

**Università di Pisa**

**Corso di laurea in Ingegneria Informatica**

**Progetto**

**Gabriele Benanti Marco Bellezza | Basi di dati | a.a 2017-2018**

Sommario

[***Introduzione*** 3](#_Toc5558626)

[***Progettazione concettuale del diagramma E-R non ristrutturato*** 7](#_Toc5558627)

[*Generalizzazioni* 7](#_Toc5558628)

[*Attributi composti* 8](#_Toc5558629)

[*Attributi multivalore* 9](#_Toc5558630)

[*Attributi derivabili* 10](#_Toc5558631)

[***Ristrutturazione del diagramma E-R*** 11](#_Toc5558632)

[*Traduzione generalizzazioni* 11](#_Toc5558633)

[*Traduzione di attributi composti* 11](#_Toc5558634)

[*Traduzione di attributi multivalore* 12](#_Toc5558635)

[*Ultima ristrutturazione* 13](#_Toc5558636)

[***Analisi delle prestazioni tramite identificazione delle operazioni*** 14](#_Toc5558637)

[***Analisi delle ridondanze*** 42](#_Toc5558638)

[***Modello relazionale*** 53](#_Toc5558639)

[***Vincoli di integrità*** 55](#_Toc5558640)

[*Vincoli generici* 56](#_Toc5558641)

[***Normalizzazione*** 57](#_Toc5558642)

[***Area Analytics*** 63](#_Toc5558643)

[*Affidabilità di un utente* 63](#_Toc5558644)

[*Tempi di percorrenza e rilevazione delle criticità* 66](#_Toc5558645)

[***Script SQL*** 68](#_Toc5558646)

# ***Introduzione***

L’obiettivo del progetto è quello di costruire da zero un database per un’azienda che opera nel settore della mobilità. I servizi offerti dall’azienda attraverso le sue piattaforme sono tre: *Car Sharing, Car Pooling* e *Ride Sharing.*

Il primo consiste nel dare a noleggio un’auto di proprietà di terzi. Il secondo offre la possibilità di condividere veicoli privati tra più individui che devono percorrere lo stesso tragitto, o parte di esso. Il terzo è un servizio di trasporto su richiesta.

L’azienda tiene traccia di tutti gli spostamenti effettuati dai veicoli predisposti ai servizi, in modo tale da avere un tracciamento preciso del percorso seguito dal veicolo, al fine di stimare una durata media del tragitto. Inoltre, possibili ritardi causati da incidenti o traffico vengono registrati al fine di segnalare tempestivamente rallentamenti per quel determinato tratto di strada, così che si eviti di avere tempi di percorrenza imprecisi. In caso di incidenti andremo a registrare nel database i veicoli convolti nell’incidente, riconducendoci all’utente al volante del veicolo incidentato con cui è stato effettuando il servizio, al fine di stabilire l’affidabilità dell’utente.

Successivamente è stato stilato un glossario per eliminare ogni ambiguità sui termini utilizzati in questa relazione.

|  |  |  |  |
| --- | --- | --- | --- |
| TERMINE | DESCRIZIONE | SINONIMI | COLLEGAMNETI |
| Account | Strumento che permette di usufruire dei servizi offerti dalla piattaforma | Profilo utente | Utente |
| Autovettura | Veicoli messi a disposizione dai vari utenti proponenti | Auto, Automobile, Vettura, Veicolo | Modello, Noleggio, Optional, Proponente, RideSharing |
| AutoCoinvolta | Autovettura coinvolta in un sinistro stradale in cui era presente un’auto di un proponente | Veicolo Coinvolto | Incidente |
| ChiamataSharing | Operazione attraverso la quale si prenota il servizio di Ride Sharing | Richiesta Ride Sharing, Chiamata di Ride Sharing | Fruitore, Sharing |
| Chilometro | Raggruppa le informazioni di un chilometro di una specifica strada | Chilometraggio | Posizione, Strada |
| Disponibilità | Indica le informazioni sulla disponibilità di un’autovettura | Date disponibilità | Autovettura |
| Documento | Contiene le informazioni relative al documento di un utente | Generalità | Utente |
| FasciaOraria | Intervalli di ore in determinati giorni della settimana in cui c’è la disponibilità di una certa auto | Giorno Disponibile | Autovettura |
| Fruitore | Utente che usufruisce di un servizio | Utente Fruitore | ChiamataSharing, PrenotazioneP, Noleggio, Utente |
| FasciaOraria | Intervalli di ore in determinati giorni della settimana in cui c’è la disponibilità di una certa auto | Giorno Disponibile | Autovettura |
| Incidente | Contiene le informazioni sulla dinamica e sull’orario del sinistro | Sinistro | AutoCoinvolta, Noleggio |
| Modello | Tipologia di autovettura |  | Autovettura |
| Noleggio | Contiene le informazioni relative al noleggio di un’auto | Car Sharing | Autovettura, Fruitore, Incidente, Valutazione |
| Optional | Accessori disponibili su un veicolo | Extra | Autovettura |
| Pool | Contiene le informazioni sul servizio di Car Pooling proposto da un utente | Istanza di Car Pooling | PrenotazioneP, Proponente,Tragitto, Variazione |
| Posizione | Informazioni sulla posizione di un autoveicolo | Locazione | Chilometro, Tragitto |
| PrenotazioneP | Contiene il codice di una prenotazione di un Pool | RichiestaPool | Fruitore, Pool |
| Proponente | Utente che propone un servizio | Utente Proponente | Autovettura, Pool,Utente |
| RideSharing | Servizio di Ride Sharing, mediante il quale un utente fruitore può usufruire di un servizio taxi offerto da un proponente | Ride Sharing | Autovettura, Tragitto |
| Strada | Raccoglie le informazioni sulla tipologia di strada |  | Chilometro, Tragitto |
| Tragitto | Insieme di strade percorse durante l’erogazione e la contemporanea fruizione di un servizio | Percorso | Noleggio, Pool, Posizione, Sharing, Strada |
| Valutazione | Feedback fornito da ogni utente al termine di un tragitto | Feedback, Recensione | Noleggio, Pool, Sharing |
| Variazione | Richiesta di una deviazione su un tragitto di Car Pooling | Deviazione | Pool |

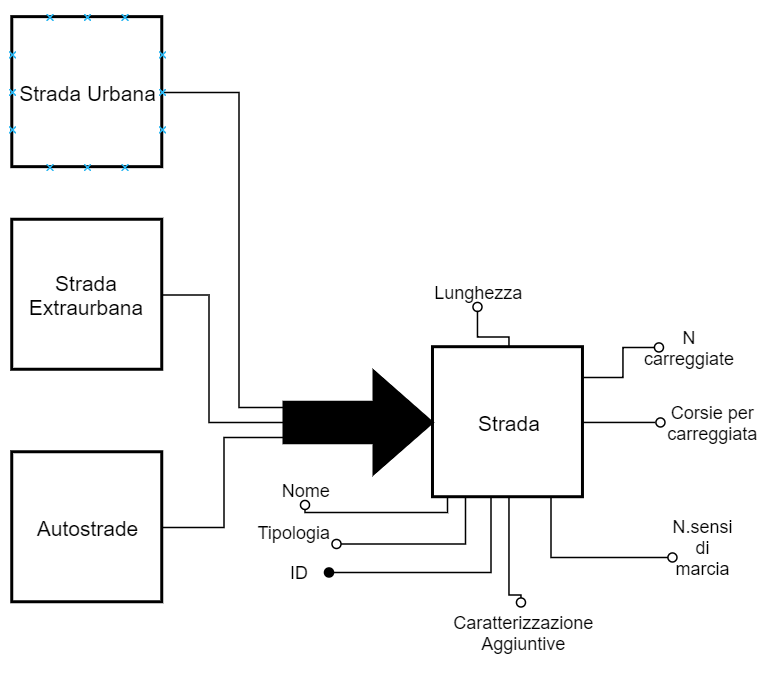
# ***Progettazione concettuale del diagramma E-R non ristrutturato***

Per lo sviluppo del di diagramma Entità-Relazione si è deciso di usare una strategia sia *top-down* sia *bottom-up.* Abbiamo iniziato a costruire il diagramma a partire da una piccola parte delle specifiche, andando ad unire ogni volta i relativi digrammi E-R prodotti, così da formare progressivamente il diagramma E-R. Seguendo questo procedimento abbiamo ottenuto come risultato finale il diagramma E-R non ristrutturato.

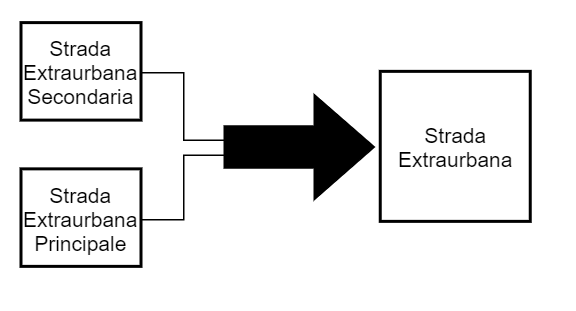
## *Generalizzazioni*

Le generalizzazioni da noi individuate riguardano le tipologie di strade: *autostrada, strada urbana e strada extraurbana.* A sua volta le strade extraurbane si suddividono in: *strade extraurbane primarie* e *strade extraurbane secondarie.*

La generalizzazione su strada è totale ed esclusiva, infatti ogni strada può appartenere ad una e una soltanto delle tre tipologie.



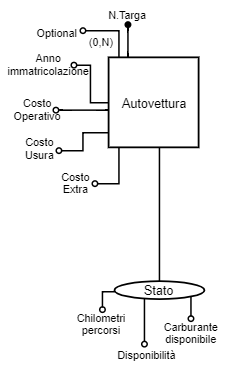
La generalizzazione su strada extraurbana è nuovamente totale ed esclusiva, sempre per il medesimo motivo.



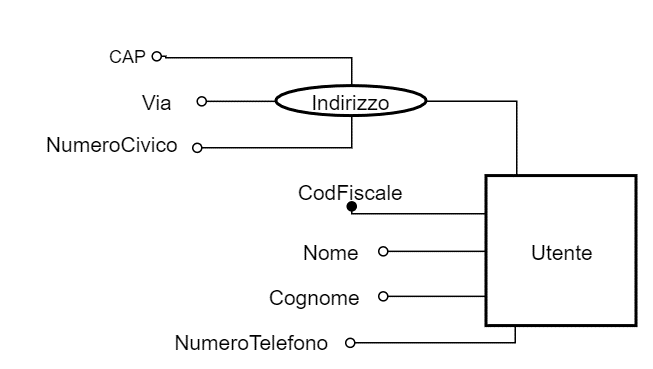
## *Attributi composti*

Abbiamo identificato due attributi composti.

Il primo attributo composto identificato è *Stato,* dell’entità **Autovettura**. Con questo attributo si tiene traccia dell’utilizzo del veicolo.

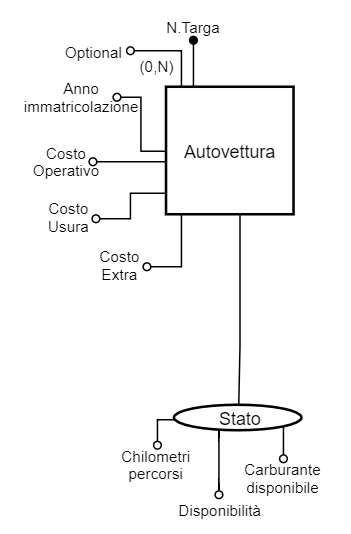


Il secondo attributo composto è *Indirizzo*, usato nell’entità **Utente**. Esso racchiude le informazioni per identificare un indirizzo.



