

# 1 Abstract

Si vuole rinnovare la base di dati di un'applicazione che offre ai suoi clienti la possibilità di cercare e confrontare diversi pacchetti di viaggio proposti dalle varie agenzie. Il precedente sistema contava all'incirca 50.000 utenti, la nuova base di dati deve supportare un nuovo bacino di possibili clienti, nell'ordine dei 500.000 utenti. All'intero dell'applicazione esistono tre attori principali: i clienti, le agenzie e le compagnie aeree.

Il cliente può: visualizzare le soluzioni di viaggio disponibili, prenotare dei pacchetti, scegliere il metodo di trasporto più adeguato per le sue esigenze e vedere lo storico degli acquisti. Durante la scelta del trasporto il sistema consiglia i voli più convenienti tra quelli presenti nel database. Il sistema ha il principale scopo di tenere traccia delle prenotazioni e delle informazioni utili al fine di guidare il cliente all'acquisto del pacchetto più adatto alle sue esigenze. Una agenzia fornisce: i pacchetti che il cliente può prenotare, e tutte le informazioni essenziali per quel pacchetto: polizza assicurativa, informazioni sull'alloggio e descrizioni testuali utili a dare un'idea del viaggio. Vengono registrati i voli offerti dalle compagnie aeree che collaborano con il servizio, così da poter calcolare durante la prenotazione il volo più conveniente, ma comunque lasciando all'utente scegliere.

Nel sistema chiunque può visualizzare i pacchetti di viaggio, anche chi non è registrato al servizio.

## 2 Raccolta e analisi dei requisiti

### 2.1 Proprietà del sistema

Gli **utenti** registrati nel sistema vengono identificati dalla loro email scelta durante la fase di registrazione. Di ogni utente vengono memorizzati i seguenti dati: la password e la data di iscrizione. Gli utenti si specializzano in tre categorie: i *clienti*, le *agenzie* e le *compagnie di volo*. Ogni **cliente** deve fornire le seguenti informazioni: il nome, il cognome, la data di nascita, in modo facoltativo il sesso e il numero di telefono. Di ogni **agenzia** viene riportato: la denominazione e la sede legale con l'indirizzo. Ogni **compagnia di volo** riporta: il nome, il codice internazionale ICAO e i voli che gestisce.

Ogni agenzia può offrire diversi *pacchetti viaggio*. Per ogni **pacchetto** vengono salvati i seguenti dati: la data di partenza, la data di ritorno, la disponibilità<sup>1</sup>, il massimo numero di persone che possono partecipare al viaggio<sup>2</sup>, il prezzo di base<sup>3</sup>, la destinazione e un ID per identificare univocamente il pacchetto. Ogni pacchetto riporta i dati dell'alloggio, identificato dal suo nome e dalla città in cui è ubicato con l'indirizzo, in più viene riportata la tipologia di struttura<sup>4</sup> e il numero di stelle se disponibile. Sia il pacchetto di viaggio e sia l'alloggio hanno una **descrizione** testuale che viene identificata nel sistema da un ID, inoltre viene riportato: un titolo della descrizione (o del viaggio) e un testo.

Ogni compagnia aerea gestisce dei voli. I **voli** sono identificati da un codice di volo e riportano le seguenti informazioni: la classe, come va fatto il check-in e il prezzo. Per ogni volo vengono riportati anche le **informazioni** riguardanti i **bagagli**: se concesso, quanto può pesare al massimo il bagaglio da mettere in stiva, e se concesso, quanto può pesare al massimo il bagaglio da portare a mano. Ogni volo ha un **aeroporto** di partenza e un arrivo identificato dal suo codice internazionale, inoltre viene tenuta traccia dell'ora e della data di partenza stimata e dell'ora e della data di arrivo stimata. Di ogni **aeroporto** viene salvata la sua ubicazione.

