

DB_AIRLINES



ANNO ACCADEMICO
2023-2024

PROPONENTI

- FRANCESCO CIANNIELLO 0124002804
- FRANCESCO FERRILLO 0124002921
- MAURIZIO DI NUZZO 0124002831

DATA DI CONSEGNA
15/07/2024



INDICE



PROGETTAZIONE
SINTESI DEI REQUISITI
GLOSSARIO
DIAGRAMMA EE/R
DIAGRAMMA RELAZIONALE
UTENTI E LORO CATEGORIE
OPERAZIONI DEGLI UTENTI
VOLUMI
VINCOLI DI INTEGRITA'
VERIFICA DI NORMALITA'

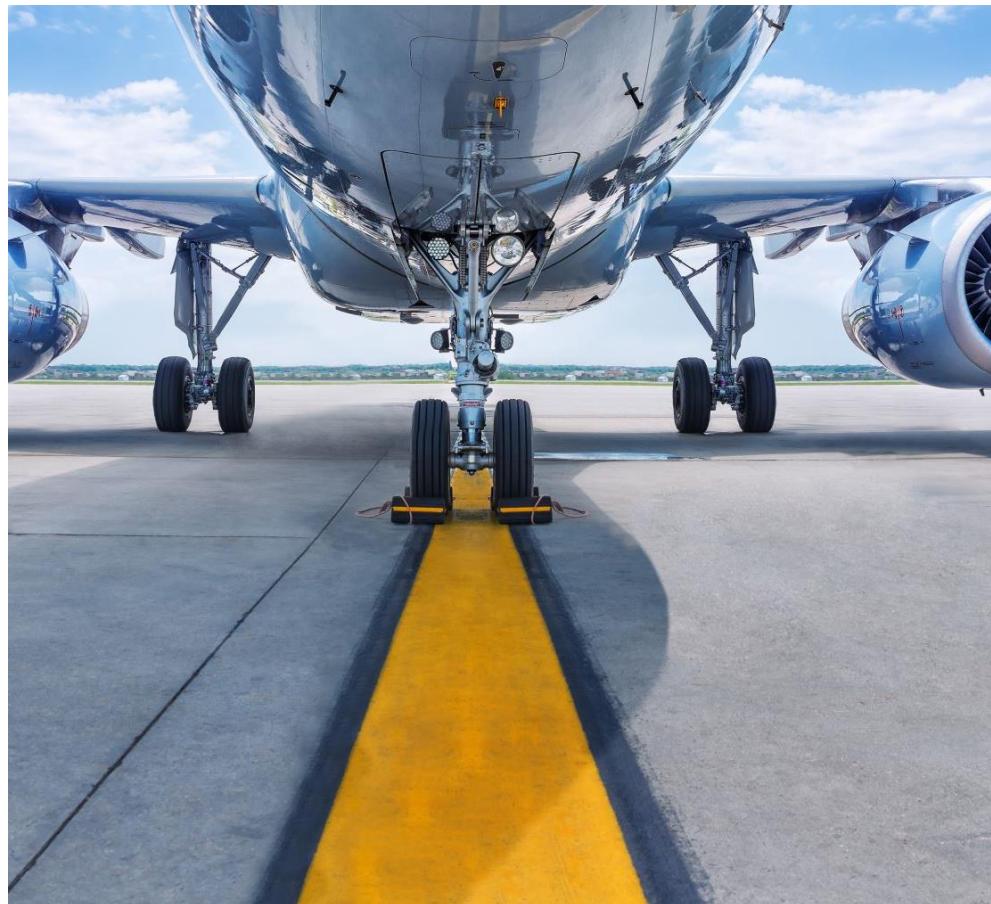


IMPLEMENTAZIONE
CREAZIONE UTENTI
DATA DEFINITION LANGUAGE
DATA MANIPULATION LANGUAGE
QUERY
TRIGGER
PROCEDURE E FUNZIONI
VISTE
DATA CONTROL LANGUAGE

PROGETTAZIONE



SINTESI DEI REQUISITI



Si vuole realizzare un database che rappresenti le informazioni più importanti degli aeroporti nazionali ed internazionali, quali organizzazione dei voli, gestione del personale e identificazione dei passeggeri.

Gli **aeroporti** sono identificati da un codice univoco denominato Icao. Di ciascun aeroporto interessa il numero di piste, la città di appartenenza e i numeri di telefono aziendali. Tra gli altri **contatti** posseduti dall'**aeroporto** vi sono l'email, il fax e la PEC univoca.

Negli aeroporti vengono gestiti arrivi e partenze, organizzando in maniera opportuna i voli.

Di ciascun **volo**, identificato da un codice, si vuole conoscerne: la durata, gli orari di arrivi e partenze previsti ma anche effettivi e il tipo di volo (internazionale o nazionale). Ogni volo è organizzato da una compagnia.

Ciascuna **compagnia**, identificata da un codice univoco denominato Iata, è caratterizzata da un nome e da uno o più numeri di telefono aziendali. Tra gli altri **contatti** posseduti dalla **compagnia** vi sono l'email, il fax e la PEC univoca.

SINTESI DEI REQUISITI



Le tratte percorse variano da volo a volo.

I voli sono eseguiti dagli aerei. Di ogni **aereo**, identificato da una marca di immatricolazione, si vuole conoscere il peso, la potenza e il numero di posti.

Ogni volo è composto da viaggiatori, i quali possono essere passeggeri o parte del personale di bordo. Di ciascun **viaggiatore** interessano il nome, il cognome, la data di nascita, l'email e il numero di telefono. Dei **passeggeri**, identificati dal codice fiscale, si vuole conoscere il numero di passaporto.

Del **personale di bordo**, identificato da una matricola, interessa il ruolo (hostess, pilota, co-pilota...). Ogni componente del personale ha una licenza.

Di ciascuna **licenza**, caratterizzata da un codice, interessano il tipo e la data di conseguimento.

Ciascun passeggero compra dei biglietti per poter viaggiare. Su ciascun **biglietto**, identificato da un codice, è presente la classe del volo. Ogni biglietto è riferito ad uno specifico volo.

SINTESI DEI REQUISITI

I biglietti sono comprati esclusivamente online. Di ciascun **pagamento** interessa il numero della carta, l'importo, la data del pagamento e la data di scadenza della carta.

Ogni passeggero possiede un documento d'identità necessario ai fini dell'imbarco. Di ciascun **documento d'identità** interessano il numero di documento, il quale lo identifica, l'ente che lo ha rilasciato e la data di scadenza.

I passeggeri portano con sé almeno un bagaglio. Di ogni **bagaglio**, invece, sono importanti il tag che lo identifica, il tipo (a mano, da stiva...) e il peso.

Se l'imbarco è consentito, al passeggero viene confermata la **carta di imbarco**, ottenuta nel momento della prenotazione del volo, di cui interessa il numero, la data del volo e il numero del posto assegnato.

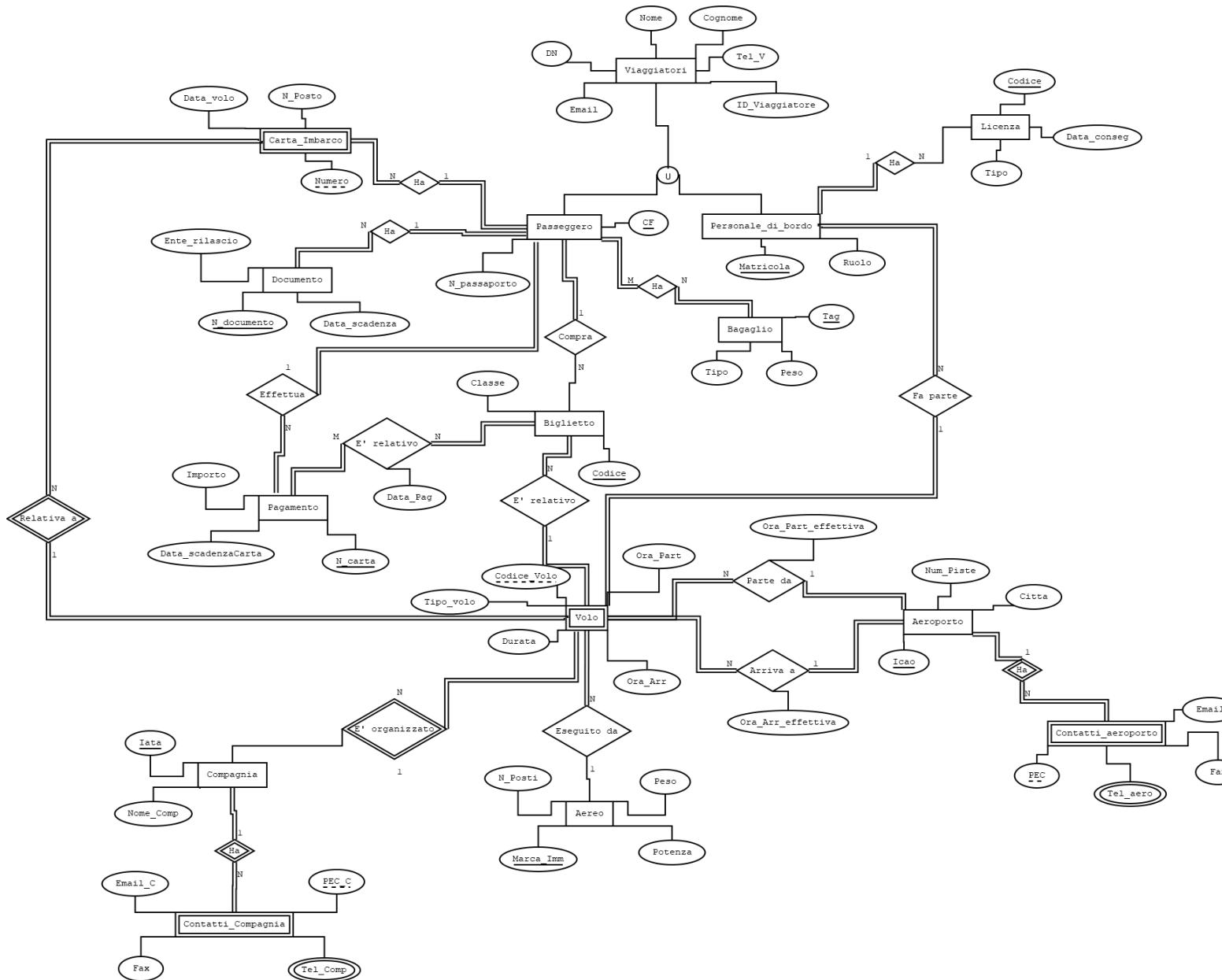


GLOSSARIO

IL GLOSSARIO EVIDENZIA I TERMINI TECNICI UTILIZZATI NEL DATABASE

<u>TERMINI</u>	<u>DEFINIZIONE</u>	<u>SINONIMI</u>	<u>OMONIMI</u>
Icao	Organizzazione internazionale dell'aviazione civile che assegna un codice ad ogni aeroporto	-	-
Codice volo	Codice numerico che identifica un particolare volo	-	-
Iata	Associazione internazionale trasporti aerei, la quale assegna codici per aerei gestiti da compagnie aeree	-	-
Marca d'Immatricolazione	Stringa alfanumerica che identifica un aeromobile (targa)	Distintivo	-
Matricola	Codice che caratterizza il personale di bordo	Albo, ruolo	Matricola ("Principiante")
Licenza	Documento che attesta la capacità, a chi la possiede, di esercitare l'attività di volo	Certificazione, patente	Licenza ("Permesso")
Classe di volo	Lettera che identifica i servizi offerti con quel tipo di biglietto	Categoria di volo	-
Data	Data di convalida del biglietto relativo ad un determinato volo	-	-
Numero di documento	Identificatore univoco che distingue una persona dall'altra	-	-
Tag	Etichetta attaccata al bagaglio con i dati del passeggero	-	-
Carta d'imbarco	Documento che permette di salire sull'aereo	Documento/certificato d'imbarco	-

DIAGRAMMA EE/R



ENTITA'

LE ENTITA' UTILIZZATE
NEL DIAGRAMMA EE/R
SONO QUI ELENcate

1. Aeroporto	2. Contatti Aeroporto	3. Volo	4. Compagnia
5. Contatti Compagnia	6. Aereo	7. Viaggiatori	8. Carta D'Imbarco
9. Biglietto	10. Bagaglio	11. Documento	12. Pagamento
13. Licenza	14. Unione:	a) Personale di bordo	b) Passeggeri

ATTRIBUTI

SONO QUI RIPORTATI GLI ATTRIBUTI RELATIVI ALLE SINGOLE ENTITA'



- 1) Aeroporto= Icao, Città, Numero Piste.
- 2) Contatti Aeroporto= Telefono Aeroporto, PEC, Email, Fax.
- 3) Volo= Durata, Tipo Volo, Ora Arrivo, Ora Partenza, Codice Volo.
- 4) Compagnia= Nome, Iata.
- 5) Contatti Compagnia= Telefono Compagnia, PEC, Email, Fax.
- 6) Aereo= Marca Immatricolazione, Peso, Potenza, Numero Posti.
- 7) Licenza= Tipo, Codice Licenza, Data Conseguimento.
- 8) Viaggiatori= Nome, Cognome, Telefono Viaggiatori, Data Di Nascita, Id Viaggiatore, Email.
- 9) Unione:
- c) Personale Di Bordo= Matricola, Ruolo.
- d) Passeggeri= Codice Fiscale, Numero Passaporto.
- 10) Biglietto= Codice, Classe.
- 11) Bagaglio = Peso, Tag, Tipo.
- 12) Carta D'imbarco= Numero, Data Volo, Numero Posto.
- 13) Documento= Numero Documento, Data Scadenza, Comune Di Rilascio.
- 14) Pagamento= Numero Carta, Importo, Data Pagamento, Data Scadenza Carta.

