Basi di dati

A.A. 2019/2020

Progettazione base di dati

Servizio di car pooling

Laura Bragagnolo

Leonardo Parancola

Alessandra Tonin

Indice

1. Descrizione del progetto3

1.1 Requisiti strutturali3

1.2 Operazioni sulla base di dati4

2. Progettazione concettuale5

2.1 Modello concettuale: Schema E-R5

2.2 Modello concettuale: Ristrutturazione schema E-R6

2.3 Schema E-R: Regole di vincolo7

2.4 Dizionario dei dati8

3. Progettazione logica10

3.1 Modello logico: Schema relazionale10

3.2 Schema relazionale: Regole di vincolo11

4. Codice SQL12

4.1 Struttura12

4.2 Interrogazioni frequenti14

5. Sito web17

6. Note18

1. Descrizione del progetto

Una start-up vuole realizzare una banca dati per gestire un servizio di car pooling a livello europeo, per promuovere il rispetto dell’ambiente e la mobilità sostenibile. I dati relativi a passeggeri, autisti, viaggi e veicoli impiegati verranno memorizzati all’interno di un database relazionale.

1. Requisiti strutturali

Ogni nuovo utente sarà identificato come *passeggero* e sarà riconosciuto univocamente dal proprio codice fiscale. Al momento dell’iscrizione dovrà inserire anche il nome, il cognome, il numero di telefono e il proprio indirizzo e-mail.

Un utente potrà decidere, inoltre, di registrarsi anche come *autista*. In questo caso, dovrà fornire il numero e la data di scadenza della patente, una foto ed eventualmente una descrizione, per esprimere preferenze come, ad esempio, ascoltare musica durante il viaggio, chiacchierare con i passeggeri, oppure per indicare se è possibile o meno portare in viaggio animali o bambini.

Ciascun autista inserisce i *veicoli* che può utilizzare per i suoi viaggi, identificati dal numero di targa. Di ogni veicolo interessa sapere il modello, il numero di posti disponibili, la casa automobilistica e il numero di polizza assicurativa.

Ogni *viaggio* è identificato univocamente dall’autista e dalla data e ora di partenza e deve riportare informazioni riguardanti la*tratta* (composta da città di partenza e di arrivo), la durata stimata, il costo totale, i posti disponibili, i dati del veicolo utilizzato ed eventuali note. L’autista si riserva di concordare l’indirizzo preciso di partenza e arrivo con i passeggeri, tramite posta elettronica o servizi di messaggistica. È presente, inoltre, un attributo che identifica lo stato delle prenotazioni (aperte o chiuse).

Al termine del viaggio, ogni passeggero può lasciare un breve commento e attribuire un voto da 1 a 10 all’autista. La media dei voti ricevuti costituirà il rating dell’autista che potrà essere preso in considerazione dagli utenti al momento della prenotazione.

Il servizio opera in gran parte dei paesi dell’area europea e le *città* sono identificate univocamente da un codice interno. A Firenze, ad esempio, corrisponde il codice ITA000035, dove ITA è il codice ISO della nazione di appartenenza e 000035 è un numero progressivo. Di ogni città si memorizza il nome, la provincia e il CAP. Ogni *nazione*, invece, è descritta dal nome e dal codice ISO corrispondente.

1. Operazioni frequenti sulla base di dati

|  |  |  |
| --- | --- | --- |
| **Operazione** | **Tipo** | **Frequenza** |
| Inserimento Utenti (passeggeri/autisti) | Aggiornamento | 50.000/giorno |
| Inserimento Viaggio | Aggiornamento | 100.000/giorno |
| Inserimento Tratta | Aggiornamento | 1.000/giorno |
| Inserimento Veicolo | Aggiornamento | 1.000/giorno |
| Inserimento Prenotazione | Aggiornamento | 500.000/giorno |
| Inserimento Feedback | Aggiornamento | 500.000/giorno |
| Aggiornamento ranking autista | Aggiornamento | 1/ora |
| Lista delle prenotazioni aperte data la tratta | Interrogazione | 1.000/ora |
| Lista delle tratte più percorse | Interrogazione | 1/settimana |
| Città più visitata nel 2019 | Interrogazione | 1/anno |
| Lista degli utenti sia passeggeri che autisti | Interrogazione | 1/mese |
| Lista dei passeggeri che non hanno mai prenotato un viaggio | Interrogazione | 2/anno |
| Modello di veicolo più utilizzato | Interrogazione | 2/anno |
| Autisti con il ranking più alto/basso | Interrogazione | 2/mese |
| Commenti dei feedback negativi (voto < 6) dati ad un autista | Interrogazione | 2/mese |

