



Progetto Aeroporto

Documento	Documentazione progetto aeroporto
Tipo	Database SQL
Versione	1.0
Data	29/09/2021



Autori	Cognome e Nome
	Capozza Emilio
	Denaro Giuseppe
	Giorgione Nicoletta
	Inglese Mario
	Luciano Salvatore
	Macrì Gabriele

Documenti di riferimento

Ref.	Titolo	Autore	Descrizione
1	Progetto Aeroporto	Sopra Steria	Database gestione voli aeroporto

Sommario

Introduzione	3
<i>Scopo e funzionalità</i>	3
<i>Acronimi e abbreviazioni</i>	3
Progettazione concettuale schema ER	4
<i>Diagramma ER</i>	4
Schema del DB	5
Interrogazioni ed analisi dei dati	6

Indice delle figure

Figura 1	4
Figura 2	5
Figura 3	6
Figura 4	6
Figura 5	6
Figura 6	7
Figura 7	7
Figura 8	7
Figura 9	7
Figura 10	8
Figura 11	8
Figura 12	8
Figura 13	9
Figura 14	9
Figura 15	9
Figura 16	9
Figura 17	9
Figura 18	10
Figura 19	10
Figura 20	10
Figura 21	10
Figura 22	11

Introduzione

Scopo e funzionalità

Nel seguente report presentiamo la documentazione riguardante la realizzazione del Database per la gestione degli aeroporti. Il Database deve rispondere ad una serie di requisiti che riguarda la gestione del traffico aeroportuale rispondendo ai seguenti requisiti di progetto assegnati:

- Le città con un aeroporto di cui non è noto il numero di piste;
- Le nazioni da cui parte e arriva il volo con codice 1;
- I tipi di aereo usati nei voli che partono da Roma;
- I tipi di aereo e il corrispondente numero di passeggeri per i voli che partono da Roma;
- Le città da cui partono voli diretti ad Amsterdam, ordinate alfabeticamente;
- Il numero di voli che partono il venerdì da Napoli;
- Le città italiane da cui partono almeno due voli alla settimana diretti in Olanda;
- Le città da cui parte l'aereo caratterizzato dal massimo numero di passeggeri;
- Le città su cui è diretto l'aereo caratterizzato dal massimo numero di passeggeri;
- Le città che sono servite dall'aereo caratterizzato dal massimo numero di passeggeri.

Acronimi e abbreviazioni

Termine	Descrizione
DB	Database
ER	Entità-Relazione

Progettazione concettuale schema ER

Al fine di soddisfare i requisiti del progetto sono state individuate tre entità distinte, *Volo*, *Aeroporto* ed *Aereo*, e i loro attributi, che permetteranno di descrivere il modello per la gestione aeroportuale.

Tali sono:

- *Volo*: i cui attributi rappresentano in maniera univoca un volo aereo: data_partenza, numero_passeggeri, aeroporto_partenza, aeroporto_arrivo, aereo, data_arrivo.
- *Aeroporto*: i cui attributi rappresentano la sua collocazione geografica e le sue proprietà strutturali: città, nazione, numero_piste.
- *Aereo*: i cui attributi rappresentano le caratteristiche costruttive dell'aereo e di proprietà: tipo_aereo, posti_disponibili, compagnia_appartenenza, nome_aereo.

Diagramma ER

Il modello ER è stato pensato focalizzandosi sull'entità volo e le sue relazioni con l'entità *Aeroporto* ed *Aereo*. L'entità *Volo* è legata dall'entità aeroporto attraverso le afferenze "parte da" e "arriva a", mentre si lega con l'aereo tramite l'id dell'aereo a cui fa riferimento il volo. Inoltre, dobbiamo controllare che il numero di posti disponibili nell'aereo sia sempre superiore al numero di passeggeri del volo. L'entità *Aereo* è legata a volo tramite l'afferenza "utilizza".