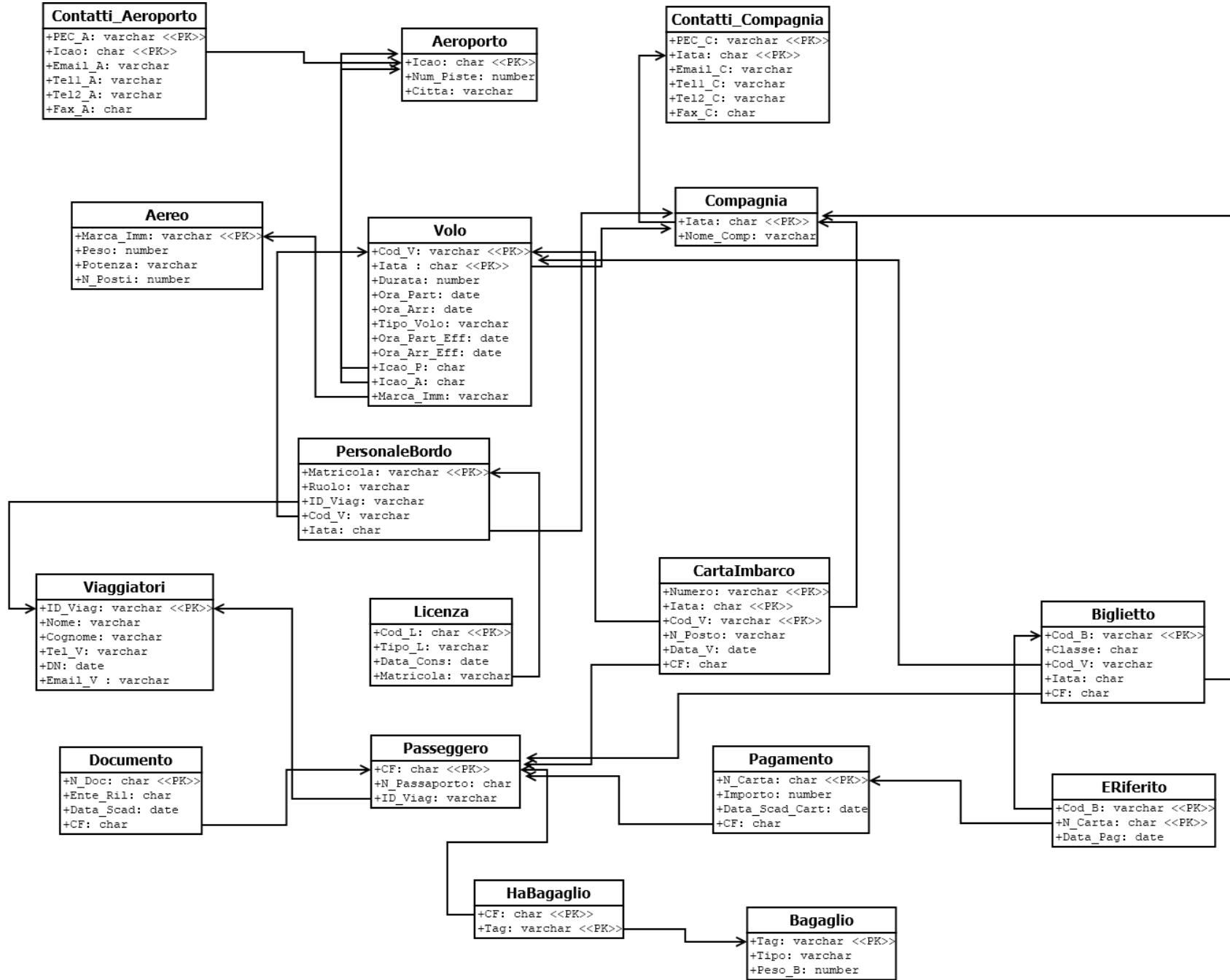


CARDINALITA'

L'ELENCO RIPORTA LE
CARDINALITA' DELLE
ASSOCIAZIONI

- Volo parte da Aeroporto (N, 1).
- Ora partenza effettiva.
- Volo arriva ad Aeroporto (N, 1).
- Ora arrivo effettiva.
- Aeroporto ha Contatti Aeroporto (1, N).
- Volo è organizzato da Compagnia (N, 1).
- Compagnia ha Contatti Compagnia (1, N).
- Volo è eseguito da Aereo (N, 1).
- Viaggiatori è un'unione che si divide in Passeggero e Personale di Bordo.
 - Passeggero compra Biglietto (1, N).
 - Biglietto è relativo a Volo (N, 1).
 - Passeggero effettua Pagamento (1, N).
 - Passeggero ha Documento (1, N).
 - Passeggero ha Bagaglio (M, N).
 - Pagamento è relativo a Biglietto (M, N).
 - Personale di Bordo ha Licenza (1, N).
 - Personale di Bordo fa parte di Volo (N, 1).
 - Passeggero ha Carta D'Imbarco (1, N).
 - Carta D'Imbarco è relativa a Volo (N, 1).

DIAGRAMMA RELAZIONALE

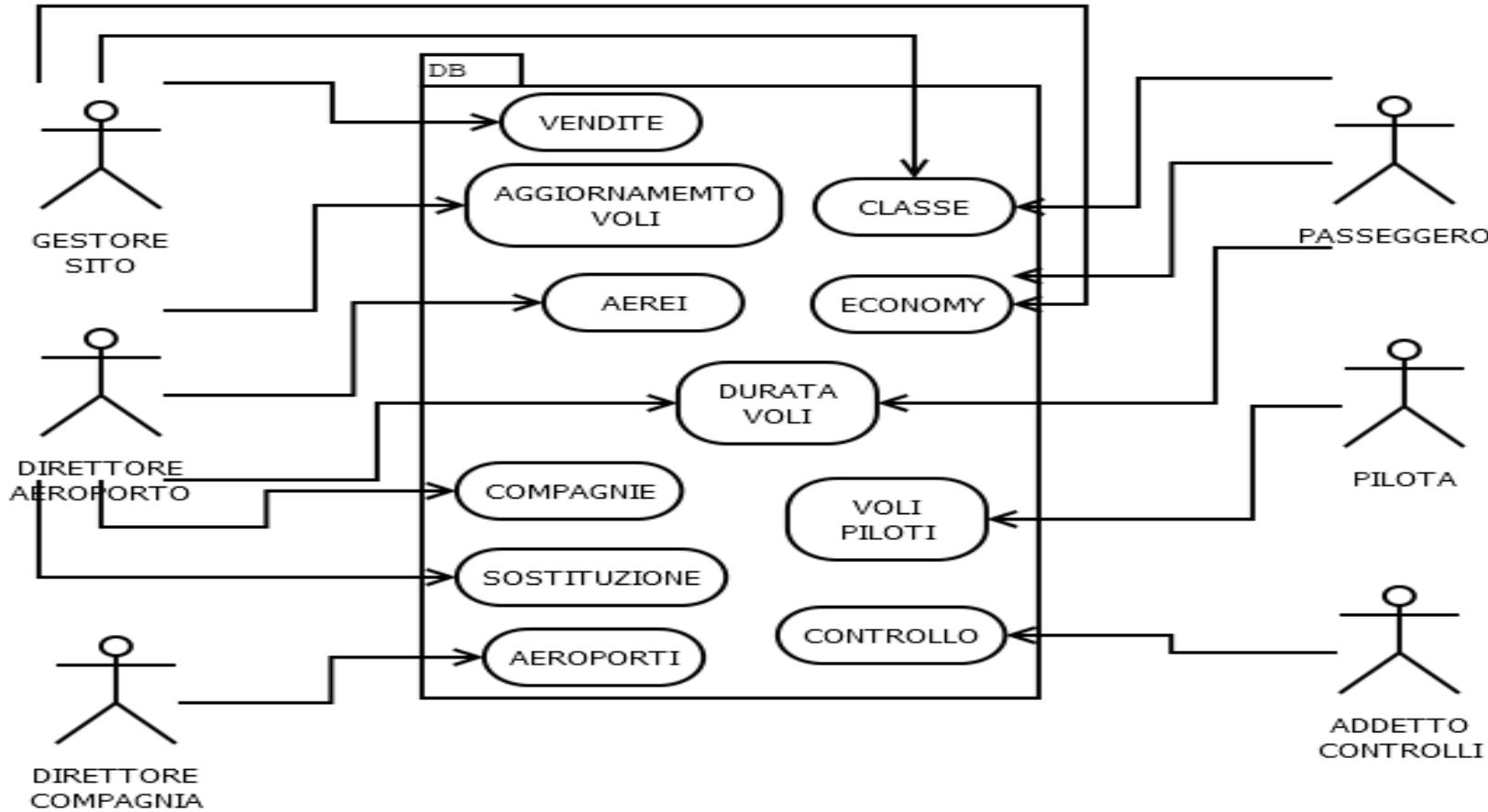


UTENTI E LE LORO CATEGORIE

IL DATABASE PRESENTA 7 UTENTI: UN AMMINISTRATORE, I DIRETTORI DEGLI AEROPORTI, IL GESTORE DEL SITO CHE VENDE BIGLIETTI ONLINE, I DIRETTORI DELLE COMPAGNIE, I PASSEGGERI, I PILOTI E GLI ADDETTI AI CONTROLLI. LA TAVOLA DEGLI UTENTI QUI PRESENTATA, NON EVIDENZIA I PRIVILEGI DI SISTEMA MA SOLO QUELLI DI OGGETTO

UTENTE	TIPO	VOLUME	PERMESSI
DB_AIRLINES	AMMINISTRATORE	1	ALL. ALL ON VOLO
DIRETTORE AEROPORTO	COMUNE	20	EXECUTE AGGIORNAMENTO VOLI. SELECT ON AEREI. SELECT ON DURATAVOLI. SELECT ON COMPAGNIE. SELECT ON SOSTITUZIONE
GESTORE SITO	COMUNE	1	SELECT ON VENDITE. EXECUTE CLASSE. EXECUTE ECONOMY
DIRETTORE COMPAGNIA	COMUNE	15	SELECT ON AEROPORTI
PASSEGGERO	COMUNE	80000	SELECT ON DURATAVOLI. EXECUTE CLASSE. EXECUTE ECONOMY
PILOTA	COMUNE	20	SELECT ON VOLIPILOTI
ADDETTO CONTROLLI	COMUNE	20	SELECT ON CONTROLLO

USE CASE



LE OPERAZIONI E GLI UTENTI COINVOLTI SONO SCHEMATIZZATI NEL
SEGUENTE DIAGRAMMA UML DEI CASI D'USO



OPERAZIONI DEGLI UTENTI

LE OPERAZIONI DEGLI UTENTI RAPPRESENTANO OPERAZIONI PIU' COMPLESSE DI SEMPLICI INSERIMENTI O MODIFICHE. PERTANTO SONO IMPLEMENTATE ATTRAVERSO LE PROCEDURE E SONO SCHEMATIZZATE TRAMITE L'UTILIZZO DELLE SCHEDE DESCRITTORE OPERAZIONE

