*FLIGHTS*

*MANAGEMENT*

*SYSTEM*

**PROGETTO SVILUPPATO PER IL CORSO DI**

**BASI DI DATI**

**CORSO DI LAUREA IN**

**SCIENZE E TECNOLOGIE INFORMATICHE**

**ANNA ACCADEMICO 2012/2013**

Riccardo Guglielmo, matr. 577079

Lavinia Mosut, matr. 577405

**STRUTTURA DELL’ELABORATO**

**1 - Introduzione**

**2 - Analisi dei requisiti**

2.1 - Intervista e definizione delle specifiche in linguaggio naturale

2.2 - Rilevamento delle ambiguità e correzioni proposte

2.3 - Estrazione dei concetti principali

**3 - Progettazione Concettuale**

3.1 - Scelta della strategia di progetto

3.2 - Schema scheletro

3.3 - Raffinamenti proposti

3.4 - Schemi concettuali parziali e integrazione delle viste

3.5 - Schema concettuale finale

**4 - Progettazione logica**

4.1 - Stima del volume dei dati

4.2 - Descrizione delle operazioni principali e stima della loro frequenza

4.3 - Schemi di navigazione e tabelle degli accessi

4.4 - Raffinamento dello schema (eliminazione di identificatori esterni, attributi composti e gerarchie, scelta delle chiavi) e analisi delle ridondanze

4.5 - Traduzione di entità e associazioni in relazioni

4.6 - Schema relazionale finale

4.7 - Traduzione delle operazioni in query SQL

**5 - Progettazione dell'applicazione**

5.1 - Descrizione dell'architettura dell'applicazione realizzata

**1 - INTRODUZIONE**

Il progetto “**Flight Management System**” consiste nello sviluppo di un portale Web, realizzato in HTML5, CSS3 e JSP, che invia query SQL a un server DB.

Il sito è stato realizzato con particolare attenzione ai suggerimenti offerti dalle implementazioni reali, cercando di rispecchiare, laddove sensato, le potenzialità e le possibilità offerte agli utenti dai servizi web di aeroporti di grandi città e/o grandi compagnie aeree.

La realizzazione del sito è stata effettuata utilizzando NetBeans, sia per le pagine JSP che per le poche pagine di HTML statico presenti e anche per la gestione delle query SQL (il tutto testato col browser “Google Chrome”).

Il sito è supportato da un unico database realizzato con Access, i cui dettagli saranno esaminati più avanti.

**2 – ANALISI DEI REQUISITI**

**2.1 – Intervista e definizione delle specifiche in linguaggio naturale**

La seguente descrizione riporta in linguaggio naturale i requisiti per il nostro sistema informativo:

**Flight Management System**

Si vuole realizzare un portale web che comunica con una base di dati, per:

- la gestione dei voli di un aeroporto, da parte dell’amministrazione di esso;

- la prenotazione dei voli, da parte degli utenti che utilizzeranno il servizio.

Nel database bisognerà tenere traccia:

* dei **voli** (numero volo, data del volo, località di partenza, località di destinazione, orario di partenza, orario di arrivo previsto, compagnia aerea che offre il volo e recapiti di essa, prezzo);
* delle **tariffe** dei voli (sovrapprezzo per viaggio in prima classe, sovrapprezzo per viaggio in seconda classe, sconto per minori, prezzo per bagaglio);
* dei **passeggeri**, ovvero degli utenti che effettuano una prenotazione (cognome, nome, codice fiscale, numero della id card e data di nascita);
* delle **prenotazioni** (numero della prenotazione, classe sulla quale si desidera viaggiare, numero di passeggieri, numero di minori, numero di bagagli, prezzo totale della prenotazione);
* dei **biglietti** delle prenotazioni (numero del biglietto);
* degli **amministratori** del sistema (cognome, nome, email, password);

**2.2 – Rilevamento delle ambiguità e correzioni proposte**

Cerchiamo ora all’interno delle specifiche fornite le parole o le espressioni che costituiscono elementi di ambiguità per il progettista del sistema; seguirà una tabella riassuntiva contenente tutte le espressioni denotate e i rispettivi chiarimenti su quest’ultime:

**Flight Management System**

Si vuole realizzare un portale web che comunica con una base di dati, per:

- la gestione dei voli di un aeroporto, da parte dell’amministrazione di esso;

- la prenotazione dei voli, da parte degli utenti che utilizzeranno il servizio.