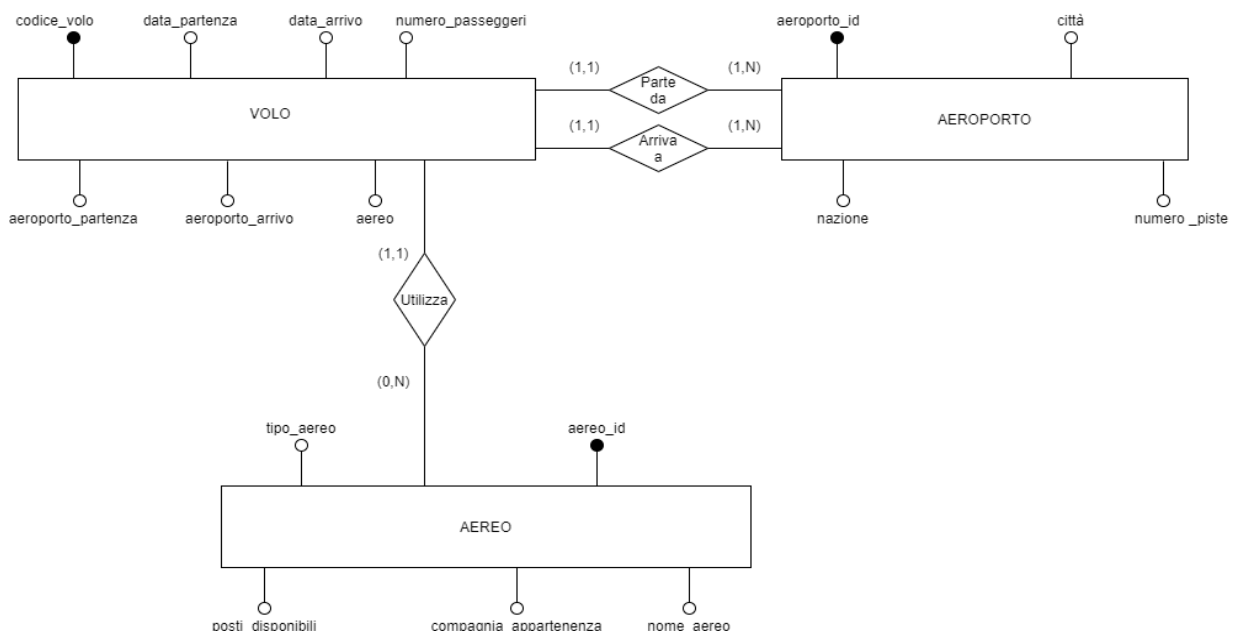


Figura 1

Schema del DB

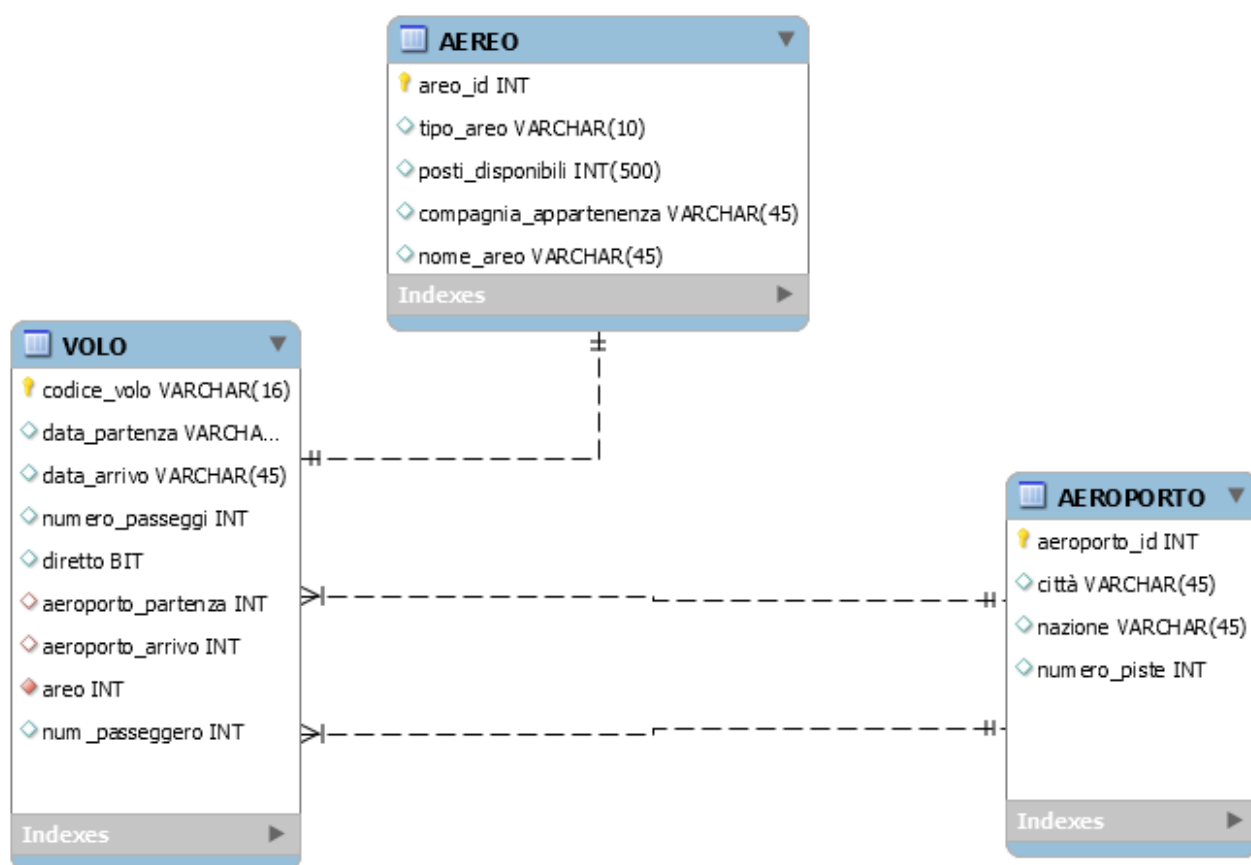


Figura 2

Le relazioni tra la tabella volo e la tabella aeroporto sono costituite dalle chiavi esterne che legano:

- l'attributo *aeroporto_partenza* della tabella *Volo* all'attributo *aeroporto_id* della tabella *Aeroporto*.
- l'attributo *aeroporto_arrivo* della tabella *Volo* all'attributo *aeroporto_id* della tabella *Aeroporto*.
- l'attributo *aereo* della tabella *Volo* all'attributo *aereo_id* della tabella *Aereo*.

Interrogazioni ed analisi dei dati

1. Le città con un aeroporto di cui non è noto il numero di piste.

L'obiettivo è interrogare la tabella *Aeroporto* in modo da ottenere i nomi delle città con un aeroporto di cui non si conosce il numero di piste.

Per farlo è possibile proiettare la colonna dei dati utili tramite il comando SELECT dalla tabella di appartenenza (*Aeroporto*), specificando tramite la clausola WHERE che il dato corrispondente al numero di piste sia nullo (NULL).

```
4 • SELECT *
5 FROM aeroporto
6 WHERE numero_piste IS NULL;
```

Figura 3

	aeroporto_id	città	nome	nazione	numero_piste
▶	7	San Pietroburgo	Pulkovo	Russia	NULL