AGGIORNAMENTO VOLI

OPERAZIONE	AGGIORNAMENTO VOLI
SCOPO:	CONTROLLARE IL PESO DELL'AEREO, STABILIRE IL NUMERO DI POSTI DISPONIBILI E AGGIORNARE L'ORA EFFETTIVA DEL VOLO
ARGOMENTI:	IATA, CODICE VOLO, ORA PARTENZA E ARRIVO EFFETTIVA, ICAO PARTENZA, ICAO ARRIVO E MARCA IMMATRICOLAZIONE
RISULTATO:	AGGIORNAMENTO POSTI DISPONIBILI E ORA EFFETTIVA VOLO, DIVIETO DI INSERIMENTO DI AEREI CON PESO INFERIORE A 10000/ERRORE
ERROTI:	PESO NON VALIDO
USA:	AEREO, CARTAIMBARCO, VOLO
MODIFICA:	AEREO, VOLO
PRIMA:	L'ORA EFFETTIVA DEL VOLO E I POSTI DISPONIBILI NON SONO AGGIORNATI E IL PESO È POSSIBILE INSERIRLO ANCHE MINORE DI 10000
POI:	ORA EFFETTIVA E POSTI DISPONIBILI AGGIORNATI E PESI DEGLI AEREI CHE VENGONO INSERITI SOLO SUPERIORI A 10000

CLASSE

OPERAZIONE	CLASSE
SCOPO:	DATO L'IMPORTO DEL PAGAMENTO, CONOSCERE SE SI VIAGGIA IN CLASSE BUSINESS
ARGOMENTI:	IMPORTO PAGAMENTO
RISULTATO:	CLASSE BUSINESS/ERRORE
ERRORI:	DATI NON PRESENTI
USA:	PAGAMENTO, BIGLIETTO, VOLO
MODIFICA:	BIGLIETTO
PRIMA:	L'IMPORTO DEL PAGAMENTO NON CORRISPONDEVA ALLA CLASSE DI VOLO
POI:	PER UN IMPORTO SUPERIORE A 500, LA CLASSE PUO' ESSERE BUSINESS

ECONOMY

OPERAZIONE	ECONOMY
SCOPO:	DATO L'IMPORTO DEL PAGAMENTO, CONOSCERE SE SI VIAGGIA IN CLASSE ECONOMY
ARGOMENTI:	IMPORTO PAGAMENTO
RISULTATO:	CLASSE ECONOMY/ERRORE
ERRORI:	DATI NON PRESENTI
USA:	PAGAMENTO, BIGLIETTO, VOLO
MODIFICA:	BIGLIETTO
PRIMA:	L'IMPORTO DEL PAGAMENTO NON CORRISPONDEVA ALLA CLASSE DI VOLO
POI:	PER UN IMPORTO SUPERIORE A 800, LA CLASSE NON PUO' ESSERE ECONOMY

TAVOLA DELLE OPERAZIONI

OPERAZIONE	TIPO	VOLUME	PERIODO
AGGIORNAMENTO VOLI	B	150	SETTIMANA
CLASSE BUSINESS	B	200	MESE
CLASSE ECONOMY	B	200	MESE



LA TAVOLA DELLE OPERAZIONI CONTA QUANTE VOLTE E' LANCIATA UN'OPERAZIONE SENZA CONSIDERARE L'ESITO

TAVOLA DEI VOLUMI

LA TAVOLA DEI VOLUMI RIPORTA IL NUMERO DI TUPLE PRESENTI IN CIASCUNA TABELLA ASSEGNAVANDO DEI VALORI VEROSIMILI. INOLTRE E' PRESENTE ANCHE LA VOCE INCREMENTO, LA QUALE INDICA DI QUANTO CAMBIA IL NUMERO DI TUPLE NELL'ARCO DI TEMPO INDICATO

TABELLA	TIPO	VOLUME	INCREMENTO	PERIODO
AEROPORTO	E	25	5	ANNO
CONTATTI AEROPORTO	ED	25	5	ANNO
VOLO	ED	500	25	GIORNO
COMPAGNIA	E	15	2	MESE
CONTATTI COMPAGNIA	ED	15	2	MESE
AEREO	E	150	10	ANNO
VIAGGIATORI	E	800000	20000	GIORNO
BIGLIETTO	E	800000	20000	GIORNO
È RELATIVO	A	800000	20000	GIORNO
PERSONALE DI BORDO	E	3000	2	GIORNO
PASSEGGERO	E	750000	20000	GIORNO
HA BAGAGLIO	A	500000	5000	GIORNO
LICENZA	E	3	0	ANNO
DOCUMENTO	E	800000	20000	GIORNO
PAGAMENTO	E	20000	2500	GIORNO
CARTA D'IMBARCO	ED	800000	20000	GIORNO
BAGAGLIO	E	500000	5000	GIORNO

VINCOLI STATICI

I VINCOLI STATICI SONO DELLE LIMITAZIONI IMPOSTE SUI VALORI CHE UN ATTRIBUTO PUO' ASSUMERE INDIPENDENTEMENTE DAL TEMPO

- I posti dell'aereo non devono essere inferiori a 20.
- L'ora partenza di un volo deve essere inferiore all'ora di arrivo.
- L'ora partenza effettiva di un volo deve essere inferiore all'ora di arrivo effettiva.
- I voli possono essere 'Nazionali' o 'Internazionali'.
- Il personale di bordo è composto da 'Piloti', 'Co-Piloti' e 'Hostess'.
- Le licenze possono essere per 'Piloti di aeromobili leggeri' o per 'Piloti di linea'.
- La data di conseguimento della licenza deve essere anteriore alla data odierna.
- L'ente di rilascio del documento d'identità può essere esclusivamente il 'Comune'.
- Le classi di voli si suddividono in 'J' (classe business), 'Y' (seconda classe) e 'F' (classe economy).



I VINCOLI DINAMICI RIGUARDANO I VALORI DEGLI ATTRIBUTI CHE VARIANO NEL TEMPO E DESCRIVONO L'EVOLUZIONE DEL SISTEMA

- Non possono essere inseriti dipendenti minorenni.
- La data di scadenza del documento d'identità deve essere posteriore alla data odierna.
- I biglietti non possono costare meno di 50.
- La data di scadenza della carta per acquisti online deve essere posteriore alla data odierna.
- Non possono esserci più di quattro dipendenti sullo stesso aereo.
- Non possono essere venduti oltre 99 biglietti per lo stesso volo.
- Non si possono portare con sé più di due bagagli.
- Non possono partire due aerei alla stessa ora.
- Non può essere assegnato a due passeggeri diversi lo stesso posto in aereo.
- Se due persone stanno comprando lo stesso biglietto, conta la data del pagamento.
- Non può essere venduto lo stesso biglietto a due persone diverse.

VINCOLI DINAMICI



VERIFICA DI NORMALITA'

LA VERIFICA DI NORMALITA' E' UTILIZZATA PER VALUTARE UNO SCHEMA DAL PUNTO DI VISTA FORMALE E PER MISURARE IL LIVELLO DI RIDONDANZA. LA PRIMA FORMA NORMALE RAPPRESENTA UNA PROPRIETA' ESTENSIONALE E QUINDI E' DEDUCIBILE DAL MODELLO RELAZIONALE. LA SECONDA E LA TERZA FORMA NORMALE RAPPRESENTANO, INVECE, UNA PROPRIETA' INTENSIONALE, OVVERO RIGUARDANO IL SIGNIFICATO DEGLI ATTRIBUTI

PRIMA FORMA NORMALE

La prima forma normale riguarda l'**atomicità**. Lo schema presenta attributi atomici fatta eccezione per le date che però possono essere accettate. Quindi lo schema è in **prima forma normale**





SECONDA FORMA NORMALE

Sono presenti **chiavi multiattributo** nell'entità Carta d'Imbarco, nell'entità Volo e nelle associazioni M a N. Nessuna di queste determina parzialmente gli attributi, i quali dipendono dalla chiave multiattributo. La **seconda forma normale** è rispettata



TERZA FORMA NORMALE

Nello schema non sono presenti dipendenze funzionali anomale, ma le dipendenze presenti sono tutte da chiave. Lo schema è quindi in **terza forma normale**

BCNF

Qualsiasi superchiave scelta, identifica in modo univoco ogni tupla. Lo schema è in BCNF



IMPLEMENTAZIONE

CREAZIONE UTENTI

IL PRIMO PASSO DELL'IMPLEMENTAZIONE E' QUELLO DI CREARE GLI UTENTI DOPO AVER EFFETTUATO L'ACCESSO COME AMMINISTRATORE

- CREATE USER DB_AIRLINES IDENTIFIED BY ADMIN;
- CREATE USER DIRETTORE_AEROPORTO IDENTIFIED BY ICAO123;
- CREATE USER GESTORE_SITO IDENTIFIED BY SHOP;
- CREATE USER DIRETTORE_COMPAGNIA IDENTIFIED BY IATA123;
- CREATE USER PASSEGGERO IDENTIFIED BY P12345;
- CREATE USER PILOTA IDENTIFIED BY PILOT747;
- CREATE USER ADDETTO_CONTROLLI IDENTIFIED BY CTRL001;
- GRANT ALL PRIVILEGES TO DB_AIRLINES;



DATA DEFINITION LANGUAGE

IL DDL RIFLETTE TUTTO CIO'
CHE E' STATO INSERITO
NELLO SCHEMA
RELAZIONALE E HA COME
FUNZIONI QUELLE DI
MODIFICARE, CREARE O
ELIMINARE LE STRUTTURE,
OVVERO LE RELAZIONI



LA TABELLA AEROPORTO E LA TABELLA COMPAGNIA NON PRESENTANO VINCOLI



AEROPORTO

```
create table Aeroporto(  
Icao char(4) primary key,  
Num_Piste number(1,0),  
Citta varchar(20)  
);
```

COMPAGNIA

```
create table Compagnia(  
Iata char(3) primary key,  
Nome_Comp varchar(20)  
);
```

LE TABELLE RELATIVE AI CONTATTI HANNO SOLO VINCOLI RIGUARDANTI L'INSERIMENTO DI EMAIL E PEC

CONTATTI_AEROPORTO

```
create table Contatti_Aeroporto(  
  
    PEC_A varchar(30),  
  
    Email_A varchar(30) unique,  
  
    Tel1_A varchar(15) not null,  
  
    Tel2_A varchar(15) not null,  
  
    Fax_A char(15),  
  
    Icao char(4),  
  
    CONSTRAINT email CHECK (REGEXP_LIKE(Email_A,'^\\w+@\\w+\\.\\w+$')),  
  
    CONSTRAINT pec CHECK (REGEXP_LIKE(PEC_A,'^\\w+@\\w+\\.\\w+$')),  
  
    constraint FK_Icao foreign key(Icao) references Aeroporto(Icao),  
  
    constraint PK_Icao primary key (Icao, PEC_A)  
);
```

CONTATTI_COMPAGNIA

```
create table Contatti_Compagnia(  
  
    PEC_C varchar(30),  
  
    Email_C varchar(30) unique,  
  
    Tel1_C varchar(15) not null,  
  
    Tel2_C varchar(15) not null,  
  
    Fax_C char(15),  
  
    Iata char(3),  
  
    constraint FK_Iata2 foreign key(Iata) references Compagnia(Iata),  
  
    CONSTRAINT email2 CHECK (REGEXP_LIKE(Email_C,'^\\w+@\\w+\\.\\w+$')),  
  
    CONSTRAINT pec2 CHECK (REGEXP_LIKE(PEC_C,'^\\w+@\\w+\\.\\w+$')),  
  
    constraint PK_Iata primary key(Iata, PEC_C)  
);
```

GLI AEREI DEVONO AVERE PIU' DI 20 POSTI
MENTRE I BAGAGLI DEVONO ESSERE A MANO O DA
STIVA ED AVERE UN PESO INFERIORE O UGUALE A
32 KG

AEREO

```
create table Aereo(  
  
Marca_Imm varchar(6) primary key,  
  
Peso number(5,0),  
  
Potenza varchar(7),  
  
N_Posti number(3,0) not null,  
  
constraint LimitePosti check(N_Posti>20)  
);
```

BAGAGLIO

```
create table Bagaglio(  
  
Tag varchar(12) primary key,  
  
Tipo varchar(8) check (Tipo in('A mano', 'Da stiva')),  
  
Peso_B number(2,0) check (Peso_B<=32)  
);
```