## *Attributi multivalore*

Nella strutturazione del diagrammo è stato identificato come attributo multivalore *Optional,* usato nell’entità ***Autovettura***. Esso precisa quali accessori sono montati su un determinato veicolo.



## *Attributi derivabili*

Nell’entità ***Variazione*** l’attributo *Entità* si calcola considerando la somma di tutte le distanze tra le posizioni che formano un tragitto. In particolare, si considera:

* Piccola: fino a 2 Km
* Media: da 2 Km a 5 Km
* Grande: da 5 Km a 10 Km

In base al peso della variazione, si effettua un aumento di prezzo unicamente per chi ha richiesto il cambio del tragitto.

* 2% se l’entità è bassa
* 5% se l’entità è media
* 8% se l’entità è alta

Nell’entità ***Pool*** l’attributo *Spesa* dell’area servizi Car Pooling si calcola considerando il *Costo usura* e il *Costo operativo* e dal *Numero Passeggeri* più eventuali aumenti dovuti a *Variazioni*. Inoltre, la *Spesa* è pro capite.

Spesa =

Nell’entità ***Pool*** l’attributo *Flessibilità* indica quanto un proponente è disposto a variare il proprio tragitto. È un valore che viene indicato dal proponete quando crea il pool.

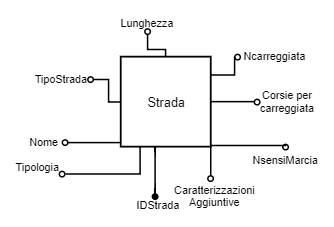
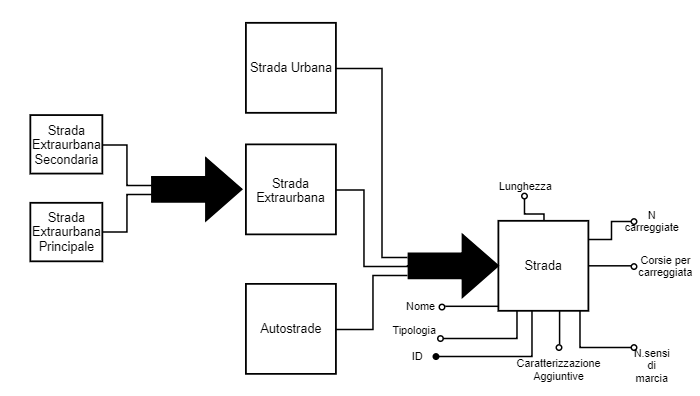
* Bassa: variazione ammessa inferiore ai 2Km
* Media: variazione ammessa compresa tra 2 Km e 5 Km
* Alta: variazione ammessa compresa tra i 5 Km e i 10 Km

# ***Ristrutturazione del diagramma E-R***

Prima della progettazione logica è stato necessario ristrutturare il diagramma E-R, essendo presenti generalizzazioni, attributi multivalore e attributi composti, elementi non traducibili per la progettazione logica.

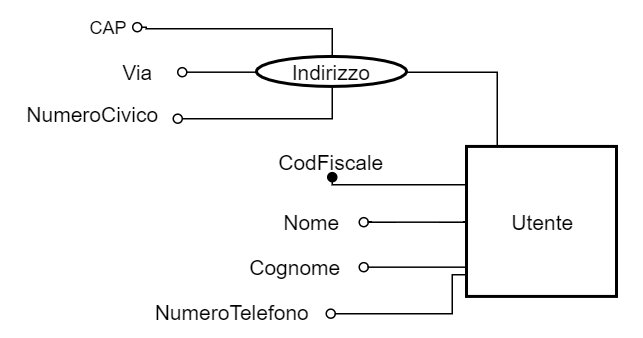
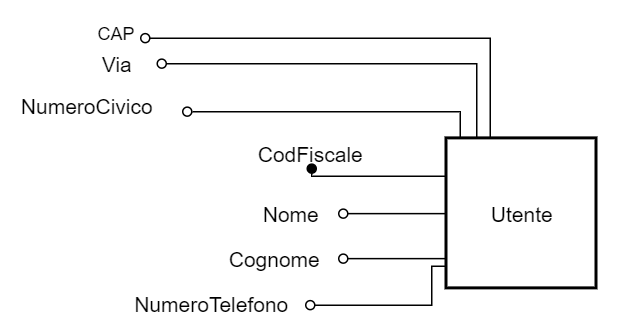
## *Traduzione generalizzazioni*

Le due generalizzazioni presenti nel diagramma E-R sono due. La prima riguarda le tipologie di *Strada.* La seconda riguarda le tipologie di strade extraurbane. Per eliminarele due generalizzazioni è stato necessario eliminare prima la seconda, essendo generalizzazione di generalizzazione, andando così ad avere due entità distinte *Strada extraurbana primaria* e *Strada extraurbana secondaria* entrambe figlie di *Strada*. Successivamente la prima generalizzazione è stata tradotta con un attributo sull’entità *Strada.*

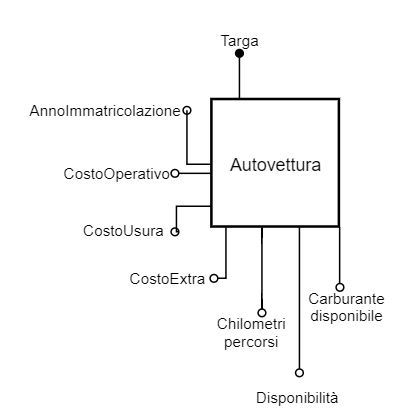
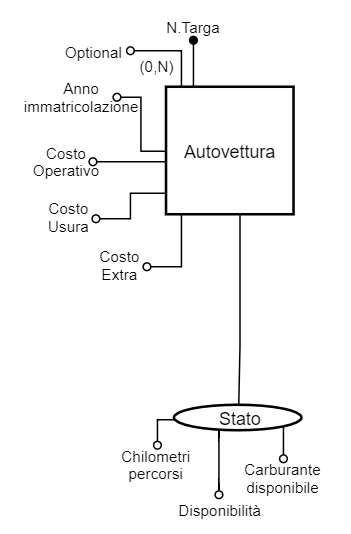


## *Traduzione di attributi composti*

Il primo attributo composto è *Indirizzo*, relativo all’entità utente. Esso è stato tradotto con tre attributi relativi all’entità Utente.

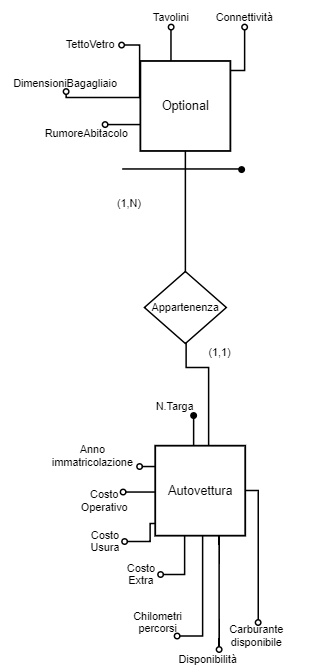
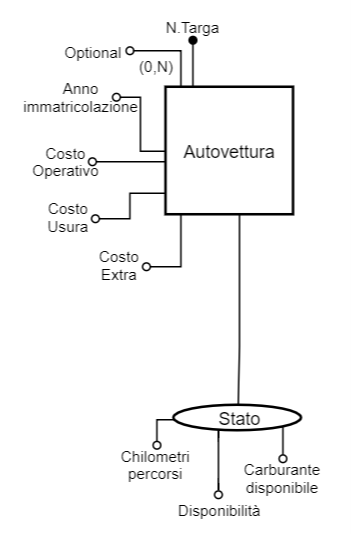
 

Il secondo attributo composto è *Stato.* Anche in questo caso l’attributo è stato suddiviso in tre attributi.



## *Traduzione di attributi multivalore*

L’attributo multivalore presente nel diagramma E-R è *Optional.* Esso è stato tramutato in una entità per via del numero elevato di informazioni che contiene.



## *Ultima ristrutturazione*

Come ultimo passaggio della ristrutturazione del diagramma E-R si sono scelti gli identificatori primari al fine di ridurre al minimo il numero di attributi da utilizzare come chiave. Per le entità ***Noleggio***, ***Pool*** e ***Sharing*** si è optato di aggiungere un nuovo attributo (Codice) al fine di facilitare le operazioni.

# ***Analisi delle prestazioni tramite identificazione delle operazioni***

*Individuazione delle operazioni più significative sulla base di dati*

Individuiamo ora le operazioni più significative sui dati, che ci permetteranno di eseguire analisi sulle prestazioni del database da noi progettato fino a questo punto, in modo da effettuare, poi, le dovute ottimizzazioni:

* Operazione 1:
  + Descrizione: Registrazione di un utente nel database
  + Frequenza: 275 volte al giorno
* Operazione 2:
  + Descrizione: Registrazione di un’autovettura
  + Frequenza: 120 volte al giorno
* Operazione 3:
  + Descrizione: Prenotazione di Car Sharing
  + Frequenza: 500 volte al giorno
* Operazione 4:
  + Descrizione: Restituzione di una vettura dopo la fruizione del Car Sharing
  + Frequenza: 500 volte al giorno
* Operazione 5:
  + Descrizione: Registrazione di un sinistro
  + Frequenza: 10 volte al giorno
* Operazione 6:
  + Descrizione: Creazione di un pool per il servizio di Car Pooling
  + Frequenza: 200 volte al giorno
* Operazione 7:
  + Descrizione: Proposta di variazione di un pool
  + Frequenza: 200 volte al giorno
* Operazione 8:
  + Descrizione: Chiamata Ride Sharing
  + Frequenza 250 volte al giorno
* Operazione 9:
  + Descrizione: Inserimento di una recensione
  + Frequenza: 900 volte al giorno
* Operazione 10:
  + Descrizione: Visualizzazione della valutazione media di un utente
  + Frequenza: 3.500 volte al giorno
* Operazione 11:
  + Descrizione: Ricerca pool ordinati per lunghezza di tragitto
  + Frequenza: 500 volte al giorno

