

CONSEGNA G4 - SISTEMA INFORMATIVO PER LA GESTIONE DELL'EMISSIONE DI BIGLIETTI AEREI

## Relazione finale del progetto di Basi di Dati

Candidato:

**Giovanni Liboni**

Matricola VR363021

## Contents

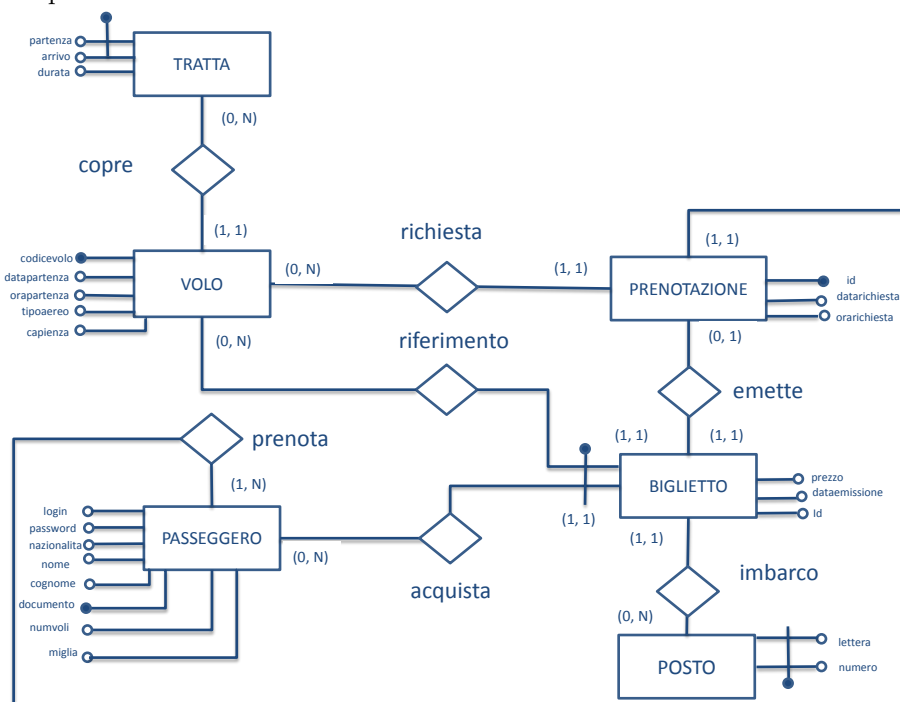
<b>I</b>	<b>Progetto della basi di dati</b>	<b>2</b>
1	Progetto concettuale	2
2	Progetto logico	3
3	Popolamento della base di dati	6
<b>II</b>	<b>Progetto del sito web</b>	<b>6</b>
4	Progettazione logica	6
5	Struttura dell'applicazione web	7
<b>III</b>	<b>Architettura MVC2</b>	<b>7</b>
<b>IV</b>	<b>Model</b>	<b>7</b>
<b>V</b>	<b>View</b>	<b>7</b>
<b>VI</b>	<b>Controller</b>	<b>7</b>
<b>VII</b>	<b>Scelte progettuali</b>	<b>8</b>
6	Hibernate	8
7	AJAX	8
8	Session	8

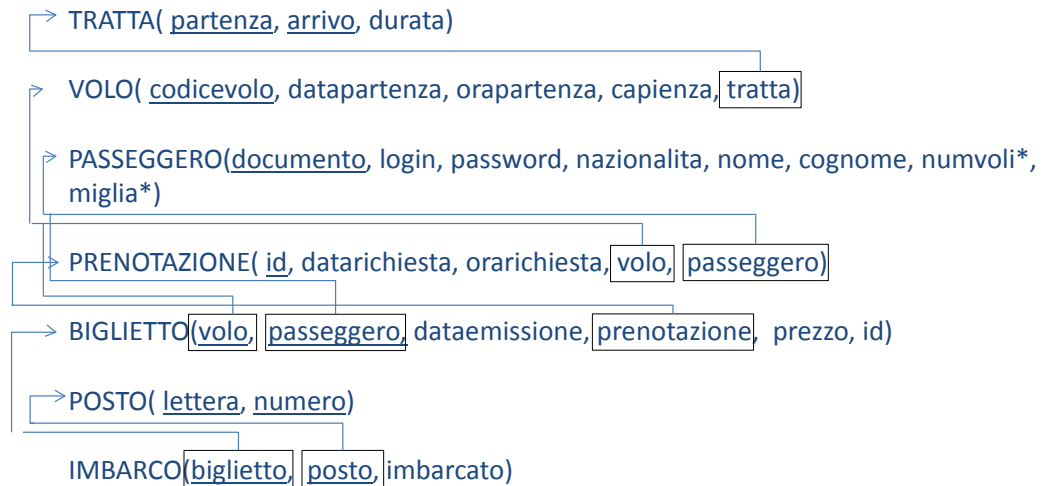
## Part I

# Progetto della basi di dati

## 1 Progetto concettuale

Qui di seguito si riporta il diagramma Entità - Relazione e lo schema concettuale ricavati dalle specifiche del progetto. A partire da questo schema si sono scritte le tabelle per la Base di Dati.