Nel database bisognerà tenere traccia:

* dei voli (numero volo, data del volo, **località di partenza**, **località di destinazione₁**, orario di partenza, orario di arrivo previsto, **compagnia aerea₂** che offre il volo e **recapiti₃** di essa, prezzo);
* delle tariffe dei voli (sovrapprezzo per viaggio in prima classe, sovrapprezzo per viaggio in seconda classe, sconto per minori, prezzo per bagaglio);
* dei passeggeri, ovvero degli utenti che effettuano una prenotazione (cognome, nome, codice fiscale, numero della id card e data di nascita);
* delle prenotazioni (numero della prenotazione, classe sulla quale si desidera viaggiare, numero di passeggieri, numero di minori, numero di bagagli, **prezzo totale della prenotazione₄**);
* dei **biglietti delle prenotazioni₅** (numero del biglietto);
* degli **amministratori del sistema₆** (cognome, nome, email, password);

**Chiarimenti:**

|  |  |  |
| --- | --- | --- |
| **Numero di nota** | **Espressione** | **Chiarimenti e scelte di sviluppo** |
| 1 | Località di partenza e località di destinazione | I voli partono da una località e giungono in un’altra. Pertanto sarebbe furbo proporre le due località, come relazioni “partenza” e “destinazione” che legano l’entità “VOLO” all’entità “LOCALITA”. Quest’ultima presenterà i seguenti attributi: nome, nazionalità, cap. |
| 2 | Compagnia aerea | Anche la compagnia aerea, come le località, dovrebbe essere registrata come entità in relazione con l’entità “VOLO”. |
| 3 | recapiti | I recapiti della compagnia aerea, saranno gli attributi  “indirizzo email” e “numero di telefono” dell’entità “COMPAGNIA\_AEREA”. |
| 5 | prezzo totale della prenotazione | Il prezzo totale della prenotazione verrà calcolato basandosi sul prezzo del volo che si desidera prenotare e sulla tariffa di esso. |
| 5 | biglietti delle prenotazioni | Ogni passeggero di ogni prenotazione, avrà un biglietto diverso. Pertanto un biglietto sarà di un solo passeggero e riguarderà una sola prenotazione (di un solo volo). |
| 6 | amministratori del sistema | Dovranno registrarsi sul portale prima di poter effettuare il login e gestire il sistema. |

Riportiamo di seguito il testo originale apportando le modifiche riassunte nella precedente tabella.

Si noti come il testo risulti ora meno elegante e spesso ripetitivo, ma chiarisca con efficacia tutti gli elementi di ambiguità introdotti dall’uso del linguaggio corrente nella stesura delle specifiche, risultando decisamente più leggibile dal punto di vista del progettista:

**Flight Management System**

Si vuole realizzare un portale web che comunica con una base di dati, per:

- la gestione dei voli di un aeroporto, da parte dell’amministrazione di esso;

- la prenotazione dei voli, da parte degli utenti che utilizzeranno il servizio.

Nel database bisognerà tenere traccia:

* dei voli (numero volo, data del volo, località di partenza, località di destinazione, orario di partenza, orario di arrivo previsto, compagnia aerea che offre il volo, prezzo);
* delle località (nome, nazionalità, cap);
* delle compagnie aeree che offrono i voli (nome della compagnia, indirizzo email, numero di telefono);
* delle tariffe dei voli (sovrapprezzo per viaggio in prima classe, sovrapprezzo per viaggio in seconda classe, sconto per minori, prezzo per bagaglio);
* dei passeggeri, ovvero degli utenti che effettuano una prenotazione (cognome, nome, codice fiscale, numero della id card e data di nascita);
* delle prenotazioni (numero della prenotazione, classe sulla quale si desidera viaggiare, numero di passeggieri, numero di minori, numero di bagagli, prezzo totale della prenotazione);
* dei biglietti delle prenotazioni (numero del biglietto);
* degli amministratori del sistema (cognome, nome, email, password);

**2.4 – Estrazione dei concetti principali**

Individuiamo adesso le parole e le espressioni chiave che ci consentiranno di realizzare un primo schema significativo del progetto e di raffinarlo successivamente per ottenere lo schema definitivo. I termini di rilievo appaiono nel testo con una sottolineatura:

**Flight Management System**

Si vuole realizzare un portale web che comunica con una base di dati, per:

- la gestione dei voli di un aeroporto, da parte dell’amministrazione di esso;

- la prenotazione dei voli, da parte degli utenti che utilizzeranno il servizio.