Tabella 1: Operazioni sulla base di dati

1. Progettazione concettuale
2. Modello Concettuale: Schema Entità-Associazione (E-R)

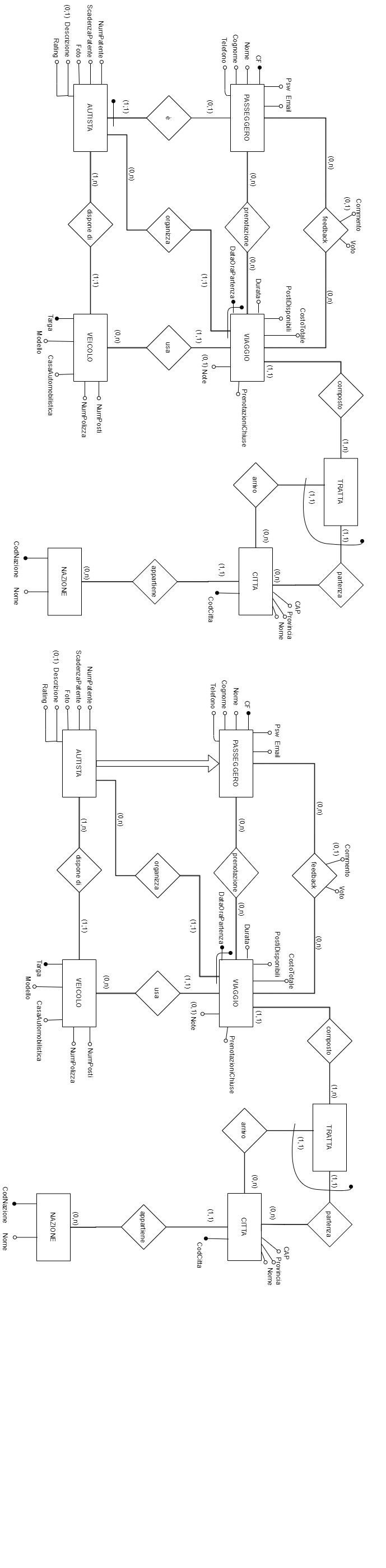


Figura 1: Schema E-R

1. Modello concettuale: Ristrutturazione schema E-R

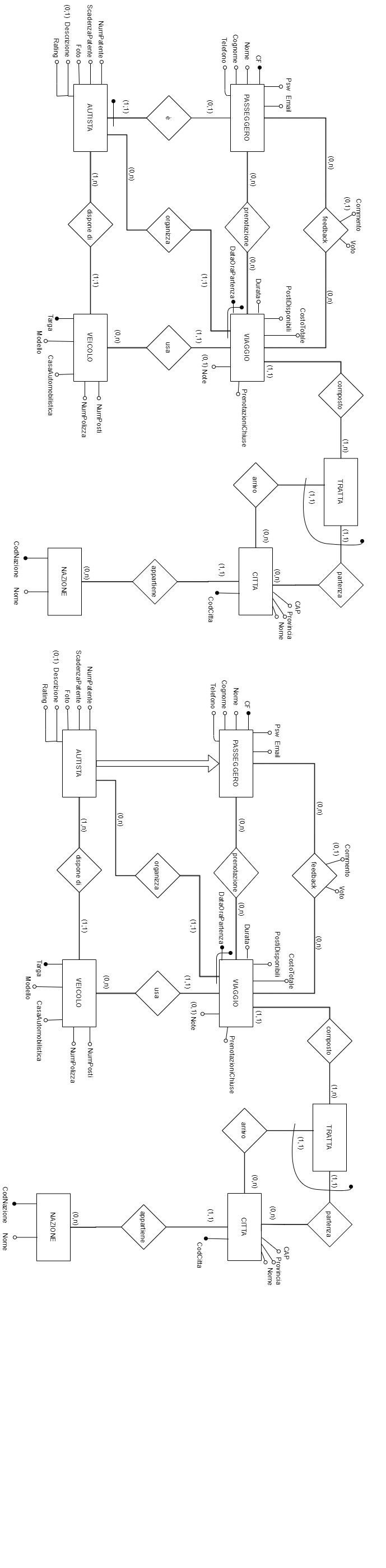


Figura 2: Schema E-R (ristrutturato)

1. Schema E-R: Regole di vincolo

Di seguito sono definite delle regole di vincolo per lo schema E-R definito in figura 2.

