

Corso Basi di Dati
Corso di Laurea Informatica per il Management
RELAZIONE di PROGETTO, A.A. 2015/2016
Web App BolognaGreen

Carlo Fanciulli
0000691815
carlo.fanciulli@studio.unibo.it

Specifiche

Si vuole realizzare una piattaforma a supporto della mobilità sostenibile ed ecologica all'interno della città di Bologna. Tale piattaforma gestisce i dati relativi alla disponibilità di mezzi di trasporto "green" (bici, scooter elettrici, auto elettriche) presenti sul territorio, e fornisce la possibilità ---agli utenti registrati--- di prenotare l'utilizzo di tali mezzi, e di accedere a servizi avanzati in tema di mobilità urbana.

In particolare, la piattaforma gestisce i dati relativi a tre tipologie di mezzi: bici, scooter elettrici ed auto elettriche. Le bici dispongono di: identificativo, marca, anno di acquisizione, colore. Le bici sono poste in postazioni di prelievo dislocate in vari punti della città. Ogni bici è collocata in una postazione; viceversa, ogni postazione di prelievo può contenere più bici. Di ogni postazione si vogliono memorizzare: indirizzo, geolocalizzazione (latitudine, longitudine), numero bici (totale, sia disponibili sia attualmente utilizzate), numero bici attualmente disponibili. Inoltre, si vuole tenere traccia delle piste ciclabili presenti all'interno della città: ogni pista dispone di: numero progressivo, lunghezza (in km), pendenza media, e di un insieme di punti (lat, long) che formano il percorso. Gli scooter e le auto elettriche dispongono di: targa, nome modello, cilindrata, colore, autonomia (numero Km), costo orario del noleggio, foto. Per le auto elettriche, si vogliono memorizzare anche: numero max di passeggeri ed attuale chilometraggio. I veicoli elettrici (scooter ed auto) sono disponibili presso i punti di noleggio, caratterizzati da: nome, indirizzo, recapito telefonico, indirizzo email, sito Web, geolocalizzazione (latitudine, longitudine). Ogni veicolo è associato ad un singolo punto di noleggio.

Oltre ad i veicoli, la piattaforma gestisce i dati degli utenti registrati. Ogni utente dispone di: email (univoca), password, nome, cognome, anno di nascita, luogo di nascita, indirizzo di residenza, recapito telefonico, foto. Inoltre, ogni utente dispone di una INBOX di messaggi di segnalazione, inviati da altri utenti.

Sono presenti tre diverse categorie di utenti: semplici, premium ed amministratori. Gli utenti (semplici/premium) possono prenotare bici o veicoli (auto/scooter) elettrici. Ogni prenotazione è associata ad un utente, ad una bici o a un veicolo, e dispone di una data/orario di inizio ed una data/orario di fine. All'atto della prenotazione online, viene generato un codice univoco della stessa. Nel caso di bici, la prenotazione non può durare più di 12 ore. Nel caso di prenotazioni di veicoli elettrici, si vuole memorizzare anche il prezzo della prenotazione ($\text{prezzo} = \text{numero ore prenotazione} * \text{costo orario del noleggio del veicolo}$). Non è possibile prenotare un mezzo già prenotato/in uso da altri utenti.

Gli utenti premium possono accedere a servizi avanzati nel caso delle auto elettriche. In particolare, essi possono prenotare slot di ricarica presso colonnine elettriche (=stazioni di ricarica) disponibili nella città. Ogni colonnina dispone di: indirizzo, geolocalizzazione (latitudine, longitudine), potenza massima di erogazione (KWh), ente fornitore (es. ENEL). Una prenotazione di ricarica fa riferimento ad una colonnina ed un utente, e dispone di codice univoco, data, slot di inizio, slot di fine; si assume che gli slot abbiano durata di mezz'ora (48 slot per giornata). Non è possibile prenotare uno slot già in uso/prenotato da altri utenti. Inoltre, gli utenti premium possono inserire segnalazioni circa lo stato delle piste ciclabili. Ogni segnalazione fa riferimento ad una pista ciclabile, e dispone di data/ora inserimento, titolo, testo del messaggio, e può contenere una foto.

Gli utenti amministratori possono inserire/cancellare/modificare i dati relativi a mezzi, utenti e prenotazioni. Inoltre, essi possono inserire dei post all'interno di un forum, segnalando eventuali condizioni circa la mobilità urbana (es. "strada X interrotta per lavori"). Ogni post dispone di: data/ora inserimento, titolo e testo del messaggio.

Operazioni principali che **sono** fornite dal sistema informativo:

Profilo utente (semplice):

- Si può visualizzare la lista delle bici, delle postazioni di prelievo di bici con numero di bici totali in ciascuna; fornita la visualizzazione attraverso tabella, e su Google Maps.
- Si possono visualizzare le piste ciclabili presenti sul territorio
- Si può visualizzare la lista dei punti di noleggio dei veicoli, con auto/scooter disponibili in ciascun punto; fornita visualizzazione attraverso tabella, e su Google Maps.
- Si possono visualizzare i vari elementi detti in precedenza posti entro un raggio di x chilometri dalla posizione corrente dell'utente
- Si può prenotare una bici o un veicolo elettrico
- Si può visualizzare le proprie prenotazioni (passate/in corso) con la possibilità di eliminare quelle in corso
- Mandare messaggi personali ad altri utenti e visualizzare i messaggi nella propria INBOX con la possibilità di eliminare i messaggi personali (sia i messaggi personali sia i messaggi che avvertono l'utente di un nuovo post nel forum)
- Visualizzare il forum
- Visualizzare il proprio profilo (con i dati inseriti dall'utente durante la registrazione) con la possibilità di cancellarsi dalla piattaforma.
- Visualizzare la Classifica degli utenti in base al numero di prenotazioni di bici/auto/scooter

Profilo utente (premium):

L'Utente Premium, ovviamente, può usufruire degli stessi servizi dell'utente semplice ma in più può:

- Inserire una segnalazione di una pista ciclabile che poi sarà visibile agli utenti che hanno una prenotazione attiva su di essa
- Visualizzare la lista delle colonnine di ricarica, con la disponibilità oraria di ciascuna (fornendo anche una visualizzazione su Google Maps potendo scegliere il raggio di chilometri dalla posizione corrente dell'utente).
- Prenotare un'operazione di ricarica presso una colonnina.
- Visualizzare dell'utilizzo medio delle colonnine di ricarica (slot occupati/slot totali) attraverso Google Charts.

Profilo amministratore:

- Inserire/visualizzare/cancellare i dati relativi a Bici, Postazioni, Piste Ciclabili, Veicoli, Punti Noleggio, Colonnine
- Possibilità di cancellare tutte le prenotazioni passate e ogni singola prenotazione in corso
- Visualizzazione di messaggi nell'INBOX potendo eliminare ogni singolo messaggio (sia personale sia globale)
- Inserire nuovi post nel forum
- Visualizzazione lista di tutti gli utenti con la possibilità di eliminarli

Tavola dei volumi (stima di entità/relazioni significative ai fini della discussione):

Concetto	Tipo	Volume
Utente	Entità	1000
Mezzo	Entità	500
Punto Noleggio	Entità	15
Postazione Prelievo	Entità	5
Pista Ciclabile	Entità	20
Segnalazione	Entità	50 (In media 1 al giorno)
Post	Entità	20 (In media 1 a settimana)
Messaggio	Entità	250 (In media 30 al giorno)
Prenotazioni Bici	Relazione	50(In media 5 al giorno)
Prenotazioni Veicoli	Relazione	150(In media 20 al giorno)
Prenotazioni Colonnine	Relazione	75(In media 7 al giorno)

Trigger previsti:

- Ogni qualvolta un utente premium inserisce una segnalazione relativa ad una pista ciclabile, viene inviato un messaggio nella INBOX degli utenti che dispongono di prenotazioni attive di bici
- Se un utente semplice effettua più di 30 prenotazioni (di bici o veicoli elettrici) viene automaticamente aggiornato il suo profilo ad utente premium
- Ogni qualvolta un utente amministratore inserisce un messaggio nel forum, viene inviato un messaggio della INBOX di tutti gli utenti, con titolo “Nuova news inserita nel forum”.
- Controllo che la prenotazione di bici non superi le 12 ore
- Controllo che le prenotazioni non siano state già effettuate da qualcun altro

File System Progetto:

Il progetto è suddiviso in diverse cartelle che hanno un diverso scopo. Le principali sono:

- “css”, “fonts”, “img”, “ico” : Cartelle per la gestione dell’interfaccia web (realizzata con bootstrap)
- “function” : Cartella per la realizzazione delle operazioni e della logica svolte nell’interfaccia attraverso il linguaggio php
- “pages”: Cartella che racchiude le pagine web mostrate poi attraverso l’interfaccia
- “js”: Cartella contenente il javascript per gli effetti dinamici interattivi nell’interfaccia web
- “uploads”: Cartella che immagazzina le foto dei veicoli e degli utenti
- “class.user.php”, “dbconfig.php”, “index.php”, “indexsignup.php”, “logout.php” e le varie Home: File principali per il primo interfacciamento con l’utente attraverso un login o una registrazione, la configurazione con il database e un file con le principali funzioni per richiamare poi le stored procedure del file sql.
- “piattaforma.sql”: file sql con la creazione della struttura del database, i vari trigger e stored procedures e infine l’inserimento automatico di vari componenti in uso nella piattaforma (ex. Bici, Veicoli, Prenotazioni)

DIZIONARIO ENTITA’

Entità	Descrizione	Attributi	Identificatore
Bici	Mezzo di trasporto	Id, Postazione Prelievo, Marca, Colore, Anno Acquisizione	Id
Pista Ciclabile	Pista dove usufruire del mezzo	Id, Chilometri, Pendenza Media, Latitudine, Longitudine	Id
Postazione Prelievo	Luogo di prelievo bici	Indirizzo, N° Bici Totale, Latitudine, Longitudine	Indirizzo
Veicolo elettrico	Mezzo di trasporto elettrico	Targa, Nome Modello, Colore, Autonomia km, Costo orario, Cilindrata, Foto	Targa
Auto	Tipo di mezzo elettrico	Max Passeggeri, Chilometraggio Attuale	Targa
Scooter	Tipo di mezzo elettrico		Targa
Punto Noleggio	Luogo di prelievo mezzi elettrici	Nome, Sito Web, Email, Telefono, Indirizzo, Latitudine, Longitudine	Nome
Utente	Persona registrata nella piattaforma	Email, Nome, Cognome, Password, Tipologia, Data	Email