Nel database bisognerà tenere traccia:

* dei **voli** (numero volo, data del volo, località di partenza, località di destinazione, orario di partenza, orario di arrivo previsto, compagnia aerea che offre il volo, prezzo);
* delle **località** (nome, nazionalità, cap);
* delle **compagnie aeree** che offrono i voli (nome della compagnia, indirizzo email, numero di telefono);
* delle **tariffe** dei voli (sovrapprezzo per viaggio in prima classe, sovrapprezzo per viaggio in seconda classe, sconto per minori, prezzo per bagaglio);
* dei **passeggeri**, ovvero degli utenti che effettuano una prenotazione (cognome, nome, codice fiscale, numero della id card e data di nascita);
* delle **prenotazioni** (numero della prenotazione, classe sulla quale si desidera viaggiare, numero di passeggieri, numero di minori, numero di bagagli, prezzo totale della prenotazione);
* dei **biglietti** delle prenotazioni (numero del biglietto);
* degli **amministratori** del sistema (cognome, nome, email, password);

**3 – PROGETTAZIONE CONCETTUALE**

**3.1 - Scelta della strategia di progetto**

Lo sviluppo dello schema Entity-Relationship procederà ora per fasi successive più o meno indipendenti tra loro. Lo schema scheletro verrà suddiviso in più sezioni che saranno raffinate e arricchite fino a rappresentare per intero la realtà in esame e tutti i vincoli che la caratterizzano.

**3.2 - Schema scheletro**

Nello schema scheletro sono state individuate 9 entità:

****

**3.3 - Raffinamenti proposti**

Lo schema concettuale nella sua versione finale si avvarrà delle seguenti entità e associazioni (per ciascuna è fornita una breve descrizione):

|  |  |  |
| --- | --- | --- |
| **NOME** | **TIPO** | **DESCRIZIONE** |
| PERSONA | E | AMMINISTRATORE e/o PASSEGGERO, ovvero utilizzatori del portale web |
| AMMINISTRATORE | E | PERSONA che amministra/gestisce il sistema |
| PASSEGGERO | E | PERSONA che effettua una PRENOTAZIONE |
| ACQUISTA | R | Relazione fra PASSEGGERO e BIGLIETTO |
| BIGLIETTO | E | Rappresenta il Biglietto di un PASSEGGERO di una PRENOTAZIONE |
| RIGUARDA | R | Relazione fra BIGLIETTO e PRENOTAZIONE |
| PRENOTAZIONE | E | Prenotazione di un VOLO, effettuata da un PASSEGGERO |
| PRENOTA | R | Relazione fra PRENOTAZIONE e VOLO |
| VOLO | E | Rappresenta un volo |
| PREVEDE | R | Relazione che lega un VOLO alla corrispettiva TARIFFA |
| TARIFFA | E | Sovrapprezzi da pagare per un determinato VOLO |
| OFFRE | R | Relazione che lega una COMPAGNIA\_AEREA al VOLO offerto da essa |
| COMPAGNIA\_AEREA | E | Rappresenta una compagnia aerea |
| PARTENZA | R | Relazione che lega la LOCALITA di partenza di un VOLO, ad esso |
| DESTINAZIONE | R | Relazione che lega la LOCALITA di destinazione di un VOLO, ad esso |
| LOCALITA | E | Rappresenta la località dalla/nella quale parte/arriva un VOLO |

**3.4 - Schemi concettuali parziali e integrazione delle viste**

* **PERSONA, AMMINISTRATORE, PASSEGGERO:**

E’ stato scelto un approccio gerarchico in quanto sia gli amministratori che i passeggeri, sono delle persone e quindi hanno un nome e un cognome. I restanti attributi caratterizzano il tipo di PERSONA, per l’appunto, AMMINISTRATORE e/o PASSEGGERO.