---

1 Quanti utenti al massimo possono comprare quel pacchetto.

2 Esempio: pacchetto famiglia da massimo 4 persone.

3 Senza contare il costo dei mezzi di trasporto per l'andata e il ritorno.

4 Esempio: hotel, bed & breakfast, eccetera.

Un cliente può scrivere un *recensione* per l'alloggio alla fine del viaggio. Le **recensioni** sono identificate da un **ID** interno e riportano: un giudizio con una scala da 0 a 5 e una motivazione testuale che può essere facoltativa. Un cliente può prenotare un pacchetto, nella **prenotazione** vengono salvati: il numero di persone che partecipano al viaggio, i dati della transizione di pagamento e le informazioni per il trasporto (sia per l'andata che per il ritorno). Una **transizione** riporta: un codice identificativo, la banca che ha preso in carico l'operazione, l'importo totale, il circuito usato e il timestamp in cui è avvenuta l'operazione. Per ogni prenotazione si può scegliere anche l'offerta più conveniente di trasporto. Le **informazioni di trasporto** riportano: il prezzo totale e le varie tratte per l'andata e per ritorno. Le tratte sono rappresentate dai voli.

Ogni luogo è riconosciuto dal sistema come una **città** identificata da un codice interno e vengono salvati: il nome e il paese dove si trova la città.

## 2.2 Glossario dei termini

Table I: Glossario dei termini

Termine	Descrizione	Sinonimo	Collegamenti
<i>Utente</i>	Utente generico iscritto al sistema		Cliente, Agenzia
<i>Cliente</i>	Specializzazione di un utente. Usuiscono del servizio		Utente
<i>Agenzia</i>	Specializzazione di un utente. Può inserire delle soluzioni di viaggio nel sistema		Utente, Pacchetto
<i>Compagnia di volo</i>	Specializzazione di un utente. Può inserire dei voli nel sistema	Compagnia aerea, Compagnia	Utente, Volo
<i>Pacchetto di viaggio</i>	Soluzione di viaggio offerta da una agenzia	Pacchetto, Soluzione di viaggio	Utente, Agenzia, Alloggio, Descrizione
<i>Alloggio</i>	Struttura che ospita il cliente durante la vacanza	Soggiorno	Città, Descrizione, Pacchetto viaggio
<i>Descrizione</i>	Descrizione testuale di un alloggio oppure di un pacchetto		Pacchetto, Alloggio
<i>Recensione</i>	Giudizio del cliente sull'alloggio offerto		Cliente, Alloggio
<i>Prenotazione</i>	Acquisto con esito positivo di un pacchetto		Cliente, Pacchetto
<i>Transizione</i>	Pagamento avvenuto con successo		Prenotazione
<i>Polizza</i>	Assicurazione per il viaggio		Pacchetto Viaggio
<i>Informazioni di trasporto</i>	Informazioni riguardanti i voli da prendere per andare e tornare dal viaggio		Prenotazione, Volo
<i>Aeroporto</i>	Luogo di partenza e arrivo degli aeri		Città, Volo
<i>Volo</i>	Volo per arrivare a destinazione o tornare	Tratte	Informazioni bagaglio, Aeroporto, Informazioni di trasporto, Compagnia di volo
<i>Informazioni bagaglio</i>	Informazioni utili al cliente sulle politiche usate per la gestione dei bagagli per un volo		Volo
<i>Città</i>	Luogo fisico		Alloggio, Aeroporto

## 2.3 Operazioni

Nel caso d'uso perso in esame il numero di operazioni effettuate non hanno una distribuzione uniforme durante tutto l'anno, ma alcune operazioni in particolare presentano un numero di richieste maggiore

durante i periodi di vacanza, cioè durante i periodi di massimo carico per il sistema, mentre in altri periodi ci sono momenti di *idle*. Ipotizziamo di seguito per le operazioni più importanti la loro frequenza.