		Nascita, Luogo Nascita, Indirizzo Residenza, Telefono, Foto	
U. Semplice	Utente con accesso servizi base		Email
U. Premium	Utente con accesso servizi avanzati		Email
Colonnina Elettrica	Luogo di ricarica veicoli elettrici	Indirizzo, Ente Fornitore, Max KWH, Data Inserimento, Longitudine, Latitudine	Indirizzo
Segnalazione	Avvertimento diretto a Inbox utenti	Id, Pista Ciclabile, Email Utente, Titolo, Testo Messaggio, Data Inserimento	Id
U. Amministratore	Utente con accesso servizi avanzati e gestione piattaforma		Email
Forum Post	Messaggio dell'amministratore diretto a forum	Id, Email Utente, Titolo, Testo Messaggio, Data Inserimento	Id
Messaggio	Messaggio INBOX che può essere Personale o Globale	Id, Email Mittente, Titolo, Email Destinatario, Tipo, Testo Messaggio, Data Invio	Id

DIZIONARIO RELAZIONI

Relazione	Descrizione	Tipo	Componenti	Attributi
Noleggio veicolo	Associa un veicolo elettrico ad un punto noleggio	1 a molti	Veicolo Elettrico, Punto Noleggio	
Noleggio bici	Associa una bici ad una postazione di prelievo	1 a molti	Bici, Postazione Prelievo	
R1	Associa un utente ad un messaggio	Molti a Molti	Utente, Messaggio	
R2	Associa una segnalazione a una pista	1 a Molti	Segnalazione, Pista Ciclabile	
Prenotazione bici	Associa due tipi di utenti ad una bici	Molti a molti	Utente Semplice, Utente Premium, Bici	Data Inizio, Data Fine, Id
Prenotazione veicolo	Associa due tipi di utenti ad un veicolo	Molti a molti	Utente Semplice, Utente Premium, Veicolo Elettrico	Data Inizio, Data Fine, Prezzo Prenotazione, Id
Prenotazione Colonnina	Associa un utente ad una colonnina elettrica	Molti a molti	Utente Premium, Colonnina Elettrica	Id, Data, Slot_Inizio, Slot_Fine
Inserimento Segnalazione	Associa un utente ad una segnalazione	Molti a molti	Utente Premium, Segnalazione	
Inserimento Post	Associa un utente ad un post	1 a molti	Utente, Forum Post	

Diagramma E-R

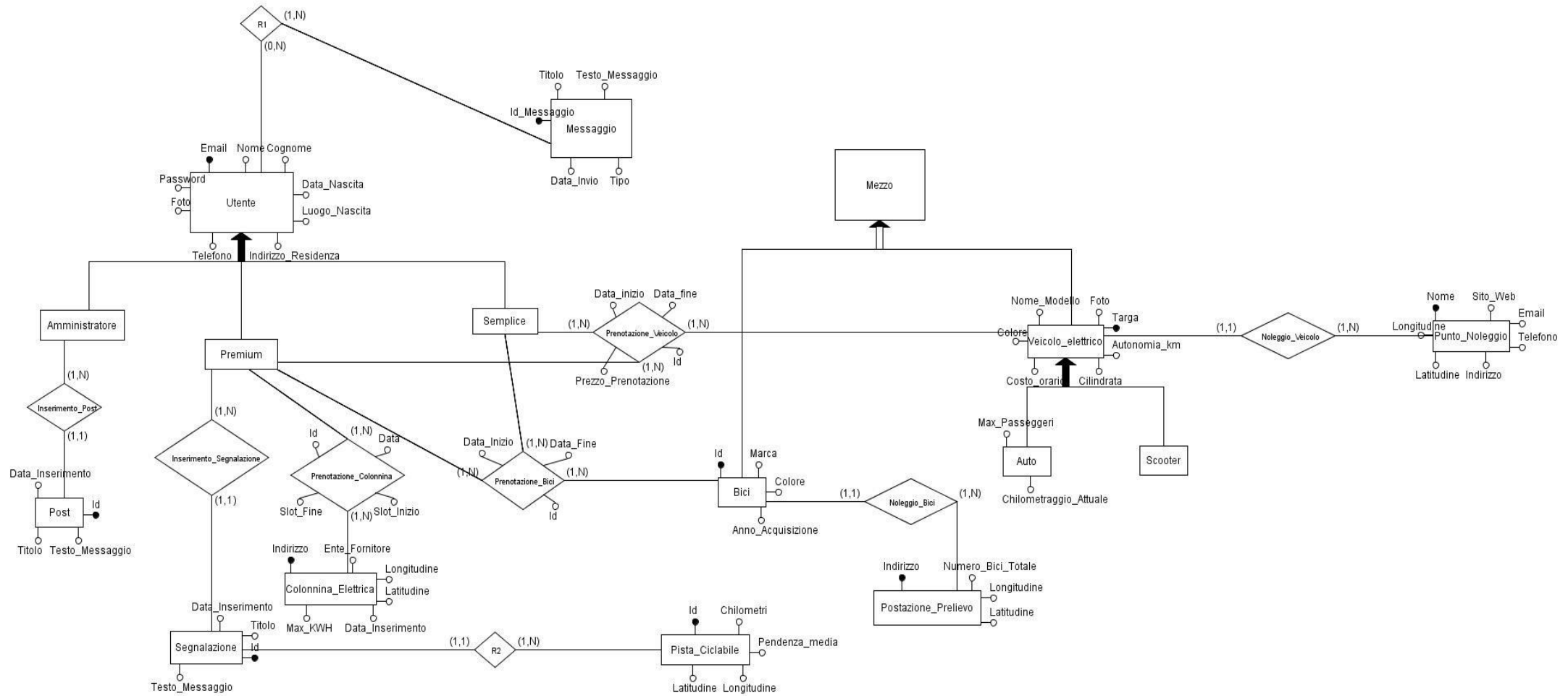


Tabella delle business rules

Regole di vincolo
La prenotazione di bici non può durare più di 12 ore
Non è possibile prenotare un mezzo già prenotato/in uso da altri utenti
Uno slot ha una durata di mezz'ora e slot max=48
Non è possibile prenotare uno slot già in uso/prenotato da altri utenti.
Quando un utente si registra al sistema, la sua categoria viene impostata a "Semplice". Non è possibile registrarsi come utente "Premium" o "Amministratore".
Regole di derivazione
Il prezzo della prenotazione è uguale al numero di ore della prenotazione moltiplicato il costo orario del noleggio del veicolo

Traduzione Diagramma E-R nel modello relazionale:

Veicolo Elettrico (Targa, Punto Noleggio, Tipologia, Nome Modello, Colore, Costo orario, Cilindrata, Autonomia km, Max passeggeri, Chilometraggio Attuale, Foto)

Punto Noleggio (Nome, Sito Web, Email, Telefono, Indirizzo, Latitudine, Longitudine)

Bici (Id, Postazione Prelievo, Marca, Colore, Anno Acquisizione)

Postazione Prelievo (Indirizzo, Numero Bici Totale, Latitudine, Longitudine)

Pista Ciclabile (Id, Chilometri, Pendenza Media, Latitudine, Longitudine)

Utente (Email, Nome, Cognome, Password, Tipologia, Data Nascita, Luogo Nascita, Indirizzo Residenza, Telefono, Foto)

Messaggio (Id_Messaggio, Email Mittente, Email Destinatario, Titolo, Tipo, Testo Messaggio, Data Invio)

Prenotazione Veicolo (Id, EmailUtente, Veicolo, Prezzo Prenotazione, Data Inizio, Data Fine)

Prenotazione Bici (Id, EmailUtente, IdBici, Data Inizio, Data Fine)

ForumPost (Id, EmailUtente, Titolo, Testo Messaggio, Data Inserimento)

Segnalazione (Id, Pista Ciclabile, EmailUtente, Titolo, Testo Messaggio, Data Inserimento)

Colonnina Elettrica (Indirizzo, Ente Fornitore, Max KWH, Data Inserimento, Latitudine, Longitudine)

Prenotazione_Colonnina (Id, EmailUtente, Indirizzo, Slot_Inizio, Slot_Fine, Data Pren)

Analisi Ridondanze:

$$W_b=0,5, W_I=0,5, \alpha=2$$

Prezzo della prenotazione

Con ridondanza: $20 \cdot 0,5 \cdot (2 \cdot 2 + 1) = 50$

Senza ridondanza: $20 \cdot 0,5 \cdot (2 \cdot 1 + 1) = 30$

Note per il docente:

E' importante sottolineare che è stata usata la versione di MySql 5.7.9 per poter eseguire alcune operazioni che altrimenti era impossibile effettuare.

Perciò versioni precedenti potrebbero creare problemi nell'esecuzione del codice sql.

Per accedere al sistema è necessario utilizzare le seguenti credenziali:

Amministratore: email → "admin@gmail.com", password → "admin".

Premium: email → "utente2@gmail.com", password → "password".

Semplice: email → "utente1@gmail.com", password → "password".

Uso dell'applicazione:

La prima operazione per poter usare il sito BolognaGreen è loggarsi al sistema oppure registrarsi. Una volta effettuato il login, nella pagina principale è possibile accedere ai servizi dati dall'applicazione.

Sono previste 3 home principali in base alla tipologia di Utente che abbiamo. (Semplice, Premium, Amministratore) in quanto ogni utente ha accesso a servizi differenti.

All'**utente Semplice** è permesso visualizzare le bici, le postazioni prelievo, le piste ciclabili, i veicoli, i punti noleggio con la possibilità di prenotare bici e veicoli.

Può visualizzare le prenotazioni effettuate nel passato e in corso.

Ha accesso ad una inbox contenente i messaggi personali (cioè quelli scambiati tra utenti) e quelli globali (cioè quelli che arrivano ad un utente ogni qual volta l'admin mette un post nel forum) con la possibilità di inviarne.

Può visualizzare il forum, alcuni tipi di classifiche e il proprio profilo con la possibilità di cancellare l'account che sta usando.

All'**utente Premium** è permesso, oltre ai servizi base già elencati precedentemente, inserire una segnalazione di una pista ciclabile (che manderà in automatico un messaggio personale di avvertimento a chi ha una prenotazione attiva su di essa), visualizzare le colonnine di ricarica con la possibilità di prenotarle.

Questo tipo di utente può visualizzare in aggiunta l'utilizzo medio degli slot delle colonnine di ricarica attraverso visualizzazione con Google Charts.