Ovviamente un amministratore può prenotare un volo e quindi essere il passeggero di un volo e viceversa.

****

* **PASSEGGERO, BIGLIETTO, PRENOTAZIONE:**

Inizialmente si era deciso di legare l’entità PASSEGGERO direttamente all’entità PRENOTAZIONE, in quanto ogni passeggero può effettuare una prenotazione per più persone e quindi più passeggeri.

Una prenotazione poteva quindi riguardare più passeggeri e inoltre, ognuno di questi, poteva effettuare più prenotazioni. Pertanto vi era una associazione **molti a molti** che poi è stata eliminata inserendo l’entità BIGLIETTO (già prevista dal progetto).

Ogni PASSEGGERO, infatti, acquista **uno o più** BIGLIETTI che riguardano ognuno PRENOTAZIONI diverse, in quanto ogni BIGLIETTO riguarda **una ed una sola** PRENOTAZIONE.

Ad ogni PRENOTAZIONE, vengono associati **tanti** BIGLIETTI quanti sono i PASSEGGERI della prenotazione stessa. Pertanto, ogni BIGLIETTO di **una ed una sola** PRENOTAZIONE, viene acquistato da **un ed un solo** PASSEGGERO.

(Ogni passeggero, come è già stato detto, può acquistare più biglietti, cioè effettuare più prenotazioni).

****

* **PRENOTAZIONE, VOLO, TARIFFA:**

Ogni PRENOTAZIONE prenota **uno ed un solo** VOLO.

Ogni VOLO **può** essere prenotato tramite **una o più** PRENOTAZIONI.

Ogni VOLO prevede **una ed una sola** TARIFFA.

Ogni TARIFFA **può** essere prevista da **uno o più** VOLI.

****

* **VOLO, LOCALITA di partenza/destinazione**

Ogni VOLO parte da **una ed una** **sola** LOCALITA e né raggiunge **una ed una** **sola** altra.

Da/in una LOCALITA **possono** partire/arrivare **uno o più** VOLI.

****

**VOLO, COMPAGNIA\_AEREA:**

Ogni COMPAGNIA\_AEREA, offre **uno o più** VOLI.

Ogni VOLO è offerto da **una ed una sola** COMPAGNIA\_AEREA.

****

**3.5 - Schema concettuale finale**

****

**4 – PROGETTAZIONE LOGICA**

**4.1 - Stima del volume dei dati**

Le seguente tabella è stata costruita considerando che ogni giorno vengono inseriti in media 100 voli e che ognuno di essi abbia in media di 200 passeggeri. Inoltre il database manterrà tutte informazioni inserite in circa 2 anni.

|  |  |  |
| --- | --- | --- |
| **CONCETTO** | **COSTRUTTO** | **VOLUME** |
| AMMINISTRATORE | E | 200 |
| PASSEGGERO | E | 20000000 |
| ACQUISTA | R | 20000000 |
| BIGLIETTO | E | 20000000 |
| RIGUARDA | R | 20000000 |
| PRENOTAZIONE | E | 10000000 |
| PRENOTA | R | 10000000 |
| VOLO | E | 100000 |
| PREVEDE | R | 100000 |
| TARIFFA | E | 100 |
| OFFRE | R | 50 |
| COMPAGNIA\_AEREA | E | 50 |
| PARTENZA | R | 100 |
| DESTINAZIONE | R | 100 |
| LOCALITA | E | 100 |

**4.2 - Descrizione delle operazioni principali**

**e stima della loro frequenza**

Funzionalità principali richieste al sistema:

1) Visualizzazione voli in una certa data;