*Table II: operazioni e costi*

<b>Numero operazione</b>	<b>Operazione</b>	<b>Descrizione</b>	<b>Numero operazioni (tempo/operazione)<sup>5</sup></b>
<i>1</i>	Inserimento pacchetto	Inserimento di un pacchetto da parte di un'agenzia	30 o/dd
<i>2</i>	Inserimento volo	Inserimento di un volo da parte di una compagnia aerea	12 000 o/dd
<i>3</i>	Inserimento cliente	Un nuovo cliente si iscrive al servizio	1 500 o/dd
<i>4</i>	Ricerca pacchetti	Consultazione dei pacchetti disponibili	260 000 o/dd
<i>5</i>	Prenotazione pacchetto	Un cliente compra una soluzione viaggio	10 000 o/dd
<i>6</i>	Controllo storico acquisti	Un cliente controlla lo storico degli acquisti	3 000 o/mm

### 3 Progettazione concettuale

---

<sup>5</sup> Riportiamo le misure di tempo: *dd* = giorni, *mm* = mesi e *yy* = anni.



	Sesso	enum('M','F')	Può essere <b>NULL</b>
	Numero telefonico	varchar(20)	Può essere <b>NULL</b>
Utente ➔ Agenzia	Denominazione	varchar(40)	
Utente ➔ Compagnia	Nome	varchar(40)	
	ICAO	char(4)	
Recensione	<u>ID</u>	integer	
	Giudizio	numeric(1,0)	∈ [0,5]
	Motivazione	varchar(200)	Può essere <b>NULL</b>
Prenotazione	<u>Codice</u> (Transazione)	varchar(16)	length(password) = 16
	Numero persone	numeric(2,0)	≥ 1
Transizione	Codice	varchar(16)	length(password) = 16
	Banca	varchar(20)	
	Importo	numeric(5,2)	≥ 0
	Circuito	varchar(10)	
	Data ora	timestamp	
Pacchetto Viaggio	<u>ID</u>	integer	
	Data di partenza	date	> Data di ritorno
	Data di ritorno	date	
	Disponibilità	integer	≥ 1
	N° persone	integer	≥ 1
	Prezzo	numeric(5,2)	≥ 0
Descrizione	<u>ID</u>	integer	
	Titolo	varchar(20)	
	Testo	varchar(400)	
Alloggio	<u>Nome</u>	varchar(25)	
	<u>ID</u> (Città)	integer	
	Stelle	integer	∈ [1,5] e può essere <b>NULL</b>
	Tipologia	varchar(10)	
Città	<u>ID</u>	integer	
	Nome	varchar(25)	
	Stato	varchar(25)	
Polizza	<u>ID</u>	integer	
	Descrizione	varchar(400)	
	Titolo	varchar(20)	
Informazioni trasporto	<u>Codice</u> (Transizione)	timestamp	length(password) = 16
	Prezzo totale	numeric(5,2)	≥ 0
Volo	Codice	integer	
	Classe	varchar(10)	
	Check-in	varchar(10)	
	Prezzo	numeric(5,2)	≥ 0
Informazioni Bagaglio	<u>ID</u>	integer	
	Max bagaglio a mano	numeric(2,0)	≥ 0 e può essere <b>NULL</b>
	Max bagaglio in stiva	numeric(2,0)	≥ 0 e può essere <b>NULL</b>
Aeroporto	<u>Codice</u>	varchar(4)	

## 3.2 Descrizione relazioni E-R

Table IV: relazioni E-R concettuale.

Relazione	Entità coinvolte	Descrizione	Attributi
Offerte Viaggio	Agenzia (0,N) Pacchetto Viaggio (1,1)	Una agenzia può offrire da 0 a N pacchetti viaggio. Un pacchetto viene offerto da una sola agenzia	
Recensione Cliente	Cliente (0,N) Recensione (1,1)	Un cliente può scrivere da 0 a N recensioni. Ogni recensione deve fare riferimento a un cliente	
Recensione Alloggio	Recensione (1,1) Alloggio (0,N)	Una recensione deve fare sempre fare riferimento ad un unico	