Ovviamente entrambi i tipi di Utente hanno la capacità di cancellare le proprie prenotazioni in corso nel caso avessero cambiato idea.

All'**utente Amministratore** è permesso inserire o cancellare i dati di bici, postazioni, piste ciclabili, veicoli, punti noleggio, colonnine di ricarica, prenotazioni, messaggi della propria inbox, post del forum e utenti.

Un meccanismo di gestione delle sessioni permette all'utente di navigare tra le pagine rimanendo loggato con l'impossibilità di accedere a pagine e servizi senza permesso. Vengono utilizzate le API di Google per determinare le coordinate dato l'indirizzo dell'utente e per la visualizzazione su Google Maps. HTML Geolocation API è invece utilizzata per determinare le posizione corrente dell'utente.

Lista Query, Trigger e Stored Procedure utilizzati:

Nome	Tipo	Descrizione
AggiornaTipoUtente	Trigger	Aggiornamento tipologia utente se vengono effettuate più di 30 prenotazioni
checkPrenotazione (Bici, Veicoli, Colonnine)	Trigger	Controllo che la prenotazione non sia già effettuata da qualcun'altro
ControlloPrenotazioneBici	Trigger	Controllo che la prenotazione non superi le 12 ore
InserimentoMessaggiInbox	Trigger	Inserimento messaggio in INBOX utenti dopo inserimento post nel forum
<u>ControlloInserimentoPostAmministratore</u>	Trigger	Controllo che la creazione dei post sia fatta solo da utenti amministratori
<u>ControlloInserimentoSegnalazione</u>	Trigger	Controllo che la prenotazione sia fatto solo da utenti Premium
<u>ControlloInserimentoColonnina</u>	Trigger	Controllo che la prenotazione sia fatto solo da utenti Premium
InserimentoMSGUtenticonPrenotazioni	Trigger	Inserimento messaggio personale agli utenti che hanno prenotazioni di bici attive dopo una segnalazione
DatePrenotazioniBici	Stored Procedure	Lista orari occupati di una prenotazione bici (serve per il trigger checkPrenotazioneBici)
DatePrenotazioniVeicoli	Stored Procedure	Lista orari occupati di una prenotazione veicoli (serve per il trigger checkPrenotazioneVeicoli)
SlotPrenotazioniColonnine	Stored Procedure	Lista slot occupati di una prenotazione colonn. (Serve per il trigger checkPrenotazioneColon.)

UtentiConPrenotazioniBici	Stored Procedure	Lista utenti con prenotazione attiva (serve per il trigger InserimentoMSGUtenticonPrenotazioni)
INSERIMENTO (Postazioni, Piste Ciclabili, Punti Noleggio, Prenotazioni, Post, Messaggi, Colonnine)	Stored Procedure	Inserisce gli elementi menzionati tra parentesi
VISUALIZZA (Postazioni, Piste Ciclabili, Punti Noleggio, Prenotazioni, Post, Messaggi, Colonnine)	Stored Procedure	Visualizza gli elementi menzionati tra parentesi
ELIMINA (Postazioni, Piste Ciclabili, Punti Noleggio, Prenotazioni, Post, Messaggi, Colonnine, Utenti, Bici, Veicoli)	Stored Procedure	Elimina gli elementi menzionati tra parentesi
ClassificaPren (Bici e Veicoli)	Stored Procedure	Classifica Utenti in base al numero di prenotazioni (Bici o Veicoli) effettuate
ListaUtenti	Stored Procedure	Ritorna lista utenti (per admin)

Codice SQL completo della base di dati:

```

CREATE DATABASE PIATTAFORMA;
USE PIATTAFORMA;
/* Definizione delle tabelle SQL */
CREATE TABLE IF NOT EXISTS Punto_Noleggio(

Nome VARCHAR(40) NOT NULL PRIMARY KEY,
Sito_Web VARCHAR(50),
Email VARCHAR(40),
Telefono VARCHAR(20),
Indirizzo VARCHAR(40),
Latitudine FLOAT(10,6),
Longitude FLOAT(10,6)
) ENGINE=INNODB;

CREATE TABLE IF NOT EXISTS Veicolo_elettrico(
Targa VARCHAR(20) NOT NULL PRIMARY KEY,
Punto_Noleggio VARCHAR(20) NOT NULL,
Tipologia ENUM('Auto','Scooter') NOT NULL,
Nome_Modello VARCHAR(40),
Colore VARCHAR(40),
Costo_orario INT,
Cilindrata INT,
Autonomia_km INT,
Max_Passeggeri INT,
Chilometraggio_Attuale INT,
Foto VARCHAR(200),
FOREIGN KEY (Punto_Noleggio) REFERENCES Punto_Noleggio(Nome)
ON DELETE CASCADE
ON UPDATE CASCADE
) ENGINE=INNODB;

```

```

CREATE TABLE IF NOT EXISTS Postazione_Prelievo(

Indirizzo VARCHAR(40) NOT NULL PRIMARY KEY,
Numero_Bici_Totale INT,
Latitudine FLOAT(10,6),
Longitudine FLOAT(10,6)
) ENGINE=INNODB;

CREATE TABLE IF NOT EXISTS Bici(

Id INT NOT NULL PRIMARY KEY,
Postazione_Prelievo VARCHAR(20),
Marca VARCHAR(20),
Colore VARCHAR(20),
Anno_Acquisizione INT,
FOREIGN KEY (Postazione_Prelievo) REFERENCES Postazione_Prelievo(Indirizzo)
ON DELETE CASCADE
ON UPDATE CASCADE
) ENGINE=INNODB;

CREATE TABLE IF NOT EXISTS Pista_Ciclabile(

Id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
Chilometri INT,
Pendenza_Media DECIMAL(6,2),
Latitudine FLOAT(10,6),
Longitudine FLOAT(10,6)
) ENGINE=INNODB;

CREATE TABLE IF NOT EXISTS Utente(

Email VARCHAR(20) NOT NULL PRIMARY KEY,
Nome VARCHAR(20) NOT NULL,
Cognome VARCHAR(20) NOT NULL,
password VARCHAR(20) NOT NULL,
Tipologia ENUM('Amministratore','Premium','Semplice') DEFAULT 'Semplice',
Data_Nascita DATE,
Luogo_Nascita VARCHAR(20),
Indirizzo_Residenza VARCHAR(40),
Telefono VARCHAR(20),
Foto VARCHAR(200)
) ENGINE=INNODB;

CREATE TABLE IF NOT EXISTS Messaggio(

Id_Messaggio INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
Email_Mittente VARCHAR(20) NOT NULL,
Titolo VARCHAR(100),
Email_Destinatario VARCHAR(20) NOT NULL,
Tipo ENUM('Personale','Globale'),
Testo_Messaggio VARCHAR(500),
DataInvio DATETIME,
FOREIGN KEY (Email_Mittente) REFERENCES Utente(Email)
ON DELETE CASCADE
ON UPDATE CASCADE,
FOREIGN KEY (Email_Destinatario) REFERENCES Utente(Email)
ON DELETE CASCADE
ON UPDATE CASCADE
) ENGINE=INNODB;

CREATE TABLE IF NOT EXISTS Prenotazione_Veicolo(

```

```

Id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
EmailUtente VARCHAR(20) NOT NULL,
Veicolo VARCHAR(20) NOT NULL,
Prezzo_Prenotazione INT,
Data_Inizio DATETIME,
Data_Fine DATETIME,
FOREIGN KEY (EmailUtente) REFERENCES Utente(Email)
ON DELETE CASCADE
ON UPDATE CASCADE,
FOREIGN KEY (Veicolo) REFERENCES Veicolo_elettrico(Targa)
ON DELETE CASCADE
ON UPDATE CASCADE
) ENGINE=INNODB;

CREATE TABLE IF NOT EXISTS Prenotazione_Bici(
Id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
EmailUtente VARCHAR(20) NOT NULL,
IdBici INT NOT NULL,
Data_Inizio DATETIME,
Data_Fine DATETIME,
FOREIGN KEY (EmailUtente) REFERENCES Utente(Email)
ON DELETE CASCADE
ON UPDATE CASCADE,
FOREIGN KEY (IdBici) REFERENCES Bici(Id)
ON DELETE CASCADE
ON UPDATE CASCADE
) ENGINE=INNODB;

CREATE TABLE IF NOT EXISTS ForumPost(

Id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
EmailUtente VARCHAR(50) NOT NULL,
Titolo VARCHAR(100),
Testo_Messaggio VARCHAR(500),
Data_Inserimento DATETIME,
FOREIGN KEY (EmailUtente) REFERENCES Utente(Email)
ON DELETE CASCADE
ON UPDATE CASCADE
) ENGINE=INNODB;

CREATE TABLE IF NOT EXISTS Segnalazione(

Id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
Pista_Ciclabile INT NOT NULL,
EmailUtente VARCHAR(50) NOT NULL,
Titolo VARCHAR(100),
Testo_Messaggio VARCHAR(500),
Data_Inserimento DATETIME,
FOREIGN KEY (Pista_Ciclabile) REFERENCES Pista_Ciclabile(Id)
ON DELETE CASCADE
ON UPDATE CASCADE,
FOREIGN KEY (EmailUtente) REFERENCES Utente(Email)
ON DELETE CASCADE
ON UPDATE CASCADE
) ENGINE=INNODB;

CREATE TABLE IF NOT EXISTS Colonnina_Elettrica(

Indirizzo VARCHAR(50) NOT NULL PRIMARY KEY,
Ente_Fornitore VARCHAR(20),
Max_KWH INT,
Data_Inserimento DATE,
Latitudine FLOAT(10,6),