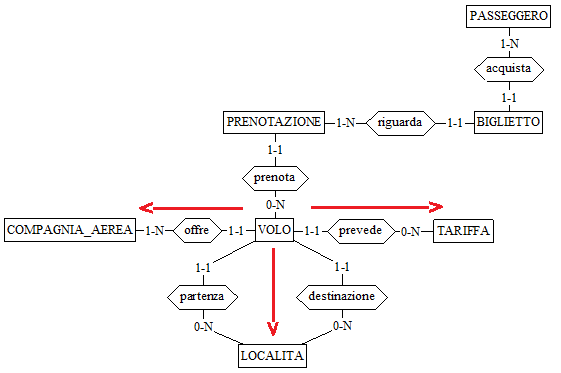
2) Prenotazione di un volo;

3) Visualizzazione delle prenotazioni di un determinato utente.

|  |  |
| --- | --- |
| **OPERAZIONE** | **FREQUENZA** |
| 1 | 100000/g |
| 2 | 20000/g |
| 3 | 20000/g |

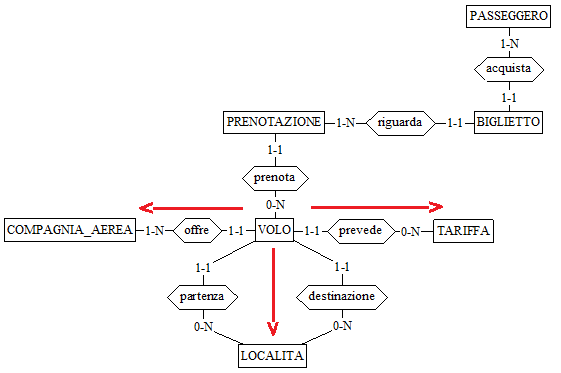
**4.3 - Schemi di navigazione e tabelle degli accessi**

**Operazione 1:**

****

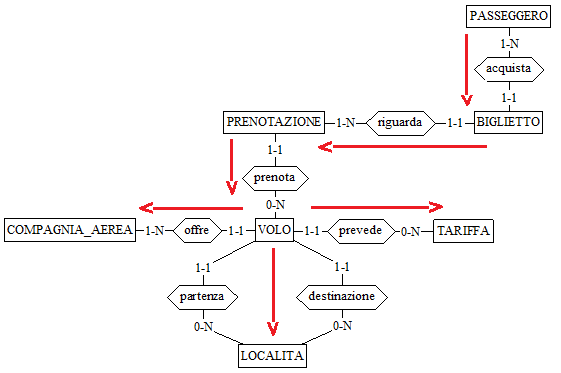
|  |  |  |  |
| --- | --- | --- | --- |
| **CONCETTO** | **COSTRUTTO** | **ACCESSI** | **TIPO** |
| VOLO | E | 100 | L |
| TARIFFA | E | 100 | L |
| COMPAGNIA\_AEREA | E | 50 | L |
| LOCALITA | E | 100 | L |

**Operazione 2:**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **CONCETTO** | **COSTRUTTO** | **ACCESSI** | **TIPO** |
| VOLO | E | 1 | L |
| TARIFFA | E | 1 | L |
| COMPAGNIA\_AEREA | E | 1 | L |
| LOCALITA | E | 2 | L |
| PRENOTAZIONE | E | 1 | S |
| BIGLIETTO | E | 2 | S |
| PASSEGGERO | E | 2 | S |

**Operazione 3:**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **CONCETTO** | **COSTRUTTO** | **ACCESSI** | **TIPO** |
| PASSEGGERO | E | 1 | L |
| BIGLIETTO | E | 1 | L |
| PRENOTAZIONE | E | 1 | L |
| VOLO | E | 1 | L |
| TARIFFA | E | 1 | L |
| COMPAGNIA\_AEREA | E | 1 | L |
| LOCALITA | E | 2 | L |

**4.4 - Raffinamento dello schema (eliminazione di identificatori esterni, attributi composti e gerarchie, scelta delle chiavi) e analisi delle ridondanze**

Il modello proposto finora non presenta **attributi composti** e/o **ridondanze**.

**Gerarchie:** AMMINISTRATORE e PASSEGGERO, nel modello concettuale sono entrambe PERSONE. Per la modellazione logica si è scelto di eliminare l’entità persona e introdurre i suoi attributi nelle entità della gerarchia.

