**Relazione Laboratorio Basi di dati**

Si vuole progettare una base di dati di supporto all’organizzazione delle attività di una compagnia aerea italiana.

La compagnia aerea offre un certo numero di voli caratterizzati da: un codice numerico che identifica univocamente il volo, il giorno in cui il volo viene effettuato (ad esempio, 24 giugno 2010; per semplicità, assumiamo che ogni volo inizi e termini lo stesso giorno), il tipo di aereo utilizzato, l’orario di partenza (ad esempio, ore 16:00) e l’orario di arrivo (ad esempio, ore 17:00), l’aeroporto di partenza (ad esempio, l’aeroporto Marco Polo di Venezia) e l’aeroporto di destinazione (ad esempio, Roma Fiumicino).

Ogni aeroporto sia identificato univocamente dal proprio nome.

Di ogni aeroporto si memorizzano la città e la nazione in cui si trova e il numero di piste disponibili.

Ogni pista sia identificata da un numero all’interno dell’aeroporto di appartenenza (ad esempio, la pista numero 1 di Roma Fiumicino).

Di ogni tipo di aereo vengono registrati la società costruttrice, il numero massimo di passeggeri e la quantità massima di merci che possono essere trasportati.

I voli si suddividono in voli nazionali ed internazionali.

Dei voli passati interessa mantenere traccia dell’orario effettivo di partenza e di arrivo (ad esempio, 16:45 e 17:50);

Dei voli futuri interessa sapere il numero di posti prenotati e il numero di posti ancora disponibili.

**Fasi della creazione della base di dati:**

1. Raccolta e analisi dei requisiti
2. Progettazione concettuale
3. Progettazione logica
4. Progettazione fisica
5. Implementazione
6. Analisi di dati in R

**1. Raccolta e analisi dei requisiti**

**1.1 Glossario**

|  |  |  |
| --- | --- | --- |
| **Termine** | **Descrizione** | **Collegamenti** |
| Volo | Viaggio di un aereo da un punto di partenza a un punto di arrivo. | Aeroporto, Tipo di aereo. |
| Tipo di aereo | Modello dell’aereo che compie un certo volo. | Casa costruttrice, Volo. |
| Aeroporto | Struttura composta da piste da cui parte o arriva un aereo. | Volo, Pista. |
| Pista | Struttura di un aeroporto da cui partono gli aerei. | Aeroporto. |
| Società Costruttrice | Entità che costruisce gli aerei, ogni casa può costruire più tipi di aereo. | Tipo di aereo. |

**1.2 Riscrittura e struttura dei requisiti**

|  |
| --- |
| **Frasi relative al volo** |
| La compagnia aerea offre un certo numero di voli caratterizzati da: un codice numerico che identifica univocamente il volo, il giorno in cui il volo viene e effettuato (ad esempio, 24 giugno 2010; per semplicità, assumiamo che ogni volo inizi e termini lo stesso giorno), il tipo di aereo utilizzato, l'orario di partenza (ad esempio, ore 16:00) e l'orario di arrivo (ad esempio, ore 17:00), l'aeroporto di partenza (ad esempio, l’aeroporto Marco Polo di Venezia) e l'aeroporto di destinazione (ad esempio, Roma Fiumicino). |
| **Frasi relative al tipo di volo** |
| I voli si suddividono in voli nazionali ed internazionali. Dei voli passati interessa mantenere traccia  dell'orario effettivo di partenza e di arrivo (ad esempio, 16:45 e 17:50); dei voli futuri interessa sapere il numero di posti prenotati e il numero di posti ancora disponibili. |
| **Frasi relative all’aeroporto** |
| Ogni aeroporto sia identificato univocamente dal proprio nome. Di ogni aeroporto si memorizzano la  città e la nazione in cui si trova e il numero di piste disponibili. |
| **Frasi relative a pista** |
| Ogni pista sia identificata da un numero  all'interno dell'aeroporto di appartenenza (ad esempio, la pista numero 1 di Roma Fiumicino). |
| **Frasi relative al tipo di aereo** |
| Di ogni tipo di aereo vengono registrati la società costruttrice, il numero massimo di passeggeri e la  quantità massima di merci che possono essere trasportati. |

**1.3 Esempio di operazioni**

* **Operazione 1:** Inserire un volo, includendo tutti i suoi parametri.
* **Operazione 2:** inserire un nuovo aeroporto per una città.
* **Operazione 3:** aggiungere una pista in un aeroporto.
* **Operazione 4:** Aggiungere un tipo di aereo.
* **Operazione 5:** Visualizzare tutti i voli passati con le relative informazioni.
* **Operazione 6:** Visualizzare il tipo di aereo per un determinato volo.
* **Operazione 7:** Visualizzare per ogni volo i posti prenotati e quelli disponibili.
* **Operazione 8:** Visualizzare tutti i voli che sono diretti in una città/aeroporto.
* **Operazione 9:** Visualizzare gli orari di tutti i voli in partenza da un aeroporto.
* **Operazione10:** Visualizzare il numero di piste per un aeroporto.