		alloggio, mentre un alloggio può avere da 0 a N recensioni	
<i>Assicurazione</i>	Pacchetto Viaggio (0,1) Polizza (1,N)	Un pacchetto viaggio può avere una polizza assicurativa. Una polizza assicurativa può essere usata per più pacchetti oppure per uno soltanto	
<i>Descrizione Viaggio</i>	Pacchetto Viaggio (1,1) Descrizione (0,1)	Ogni pacchetto viaggio deve avere una descrizione. Una descrizione non necessariamente deve fare riferimento a un pacchetto	
<i>Descrizione Alloggio</i>	Descrizione (0,1) Alloggio (1,1)	Ogni alloggio deve avere una descrizione. Una descrizione non necessariamente deve fare riferimento a un alloggio	
<i>Destinazione</i>	Pacchetto viaggio (1,1) Città (0,N)	Un pacchetto ha una sola destinazione. Una città può essere più volte destinazione di un viaggio oppure non è mai una destinazione di un viaggio	
<i>Informazioni Alloggio</i>	Pacchetto Viaggio (1,1) Alloggio (1,N)	Ogni pacchetto viaggio deve riferirsi ad un alloggio. Un alloggio può essere riferito da più pacchetti	
<i>Sede Alloggio</i>	Alloggio (1,1) Città (0,N)	Ogni alloggio (fisico) ha una sola ubicazione. Una città può essere sede di più alloggi oppure di nessun alloggio	Indirizzo – varchar(40)
<i>Sede Agenzia</i>	Agenzia (1,1) Città (0,N)	Un agenzia ha una sola città come sede. Una città può essere la sede di più agenzia oppure di nessuna	Indirizzo – varchar(40)
<i>Prenotazione Cliente</i>	Cliente (0,N) Prenotazione (1,1)	Un cliente può eseguire da 0 a N prenotazioni. Ogni prenotazione deve riferirsi a un cliente	
<i>Prenotazione Pacchetto</i>	Prenotazione (1,1) Pacchetto Viaggio (0,N)	Una prenotazione deve riferirsi a un pacchetto. Un pacchetto può apparire da 0 a N volte in una prenotazione	
<i>Pagamento</i>	Prenotazione (1,1) Transizione (1,1)	Ogni prenotazione ha un transizione e viceversa	
<i>Trasporto</i>	Prenotazione (0,1) Informazioni trasporto (1,1)	Una prenotazione può usufruire del trasporto. Le informazioni di trasporto devono fare riferimento ad una prenotazione	
<i>Ritorno</i>	Informazioni trasporto (1,N) Volo (0,1)	Una informazioni di trasporto può avere più voli di ritorno. Un volo può presentarsi più volte oppure nessuna	
<i>Andata</i>	Informazioni trasporto (1,N) Volo (0,1)	Una informazioni di trasporto può avere più voli di andata. Un volo può presentarsi più volte oppure nessuna	
<i>Gestisce</i>	Compagnia (1,N) Volo (1,1)	Una compagnia può gestire più voli. Ogni volo deve essere gestito da una compagnia	
<i>Dettagli Volo</i>	Informazioni Bagaglio (1,N) Volo (1,1)	Ogni volo ha le informazioni riguardanti il trasporto dei bagagli. Ogni informazione sui bagagli può essere riferita a più	

		voli	
<i>Partenza Volo</i>	Volo (1,1) Aeroporto (0,N)	Ogni volo ha una aeroporto di partenza. Un aeroporto può essere punto di partenza per un volo	Data ora – timestamp
<i>Arrivo Volo</i>	Volo (1,1) Aeroporto (0,N)	Ogni volo ha una aeroporto di arrivo. Un aeroporto può essere punto di arrivo per un volo	Data ora – timestamp
<i>Aeroporto Città</i>	Aeroporto (1,1) Città (0,N)	Ogni aeroporto è situato in una città. Una città può ospitare da 0 a N aeroporti.	

### 3.3 Regole aziendali<sup>6</sup>

#### Regole di vincolo (RV):

1. un pacchetto non deve avere la *data di ritorno* inferiore alla *data di partenza*;
2. un volo non deve avere il *data ora* dell'arrivo inferiore al *data ora* di partenza;
3. nella relazione andata e ritorno dell'entità informazioni trasporto non devono esserci dei voli che vengono effettuati contemporaneamente;
4. un cliente non deve scrivere una recensione per un'alloggio non presente nello storico dei pacchetti viaggio che ha prenotato;
5. l'attributo *numero di persone* nell'entità prenotazione non deve essere superiore al valore del *N° persone* del pacchetto viaggio prenotato;
6. una descrizione non deve essere in relazione allo stesso tempo sia con il pacchetto viaggio e sia con l'alloggio;
7. la destinazione del pacchetto non deve differire dalla sede dell'alloggio prenotato;
8. la *data di acquisto* della prenotazione non deve differire da il *data ora* di transizione;
9. un pacchetto non deve essere prenotato più volte di quanto riportato nella sua disponibilità;