```

```

Longitude FLOAT(10,6)
) ENGINE=INNODB;

CREATE TABLE IF NOT EXISTS Prenotazione_Colonnina(

Id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
EmailUtente VARCHAR(20) NOT NULL,
Indirizzo VARCHAR(50) NOT NULL,
Slot_Inizio INT,
Slot_Fine INT,
Data_pren DATE,
FOREIGN KEY (EmailUtente) REFERENCES Utente(Email)
ON DELETE CASCADE
ON UPDATE CASCADE,
FOREIGN KEY (Indirizzo) REFERENCES Colonnina_Elettrica(Indirizzo)
ON DELETE CASCADE
ON UPDATE CASCADE
) ENGINE=INNODB;

/* -----*/
/* -----Definizione TRIGGER----- */

/*Controllo che la prenotazione sia fatta solo da utenti premium*/
DELIMITER *
CREATE TRIGGER controlloInserimentoSegnalazione BEFORE INSERT ON Segnalazione
FOR EACH ROW
BEGIN
DECLARE premium_status VARCHAR(20);
SET premium_status = (SELECT Tipologia FROM Utente WHERE Email =
NEW.EmailUtente);
IF (STRCMP(premium_status,'Premium')not like 0)
THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'Vincolo violato: non sei un utente premium';
END IF;
END
*
DELIMITER ;

/*Controllo che la prenotazione Colonnina sia fatta solo da utenti premium*/
DELIMITER ^
CREATE TRIGGER controlloPrenotazioneColonnina BEFORE INSERT ON
Prenotazione_Colonnina
FOR EACH ROW
BEGIN
DECLARE premium_status VARCHAR(20);
SET premium_status = (SELECT Tipologia FROM Utente WHERE Email =
NEW.EmailUtente);
IF (STRCMP(premium_status,'Premium')not like 0)
THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'Vincolo violato: non sei un utente premium';
END IF;
END
^
DELIMITER ;

/*Inserisco messaggio personale agli utenti che hanno prenotazioni di bici
attive*/

DELIMITER ^
CREATE TRIGGER InserimentoMSGUtenticonPrenotazioni AFTER INSERT ON Segnalazione
FOR EACH ROW

```

```

BEGIN
DECLARE email_dest VARCHAR(20);
DECLARE rowcount INT;
CALL UtentiConPrenotazioniBici();
SET @rowcount = (SELECT COUNT(*) FROM ListaUtentiConPren);
WHILE (@rowcount > 0) DO
SET @email_dest = (SELECT Email FROM ListaUtentiConPren WHERE Id = @rowcount);
INSERT INTO
Messaggio(Email_Mittente,Titolo,Email_Destinatario,Tipo,Testo_Messaggio,DataInvi
o) VALUES
(NEW.EmailUtente,NEW.Titolo,@email_dest,'Personale',NEW.Testo_Messaggio,
NEW.Data_Inserimento);
SET @rowcount = @rowcount -1;
END WHILE;
END
^
DELIMITER ;

/*Controllo che la creazione dei post sia fatta solo da utenti amministratori*/
DELIMITER *
CREATE TRIGGER controlloInserimentoPostAmministratore BEFORE INSERT ON ForumPost
FOR EACH ROW
BEGIN
DECLARE admin_status VARCHAR(20);
SET admin_status = (SELECT Tipologia FROM Utente WHERE Email = NEW.EmailUtente);
IF (STRCMP(admin_status,'Amministratore')not like 0)
THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'Vincolo violato: non sei un utente amministratore';
END IF;
END
*
DELIMITER ;

/*Inserisco messaggio inbox utenti dopo inserimento forum post*/
DELIMITER *
CREATE TRIGGER InserimentoMessaggioInbox AFTER INSERT ON ForumPost
FOR EACH ROW
BEGIN
DECLARE testo VARCHAR(500);
DECLARE dat DATETIME;
DECLARE email VARCHAR(20);
SET @testo = 'Nuova news inserita nel forum';
SET @dat = NEW.Data_Inserimento;
SET @email = NEW.EmailUtente;
SET @tit = NEW.Titolo;
INSERT INTO
Messaggio(Email_Mittente,Titolo,Email_Destinatario,Tipo,Testo_Messaggio,DataInvi
o) VALUES (@email,@tit,'admin@gmail.com','Globale',@testo,@dat);
END
*
DELIMITER ;

/*Controllo che la prenotazione non superi le 12 ore*/
DELIMITER *
CREATE TRIGGER controlloPrenotazioneBici BEFORE INSERT ON Prenotazione_Bici
FOR EACH ROW
BEGIN
DECLARE prov datetime;
SET prov = NEW.Data_Inizio + INTERVAL 12 HOUR;
IF (NEW.Data_Fine > prov)
THEN

```



```

SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'Vincolo violato: non si puo prenotare per piu di 12 ore';
END IF;
END
*
DELIMITER ;

/*Controllo che la prenotazione non sia già effettuata da qualcun'altro*/
/*trigger che segue il trigger precedente, entrambi effettuati prima di un
inserimento in prenotazione bici*/
DELIMITER *
CREATE TRIGGER checkPrenotazioneBici BEFORE INSERT ON Prenotazione_Bici
FOR EACH ROW FOLLOWS controlloPrenotazioneBici
BEGIN
CALL DatePrenotazioniBici(NEW.IdBici);
SET @righecount = (SELECT COUNT(Id) FROM ListaOrariBiciOccupate);

WHILE (@righecount > 0) DO
SET @dataInizio = (SELECT Data_Inizio FROM ListaOrariBiciOccupate WHERE Id=
@righecount);
SET @dataFine = (SELECT Data_Fine FROM ListaOrariBiciOccupate WHERE Id=
@righecount);
IF (( NEW.Data_Inizio BETWEEN @dataInizio AND @dataFine) OR (NEW.Data_Fine
BETWEEN @dataInizio AND @dataFine))
THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'Vincolo violato: non prenotabile';
END IF;
SET @righecount = @righecount -1;
END WHILE;
END
*
DELIMITER ;

/*Controllo che la prenotazione non sia già effettuata da qualcun'altro*/
DELIMITER *
CREATE TRIGGER checkPrenotazioneVeicoli BEFORE INSERT ON Prenotazione_Veicolo
FOR EACH ROW
BEGIN
CALL DatePrenotazioniVeicoli(NEW.Veicolo);
SET @righecount = (SELECT COUNT(Id) FROM ListaOrariVeicoliOccupati);

WHILE (@righecount > 0) DO
SET @dataInizio = (SELECT Data_Inizio FROM ListaOrariVeicoliOccupati WHERE Id=
@righecount);
SET @dataFine = (SELECT Data_Fine FROM ListaOrariVeicoliOccupati WHERE Id=
@righecount);
IF (( NEW.Data_Inizio BETWEEN @dataInizio AND @dataFine) OR (NEW.Data_Fine
BETWEEN @dataInizio AND @dataFine))
THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'Vincolo violato: non prenotabile';
END IF;
SET @righecount = @righecount -1;
END WHILE;
END
*
DELIMITER ;

/*Controllo che la prenotazione non sia già effettuata da qualcun'altro*/
DELIMITER *

```

```

CREATE TRIGGER checkPrenotazioneColonnine BEFORE INSERT ON
Prenotazione_Colonnina
FOR EACH ROW
BEGIN
CALL SlotPrenotazioniColonnine(NEW.Data_pren);
SET @righecount = (SELECT COUNT(Id) FROM ListaColOccupate);
WHILE (@righecount > 0) DO
SET @slotInizio = (SELECT Slot_Inizio FROM ListaColOccupate WHERE Id=
@righecount);
SET @slotFine = (SELECT Slot_Fine FROM ListaColOccupate WHERE Id= @righecount);
IF (( NEW.Slot_Inizio BETWEEN @slotInizio AND @slotFine) OR (NEW.Slot_Fine
BETWEEN @slotInizio AND @slotFine))
THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'Vincolo violato: slot occupati';
END IF;
SET @righecount = @righecount -1;
END WHILE;
END
*

DELIMITER ;
/* Aggiornamento tipo Utente*/
DELIMITER $
CREATE TRIGGER AggiornaTipoUtente1 AFTER INSERT ON Prenotazione_Bici
FOR EACH ROW
BEGIN
DECLARE num_prenotazioni_bici INT;
DECLARE num_prenotazioni_veicoli INT;
SET num_prenotazioni_bici = (SELECT COUNT(*) FROM Prenotazione_Bici WHERE
EmailUtente = NEW.EmailUtente);
SET num_prenotazioni_veicoli = (SELECT COUNT(*) FROM Prenotazione_Veicolo WHERE
EmailUtente = NEW.EmailUtente);
IF( num_prenotazioni_bici+num_prenotazioni_veicoli >= 30)
THEN
UPDATE Utente SET Tipologia='Premium' WHERE Email = NEW.EmailUtente;
END IF;
END;
$
DELIMITER ;

/* Aggiornamento tipo Utente*/
DELIMITER $
CREATE TRIGGER AggiornaTipoUtente2 AFTER INSERT ON Prenotazione_Veicolo
FOR EACH ROW
BEGIN
DECLARE num_prenotazioni_bici INT;
DECLARE num_prenotazioni_veicoli INT;
SET num_prenotazioni_bici = (SELECT COUNT(*) FROM Prenotazione_Bici WHERE
EmailUtente = NEW.EmailUtente);
SET num_prenotazioni_veicoli = (SELECT COUNT(*) FROM Prenotazione_Veicolo WHERE
EmailUtente = NEW.EmailUtente);
IF( num_prenotazioni_bici+num_prenotazioni_veicoli >= 30)
THEN
UPDATE Utente SET Tipologia='Premium' WHERE Email = NEW.EmailUtente;
END IF;
END;
$
DELIMITER ;

/* -----STORED PROCEDURES-----*/
/*-----table temporanea per il trigger checkPrenotazioneColonnine----*/
DELIMITER ^

```

```

CREATE PROCEDURE SlotPrenotazioniColonnine(IN datap DATE)
BEGIN
SET @n = 0;
DROP TEMPORARY TABLE IF EXISTS ListaColOccupate;
CREATE TEMPORARY TABLE ListaColOccupate SELECT @n := @n + 1 AS
Id,Slot_Inizio,Slot_Fine
FROM Prenotazione_Colonnina , (SELECT @n := 0) m
WHERE Data_pren = datap;
END;
^

DELIMITER ;
/*-----table temporanea per il trigger checkPrenotazioneBici-----*/
DELIMITER ^
CREATE PROCEDURE DatePrenotazioniBici(IN Id INT)
BEGIN
SET @n = 0;
DROP TEMPORARY TABLE IF EXISTS ListaOrariBiciOccupate;
CREATE TEMPORARY TABLE ListaOrariBiciOccupate SELECT @n := @n + 1 AS
Id,Data_Inizio,Data_Fine
FROM Prenotazione_Bici , (SELECT @n := 0) m
WHERE IdBici = Id;
END;
^

DELIMITER ;
/*-----table temporanea per il trigger checkPrenotazioneVeicoli-----*/
DELIMITER ^
CREATE PROCEDURE DatePrenotazioniVeicoli(IN TargaVeicolo VARCHAR(30))
BEGIN
SET @n = 0;
DROP TEMPORARY TABLE IF EXISTS ListaOrariVeicoliOccupati;
CREATE TEMPORARY TABLE ListaOrariVeicoliOccupati SELECT @n := @n + 1 AS
Id,Data_Inizio,Data_Fine
FROM Prenotazione_Veicolo , (SELECT @n := 0) m
WHERE Veicolo = TargaVeicolo;
END;
^

DELIMITER ;
/*-----table temporanea per il trigger InserimentoMSGUtenticonPrenotazioni-----*/
DELIMITER ^
CREATE PROCEDURE UtentiConPrenotazioniBici()
BEGIN
SET @n = 0;
DROP TEMPORARY TABLE IF EXISTS ListaUtentiConPren;
CREATE TEMPORARY TABLE ListaUtentiConPren SELECT DISTINCT @n := @n + 1 AS Id
,Email
FROM Utente,Prenotazione_Bici, (SELECT @n := 0) m
WHERE ((Utente.Email=Prenotazione_Bici.EmailUtente) AND
(Prenotazione_Bici.Data_Inizio > now()));
END;
^

DELIMITER ;

/*----- VISUALIZZA POSTAZIONI-----*/
DELIMITER ^
CREATE PROCEDURE VisualizzaPostazioni()
BEGIN
SET AUTOCOMMIT = 0;
START TRANSACTION;
SELECT Indirizzo,Numero_Bici_Totale,Latitudine,Longitudine FROM
Postazione_Prelievo;
COMMIT;
END;
^