VOLO

LA TABELLA VOLO PRESENTA VINCOLI RIGUARDANTI
GLI ORARI DI PARTENZA (MINORI DI QUELLI DI ARRIVO)
E SUL TIPO DI VOLO, IL QUALE PUO' ESSERE
NAZIONALE O INTERNAZIONALE

```
create table Volo(
    Cod_V varchar(7),
    Durata number(3,0),
    Ora_Part date,
    Ora_Arr date,
    Tipo_Volo varchar(14),
    Ora_Part_Eff date,
    Ora_Arr_Eff date,
    Icao_P char(4),
    Icao_A char(4),
    lata char(3),
    Marca_Imm varchar(6),
    constraint Ora check (Ora_Part<Ora_Arr),
    constraint OraF check
        (Ora_Part_Eff<Ora_Arr_Eff),
    constraint TipoVolo check( Tipo_Volo in
        ('Nazionale', 'Internazionale')),
    constraint FK_Icao2 foreign key(Icao_P)
        references Aeroporto(Icao),
    constraint FK_Icao3 foreign key(Icao_A)
        references Aeroporto(Icao),
    constraint FK_lata foreign key(lata) references
        Compagnia(lata),
    constraint FK_Marcalmm foreign
        key(Marca_Imm) references Aereo(Marca_Imm),
    constraint PK_Volo primary key (lata, Cod_V)
);
```

VIAGGIATORI

```
create table Viaggiatori(
```

```
    ID_Viag varchar(4) primary key,
```

```
    Nome varchar(20),
```

```
    Cognome varchar(15),
```

```
    Tel_V varchar(15) unique,
```

```
    DN date,
```

```
    Email_V varchar(30) unique
```

```
CONSTRAINT email3 CHECK  
(REGEXP_LIKE>Email_V,'^\\w+@\\w+\\.\\w+$')
```

```
);
```

LA TABELLA VIAGGIATORI
PRESENTA SOLO IL VINCOLO PER
LA SCRITTURA DELL'EMAIL



IL PERSONALE DI BORDO E' COMPOSTO SOLO DA PILOTI, CO-PILOTI E HOSTESS, MENTRE LA TABELLA LICENZA HA VINCOLI SULLA DATA DI CONSEGUIMENTO (PRECEDENTE AL 24 MAGGIO) E SUL TIPO DI LICENZA (PILOTA DI LINEA O DI AEROMOBILI LEGGERI)

PERSONALEBORDO

```
create table PersonaleBordo(  
    Matricola varchar(7) primary key,  
    Ruolo varchar(9),  
    ID_Viag varchar(4),  
    Cod_V varchar(7),  
    Iata char(3),  
    constraint Ruolo check (Ruolo in ('Pilota', 'Co-pilota',  
    'Hostess')),  
    constraint FKID_Viag foreign key(ID_Viag) references  
    Viaggiatori(ID_Viag),  
    constraint FKCod_V foreign key(Iata, Cod_V) references  
    Volo(Iata, Cod_V)  
);
```

LICENZA

```
create table Licenza(  
    Cod_L char(5) primary key,  
    Tipo_L varchar (30),  
    Data_Cons date,  
    Matricola varchar(7),  
    constraint Tipo_L check(Tipo_L in('Pilota di aeromobili  
leggeri', 'Pilota di linea')),  
    constraint Data_Cons check  
(Data_Cons<TO_DATE('24/may/2024 00:00:00',  
'dd-mon/yyyy hh24:mi:ss')),  
    constraint FKMat foreign key(Matricola) references  
    PersonaleBordo(Matricola)  
);
```

LE TABELLE PASSEGGERO E CARTAIMBARCO NON HANNO VINCOLI PARTICOLARI

PASSEGGERO

```
create table Passeggero(  
    CF char(16) primary key,  
    N_Passaporto char(9) unique,  
    ID_Viag varchar(4),  
    constraint FKViag foreign key(ID_Viag) references  
    Viaggiatori(ID_Viag)  
)
```

CARTAIMBARCO

```
create table Cartaimbarco(  
    Numero varchar(20),  
    N_Posto varchar(4),  
    Data_V date,  
    Cod_V varchar(7),  
    lata char(3),  
    CF char(16),  
    constraint FKCV foreign key(lata, Cod_V) references Volo(lata,  
    Cod_V),  
    constraint FKCF3 foreign key(CF) references Passeggero(CF),  
    constraint PKCB primary key (Numero, lata, Cod_V)  
)
```

I DOCUMENTI DEVONO ESSERE RILASCIATI NECESSARIAMENTE DAL COMUNE

DOCUMENTO

```
create table Documento(  
    N_Doc char(9) primary key,  
    Ente_Ril char(6),  
    Data_Scad date,  
    CF char(16),  
    constraint Ente_Ril check(Ente_Ril  
        in('Comune')),  
    constraint FKCF foreign key(CF) references  
        Passeggero(CF)  
);
```

HABAGAGLIO

```
create table HaBagaglio(  
    CF char(16),  
    Tag varchar(12),  
    constraint FKCF2 foreign key(CF) references  
        Passeggero(CF),  
    constraint FKTag foreign key(Tag) references  
        Bagaglio(Tag),  
    constraint PKHB primary key(CF, Tag)  
);
```

BIGLIETTO

LA CLASSE DI VOLO PUO' ESSERE J
(CLASSE BUSINESS), Y (SECONDA
CLASSE) E F (ECONOMY)

create table Biglietto(

Cod_B varchar(20) primary key,

Classe char(1),

Cod_V varchar(7),

Iata char(3),

CF char(16),

constraint Classe check(Classe in ('J', 'F', 'Y')),

constraint FKCV3 foreign key(Iata, Cod_V) references Volo(Iata, Cod_V),

constraint FKCF5 foreign key(CF) references Passeggero(CF)

);



LE TABELLE PAGAMENTO ED ERIFERITO NON PRESENTANO VINCOLI PARTICOLARI

PAGAMENTO

```
create table Pagamento(  
    N_Carta char(16) primary key,  
    Importo number(4,0),  
    Data_Scad_Cart date,  
    CF char(16),  
    constraint FKCF4 foreign key(CF) references  
    Passeggero(CF)
```

```
);
```



ERIFERITO

```
create table ERiferito(  
    N_Carta char(16),  
    Cod_B varchar(20),  
    Data_Pag date,  
    constraint FKPB foreign key(N_Carta)  
    references Pagamento(N_Carta),  
    constraint FKPB2 foreign key(Cod_B)  
    references Biglietto(Cod_B),  
    constraint PKPB primary key(N_Carta, Cod_B)  
);
```

DATA MANIPULATION LANGUAGE

IL DML E' L'INSIEME DI
OPERAZIONI UTILIZZATE PER
INSERIRE, RIMUOVERE O
MODIFICARE I DATI

```
    if operation == "MIRROR_X":  
        mirror_mod.use_x = True  
        mirror_mod.use_y = False  
        mirror_mod.use_z = False  
    elif operation == "MIRROR_Y":  
        mirror_mod.use_x = False  
        mirror_mod.use_y = True  
        mirror_mod.use_z = False  
    elif operation == "MIRROR_Z":  
        mirror_mod.use_x = False  
        mirror_mod.use_y = False  
        mirror_mod.use_z = True  
  
#selection at the end -add  
#    bpy.context.scene.objects.active  
#    bpy.context.scene.objects.active.select= 1  
#    mirror_ob.select=1  
#    context.scene.objects.active.select= 0  
#  
#    print("Selected" + str(modifier))  
#    mirror_ob.select = 0  
#    bpy.context.selected_objects[0].select= 1  
#    data.objects[one.name].select= 1  
  
#    print("please select exactly one object")  
  
- OPERATOR CLASSES ---
```

```
class MirrorOperator(bpy.types.Operator):  
    bl_idname = "object.mirror"  
    bl_label = "X mirror to the selected"  
    bl_options = {'REGISTER', 'UNDO'}  
    bl_description = "mirror mirror_x"  
    bl_context = "object mode"  
    bl_space_type = "VIEW_3D"  
  
    def execute(self, context):  
        if context.active_object is not None:  
            ob = context.active_object
```

AEROPORTO

```
insert into Aeroporto values ('LIMC', 2, 'Milano')
insert into Aeroporto values ('LIRF', 4, 'Roma')
insert into Aeroporto values ('LIRN', 1, 'Napoli')
insert into Aeroporto values ('ZBAD', 7, 'Pechino')
insert into Aeroporto values ('LEMD', 4, 'Madrid')
insert into Aeroporto values ('LFPO', 3, 'Parigi')
insert into Aeroporto values ('KJFK', 4, 'New York')
insert into Aeroporto values ('EDDB', 2, 'Berlino')
insert into Aeroporto values ('EHAM', 6, 'Amsterdam')
insert into Aeroporto values ('KLAX', 4, 'Los Angeles')
insert into Aeroporto values ('LEBL', 3, 'Barcellona')
insert into Aeroporto values ('OMDB', 2, 'Dubai')
```



COMPAGNIA

```
insert into Compagnia values ('RYR', 'Ryanair')
insert into Compagnia values ('UAE', 'Emirates')
insert into Compagnia values ('EZY', 'easyJet')
insert into Compagnia values ('DLH', 'Lufthansa')
insert into Compagnia values ('AFR', 'Air France')
insert into Compagnia values ('BAW', 'British Airways')
insert into Compagnia values ('JAL', 'Japan Airlines')
insert into Compagnia values ('ABB', 'Air Belgium')
insert into Compagnia values ('SWR', 'Swiss')
insert into Compagnia values ('CCA', 'Air China')
insert into Compagnia values ('OAL', 'Olympic Air')
insert into Compagnia values ('TAP', 'Tap Portugal')
```





CONTATTI_AEROPORTO

```
insert into Contatti_Aeroporto  
values('doganemalpensa@pecgov.it',  
'doganemalpensa@adm.it', '02 232323', '02  
74863018', '02 69633615', 'LIMC')
```

```
insert into Contatti_Aeroporto  
values('doganeroma2@pecgov.it',  
'doganeroma2@adm.it', '06 65951', '39  
0665957249', '06 659520954', 'LIRF')
```

```
insert into Contatti_Aeroporto  
values('doganenapoli1@pecgov.it',  
'doganenapoli1@adm.it', '081 7896259', '081  
19334619', '39 0817802546', 'LIRN')
```

```
insert into Contatti_Aeroporto  
values('BeijingAirport@PEC.IT', 'beijingair@bj.com', '02  
69633602', '86 2168341000', '86 2168342735',  
'ZBAD')
```

```
insert into Contatti_Aeroporto  
values('MadridAir@PEC.IT', 'clientesmad@aena.es',  
'34 91 3211000', '913 938380', '34 917466112',  
'LEMD')
```

```
insert into Contatti_Aeroporto  
values('AirParis@PEC.IT',  
'serviceclients@aeroport.com', '33 148622280', '33  
892563950', '33 148620752', 'LFPO')
```

```
insert into Contatti_Aeroporto values('NYAir@PEC.IT',  
'newyorkinfo@esteri.it', '1 (718)2444444', '1  
(718)9173999', '1 (212)2494945', 'KJFK')
```

CONTATTI_COMPAGNIA



insert into Contatti_Compagnia values('FRITALY@LEGALMAIL.IT',
'ryanairen@ryanair.ie', '02 89980500', ' 800582717', '3538121672', 'RYR')

insert into Contatti_Compagnia values('EMIRATESITALY@PEC.IT',
'emiratesen@emirates.ie', '39 02 914 83383', ' 06 6501 1417', '971 4
2864120', 'UAE')

insert into Contatti_Compagnia values('EASYJET@LEGALMAIL.IT',
'groupit@easyJet.com', '39 0232068899', ' 44 3305515151', '44
1582443355', 'EZY')

insert into Contatti_Compagnia values ('AMMINISTRAZIONELH@PEC.IT',
'customerrelation@lufthansa.com', '39 0899198000', ' 39 0899198004',
'02 58372226', 'DLH')

insert into Contatti_Compagnia values('AFADMIN@PECAFKL.IT',
'groupit@airfrance.com', '02 38591272', ' 06 48791', '06 4817079', 'AFR')

insert into Contatti_Compagnia values('BRITISHAIRWAYSPLC@PEC.IT',
'groupsitaly@ba.com', '02 69633602', ' 39 0689386034', '02 69633615',
'BAW')

insert into Contatti_Compagnia values('TAPPORTUGAL@LEGALMAIL.IT',
'info@tapportugal.com', '39 0269682334', '39 0269682334', '351
707205700', 'TAP')