	12	Istanbul	Atatürk	Turchia	NULL
	16	Oslo	Gardermoen	Norvegia	NULL
	18	Atene	Eleftherios Venizelos	Grecia	NULL

Figura 4

2. Le nazioni da cui parte e arriva il volo con codice 1.

L'obiettivo è ottenere la proiezione in cui siano presenti la nazione di partenza e la nazione di arrivo del volo con codice 1. Per come è stato strutturato il database, si ha all'interno della tabella volo sia il riferimento all'aeroporto di andata che a quello di ritorno, e, non potendo fare riferimento alla stessa tabella per entrambi, si utilizzano 2 alias per creare 2 tabelle temporanee. In una tabella si fa riferimento all'aeroporto di arrivo e nell'altra a quello di partenza.

In questo modo si realizza il prodotto cartesiano tra le tre tabelle *Volo* (V), *Aeroporto Arrivo* (AA) e *Aeroporto partenza* (AP) operando con una EQUI JOIN sulle righe corrispondenti al campo di interesse. In questo caso, l'EQUI JOIN viene fatta una volta tra l'attributo aeroporto_id della tabella *Aeroporto di partenza* e l'attributo aeroporto_partenza della tabella *Volo*, mentre l'altra tra l'attributo aeroporto_id con riferimento alla tabella *Aeroporto di arrivo* e l'attributo aeroporto_arrivo della tabella *Volo*.

