

# FLAPP - Sistema di gestione per compagnie aeree

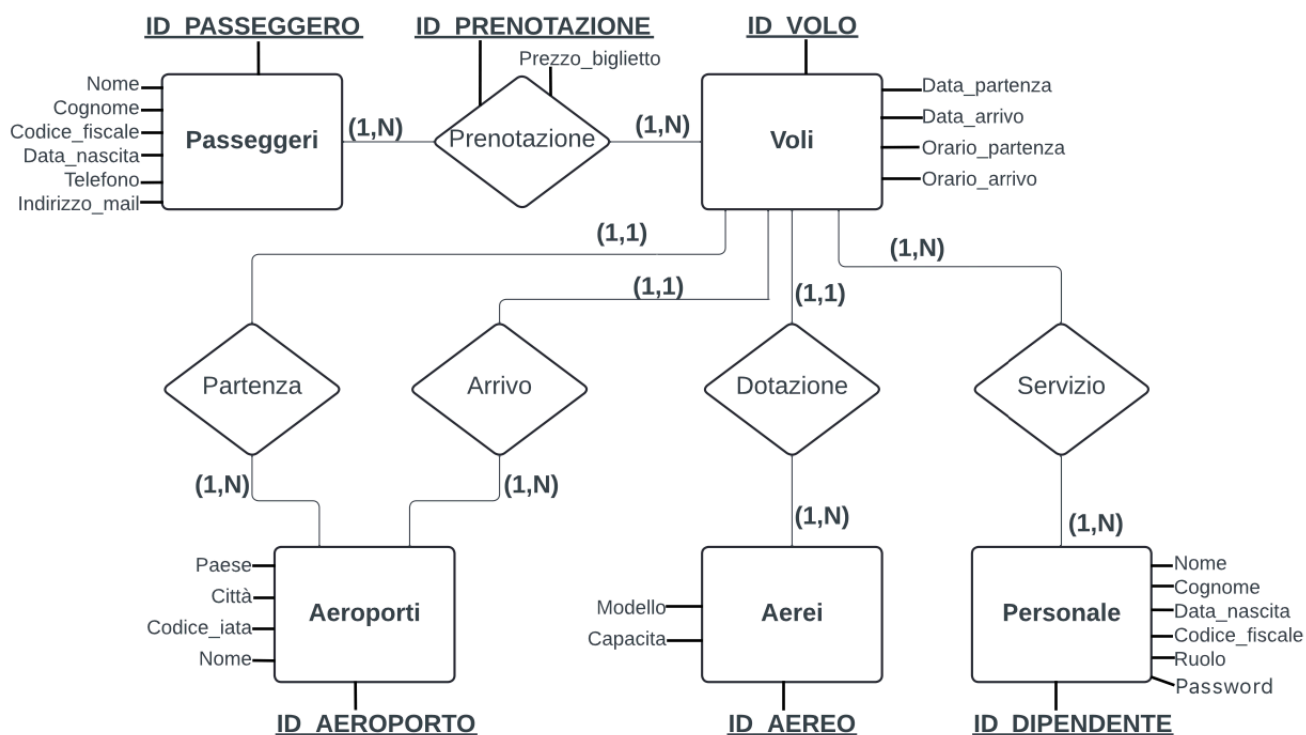
## 1. Introduzione

Flapp è una piattaforma web che consente a clienti e staff di interrogare il database di una compagnia aerea. In particolare offre le seguenti funzionalità:

- Nell'area utente, i clienti possono visualizzare i dettagli delle prenotazioni in base al codice prenotazione. Inoltre, possono inserire il proprio codice utente per vedere le proprie prenotazioni ed eventualmente modificarle.
- Nell'area staff, si possono visualizzare i voli su cui un operatore ha lavorato e anche le ore di lavoro svolte.
- Il pubblico generale può visualizzare statistiche come gli aeroporti con più arrivi o partenze e infine, può lasciare una recensione del sito web.

## 2. Schema concettuale

Le entità principali del database sono Passeggeri, Prenotazioni, Voli, Aerei, Aeroporti, Servizio, e Personale. I Passeggeri effettuano una o più Prenotazioni, associate ai Voli che partono e arrivano in specifici Aeroporti. Ogni volo è operato da un Aereo e servito dal Personale, che comprende dipendenti con ruoli definiti. Gli aeroporti sono identificati da un codice IATA e localizzati in specifiche città e paesi. Gli attributi dettagliati aiutano a gestire le informazioni operative. Le cardinalità indicano il numero di istanze delle entità coinvolte nelle relazioni, ad esempio, un volo può essere associato a molti passeggeri e prenotazioni, ma ogni prenotazione è legata a un solo volo.