AEREO



insert into Aereo values ('I-AAAB', 22680, '170 kW', 70)
insert into Aereo values ('I-CACB', 23000, '200 kW', 50)
insert into Aereo values ('B-2000', 23500, '175 kW', 70)
insert into Aereo values ('B-2010', 23100, '210 kW', 50)
insert into Aereo values ('EC-AAB', 22700, '175 kW', 80)
insert into Aereo values ('EC-RTT', 23050, '110 kW', 25)
insert into Aereo values ('F-CAAB', 23400, '150 kW', 70)
insert into Aereo values ('N-9800', 23350, '160 kW', 55)
insert into Aereo values ('D-VERL', 24000, '200 kW', 100)
insert into Aereo values ('PH-AAA', 22890, '130 kW', 65)
insert into Aereo values ('HS-POL', 22780, '120 kW', 60)
insert into Aereo values ('SU-AAC', 22950, '100 kW', 50)

```
insert into Volo values('RYR001', 135, TO_DATE('03/mar/2024 21:00:00', 'dd/month/yyyy hh24:mi:ss'), TO_DATE('03/mar/2024 23:15:00', 'dd/month/yyyy hh24:mi:ss'), 'Internazionale', TO_DATE('03/mar/2024 21:05:44', 'dd/month/yyyy hh24:mi:ss'), TO_DATE('03/mar/2024 23:20:00', 'dd/month/yyyy hh24:mi:ss'), 'LIRN', 'LFPO', 'RYR', 'I-AAAB')
```

```
insert into Volo values('EZY001', 95, TO_DATE('25/may/2024 21:55:00', 'dd/month/yyyy hh24:mi:ss'), TO_DATE('25/may/2024 23:30:00', 'dd/month/yyyy hh24:mi:ss'), 'Nazionale', TO_DATE('25/may/2024 21:58:00', 'dd/month/yyyy hh24:mi:ss'), TO_DATE('25/may/2024 23:35:00', 'dd/month/yyyy hh24:mi:ss'), 'LIRN', 'LIMC', 'EZY', 'I-CACB')
```

```
insert into Volo values('AFR001', 130, TO_DATE('01/jun/2024 18:25:00', 'dd/month/yyyy hh24:mi:ss'), TO_DATE('01/jun/2024 20:35:00', 'dd/month/yyyy hh24:mi:ss'), 'Internazionale', TO_DATE('01/jun/2024 18:30:00', 'dd/month/yyyy hh24:mi:ss'), TO_DATE('01/jun/2024 20:40:00', 'dd/month/yyyy hh24:mi:ss'), 'LIRF', 'LFPO', 'AFR', 'I-RSTS')
```

```
insert into Volo values('DLH001', 800, TO_DATE('04/jul/2024 19:50:00', 'dd/month/yyyy hh24:mi:ss'), TO_DATE('05/jul/2024 08:00:00', 'dd/month/yyyy hh24:mi:ss'), 'Internazionale', TO_DATE('04/jul/2024 20:00:00', 'dd/month/yyyy hh24:mi:ss'), TO_DATE('05/jul/2024 08:10:00', 'dd/month/yyyy hh24:mi:ss'), 'EDDB', 'VTBD', 'DLH', 'D-AAAB')
```

```
insert into Volo values('JAL001', 665, TO_DATE('18/jul/2024 12:40:00', 'dd/month/yyyy hh24:mi:ss'), TO_DATE('18/jul/2024 23:40:00', 'dd/month/yyyy hh24:mi:ss'), 'Internazionale', TO_DATE('18/jul/2024 12:42:00', 'dd/month/yyyy hh24:mi:ss'), TO_DATE('18/jul/2024 23:43:00', 'dd/month/yyyy hh24:mi:ss'), 'LGAV', 'KJFK', 'JAL', 'EC-FFG')
```

```
insert into Volo values('ABB001', 718, TO_DATE('30/mar/2024 16:00:00', 'dd/month/yyyy hh24:mi:ss'), TO_DATE('31/mar/2024 03:58:00', 'dd/month/yyyy hh24:mi:ss'), 'Internazionale', TO_DATE('30/mar/2024 16:02:00', 'dd/month/yyyy hh24:mi:ss'), TO_DATE('31/mar/2024 02:00:00', 'dd/month/yyyy hh24:mi:ss'), 'LEMD', 'ZBAD', 'ABB', 'EC-FFG')
```

```
insert into Volo values('BAW004', 530, TO_DATE('26/apr/2024 06:50:00', 'dd/month/yyyy hh24:mi:ss'), TO_DATE('26/apr/2024 14:50:00', 'dd/month/yyyy hh24:mi:ss'), 'Internazionale', TO_DATE('26/apr/2024 06:52:00', 'dd/month/yyyy hh24:mi:ss'), TO_DATE('26/apr/2024 14:51:00', 'dd/month/yyyy hh24:mi:ss'), 'ZBAD', 'OMDB', 'BAW', 'B-9872')
```

VOLO



BAGAGLIO



Insert into Bagaglio values('LIRNLFPO001', 'A mano', 5)
Insert into Bagaglio values('LIRNLFPO002', 'A mano', 7)
Insert into Bagaglio values('LIRFLFPO003', 'A mano', 6)
Insert into Bagaglio values('EDDBVTBD001', 'A mano', 8)
Insert into Bagaglio values('LIRNLIMC002', 'A mano', 7)
Insert into Bagaglio values('LGAVKJFK003', 'A mano', 10)
Insert into Bagaglio values('LFPOKLAX001', 'A mano', 10)
Insert into Bagaglio values('LFPOKLAX002', 'Da stiva', 27)
Insert into Bagaglio values('LEMDLIRF002', 'A mano', 10)
Insert into Bagaglio values('LEMDLIRF003', 'A mano', 8)
Insert into Bagaglio values('RJTTZSPD001', 'A mano', 7)
Insert into Bagaglio values('KJFKHECA001', 'Da stiva', 23)
Insert into Bagaglio values('KJFKHECA002', 'A mano', 7)

VIAGGIATORI



Insert into Viaggiatori values ('0001', 'Mario', 'Rossi', '3334449987',
TO_DATE('12/jul/1990', 'dd/month/yyyy'), 'Mariorossi@gmail.com')

Insert into Viaggiatori values ('0002', 'Maria', 'Verdi', '3216549870',
TO_DATE('01/feb/1978', 'dd/month/yyyy'), 'Mariaverd@gmail.com')

Insert into Viaggiatori values ('0012', 'Roberta', 'Giordano',
'329099876', TO_DATE('05/aug/2004', 'dd/month/yyyy'),
'Rob@gmail.com')

Insert into Viaggiatori values ('0013', 'Francesco', 'Cenci',
'3409893459', TO_DATE('30/jan/2003', 'dd/month/yyyy'),
'Franc@gmail.com')

Insert into Viaggiatori values ('0014', 'Melissa', 'Jones', '45677465',
TO_DATE('25/dec/2005', 'dd/month/yyyy'), 'mmali@icloud.it')

Insert into Viaggiatori values ('0026', 'Giorgio', 'Bianchi', '345678902',
TO_DATE('10/aug/1950', 'dd/month/yyyy'), 'Gb@gmail.com')

Insert into Viaggiatori values ('0027', 'Ava', 'Liu', '3768992300',
TO_DATE('25/jun/1985', 'dd/month/yyyy'), 'aveliina@icloud.it')

Insert into Viaggiatori values ('0040', 'Igor', 'Miti', '8976543029',
TO_DATE('20/jul/1948', 'dd/month/yyyy'), 'mitini@gmail.com')

Insert into Viaggiatori values ('0041', 'Claudio', 'Langella',
'8009005000', TO_DATE('17/aug/1950', 'dd/month/yyyy'),
'claudiolangella71@gmail.com')



PERSONALEBORDO

Insert into PersonaleBordo values ('0450001', 'Pilota', '0001', 'RYR001', 'RYR')

Insert into PersonaleBordo values ('0450002', 'Co-pilota', '0004', 'RYR001', 'RYR')

Insert into PersonaleBordo values ('0450003', 'Hostess', '0008', 'RYR001', 'RYR')

Insert into PersonaleBordo values ('0450009', 'Co-pilota', '0012', 'BAW002', 'BAW')

Insert into PersonaleBordo values ('0450010', 'Hostess', '0006', 'BAW002', 'BAW')

Insert into PersonaleBordo values ('0450011', 'Pilota', '0014', 'AFR001', 'AFR')

Insert into PersonaleBordo values ('0450019', 'Hostess', '0019', 'DLH003', 'DLH')

Insert into PersonaleBordo values ('0450020', 'Pilota', '0020', 'UAE001', 'UAE')

Insert into PersonaleBordo values ('0450021', 'Co-pilota', '0022', 'UAE001', 'UAE')

Insert into PersonaleBordo values ('0450022', 'Hostess', '0024', 'UAE001', 'UAE')

Insert into PersonaleBordo values ('0450015', 'Co-pilota', '0003', 'EZY002', 'EZY')

Insert into PersonaleBordo values ('0450016', 'Hostess', '0007', 'EZY002', 'EZY')

LICENZA

Insert into Licenza values ('LAPL', 'Pilota di linea',
TO_DATE('08/aug/2008', 'dd/month/yyyy'), '0450001')

Insert into Licenza values ('LAPL2', 'Pilota di linea',
TO_DATE('10/sep/2010', 'dd/month/yyyy'), '0450002')

Insert into Licenza values ('ATPL', 'Pilota di aeromobili leggeri',
TO_DATE('07/may/2015', 'dd/month/yyyy'), '0450005')

Insert into Licenza values ('ATPL2', 'Pilota di aeromobili leggeri',
TO_DATE('26/jan/2000', 'dd/month/yyyy'), '0450006')

Insert into Licenza values ('ATPL5', 'Pilota di aeromobili leggeri',
TO_DATE('05/jul/2004', 'dd/month/yyyy'), '0450017')

Insert into Licenza values ('ATPL6', 'Pilota di aeromobili leggeri',
TO_DATE('04/sep/1995', 'dd/month/yyyy'), '0450018')

Insert into Licenza values ('LAPL7', 'Pilota di linea',
TO_DATE('10/feb/2012', 'dd/month/yyyy'), '0450020')

Insert into Licenza values ('ATPL7', 'Pilota di aeromobili leggeri',
TO_DATE('11/dec/2022', 'dd/month/yyyy'), '0450021')

Insert into Licenza values ('ATPL8', 'Pilota di aeromobili leggeri',
TO_DATE('10/mar/2020', 'dd/month/yyyy'), '0450066')





PASSEGGERO

Insert into Passeggero values ('GRNLNE01C43G273A', 'AB0000001', '0021')

Insert into Passeggero values ('SRDFNC59C42G081K', 'AB0000002', '0023')

Insert into Passeggero values ('MNTLRT07C22A500N', 'AB0000003', '0025')

Insert into Passeggero values ('PRGNHL04M02C070L', 'AB0000013', '0035')

Insert into Passeggero values ('RDTMDL08C65A558V', 'AB0000014', '0036')

Insert into Passeggero values ('SVVNSH79A43D415J', 'AB0000015', '0037')

Insert into Passeggero values ('KMRMRA75L53F378O', 'AB0000016', '0038')

Insert into Passeggero values ('BRSDDA05T25F839S', 'AB0000017', '0039')

Insert into Passeggero values ('MTIGRI48L20F205V', 'AB0000018', '0040')

Insert into Passeggero values ('MNTDMN96R24L521V', 'AB0000040', '0062')

Insert into Passeggero values ('TRRLND03T47E319P', 'AB0000041', '0063')

Insert into Passeggero values ('GTTGLC81S08L521H', 'AB0000042', '0064')

Insert into Passeggero values ('LVRNGL92C70D024T', 'AB0000043', '0065')

DOCUMENTO



Insert into documento values ('CA00000AA', 'Comune', TO_DATE('22/may/2028', 'dd/month/yyyy'), 'GRNLNE01C43G273A')

Insert into documento values ('CA00001BB', 'Comune', TO_DATE('24/jan/2029', 'dd/month/yyyy'), 'SRDFNC59C42G081K')

Insert into documento values ('CA00002CC', 'Comune', TO_DATE('06/jan/2028', 'dd/month/yyyy'), 'MNTLRT07C22A500N')

Insert into documento values ('CA00010BB', 'Comune', TO_DATE('31/jan/2025', 'dd/month/yyyy'), 'MNFNTN82M10D292R')

Insert into documento values ('CA00013GG', 'Comune', TO_DATE('24/may/2025', 'dd/month/yyyy'), 'SHRSHA08H45C698U')

Insert into documento values ('CA00079OO', 'Comune', TO_DATE('13/jul/2030', 'dd/month/yyyy'), 'TTNNEE07L19C737W')

Insert into documento values ('CA00990AA', 'Comune', TO_DATE('23/may/2028', 'dd/month/yyyy'), 'MTIGRI48L20F205V')

Insert into documento values ('CA00009BS', 'Comune', TO_DATE('02/apr/2026', 'dd/month/yyyy'), 'LNGCLD50M17A300U')

