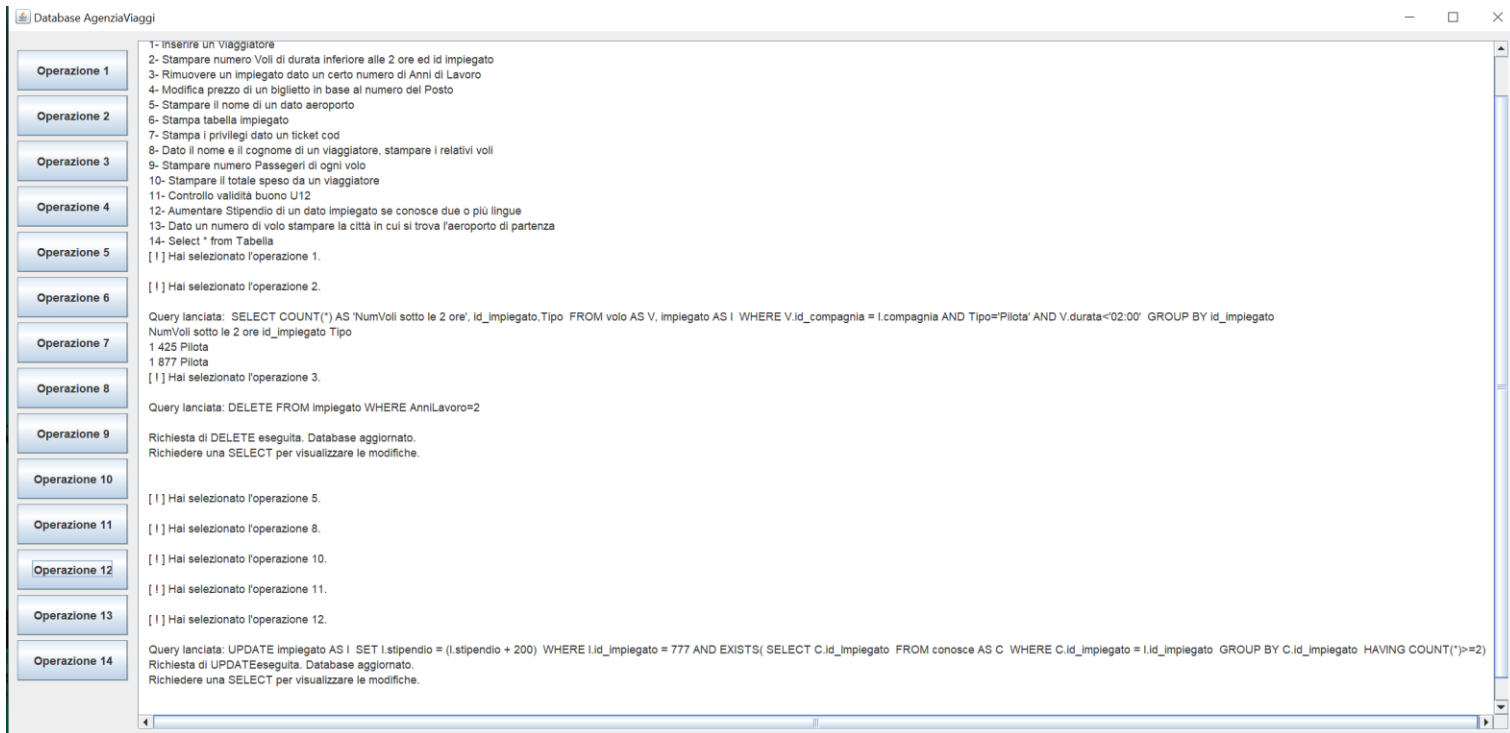
Per esempio, alla richiesta dell'esecuzione dell'operazione 1, verrà generata un ulteriore frame dove inserire i campi necessari all'entità.



Per quanto riguarda le operazioni di tipo Select (es. Operazione 11), che stampano dei dati, questi vengono mostrati all'interno del frame principale, in questo modo:



L'applicazione stampa anche l'esito delle operazioni. Questo è particolarmente importante nei casi di UPDATE, dove è necessario controllare che l'operazione abbia prodotto gli effetti desiderati e che non ci siano effetti collaterali.



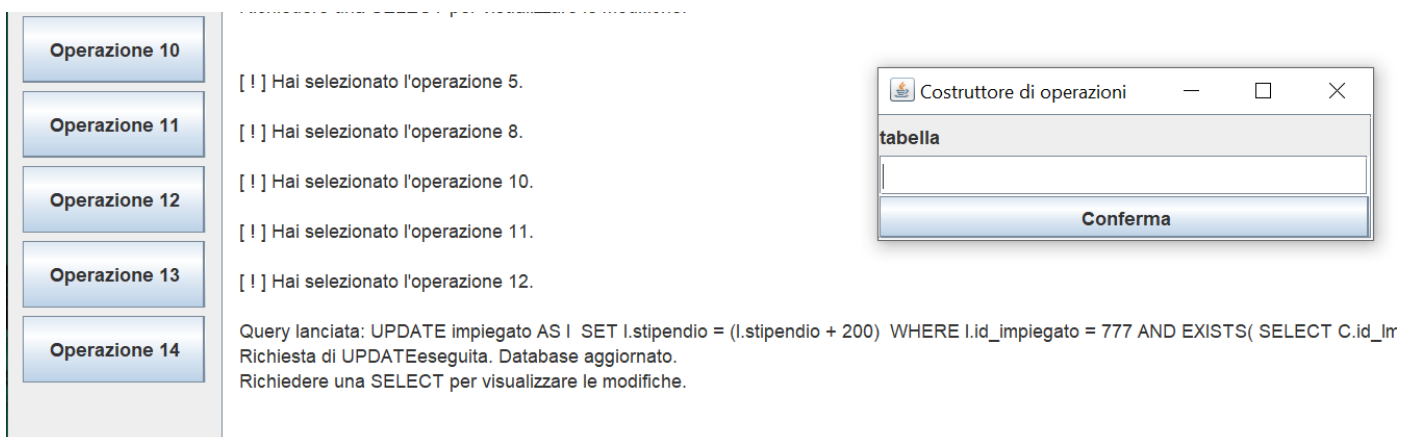
Tuttavia lo stesso ragionamento può essere applicato per le SELECT e le DELETE:

Query lanciata: DELETE FROM impiegato WHERE AnniLavoro=2

Richiesta di DELETE eseguita. Database aggiornato.

Richiedere una SELECT per visualizzare le modifiche.

Dunque, a tal proposito è stata inserita l'operazione 14 che stampa una tabella a scelta.



Ad esempio:

The screenshot shows a database application interface. On the left, there is a vertical sidebar with four buttons labeled "Operazione 11", "Operazione 12", "Operazione 13", and "Operazione 14". The "Operazione 14" button is highlighted with a blue border. To the right of the sidebar, the main area displays the text "Query lanciata: SELECT * FROM lingua" followed by a list of language names: "Nome Lingua", "Francese", "Inglese", "Italiano", "Russo", "Spagnolo", and "Tedesco". At the bottom of the main area, there is a horizontal scrollbar with a blue track and a white slider.

Nome Lingua
Francese
Inglese
Italiano
Russo
Spagnolo
Tedesco