### 3. Schema logico

Passeggeri (**ID\_PASSEGGERO**, Nome, Cognome, Codice\_fiscale, Data\_nascita, Telefono, Indirizzo\_mail)  
Voli (**ID\_VOLO**, Data\_partenza, Data\_arrivo, Orario\_partenza, Orario\_arrivo, Durata, Aeroporto\_partenza, Aeroporto\_arrivo, Aereo)  
Aeroporti ( **ID\_AEROPORTO**, Paese, Città, Codice\_ata, Nome)  
Aerei (**ID\_AEREO**, Modello, Capacità)  
Personale (**ID\_DIPENDENTE**, Nome, Cognome, Data\_nascita, Codice\_fiscale, Ruolo, Password)  
Prenotazione (**ID\_PRENOTAZIONE**, Passeggero, Volo, Prezzo\_biglietto)  
Servizio (ID\_dipendente, ID\_volo)

### 4. Vincoli di integrità relazionale

Voli.Aeroporto\_partenza -> Aeroporti.ID\_AEROPORTO  
Voli.Aeroporto\_arrivo -> Aeroporti.ID\_AEROPORTO  
Voli.Aereo -> Aerei.ID\_AEREO  
Prenotazione.Passeggero -> Passeggeri.ID\_PASSEGGERO  
Prenotazione.Volo -> Voli.ID\_VOLO  
Servizio.ID\_dipendente -> Personale.ID\_DIPENDENTE  
Servizio.ID\_volo -> Voli.ID\_VOLO

### 5. Query utilizzate per implementare le funzionalità

<pre>SELECT V.ID_VOLO, V.Data_partenza, V.Orario_partenza, V.Durata, A1.Nome AS Aeroporto_partenza, A2.Nome AS Aeroporto_arrivo, AER.Modello AS Aereo_Impiegato, P.Nome AS Nome_Pilota, P.Cognome AS Cognome_Pilota  FROM Voli V JOIN Aeroporti A1 ON V.Aeroporto_partenza = A1.ID_AEROPORTO JOIN Aeroporti A2 ON V.Aeroporto_arrivo = A2.ID_AEROPORTO JOIN Aerei AER ON V.Aereo = AER.ID_AEREO JOIN Servizio S ON V.ID_VOLO = S.ID_volo JOIN Personale P ON S.ID_dipendente = P.ID_DIPENDENTE  WHERE V.ID_VOLO = 'FL00000002' AND P.Ruolo = 'Pilota';</pre>	Restituisce informazioni dettagliate su un volo specifico, tra cui ID volo, data, orario, durata, aeroporti di partenza e arrivo, aereo impiegato e pilota.
--	---

<pre> SELECT   a.Modello,   COUNT(v.ID_VOLO) AS Numero_voli FROM   Aerei a JOIN   Voli v ON a.ID_AEREO = v.Aereo GROUP BY   a.ID_AEREO, a.Modello ORDER BY   Numero_voli DESC LIMIT 1; </pre>	<p>Trova l'aereo che è stato coinvolto nel maggior numero di voli, ordinando i risultati per numero di voli in ordine decrescente.</p>
<pre> SELECT   a.ID_AEROPORTO,   a.Nome,   a.Codice_iata,   (SELECT COUNT(v1.ID_VOLO)    FROM Voli v1    WHERE v1.Aeroporto_arrivo = a.ID_AEROPORTO) AS   Numero_arrivi FROM   Aeroporti a WHERE   (SELECT COUNT(v2.ID_VOLO)    FROM Voli v2    WHERE v2.Aeroporto_arrivo = a.ID_AEROPORTO) &gt;=   ALL   (SELECT COUNT(v3.ID_VOLO)    FROM Voli v3    GROUP BY v3.Aeroporto_arrivo); </pre>	<p>Trova l'aeroporto con il maggior numero di voli in arrivo, mostrando il nome, il codice IATA e il numero di arrivi.</p>
<pre> SELECT   a.ID_AEROPORTO,   a.Nome,   a.Codice_iata,   (SELECT COUNT(v1.ID_VOLO)    FROM Voli v1    WHERE v1.Aeroporto_partenza = a.ID_AEROPORTO)   AS Numero_partenze FROM   Aeroporti a WHERE   (SELECT COUNT(v2.ID_VOLO)    FROM Voli v2    WHERE v2.Aeroporto_partenza = a.ID_AEROPORTO)   &gt;= ALL   (SELECT COUNT(v3.ID_VOLO)    FROM Voli v3    GROUP BY v3.Aeroporto_partenza); </pre>	<p>Trova l'aeroporto con il maggior numero di voli in partenza, mostrando il nome, il codice IATA e il numero di partenze.</p>