Insert into documento values ('CB00000VV', 'Comune', TO_DATE('05/sep/2035', 'dd/month/yyyy'), 'RRVHRY13S23C698N')

HABAGAGLIO

Insert into HaBagaglio values ('GRNLNE01C43G273A', 'LIRNLFPO001')

Insert into HaBagaglio values ('SRDFNC59C42G081K', 'LIRNLFPO002')

Insert into HaBagaglio values ('MNTLRT07C22A500N', 'LIRNLFPO003')

Insert into HaBagaglio values ('BNCGRG50M10F839P', 'LIRNLIMC001')

Insert into HaBagaglio values ('PLSGNN78E06F835Y', 'LIMCLICC003')

Insert into HaBagaglio values ('RRVJMN96B68G014O', 'LEMDZBAD001')

Insert into HaBagaglio values ('PTTSRN09L55E389R', 'ESSALEBL004')

Insert into HaBagaglio values ('NRDGNS69E42F974Q', 'EHAMLIICC001')

Insert into HaBagaglio values ('NRDMRA99D29Z160A', 'EHAMLIICC002')

Insert into HaBagaglio values ('CSCCRD04C10A126S', 'RJTTZSPD003')

Insert into HaBagaglio values ('TNKKAI08M08A558B', 'LFPOLIRN001')

Insert into HaBagaglio values ('LVRNGL92C70D024T', 'KJFKHECA001')

Insert into HaBagaglio values ('LVRNGL92C70D024T', 'LIRFLICC001')



CARTAIMBARCO



Insert into Cartalmbarco values ('100000020', 'AB00', TO_DATE('03/mar/2024', 'dd-mon/yyyy'), 'RYR001', 'RYR', 'GRNLNE01C43G273A')

Insert into Cartalmbarco values ('100000021', 'AB01', TO_DATE('28/dec/2024', 'dd-mon/yyyy'), 'RYR002', 'RYR', 'SRDFNC59C42G081K')

Insert into Cartalmbarco values ('100000022', 'AB00', TO_DATE('28/dec/2024', 'dd-mon/yyyy'), 'RYR002', 'RYR', 'MNTLRT07C22A500N')

Insert into Cartalmbarco values ('100000030', 'AC22', TO_DATE('21/jun/2024', 'dd-mon/yyyy'), 'AFR002', 'AFR', 'CHNDNG80D04Z729F')

Insert into Cartalmbarco values ('100000031', 'AD20', TO_DATE('21/jun/2024', 'dd-mon/yyyy'), 'AFR002', 'AFR', 'NRDZRR10E42B164Y')

Insert into Cartalmbarco values ('100000032', 'AB04', TO_DATE('02/aug/2024', 'dd-mon/yyyy'), 'UAE001', 'UAE', 'BRSDDA05T25F839S')

Insert into Cartalmbarco values ('100000033', 'AB22', TO_DATE('02/aug/2024', 'dd-mon/yyyy'), 'UAE001', 'UAE', 'PLSGNN78E06F835Y')

Insert into Cartalmbarco values ('100000036', 'AB07', TO_DATE('03/apr/2024', 'dd-mon/yyyy'), 'IBE002', 'IBE', 'TTNNSEE07L19C737W')

Insert into Cartalmbarco values ('100000037', 'AB08', TO_DATE('03/apr/2024', 'dd-mon/yyyy'), 'IBE002', 'IBE', 'CSCCRD04C10A126S')

Insert into Cartalmbarco values ('100000041', 'AV03', TO_DATE('07/may/2024', 'dd-mon/yyyy'), 'BAW001', 'BAW', 'HFRNGS69H07E320S')

PAGAMENTO



Insert into Pagamento values ('5333171000000000', 75,
TO_DATE('30/apr/2027', 'dd-mon/yyyy'), 'GRNLNE01C43G273A')

Insert into Pagamento values ('5333171000001000', 900,
TO_DATE('22/nov/2026', 'dd-mon/yyyy'), 'SRDFNC59C42G081K')

Insert into Pagamento values ('5333171000000020', 750,
TO_DATE('29/jan/2030', 'dd-mon/yyyy'), 'MNTLRT07C22A500N')

Insert into Pagamento values ('5333171000301000', 550,
TO_DATE('22/nov/2026', 'dd-mon/yyyy'), 'RRVJMN96B68G014O')

Insert into Pagamento values ('5333171000001010', 430,
TO_DATE('03/sep/2035', 'dd-mon/yyyy'), 'PTTSRN09L55E389R')

Insert into Pagamento values ('5333171100001000', 735,
TO_DATE('30/mar/2031', 'dd-mon/yyyy'), 'NRDMRA99D29Z160A')

Insert into Pagamento values ('5333171104201000', 1000,
TO_DATE('26/feb/2027', 'dd-mon/yyyy'), 'LVRNGL92C70D024T')

Insert into Pagamento values ('5333171100511000', 80,
TO_DATE('20/aug/2025', 'dd-mon/yyyy'), 'THMCRL69E42Z510M')

Insert into Pagamento values ('5333171100060000', 110,
TO_DATE('18/jul/2024', 'dd-mon/yyyy'), 'NRDGNS69E42F974Q')

Insert into Pagamento values ('5333171100061711', 300,
TO_DATE('14/jan/2030', 'dd-mon/yyyy'), 'HBRLBN09L55A558P')

BIGLIETTO

Insert into Biglietto values ('10000098720009110000', 'Y', 'RYR001', 'RYR', 'GRNLNE01C43G273A')

Insert into Biglietto values ('10000098720009110001', 'Y', 'UAE001', 'UAE', 'SRDFNC59C42G081K')

Insert into Biglietto values ('10000098720009110002', 'J', 'UAE001', 'UAE', 'MNTLRT07C22A500N')

Insert into Biglietto values ('10000098720009110006', 'Y', 'DLH001', 'DLH', 'RRVJMN96B68G014O')

Insert into Biglietto values ('10000098720009110007', 'F', 'DLH001', 'DLH', 'PTTSRN09L55E389R')

Insert into Biglietto values ('10000098720009110008', 'J', 'RYR002', 'RYR', 'NRDMRA99D29Z160A')

Insert into Biglietto values ('10000098720009110014', 'Y', 'RYR004', 'RYR', 'THMCRL69E42Z510M')

Insert into Biglietto values ('10000098720009110015', 'F', 'IBE003', 'IBE', 'NRDGNS69E42F974Q')

Insert into Biglietto values ('10000098720009110016', 'F', 'EZY003', 'EZY', 'HBRLBN09L55A558P')

Insert into Biglietto values ('10000098720009110017', 'Y', 'DLH003', 'DLH', 'MGLVTR20L20F836E')



ERIFERITO

Insert into ERiferito values ('5333171000000000', '10000098720009110000',
TO_DATE('26/feb/2024', 'dd/month/yyyy'))

Insert into ERiferito values ('5333171000001000', '10000098720009110001',
TO_DATE('20/dec/2024', 'dd/month/yyyy'))

Insert into ERiferito values ('5333171000000020', '10000098720009110002',
TO_DATE('20/dec/2024', 'dd/month/yyyy'))

Insert into ERiferito values ('5333171000001010', '10000098720009110007',
TO_DATE('02/aug/2024', 'dd/month/yyyy'))

Insert into ERiferito values ('5333171100001000', '10000098720009110008',
TO_DATE('21/nov/2024', 'dd/month/yyyy'))

Insert into ERiferito values ('5333171104001000', '10000098720009110009',
TO_DATE('01/jan/2024', 'dd/month/yyyy'))

Insert into ERiferito values ('5333171100501000', '10000098720009110010',
TO_DATE('09/oct/2024', 'dd/month/yyyy'))

Insert into ERiferito values ('5333171100581000', '10000098720009110018',
TO_DATE('28/mar/2024', 'dd/month/yyyy'))

Insert into ERiferito values ('5333171120061000', '10000098720009110019',
TO_DATE('21/jul/2024', 'dd/month/yyyy'))

Insert into ERiferito values ('5333171100060000', '10000098720009110015',
TO_DATE('18/jun/2024', 'dd/month/yyyy'))



QUERY



LE QUERY SONO INTERROGAZIONI, IN LINGUAGGIO SQL, CHE PERMETTONO DI RICAVARE INFORMAZIONI DA UN DATABASE

TROVARE IL NUMERO DI
PASSAPORTO E IL CODICE
FISCALE DEI PASSEGGERI CHE
HANNO VIAGGIATO IN CLASSE
BUSINESS A TARIFFA INTERA IN
UN VOLO CON DURATA MINORE
DI 200 ORE

```
select N_Passaporto, P.CF  
from Biglietto B join  
Passeggero P on B.CF=P.CF  
join Volo V on  
B.Cod_V=V.Cod_V  
where  
B.Classe='J' and V.Durata<200
```



TROVARE IL NUMERO DI DOCUMENTI CHE HANNO LA STESSA DATA DI SCADENZA

```
select count(distinct Data_Scad) from Documento  
D1  
  
where exists(  
  
    select * from Documento D2  
  
        where D2.Data_Scad=D1.Data_Scad and  
D1.CF<>D2.CF  
  
)  
  
group by Data_Scad
```

TROVARE TUTTI I VOLI INTERNAZIONALI CHE PARTONO DA NAPOLI

```
select *  
  
from Volo V join Compagnia C on V.Iata=C.Iata  
  
where Cod_V=(select Cod_V from Volo  
  
        where Icao_P='LIRN' and  
Tipo_Volo='Internazionale')
```



TROVARE IL CODICE FISCALE DEI PASSEGGERI CHE HANNO COMPRATO IL BIGLIETTO PRIMA DEL 15 GIUGNO 2024 CON UNA SPESA MINORE DI 900 EURO

select CF

```
from ERiferito E join Pagamento P on E.N_Carta=P.N_Carta  
where Data_Pag<TO_DATE('15/jun/2024', 'dd-mon/yyyy') and  
Importo>=900
```

TROVARE IL COMPONENTE PIU' ANZIANO DEL PERSONALE DI BORDO

```
select Nome, Cognome, floor(MONTHS_BETWEEN(sysdate,  
DN)/12) as Età, Matricola, Ruolo from Viaggiatori V1  
join PersonaleBordo PB on V1.ID_Viag=PB.ID_Viag  
where not exists(select * from Viaggiatori V2 join  
PersonaleBordo PB on V2.ID_Viag=PB.ID_Viag  
where V2.DN<V1.DN)
```

TROVARE NOME, COGNOME, ETA', CODICE VOLO E CODICE BIGLIETTO DI COLORO CHE HANNO VIAGGIATO IN BUSINESS CLASS SU VOLI CHE PARTONO DA MADRID ORGANIZZATI DALLE COMPAGNIE

```
select Nome, Cognome, floor(MONTHS_BETWEEN(sysdate, DN)/12) as Età,  
VO.Cod_V, P.CF, Cod_B
```

```
from Viaggiatori V join Passeggero P on V.ID_Viag=P.ID_Viag
```

```
join Biglietto B on B.CF=P.CF
```

```
join Volo VO on VO.Cod_V=B.Cod_V
```

```
where B.Classe='J' and VO.Iata=
```

```
(select max(Iata) from Volo where Icao_P='LEMD')
```

TROVARE NOME, COGNOME ED ETA' DEI VIAGGIATORI CHE PRENDONO VOLI PER ARRIVARE A CATANIA E CHE HANNO COMPRATO IL BIGLIETTO PRIMA DEL 16 GIUGNO 2024

```
select Nome, Cognome, floor(MONTHS_BETWEEN(sysdate, DN)/12) as Età
```

```
from Viaggiatori V join Passeggero P on V.ID_Viag=P.ID_Viag
```

```
join Biglietto B on B.CF=P.CF
```

```
join Volo VO on B.Cod_V=VO.Cod_V
```

```
join ERiferito E on E.Cod_B=B.Cod_B
```

```
where Icao_A='LICC' and Data_Pag>TO_DATE('16/jun/2024', 'dd/month/yyyy')
```



CONTARE PER OGNI BAGAGLIO QUANTI HANNO LO STESSO PESO E
VISUALIZZARNE IL PESO E IL TIPO

```
select count (Peso_B), Peso_B, Tipo  
from Bagaglio B1  
where exists(  
    select * from Bagaglio B2  
    where B2.Peso_B=B1.Peso_B and B1.Tag<>B2.Tag  
)  
group by Peso_B, Tipo
```