*Tavola dei volumi*

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | E/R | Numero istanze | Motivazione |
| Account | E | 50.000 | Ipotesi iniziale |
| Appartenenza | R | 50.000 | Cardinalità (1,1) con Documento |
| Autovettura | E | 22.000 | 2/5 degli iscritti rende disponibile un veicolo, dei quali 1/10 ne registra almeno due |
| AutoCoinvolta | E | 3.780 | 9/10 dei sinistri coinvolgono 2 auto, i restanti almeno 3 |
| ChiamataSharing | E | 45.000 | Ipotesi chiamate totali |
| Chilometro | E | 750 | Presa come riferimento la lunghezza totale delle strade di una città di medie dimensioni, dove si suppone sia concentrato l’uso dell’applicazione |
| Coinvolgimento | R | 3.780 | Cardinalità (1,1) con AutoConivolta |
| CorsaSharing | R | 22.500 | Vengono accettate metà delle chiamate |
| Disponibilità | R | 66.000 | In media un’auto è disponibile per tre giorni alla settimana |
| Documento | E | 50.000 | Pari al numero di iscritti |
| Effettua | R | 45.000 | Cardinalità (1,1) con ChiamataSharing |
| Extra | R | 22.000 | Cardinalità (1,1) con Autovettura |
| Feedback | R | 162.000 | Cardinalità (1,1) con valutazione |
| Fruitore | E | 40.000 | 3/5 non registrano auto, quindi sono solo fruitori, un altro quinto hanno entrambi i ruoli, quindi 4/5 degli iscritti |
| FasciaOraria | E | 66.000 | Cardinalità (1,1) con Disponibilità |
| Incidente | E | 1.800 | Ipotesi numero di sinistri |
| Intersezione | R | 2.250 | Ogni chilometro ha in media 3 intersezioni |
| Iscrizione | R | 50.000 | Cardinalità (1,1) con Account |
| InserimentoValutazione | R | 162.000 | Cardinalità (1,1) con Valutazione |
| Modello | E | 500 | Numero ipotizzato di modelli diversi |
| Noleggio | E | 90.000 | Ipotesi noleggi totali |
| Optional | E | 22.000 | Cardinalità (1,1) con Autovettura |
| Pool | E | 36.000 | Ipotesi pool totali |
| Posizionamento | R | 750 | Cardinalità (1,1) con Chilometro |
| Posizione | E | 9.576.000 | In media tragitti di 7 km, la posizione viene presa circa 8 volte per chilometro |
| PrenotazioneN | R | 90.000 | Cardinalità (1,1) con Noleggio |
| PrenotazioneP | E | 72.000 | In media due utenti prenotano un pool |
| PrenotazionePool | R | 72.000 | Cardinalità (1,1) con PrenotazioneP |
| Proprietà | R | 22.000 | Cardinalità (1,1) con Autovettura |
| Proponente | E | 20.000 | 2/5 degli iscritti rende disponibile un veicolo |
| PropostaPool | R | 36.000 | Cardinalità (1,1) con Pool |
| PropostaVariazione | R | 36.000 | Cardinalità (1,1) con Variazione |
| Relativa | R | 9.600.000 | Si suppone che posizioni relative ad incroci generino circa 24.000 entrate in più |
| RichiestaPrenotazione | R | 72.000 | Cardinalità (1,1) con PrenotazioneP |
| RideSharing | E | 18.000 | 4.500 sono sharing multipli |
| RuoloP | R | 20.000 | Cardinalità (1,1) con Proponente |
| RuoloF | R | 40.000 | Cardinalità (1,1) con Fruitore |
| Sinistro | R | 1.800 | Cardinalità (1,1) con Incidente |
| Strada | E | 750 | In media una strada è lunga 1 km |
| Tipologia | R | 22.000 | Cardinalità (1,1) con Autovettura |
| Tracciamento | R | 9.576.000 | Cardinalità (1,1) con posizione |
| Tragitto | E | 171.000 | Ipotesi numero tragitti |
| TragittoNoleggio | R | 90.000 | Cardinalità (1,1) con Noleggio |
| TragittoPool | R | 36.000 | Cardinalità (1,1) con Pool |
| TragittoSharing | R | 18.000 | Cardinalità (1,1) con RideSharing |
| TragittoVariazione | R | 36.000 | Cardinalità (1,1) con Variazione |
| UsoNoleggio | R | 90.000 | Cardinalità (1,1) con Noleggio |
| UsoPool | R | 36.000 | Cardinalità (1,1) con Pool |
| UsoSharing | R | 18.000 | Cardinalità (1,1) con RideSharing |
| Utente | E | 50.000 | Cardinalità (1,1) con Iscrizione |
| Valutazione | E | 162.000 | Si suppone che circa al 95% dei tragitti corrisponda una valutazione |
| Variazione | E | 36.000 | Supposta almeno una richiesta di variazione in media per pool |
| VariazioneTragitto | R | 36.000 | Cardinalità (1,1) con Varizione |

1. **Registrazione nuovo account**

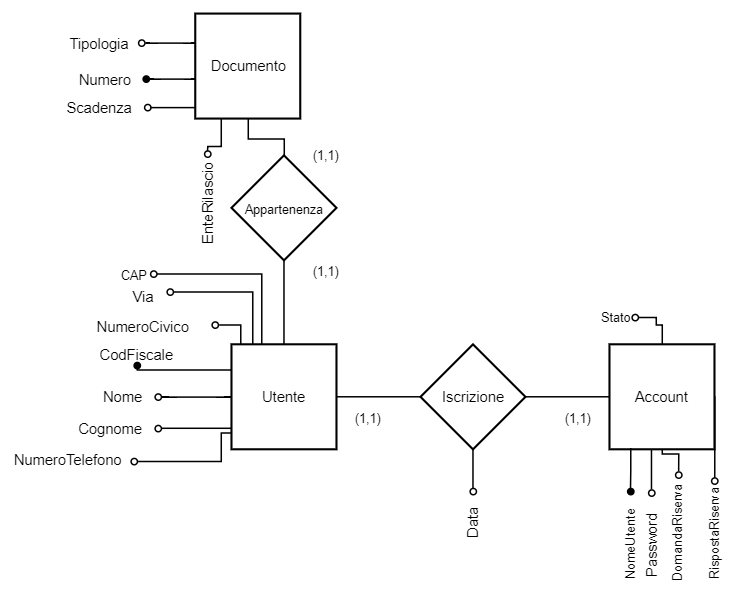
**Descrizione**: Viene creato un nuovo account inserendo le informazioni necessarie all’utilizzo dei servizi.

**Input**: Codice Fiscale, Nome, Cognome, Telefono, Indirizzo (CAP, Via, Numero Civico), Numero documento, Tipo documento, Scadenza documento, Ente di rilascio, Password, Domanda di riserva, Riposta di riserva.

**Output**: Registrazione account.

**Frequenza giornaliera**: 275.

**Parte di diagramma interessata:**

****

**Tavola dei volumi interessata:**

|  |  |  |  |
| --- | --- | --- | --- |
| Account | Entità | 50.000 | |
| Iscrizione | Relazione | 50.000 | |
| Utente | Entità | 50.000 |
| Appartenenza | Relazione | 50**.**000 | |
| Documento | Entità | 50.000 |

**Tavola degli accessi interessata:**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome Costrutto | Numero Operazioni elementari | Tipo Operazione | Descrizione |
| Account | 2 | S | Viene registrato un nuovo account |
| Iscrizione | 2 | S | Viene scritta una nuova occorrenza tra Account e Utente |
| Utente | 2 | S | Vengono salvate le generalità dell’utente |
| Appartenenza | 2 | S | Viene creata un’occorrenza tra Documento e Utente |
| Documento | 2 | S | Viene registrato il Documento dell’utente |

**Tavola delle operazioni:**

|  |  |  |
| --- | --- | --- |
| Totale operazioni elementari per singola operazione | 2+2+2+2+2+2 = 10 | |
| Totale operazioni elementari al giorno | 10 × 275 = 2.750 |

1. **Registrazione autovettura**

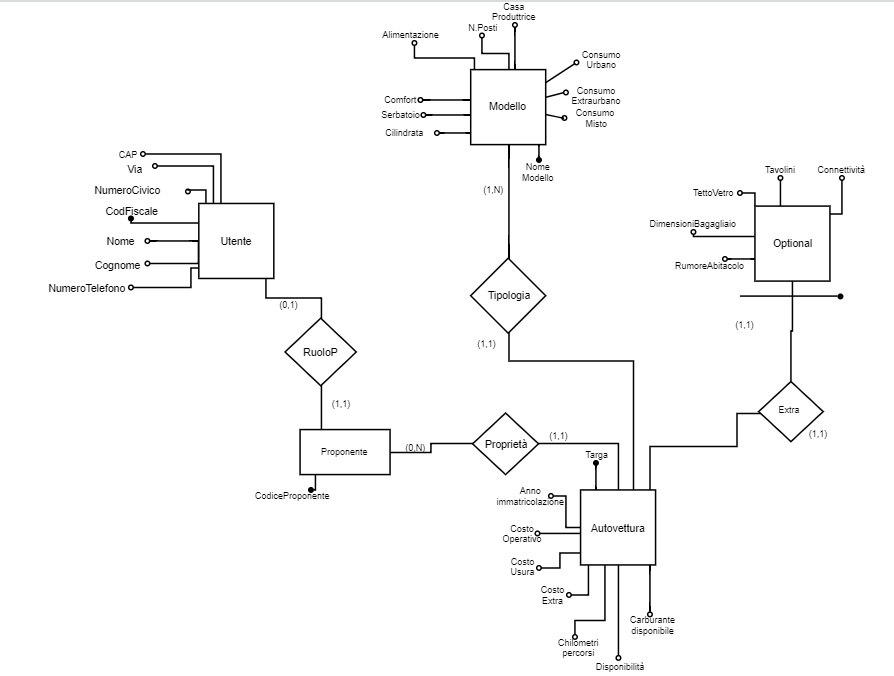
**Descrizione:** Un utente proponente registra i dati della propria autovettura, se non compare come proponente lo registra come tale.

**Input:** Codice del proponete, Targa, Anno immatricolazione, Costo Operativo, Costo Usura, Costo Extra, Carburante Disponibile, Chilometri Percorsi, Modello, Optional, Disponibilità.

**Output:** Registrazione nuova auto, eventuale registrazione dell’utente come proponente.

**Frequenza giornaliera:** 120.

**Porzione di diagramma interessata:**

****

**Porzione tavola dei volumi interessata:**

|  |  |  |
| --- | --- | --- |
| Account | Entità | 50.000 |
| RuoloP | Relazione | 20.000 |
| Proponente | Entità | 20.000 |
| Proprietà | Relazione | 22.000 |
| Autovettura | Entità | 22.000 |
| Tipologia | Relazione | 22.000 |
| Modello | Entità | 500 |
| Extra | Relazione | 22.000 |
| Optional | Entità | 22.000 |

**Tavola degli accessi:**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome Costrutto | Numero Operazioni elementari | Tipo Operazione | Descrizione |
| Proponente | 1 | L | Leggo le informazioni su Proponente |
| RuoloP | 2 | S | Creo una nuova occorrenza tra Utente e Proponente |
| Proponente | 2 | S | Inserisco l’identificatore del proponente |
| Proprietà | 2 | S | Scrivo una nuova occorrenza tra Proponente e Autovettura |
| Autovettura | 2 | S | Vengono inserite le informazioni generali riguardanti il veicolo |
| Tipologia | 2 | S | Scrivo una nuova occorrenza tra Autovettura e Modello |
| Modello | 2 | S | Vengono inserite informazioni aggiuntive del veicolo |
| Extra | 2 | S | Scrivo una nuova occorrenza tra Autovettura e Optional |
| Optional | 2 | S | Vengono inserite caratteristiche secondarie del veicolo |

**Tavola delle operazioni:**

|  |  |
| --- | --- |
| Totale operazioni elementari per singola operazione | 1+2+2+2+2+2+2+2+2 = 17 |
| Totale operazioni elementari al giorno | 17 × 120=2.040 |

1. **Prenotazione di Car Sharing**

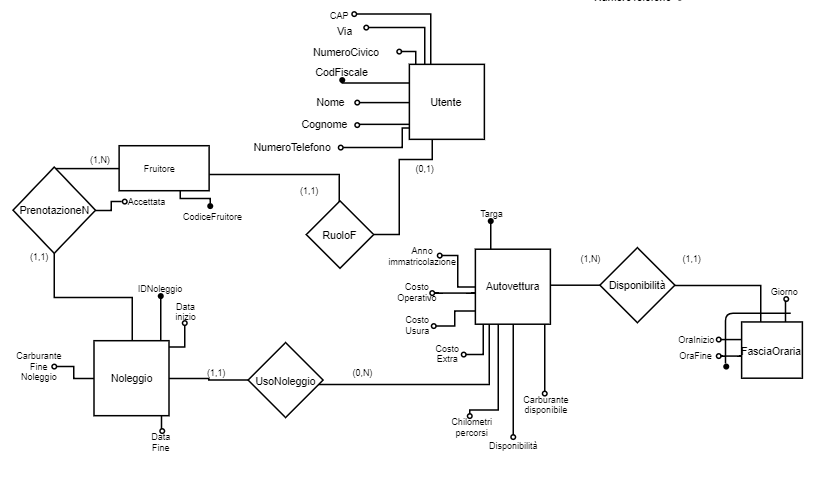
**Descrizione**: Un utente fruitore prenota un’auto disponibile in una certa fascia oraria di un determinato giorno per usufruire del Car Sharing, se non compare come fruitore lo registra come tale.