<pre> SELECT   V.ID_VOLO,   AP.Nome AS Nome_Aeroporto_Partenza,   AP1.Nome AS Nome_Aeroporto_Arrivo,   V.Data_partenza,   V.Orario_partenza FROM   Servizio S JOIN   Voli V ON S.ID_volo = V.ID_VOLO JOIN   Aeroporti AP ON V.Aeroporto_partenza =   AP.ID_AEROPORTO JOIN   Aeroporti AP1 ON V.Aeroporto_arrivo =   AP1.ID_AEROPORTO WHERE   S.ID_dipendente = 'DIP0000006' AND   v.data_partenza &lt;= CURRENT_DATE; </pre>	<p>Mostra i voli operati da un dipendente (identificato dal suo ID) fino alla data odierna.</p>
<pre> SELECT SEC_TO_TIME(SUM(TIME_TO_SEC(V.Durata))) AS Totale_ore FROM Voli V JOIN Servizio S ON V.ID_VOLO = S.ID_volo JOIN Personale P ON S.ID_dipendente = P.ID_DIPENDENTE WHERE P.ID_DIPENDENTE = ? AND MONTH(V.Data_partenza) = MONTH(CURDATE()) AND YEAR(V.Data_partenza) = YEAR(CURDATE()) AND V.Data_partenza &lt;= CURRENT_DATE </pre>	<p>Calcola le ore totali lavorate da un dipendente nel mese corrente, sulla base della durata dei voli effettuati.</p>
<pre> SELECT   P.Nome AS Nome_Passeggero,   P.Cognome AS Cognome_Passeggero,   Pr.ID_PRENOTAZIONE AS Codice_prenotazione,   V.Data_partenza,   A1.Città AS Città_partenza,   A1.Nome AS Aeroporto_partenza,   A2.Città AS Città_arrivo,   A2.Nome AS Aeroporto_arrivo,   Pr.Prezzo_biglietto FROM   Prenotazioni Pr JOIN   Passeggeri P ON Pr.Passeggero = P.ID_PASSEGGERO JOIN   Voli V ON Pr.Volo = V.ID_VOLO JOIN   Aeroporti A1 ON V.Aeroporto_partenza =   A1.ID_AEROPORTO JOIN   Aeroporti A2 ON V.Aeroporto_arrivo =   A2.ID_AEROPORTO  WHERE P.ID_PASSEGGERO = '1122334455'; </pre>	<p>Restituisce tutti i voli prenotati da un passeggero specificato, inclusi dettagli come aeroporto di partenza e arrivo, data e orario del volo e prezzo del biglietto.</p>

<pre> SELECT   V.Data_partenza,   V.Orario_partenza,   A1.Città AS Città_partenza,   A1.Nome AS Aeroporto_partenza,   V.Data_arrivo,   V.Orario_arrivo,   A2.Città AS Città_arrivo,   A2.Nome AS Aeroporto_arrivo,   V.Durata FROM   Prenotazioni Pr JOIN   Voli V ON Pr.Volo = V.ID_VOLO JOIN   Aeroporti A1 ON V.Aeroporto_partenza = A1.ID_AEROPORTO JOIN   Aeroporti A2 ON V.Aeroporto_arrivo = A2.ID_AEROPORTO WHERE   Pr.ID_PRENOTAZIONE = 'BK00000003'; </pre>	<p>Restituisce i dettagli di un volo specifico dato il codice di prenotazione, inclusi orari di partenza e arrivo e aeroporti.</p>
<pre> SELECT   V.Data_partenza,   V.Orario_partenza,   A1.Città AS Città_partenza,   A1.Codice_iata AS Aeroporto_partenza,   V.Data_arrivo,   V.Orario_arrivo,   A2.Città AS Città_arrivo,   A2.Codice_iata AS Aeroporto_arrivo,   V.Durata FROM   Voli AS V JOIN   Aeroporti A1 ON V.Aeroporto_partenza = A1.ID_AEROPORTO JOIN   Aeroporti A2 ON V.Aeroporto_arrivo = A2.ID_AEROPORTO WHERE   A1.Codice_iata = "MAD" AND   A2.Codice_iata = "NRT" AND V.Data_partenza &gt; CURRENT_DATE(); </pre>	<p>Trova i voli alternativi per un dato itinerario, in base ai codici IATA di partenza e arrivo e alla data di partenza futura.</p>
<pre> SELECT   Volo AS ID_VOLO,   COUNT(*) AS Posti_occupati FROM   Prenotazioni WHERE   Volo = "FL00000001" GROUP BY   Volo; </pre>	<p>Conta il numero di prenotazioni per un volo specifico, indicando quanti posti sono stati occupati.</p>

<pre> SELECT   A.Capacità - (SELECT COUNT(*)                 FROM Prenotazioni P                 WHERE P.Volo = V.ID_VOLO) AS Posti_disponibili FROM   Aerei A JOIN   Voli V ON A.ID_AEREO = V.Aereo WHERE   V.ID_VOLO = "FL00000002"; </pre>	<p>Calcola il numero di posti disponibili per un volo specifico, sottraendo i posti occupati dalla capacità totale dell'aereo.</p>
<pre> SELECT Prezzo_biglietto FROM Prenotazioni WHERE ID_PRENOTAZIONE = "BK00000001"; </pre>	<p>Restituisce il prezzo di una prenotazione esistente, utilizzato per mantenere il prezzo di un nuovo biglietto.</p>
<pre> SELECT   CONCAT("BK",LPAD(CAST(MAX(CAST(SUBSTRING(ID_PRE NOTAZIONE, 3) AS UNSIGNED)) + 1 AS CHAR), 8, '0')) AS NuovoID_Prenotazione FROM Prenotazioni; </pre>	<p>Genera un nuovo ID di prenotazione univoco, incrementando l'ultimo ID esistente di una unità.</p>