#### Regole di derivazione (RD):

1. il *prezzo totale* in informazioni trasporto deve essere la somma del prezzo di ogni volo per l'andata e il ritorno;
2. l'*importo* in transizione deve essere la somma del prezzo del pacchetto di viaggio acquisto più il prezzo totale delle informazioni di trasporto.

## 4 Progettazione logica

### 4.1 Ristrutturazione dello schema E-R

#### 4.1.1 Analisi ridondanze

L'attributo *prezzo totale* dell'entità Informazioni Trasporto può essere derivato attraverso la somma dei *prezzi* di ogni volo in relazione con l'entità. L'attributo *importo* dell'entità Transizione può essere calcolato come la somma del *prezzo* del pacchetto acquistato e del *prezzo totale* per il trasporto. La destinazione del pacchetto può essere derivata dalla sede dell'alloggio prenotato. Riportiamo dunque le tavole per le entità e le operazioni di nostro interesse:

<sup>6</sup> O vincoli di integrità. Usiamo la stessa terminologia che viene usata nel libro presentato a inizio corso.

Table V: Tabella dei volumi

Concetto	Tipo	Volume (B)
Transizione	E	760 000
Informazioni	E	754 000
Trasporto		

Table VI: Tabella delle operazioni

Operazione n°	Tipo <sup>7</sup>	Frequenza
5	I	10 000 al giorno
6	I	8 000 al giorno

Prezzo totale è un Numeric(5,2) dunque occupa 5 byte, calcoliamo il volume occupato:  $5B \cdot 760000 = 3,8\text{Mb}$ , mentre le operazioni richieste per il calcolo di questo dato sono la 5 e 6, ipotizziamo che un utente in media prenota 13 pacchetti:

#### Tavole degli accessi in presenza di ridondanza

Table VII: operazione 5

Concetto	Costr.	Acc.	Tipo
Pacchetto Viaggio	E	1	L
Prenotazione	R	1	S
Prenotazione Cliente	R	1	S
Prenotazione Pacchetto	R	1	S
Volo <sup>8</sup>	E	2	L
Informazioni trasporto	E	1	S
Trasporto	R	1	S
Ritorno	R	1	S
Andata	R	1	S
Transazione	E	1	S
Pagamento	R	1	S

Table VIII: operazione 6

Concetto	Costr.	Acc.	Tipo
Cliente	E	1	L
Prenotazione cliente	R	13	L
Prenotazione	E	13	L
Prenotazione Pacchetto	R	13	L
Pacchetto Viaggio	E	13	L
Pagamento	R	13	L
Transazione	E	13	L

#### Tavole degli accessi in assenza della ridondanza

Notiamo che le operazioni della tavola 5 non variano in assenza di ridondanza. Questo perché gli accessi riportati sono sempre li stessi per l'inserimento di una nuova prenotazione.

Table IX: operazione 6

Concetto	Costr.	Acc.	Tipo
Cliente	E	1	L
Prenotazione cliente	R	13	L
Prenotazione	E	13	L
Prenotazione Pacchetto	R	13	L
Pacchetto Viaggio	E	13	L

<sup>7</sup> I: interattiva. B: batch.

<sup>8</sup> La maggior parte dei viaggi ha solo due 2 voli: andata e ritorno.



Ipotizzando che una scrittura equivale a 2 letture abbiamo che: in presenza di ridondanza avvengono 21 accessi per l'operazione 5, mentre 79 accessi in media per l'operazione 6. Otteniamo al giorno quante operazioni in presenza di ridondanza abbiamo:  $21 \cdot 10000 + 79 \cdot 2000 = 368\,000$  accessi in media. Mentre in assenza di ridondanza abbiamo che: l'operazione 5 non varia il suo numero di accessi, mentre l'operazione 6 ha in media 105 accessi. Calcoliamo in assenza di ridondanza il numero medio di accessi:  $21 \cdot 10000 + 105 \cdot 3000 = 525\,000$  accessi. Otteniamo dunque 157 000 accessi in più in assenza di ridondanza.