```

```

DELIMITER ;

/*----- VISUALIZZA PISTE CICLABILI-----*/
DELIMITER ^
CREATE PROCEDURE VisualizzaPisteCiclabili()
BEGIN
SET AUTOCOMMIT = 0;
START TRANSACTION;
SELECT Id,Chilometri,Pendenza_Media,Latitudine,Longitudine FROM Pista_Ciclabile;
COMMIT;
END;
^
DELIMITER ;

/*----- VISUALIZZA PUNTI NOLEGGIO-----*/

DELIMITER ^
CREATE PROCEDURE VisualizzaPuntiNoleggio()
BEGIN
CREATE OR REPLACE VIEW ScooterinNoleggioParziale AS
SELECT Punto_Noleggio.Nome,COUNT(*) AS Numero_Scooter
FROM Veicolo_elettrico,Punto_Noleggio
WHERE Punto_Noleggio.Nome = Veicolo_elettrico.Punto_Noleggio AND
Veicolo_elettrico.Tipologia ='Scooter'
GROUP BY Punto_Noleggio.Nome;

CREATE OR REPLACE VIEW AutoinNoleggioParziale AS
SELECT Punto_Noleggio.Nome,COUNT(*) AS Numero_Auto
FROM Veicolo_elettrico,Punto_Noleggio
WHERE Punto_Noleggio.Nome = Veicolo_elettrico.Punto_Noleggio AND
Veicolo_elettrico.Tipologia ='Auto'
GROUP BY Punto_Noleggio.Nome;

CREATE OR REPLACE VIEW ScooterInNoleggio AS
SELECT Punto_Noleggio.Nome, ScooterinNoleggioParziale.Numero_Scooter
FROM Punto_Noleggio
LEFT OUTER JOIN ScooterinNoleggioParziale ON Punto_Noleggio.Nome =
ScooterinNoleggioParziale.Nome;

CREATE OR REPLACE VIEW AutoInNoleggio AS
SELECT Punto_Noleggio.Nome,AutoinNoleggioParziale.Numero_Auto
FROM Punto_Noleggio
LEFT OUTER JOIN AutoinNoleggioParziale ON Punto_Noleggio.Nome =
AutoinNoleggioParziale.Nome;

SELECT ScooterInNoleggio.Nome, IFNULL(ScooterInNoleggio.Numero_Scooter,0) AS
Numero_Scooter, IFNULL(AutoInNoleggio.Numero_Auto,0) AS Numero_Auto
FROM ScooterInNoleggio
INNER JOIN AutoInNoleggio
ON ScooterInNoleggio.Nome=AutoInNoleggio.Nome;
END;
^
DELIMITER ;

/*----- VISUALIZZAPRENOTAZIONINCORSO/PASSATE-----*/
DELIMITER ^
CREATE PROCEDURE VisualizzaPrenotazioninCorso(IN Email VARCHAR(50))
BEGIN
SELECT * FROM Prenotazione_Bici WHERE ((EmailUtente = Email) AND (Data_Inizio >
NOW())) ;
SELECT * FROM Prenotazione_Veicolo WHERE ((EmailUtente = Email) AND (Data_Inizio
> NOW())) ;

```

```

SELECT * FROM Prenotazione_Colonnina WHERE ((EmailUtente = Email) AND (Data_pren
> NOW())) ;
END;
^

DELIMITER ;
/*-----*/
DELIMITER ^
CREATE PROCEDURE VisualizzaPrenotazioniPassate(IN Email VARCHAR(50))
BEGIN
SELECT * FROM Prenotazione_Bici WHERE ((EmailUtente = Email) AND (Data_Inizio <
NOW())) ;
SELECT * FROM Prenotazione_Veicolo WHERE ((EmailUtente = Email) AND (Data_Inizio
< NOW())) ;
SELECT * FROM Prenotazione_Colonnina WHERE ((EmailUtente = Email) AND (Data_pren
< NOW())) ;
END;
^

DELIMITER ;
/*-----per l'admin-----*/
DELIMITER ^
CREATE PROCEDURE VisualizzaPrenotazioniTotaliPassate()
BEGIN
SELECT * FROM Prenotazione_Bici WHERE (Data_Inizio < NOW());
SELECT * FROM Prenotazione_Veicolo WHERE (Data_Inizio < NOW());
SELECT * FROM Prenotazione_Colonnina WHERE (Data_pren < NOW());
END;
^

DELIMITER ;
/*-----per l'admin-----*/
DELIMITER ^
CREATE PROCEDURE VisualizzaPrenotazioniTotaliinCorso()
BEGIN
SELECT * FROM Prenotazione_Bici WHERE (Data_Inizio > NOW()) ;
SELECT * FROM Prenotazione_Veicolo WHERE (Data_Inizio > NOW()) ;
SELECT * FROM Prenotazione_Colonnina WHERE (Data_pren > NOW()) ;
END;
^

DELIMITER ;
/*-----*/
/*----- VISUALIZZAPOSTFORUM-----*/
DELIMITER ^
CREATE PROCEDURE VisualizzaForum()
BEGIN
SELECT * FROM ForumPost
ORDER BY Data_Inserimento DESC;
END;
^

DELIMITER ;
/*-----*/
/*----- VISUALIZZAINBOX-----*/
DELIMITER ^
CREATE PROCEDURE VisualizzaINBOX(IN Email VARCHAR(50))
BEGIN
SELECT
Id_Messaggio,Email_Mittente,Titolo,Email_Destinatario,Tipo,Testo_Messaggio,DataI
nvio
FROM Messaggio WHERE (Email_Destinatario = Email) OR (Tipo = 'Globale')
ORDER BY DataInvio DESC;
END;
^

DELIMITER;

```

```

/*-----*/
/*----- VISUALIZZACOLONNINE-----*/

DELIMITER ^
CREATE PROCEDURE VisualizzaColonnine()
BEGIN
SELECT * FROM Colonnina_Elettrica;
END;
^

DELIMITER ;
/*-----*/
/*----- ELIMINAMSGINBOX-----*/
DELIMITER ^
CREATE PROCEDURE EliminaINBOX(IN Id VARCHAR(50))
BEGIN
SET AUTOCOMMIT = 0;
START TRANSACTION;
DELETE FROM Messaggio WHERE (Id_Messaggio = Id);
COMMIT;
END;
^

DELIMITER ;
/*-----*/
/*----- ELIMINAPOST-----*/
DELIMITER ^
CREATE PROCEDURE EliminaPOST(IN IdInserito VARCHAR(50))
BEGIN
SET AUTOCOMMIT = 0;
START TRANSACTION;
DELETE FROM ForumPost WHERE (Id = IdInserito);
COMMIT;
END;
^

DELIMITER ;
/*-----*/
/*----- ELIMINAUTENTE-----*/
DELIMITER ^
CREATE PROCEDURE EliminaUTENTE(IN EmailInserito VARCHAR(50))
BEGIN
SET AUTOCOMMIT = 0;
START TRANSACTION;
DELETE FROM Utente WHERE (Email = EmailInserito);
COMMIT;
END;
^

DELIMITER ;
/*-----*/
/*----- ELIMINAVEICOLO-----*/
DELIMITER ^
CREATE PROCEDURE EliminaVEICOLO(IN TargaInserita VARCHAR(50))
BEGIN
SET AUTOCOMMIT = 0;
START TRANSACTION;
DELETE FROM Veicolo_Elettrico WHERE (Targa = TargaInserita);
COMMIT;
END;
^

DELIMITER ;
/*----- ELIMINABICI-----*/
DELIMITER ^
CREATE PROCEDURE EliminaBICI(IN IdInserito INT)
BEGIN
SET AUTOCOMMIT = 0;

```

```

START TRANSACTION;
DELETE FROM Bici WHERE (Id = IdInserito);
COMMIT;
END;
^

DELIMITER ;
/*----- ELIMINACOLONNINA-----*/
DELIMITER ^
CREATE PROCEDURE EliminaCOLONNINA(IN Ind VARCHAR(40))
BEGIN
SET AUTOCOMMIT = 0;
START TRANSACTION;
DELETE FROM Colonnina_Elettrica WHERE (Indirizzo = Ind);
COMMIT;
END;
^

DELIMITER ;
/*----- ELIMINAPOSTAZIONE-----*/
DELIMITER ^
CREATE PROCEDURE EliminaPOSTAZIONE(IN Ind VARCHAR(40))
BEGIN
SET AUTOCOMMIT = 0;
START TRANSACTION;
DELETE FROM Postazione_Prelievo WHERE (Indirizzo = Ind);
COMMIT;
END;
^

DELIMITER ;
/*----- ELIMINAPISTA-----*/
DELIMITER ^
CREATE PROCEDURE EliminaPISTA(IN IdIns INT)
BEGIN
SET AUTOCOMMIT = 0;
START TRANSACTION;
DELETE FROM Pista_Ciclabile WHERE (Id = IdIns);
COMMIT;
END;
^

DELIMITER ;
/*----- ELIMINAPUNTONOLEGGIO-----*/
DELIMITER ^
CREATE PROCEDURE EliminaPUNTONOLEGGIO(IN NomeInserito VARCHAR(40))
BEGIN
SET AUTOCOMMIT = 0;
START TRANSACTION;
DELETE FROM Punto_Noleggior WHERE (Nome = NomeInserito);
COMMIT;
END;
^

DELIMITER ;
/*----- ELIMINAPRENINCORSO BICI-----*/
DELIMITER ^
CREATE PROCEDURE EliminaPrenInCorsoBici(IN IdInserito INT)
BEGIN
SET AUTOCOMMIT = 0;
START TRANSACTION;
DELETE FROM Prenotazione_Bici WHERE (Id= IdInserito);
COMMIT;
END;
^

DELIMITER ;
/*----- ELIMINAPRENINCORSO VEICOLO-----*/
DELIMITER ^