1. Il veicolo associato a ciascun viaggio deve essere presente tra i veicoli dell’autista.
2. Un passeggero può lasciare un feedback soltanto a viaggi a cui ha partecipato. Pertanto, l’insieme degli elementi presenti nell’associazione Feedback deve essere un sottoinsieme dell’associazione Prenotazione.
3. Un passeggero può prenotare un viaggio solamente se l’attributo PrenotazioniChiuse ha valore False. Tale attributo, che ha valore False di default, assume valore True quando PostiDisponibili è uguale a zero, o quando DataOraPartenza è maggiore o uguale alla data e ora correnti.
4. L’attributo NumPatente di Autista è unico.
5. Gli attributi Telefono e Email di Passeggero sono unici.
6. Dizionario dei dati

Di seguito sono riportate due tabelle (tabella 2 e tabella 3) nelle quali vi è una descrizione per ogni entità e per ogni associazione con il relativo volume.

Si suppone che siano registrati alla piattaforma all’incirca 10 milioni di utenti dei quali, mediamente, 1 su 5 è sia passeggero che autista. Ogni autista registra in media 2 veicoli e organizza approssimativamente 7 viaggi.

Il database contiene tutte le città europee, intese come *comuni* o *unità amministrative locali* (circa 200.000 elementi).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entità** | **Descrizione** | **Attributi** | **Identificatore** | **Volume** |
| PASSEGGERO | Utente registrato | CF, Nome, Cognome, Telefono, Email, Psw | CF | 10.000.000 |
| AUTISTA | Autista | Passeggero, NumPatente, ScadenzaPatente, Foto, Descrizione, Rating | Passeggero | 2.000.000 |
| VEICOLO | Veicolo appartenente all’autista | Targa, Modello, CasaAutomobilistica, NumPolizza, NumPosti | Targa | 4.000.000 |
| VIAGGIO | Viaggio organizzato | Autista, DataOraPartenza, CostoTotale, Durata, PostiDisponibili, PrenotazioniChiuse, Note, Veicolo, CittaP, CittaA | Autista, DataOraPartenza | 14.000.000 |
| TRATTA | Partenza e Destinazione | CittaP, CittaA | CittaP, CittaA | 500.000 |
| CITTÀ | Città di partenza o destinazione | CodCitta, Nazione, Nome, CAP, Provincia | CodCittà | 200.000 |
| NAZIONE | Nazione | CodNazione, Nome | CodNazione | 16 |

Tabella 2: Descrizione delle entità

|  |  |  |  |
| --- | --- | --- | --- |
| **Associazione** | **Attributi** | **Entità collegate** | **Volume** |
| FEEDBACK | Commento, Voto | Passeggero (0,n), Viaggio (0,n) | 38.000.000 |
| PRENOTAZIONE |  | Passeggero (0,n), Viaggio (0,n) | 40.000.000 |
| ORGANIZZA |  | Autista (0,n), Viaggio (1,1) | 14.000.000 |
| UTILIZZA |  | Autista (1,n), Veicolo (1,1) | 4.000.000 |
| USA |  | Viaggio (1,1), Veicolo (0,n) | 14.000.000 |
| COMPOSTO |  | Viaggio (1,1), Tratta (1,n) | 14.000.000 |
| PARTENZA |  | Tratta (1,1), Citta (0,n) | 500.000 |
| ARRIVO |  | Tratta (1,1), Citta (0,n) | 500.000 |
| APPARTIENE |  | Citta (1,1), Nazione (0,n) | 200.000 |