TROVARE LA CITTA' E IL RELATIVO NUMERO PISTE DELL'AEROPORTO I CUI VOLI SONO ORGANIZZATI DA RYANAIR E IL PERSONALE DI BORDO È CARATTERIZZATO DA PILOTI DI LINEA

```
select Citta, Num_Piste  
from Aeroporto join Volo V on Icao_A=Icao  
join PersonaleBordo PB on PB.Cod_V=V.Cod_V  
join Licenza L on L.Matricola=PB.Matricola  
where V.Iata='RYR' and Ruolo='Pilota'  
and Tipo_L=(  
    select max(Tipo_L) from Licenza  
    where Tipo_L='Pilota di linea'  
)
```



TROVARE NUMERO POSTO E LA DATA DEL VOLO SULLE CARTE D'IMBARCO DEI PASSEGGERI CHE PRENDONO UN VOLO PER BARCELLONA PAGANDO MENO DI 900 EURO ED IDENTIFICARLI ATTRAVERSO IL CODICE FISCALE E TROVARE LA CLASSE IN CUI VIAGGIANO

```
select N_Posto, Data_V, C.CF, Classe  
from Cartalmbarco C join Volo V on  
C.Cod_V=V.Cod_V  
join Passeggero P on C.CF=P.CF  
join Pagamento PA on PA.CF=P.CF  
join Biglietto B on B.CF=P.CF  
where Icao_A='LEBL' and Importo<900
```





TRIGGER

I TRIGGER SONO UTILIZZATI PER IL CONTROLLO
DEI VINCOLI DINAMICI E AGISCONO SU
INSERIMENTO E MODIFICA DEI DATI

PERSONALE



IL TRIGGER EVITA L'INSERIMENTO DI PERSONALE DI BORDO CON
ETA' INFERIORE AI 18 ANNI

create or replace trigger Personale

before insert on PersonaleBordo

for each row

declare

Eta number(2,0);

Minorenne exception;

begin

select floor(MONTHS_BETWEEN(sysdate, DN)/12) into Eta

from Viaggiatori

where ID_Viag=:new.ID_Viag;

if Eta<18 then

raise Minorenne;

end if;

exception

when Minorenne then

RAISE_APPLICATION_ERROR(-20001,'Personale minorenne');

end;

DOCUMENTO

IL TRIGGER NON PERMETTE L'INSERIMENTO DI DOCUMENTI CHE HANNO SCADENZA PRECEDENTE ALLA DATA ODIENA

```
create or replace trigger Documento
```

```
before insert or update on Documento
```

```
for each row
```

```
declare
```

```
Scaduto exception;
```

```
begin
```

```
if :new.Data_Scad<SYSDATE then
```

```
    raise Scaduto;
```

```
end if;
```

```
exception
```

```
when NO_DATA_FOUND then
```

```
null;
```

```
when Scaduto then
```

```
    RAISE_APPLICATION_ERROR(-20002, 'Documento scaduto');
```

```
end;
```



VIAGGIO



IL TRIGGER IMPEDISCE L'ESISTENZA DI VOLI IL CUI IMPORTO DI PAGAMENTO E' INFERIORE A 50

create or replace trigger Viaggio

before insert on Pagamento

for each row

declare

PrezzoBasso exception;

begin

if :new.Importo<50 then

raise PrezzoBasso;

end if;

exception

when NO_DATA_FOUND then

null;

when PrezzoBasso then

RAISE_APPLICATION_ERROR(-20003, 'Prezzo basso');

end;

CARTA



IL TRIGGER IMPedisce l'INSERIMENTO DI CARTE DI CREDITO CON DATA DI SCADENZA PRECEDENTE ALLA DATA ODIERNA

```
create or replace trigger Carta
before insert or update on Pagamento
for each row
declare
Carta_Scaduta exception;
begin
if :new.Data_Scad_Cart< SYSDATE then
    raise Carta_Scaduta;
end if;
exception
when NO_DATA_FOUND then
null;
when Carta_Scaduta then
    RAISE_APPLICATION_ERROR(-20004, 'Carta scaduta');
end;
```

LAVORATORI

IL TRIGGER NON PERMETTE L'INSERIMENTO DI PIU' DI 4 LAVORATORI PER OGNI VOLO

create or replace trigger Lavoratori

before insert on PersonaleBordo

for each row

declare

Contatore number(1,0);

Troppi_Lavoratori exception;

begin

select count(*) into Contatore

from PersonaleBordo

where Cod_V=:new.Cod_V;

if Contatore>4 then

raise Troppi_Lavoratori;

end if;

exception

when Troppi_Lavoratori then

RAISE_APPLICATION_ERROR(-20005, 'Ci sono troppi lavoratori');

end;



VENDITA

IL TRIGGER IMPEDISCE LA VENDITA DI PIU' DI 99 BIGLIETTI PER LO STESSO VOLO

create or replace trigger Vendita

before insert on Biglietto

for each row

declare

Biglietti number(3,0);

Tutto_Esaurito exception;

begin

select count(*) into Biglietti

from Biglietto

where Cod_V=:new.Cod_V;

if Biglietti>99 then

raise Tutto_Esaurito;

end if;

exception

when Tutto_Esaurito then

RAISE_APPLICATION_ERROR(-20007, 'Biglietti finiti');

end;



BAGAGLI_A_PERSONA

IL TRIGGER LIMITA IL NUMERO DI BAGAGLI A PERSONA A 2

```
create or replace trigger Bagagli_A_Persona
```

```
before insert on HaBagaglio
```

```
for each row
```

```
declare
```

```
Bagagli number(1,0);
```

```
Limite exception;
```

```
begin
```

```
select count(*) into Bagagli
```

```
from HaBagaglio
```

```
where CF=:new.CF;
```

```
if Bagagli>2 then
```

```
raise Limite;
```

```
end if;
```

```
exception
```

```
when Limite then
```

```
RAISE_APPLICATION_ERROR(-20008, 'Il bagaglio rimane qua');
```

```
end;
```



VOLI_INSIEME



IL TRIGGER EVITA CHE DUE AEREI PARTANO NELLO STESSO ISTANTE

create or replace trigger Voli_Insieme

before insert or update on Volo

for each row

declare

Partenza Volo.Cod_V%type;

NonPartire exception;

begin

select Cod_V into Partenza

from Volo

where Ora_Part_Eff=:new.Ora_Part_Eff

and rownum=1;

raise NonPartire;

exception

when NO_DATA_FOUND then

null;

when NonPartire then

RAISE_APPLICATION_ERROR(-20009, 'Aspetta e poi vai');

end;

POSTOMIO



IL TRIGGER EVITA L'ASSEGNAZIONE DELLO STESSO POSTO SULLO
STESO AEREO A DUE PASSEGGERI DIVERSI

```
create or replace trigger PostoMio
before insert or update on Cartalmbarco
for each row
declare
Posto Cartalmbarco.N_Posto%type;
NonPermetterti exception;
begin
select N_Posto into Posto
from Cartalmbarco
where Cod_V=:new.Cod_V;
if Posto=:new.N_Posto then
raise NonPermetterti;
end if;
exception
when NO_DATA_FOUND then
null;
when NonPermetterti then
RAISE_APPLICATION_ERROR(-20010, 'Questo è il mio posto');
end;
```

ACQUISTO

IL TRIGGER EVITA CHE LO STESSO BIGLIETTO SIA COMPRATO DA DUE PERSONE NELLO STESSO ISTANTE

```
create or replace trigger Acquisto
```

```
before insert on ERiferito
```

```
for each row
```

```
declare
```

```
DataAcquisto date;
```

```
Mio exception;
```

```
begin
```

```
select Data_Pag into DataAcquisto
```

```
from ERiferito
```

```
where Cod_B=:new.Cod_B;
```

```
if DataAcquisto=:new.Data_Pag then
```

```
raise Mio;
```

```
end if;
```

```
exception
```

```
when Mio then
```

```
RAISE_APPLICATION_ERROR(-20011, 'Comprato prima io');
```

```
end;
```





PROCEDURE E FUNZIONI

LE PROCEDURE E LE FUNZIONI SEMPLIFICANO LE OPERAZIONI DEGLI UTENTI E SONO IMPLEMENTABILI QUANDO CI SONO PIU' CONTROLLI DA EFFETTUARE. LE PROCEDURE SONO MEMORIZZATE PERMANENTEMENTE NEL DATABASE E NON HANNO NECESSARIAMENTE UN VALORE DI OUTPUT. LE FUNZIONI, INVECE, HANNO NECESSARIAMENTE UN VALORE DI OUTPUT

AGGIORNAMENTO VOLI

```
create or replace procedure AggiornamentoVoli (
    lataN Compagnia.lata%type,
    Cod_VN Volo.Cod_V%type,
    Ora_Part_EffN date,
    Ora_Arr_EffN date,
    Icao_PN Volo.Icao_P%type,
    Icao_AN Volo.Icao_A%type,
    Marca_ImmN Aereo.Marca_Imm%type
)
as
    Peso_Aereo number(4,0);
    N_Posti_Occupati number(3,0);
    NoAereo exception;
begin
    --Aggiornamento ora effettiva del volo
    update Volo
```

```
        set Ora_Part_Eff = Ora_Part_EffN,
            Ora_Arr_Eff = Ora_Arr_EffN,
            Icao_P = Icao_PN,
            Icao_A = Icao_AN,
            Marca_Imm = Marca_ImmN
        where lata = lataN and Cod_V = Cod_VN;

        --Calcolo posti occupati
        select count (N_Posto) into N_Posti_Occupati
        from Cartalbarco
        where Cod_V=Cod_VN;

        -- Aggiorna il numero di posti disponibili
        -- nell'aereo
        update Aereo
        set N_Posti = N_Posti - N_Posti_Occupati
        where Marca_Imm = Marca_ImmN;
```

QUESTA PROCEDURA AGGIORNA L'ORA
EFFETTIVA DEL VOLO, AGGIORNA I POSTI
DISPONIBILI SULL'AEREO E IMPEDISCE
L'INSERIMENTO DI AEREI CON PESO
INFERIORE A 10000

```
--Il peso dell'aereo non deve essere minore di
10000
select Peso into Peso_Aereo
from Aereo
where Marca_Imm = Marca_ImmN;
if Peso_Aereo<10000 then
    raise NoAereo;
end if;
commit;
exception
when NoAereo then
    RAISE_APPLICATION_ERROR(-20020, 'Peso
aereo non valido');
rollback;
end;
```

GESTIONEA

QUESTA FUNCTION RESTITUISCE LA MATRICOLA DEL
COMPONENTE DEL PERSONALE DI BORDO CON ETA'
INFERIORE PER POTER SOSTituIRE CHI HA MENO ESPERIENZA

create or replace function GestioneA

return PersonaleBordo.Matricola%type

as

PiuPiccolo PersonaleBordo.ID_Viag%type;

PiuPiccoloM PersonaleBordo.Matricola%type;

PiuPiccoloR PersonaleBordo.Ruolo%type;

Dura number(4,0):=0;

Negativo exception;

begin

select PB.ID_Viag into PiuPiccolo

from Viaggiatori V join PersonaleBordo PB on V.ID_Viag=PB.ID_Viag

where V.DN= (select min(DN) from Viaggiatori);

select PB.Matricola into PiuPiccoloM

from Viaggiatori V join PersonaleBordo PB on V.ID_Viag=PB.ID_Viag

where V.DN= (select min(DN) from Viaggiatori);

return PiupiccoloM;

select Ruolo into PiuPiccoloR

from Viaggiatori V join PersonaleBordo PB on V.ID_Viag=PB.ID_Viag

where V.DN= (select min(DN) from Viaggiatori);

select Durata into Dura

from PersonaleBordo PB join Viaggiatori V on V.ID_Viag=PB.ID_Viag join
Volo VO on PB.Cod_V=VO.Cod_V

where V.DN= (select min(DN) from Viaggiatori);

if Dura<0 then

raise Negativo;

end if;

exception

when Negativo then

RAISE_APPLICATION_ERROR(-20022, 'Impossibile');

end;

DURATA

QUESTA FUNCTION RESTITUISCE LA DURATA
EFFETTIVA DI CIASCUN VOLO

```
create or replace function Durata(Cod_VN Volo.Cod_V%type, lataN  
Volo.lata%type)
```

```
    return number
```

```
as
```

```
    durataEff number(5,0);
```

```
CFn Passeggero.CF%type;
```

```
Marca_Imm Aereo.Marc_Imm%type;
```

```
begin
```

```
select (Ora_Arr_Eff-Ora_Part_Eff)*24*60 into durataEff
```

```
from Volo
```

```
where lata=lataN and Cod_V=Cod_VN;
```

```
return durataEff;
```

```
select listagg(CF,',') within group (order by CF) into CFn  
from Cartalmbarco
```

```
        where lata=lataN and Cod_V=Cod_VN;
```

```
select A.Marc_Imm into Marca_Imm  
from Aereo A join Volo V on A.Marc_Imm=V.Marc_Imm  
where A.Marc_Imm=(select Marca_Imm from Aereo  
where lata=lataN and Cod_V=Cod_VN);
```

```
exception
```

```
when NO_DATA_FOUND then  
return 'Nessun dato per questo volo';  
end;
```