Possiamo notare che anche *Importo* di transizione richiede la stessa analisi fatta per il *prezzo totale*, dunque otteniamo alla fine eliminando 7,6 Mb risparmiati a discapito di 314 000 accessi aggiuntivi, per un totale di 1 050 000 accessi per entrambi gli attributi. Dato il risparmio è abbastanza irrilevante rispetto a un numero cospicuo di accessi, si è deciso mantenere entrambi gli attributi.

La relazione *Destinazione* tra pacchetto viaggio e città è una ridondanza, decidiamo di rimuoverla visto che: non abbiamo problemi di inconsistenza, non serve applicare più la regola aziendale RV7 e semplifica la struttura della base di dati.

### 4.1.2 Eliminazione generalizzazioni

Notiamo che l'unica generalizzazione presente nel diagramma E-R (figura 1, pagina 4) è totale. La strategia che usiamo è quella dell'*accorpamento del genitore della generalizzazione nelle figlie*. Le altre tecniche sono state escluse perché:

- in questo caso è importante distinguere l'entità figlie per eseguire le operazioni tutte distinte tra loro, dunque è da escludere l'*accorpamento delle figlie della generalizzazione nel genitore*;
- non conviene usare la tecnica di *sostituzione della generalizzazione con associazioni* perché richiede l'aggiunta di vincoli e inoltre aggiunge ulteriori accessi.

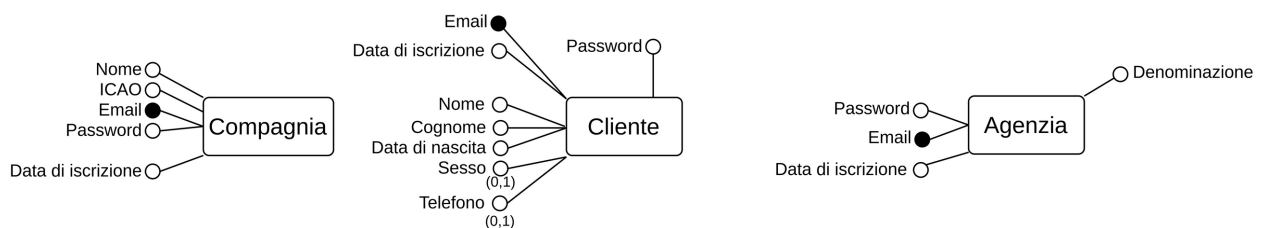


Figure 2: risultato della eliminazione della generalizzazione Utente.

### 4.1.3 Partizionamento accorpamento di entità e associazioni

Gli attributi *Indirizzo* della relazione sede agenzia e sede alloggio potrebbero avere delle ridondanze, ma dato che è estremamente raro e il numero di agenzie è abbastanza contenuto rispetto a quello degli alloggi, si è deciso di lasciare invariata la struttura delle relazioni.

Viene partizionata la nuova struttura dell'entità Cliente in modo da distinguere i dati le cliente da quelli utente ereditati dal genitore, questo con lo scopo di semplificare e separare i concetti di dati dell'utente e dati personali.

### 4.1.4 Scelta degli identificatore principali

Gli identificatori principali rimano sempre quelli segnati sullo schema E-R.

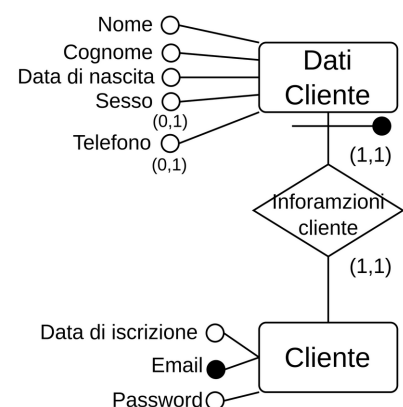


Figure 3: Risultati accorpamento.