```

```

CREATE PROCEDURE EliminaPrenInCorsoVeicoli(IN IdInserito INT)
BEGIN
SET AUTOCOMMIT = 0;
START TRANSACTION;
DELETE FROM Prenotazione_Veicolo WHERE (Id=IdInserito);
COMMIT;
END;
^

DELIMITER ;
/*----- ELIMINAPRENINCORSO COLONNINA-----*/
DELIMITER ^
CREATE PROCEDURE EliminaPrenInCorsoColonnina(IN IdInserito INT)
BEGIN
SET AUTOCOMMIT = 0;
START TRANSACTION;
DELETE FROM Prenotazione_Colonnina WHERE (Id=IdInserito);
COMMIT;
END;
^

DELIMITER ;
/*----- ELIMINAZIONE PRENOTAZIONI PASSATE (ADMIN) -----*/
DELIMITER ^
CREATE PROCEDURE EliminaPrenotazioni()
BEGIN
SET AUTOCOMMIT = 0;
START TRANSACTION;
DELETE FROM Prenotazione_Bici WHERE (Data_Inizio < NOW());
DELETE FROM Prenotazione_Veicolo WHERE (Data_Inizio < NOW());
DELETE FROM Prenotazione_Colonnina WHERE (Data_pren < NOW());
COMMIT;
END;
^

DELIMITER ;
/*----- inserimento Prenotazioni BICI e VEICOLI-----*/
DELIMITER ^
CREATE PROCEDURE PrenotazioneBici(IN EmailUtente VARCHAR(50), IN IdBici INT, IN
Data_Inizio DATETIME, IN Data_Fine DATETIME)
BEGIN
SET AUTOCOMMIT = 0;
START TRANSACTION;
IF(Data_Fine > Data_Inizio) THEN
INSERT INTO Prenotazione_Bici(EmailUtente, IdBici, Data_Inizio, Data_Fine) VALUES
(EmailUtente, IdBici, Data_Inizio, Data_Fine);
ELSE
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'Vincolo violato: non prenotabile';
END IF;
COMMIT;
END;
^

DELIMITER ;

DELIMITER ^
CREATE PROCEDURE PrenotazioneVeicolo(IN EmailUtente VARCHAR(50), IN TargaVeicolo
VARCHAR(50), IN Data_Inizio DATETIME, IN Data_Fine DATETIME)
BEGIN
DECLARE prezzo INT;
DECLARE datadiff INT;
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET prezzo = ((SELECT Costo_orario FROM Veicolo_elettrico WHERE Targa =
TargaVeicolo)) * (SELECT HOUR(TIMEDIFF(Data_Fine, Data_Inizio)));
IF(Data_Fine > Data_Inizio) THEN

```



```

INSERT INTO
Prenotazione_Veicolo(EmailUtente,Veicolo,Data_Inizio,Data_Fine,Prezzo_Prenotazio
ne) VALUES (EmailUtente,TargaVeicolo,Data_Inizio,Data_Fine,prezzo);
ELSE
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'Vincolo violato: non prenotabile';
END IF;
COMMIT;
END;
^

DELIMITER ;

/*----- Inserimento SEGNALAZIONE-----*/

DELIMITER ^
CREATE PROCEDURE InserimentoSegnalazione(IN Pista_Ciclabile INT,IN EmailUtente
VARCHAR(50),IN Titolo VARCHAR(100),IN Testo_Messaggio VARCHAR(500))
BEGIN

DECLARE timenow DATETIME;
SET timenow = NOW();
SET AUTOCOMMIT = 0;
START TRANSACTION;
INSERT INTO
Segnalazione(Pista_Ciclabile,EmailUtente,Titolo,Testo_Messaggio,Data_Inserimento
) VALUES (Pista_Ciclabile,EmailUtente,Titolo,Testo_Messaggio,timenow);
COMMIT;
END;
^

DELIMITER ;

/*----- inserimento PRENOTAZIONE COLONNINA-----*/
DELIMITER ^
CREATE PROCEDURE PrenotazioneColonnina(IN EmailUtente VARCHAR(50),IN Indirizzo
VARCHAR(50),IN Slot_Inizio INT,IN Slot_Fine INT,IN Data_pren DATE)
BEGIN
SET AUTOCOMMIT = 0;
START TRANSACTION;
IF(Slot_Fine > Slot_Inizio) THEN
INSERT INTO
Prenotazione_Colonnina(EmailUtente,Indirizzo,Slot_Inizio,Slot_Fine,Data_pren)
VALUES (EmailUtente,Indirizzo,Slot_Inizio,Slot_Fine,Data_pren);
ELSE
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'Vincolo violato: non prenotabile';
END IF;
COMMIT;
END;
^

DELIMITER ;

/*-----inserimento POST-----*/
DELIMITER ^
CREATE PROCEDURE InserimentoPost(IN EmailUtente VARCHAR(50),IN Titolo
VARCHAR(100),IN Testo_Messaggio VARCHAR(500))
BEGIN
DECLARE timenow DATETIME;
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET timenow = NOW();
INSERT INTO ForumPost(EmailUtente,Titolo,Testo_Messaggio,Data_Inserimento)
VALUES (EmailUtente,Titolo,Testo_Messaggio,timenow);
COMMIT;
END;
^

DELIMITER ;

```

```

/*-----inserimento NUOVA BICI-----*/
DELIMITER ^
CREATE PROCEDURE InserimentonewBici(IN Id INT , IN Postazione VARCHAR(20), IN
Marca VARCHAR(20), IN Colore VARCHAR(20), IN Anno INT)
BEGIN
SET AUTOCOMMIT = 0;
START TRANSACTION;
INSERT INTO Bici(Id,Postazione_Prelievo,Marca,Colore,Anno_Acquisizione) VALUES
(Id,Postazione,Marca,Colore,Anno);
COMMIT;
END;
^

DELIMITER ;
/*-----inserimento NUOVO VEICOLO-----*/
DELIMITER ^
CREATE PROCEDURE InserimentonewVeicolo(IN Targa VARCHAR(20), IN PuntoNoleggio
VARCHAR(20) , IN Tipologia VARCHAR(20), IN NomeModello VARCHAR(40), IN Colore
VARCHAR(40), IN CostoH INT, IN Cilindrata INT, IN AutonomiaKm INT, IN MaxPasseggeri
INT, IN Km_Attuale INT, IN Foto BLOB)
BEGIN
SET AUTOCOMMIT = 0;
START TRANSACTION;
INSERT INTO
Veicolo_elettrico(Targa,Punto_Noleggio,Tipologia,Nome_Modello,Colore,Costo_orari
o,Cilindrata,Autonomia_km,Max_Passeggeri,Chilometraggio_Attuale,Foto) VALUES
(Targa,PuntoNoleggio,Tipologia,NomeModello,Colore,CostoH,Cilindrata,AutonomiaKm,
MaxPasseggeri,Km_Attuale,Foto);
COMMIT;
END;
^

DELIMITER ;
/*-----inserimento NUOVA POSTAZIONE-----*/
DELIMITER ^
CREATE PROCEDURE InserimentonewPostazione(IN Indirizzo VARCHAR(40), IN
Numero_Bici_Totale INT, IN Numero_Bici_Disponibili INT, IN Latitudine
FLOAT(10,6), IN Longitudine FLOAT(10,6))
BEGIN
SET AUTOCOMMIT = 0;
START TRANSACTION;
INSERT INTO
Postazione_Prelievo(Indirizzo,Numero_Bici_Totale,Numero_Bici_Disponibili,Latitudi
ne,Longitudine) VALUES
(Indirizzo,Numero_Bici_Totale,Numero_Bici_Disponibili,Latitudine,Longitudine);
COMMIT;
END;
^

DELIMITER ;
/*-----inserimento NUOVA PISTA CICLABILE-----*/
DELIMITER ^
CREATE PROCEDURE InserimentonewPistaCiclabile(IN Chilometri INT, IN
Pendenza_Media DECIMAL(6,4), IN Latitudine FLOAT(10,6), IN Longitudine
FLOAT(10,6))
BEGIN
SET AUTOCOMMIT = 0;
START TRANSACTION;
INSERT INTO Pista_Ciclabile(Chilometri,Pendenza_Media,Latitudine,Longitudine)
VALUES (Chilometri,Pendenza_Media,Latitudine,Longitudine);
COMMIT;
END;
^

DELIMITER ;
/*-----inserimento NUOVO PUNTO NOLEGGIO-----*/
DELIMITER ^

```

```

CREATE PROCEDURE InserimentonewPuntoNoleggio(IN Nome VARCHAR(20), IN Sito_Web
VARCHAR(50), IN Email VARCHAR(40), IN Telefono VARCHAR(20), IN Indirizzo
VARCHAR(40), IN Latitudine FLOAT(10,6), IN Longitudine FLOAT(10,6))
BEGIN
SET AUTOCOMMIT = 0;
START TRANSACTION;
INSERT INTO
Punto_Noleggio(Nome,Sito_Web,Email,Telefono,Indirizzo,Latitudine,Longitudine)
VALUES (Nome,Sito_Web,Email,Telefono,Indirizzo,Latitudine,Longitudine);
COMMIT;
END;
^

DELIMITER ;
/*-----inserimento NUOVA COLONNINA RICARICA-----*/
DELIMITER ^
CREATE PROCEDURE InserimentonewColonninaRicarica(IN Indirizzo VARCHAR(40), IN
Ente_Fornitore VARCHAR(20), IN Max_KWH INT, IN Data_Inserimento DATE, IN
Latitudine FLOAT(10,6), IN Longitudine FLOAT(10,6))
BEGIN
SET AUTOCOMMIT = 0;
START TRANSACTION;
INSERT INTO
Colonnina_Elettrica(Indirizzo,Ente_Fornitore,Max_KWH,Data_Inserimento,Latitudine
,Longitudine) VALUES
(Indirizzo,Ente_Fornitore,Max_KWH,Data_Inserimento,Latitudine,Longitudine);
COMMIT;
END;
^

DELIMITER ;
/*----- CLASSIFICA UTENTI ORDER BY Numero PRENOTAZIONI BICI-----*/
DELIMITER ^
CREATE PROCEDURE ClassificaPrenBici()
BEGIN
DROP TABLE IF EXISTS PrenBici;
CREATE TEMPORARY TABLE PrenBici
SELECT EmailUtente,COUNT(*) AS Numero_Prenotazioni_Bici FROM Prenotazione_Bici
GROUP BY EmailUtente;

SELECT * from PrenBici ORDER BY Numero_Prenotazioni_Bici DESC;
END;
^

DELIMITER ;
/*----- CLASSIFICA UTENTI ORDER BY Numero PRENOTAZIONI Veicoli-----*/
DELIMITER ^
CREATE PROCEDURE ClassificaPrenVeicoli()
BEGIN
DROP TABLE IF EXISTS PrenVeicoli;
CREATE TEMPORARY TABLE PrenVeicoli
SELECT EmailUtente,COUNT(*) AS Numero_Prenotazioni_Veicoli FROM
Prenotazione_Veicolo GROUP BY EmailUtente;

SELECT * FROM PrenVeicoli ORDER BY Numero_Prenotazioni_Veicoli DESC;
END;
^

DELIMITER ;