Per concludere utilizziamo il comando WHERE per selezionare le righe con il codice di volo uguale a 1.

```
11 • SELECT AP.nazione AS nazione_partenza, AA.nazione AS nazione_arrivo
12 FROM volo AS V
13 JOIN aeroporto AP ON AP.aeroporto_id = V.aeroporto_partenza
14 JOIN aeroporto AA ON AA.aeroporto_id = V.aeroporto_arrivo
15 WHERE V.codice_volo = 1;
```

Figura 5

Come riportato nella tabella sottostante, il volo con codice 1 serve gli aeroporti in Italia e in Portogallo.

	nazione_partenza	nazione_arrivo
►	Italia	Portogallo

Figura 6

3. I tipi di aereo usati nei voli che partono da Roma.

Per ottenere la proiezione del tipo di aereo occorre interrogare la tabella *Aereo* (in questo caso con ALIAS AS) unita tramite una doppia JOIN alla tabella *Volo* e alla tabella *Aeroporto* rispettivamente con ALIAS V e AP.

La clausola WHERE filtra i tipi di aerei che partono dalla città di Roma.

```
20 • SELECT tipo_aereo, AP.città
21 FROM aereo AS A
22 JOIN volo AS V ON V.aereo = A.aereo_id
23 JOIN aeroporto as AP ON AP.aeroporto_id = V.aeroporto_partenza
24 WHERE città='ROMA';
```

Figura 7

Come risultato si hanno i seguenti tipi di aerei.

	tipo_aereo	città
►	Di stato	Roma
	Charter	Roma

Figura 8

4. I tipi di aereo e il corrispondente numero di passeggeri per i voli che partono da Roma.

Il tipo di aereo con il numero di passeggeri relativi ai voli che partono da Roma vengono prelevati interrogando le tabelle *Aereo*, *Volo* e *Aeroporto* messe in relazione tramite il comando WHERE in cui sono presenti le FOREIGN KEY della tabella *Volo* e della tabella *Aereo*. I risultati vengono poi filtrati tramite l'ulteriore clausola WHERE per la città di Roma.

```
29 • SELECT tipo_aereo, V.data_partenza, V.numero_passeggeri, V.aeroporto_partenza, V.aeroporto_arrivo
30 FROM aereo AS A, aeroporto AS Ar, volo AS V
31 WHERE V.aereo = A.aereo_id
32 AND Ar.aeroporto_id = V.aeroporto_partenza
33 AND città='ROMA';
```

Figura 9

Al risultato della precedente query viene aggiunto il rispettivo numero di passeggeri, come richiesto.

	tipo_aereo	data_partenza	numero_passeggeri	aeroporto_partenza	aeroporto_arrivo
►	Di stato	2021/09/21	212	14	18
	Charter	2021/07/28	322	14	11

Figura 10

5. *Le città da cui partono voli diretti ad Amsterdam, ordinate alfabeticamente.*

Tramite il comando SELECT si visualizza la proiezione con le città di partenza per i voli diretti ad Amsterdam (*AP.città*) della tabella ottenuta con una doppia JOIN tra la tabella *Volo* e le tabelle *Aeroporto AP* e *Aeroporto AA*.

I risultati sono stati filtrati tramite la clausola WHERE, in cui definiamo come città di arrivo Amsterdam, e ordinate alfabeticamente tramite il comando ORDER BY, riferito alle *AP.città*.

```

38 SELECT DISTINCT AP.città AS città_partenza, AA.città AS città_arrivo
39 FROM volo AS V
40 JOIN aeroporto AP ON AP.aeroporto_id = V.aeroporto_partenza
41 JOIN aeroporto AA ON AA.aeroporto_id = V.aeroporto_arrivo
42 WHERE AA.città = 'Amsterdam'
43 ORDER BY AP.città;
```

Figura 11

Le città da cui partono i voli diretti ad Amsterdam sono Francoforte, Milano, Napoli, Torino e Verona.

	città_partenza	città_arrivo
►	Francoforte	Amsterdam
	Milano	Amsterdam
	Napoli	Amsterdam
	Torino	Amsterdam
	Verona	Amsterdam

Figura 12

6. *Il numero di voli che partono il venerdì da Napoli.*

Il numero di voli è possibile ricavarlo tramite la funzione COUNT che conta le città di arrivo dalla tabella *Volo* in relazione con la tabella *Aeroporto* individuate dalla WHERE clause in cui la città di partenza è Napoli e il giorno della settimana è venerdì. Il COUNT è poi supportato dalla funzione GROUP BY.


```

48 • SELECT Ar.città, COUNT(Ar.città)
49 FROM aeroporto as Ar, volo as V
50 WHERE Ar.città = 'Napoli' AND V.aeroporto_partenza = Ar.aeroporto_id
51 AND DATE_FORMAT(V.data_partenza, ' %W %M %D %Y') LIKE ' Friday %';

```

Figura 13

	città	COUNT(Ar.città)
►	Napoli	2

Figura 14

7. Le città italiane da cui partono almeno 2 voli alla settimana diretti in Olanda.

L'obiettivo è visualizzare la lista delle città italiane ottenuta con una doppia JOIN legata all'aeroporto di partenza e di arrivo.