## 4.1.5 E-R Logico

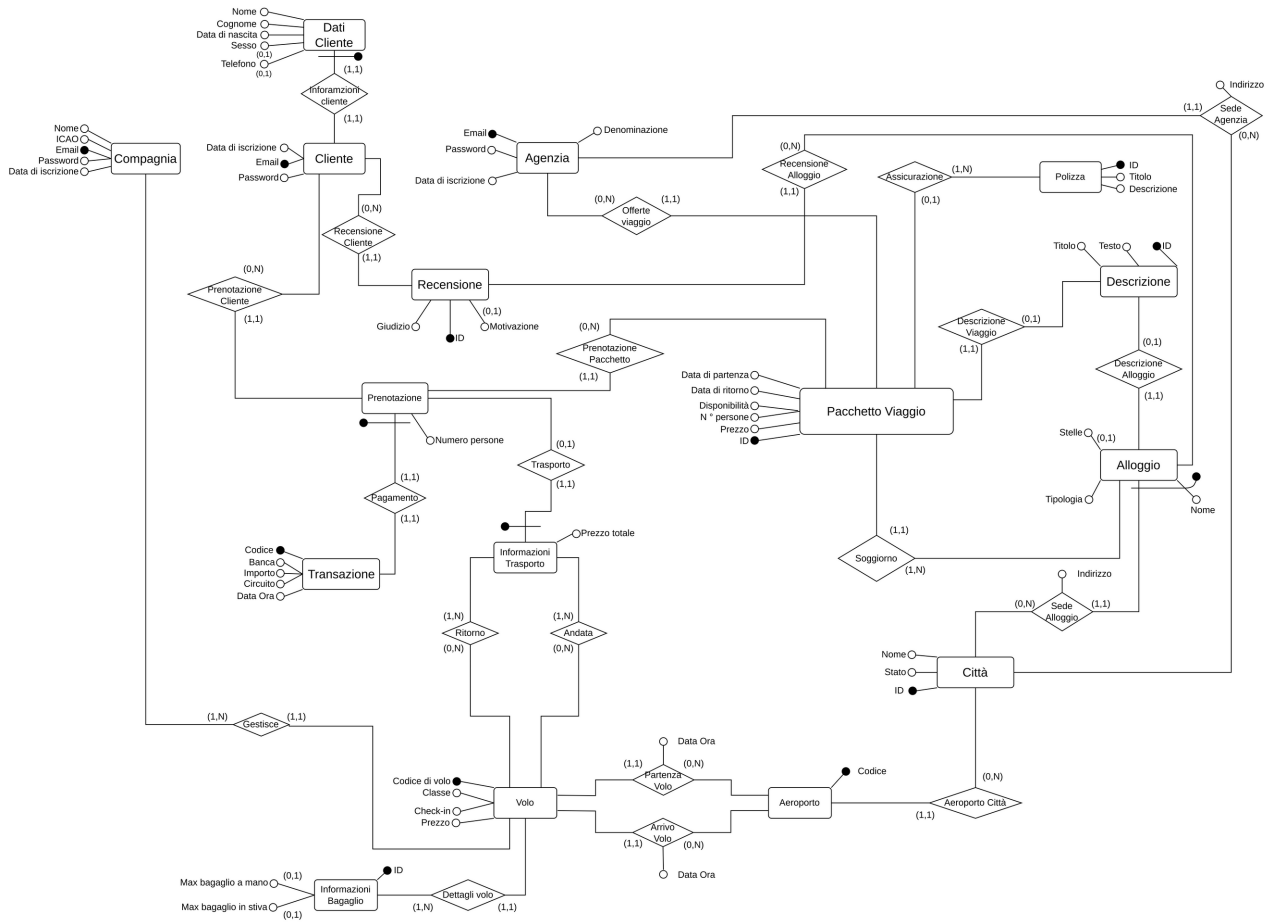


Figure 4: E-R logico.