```

```

/*----- LISTA UTENTI-----*/
DELIMITER ^
CREATE PROCEDURE ListaUtenti()
BEGIN
SELECT
Email, Nome, Cognome, password, Tipologia, Data_Nascita, Luogo_Nascita, Indirizzo_Resid
enza, Telefono
FROM Utente;

END;
^
DELIMITER ;

/* -----*/
/* -----Inserimento dati nelle tabelle-----*/

INSERT INTO
Utente(Email, Nome, Cognome, password, Tipologia, Data_Nascita, Luogo_Nascita, Indirizz
o_Residenza, Telefono, Foto) VALUES
('utente1@gmail.com', 'Franco', 'Battelli', 'password', 'Semplice', '1989-12-
12', 'Monteparco', 'via santoparco', 331231231, 'utente1.jpg');
INSERT INTO
Utente(Email, Nome, Cognome, password, Tipologia, Data_Nascita, Luogo_Nascita, Indirizz
o_Residenza, Telefono, Foto) VALUES
('utente2@gmail.com', 'Pippo', 'Peppo', 'password', 'Premium', '1993-05-
11', 'Bononia', 'via ciaciao 6', 323333222, 'utente2.jpg');
INSERT INTO
Utente(Email, Nome, Cognome, password, Tipologia, Data_Nascita, Luogo_Nascita, Indirizz
o_Residenza, Telefono, Foto) VALUES
('utente3@gmail.com', 'ProvaNome', 'ProvaCognome', 'password', 'Premium', '1992-11-
11', 'Tralalla', 'via saluti 9', 323333222, 'utente3.jpg');
INSERT INTO
Utente(Email, Nome, Cognome, password, Tipologia, Data_Nascita, Luogo_Nascita, Indirizz
o_Residenza, Telefono, Foto) VALUES
('admin@gmail.com', 'Zeus', 'Dei', 'admin', 'Amministratore', '1969-05-
01', 'Olimpo', 'via degli dei 66', 12345678, 'zeus.jpg');

INSERT INTO
Punto_Noleggio(Nome, Sito_Web, Email, Telefono, Indirizzo, Latitudine, Longitudine)
VALUES ('Hertz', 'www.hertz.it', 'hertz@gmail.com', 3441908587, 'Via Boldrini,
4', 44.491131, 11.335640);
INSERT INTO
Punto_Noleggio(Nome, Sito_Web, Email, Telefono, Indirizzo, Latitudine, Longitudine)
VALUES ('Car Italy', 'www.caritaly.it', 'caritaly@gmail.com', 0515876770, 'Via
Milazzo 12/A', 44.486025, 11.347772);
INSERT INTO
Punto_Noleggio(Nome, Sito_Web, Email, Telefono, Indirizzo, Latitudine, Longitudine)
VALUES ('Avis', 'www.avisautonoleggio.it', 'avis@gmail.com', 0516341632, 'Via Nicolo
Dall Arca 2', 44.506678, 11.330749);

INSERT INTO
Veicolo_elettrico(Targa, Punto_Noleggio, Tipologia, Nome_Modello, Colore, Costo_orari
o, Cilindrata, Autonomia_km, Max_Passeggeri, Chilometraggio_Attuale, Foto) VALUES
('123ABC456', 'Hertz', 'Auto', 'Renault
Zoe', 'Grigio', 25, 88, 125, 4, 12500, 'zoe1.jpg');
INSERT INTO
Veicolo_elettrico(Targa, Punto_Noleggio, Tipologia, Nome_Modello, Colore, Costo_orari
o, Cilindrata, Autonomia_km, Max_Passeggeri, Chilometraggio_Attuale, Foto) VALUES
('4BCG554KS', 'Hertz', 'Scooter', 'Volt City60', 'Rosso', 15, 43, 99, NULL, NULL, 'City60-
volt1.png');

```

```

INSERT INTO
Veicolo_elettrico(Targa,Punto_Noleggior,Tipologia,Nome_Modello,Colore,Costo_orari
o,Cilindrata,Autonomia_km,Max_Passeggeri,Chilometraggio_Attuale,Foto) VALUES
('LAL123TFG','Avis','Auto','Nissan Leaf','Rosso',30,99,199,5,15000,'nissan-
leaf1.jpg');
INSERT INTO
Veicolo_elettrico(Targa,Punto_Noleggior,Tipologia,Nome_Modello,Colore,Costo_orari
o,Cilindrata,Autonomia_km,Max_Passeggeri,Chilometraggio_Attuale,Foto) VALUES
('FF95JGKHF','Hertz','Scooter','Trefor
Tripl','Bianco',23,39,125,NULL,NULL,'trefor-tripl.jpg');
INSERT INTO
Veicolo_elettrico(Targa,Punto_Noleggior,Tipologia,Nome_Modello,Colore,Costo_orari
o,Cilindrata,Autonomia_km,Max_Passeggeri,Chilometraggio_Attuale,Foto) VALUES
('ALOPR6D6F','Avis','Auto','Tesla ModelS','Rosso',45,77,144,5,11000,'Tesla-
modelS.jpg');
INSERT INTO
Veicolo_elettrico(Targa,Punto_Noleggior,Tipologia,Nome_Modello,Colore,Costo_orari
o,Cilindrata,Autonomia_km,Max_Passeggeri,Chilometraggio_Attuale,Foto) VALUES
('WOW202049','Car Italy','Auto','E-UP','Bianca',33,34,166,5,3000,'up_e-up.jpg');
INSERT INTO
Veicolo_elettrico(Targa,Punto_Noleggior,Tipologia,Nome_Modello,Colore,Costo_orari
o,Cilindrata,Autonomia_km,Max_Passeggeri,Chilometraggio_Attuale,Foto) VALUES
('Z58GZ36HZ','Car Italy','Scooter','Askoll
es1','Bianco',64,88,123,NULL,NULL,NULL);

INSERT INTO
Postazione_Prelievo(Indirizzo,Numero_Bici_Totale,Latitudine,Longitudine) VALUES
('via santo stefano 3',10,44.495603, 11.353665);
INSERT INTO
Postazione_Prelievo(Indirizzo,Numero_Bici_Totale,Latitudine,Longitudine) VALUES
('via irnerio 4',25,44.500859, 11.346164);
INSERT INTO
Postazione_Prelievo(Indirizzo,Numero_Bici_Totale,Latitudine,Longitudine) VALUES
('via clavature 2',22,44.493449, 11.343733);
INSERT INTO
Postazione_Prelievo(Indirizzo,Numero_Bici_Totale,Latitudine,Longitudine) VALUES
('via paglia 2',15,44.488376, 11.340799);

INSERT INTO Bici(Id,Postazione_Prelievo,Marca,Colore,Anno_Acquisizione) VALUES
('01','via santo stefano 3','Scott Genius','Blu','2014');
INSERT INTO Bici(Id,Postazione_Prelievo,Marca,Colore,Anno_Acquisizione) VALUES
('02','via irnerio 4','Diadora','Rosso','2011');
INSERT INTO Bici(Id,Postazione_Prelievo,Marca,Colore,Anno_Acquisizione) VALUES
('03','via paglia 2','SRAM','Azzurro','2010');
INSERT INTO Bici(Id,Postazione_Prelievo,Marca,Colore,Anno_Acquisizione) VALUES
('04','via clavature 2','Synchros','Nero','2016');
INSERT INTO Bici(Id,Postazione_Prelievo,Marca,Colore,Anno_Acquisizione) VALUES
('05','via irnerio 4','XBionic','Bianco','2008');
INSERT INTO Bici(Id,Postazione_Prelievo,Marca,Colore,Anno_Acquisizione) VALUES
('06','via clavature 2','Topeak','Grigio','2999');

INSERT INTO Pista_Ciclabile(Chilometri,Pendenza_Media,Latitudine,Longitudine)
VALUES (25,30.12,44.499554, 11.326543);
INSERT INTO Pista_Ciclabile(Chilometri,Pendenza_Media,Latitudine,Longitudine)
VALUES (11,11,44.504853,11.345467);
INSERT INTO Pista_Ciclabile(Chilometri,Pendenza_Media,Latitudine,Longitudine)
VALUES (8,0,44.486669, 11.339229);
INSERT INTO Pista_Ciclabile(Chilometri,Pendenza_Media,Latitudine,Longitudine)
VALUES (3,99.99,44.521599, 11.362375);