Descrivere il diagramma e lo schema logico.

## 2 Progetto logico

Abbiamo usato un singolo script per la creazione delle tabelle sul database. Il codice viene riportato qui di seguito.

```

1 CREATE TABLE tratta(
2   partenza VARCHAR(100) NOT NULL,
3   arrivo   VARCHAR(100) NOT NULL,
4   durata   INTEGER NOT NULL,
5   distanza NUMERIC(10,1) NOT NULL DEFAULT '0.0',
6   PRIMARY KEY ( partenza, arrivo ),
7   UNIQUE ( partenza, arrivo )
8 );
9 CREATE TABLE volo(
10  codicevolo VARCHAR(10) NOT NULL,
11  partenza   VARCHAR(100) NOT NULL,
12  arrivo     VARCHAR(100) NOT NULL,
13  datapartenza DATE NOT NULL,
14  orapartenza TIME NOT NULL,
15  tipoaereo   VARCHAR(50) NOT NULL,
16  capienza    INTEGER NOT NULL,
17  PRIMARY KEY( codicevolo ),
18  UNIQUE ( codicevolo ),
19  FOREIGN KEY( partenza, arrivo )

```

```

20 REFERENCES tratta( partenza , arrivo )
21 ON UPDATE CASCADE
22 ON DELETE CASCADE
23 );
24 CREATE TABLE passeggero(
25 login VARCHAR(100) NOT NULL,
26 password VARCHAR(100) NOT NULL,
27 nazionalita VARCHAR(100) NOT NULL,
28 nome VARCHAR(100) NOT NULL,
29 cognome VARCHAR(100) NOT NULL,
30 documento VARCHAR(50) NOT NULL,
31 picture BYTEA,
32
33 numvoli INTEGER DEFAULT 0,
34 miglia FLOAT DEFAULT 0,
35 tessera BOOLEAN DEFAULT FALSE,
36 PRIMARY KEY( documento ),
37 UNIQUE ( login )
38 );
39 CREATE TABLE prenotazione(
40 id SERIAL,
41 codicevolo VARCHAR(10) NOT NULL
42 REFERENCES volo( codicevolo )
43 ON UPDATE CASCADE
44 ON DELETE CASCADE,
45 documento VARCHAR(50) NOT NULL
46 REFERENCES passeggero( documento )
47 ON UPDATE CASCADE
48 ON DELETE CASCADE,
49 datarichiesta DATE,
50 orarichiesta TIME,
51 PRIMARY KEY( id ),
52 UNIQUE ( codicevolo , documento )
53 );
54 CREATE TABLE biglietto(
55 id SERIAL,
56 codicevolo VARCHAR(10) NOT NULL
57 REFERENCES volo( codicevolo )
58 ON UPDATE CASCADE
59 ON DELETE CASCADE,
60 documento VARCHAR(50) NOT NULL
61 REFERENCES passeggero( documento )
62 ON UPDATE CASCADE
63 ON DELETE CASCADE,
64 id_prenotazione INTEGER NOT NULL
65 REFERENCES prenotazione( id )
66 ON UPDATE CASCADE
67 ON DELETE CASCADE,
68 dataemissione DATE NOT NULL,
69 prezzo NUMERIC(10,2) NOT NULL,
70
71 PRIMARY KEY( codicevolo , documento ),
72 UNIQUE ( codicevolo , documento , id )
73 );
74
75 CREATE TABLE posto(
76 lettera CHAR NOT NULL,
77 numero INTEGER DEFAULT 0,
78 PRIMARY KEY ( lettera , numero ),
79 UNIQUE ( lettera , numero )
80 );
81 CREATE TABLE imbarco(

```

```

82 lettera CHAR NOT NULL,
83 numero INTEGER DEFAULT 0,
84 imbarcato BOOLEAN DEFAULT FALSE,
85 -- PRIMARY KEY DEL BIGLIETTO
86 codicevolo VARCHAR(10) NOT NULL,
87 documento VARCHAR(50) NOT NULL,
88 -- IDENTIFICATO DAL BIGLIETTO E DAL POSTO
89 PRIMARY KEY ( codicevolo , documento , lettera , numero),
90
91 FOREIGN KEY( lettera , numero )
92 REFERENCES posto( lettera , numero )
93 ON UPDATE CASCADE
94 ON DELETE CASCADE,
95 FOREIGN KEY( codicevolo , documento )
96 REFERENCES biglietto( codicevolo , documento )
97 ON UPDATE CASCADE
98 ON DELETE CASCADE
99 );

