

# Indice

<b>1</b>	<b>Piattaforma di Car Sharing</b>	<b>1</b>
1.1	Introduzione al progetto . . . . .	1
1.2	Progettazione . . . . .	1
1.2.1	Analisi dei requisiti . . . . .	1
1.2.2	Diagramma Entità-Relazione (ER) . . . . .	2
1.2.3	Schema logico . . . . .	4
1.2.4	Dizionario dei dati . . . . .	5
1.3	Conclusioni . . . . .	7

## 1 Piattaforma di Car Sharing

### 1.1 Introduzione al progetto

Si vuole creare un software per la gestione di un servizio di Car Sharing. Si deve realizzare uno schema ER dell'applicazione con il relativo schema logico con annessa spiegazione delle scelte progettuali. Una volta che la progettazione è completata è richiesta la creazione di uno Script SQL in cui si includono le seguenti parti:

- Creazione del DataBase
- Creazione di tutte le tabelle con relativi vincoli
- Popolazione del DataBase con dati di esempio (almeno 5 record per tabella)

Inoltre è anche richiesto di creare un dizionario dei dati al fine di descrivere le tabelle e i campi. Infine sono richieste delle considerazioni finali sul progetto.

### 1.2 Progettazione

#### 1.2.1 Analisi dei requisiti

La piattaforma deve gestire tutto il processo di un noleggio, che è costituito dai seguenti passaggi:

- Memorizzare i dati degli utenti e delle loro patenti.
- Registrare i dati dei veicoli.
- Registrare i luoghi in cui è possibile ritirare o restituire i veicoli, che possono essere stazioni o parcheggi.
- Si devono registrare le prenotazioni con data e ora di inizio del noleggio e data e ora della riconsegna del veicolo.
- Si deve tenere traccia delle manutenzioni che vengono eseguite su un veicolo.
- In un noleggio possono essere effettuati dei danni sul veicolo o possono essere prese delle multe.
- Si deve anche gestire il sistema tariffario dei noleggi
- Inoltre si devono poter scrivere delle recensioni in merito ad un noleggio.
- Infine vanno inseriti dei vincoli: Un veicolo può essere noleggiato da un solo utente alla volta, calcolo automatico tariffe, stato veicolo (disponibile, in uso, in manutenzione).

### 1.2.2 Diagramma Entità-Relazione (ER)

```

erDiagram
    OFFICINA ||--o{ MANUTENZIONE : "Eseguita da"
    MANUTENZIONE o{ -- || VEICOLO : "Viene eseguita"
    VEICOLO || -- o| PRENOTAZIONE : "è assegnato"
    PRENOTAZIONE ||--O| PAGAMENTO : "Viene effettuato"
    PAGAMENTO O{ -- || mETODOdIpAGAMENTO : Accetta
    PRENOTAZIONE ||--|| NOLEGGIO : Crea
    NOLEGGIO || -- O| RECENZIONE : "Viene scritta"
    UTENTE ||--O{ PRENOTAZIONE : Effettua
    PATENTE || -- || UTENTE : Possiede
    STAZIONE || --O{ PRENOTAZIONE : Presso
    NOLEGGIO O{ --|| PAGAMENTO : "Extra saldati con"
    OFFICINA {
        string ragioneSociale pk
        string Indirizzo_Comune
        string Indirizzo_Via
        string Indirizzo_Civico
        int Indirizzo_Cap
        int partitaIva
        int telefono
        string email
        set tipologia
    }
    VEICOLO {
        string targa pk
        string marca
        string modello
        string posizioneGPS
        float prezzoAlgiorno
        enum categoria
        int cc
        int cv
        int posti
        set optionals*
    }
    MANUTENZIONE {
        string idManutenzione pk
        string fkVeicolo fk
        string fkOfficina fk
        date data
        string descrizione
        set tipologia
    }
    mETODOdIpAGAMENTO {
        int idMetodo pk
        string nomeTitolare
        string cognomeTitolare
    }

