

DOCUMENTAZIONE DATABASE

Corso di Basi di Dati I



UNIVERSITÀ DEGLI STUDI
DI NAPOLI FEDERICO II

STUDENTI:

Cognome: Buonanno
Cognome: De Marco

Nome: Manuel
Nome: Luigi

Matricola: N86003256
Matricola: N86003287

ID Gruppo: 2
Traccia: 4

INDICE

1. Analisi del problema
2. Class diagram
3. Dizionario delle classi
4. Dizionario delle associazioni
5. Dizionario dei vincoli
6. Schema logico
7. Descrizione dei triggers
8. Descrizione delle function e delle procedures
9. Definizione delle tabelle

1. ANALISI DEL PROBLEMA

Si vuole sviluppare un Database in grado di gestire i dati riguardanti uno scalo aeroportuale. La gestione deve comprendere: le tratte, i gates, le compagnie aeree che riguardano l'aeroporto e le code di imbarco. Un gate può essere associato ad un volo nella giornata, ma mai a più di uno contemporaneamente. Un gate e un volo hanno dei tempi di imbarco effettivi e stimati, per l'eventuale calcolo dei ritardi. Il tempo stimato di un gate sarà dato dai tempi di imbarco delle varie code di imbarco che vengono associate ad esso. Ogni tratta sarà resa disponibile da una compagnia aerea registrata all'aeroporto. Ogni volo in partenza avrà la propria tratta, data e orario, status imbarco e status volo.

2.CLASS DIAGRAM

Class diagram presente nell'altro PDF.

3. DIZIONARIO DELLE CLASSI.

CLASSE	DESCRIZIONE	ATTRIBUTI
VoloPartenza	Gestisce i voli in partenza dall'aeroporto.	<ul style="list-style-type: none">• codiceVoloPartenza char(6): codice univoco del volo in partenza;• dataOrarioPartenza timestamp: data e orario del volo;• numeroPrenotazioni varchar(3): numero delle prenotazioni per il volo;• tempoDilbarcoEffettivo time: tempo di imbarco al gate del volo;• statusVolo varchar(20): status del volo che ne descrive il progresso, se in programma, in preparazione, in partenza o ritardo;• statusImbarco varchar(20): status dell'imbarco, se in orario o in ritardo.
Tratta	Gestisce le tratte che prevede l'aeroporto.	<ul style="list-style-type: none">• codiceTratta char(6): codice univoco della tratta;• cittaPartenza varchar(25): città di partenza della tratta;• cittaArrivo varchar(25): città di arrivo della tratta.
CompagniaAerea	Gestisce le compagnie aeree affiliate all'aeroporto.	<ul style="list-style-type: none">• codiceCompagniaAerea char(6): codice univoco della compagnia aerea;• nome varchar(25): nome della compagnia aerea.
Gate	Gestisce i gate presenti in aeroporto.	<ul style="list-style-type: none">• codiceGate char(6): codice univoco del gate;• numeroPorta varchar(2): numero della porta del gate;• tempoDilbarcoStimato time: indica il tempo stimato di imbarco al gate;• tempoChiusuraGate time: indica il tempo in cui il gate chiude.
CodaDilbarco	Gestisce le code di imbarco dell'aeroporto.	<ul style="list-style-type: none">• codiceCodaDilbarco char(6): codice univoco della coda di imbarco;• nomeCoda varchar(25): nome della coda di imbarco;• tempoDilbarcoStimato codi time: indica il tempo stimato di imbarco per la coda di imbarco.
VoloArrivo	Gestisce i voli in arrivo all'aeroporto.	<ul style="list-style-type: none">• codiceVoloArrivo char(6): codice univoco del volo in arrivo;• cittaPartenza varchar(30): nome della città di arrivo;• dataOrarioArrivo timestamp: indica la data e l'orario di arrivo del volo.
Utente	Gestisce gli utenti registrati all'applicativo.	<ul style="list-style-type: none">• nome varchar(30): nome dell'utente che si registra;• cognome varchar(30): cognome dell'utente che si registra;• email varchar(30): e-mail dell'utente che si registra;• password varchar(30): password dell'utente che si registra.

4. DIZIONARIO DELLE ASSOCIAZIONI.

ASSOCIAZIONE	DESCRIZIONE	PARTECIPANTI
ospita	Descrive il legame tra VoloPartenza e Gate , in quanto un gate può ospitare più voli nella giornata, ma mai nello stesso orario.	VoloPartenza; Ruolo: è ospitato (n,1) Gate; Ruolo: ospita (1,n)
appartiene	Descrive il legame tra VoloPartenza e Tratta , in quanto una determinata tratta appartiene a un volo in partenza.	VoloPartenza; Ruolo: ha (n,1) Tratta; Ruolo: appartiene (1,n)
gestisce	Descrive il legame tra Tratta e CompagniaAerea , in quanto è la compagnia aerea che si occupa di gestire tutte le tratte.	Tratta ; Ruolo: e gestita (n,1) CompagniaAerea ; Ruolo: gestisce (1,n)
popola	Descrive il legame tra Gate e CodaDilImbarco , in quanto un gate ospita diverse code di imbarco.	Gate ; Ruolo: e popolato (n,n) CodaDilImbarco ; Ruolo: popola (n,n)

5. DIZIONARIO DEI VINCOLI

- 1) La password nella fase di registrazione account deve essere compresa tra gli 8 e i 16 caratteri.