Tabella 3: Descrizione delle associazioni

1. Progettazione logica
   1. Modello logico: schema relazionale

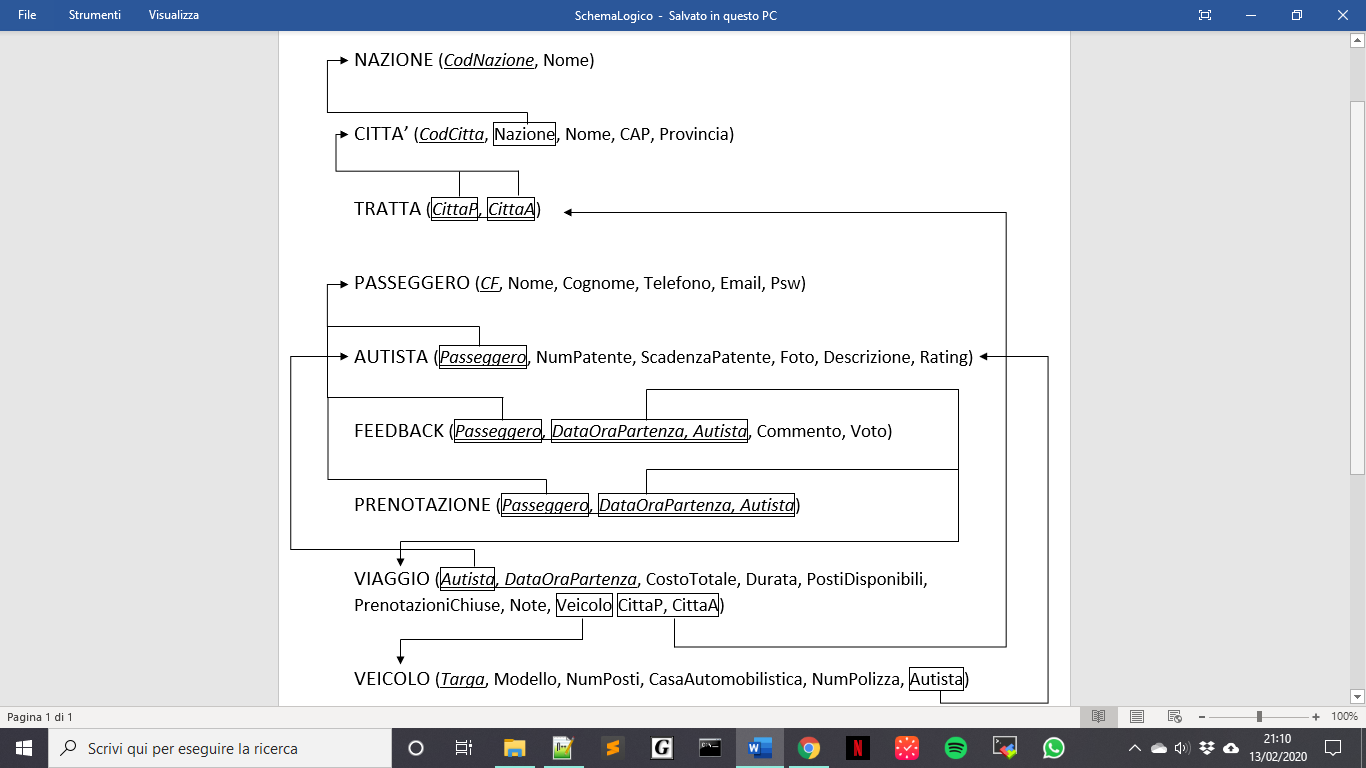


Figura 3: Schema relazionale

* 1. Schema relazionale: Regole di vincolo

Di seguito sono riportate le regole di vincolo relative agli attributi dello schema relazionale definito in figura 3.

1. Gli attributi di Passeggero non possono essere nulli. Gli attributi Email e Telefono devono essere unici.
2. Gli attributi NumPatente, ScadenzaPatente, Foto e Rating di Autista non possono essere nulli. L’attributo NumPatente, inoltre, deve essere unico.
3. L’attributo Descrizione di Autista può assumere valore nullo.
4. Gli attributi di Viaggio non possono essere nulli ad eccezione di Note.
5. L’attributo Voto di Feedback non può essere nullo, mentre Commento è un attributo opzionale.
6. Gli attributi di Veicolo non possono essere nulli.
7. Gli attributi di Città non possono essere nulli.
8. Gli attributi di Nazione non possono essere nulli.
9. Codice SQL
   1. Struttura