Le **chiavi** scelte per le entità proposte sono le seguenti:

|  |  |
| --- | --- |
| **ENTITA’** | **CHIAVE PRIMARIA** |
| AMMINISTRATORE | Ogni AMMINISTRATORE è identificato univocamente dall’indirizzo email con la quale si registra sul portale web. |
| PASSEGGERO | Ogni PASSEGGERO è identificato dal proprio codice fiscale. |
| PRENOTAZIONE | Ogni PRENOTAZIONE ha un numero progressivo (univoco) che la identifica. |
| BIGLIETTO | Ogni BIGLIETTO è identificato da un numero progressivo, dal codice fiscale del PASSEGGERO che lo ha acquistato e dal numero della PRENOTAZIONE che lo riguarda. |
| VOLO | Ogni VOLO è identificato da un numero progressivo (univoco). |
| TARIFFA | Ogni TARIFFA ha un codice univoco che la identifica. |
| COMPAGNIA\_AEREA | Ogni COMPAGNIA\_AEREA è identificata dal suo nome. |
| LOCALITA | Ogni LOCALITA è identificata univocamente dai campi “nome”, “nazionalità”, “CAP”. |

**4.5 - Traduzione di entità e associazioni in relazioni**

* **AMMINISTRATORE** (email, cognome, nome, password)
* **PASSEGGERO** (codice\_fiscale, cognome, nome, data\_nascita, id\_card)
* **BIGLIETTO** (num\_b, codice\_fiscale, num\_p)

FK: codice\_fiscale REFERENCES: PASSEGGERO

FK: num\_p REFERENCES: PRENOTAZIONE

* **PRENOTAZIONE** (num\_p, classe, num\_passeggeri, num\_minori, num\_bagagli, costo\_p, num\_v)

FK: num\_v REFERENCES: VOLO

* **VOLO** (num\_v, data, num\_pc1, num\_pc2, ora\_partenza, ora\_arrivo, prezzo, nome, par\_nome,

par\_naz, par\_cap, des\_nome, des\_naz, des\_cap, codice\_t)

FK: nome REFERENCES: COMPAGNIA\_AEREA

FK: par\_nome, par\_naz, par\_cap REFERENCES: LOCALITA

FK: des\_nome, des\_naz, des\_cap REFERENCES: LOCALITA

FK: codice\_t REFERENCES: TARIFFA

* **COMPAGNIA**\_AEREA (nome, email, telefono)
* **TARIFFA** (codice\_t, classe1, classe2, sconto\_minori, costo\_bagaglio)
* **LOCALITA** (nome, naz, cap)

**4.6 - Schema relazionale finale**

****

**4.7 - Traduzione delle operazioni in query SQL**

**INSERIMENTI:**

***Utente che si registra come amministratore***

INSERT INTO AMMINISTRATORE

VALUES ('nome','cognome','email','password')

***Nuova compagnia aerea***

INSERT INTO COMPAGNIA\_AEREA

VALUES ('nome','email','telefono')

***Nuova località***

INSERT INTO LOCALITA

VALUES ('nome','nazionalità','cap')

***Nuova tariffa***

INSERT INTO TARIFFA (classe1, classe2, sconto\_minori, costo\_bagaglio)

VALUES ('sovrapprezzo classe 1','sovrapprezzo classe 2','sconto minori','costo bagaglio')

***Nuovo volo***

INSERT INTO VOLO (data, num\_pc1, num\_pc2, ora\_partenza, ora\_arrivo, prezzo, nome, par\_nome, par\_naz, par\_cap, des\_nome, des\_naz, des\_cap, codice\_t)

VALUES ('data', 'posti classe 1', 'posti classe 2','orario di partenza','orario di arrivo','prezzo','compagnia aerea','nome della località di partenza','nazionalità di partenza','cap di partenza','nome della destinazione','nazionalità di destinazione','cap della destinazione','codice tariffa')

***Nuovo passeggero***

INSERT INTO PASSEGGERO

VALUES ('cognome','nome','codice fiscale','data di nascita','id card')

***Nuova prenotazione***

INSERT INTO PRENOTAZIONE (classe, num\_passeggeri, num\_minori, num\_bagagli, costo\_p, num\_v)

VALUES ('classe','# passeggeri','# minori','# bagagli','costo','numero del volo')