```

```
    enum tipologia
    int numeroCarta*
    date scadenzaCarta*
    string numeroConto*
    string email*
}
STAZIONE{
    string idStazione pk
    string Indirizzo_Comune
    int Indirizzo_Cap
    string Indirizzo_Via
    string Indirizzo_Civico
    bool coperto
}
UTENTE{
    string codiceFiscale pk
    string nome
    string cognome
    date dataDiNascita
    string comuneNascita
}
PAGAMENTO{
    int idPagamento pk
    string fkUtente fk
    int fkMetodoDiPagamento fk
    float importo
    string causale
    date dataPagamento
    time oraPagamento
}
PATENTE{
    string numeroPatente pk
    string fkUtente fk
    string enteRilasciatore
    date dataRilascio
    date dataScadenza
    set tipologia
}
NOLEGGIO{
    string fkPrenotazione PK, FK
    date dataRestituzione pk
    int kmPercorsi
    float prezzoAlChilometro
    set extra
    float importo*
    string descrizione*
    int tempoExtra*
```

```

    int puntiPatenteTolti*
}
RECENZIONE{
    string idRecensione pk
    string fkNoleggior pk
    date dataPubblicazione
    int punteggio
    string descrizione
}
PRENOTAZIONE{
    string idPrenotazione pk
    string fkVeicolo fk
    string fkUtente fk
    string fkMetodoDiPagamento fk
    string fkStazione fk
    date dataPrenotazione
    date dataInizioNoleggior
    time oraInizioNoleggior
    date dataFineNoleggior
    time oraFineNoleggior
    int maxKmPrevisti
}

```

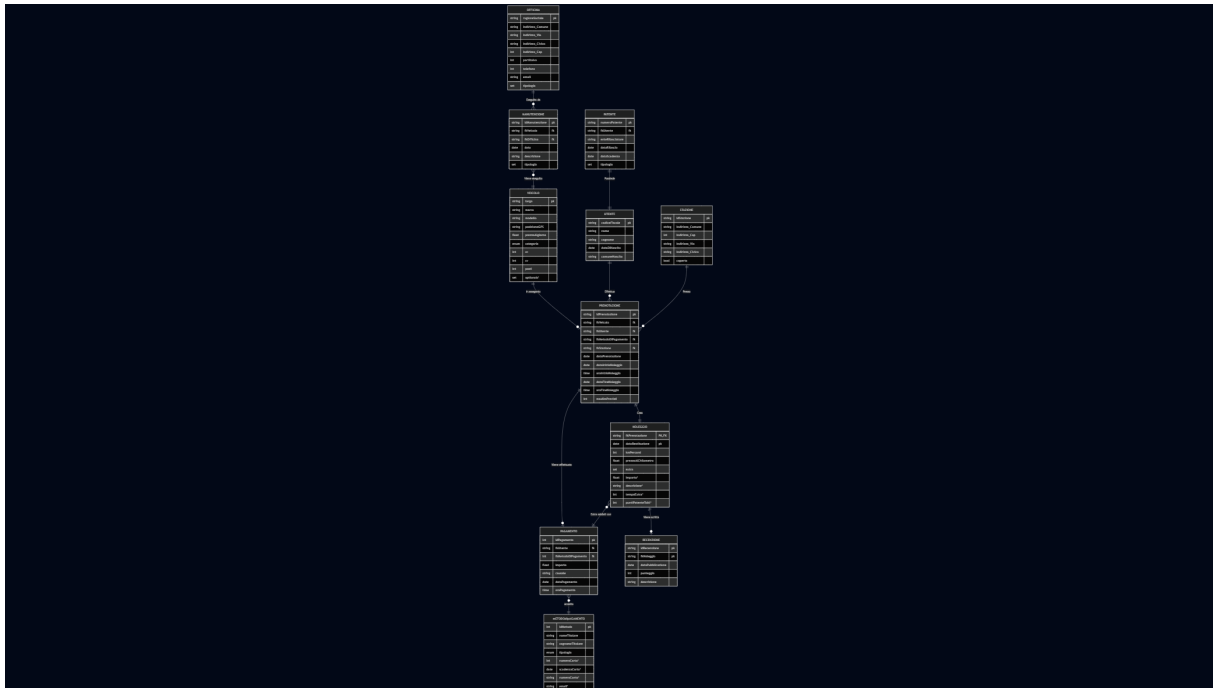


Figura 1: Diagramma ER

### 1.2.3 Schema logico

Veicolo(Targa, marca, modello, latitudine, longitudine, prezzoAlGiorno, categoria, cc, cv, posti,optionals\*)

Officina(ragioneSociale, indirizzo\_Comune, indirizzo\_Via, indirizzo\_Civico, indirizzo\_Cap,partitaIva, Telefono, email, tipologia)

Manutenzione(idManutenzione, fkVeicolo, fkOfficina, kmVeicolo, data, descrizione)

metodoDiPagamento(idMetodo, nomeTitolare, cognomeTitolare, Tipogia, numeroCarta, *scadenzaCarta*,numeroConto,*email*)