I risultati vengono filtrati con il comando WHERE, in cui definiamo le righe con aeroporto di partenza in Olanda e con almeno 2 voli a settimana.

Queste vengono raggruppate per data e città di partenza e vengono filtrate scegliendo quelle la cui ricorrenza è maggiore di uno.

```

55 • SELECT WEEK(V.data_partenza) AS partenza, COUNT(V.data_partenza),
56 AP.città AS città_partenza, AA.città AS città_arrivo, AA.nazione, V.codice_volo
57 FROM volo AS V
58 JOIN aeroporto AP ON AP.aeroporto_id = V.aeroporto_partenza
59 JOIN aeroporto AA ON AA.aeroporto_id = V.aeroporto_arrivo
60 WHERE AA.nazione = 'Paesi Bassi' AND AP.nazione = 'Italia'
61 GROUP BY partenza, città_partenza
62 HAVING count(V.data_partenza) > 1
63 ORDER BY città_partenza;

```

Figura 15

	partenza	count(V.data_partenza)	città_partenza	città_arrivo	nazione	codice_volo
►	44	2	Napoli	Amsterdam	Paesi Bassi	40
	29	2	Verona	LAia	Paesi Bassi	35

Figura 16

8. Le città da cui parte l'aereo caratterizzato dal massimo numero di passeggeri.

Selezioniamo tramite la funzione SELECT le città da cui parte l'aereo (AP.città). Da questa colonna è stato selezionato il campo cui corrispondeva l'aeroporto_id del volo che fa riferimento all'aereo col numero massimo di passeggeri.

```

70 • SELECT AP.città
71 FROM Volo AS V, Aereo AS A, Aeroporto AS AP
72 WHERE AP.aeroporto_id=V.aeroporto_partenza AND V.aereo=A.aereo_id
73 AND V.numero_passeggeri= (select max(numero_passeggeri) from Volo);

```

Figura 17

La città da cui parte l'aereo con il numero massimo di passeggeri è unica ed è Torino.

	città
►	Torino

Figura 18

9. Le città su cui è diretto l'aereo caratterizzato dal massimo numero di passeggeri.

Si procede come nel precedente caso, cambiando nella SELECT il campo relativo alla città di arrivo.

```

77 • SELECT AA.città
78 FROM Volo AS V, Aereo AS A, Aeroporto AS AA
79 WHERE AA.aeroporto_id=V.aeroporto_arrivo AND V.aereo=A.aereo_id
80 AND V.numero_passeggeri= (select max(numero_passeggeri) from Volo);

```

Figura 19

Come risultato otteniamo la città di Amsterdam.

	città
►	Amsterdam

Figura 20

10. Le città che sono servite dall'aereo caratterizzato dal massimo numero di passeggeri.

Per ottenere le città servite dall'aereo con il numero massimo di passeggeri si procede come nei casi precedenti a differenza di una doppia JOIN legata all'aeroporto di partenza e di arrivo, e di una SUB QUERY per individuare l'aereo con il numero massimo di posti disponibili.

```

82 • SELECT AP.città, AA.città, A.aereo_id
83 FROM aereo as A, volo AS V
84 JOIN aeroporto AP ON AP.aeroporto_id = V.aeroporto_partenza
85 JOIN aeroporto AA ON AA.aeroporto_id = V.aeroporto_arrivo
86 WHERE V.aereo = A.aereo_id
87 AND V.aereo = (
88     SELECT A.aereo_id
89     FROM aereo AS A
90     WHERE A.posti_disponibili = (
91         SELECT MAX(posti_disponibili)
92         FROM aereo
93     )
94 );

```

Figura 21

Il risultato della Query sono le città di Torino e Amsterdam.

	città	città	aereo_id
►	Budapest	Torino	171
	Francoforte	Amsterdam	171

Figura 22

sopra  steria