**Input**: CodiceFiscale, Data Inizio, Data Fine.

**Output**: Inserimento di una prenotazione di Car Sharing nel database, eventuale registrazione dell’utente come fruitore.

**Frequenza giornaliera**: 500.

**Parte di diagramma interessata:**

****

**Tavola dei volumi interessata:**

|  |  |  |  |
| --- | --- | --- | --- |
| Utente | Entità | 50.000 | |
| RuoloF | Fruitore | 40.000 | |
| Fruitore | Entità | 40.000 | |
| PrenotazioneN | Relazione | 90.000 |
| Noleggio | Entità | 90.000 | |
| UsoNoleggio | Relazione | 90.000 | |
| Autovettura | Entità | 22.000 |
| Disponibilità | Relazione | 66.000 |
| FasciaOraria | Entità | 66.000 |

**Tavola degli accessi interessata:**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome Costrutto | Numero Operazioni elementari | Tipo Operazione | Descrizione |
| Fruitore | 1 | L | Leggo le informazioni su Fruitore |
| RuoloF | 2 | S | Scrivo una nuova occorrenza tra Utente e Fruitore |
| Fruitore | 2 | S | Scrivo il codice identificativo del fruitore |
| PrenotazioneN | 2 | S | Associo una richiesta di noleggio ad un utente fruitore |
| Noleggio | 2 | S | Inserisco una nuova occorrenza di noleggio inserendo i dati relativi alla prenotazione |
| UsoNoleggio | 2 | S | Associo un’auto ad un noleggio |
| Autovettura | 1 | L | Leggo le informazioni dell’auto |
| Disponibilità | 1 | L | Associo la disponibilità ad un’auto |
| FasciaOraria | 1 | L | Verifico la disponibilità dell’auto |

**Tavola delle operazioni:**

|  |  |
| --- | --- |
| Totale operazioni elementari per singola operazione | 1+2+2+2+2+2+1+1+1 = 14 |
| Totale operazioni elementari al giorno | 14 × 500 = 7.000. |

1. **Restituzione di una vettura dopo la fruizione del Car Sharing**

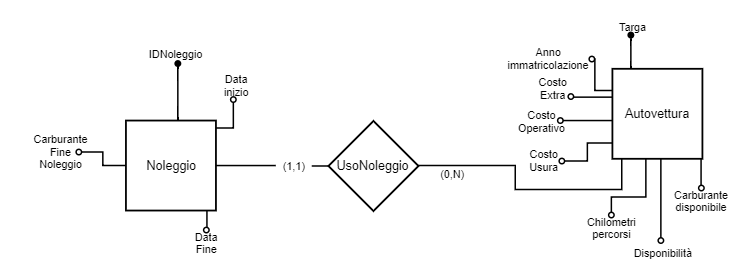
**Descrizione**: Restituzione di un’auto dopo aver usufruito del servizio di Car Sharing.

**Input**: IDNoleggio, Targa, Chilometri Percorsi, Carburante Fine Noleggio.

**Output**: Aggiornamento delle informazioni relative al Noleggio e all’Autovettura dopo il termine del servizio.

**Frequenza giornaliera**: 500 restituzioni al giorno (in media un noleggio dura meno di un giorno).

**Parte di diagramma interessata:**



**Tavola dei volumi interessata:**

|  |  |  |  |
| --- | --- | --- | --- |
| Noleggio | Entità | 90.000 | |
| UsoNoleggio | Relazione | 90.000 | |
| Autovettura | Entità | 22.000 |

**Tavola degli accessi interessata:**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome Costrutto | Numero Operazioni elementari | Tipo Operazione | Descrizione |
| Autovettura | 1 | L | Controllo il livello di carburante iniziale per stabilire se la restituzione può avvenire |
| Noleggio | 2 | S | Vengono inserite le informazioni di fine noleggio |
| Autovettura | 2 | S | Vengono aggiornati i dati relativi al veicolo noleggiato |

**Tavola delle operazioni:**

|  |  |
| --- | --- |
| Totale operazioni elementari per singola operazione | 1+2+2 = 5 |
| Totale operazioni elementari al giorno | 5 × 500 = 2.500 |

1. **Registrazione di un sinistro**

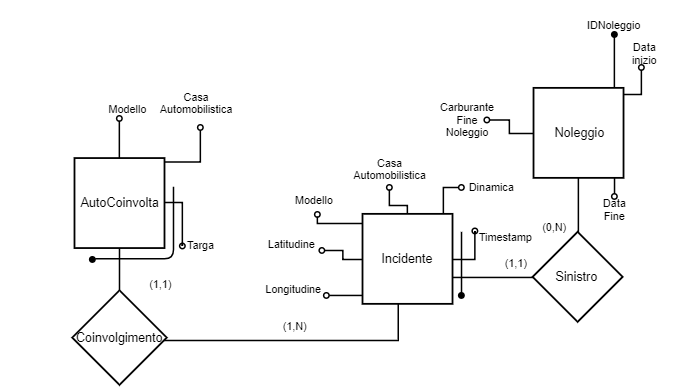
**Descrizione**: Durante il servizio di Car Sharing il fruitore commette un incidente. Vengono pertanto registrati i dati dell’auto convolta nell’incidente e le dinamiche.

**Input**: Auto Coinvolta, Incidente, IDNoleggio.

**Output**: Registrazione incidente.

**Frequenza giornaliera:** 10.

**Parte di diagramma interessata:**

****

**Tavola dei volumi interessata:**

|  |  |  |  |
| --- | --- | --- | --- |
| Noleggio | Entità | 90.000 | |
| Sinistro | Relazione | 1.800 | |
| Incidente | Entità | 1.800 |
| Coinvolgimento | Relazione | 3.780 | |
| AutoCoinvolta | Entità | 3.780 |

**Tavola degli accessi interessata:**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome Costrutto | Numero Operazioni elementari | Tipo Operazione | Descrizione |
| Sinistro | 2 | S | Creo un’occorrenza tra Noleggio e Incidente |
| Incidente | 2 | S | Inserisco le informazioni relative all’incidente |
| Coinvolgimento | 2 | S | Creo un’occorrenza tra AutoCoinvolta e Incidente |
| AutoCoinvolta | 2 | S | Inserisco le informazioni relative all’auto coinvolta nell’incidente |

**Tavola delle operazioni:**

|  |  |
| --- | --- |
| Totale operazioni elementari per singola operazione | 2+2+2+2=8 |
| Totale operazioni elementari al giorno | 8 × 10= 80 |

1. **Creazione di un pool per il servizio di Car Pooling**

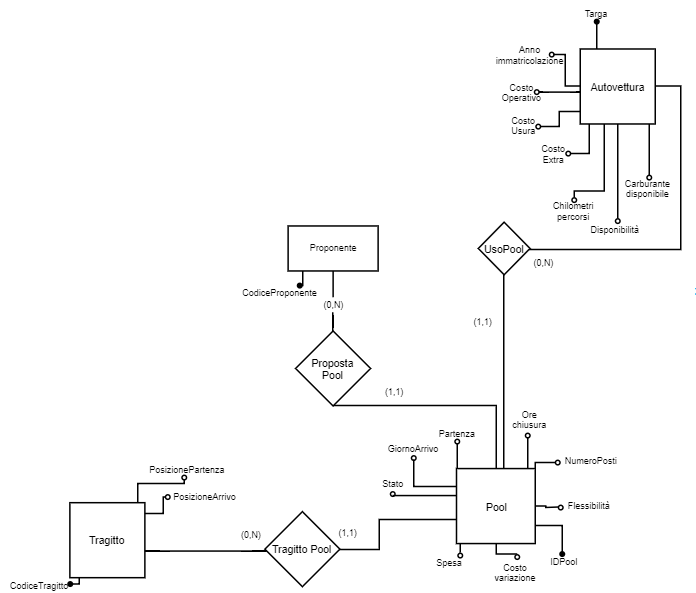
**Descrizione**: Viene creato un nuovo pool, con tutte le informazioni necessarie.

**Input**: Codice Tragitto, Targa, Codice Proponente, Partenza, Giorno di arrivo, Flessibilità, Numero Posti, Spesa.

**Output**: Creazione pool.

**Frequenza giornaliera**: 200.

**Parte di diagramma interessata:**

****

**Tavola dei volumi interessata:**

|  |  |  |
| --- | --- | --- |
| Autovettura | Entità | 22.000 |
| Proponente | Entità | 20.000 |
| PropostaPool | Relazione | 36.000 |
| UsoPool | Relazione | 36.000 |
| Pool | Entità | 36.000 |
| TragittoPool | Relazione | 36.000 |
| Tragitto | Entità | 171.000 |

**Tavola degli accessi interessata:**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome Costrutto | Numero Operazioni elementari | Tipo Operazione | Descrizione |
| PropostaPool | 2 | S | Creo una occorrenza tra Proponente e Pool |
| Pool | 2 | S | Creo un nuovo pool |
| UsoPool | 2 | S | Scrivo una nuova occorrenza tra Pool e Autovettura |
| TragittoPool | 2 | S | Creo una nuova occorrenza tra Pool e Tragitto |
| Tragitto | 2 | S | Inserisco le informazioni sul tragitto del Pool |

**Tavola delle operazioni:**

|  |  |
| --- | --- |
| Totale operazioni elementari per singola operazione | 2+2+1+2+2= 10 |
| Totale operazioni elementari al giorno | 10 × 200 = 2.000 |

1. **Proposta di variazione di un pool**

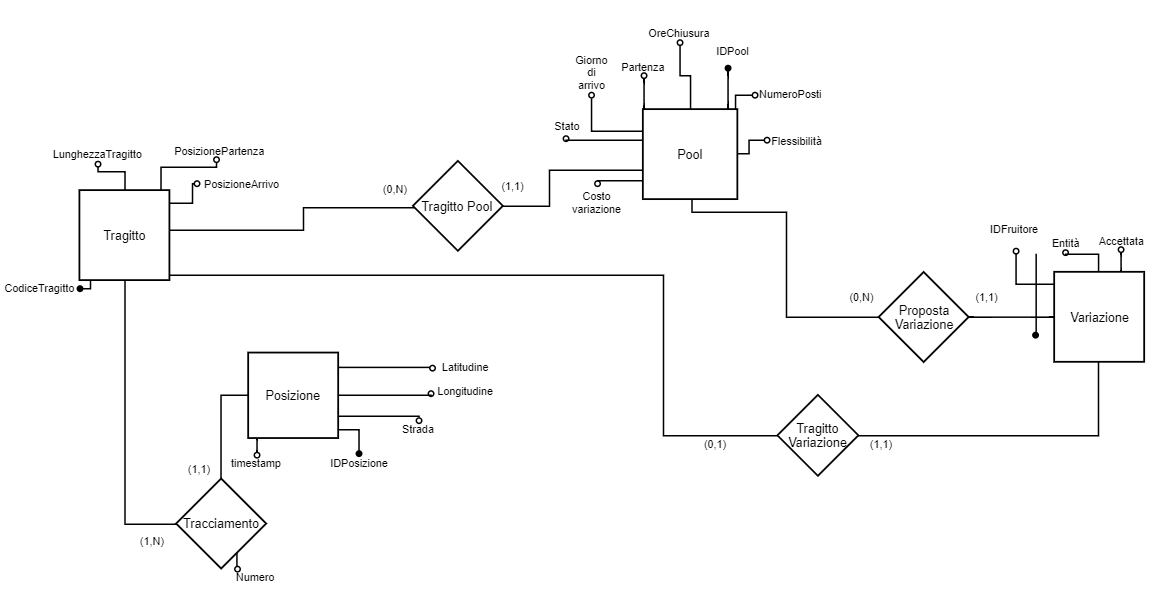
**Descrizione**: Viene inserita una proposta di variazione su un pool in stato *aperto*.