```

```

INSERT INTO
Prenotazione_Veicolo(EmailUtente,Veicolo,Prezzo_Prenotazione,Data_Inizio,Data_Fi
ne) VALUES ('utente1@gmail.com','WOW202049',12,'2016-04-04 05:05:00','2016-04-05
11:05:00');
INSERT INTO
Prenotazione_Veicolo(EmailUtente,Veicolo,Prezzo_Prenotazione,Data_Inizio,Data_Fi
ne) VALUES ('utente2@gmail.com','123ABC456',200,'2016-05-11 04:01:00','2016-05-
11 14:23:00');
INSERT INTO
Prenotazione_Veicolo(EmailUtente,Veicolo,Prezzo_Prenotazione,Data_Inizio,Data_Fi
ne) VALUES ('utente3@gmail.com','123ABC456',50,'2016-04-16 08:05:00','2016-04-23
08:05:00');
INSERT INTO
Prenotazione_Veicolo(EmailUtente,Veicolo,Prezzo_Prenotazione,Data_Inizio,Data_Fi
ne) VALUES ('utente2@gmail.com','WOW202049',12,'2016-01-02 10:05:22','2016-01-03
12:05:22');
INSERT INTO
Prenotazione_Veicolo(EmailUtente,Veicolo,Prezzo_Prenotazione,Data_Inizio,Data_Fi
ne) VALUES ('utente1@gmail.com','LAL123TFG',200,'2016-05-01 20:00:00','2016-05-
02 10:30:00');
INSERT INTO
Prenotazione_Veicolo(EmailUtente,Veicolo,Prezzo_Prenotazione,Data_Inizio,Data_Fi
ne) VALUES ('utente3@gmail.com','123ABC456',50,'2016-04-01 00:01:00','2016-04-03
10:00:00');
INSERT INTO
Prenotazione_Veicolo(EmailUtente,Veicolo,Prezzo_Prenotazione,Data_Inizio,Data_Fi
ne) VALUES ('utente3@gmail.com','WOW202049',12,'2016-03-19 19:45:52','2016-03-21
21:05:39');
INSERT INTO
Prenotazione_Veicolo(EmailUtente,Veicolo,Prezzo_Prenotazione,Data_Inizio,Data_Fi
ne) VALUES ('utente3@gmail.com','Z58GZ36HZ',200,'2016-05-21 09:15:00','2016-05-
22 16:35:40');
INSERT INTO
Prenotazione_Veicolo(EmailUtente,Veicolo,Prezzo_Prenotazione,Data_Inizio,Data_Fi
ne) VALUES ('utente1@gmail.com','Z58GZ36HZ',50,'2016-04-03 17:05:22','2016-04-07
20:15:00');
INSERT INTO
Prenotazione_Veicolo(EmailUtente,Veicolo,Prezzo_Prenotazione,Data_Inizio,Data_Fi
ne) VALUES ('utente3@gmail.com','123ABC456',50,'2016-07-16 08:05:00','2016-07-23
08:05:00');
INSERT INTO
Prenotazione_Veicolo(EmailUtente,Veicolo,Prezzo_Prenotazione,Data_Inizio,Data_Fi
ne) VALUES ('utente2@gmail.com','WOW202049',12,'2016-06-02 10:05:22','2016-06-03
12:05:22');
INSERT INTO
Prenotazione_Veicolo(EmailUtente,Veicolo,Prezzo_Prenotazione,Data_Inizio,Data_Fi
ne) VALUES ('utente1@gmail.com','LAL123TFG',200,'2016-08-01 20:00:00','2016-08-
02 10:30:00');

```

```

INSERT INTO Prenotazione_Bici(EmailUtente,IdBici,Data_Inizio,Data_Fine) VALUES
('utente3@gmail.com','01','2016-05-01 01:05:00','2016-05-01 03:55:00');
INSERT INTO Prenotazione_Bici(EmailUtente,IdBici,Data_Inizio,Data_Fine) VALUES
('utente2@gmail.com','02','2016-04-22 13:25:02','2016-04-22 22:05:41');
INSERT INTO Prenotazione_Bici(EmailUtente,IdBici,Data_Inizio,Data_Fine) VALUES
('utente2@gmail.com','05','2016-05-23 09:19:19','2016-05-23 19:27:56');
INSERT INTO Prenotazione_Bici(EmailUtente,IdBici,Data_Inizio,Data_Fine) VALUES
('utente1@gmail.com','05','2016-05-03 15:29:23','2016-05-03 18:45:12');
INSERT INTO Prenotazione_Bici(EmailUtente,IdBici,Data_Inizio,Data_Fine) VALUES
('utente3@gmail.com','01','2016-03-12 04:05:00','2016-03-12 07:23:10');
INSERT INTO Prenotazione_Bici(EmailUtente,IdBici,Data_Inizio,Data_Fine) VALUES
('utente1@gmail.com','03','2016-04-18 16:05:22','2016-04-18 16:55:22');

```

```

INSERT INTO Prenotazione_Bici(EmailUtente,IdBici,Data_Inizio,Data_Fine) VALUES
('utente3@gmail.com','02','2016-04-04 01:00:00','2016-04-04 01:15:00');
INSERT INTO Prenotazione_Bici(EmailUtente,IdBici,Data_Inizio,Data_Fine) VALUES
('utente1@gmail.com','01','2016-05-09 16:33:10','2016-05-09 22:00:00');
INSERT INTO Prenotazione_Bici(EmailUtente,IdBici,Data_Inizio,Data_Fine) VALUES
('utente1@gmail.com','01','2016-05-11 07:05:00','2016-05-11 14:50:24');
INSERT INTO Prenotazione_Bici(EmailUtente,IdBici,Data_Inizio,Data_Fine) VALUES
('utente2@gmail.com','06','2016-04-26 04:11:25','2016-04-26 08:22:22');
INSERT INTO Prenotazione_Bici(EmailUtente,IdBici,Data_Inizio,Data_Fine) VALUES
('utente3@gmail.com','04','2016-02-29 05:05:05','2016-02-29 07:07:07');
INSERT INTO Prenotazione_Bici(EmailUtente,IdBici,Data_Inizio,Data_Fine) VALUES
('utente1@gmail.com','02','2016-03-18 22:45:00','2016-03-19 04:12:12');
INSERT INTO Prenotazione_Bici(EmailUtente,IdBici,Data_Inizio,Data_Fine) VALUES
('utente2@gmail.com','06','2016-06-26 04:11:25','2016-06-26 08:22:22');
INSERT INTO Prenotazione_Bici(EmailUtente,IdBici,Data_Inizio,Data_Fine) VALUES
('utente3@gmail.com','04','2016-07-29 05:05:05','2016-07-29 07:07:07');
INSERT INTO Prenotazione_Bici(EmailUtente,IdBici,Data_Inizio,Data_Fine) VALUES
('utente1@gmail.com','02','2016-06-18 22:45:00','2016-06-19 04:12:12');

INSERT INTO ForumPost(EmailUtente,Titolo,Testo_Messaggio,Data_Inserimento)
VALUES ('admin@gmail.com','News','A breve nuove news riguardanti il sito','2016-
05-11 17:13:22');
INSERT INTO ForumPost(EmailUtente,Titolo,Testo_Messaggio,Data_Inserimento)
VALUES ('admin@gmail.com','Aggiornamento sito','il sito &eacute; stato
aggiornato. A breve le info','2016-04-21 11:43:41');
INSERT INTO ForumPost(EmailUtente,Titolo,Testo_Messaggio,Data_Inserimento)
VALUES ('admin@gmail.com','Inserimento veicolo','Inserito nuovo veicolo','2016-
05-17 22:11:10');

INSERT INTO
Colonnina_Elettrica(Indirizzo,Ente_Fornitore,Max_KWH,Data_Inserimento,Latitudine
,Longitudine) VALUES ('via panto cane 5','Enel',150,'2016-05-02',44.501218,
11.361283);
INSERT INTO
Colonnina_Elettrica(Indirizzo,Ente_Fornitore,Max_KWH,Data_Inserimento,Latitudine
,Longitudine) VALUES ('via parigi 12','Hera',200,'2016-05-07',44.496566,
11.341283);
INSERT INTO
Colonnina_Elettrica(Indirizzo,Ente_Fornitore,Max_KWH,Data_Inserimento,Latitudine
,Longitudine) VALUES ('via saragozza 74','AIM Energy',233,'2016-05-
02',44.490060, 11.332197);
INSERT INTO
Colonnina_Elettrica(Indirizzo,Ente_Fornitore,Max_KWH,Data_Inserimento,Latitudine
,Longitudine) VALUES ('via santo stefano 103','Hera',145,'2016-05-01',44.486947,
11.353282);
INSERT INTO
Colonnina_Elettrica(Indirizzo,Ente_Fornitore,Max_KWH,Data_Inserimento,Latitudine
,Longitudine) VALUES ('via Giovanni Amendola 12','Edison Energia',200,'2016-05-
12',44.504589, 11.341035);

INSERT INTO
Prenotazione_Colonnina(EmailUtente,Indirizzo,Slot_Inizio,Slot_Fine,Data_pren)
VALUES ('utente2@gmail.com','via panto cane 5',5,7,'2016-06-15');
INSERT INTO
Prenotazione_Colonnina(EmailUtente,Indirizzo,Slot_Inizio,Slot_Fine,Data_pren)
VALUES ('utente2@gmail.com','via parigi 12',3,4,'2016-06-01');
INSERT INTO
Prenotazione_Colonnina(EmailUtente,Indirizzo,Slot_Inizio,Slot_Fine,Data_pren)
VALUES ('utente3@gmail.com','via saragozza 74',3,4,'2016-08-15');

```

```
INSERT INTO
Prenotazione_Colonnina(EmailUtente,Indirizzo,Slot_Inizio,Slot_Fine,Data_pren)
VALUES ('utente3@gmail.com','via santo stefano 103',7,9,'2016-08-15');
INSERT INTO
Prenotazione_Colonnina(EmailUtente,Indirizzo,Slot_Inizio,Slot_Fine,Data_pren)
VALUES ('utente2@gmail.com','via Giovanni Amendola 12',12,17,'2016-07-01');
INSERT INTO
Prenotazione_Colonnina(EmailUtente,Indirizzo,Slot_Inizio,Slot_Fine,Data_pren)
VALUES ('utente3@gmail.com','via santo stefano 103',1,6,'2016-09-22');
```

```
INSERT INTO
Segnalazione(Pista_Ciclabile,EmailUtente,Titolo,Testo_Messaggio,Data_Inserimento)
VALUES ('1','utente2@gmail.com','Buca','incidente','2015-11-12 13:49:02');
INSERT INTO
Segnalazione(Pista_Ciclabile,EmailUtente,Titolo,Testo_Messaggio,Data_Inserimento)
VALUES ('3','utente3@gmail.com','Mancanza Segnaletica','Chiamate la
polizia','2016-04-18 16:17:02');
INSERT INTO
Segnalazione(Pista_Ciclabile,EmailUtente,Titolo,Testo_Messaggio,Data_Inserimento)
VALUES ('4','utente2@gmail.com','Segnalazione di prova','segnalo
qualcosa','2015-11-12 13:49:02');
INSERT INTO
Segnalazione(Pista_Ciclabile,EmailUtente,Titolo,Testo_Messaggio,Data_Inserimento)
VALUES ('3','utente3@gmail.com','Incidente','Bici tutte rotte','2016-04-18
16:17:02');
```