**CREATE TABLE** Passeggero**(**

CF **CHAR(16) UNIQUE,**

Nome **VARCHAR(20) NOT NULL,**

Cognome **VARCHAR(20) NOT NULL,**

Telefono **VARCHAR(14) NOT NULL,**

Email **VARCHAR(30) NOT NULL,**

Psw **CHAR(8) NOT NULL,**

**UNIQUE(**Telefono**,** Email**),**

**PRIMARY KEY(**CF**)**

**);**

**CREATE TABLE** Autista**(**

Passeggero **CHAR(16) UNIQUE,**

NumPatente **CHAR(10) NOT NULL,**

ScadenzaPatente **DATE NOT NULL,**

Foto **VARCHAR(10) NOT NULL,**

Descrizione **VARCHAR(250) NOT NULL,**

Rating **FLOAT NOT NULL,**

**PRIMARY KEY(**Passeggero**),**

**UNIQUE(**NumPatente**),**

**FOREIGN KEY (**Passeggero**) REFERENCES Passeggero(**CF**) ON UPDATE CASCADE**

**);**

**CREATE TABLE** Nazione**(**

CodNazione **CHAR(3) UNIQUE,**

Nome **VARCHAR(20) NOT NULL,**

**PRIMARY KEY(**CodNazione**)**

**);**

**CREATE TABLE** Citta**(**

CodCitta **CHAR(9) UNIQUE,**

Nome **VARCHAR(20) NOT NULL,**

CAP **CHAR(5) NOT NULL,**

Provincia **VARCHAR(2) NOT NULL,**

Nazione **CHAR(3) NOT NULL,**

**PRIMARY KEY(**CodCitta**),**

**FOREIGN KEY (**Nazione**) REFERENCES Nazione(**CodNazione**) ON UPDATE CASCADE**

**);**

**CREATE TABLE** Tratta**(**

CittaP **CHAR(9),**

CittaA **CHAR(9),**

**PRIMARY KEY(**CittaA**,** CittaP**),**

**UNIQUE(**CittaP**,** CittaA**),**

**FOREIGN KEY (**CittaA**) REFERENCES Citta(**CodCitta**) ON UPDATE CASCADE,**

**FOREIGN KEY (**CittaP**) REFERENCES Citta(**CodCitta**) ON UPDATE CASCADE**

**);**

**CREATE TABLE** Veicolo**(**

Targa **CHAR(7) UNIQUE,**

Modello **VARCHAR(20) NOT NULL,**

NumPosti **INT NOT NULL,**

CasaAutomobilistica **VARCHAR(20) NOT NULL,**

NumPolizza **INT NOT NULL,**

Autista **CHAR(16) NOT NULL,**

**UNIQUE(**NumPolizza**),**

**PRIMARY KEY(**Targa**),**

**FOREIGN KEY (**Autista**) REFERENCES Autista(**Passeggero**) ON UPDATE CASCADE**

**);**

**CREATE TABLE** Viaggio**(**

Autista **CHAR(16),**

DataOraPartenza **TIMESTAMP,**

CostoTotale **INT NOT NULL,**

Durata **INT NOT NULL,**

PostiDisponibili **INT NOT NULL,**

PrenotazioniChiuse **BOOLEAN DEFAULT FALSE,**

Note **VARCHAR(250),**

Veicolo **CHAR(7) NOT NULL,**

CittaP **CHAR(9) NOT NULL,**

CittaA **CHAR(9) NOT NULL,**

**PRIMARY KEY(**Autista**,** DataOraPartenza**),**

**UNIQUE(**Autista**,** DataOraPartenza**),**

**FOREIGN KEY (**Autista**) REFERENCES Autista(**Passeggero**) ON UPDATE CASCADE,**

**FOREIGN KEY (**Veicolo**) REFERENCES Veicolo(**Targa**) ON UPDATE CASCADE,**

**FOREIGN KEY (**CittaP**,** CittaA**) REFERENCES Tratta(**CittaP**,** CittaA**) ON UPDATE CASCADE**

**);**

**CREATE TABLE** Prenotazione**(**

Passeggero **CHAR(16),**

DataOraPartenza **TIMESTAMP,**

Autista **CHAR(16),**

**PRIMARY KEY(**Passeggero**,** DataOraPartenza**,** Autista**),**

**UNIQUE(**Passeggero**,** DataOraPartenza**,** Autista**),**

**FOREIGN KEY (**Autista**,** DataOraPartenza**) REFERENCES Viaggio(**Autista**,** DataOraPartenza**) ON UPDATE CASCADE,**