**2. Progettazione concettuale**

**2.1 Schema Entità/Relazione**

Abbiamo utilizzato un approccio di tipo bottom-up:

1. Definizione delle entità principali:

Immagine che contiene poligono

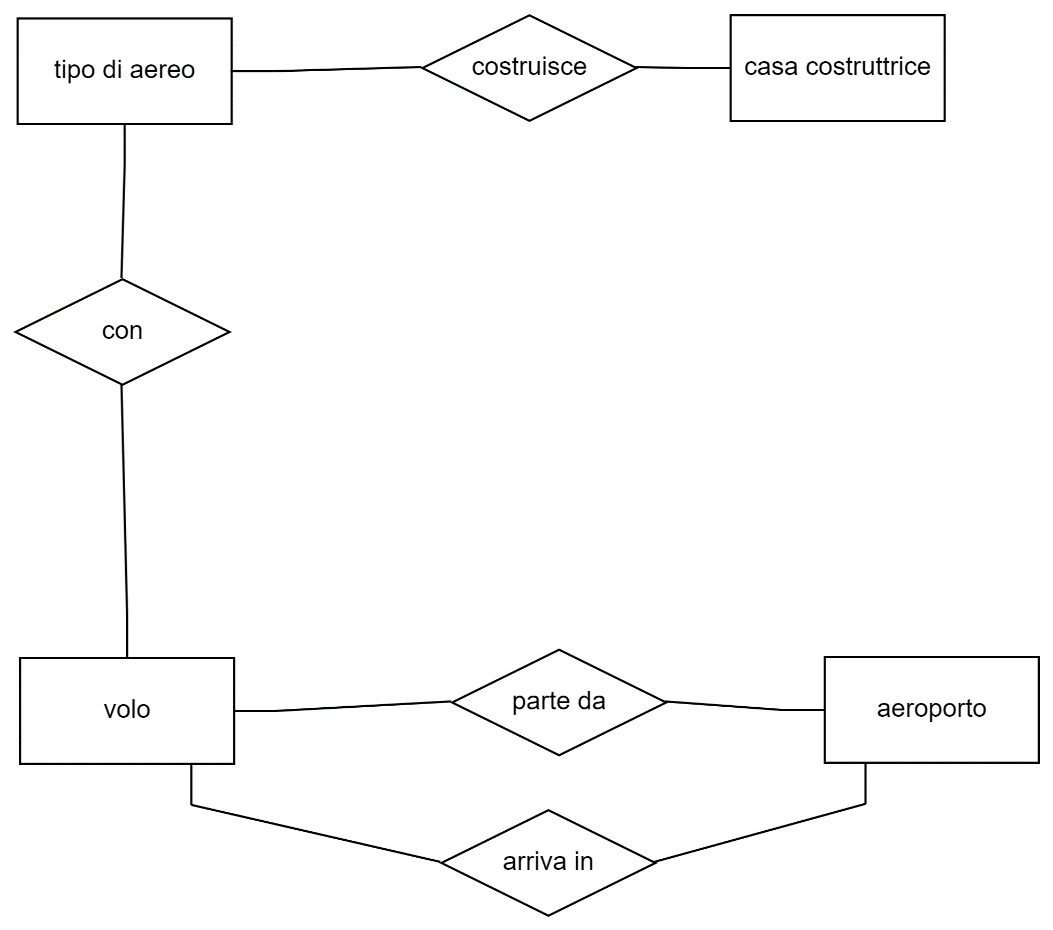
Descrizione generata automaticamente

1. Studio della relazione tra pista e aeroporto:

Immagine che contiene diagramma

Descrizione generata automaticamente

1. Aggiunta dell’entità “tipo di aereo” allo schema principale:



1. Studio della specializzazione dell’entità “volo”:

Immagine che contiene diagramma

Descrizione generata automaticamente

1. Unione di tutti i componenti e aggiunta dei relativi attributi.

**Immagine che contiene diagramma, schizzo, disegno, testo

Descrizione generata automaticamente**

**3. Progettazione Logica**

**3.1 Analisi delle ridondanze**

L’entità *aeroporto*ha un attributo derivato che è *#piste* (numero piste), che è il numero delle istanze della relazione “si trova in” tra entità *aeroporto*e entità *pista***.**

Prendiamo in considerazione le due operazioni disponibili su *pista* e *aeroporto*:

- **Operazione 1:** aggiungere una pista in un aeroporto.