**Input**: IDFruitore, IDPool, Codice Tragitto, Entità.

**Output**: Registrazione di una proposta di variazione.

**Frequenza giornaliera**: 200.

**Parte di diagramma interessata:**

****

**Tavola dei volumi interessata:**

|  |  |  |  |
| --- | --- | --- | --- |
| Tracciamento | Relazione | 9.576.000 | |
| Posizione | Entità | 9.576.000 | |
| Pool | Entità | 36.000 | |
| PropostaVariazione | Relazione | 36.000 | |
| Tragitto | Entità | 171.000 | |
| TragittoPool | Relazione | 36.000 | |
| TragittoVariazione | Relazione | 36.000 |
| Variazione | Entità | 36.000 | |

**Tavola degli accessi interessata:**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome Costrutto | Numero Operazioni elementari | Tipo Operazione | Descrizione |
| Posizione | 1 × 56 = 56 | L | Leggo tutte le posizioni del tragitto di variazione per calcolare l’entità della variazione del tragitto (9.576.000 ÷ 171.000 = 56 posizioni in media) |
| PropostaVariazione | 2 | S | Creo un’occorrenza tra Variazione e Pool |
| Variazione | 2 | S | Scrivo le informazioni sulla variazione nel database |
| TragittoVariazione | 2 | S | Creo un’occorrenza tra Variazione e Tragitto |

**Tavola delle operazioni:**

|  |  |
| --- | --- |
| Totale operazioni elementari per singola operazione | 56+2+2+2=62 |
| Totale operazioni elementari al giorno | 62×200=12.400 |

1. **Chiamata Ride Sharing**

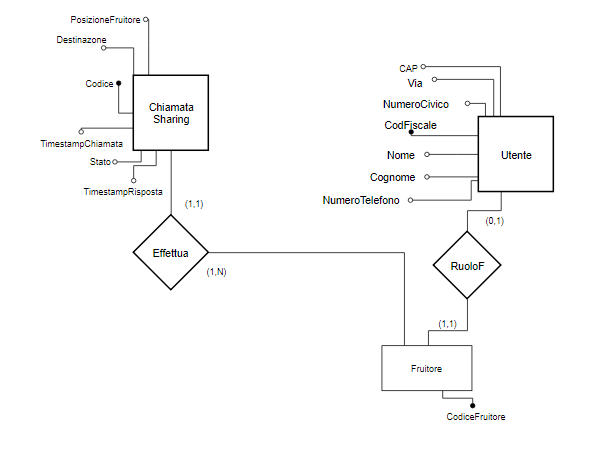
**Descrizione**: Inserimento di una chiamata di Ride Sharing, se non compare come fruitore lo registra come tale.

**Input**: PosizioneFruitore, Destinazione, TimestampChiamata, Codice, TimestampRisposta, CodiceFruitore

**Output**: Inserimento dei dati di una chiamata di Ride Sharing nel database, eventuale registrazione dell’utente come fruitore.

**Frequenza giornaliera**: 250.

**Parte di diagramma interessata:**



**Tavola dei volumi interessata:**

|  |  |  |
| --- | --- | --- |
| ChiamataSharing | Entità | 45.000 |
| Effettua | Relazione | 45.000 |
| Fruitore | Entità | 40.000 |
| Utente | Entità | 50.000 |

**Tavola degli accessi interessata:**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome Costrutto | Numero Operazioni elementari | Tipo Operazione | Descrizione |
| Fruitore | 1 | L | Leggo le informazioni su Fruitore |
| RuoloF | 2 | S | Creo un’occorrenza tra Utente e Fruitore |
| Fruitore | 2 | S | Aggiorno il ruolo dell’utente |
| ChiamataSharing | 2 | S | Scrivo informazioni sulla chiamata |
| Effettua | 2 | S | Creo una nuova occorrenza tra ChiamataSharing e Fruitore |

**Tavola delle operazioni:**

|  |  |
| --- | --- |
| Totale operazioni elementari per singola operazione | 1+2+2+2+2 = 9 |
| Totale operazioni elementari al giorno | 9 × 250 = 2.250 |

1. **Inserimento di una recensione**

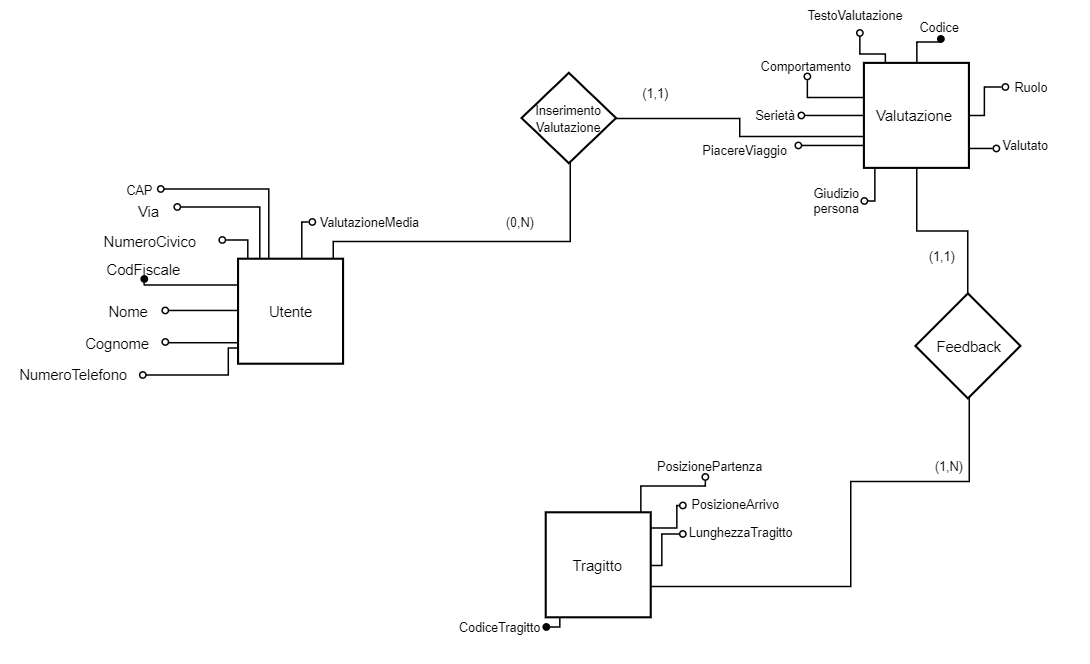
**Descrizione**: Rilascio di feedback sul tragitto.

**Input**: TestoValutazione, Codice, CodiceTragitto, CodFiscale, Valutato, Serietà, Comportamento, PiacereViaggio, GiudizioPersona, Ruolo.

**Output**: Inserimento di una recensione nel database.

**Frequenza giornaliera**: 900.

**Parte di diagramma interessata:**

****

**Tavola dei volumi interessata:**

|  |  |  |  |
| --- | --- | --- | --- |
| Feedback | Relazione | 162.000 | |
| InserimentoValutazione | Relazione | 162.000 | |
| Tragitto | Entità | 171.000 | |
| Utente | Entità | 50.000 | |
| Valutazione | Entità | 162.000 |

**Tavola degli accessi interessata:**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome Costrutto | Numero Operazioni elementari | Tipo Operazione | Descrizione |
| InserimentoValutazione | 2 | S | Creazione di un’occorrenza tra Valutazione e Utente |
| Valutazione | 2 | S | Inserimento della valutazione nel database |
| Feedback | 2 | S | Creazione di un’occorrenza tra Valutazione e Tragitto |

**Tavola delle operazioni:**

|  |  |
| --- | --- |
| Totale operazioni elementari per singola operazione | 2+2+2 = 6 |
| Totale operazioni elementari al giorno | 6 × 900 = 5.400 |

1. **Visualizzazione della valutazione media di un utente**

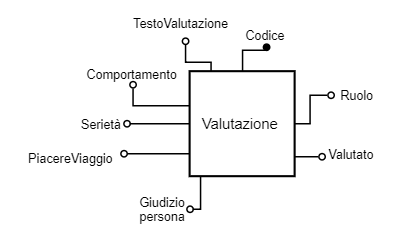
**Descrizione**: Visualizzazione della valutazione media di un utente.

**Input**: Valutato

**Output**: Valutazione media dell’utente.

**Frequenza giornaliera**: 3.500.

**Parte di diagramma interessata:**

****

**Tavola dei volumi interessata:**

|  |  |  |
| --- | --- | --- |
| Valutazione | Entità | 162.000 |

**Tavola degli accessi interessata:**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome Costrutto | Numero Operazioni elementari | Tipo Operazione | Descrizione |
| Valutazione | 1 × 3 | L | Leggo tutte le valutazioni dell’utente |

**Tavola delle operazioni:**

|  |  |
| --- | --- |
| Totale operazioni elementari per singola operazione | 3 |
| Totale operazioni elementari al giorno | 3 × 3.500 = 10.500 |

1. **Ricerca Pool ordinati per lunghezza del tragitto**

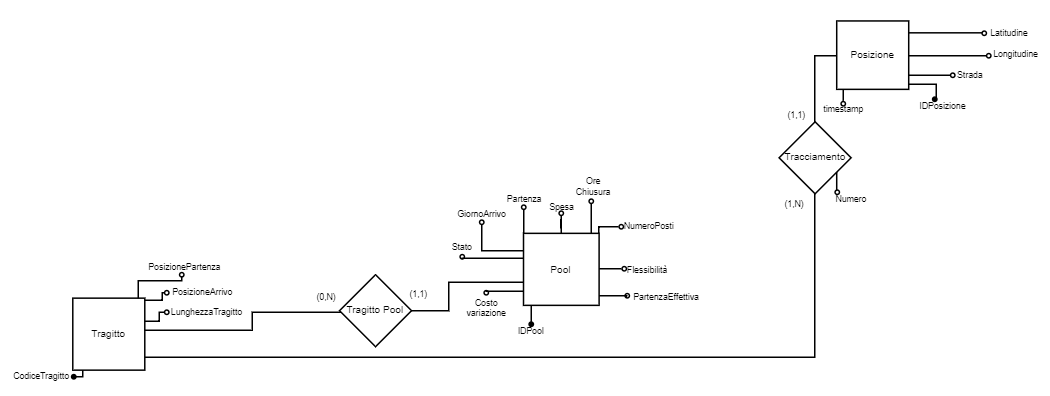
**Descrizione**: Visualizza in ordine di lunghezza i pool disponibili per un tragitto