Stazione(idStazione, indirizzo\_Comune,indirizzo\_Cap, indirizzo\_Via, indirizzo\_Civico, coperto )

Utente(codiceFiscale, nome, cognome, dataDiNascita, comuneDiNacita)

Pagamento(idPagamento, fkUtente, fkMetodoDiPagamento, importo, causale\*,dataPagamento, oraPagamento)

Patente(numeroPatente, fkUtente, enteRilasciatore, dataRilascio, dataScadenza, tipologia)

Prenotazione(idPrenotazione, fkVeicolo, fkUtente, fkMetodoDIPagamento, fkStazione,dataPrenotazione, dataInizioNoleggio, oraInizioNoleggio, dataFineNoleggio, oraFineNoleggio, MaxkmPrevisti)

Noleggio(fkPrenotazione,dataRestituzione, kmPercorsi, extra, importoExtra, *tempoExtra*, puntiPatenteTolti\*)

Recensione(idRecensione, fkNoleggio, dataPubblicazione, punteggio, descrizione)

## 1.2.4 Dizionario dei dati

Descrizione di tabelle e campi:

**1.2.4.1 Veicolo** La tabella veicolo contiene i dati di tutti i veicoli presenti sulla piattaforma. Pero ogni veicolo si tiene la targa per identificare univocamente un veicolo, la targa è una stringa di 6 caratteri. Inoltre risulta fondamentale salvarsi marca e modello per aiutare gli utenti a scegliere un veicolo. Come in ogni piattaforma, per l'azienda che affitta i veicoli è fondamentale sapere in ogni momento dove si trovano i propri veicoli e ciò viene fatto con i due attributi che indicano latitudine e longitudine. Inoltre per ogni veicolo si tiene conto del prezzo al giorno in modo da riuscire poi a calcolare il prezzo di ogni noleggio in base al numero di giorni del noleggio. Dopodichè i veicoli si distinguono con un attributo di tipo enum in autovetture e motocicli. Per ogni veicolo si tiene conto dei parametri del motore quali cavalli e cilindrata e i posti omologati. Nel caso dei motocicli i posti omologati massimi sono 2 e nel caso delle autovetture il limite di 9 posti (limite imposto per legge a autovetture e autocarri) è imposto dalla lunghezza della variabile che prevede solo un numero intero da una cifra (da 0 a 9) e limitata da un vincolo che lo vincola >0. Infine c'è un attributo di tipo set per includere gli optionals, esso è disponibile solo se il veicolo in questione è un'autovettura.

**1.2.4.2 Officina** La tabella officina contiene i dati delle officine. Ogni officina è identificata univocamente dalla ragione sociale, è caratterizzata da suo indirizzo (Comune, cap, via, civico), dalla partita iva (numero intero da 11 caratteri), da un numero di telefono e un'email. Infine si vuole identificare il tipo di officina tramite un attributo di tipo set che

indica i tipi di lavori che vengono svolti all'interno di essa, ad esempio lavori di carrozzeria, lavori di officina e lavori di elettrauto.

**1.2.4.3 Manutenzione** La manutenzione di un veicolo dell'azienda di car sharing è identificata da un numero intero, nell'sql l'ho messo `auto_increment` tanto è progressivo. Per ogni manutenzione si tiene la targa del veicolo a cui è stata eseguita la manutenzione in quanto chiave primaria dell'entità veicolo. Inoltre c'è la chiave `pprimaria` dell'officina in cui è eseguita la manutenzione. Si tiene anche traccia dei chilometri del veicolo al momento della manutenzione (numero intero da 6 cifre in quanto è difficile che un'auto arrivi a più di 999.999 chilometri) e infine c'è una breve descrizione dei lavori eseguiti.

**1.2.4.4 Metodo di pagamento** Un metodo di pagamento descrive i dati di pagamento di una persona. Si ha un id del metodo di pagamento che è un numero intero da 6 caratteri, ogni metodo di pagamento di ogni persona ha un numero diverso, anch'esso ha l'`auto_increment` in quanto è un numero progressivo. Poi ci sono il nome e il cognome del titolare del metodo di pagamento e un attributo `enum` che indica i 3 tipi di pagamento disponibili quali bonifico, carta di credito e paypal. Nel caso del bonifico sarà necessario inserire il numero del proprio conto bancario, nel caso della carta di credito bisogna inserire il numero della carta di credito e la sua data di scadenza, mentre nel caso di Paypal è necessario inserire l'email con cui si è registrati alla piattaforma di pagamento. Questi ultimi dati sono limitati da un vincolo di tupla. Si potrebbe pensare di sostituire il nome e cognome dell'intestatario con una `fk` all'entità utente descritta successivamente, ma fare ciò non conviene perché un'utente che fa un' prenotazione potrebbe pagare con un conto o una carta di credito non intestato a se stesso.