- **Operazione 2:** visualizzare il numero di piste per un aeroporto.

Assumiamo che i volumi di dati per ogni entità/relazione siano rappresentati dalla seguente tabella:

|  |  |  |
| --- | --- | --- |
| **Entità/Relazione** | **Tipo** | **Volume** |
| *aeroporto* | E | 500 |
| *pista* | E | 1000 |
| *si trova in* | R | 1000 |

Assumiamo anche che ogni aeroporto in media ha due piste.

Assumiamo che la frequenza di queste operazioni sia rappresentata dalla seguente tabella:

|  |  |  |
| --- | --- | --- |
| **Operazione** | **Tipo** | **Frequenza(volte/anno)** |
| Operazione1 | W | 5 |
| Operazione2 | R | 10.000 |

- Analisi in due casi per l’operazione1:

* **Caso 1**: manteniamo l’attributo derivato *numero piste*, quindi si hanno 1 lettura in *pista*, 1 scrittura in *pista*, 1 scrittura in *si trova* in, 1 lettura in *aeroporto*, 1 scrittura in *aeroporto*. Considerando che per convenzione una scrittura equivale a due letture, otteniamo 8 letture, per un totale di 40 accessi all’anno.
* **Caso 2**: eliminiamo l’attributo derivato *numero piste*, quindi si hanno 1 lettura in *pista*, 1 scrittura in *pista*, 1 scrittura in *si trova in*. Otteniamo così 5 letture, per un totale di 25 accessi all’anno.

- Analisi in due casi per l’operazione2:

* **Caso 1**: manteniamo l’attributo derivato *numero piste*, quindi si ha 1lettura sull’entità *aeroporto*, e quindi nel nostro caso avremmo 10.000 letture all’anno.
* **Caso 2**: eliminiamo l’attributo derivato *numero piste*, si hanno 1 lettura su *aeroporto* e 2 letture su *si trova in*, per un totale di 30.000 letture all’anno.

- Conclusione: risulta conveniente **mantenere l’attributo ridondante** *numero piste* poiché il numero di accessi sarebbe di 10.000 + 40 = 10.040, mentre se venisse eliminato gli accessi sarebbero 30.000 + 25 = 30.025. Inoltre, l’operazione di scrittura risulta molto meno frequente rispetto quella di lettura.

**3.2 Eliminazione delle generalizzazioni**

Nello schema entità-relazione abbiamo una generalizzazione sull’entità *volo*.

Elimineremo tale generalizzazione nel seguente modo:

Immagine che contiene diagramma, schizzo, disegno, testo

Descrizione generata automaticamente

Gli attributi delle entità specializzanti sono stati trasferiti all’entità genitore ed è stato aggiunto un attributo che identifica il *tipo di volo*, che potrà essere *futuro* oppure *passato*.

**3.3 Individuazione degli identificatori**

|  |  |
| --- | --- |
| **Entità** | **Identificatore** |
| *Aeroporto* | *Nome* |
| *Volo* | *#volo* |
| *Pista* | *Aeroporto, #pista (entità debole)* |
| *Tipo di Aereo* | *Nome* |
| *Casa Costruttrice* | *Nome* |

**3.4 Creazione del modello relazionale**

*Volo* (*#volo*, *orarioPartenza*, *orarioArrivo*, *Nazionale*, *OraPartenzaEffettiva*, *OraArrivoEffettiva*, *postiDisponbili*, *postiPrenotati*)

*Aeroporto* (*Nome*, *Nazione*, *Città*, *#piste*)

*TipoDiAereo* (*Nome*, *NumeroMassimoMerci*, *NumeroMassimoPasseggeri*)

*CasaCostruttrice* (*nome*)

*Partenza* (*#volo*, *nomeAeroporto*)

*Arrivo* (*#volo*, *NomeAeroporto*)

*Utilizza* (*#volo*, *Nome*)

*Costruisce* (*Nome*, *tipoDiAereo*)

*Pista* (*#pista*, *NomeAeroporto*)