**FOREIGN KEY (**Passeggero**) REFERENCES Passeggero(**CF**) ON UPDATE CASCADE**

**);**

**CREATE TABLE** Feedback**(**

Passeggero **CHAR(16),**

DataOraPartenza **TIMESTAMP,**

Autista **CHAR(16),**

Commento **VARCHAR(140),**

Voto **INT NOT NULL CHECK(**Voto **> 0 AND** Voto **<= 10),**

**PRIMARY KEY(**Passeggero**,** DataOraPartenza**,** Autista**),**

**UNIQUE(**Passeggero**,** DataOraPartenza**,** Autista**),**

**FOREIGN KEY (**Autista**,** DataOraPartenza**) REFERENCES Viaggio(**Autista**,** DataOraPartenza**) ON UPDATE CASCADE,**

**FOREIGN KEY (**Passeggero**) REFERENCES Passeggero(**CF**) ON UPDATE CASCADE**

**);**

* 1. Interrogazioni frequenti

1. Lista delle prenotazioni aperte data la tratta

**SELECT** V.Autista, V.DataOraPartenza **AS** Partenza, V.Note, CostoTotale**/(**NumPosti **-** PostiDisponibili**)** **AS** Costo

**FROM** Viaggio **AS** V

**JOIN** Citta **AS** C1

**ON** V. CittaP **=** C1.CodCitta

**JOIN** Citta **AS** C2

**ON** CittaA **=** C2.CodCitta

**JOIN** Veicolo

**ON** Veicolo **=** Targa

**WHERE** C1.Nome **=** **'Padova'** **AND** C2.Nome **=** **'Padova'** **AND** PrenotazioniChiuse **=** **false;**

2. Lista delle 5 tratte più percorse

**SELECT** C1.Nome **AS** Partenza**,** C2.Nome **AS** Destinazione

**FROM** Viaggio **AS** V

**JOIN** Citta **AS** C1

**ON** V. CittaP **=** C1.CodCitta

**JOIN** Citta **AS** C2

**ON** CittaA **=** C2.CodCitta

**GROUP BY** C1.Nome**,** C2.Nome

**ORDER BY** **COUNT(\*)** **DESC**

**LIMIT** **5;**

3. Lista degli utenti

**SELECT** Passeggero.\***,** NumPatente

**FROM** Autista **RIGHT JOIN** Passeggero **ON** Passeggero **=** CF**;**

4. Lista dei passeggeri che non hanno mai prenotato un viaggio

**SELECT** \*

**FROM** Passeggero **AS** P

**WHERE NOT EXISTS (** **SELECT** \*

**FROM** Prenotazione

**WHERE** Passeggero **=** P.CF **);**

5. Modello di veicolo più utilizzato

**CREATE VIEW** VeicoloNumViaggi **AS(**

**SELECT** Modello**, COUNT(\*) AS** NumViaggi

**FROM** Viaggio **JOIN** Veicolo **ON** Veicolo **=** Targa

**GROUP BY** Modello

**);**

**SELECT** Modello, CasaAutomobilistica**,** NumPosti

**FROM** VeicoloNumViaggi **NATURAL JOIN** Veicolo

**WHERE** NumViaggi **= ( SELECT MAX(**NumViaggi**)**

**FROM** VeicoloNumViaggi **);**

6. Lista dei 5 autisti con il ranking più alto

**SELECT** Passeggero.\*, NumPatente

**FROM** Autista **JOIN** Passeggero **ON** Passeggero **=** CF

**ORDER BY** Rating **DESC**

**LIMIT** **5;**

7. Lista degli Autisti che hanno ricevuto voti minori di 6 e i relativi commenti.

**SELECT** Passeggero.\***,** Commento**,** Voto

**FROM** Feedback **JOIN** Passeggero **ON** Autista **=** CF

**WHERE** Voto **<** **6;**

8. Città più visitata nel 2019

**CREATE VIEW** CittaNumViaggi **AS(**

**SELECT** CittaA**, COUNT(\*) AS** NumViaggi

**FROM** Viaggio

**WHERE CAST(**DataOraPartenza **AS** text**)** **LIKE** **'%2019%'**

**GROUP BY** CittaA

**);**

**SELECT** Nome**,** Nazione

**FROM** Citta **JOIN** CittaNumViaggi **ON** CodCitta **=** CittaA

**WHERE** NumViaggi **= ( SELECT MAX(**NumViaggi**)**

**FROM** CittaNumViaggi **);**

1. Sito web

La piattaforma viene gestita attraverso un semplice sito web dinamico che si interfaccia con un database realizzato in PostgreSQL, generando pagine diverse a seconda delle richieste dell’utente.