**Input**: PosizionePartenza, PosizioneArrivo

**Output**: Visualizzazione in ordine decrescente di lunghezza dei pool disponibili.

**Frequenza giornaliera**: 500

**Parte di diagramma interessata:**

****

**Tavola dei volumi interessata:**

|  |  |  |
| --- | --- | --- |
| Posizone | Entità | 9.576.000 |
| Tracciamento | Relazione | 9.576.000 |
| Tragitto | Entità | 171.000 |
| TragittoPool | Relazione | 36.000 |
| Pool | Entita | 36.000 |

**Tavola degli accessi interessata:**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome Costrutto | Numero Operazioni elementari | Tipo Operazione | Descrizione |
| Pool | 1 × 10=10 | L | Leggo il codice del pool. Si suppone che si abbiano 10 pool in stato aperto per le posizioni date in input |
| TragittoPool | 1 × 10=10 | L | Leggo un’occorrenza tra Pool e Tragitto |
| Tragitto | 1 × 10=10 | L | Accedo alla tabella tragitto. |
| Tracciamento | 1×10×56=560 | L | Leggo una occorrenza tra Tragitto e Tracciamento |
| Posizione | 1 × 56 × 10=560 | L | Leggo tutte le posizioni del tragitto (9.576.000 ÷ 171.000 = 56 posizioni in media) |

**Tavola delle operazioni:**

|  |  |
| --- | --- |
| Totale operazioni elementari per singola operazione | 10+10+10+560+560=1.150 |
| Totale operazioni elementari al giorno | 1.150 × 500 = 575.000 |

# ***Analisi delle ridondanze***

Al fine di migliorare le prestazioni delle operazioni da noi sopra descritte andiamo ad analizzare i possibili benefici dati da l’aggiunta di attributi ridondanti.

**Ridondanza 1**

Attributo *LunghezzaTragitto:* Andiamo ad analizzare come le operazioni *‘Proposta variazione di un pool’* e ‘*Ricerca pool ordinati per lunghezza di tragitto ’* possano beneficiare dall’aggiunta dell’attributo *LunghezzaTragitto* sull’entità **Tragitto.**

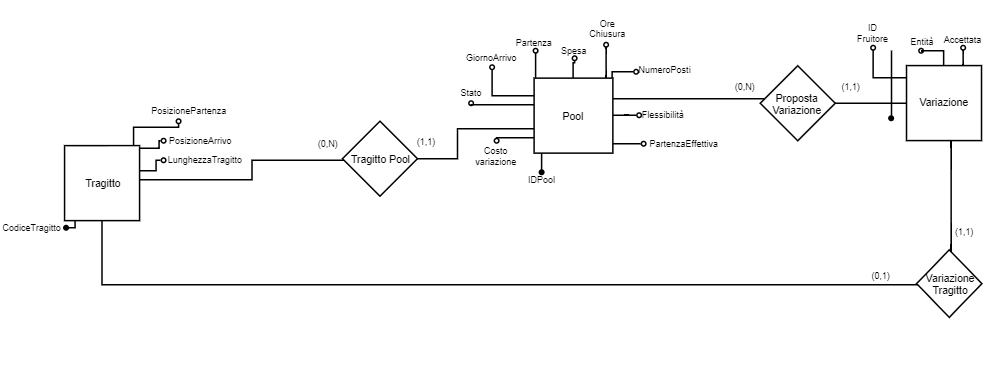
**Descrizione**: Viene inserita una proposta di variazione su un pool in stato *aperto*.

**Input**: IDFruitore, IDPool, Codice Tragitto, Entità, LunghezzaTragitto.

**Output**: Registrazione di una proposta di variazione.

**Frequenza giornaliera**: 200.

**Parte di diagramma interessata:**

****

**Tavola dei volumi interessata:**

|  |  |  |  |
| --- | --- | --- | --- |
| Tracciamento | Relazione | 9.576.000 | |
| Pool | Entità | 36.000 | |
| PropostaVariazione | Relazione | 36.000 | |
| Tragitto | Entità | 171.000 | |
| TragittoPool | Relazione | 36.000 | |
| TragittoVariazione | Relazione | 36.000 |
| Variazione | Entità | 36.000 | |

**Tavola degli accessi interessata:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome Costrutto | Numero Operazioni elementari | Tipo Operazione | Descrizione | |
| Tragitto | 1 | L | Leggo la lunghezza  della variazione del tragitto | |
| PropostaVariazione | 2 | S | Creo un’occorrenza tra Variazione e Pool | |
| Variazione | 2 | S | Scrivo le informazioni sulla variazione nel database | |
| TragittoVariazione | 2 | S | Creo un’occorrenza tra Variazione e Tragitto |

**Tavola delle operazioni:**

|  |  |
| --- | --- |
| Totale operazioni elementari per singola operazione | 1+2+2+2 = 7 |
| Totale operazioni elementari al giorno | 7 × 200 = 1.400 |

fT = 200

oT = 62

nT = fT × oT = 12.400

oTRID = 7

nTRID = fT × oTRID = 1.400

∆read = nT - nTRID = 12.400 – 1.400 = 11.000

Andiamo a vedere come l’operazione 11 possa trarre vantaggio dall’aggiunta della ridondanza.

**11)**

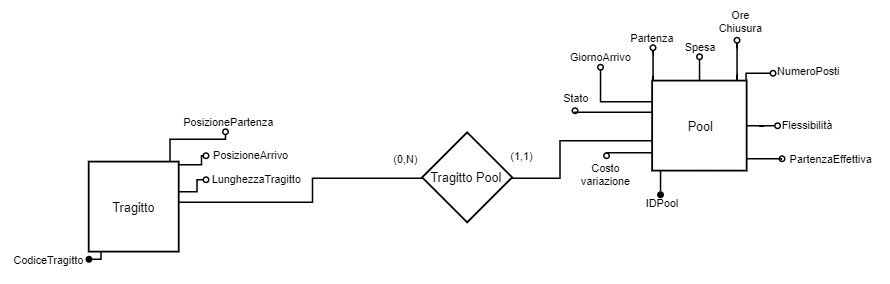
**Descrizione**: Visualizza in ordine di lunghezza i pool disponibili per un tragitto

**Input**: PosizionePartenza, PosizioneArrivo

**Output**: Visualizzazione in ordine decrescente di lunghezza dei pool disponibili.

**Frequenza giornaliera**: 500

**Parte di diagramma interessata:**

****

**Tavola dei volumi interessata:**

|  |  |  |
| --- | --- | --- |
| Tragitto | Entità | 171.000 |
| TragittoPool | Relazione | 36.000 |
| Pool | Entita | 36.000 |

**Tavola degli accessi interessata:**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome Costrutto | Numero Operazioni elementari | Tipo Operazione | Descrizione |
| Pool | 1 × 10=10 | L | Leggo il codice del pool. Si suppone che si abbiano 10 pool in stato aperto per le posizioni date in input |
| TragittoPool | 1 × 10=10 | L | Leggo un’occorrenza tra Pool e Tragitto |
| Tragitto | 1 × 10=10 | L | Accedo alla tabella tragitto. |

**Tavola delle operazioni:**

|  |  |
| --- | --- |
| Totale operazioni elementari per singola operazione | 10+10+10=30 |
| Totale operazioni elementari al giorno | 30 × 500 = 15.000 |

fT = 500

oT = 1.150

nT = fT × oT = 575.000

oTRID = 30

nTRID = fT × oTRID = 15.000

∆read = nT - nTRID = 575.000 – 15.000 = 560.000

**Operazione di aggiornamento:** aggiorno la lunghezza di un tragitto ogni volta che viene inserita una nuova posizione per quel tragitto.

**Tavola degli accessi interessata:**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome Costrutto | Numero Operazioni elementari | Tipo Operazione | Descrizione |
| Posizione | 1 × 2 = 2 | L | Leggo le due ultime posizioni che formano il tragitto |
| Tracciamento | 1 | L | Leggo un’occorrenza tra Posizione e Tragitto |
| Tragitto | 2 | S | Aggiorno la lunghezza del tragitto. |

**Tavola delle operazioni:**

|  |  |
| --- | --- |
| Totale operazioni elementari per singola operazione | 2+1+2=5 |
| Totale operazioni elementari al giorno | 5 × (950 × 56) = 266.000 |

fT = 53.200

oT = 5

nT = fT × oT = 266.000

L’operazione di aggiornamento è stata implementata con un trigger.

|  |  |  |
| --- | --- | --- |
| Senza Ridondanza 2 |  | |
| Operazione | **Numero operazioni elementari al giorno** | |
| Operazione 7 | 12.400 |
| Operazione 11 | 575.000 |
| Totale | 587.400 |

|  |  |  |
| --- | --- | --- |
| Con Ridondanza 1 |  | |
| Operazione | **Numero operazioni elementari al giorno** | |
| Operazione 7 | 1.400 |
| Operazione 11 | 15.000 |
| Operazione di aggiornamento | 266.000 |
| Totale | 282.400 |

Con l’aggiunta dell’attributo *LunghezzaTragitto* è possibile determinare il valore dell’attributo *Entità* con un numero molto inferiore di operazioni rispetto al caso noi non avessimo la ridondanza. Infatti, nel caso noi operassimo senza la ridondanza sarebbe necessario leggere tutte le posizioni che compongono il tragitto proposto come variazione al fine di calcolare la lunghezza della variazione.

Notiamo che gli accessi totali diminuiscono complessivamente di 587.400 – 282.400 = 305.000 e ciò è più che sufficiente per stabilire che la ridondanza è stata introdotta efficacemente.

**Ridondanza 2**

Attributo *ValutazioneMedia*: Analizziamo come l’aggiunta dell’attributo *ValutazioneMedia* sull’entità ***Utente*** possa migliorare l’efficienza dell’operazione 10.

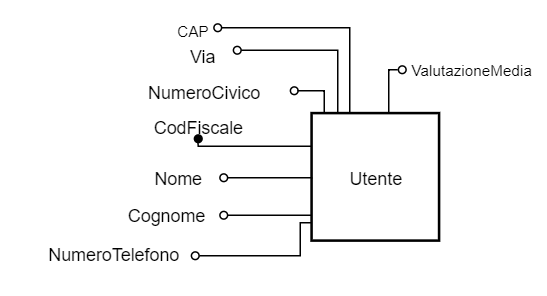
**Descrizione**: Visualizzazione della valutazione media di un utente.