**4. Progettazione fisica**

**4.1 Modalità di memorizzazione dei dati**

*Volo* (

*#volo*  Integer

*orarioPartenza*  TimeStamp

*orarioArrivo*  TimeStamp

*Nazionale*  Boolean

*OraPartenzaEffettiva*  TimeStamp

*OraArrivoEffettiva*  TimeStamp

*postiDisponibili*  Integer

*postiPrenotati*  Integer

)

*Aeroporto* (

*Nome* String

*Nazione* String

*Città* String

*#piste*  Integer

)

*TipoDiAereo* (

*Nome* String

*NumeroMassimoMerci* Integer

*NumeroMassimoPasseggeri*  Integer

)

*CasaCostruttrice* (

*Nome*  String

)

*Partenza* (

*#volo*  Integer

*nomeAeroporto* String

)

*Arrivo* (

*#volo*  Integer

*NomeAeroporto* String

)

*Utilizza* (

*#volo* Integer

*Nome* String

)

*Costruisce* (

*Nome* String

*tipoDiAereo*  String

)

*Pista* (

*#pista*  Integer

*NomeAeroporto* String

)

**4.2 Definizione delle relazioni in SQL**

create table partenza(

    #volo integer primary key,

    nomeAeroporto varchar(10),

    foreign key (nomeAeroporto) references aeroporto(nome),

    foreign key (#volo) references volo(#volo)

);

create table arrivo(

    #volo integer primary key,

    nomeAeroporto varchar(10),

    foreign key (nomeAeroporto) references aeroporto(nome),

    foreign key (#volo) references volo(#volo)

);

create table utilizza(

    #volo integer primary key,

    nome varchar(10),

    foreign key (#volo) references volo(#volo),

    foreign key (nome) references tipoDiAereo(nome)

);

create table costruisce(

    nome varchar(10),

    tipoDiAereo varchar(10),

    primary key(nome,tipoDiAereo),

    foreign key (nome) references casaCostruttrice(nome),

    foreign key (tipoDiAereo) references tipoDiAereo(nome)

);

create table pista(

    #pista integer,

    nomeAeroporto varchar(10),

    primary key(#pista,nomeAeroporto)

    foreign key (nomeAeroporto) references aeroporto(nome)

);

**5. Implementazione**

**5.1 Trigger**

1. Un volo non puo partire e arrivare nello stesso posto:

create trigger controlla\_AerArrivo  
before insert or update on arrivo  
for each row  
execute procedure controlla\_Ar\_arrivo();

create or replace function controlla\_Ar\_arrivo()  
returns trigger language plpgsql as   
$$  
    declare   
        aeroporto\_partenza varchar(10);  
    begin   
  
    select nomeAeroporto into aeroporto\_partenza from partenza where new.nvolo = nvolo;  
  
    if aeroporto\_partenza = new.nomeAeroporto then  
       return null;  
    end if;  
  
    return new;  
  
    end;  
$$;

1. Per modificare l'orario effettivo di arrivo di un volo l'orario effettivo di partenza non può essere null:

create trigger controlla\_data  
before update on volo  
for each row  
execute procedure controlla\_arrivo();

create or replace function controlla\_arrivo()  
returns trigger language plpgsql as   
$$  
    begin   
  
    if new.oraPartenzaEffettiva IS NULL then  
       return null;  
    end if;  
  
    return new;  
  
    end;  
$$;

**5.2 Query**

1. Visualizzare il numero di piste per ogni aeroporto:

select NomeAeroporto, count(\*) from Pista group by NomeAeroporto

1. Visualizzare i voli futuri che arrivano in un aeroporto e da dove partono:

select  #volo as codice\_volo, P.nomeAeroporto as parte\_da

from Arrivo as A, Partenza as P, Volo as V

where A.nomeAeroporto = 'nom\_aeroporto' AND A.#volo = P.#volo AND A.#volo = V.#volo AND v.OraArrivoEffettiva IS NOT NULL

1. Visualizzare le case costruttrici che producono un solo tipo di aereo:

select Nome from Costruisce as c1,c2

where c1.Nome = c2.Nome AND c1.tipoDiAereo != c2.tipoDiAereo

**6. Analisi dei dati in R**

**6.1 Query e visualizzazione del risultato tramite grafici**

1. Numero di voli in arrivo per ciascun aeroporto:

select nomeAeroporto, count(\*) from arrivo group by nomeAeroporto

Immagine che contiene testo, schermata, numero, Carattere

Descrizione generata automaticamente

1. Tipi di aereo più usati:

select nome, count(\*) from utilizza group by nome

Immagine che contiene testo, Policromia, diagramma, Elementi grafici

Descrizione generata automaticamente