CLASSE

QUESTA PROCEDURA, IN BASE AL VALORE SPECIFICATO NELLA CHIAMATA, IDENTIFICA I VOLI PER CUI LA CLASSE E' BUSINESS, QUELLI NON BUSINESS E QUELLI PER CUI IL VOLO NON E' DEFINITO

```
create or replace procedure Classe(Importon
Pagamento.Importo%type)
as
imp Pagamento.Importo%type;
begin
select P.Importo into imp
from Pagamento P join Biglietto B on P.CF=B.CF
join Volo V on B.Cod_V=V.Cod_V
where P.Importo=Importon and Classe='J' and
V.Durata>100;
if imp<500 then
    DBMS_OUTPUT.PUT_LINE('Non è classe
business');
else
    DBMS_OUTPUT.PUT_LINE('Classe business');
end if;
commit;
exception
when NO_DATA_FOUND then
    DBMS_OUTPUT.PUT_LINE('Non esiste');
rollback;
end;
```

VRYANAIR

QUESTA FUNCTION TROVA I CODICI VOLO
RELATIVI AL TIPO DI LICENZA SPECIFICATO

```
create or replace function VRyanair (Tipo_Ln
Licenza.Tipo_L%type, Cod_Vn
Volo.Cod_V%type)
return varchar
as
cod Volo.Cod_V%type;
begin
select max(V.Cod_V) into cod
from Compagnia C join Volo V on C.lata=V.lata
join PersonaleBordo PB on
```

```
PB.Cod_V=V.Cod_V
join Licenza L on L.Matricola=PB.Matricola
where Tipo_L=Tipo_Ln and V.Cod_V=Cod_Vn;
return cod;
exception
when NO_DATA_FOUND then
DBMS_OUTPUT.PUT_LINE('Voli non trovati');
end;
```

ECONOMY

```
create or replace procedure Economy(Importoe
Pagamento.Importo%type)
as
importo Pagamento.Importo%type;
begin
select P.Importo into importo
from Pagamento P join Biglietto B on P.CF=B.CF join
Volo V on B.Cod_V=V.Cod_V
where P.Importo=Importoe and Classe='F' and
V.Tipo_Volo='Internazionale';
if importo>800 then
DBMS_OUTPUT.PUT_LINE('Non è classe economy');
else
```

QUESTA PROCEDURA , IN BASE AL VALORE SPECIFICATO
NELLA CHIAMATA, IDENTIFICA I VOLI PER CUI LA CLASSE E'
ECONOMY, QUELLI NON ECONOMY E QUELLI PER CUI IL VOLO
NON E' DEFINITO

```
DBMS_OUTPUT.PUT_LINE('Classe economy');
end if;
commit;
exception
when NO_DATA_FOUND then
DBMS_OUTPUT.PUT_LINE('Non esiste');
rollback;
end;
```

VISTE

UNA VISTA E' UN ALIAS PER UNA QUERY. IN GENERALE, UNA VISTA E' UNA TABELLA VIRTUALE NON SALVATA SU DISCO (NON CREA RIDONDANZA) E VISIBILE AD UN UTENTE PER NON PERMETTERE L'ACCESSO ALLE TABELLE MA A DATI LIMITATI

VOLIPILOTI

QUESTA VISTA E' UTILE ALL'UTENTE PILOTA E PERMETTE DI VISUALIZZARE TUTTI I VOLI CHE HANNO GIA' UN PILOTA O UN CO-PILOTA ASSEGNAUTO

create or replace view VoliPiloti as

select VRyanair(Tipo_L, V.Cod_V) as VP,

Marca_Imm as Aereo,

Nome_Comp as Compagnia,

Tipo_L as L,

Ruolo as R

from Compagnia C join Volo V on C.lata=V.lata

join PersonaleBordo PB on PB.Cod_V=V.Cod_V

join Licenza L on L.Matricola=PB.Matricola

order by VP



DURATAVOLI

QUESTA VISTA E' UTILE AL PASSEGGERO PER CONOSCERE TUTTO CIO' CHE E' NECESSARIO RELATIVAMENTE AL PROPRIO VOLO ED E' UTILE AL DIRETTORE DELL'AEROPORTO PER AVERE I VOLI SOTTO CONTROLLO

create or replace view DurataVoli as

select distinct Durata(V.Cod_V, V.lata) as DV,

V.Marca_Imm as AereoVolo,

V.Cod_V as CV,

Data_V as DataVolo,

V.Icao_P as Partenza,

V.Icao_A as Arrivo

from Cartalbarco C join Volo V on V.Cod_V=C.Cod_V

join Aeroporto A on A.Icao=V.Icao_P

order by DV



SOSTITUZIONE

QUESTA VISTA E' UTILE AL DIRETTORE
DELL'AEROPORTO PER POTER EFFETTUARE
UNA SOSTITUZIONE CON IL COMPONENTE
DEL PERSONALE DI BORDO CHE HA MENO
ESPERIENZA

create or replace view Sostituzione as

select distinct GestioneA as Giovane

from PersonaleBordo





COMPAGNIE

QUESTA VISTA E' UTILE AL DIRETTORE
DELL'AEROPORTO PER CONOSCERE LE COMPAGNIE

create or replace view Compagnie as

select C.lata,

Nome_Comp as Compagnia,

Pec_C as Pec,

Tel1_C as Telefono_Principale,

Tel2_C as Telefono_Secondario

from Compagnia C join Contatti_Compagnia CC on
C.lata=CC.lata

order by lata

AEROPORTI

QUESTA VISTA E' UTILE AL DIRETTORE DELLA
COMPAGNIA PER TENERE SOTTO CONTROLLO
GLI AEROPORTI IN CUI OPERANO LE
COMPAGNIE

create or replace view Aeroporti as

select A.Icao,

Citta as Citta,

PEC_A as Pec,

Tel1_A as Telefono_Principale,

Tel2_A as Telefono_Secondario

from Aeroporto A join Contatti_Aeroporto CA on
A.Icao=CA.Icao

order by Icao





VENDITE

QUESTA VISTA E' UTILE AL GESTORE DEL SITO ONLINE DOVE VENGONO COMPRATI I BIGLIETTI PER I VOLI

```
create or replace view Vendite as  
select E.Cod_B as CodiceBiglietto,  
E.N_Carta as Carta,  
Data_Pag as Data_Pagamento,  
Importo as Importo_Pagato,  
PCF as Codice_Fiscale_Passeggero,  
Cod_V as Volo,  
Classe as Classe_Volo  
from ERiferito E join Pagamento P on E.N_Carta=P.N_Carta  
join Biglietto B on E.Cod_B=B.Cod_B  
order by Importo_Pagato
```

QUESTA VISTA E' UTILE
ALL'ADDETTO AL
CONTROLLO PER AVERE
LE INFORMAZIONI
RIGUARDANTI IL CHECK-
IN DEI PASSEGGERI

CONTROLLO

create or replace view Controllo as
select B.Cod_B as CodiceBiglietto,
Numero as Carta_Imbarco,
Data_V as Data_Volo,
N_Passaporto as Numero_Passaporto,
C.CF as Codice_Fiscale_Passeggero,
N_Doc as Numero_Documento,
Data_Scad as Scadenza_Documento,
BL.Tag as Tag_Bagaglio,
Peso_B as Peso_Controllato,
C.Cod_V as Codice_Volo

from Cartalmbarco C join Passeggero
P on C.CF=P.CF
join Biglietto B on P.CF=B.CF
join Documento D on P.CF=D.CF
join HaBagaglio HB on HB.CF=P.CF
join Bagaglio BL on BL.Tag=HB.Tag
order by Peso_Controllato





AEREI

QUESTA VISTA E' UTILE AL
DIRETTORE DELL'AEROPORTO
PER TENERE TRACCIA DEGLI
AEREI

create or replace view Aerei as
select Peso as Peso_Aereo,
A.Marcia_Imm as Targa,
N_Posti as Posti,
Cod_V as Codice_Volo
From Aereo A join Volo V on
A.Marcia_Imm=V.Marcia_Imm
Order by Peso



DATA CONTROL LANGUAGE

IL DATA CONTROL LANGUAGE DEFINISCE I PERMESSI PER OGNI UTENTI SIA PER QUANTO RIGUARDA LE PROCEDURE DA ESEGUIRE CHE LE VISTE A CUI POTER ACCEDERE

DIRETTORE_AEROPORTO



IL DIRETTORE DELL'AEROPORTO PUO' ACCEDERE ALLE VISTE
CHE RIGUARDANO IL TRACCIAMENTO DEGLI AEREI, LE
COMPAGNIE PRESENTI, LE INFORMAZIONI SUI VOLI E LE
INFORMAZIONI RIGUARDO POSSIBILI SOSTITUZIONI. INOLTRE,
PUO' ESEGUIRE LA PROCEDURA DI AGGIORNAMENTO DEI VOLI

GRANT CONNECT, CREATE SESSION TO
DIRETTORE_AEROPORTO;

GRANT EXECUTE ON AGGIORNAMENTOVOLI TO
DIRETTORE_AEROPORTO;

GRANT SELECT ON AEREI TO DIRETTORE_AEROPORTO;

GRANT SELECT ON DURATAVOLI TO DIRETTORE_AEROPORTO;

GRANT SELECT ON COMPAGNIE TO DIRETTORE_AEROPORTO;

GRANT SELECT ON SOSTITUZIONE TO
DIRETTORE_AEROPORTO;

GESTORE_SITO

IL GESTORE DEL SITO ONLINE CHE VENDE BIGLIETTI PUO'
ACCEDERE ALLA VISTA VENDITE PER TENERE TRACCIA
DEGLI ACQUISTI E PUO' ESEGUIRE LE PROCEDURE
CLASSE E ECONOMY IN MODO DA CONTROLLARE SE CI
SONO STATE VENDITE A PREZZI ANOMALI

GRANT CONNECT, CREATE SESSION TO GESTORE_SITO;
GRANT SELECT ON VENDITE TO GESTORE_SITO;
GRANT EXECUTE ON CLASSE TO GESTORE_SITO;
GRANT EXECUTE ON ECONOMY TO GESTORE_SITO;



DIRETTORE_COMPAGNIA

IL DIRETTORE DELLA COMPAGNIA PUO'
ACCEDERE ALLA VISTA PER CONOSCERE
GLI AEROPORTI IN CUI OPERA LA PROPRIA
COMPAGNIA MA NON PUO' ESEGUIRE
ALCUNA PROCEDURA

GRANT CONNECT, CREATE SESSION TO
DIRETTORE_COMPAGNIA;

GRANT SELECT ON AEROPORTI TO
DIRETTORE_COMPAGNIA;





PASSEGGERO

IL PASSEGGERO PUO' ACCEDERE
ALLA VISTA PER CONOSCERE LE
INFORMAZIONI RELATIVE AI VOLI E
PUO' ESEGUIRE LE PROCEDURE
PER SAPERE LA CLASSE RELATIVA
AL PROPRIO VOLO

GRANT CONNECT, CREATE
SESSION TO PASSEGGERO;

GRANT SELECT ON DURATAVOLI TO
PASSEGGERO;

GRANT EXECUTE ON CLASSE TO
PASSEGGERO;

GRANT EXECUTE ON ECONOMY TO
PASSEGGERO;



PILOTA

IL PILOTA PUO' ACCEDERE ALLA VISTA
PER VISUALIZZARE I VOLI A CUI GIA' E'
STATO ASSEGNATO UN PILOTA O UN
CO-PILOTA MA NON PUO' ESEGUIRE
PROCEDURE

GRANT CONNECT, CREATE SESSION TO
PILOTA;

GRANT SELECT ON VOLIPILOTI TO
PILOTA;



ADDETTO_CONTROLLI

L'ADDETTO AI CONTROLLI PUO' ACCEDERE ALLA VISTA
CONTROLLO PER AVERE LE INFORMAZIONI NECESSARIE AL
CHECK-IN DEI PASSEGGERI MA NON PUO' ESEGUIRE PROCEDURE

GRANT CONNECT, CREATE SESSION TO ADDETTO_CONTROLLI;
GRANT SELECT ON CONTROLLO TO ADDETTO_CONTROLLI;