**Input**: Utente.

**Output**: Valutazione media dell’utente.

**Frequenza giornaliera**: 3.500.

**Parte di diagramma interessata:**

****

**Tavola dei volumi interessata:**

|  |  |  |
| --- | --- | --- |
| Utente | Entità | 50.000 |

**Tavola degli accessi interessata:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nome Costrutto | Numero Operazioni elementari | Tipo Operazione | | Descrizione | | |
| Utente | 1 | | L | | Leggo la valutazione media |

**Tavole delle operazioni:**

|  |  |
| --- | --- |
| Totale operazioni elementari per singola operazione | 1 |
| Totale operazioni elementari al giorno | 1 × 3.500 = 3.500 |

fT = 3.500

oT = 3

nT = fT × oT = 10.500

oTRID = 1

nTRID = fT × oTRID = 3.500

∆read = nT - nTRID = 10.500 – 3.500 = 7.000

L’aggiunta dell’attributo *ValutazioneMedia* ci permette di ottenere immediatamente la valutazione complessiva dell’utente. L’attributo appena introdotto deve essere aggiornato ogni qualvolta sarà aggiunta una valutazione. Pertanto dobbiamo andare a modificare l’operazione 9.

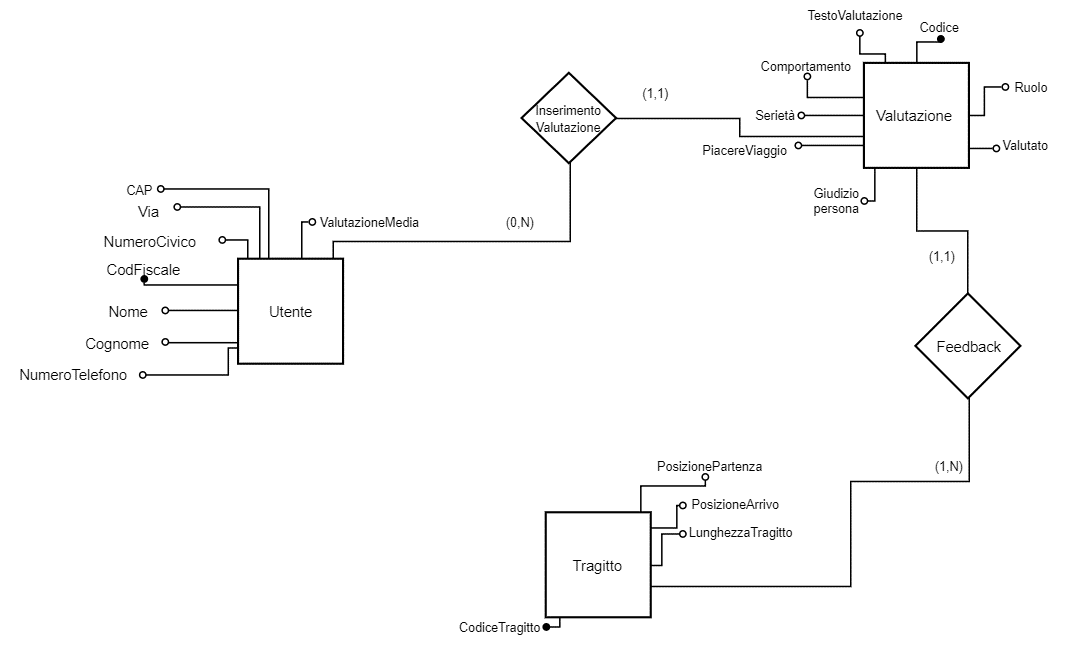
**Descrizione**: Rilascio di feedback sul tragitto.

**Input**: TestoValutazione, Codice, Valutante, Valutato, Serietà, Comportamento, PiacereViaggio, GiudizioPersona.

**Output**: Inserimento di una recensione nel database e aggiornamento della valutazione media dell’utente valutato.

**Frequenza giornaliera**: 900.

**Parte di diagramma interessata:**



**Tavola dei volumi interessata:**

|  |  |  |  |
| --- | --- | --- | --- |
| Feedback | Relazione | 162.000 | |
| InserimentoValutazione | Relazione | 162.000 | |
| Tragitto | Entità | 171.000 | |
| Utente | Entità | 50.000 | |
| Valutazione | Entità | 162.000 |

**Tavola degli accessi interessata:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nome Costrutto | Numero Operazioni elementari | | Tipo Operazione | | Descrizione | |
| InserimentoValutazione | | 2 | | S | | Creazione di un’occorrenza tra Valutazione e Utente |
| Valutazione | | 2 | | S | | Inserimento della valutazione nel database |
| Feedback | | 2 | | S | | Creazione di un’occorrenza tra Valutazione e Tragitto |
| Utente | | 2 | | S | | Aggiorno l’attributo ValutazioneMedia dell’utente valutato |

**Tavole delle operazioni:**

|  |  |
| --- | --- |
| Totale operazioni elementari per singola operazione | 2+2+2+2=8 |
| Totale operazioni elementari al giorno | 8 × 900 = 7.200 |

fT = 900

oT = 6

nT = fT × oT = 5.400

oTRID = 8

nTRID = fT × oTRID = 7.200

∆read = nT - nTRID = 5.400 – 7.200 = -1.800

|  |  |
| --- | --- |
| Senza Ridondanza 2 |  |
| Operazione | **Numero operazioni elementari al giorno** |
| Operazione 9 | 5.400 |
| Operazione 10 | 10.500 |
| Totale | 15.900 |

|  |  |
| --- | --- |
| Con Ridondanza 2 |  |
| Operazione | **Numero operazioni elementari al giorno** |
| Operazione 9 | 7.200 |
| Operazione 10 | 3.500 |
| Totale | 10.700 |

Notiamo che sebbene l’operazione 9 sia influenzata negativamente dalla ridondanza a causa dell’operazione di aggiornamento che è chiamata a svolgere, gli accessi totali diminuiscono complessivamente di 15.900 - 10.700 = 5.200 e ciò è più che sufficiente per stabilire che la ridondanza è stata introdotta efficacemente.

# ***Modello relazionale***

Andiamo adesso a tradurre il nostro diagramma E-R in uno schema relazionale, ovvero traduciamo le entità e le relazioni in tabelle.

* UTENTE (CodFiscale, Nome, Cognome, NumeroTelefono, NumeroCivico, Via, CAP, ValutazioneMedia)
* ACCOUNT (NomeUtente, Data, CodFiscale, Password, DomandaRiserva, RispostaRiserva, Stato)
* DOCUMENTO (Numero, CodFiscale, Tipologia, Scadenza, EnteRilascio)
* FRUITORE (CodiceFruitore, CodFiscale)
* PROPONENTE (CodiceProponente, CodFiscale)
* VALUTAZIONE (Codice, CodiceTragitto, CodFiscale, TestoValutazione, Comportamento, Serieta, PiacereViaggio, GiudizioPersona, Ruolo, Valutato)
* AUTOVETTURA (Targa, CodiceProponente, AnnoImmatricolazione, CostoOperativo, CostoUsura, CostoExtra, ChilometriPercorsi, Disponibilita, CarburanteDisponibile, NomeModello)
* OPTIONAL (Targa, Connettivita, Tavolini, TettoVetro, DimensioneBagagliaio, RumoreAbitacolo)
* CHIAMATASHARING (Codice, CodiceFruitore, PosizioneFruitore, Destinazione, TimestampChiamata, Stato, TimestampRisposta)
* PRENOTAZIONEN (CodiceFruitore, IDNoleggio, Accettata)
* NOLEGGIO (IDNoleggio, CodiceFruitore, Targa, DataInizio, CarburanteFineNoleggio, DataFine, CodiceTragitto)
* INCIDENTE (IDNoleggio, Timestamp, Dinamica, CasaAutomobilistica, Modello, Latitudine, Longitudine)
* AUTOCOINVOLTA (Targa,IDNoleggio,Timestamp, CasaAutomobilistica, Modello)
* RIDESHARING (IDSharing, CodiceTragitto, Targa, OrarioArrivo, OrarioPartenza, PartenzaEffettiva)
* CORSASHARING (IDSharing, Codice)
* PRENOTAZIONEP (CodiceFruitore, IDPool, CodicePrenotazione)
* MODELLO (NomeModello, ConsumoMisto, ConsumoUrbano, ConsumoExtraurbano, CasaProdruttrice, NPosti, Alimentazione, Comfort, Serbatoio, Cilindrata,VelocitaMax)
* FASCIAORARIA (Targa, OraInizio, OraFine, Giorno)
* POOL (IDPool, CodiceTragitto, CodiceProponente, Targa, Flessibilita, NumeroPosti, OreChiusura, Spesa, Partenza, GiornoArrivo, Stato, CostoVariazione, PartenzaEffettiva)
* VARIAZIONE (IDPool, IDFruitore, Accettata, Entita, CodiceTragitto)
* TRAGITTO (CodiceTragitto, LunghezzaTragitto, PosizioneArrivo, PosizionePartenza)
* POSIZIONE (IDPosizione, Strada, Latitudine, Longitudine, Timestamp, Numero, CodiceTragitto)
* RELATIVA (IDPosizione, Numero, IDStrada)
* CHILOMETRO (IDStrada, Numero, Latitudine, Longitudine, LimiteVelocita, Pedaggio, TempoMedio)
* INTERSEZIONE (Numero1, IDStrada1, Numero2, IDStrada2)
* STRADA (IDStrada, Tipologia, Nome, TipoStrada, Lunghezza, NCarreggiate, CorsieCarreggiata, NSensiMarcia, CaratterizzazioniAggiuntive)

# ***Vincoli di integrità***

Vengono elencati qui di seguito i vincoli di integrità referenziali.