```

Abbiamo aggiunto delle funzioni scritte in *plpgsql* per eseguire azioni di routine sul base di dati e per inserire automaticamente la data e l'ora di inserimento di determinate tuple. Si riporta di seguito il codice delle funzioni create.

```

1 CREATE OR REPLACE FUNCTION update_time_richiesta()
2 RETURNS TRIGGER AS '
3 BEGIN
4     IF NEW.datarichiesta IS NULL THEN
5         NEW.datarichiesta := current_date;
6     END IF;
7     IF NEW.orarichiesta IS NULL THEN
8         NEW.orarichiesta := current_time;
9     END IF;
10    RETURN NEW;
11 END' LANGUAGE 'plpgsql';
12
13
14
15 CREATE OR REPLACE FUNCTION update_time_biglietto()
16 RETURNS TRIGGER AS '
17 BEGIN
18     IF NEW.dataemissione IS NULL THEN
19         NEW.dataemissione := current_date;
20     END IF;
21    RETURN NEW;
22 END' LANGUAGE 'plpgsql';
23
24
25
26 CREATE OR REPLACE FUNCTION update_numvoli()
27 RETURNS TRIGGER AS '
28 BEGIN
29     IF NEW.dataemissione IS NULL THEN
30         NEW.dataemissione := current_date;
31     END IF;
32    RETURN NEW;
33 END' LANGUAGE 'plpgsql';
34
35 CREATE OR REPLACE FUNCTION update_tessera() RETURNS trigger AS $$
36 BEGIN
37     UPDATE passeggero
38     SET numvoli = (SELECT count(*)
39                     FROM volo JOIN biglietto ON biglietto.codicevolo = volo.
                     codicevolo

```

```

40         JOIN tratta ON (volo.partenza = tratta.partenza
41             AND volo.arrivo = tratta.arrivo )
42     WHERE passeggero.tessera=true
43         AND passeggero.documento = biglietto.documento
44     GROUP BY passeggero.documento ),
45     miglia = (SELECT sum(distanza)
46 FROM volo JOIN biglietto ON biglietto.codicevolo = volo.
47     codicevolo
48     JOIN tratta ON (volo.partenza = tratta.partenza AND volo.
49     arrivo = tratta.arrivo )
50 WHERE passeggero.tessera=true AND passeggero.documento =
51     biglietto.documento
52     GROUP BY passeggero.documento )
53 FROM biglietto
54 WHERE biglietto.dataemissione >= ( SELECT date('now') - interval
55     '3 year')
56     AND NEW.documento = passeggero.documento;
57 RETURN NEW;
58 END;
59 $$ LANGUAGE 'plpgsql';

```

Per automatizzare l'esecuzione delle funzioni si sono scritti dei trigger, azioni eseguite al verificarsi di certe condizioni. I trigger creati sono i seguenti:

```

1 CREATE TRIGGER update_time_richiesta
2 BEFORE INSERT ON prenotazione
3     FOR EACH ROW EXECUTE PROCEDURE update_time_richiesta();
4
5 CREATE TRIGGER update_tessera
6 AFTER INSERT OR DELETE ON biglietto
7     FOR EACH ROW EXECUTE PROCEDURE update_tessera();
8
9 CREATE TRIGGER update_time_biglietto
10 BEFORE INSERT ON biglietto
11     FOR EACH ROW EXECUTE PROCEDURE update_time_biglietto();

```

### 3 Popolamento della base di dati

## Part II

# Progetto del sito web

### 4 Progettazione logica

Si riportano di seguito gli schemi di pagina seguiti durante la creazione del sito.

## 5 Struttura dell'applicazione web

### Part III

## Architettura MVC2

L'elaborato è stato sviluppato seguendo il design pattern MVC2 e seguendo l'approccio servlet-centric. Questo pattern è composto da tre moduli:

- Model - Comprende la classe DBMS.java e i Java Data Beans. I beans sono contenuti nel package *bean*, mentre la classe DBMS.java è contenuta nel package *database*;
- View - Comprende tutte le JSPs, i javascript e i css;
- Controller - Comprende le servlet main.java, picture.java. Ogni servlet ha un compito specifico: ... ;

### Part IV

## Model

Fanno parte del modulo Model il DBMS e i javabeans.

### Part V

## View

Fanno parte del modulo View tutte la JSPs, i javascript e i css.

### Part VI

## Controller

Fanno parte del modulo Controller le due servlet. Le due servlet che controllano il flusso sono *main.java* e *picture.java*.

La servlet *picture.java* gestisce la parte multimediale dell'applicazione, caricando e scaricando dalla base di dati le foto profilo dei passeggeri. Per ogni richiesta viene passato il parametro *ps* con un determinato valore e può assumere questi valori:

- uploadimage
- downloadimage

La servlet *main.java* controlla le richieste da parte delle JSPs.



- areapersonale - Vengono caricati dalla base di dati i voli e le prenotazioni dell'utente specificato nel parametro *pass*.
- ricercavolo
- volipage
- prenotazione
- nuovaprenotazione
- logout
- login
- contatti
- emettibiglietto
- newbiglietto
- chisiamo
- ajaxricercavolo
- checkusername
- checkdocumento

## Part VII

# Scelte progettuali

## 6 Hibernate

## 7 AJAX

## 8 Session