***Nuovo biglietto***

INSERT INTO BIGLIETTO (codice\_fiscale, num\_p)

VALUES ('codice fiscale','numero di prenotazione')

**ESTRAZIONI:**

***Amministratore che ha effettuato il login***

SELECT \*

FROM AMMINISTRATORE

WHERE email='email\_admin'

***Voli previsti in una data inserita dall'utente, con rispettive tariffe***

SELECT num\_v, ora\_partenza, par\_nome, par\_naz, par\_cap, ora\_arrivo, des\_nome, des\_naz, des\_cap, VOLO.nome, prezzo, classe1, classe2, sconto\_minori, costo\_bagaglio

FROM VOLO, TARIFFA

WHERE data = data AND VOLO.codice\_t = TARIFFA.codice\_t

ORDER BY num\_v

***Volo scelto dall'utente per effettuare una prenotazione, con corrispondente tariffa***

SELECT num\_v, ora\_partenza, par\_nome, par\_naz, par\_cap, ora\_arrivo, des\_nome, des\_naz, des\_cap, data, VOLO.nome, prezzo, classe1, classe2, sconto\_minori, costo\_bagaglio

FROM VOLO, TARIFFA

WHERE num\_v = 'numero del volo' AND VOLO.codice\_t = TARIFFA.codice\_t

***Passeggero che inserisce il suo codice fiscale per visualizzare le sue prenotazioni***

SELECT codice\_fiscale

FROM PASSEGGERO

WHERE codice\_fiscale = 'codice\_fiscale'

***Prenotazioni effettuate da un determinato passeggero, con corrispondenti biglietti e voli prenotati***

SELECT VOLO.num\_v AS numv, ora\_partenza, par\_nome, par\_naz, par\_cap, ora\_arrivo, des\_nome, des\_naz, des\_cap, data, VOLO.nome AS nomeca, classe, num\_passeggeri, num\_minori, num\_bagagli

FROM VOLO, PRENOTAZIONE, BIGLIETTO, PASSEGGERO

WHERE PASSEGGERO.codice\_fiscale = 'codice\_fiscale' AND PASSEGGERO.codice\_fiscale = BIGLIETTO.codice\_fiscale AND BIGLIETTO.num\_p = PRENOTAZIONE.num\_p AND PRENOTAZIONE.num\_v = VOLO.num\_v

***Tariffe da scegliere per l'inserimento di un nuovo volo***

SELECT \* FROM TARIFFA

**5 - PROGETTAZIONE DELL’ APPLICAZIONE**

**5.1 - Descrizione dell'architettura dell'applicazione realizzata**

Il portale web realizzato è composto da 1 database, 3 fragments JSPF, 1 foglio di stile CSS3 e 18 pagine JSP, il tutto per realizzare:

* una homepage che presenta il portale;
* una sessione composta da 3 pagine, una per la ricerca dei voli in un certa data ed eventualmente per la prenotazione di uno di essi, una per inserire tutti i dati necessari per la prenotazione ed una per l’invio al database delle query SQL, per la conferma dell’avvenuta prenotazione e per la stampa nel numero dei biglietti associati ad essa;
* una sessione composta da 2 pagine, una l’inserimento del codice fiscale dell’utente della quale si vuole visualizzare ogni prenotazione da esso effettuata e una per visualizzare, appunto, tutte le prenotazioni di quel determinato utente (mediante invio di query SQL al DB);
* una sessione composta da 12 pagine: di cui 5 dedicate alla registrazione, e login di un amministratore e 7 per l’inserimento nel database di un nuovo volo e/o una nuova località, tariffa, compagnia aerea.

**Presentazione proposta nella homepage del portale web:**

Benvenuto nel **FLIGHTS MANAGEMENT SYSTEM**!

Gli **utenti** possono:

* cercare i voli previsti in un certa data;
* prenotare un volo;
* visualizzare tutte le prenotazioni effettuate.

Gli **amministratori** del sistema possono:

* registrarsi se non lo si è già fatto;
* effettuare il login;
* inserire nuovi voli;
* inserire nuove località;
* inserire nuove tariffe;
* inserire nuove compagnie aeree.