Le varie pagine, tutte realizzate in PHP, permettono all’utente di svolgere sia operazioni di ricerca di dati all’interno del database, sia di utilizzare funzioni interattive come registrazione, login e logout.

La homepage del sito prevede la visualizzazione del logo dell’azienda, dello slogan e di un menù laterale che permette all’utente di navigare tra le varie sezioni. Di seguito verranno spiegate brevemente le varie funzionalità del sito, in particolare per quanto riguarda le operazioni di interrogazione e aggiornamento della base di dati.

Il menù laterale contiene le seguenti voci:

* CERCA VIAGGI: permette di cercare i viaggi attualmente disponibili selezionando le città di partenza e arrivo attraverso dei menù a tendina, che contengono i nomi delle città inserite nel database. Verrà visualizzato anche il costo del viaggio allo stato attuale, inteso come costo totale diviso il numero di prenotazioni finora ricevute: questo sarà quindi il prezzo che dovrà pagare l’utente se nessun altro passeggero si aggiungerà/toglierà.
* TOP 5 TRATTE: permette di visualizzare le 5 tratte più frequentate, in base ai viaggi inseriti nel portale;
* TOP CITTÀ 2019: permette di visualizzare la città più visitata nel 2019, indicando anche la nazione di appartenenza;
* AUTISTI MIGLIORI: visualizza i 5 autisti con rating più alto, in base alle valutazioni lasciate dai passeggeri;
* AUTISTI PEGGIORI: visualizza i nomi degli autisti peggiori, ossia quelli che hanno ricevuto una valutazione inferiore a 6/10, riportando anche i commenti lasciati insieme al voto;
* UTENTI: permette di visualizzare l’elenco completo degli utenti attualmente iscritti alla piattaforma, sia come passeggeri che come autisti;
* MODELLO DI VEICOLO PIÙ UTILIZZATO: visualizza le informazioni del veicolo che è stato utilizzato per il maggior numero di viaggi;
* PASSEGGERI SENZA PRENOTAZIONI: visualizza l’elenco degli utenti iscritti alla piattaforma come passeggeri, ma che non hanno ancora effettuato una prenotazione per un viaggio.

Meritano un’attenzione particolare le funzionalità che riguardano la registrazione di un utente e la gestione della sessione di login.

Nella parte destra del sito sono presenti due link, rispettivamente “*Accedi*” e “*Registrati*”.

“Registrati” permette a un nuovo utente di registrarsi come passeggero attraverso la compilazione di un form, che prevede l’inserimento di nome, cognome, codice fiscale, email, password e numero di telefono. Se i dati sono corretti e completi, il nuovo profilo verrà inserito all’interno del database e si aprirà la pagina di accesso, in cui verranno richiesti email e password per iniziare una nuova sessione come utente; in caso contrario, verranno visualizzati dei messaggi d’errore se la compilazione è incompleta o non rispetta i requisiti.

“Accedi” permette invece ad un utente registrato di effettuare il login alla piattaforma, inserendo le sue credenziali e iniziando quindi una nuova sessione all’interno del sito. Se l’accesso va a buon fine, sarà possibile visualizzare due nuovi elementi nella pagina:

1. Un messaggio di benvenuto all’utente (“Ciao $nome!”) sotto al logo aziendale
2. Una nuova voce nel menù laterale, “*LOGOUT*”, che permetterà all’utente di terminare la sessione corrente e di effettuare la disconnessione dal portale, ritornando alla homepage e visualizzando un messaggio di ringraziamento per l’utilizzo del servizio.

Infine, è possibile ritornare in qualsiasi momento alla pagina principale del sito cliccando sul logo dell’azienda.

1. Note

Strumenti utilizzati: Microsoft Word[[1]](#footnote-2), PostgreSQL[[2]](#footnote-3), Draw.io[[3]](#footnote-4), Notepad++[[4]](#footnote-5), XAMPP5.

1. https://products.office.com/it-it/word [↑](#footnote-ref-2)
2. https://www.postgresql.org/ [↑](#footnote-ref-3)
3. https://www.draw.io/ [↑](#footnote-ref-4)
4. <https://notepad-plus-plus.org/>

   5 <https://www.apachefriends.org/it/index.html> [↑](#footnote-ref-5)