## 4.2 Traduzione verso il modello relazionale

Usiamo le notazioni usate per l'E-R concettuale. Nei vincoli referenziali indichiamo che un insieme di attributi  $X$  della relazione  $R_1$  è una **chiave referenziale** per la relazione  $R_2$  rispetto alla sua chiave primaria  $K$ :  $R_1(X) \rightarrow R_2(K)$ .

**Cliente**(Email, Password, DataIscrizione)

- $\text{Cliente}(\text{Email}) \rightarrow \text{DatiCliente}(\text{Email})$

**DatiCliente**(Email, Nome, Cognome, DataNascita, Sesso, Telefono)

**Compagnia**(Email, Password, DataIscrizione, Nome, ICAO)

**Volo**<sup>9</sup>(Codice, Classe, CheckIn, Prezzo, EmailCompagnia, AeroportoPartenza, TimestampPartenza, AeroportoArrivo, TimestampArrivo, IDBagagli)

- $\text{Volo}(\text{EmailCompagnia}) \rightarrow \text{Compagnia}(\text{Email})$
- $\text{Volo}(\text{AeroportoPartenza}) \rightarrow \text{Aeroporto}(\text{Codice})$
- $\text{Volo}(\text{AeroportoArrivo}) \rightarrow \text{Aeroporto}(\text{Codice})$

9 Rinominamo l'attributo DataOra ereditato dalla associazione *Partenza Volo* in TimestampPartenza. In modo analogo per l'arrivo.

- Volo(IDBagagli) → InformazioniBagagli(ID)

**InformazioniBagaglio**(ID, BagaglioMano, BagaglioStiva)

**Agenzia**(Email, Password, DataIscrizione, Denominazione, IDCittà, Indirizzo)

- Agenzia(IDCittà) → Città(ID)

**PacchettoViaggio**(ID, Prezzo, NumeroPersone, Disponibilità, DataRitorno, DataPartenza, EmailAgenzia, IDPolizza, IDDescrizione, IDCittàAlloggio, NomeAlloggio)

- PacchettoViaggio(EmailAgenzia) → Agenzia(Email)
- PacchettoViaggio(IDPolizza) → Polizza(ID)
- PacchettoViaggio(IDDescrizione) → Descrizione(ID)
- PacchettoViaggio(IDCittàAlloggio, NomeAlloggio) → Alloggio(IDCittà, Nome)

**Polizza**(ID, Descrizione, Assicuratore)

**Descrizione**(ID, Testo, Titolo)

**Alloggio**(IDCittà, Nome, Stelle, Tipologia, IDDescrizione, Indirizzo)

- Alloggio(IDCittà) → Città(ID)
- Alloggio(IDDescrizione) → Descrizione(ID)

**Città**(ID, Nome, Stato)

**Aeroporto**(Codice, IDCittà)

- Aeroporto(IDCittà) → Città(ID)

**Prenotazione**(CodiceTransizione, EmailCliente, NumeroPersone, IDPacchettoViaggio)

- Prenotazione(CodiceTransizione) → Transizione(Codice)
- Prenotazione(IDPacchettoViaggio) → PacchettoViaggio(ID)

**Transazione**(Codice, Banca, Importo, Circuito, DataOra)

**InformazioniTrasporto**(CodiceTransizione, PrezzoTotale)

- InformazioniTrasporto(CodiceTransizione) → Prenotazione(CodiceTransizione)

**Ritorno**(CodiceTransizione, CodiceVolo)

- Ritorno(CodiceTransizione) → InformazioniTrasporto(CodiceTransizione)
- Ritorno(CodiceVolo) → Volo(Codice)

**Andata**(CodiceTransizione, CodiceVolo)

- Andata(CodiceTransizione) → InformazioniTrasporto(CodiceTransizione)
- Andata(CodiceVolo) → Volo(Codice)

**Recensione**(ID, Giudizio, Motivazione, EmailCliente, IDCittà, NomeAlloggio)

- Recensione(EmailCliente) → Cliente(Email)
- Recensione(IDCittà, NomeAlloggio) → Alloggio(IDCittà, NomeAlloggio)

**5    Query SQL**

**6    Indice**

**7    Client (software C++)**