**1.2.4.5 Stazione** L'entità stazione indica un luogo in cui si ritirano e riconsegnano i veicoli. Per ogni stazione si ha un id della stazione, che consiste in una stringa composta dalla provincia in cui si trova e un numero intero di 3 cifre per identificare più stazioni all'interno della provincia. Ogni stazione è caratterizzata dal suo indirizzo (comune, cap, via, civico) e una variabile booleana che indica se la stazione è coperta o no.

**1.2.4.6 Utente** L'entità utente ci identifica una persona che richiede di fare un noleggio. Ogni utente viene identificato univocamente dal codice fiscale e caratterizzato dal nome, dal cognome, la data di nascita e il nome del comune di nascita.

**1.2.4.7 Pagamento** Il pagamento indica un'operazione di pagamento seguendo un metodo di pagamento. Ogni pagamento è identificato univocamente da un id intero di 9 cifre con `auto_increment` in quanto è un numero progressivo. È caratterizzato dal codice fiscale dell'utente che esegue la prenotazione, e l'id del metodo di pagamento usato. Inoltre è caratterizzato dall'importo del pagamento, una causale e la data e ora in cui viene fatto il pagamento.

**1.2.4.8 Patente** Ogni patente è identificata univocamente dal codice della patente che è una stringa di 10 caratteri, mentre è caratterizzata dal riferimento della persona a cui è intestata, dall'ente che l'ha rilasciata, dalla data di conseguimento e dalla data di scadenza. Inoltre c'è un attributo di tipo `set` per indicare quali tipi di patente ha conseguito l'utente.

**1.2.4.9 Prenotazione** Una prenotazione tiene i dati di un noleggio al momento in cui un utente fa una richiesta a quando non riconsegna il veicolo. Una prenotazione è identificata univocamente da un codice alfanumerico di 7 caratteri ed è caratterizzato dalla targa del veicolo che è stato scelto, dal codice fiscale dell'utente che esegue la prenotazione, dall'id del metodo di pagamento che verrà usato per saldare il noleggio e dall'id della stazione in cui verrà ritirato il veicolo. Inoltre per ogni prenotazione si tiene traccia della data della prenotazione, della data e dell'ora da cui è possibile ritirare il veicolo e dalla data e ora previste per la fine del noleggio. Per ogni prenotazione si impone anche un numero massimo di chilometri percorribili dall'utente, dopo lo sforamento di questa soglia si deve pagare una penale.

**1.2.4.10 Noleggio** L'entità noleggio tiene traccia di un noleggio il cui veicolo è già stato restituito, è identificato univocamente dalla prenotazione a cui fa riferimento e dalla data di restituzione effettiva del veicolo per avere la conferma che il veicolo sia stato effettivamente restituito e valutare eventuali ritardi nella riconsegna di esso. Inoltre si salva anche il numero di chilometri percorsi dall'utente con il veicolo noleggiato. Infine c'è un attributo di tipo set che indica gli extra quali munte, danni, sovrapprezzo data di restituzione o sforamento numero chilometri previsti per il noleggio. Infine ci sono 2 attributi nullable che vengono usati solo nel caso in cui ci sia uno sforamento di tempo (tempoExtra) espressa in ore arrotondando il tempo extra per eccesso e nel caso in cui ci sia una multa e dunque una perdita di punti dalla patente dell'utente (puntiPatenteTolti).

**1.2.4.11 Recensione** In seguito ad aver fatto un noleggio l'utente può scrivere una recensione. Le recensioni sono identificate univocamente da un attributo intero con auto\_increment essendo un numero progressivo, HA un riferimento diretto al noleggio effettuato per evitare che alcune persone possano lasciare una recensione senza aver neanche fatto un noleggio. Inoltre si tiene traccia della data di pubblicazione della recensione, di un punteggio (da 0 a 9) e una descrizione.

## 1.3 Conclusioni

Per migliorare l'applicazione di gestione del car sharing sarebbe necessario implementare uno script php per rendere più precisi i collegamenti degli extra con dei controlli ad esempio sulle date di restituzione o i chilometri percorsi. Inoltre facendo questo script si riesce a calcolare il costo di ogni noleggio.