6. SCHEMA LOGICO

Utente (nome, cognome, email, password)

VoloArrivo (codiceVoloArrivo, cittaPartenza, dataOrarioArrivo)

VoloPartenza (codiceVoloPartenza, dataOrarioArrivo, numeroPrenotazioni, tempoDilImbarcoEffettivo, statusVolo, statusImbarco, xCodiceTratta, xCodiceGate)

Tratta (codiceTratta, cittaPartenza, CittaArrivo, xCodiceCompagniaAerea)

CompagniaAerea (codiceCompagniaAerea, nome)

Gate (codiceGate, numeroPorta, tempoDilImbarcoStimato, tempoChiusuraGate)

CodaDilImbarco (codiceCodaDilImbarco, nomeCoda, tempoDilImbarcoStimato)

CodaDilImbarcoGate (xCodiceGate, xCodiceCodaDilImbarco).

7. DESCRIZIONE DEI TRIGGERS

- Il trigger **CalcoloTempolImbarcoStimato** attiva la *function* **CalcoloTempolImbarcoStimatoDaCodaDilImbarco()**, dopo un update alla tabella *CodaDilImbarco*;
- Il trigger **CalcoloTempolImbarcoStimatoElimina** attiva la *function* **CalcoloTempolImbarcoStimatoElimina()**, dopo un delete alla tabella *CodaDilImbarcoGate*;
- Il trigger **ControlloCodaAssociata** attiva la *function* **ControlloCodaAssociata ()**, prima di un insert alla tabella *CodaDilImbarcoGate*;
- Il trigger **AggiornaStatusImbarco** attiva la *function* **AggiornaStatusImbarco ()**, dopo un insert o update alla tabella *VoloPartenza*;
- Il trigger **VoloContemporaneamenteInGate** attiva la *function* **VoloContemporaneamenteInGate()**, prima di un insert sulla tabella *VoloPartenza*.

8. DESCRIZIONE FUNCTIONS E PROCEDURES

- **CalcoloTempolImbarcoStimatoAggiunta()**: questa *function* permette di calcolare il tempo di imbarco stimato di un gate all'aggiunta di una relazione tra *gate e coda di imbarco*;
- **AggiornaStatusImbarco()**: questa *function* ci informa sullo status dell'imbarco, andando a prendere i tempi di imbarco effettivo e stimato;
- **CalcoloTempolImbarcoStimatoDaCodaDiImbarco()**: questa *function* permette di calcolare il tempo di imbarco stimato di un gate nel momento in cui il tempo della coda di imbarco viene modificato;
- **CalcoloTempolImbarcoStimatoElimina()**: questa *function* calcola il tempo di imbarco stimato di un gate nel momento in cui viene eliminata un'associazione tra un **gate e una coda di imbarco**;
- **ControlloCodaAssociata()**: questa *function* controlla se un gate e una coda di imbarco sono già associate andando a evitare duplicazioni di associazioni;
- **DatoInesistenteVoloPartenza()**: questa *function* va a verificare l'esistenza di un volo andando a controllare la presenza di un dato codice nel database;
- **VoloContemporaneamenteInGate()**: questa *function* evita che due voli siano registrati nello stesso gate allo stesso orario;
- **CalcoloTempolImbarcoStimatoAggiunta(codicegt character)**: questa *procedure* calcola il tempo di imbarco stimato di un gate in base al parametro *codicegt*;
- **CalcoloTempolImbarcoStimatoAggiunta(codicegt character, tempodiimbarcostimatoattuale time without time zone)**: questa *procedure* calcola il tempo di imbarco stimato di un gate tenendo conto dei parametri *codicegt* e *tempodiimbarcostimatoattuale*, passati dalla *function* che lo richiama.

9. DEFINIZIONE DELLE TABELLE

```
CREATE TABLE public.codadiimbarco (  
    codiccodadiimbarco character(6) NOT NULL,  
    nomecoda character varying(25),  
    tempodiimbarcostimatocdi time without time zone NOT NULL  
);  
  
CREATE TABLE public.codadiimbarcogate (  
    xcodicegate character(6),  
    xcodicecodadiimbarco character(6)  
);  
  
CREATE TABLE public.compagniaaerea (  
    codicecompagniaaerea character(6) NOT NULL,  
    nome character varying(25) NOT NULL  
);  
  
CREATE TABLE public.gate (  
    codicegate character(6) NOT NULL,  
    numeroporta character varying(2) NOT NULL,  
    tempodiimbarcostimato time without time zone DEFAULT '00:00:00'::time without time zone,  
    tempochiusuragate time without time zone  
);  
  
CREATE TABLE public.tratta (  
    codicetratta character(6) NOT NULL,  
    cittapartenza character varying(25) NOT NULL,  
    cittaarrivo character varying(25) NOT NULL,  
    xcodicecompagniaaerea character(6)  
);  
  
CREATE TABLE public.utente (  
    nome character varying(30) NOT NULL,  
    cognome character varying(30) NOT NULL,  
    email character varying(30) NOT NULL,  
    password character varying(30) NOT NULL,  
    CONSTRAINT minlengthconstraint CHECK ((length((password)::text) > 7))  
);
```

```
CREATE TABLE public.voloarrivo (  
    codicevoloarrivo character(6) NOT NULL,  
    cittapartenza character varying(30) NOT NULL,  
    dataorarioarrivo timestamp without time zone
```

```
);
```

```
CREATE TABLE public.volopartenza (  
    codicevolopartenza character(6) NOT NULL,  
    dataorariopartenza timestamp without time zone NOT NULL,  
    numeroprenotazioni character varying(3) NOT NULL,  
    tempodiimbarcoeffettivo time without time zone,  
    statusvolo character varying(20) NOT NULL,  
    statusimbarco character varying(20),  
    xcodicegate character(6),  
    xcodicetratta character(6),  
    CONSTRAINT aggiungivolovecchio CHECK ((dataorariopartenza > now()))
```

```
);
```