* Esiste un vincolo di integrità referenziale tra l’attributo CodFiscale della tabella UTENTE e l’attributo omonimo della tabella ACCOUNT
* Esiste un vincolo di integrità referenziale tra l’attributo CodFiscale della tabella DOCUMENTO e l’attributo omonimo della tabella UTENTE
* Esiste un vincolo di integrità referenziale tra l’attributo CodFiscale della tabella FRUITORE e l’attributo omonimo della tabella UTENTE
* Esiste un vincolo di integrità referenziale tra l’attributo CodFiscale della tabella PROPONENTE e l’attributo omonimo della tabella UTENTE
* Esiste un vincolo di integrità referenziale tra l’attributo CodFiscale della tabella VALUTAZIONE e l’attributo omonimo della tabella UTENTE
* Esiste un vincolo di integrità referenziale tra l’attributo CodiceTragitto della tabella VALUTAZIONE e l’attributo omonimo della tabella TRAGITTO
* Esiste un vincolo di integrità referenziale tra l’attributo CodiceProponente della tabella AUTOVETTURA e l’attributo omonimo della tabella PROPONENTE
* Esiste un vincolo di integrità referenziale tra l’attributo Targa della tabella OPTIONAL e l’attributo omonimo della tabella AUTOVETTURA
* Esiste un vincolo di integrità referenziale tra l’attributo CodiceFruitore della tabella CHIAMATASHARING e l’attributo omonimo della tabella FRUITORE
* Esiste un vincolo di integrità referenziale tra l’attributo CodiceFruitore della tabella PRENOTAZIONEN e l’attributo omonimo della tabella FRUITORE
* Esiste un vincolo di integrità referenziale tra l’attributo IDNoleggio della tabella PRENOTAZIONEN e l’attributo omonimo della tabella NOLEGGIO
* Esiste un vincolo di integrità referenziale tra l’attributo Targa della tabella NOLEGGIO e l’attributo omonimo della tabella AUTOVETTURA
* Esiste un vincolo di integrità referenziale tra l’attributo IDNoleggio della tabella INCIDENTE e l’attributo omonimo della tabella NOLEGGIO
* Esiste un vincolo di integrità referenziale tra l’attributo Codice della tabella CORSASHARING e l’attributo omonimo della tabella CHIAMATASHARING
* Esiste un vincolo di integrità referenziale tra l’attributo IDSharing della tabella CORSASHARING e l’attributo omonimo della tabella RIDESHARING
* Esiste un vincolo di integrità referenziale tra l’attributo CodiceTragitto della tabella RIDESHARING e l’attributo omonimo della tabella TRAGITTO
* Esiste un vincolo di integrità referenziale tra l’attributo Targa della tabella RIDESHARING e l’attributo omonimo della tabella AUTOVETTURA
* Esiste un vincolo di integrità referenziale tra l’attributo CodiceFruitore della tabella PRENOTAZIONEP e l’attributo omonimo della tabella FRUITORE
* Esiste un vincolo di integrità referenziale tra l’attributo IDPool della tabella PRENOTAZIONEP e l’attributo omonimo della tabella POOL
* Esiste un vincolo di integrità referenziale tra l’attributo NomeModello della tabella AUTOVETTURA e l’attributo omonimo della tabella MODELLO
* Esiste un vincolo di integrità referenziale tra l’attributo Targa della tabella GIORNO e l’attributo omonimo della tabella AUTOVETTURA
* Esiste un vincolo di integrità referenziale tra l’attributo Targa della tabella POOL e l’attributo omonimo della tabella AUTOVETTURA
* Esiste un vincolo di integrità referenziale tra l’attributo CodiceTragitto della tabella POOL e l’attributo omonimo della tabella TRAGITTO
* Esiste un vincolo di integrità referenziale tra l’attributo CodiceProponente della tabella POOL e l’attributo omonimo della tabella PROPONENTE
* Esiste un vincolo di integrità referenziale tra l’attributo IDPool della tabella VARIAZIONE e l’attributo omonimo della tabella POOL
* Esiste un vincolo di integrità referenziale tra l’attributo CodiceTragitto della tabella VARIAZIONE e l’attributo omonimo della tabella TRAGITTO
* Esiste un vincolo di integrità referenziale tra l’attributo CodiceTragitto della tabella NOLEGGIO e l’attributo omonimo della tabella TRAGITTO
* Esiste un vincolo di integrità referenziale tra l’attributo CodiceTragitto della tabella POSIZIONE e l’attributo omonimo della tabella TRAGITTO
* Esiste un vincolo di integrità referenziale tra l’attributo IDPosizione della tabella RELATIVA e l’attributo omonimo della tabella POSIZIONE
* Esiste un vincolo di integrità referenziale tra la coppia di attributi Numero e IDStrada della tabella RELATIVA e i due attributi omonimi della tabella CHILOMETRO
* Esiste un vincolo di integrità referenziale tra l’attributo IDStrada della tabella CHILOMETRO e l’attributo omonimo della tabella STRADA
* Esiste un vincolo di integrità referenziale tra la coppia di attributi Numero1 e IDStrada1 della tabella INTERSEZIONE e la coppia di attributi Numero e IDStrada della tabella CHILOMETRO
* Esiste un vincolo di integrità referenziale tra la coppia di attributi Numero2 e IDStrada2 della tabella INTERSEZIONE e la coppia di attributi Numero e IDStrada della tabella CHILOMETRO
* Esiste un vincolo di integrità referenziale tra la coppia di attributi IDNoleggio e Timestamp della tabella AUTOCOINVOLTA e i due attributi omonimi della tabella INCIDENTE

## *Vincoli generici*

* L’entità della variazione relativa ad un pool deve essere minore o uguale alla flessibilità espressa dal proponente.
* Un fruitore non può prenotare un pool o richiedere variazioni di un pool se è nello stato *chiuso.*
* Il documento inserito in fase di registrazione non deve essere scaduto.
* Una variazione deve avere il punto di arrivo coincidente con il punto di partenza.
* Un’auto non può essere restituita al termine del servizio di Car Sharing finché il carburante finale non è almeno il 95% del carburante iniziale.

# ***Normalizzazione***

Vengono qui riportate le dipendenze funzionali sopra identificate con il relativo studio della forma normale.

**Tabella UTENTE:**

UTENTE (CodFiscale, Nome, Cognome, NumeroTelefono, NumeroCivico, Via, CAP, ValutazioneMedia)

* CodFiscale →Nome, Cognome, NumeroTelefono, NumeroCivico, Via, CAP, ValutazioneMedia

Poiché la parte sinistra è superchiave, UTENTE è in BCNF.

**Tabella ACCOUNT:**

ACCOUNT (NomeUtente, Data, CodFiscale, Password, DomandaRiserva, RispostaRiserva, Stato)

* NomeUtente → Data, CodFiscale, Password, DomandaRiserva, RispostaRiserva, Stato
* CodFiscale → NomeUtente, Data, Password, DomandaRiserva, RispostaRiserva, Stato

Poiché la parte sinistra è superchiave, ACCOUNT è in BCNF.

**Tabella DOCUMENTO:**

DOCUMENTO (Numero, CodFiscale, Tipologia, Scadenza, EnteRilascio)

* Numero → CodFiscale, Tipologia, Scadenza, EnteRilascio

Poiché la parte sinistra è superchiave, DOCUMENTO è in BCNF.

**Tabella FRUITORE:**

FRUITORE (CodiceFruitore, CodFiscale)

* CodiceFruitore → CodFiscale
* CodFiscale → CodiceFruitore

Poiché la parte sinistra è superchiave, FRUITORE è in BCNF.

**Tabella PROPONENTE:**

PROPONENTE (CodiceProponente, CodFiscale)

* CodiceProponente → CodFiscale
* CodFiscale → CodiceProponente

Poiché la parte sinistra è superchiave, PROPONENTE è in BCNF.

**Tabella VALUTAZIONE:**

VALUTAZIONE (Codice, CodiceTragitto, CodFiscale, TestoValutazione, Comportamento, Serieta, PiacereViaggio, GiudizioPersona, Ruolo, Valutato)

* Codice → CodiceTragitto, CodFiscale, TestoValutazione, Comportamento, Serieta, PiacereViaggio, GiudizioPersona, Ruolo, Valutato

Poiché la parte sinistra è superchiave, VALUTAZIONE è in BCNF.

**Tabella AUTOVETTURA:**

AUTOVETTURA (Targa, CodiceProponente, AnnoImmatricolazione, CostoOperativo, CostoUsura, CostoExtra, ChilometriPercorsi, Disponibilita, CarburanteDisponibile, NomeModello)

* Targa → CodiceProponente, AnnoImmatricolazione, CostoOperativo, CostoUsura, CostoExtra, ChilometriPercorsi, Disponibilita, CarburanteDisponibile, NomeModello

Poiché la parte sinistra è superchiave, AUTOVETTURA è in BCNF.

**Tabella OPTIONAL:**

OPTIONAL (Targa, Connettivita, Tavolini, TettoVetro, DimensioneBagagliaio, RumoreAbitacolo)

* Targa → Connettivita, Tavolini, TettoVetro, DimensioneBagagliaio, RumoreAbitacolo

Poiché la parte sinistra è superchiave, OPTIONAL è in BCNF.

**Tabella CHIAMATASHARING:**

CHIAMATASHARING (Codice, CodiceFruitore, PosizioneFruitore, Destinazione, TimestampChiamata, Stato, TimestampRisposta)

* Codice → CodiceFruitore, PosizioneFruitore, Destinazione, TimestampChiamata, Stato, TimestampRisposta

Poiché la parte sinistra è superchiave, CHIAMATASHARING è in BCNF.

**Tabella PRENOTAZIONEN:**

PRENOTAZIONEN (CodiceFruitore, IDNoleggio, Accettata)

* CodiceFruitore, IDNoleggio → Accettata

Poiché la parte sinistra è superchiave, PRENOTAZIONEN è in BCNF.

**Tabella NOLEGGIO:**

NOLEGGIO (IDNoleggio, CodiceFruitore, Targa, DataInizio, CarburanteFineNoleggio, DataFine, CodiceTragitto)

* IDNoleggio → CodiceFruitore, Targa, DataInizio, CarburanteFineNoleggio, DataFine, CodiceTragitto

Poiché la parte sinistra è superchiave, NOLEGGIO è in BCNF.

**Tabella INCIDENTE:**

INCIDENTE (IDNoleggio, Timestamp, Dinamica, Latitudine, Longitudine)

* IDNoleggio, Timestamp → Dinamica, Latitudine, Longitudine

Poiché la parte sinistra è superchiave, INCIDENTE è in BCNF.

**Tabella AUTOCOINVOLTA:**

AUTOCOINVOLTA (Targa,IDNoleggio,Timestamp, CasaAutomobilistica, Modello)

* Targa,IDNoleggio,Timestamp → CasaAutomobilistica, Modello

Poiché la parte sinistra è superchiave, AUTOCOINVOLTA è in BCNF.

**Tabella RIDESHARING:**

RIDESHARING (IDSharing, CodiceTragitto, Targa, OrarioArrivo, OrarioPartenza)

* IDSharing → CodiceTragitto, Targa, OrarioArrivo, OrarioPartenza, PartenzaEffettiva

Poiché la parte sinistra è superchiave, RIDESHARING è in BCNF.

**Tabella CORSASHARING:**

CORSASHARING (IDSharing, Codice)

Poiché non esistono dipendenze funzionali, CORSASHARING è in BCNF.

**Tabella PRENOTAZIONEP:**

PRENOTAZIONEP (CodiceFruitore, IDPool, CodicePrenotazione)

Poiché non esistono dipendenze funzionali, PRENOTAZIONEP è in forma BCNF.

**Tabella MODELLO:**

MODELLO (NomeModello, ConsumoMisto, ConsumoUrbano, ConsumoExtraurbano, CasaProdruttrice, NPosti, Alimentazione, Comfort, Serbatoio, Cilindrata, VelocitaMax)

* NomeModello → ConsumoMisto, ConsumoUrbano, ConsumoExtraurbano, CasaProdruttrice, NPosti, Alimentazione, Comfort, Serbatoio, Cilindrata,VelocitaMax

Poiché la parte sinistra è superchiave, MODELLO è in BCNF.

**Tabella FASCIAORARIA:**

FASCIAORARIA (Targa, OraInizio, OraFine, Giorno)

Poiché non esistono dipendenze funzionali, FASCIAORARIA è in BCNF.

**Tabella POOL:**

POOL (IDPool, CodiceTragitto, CodiceProponente, Targa, Flessibilita, NumeroPosti, OreChiusura, Spesa, Partenza, GiornoArrivo, Stato, CostoVariazione)

* IDPool → CodiceTragitto, CodiceProponente, Targa, Flessibilita, NumeroPosti, OreChiusura, Spesa, Partenza, GiornoArrivo, Stato, CostoVariazione, PartenzaEffettiva

Poiché la parte sinistra è superchiave, POOL è in BCNF.

**Tabella VARIAZIONE:**

VARIAZIONE (IDPool, IDFruitore, Accettata, Entita, CodiceTragitto)

* IDPool, IDFruitore → Accettata, Entita, CodiceTragitto

Poiché la parte sinistra è superchiave, VARIAZIONE è in BCNF.

**Tabella AUTOVETTURA:**

TRAGITTO (CodiceTragitto, LunghezzaTragitto, PosizioneArrivo, PosizionePartenza)

* CodiceTragitto → LunghezzaTragitto, PosizioneArrivo, PosizionePartenza

Poiché la parte sinistra è superchiave, AUTOVETTURA è in BCNF.

**Tabella POSIZIONE:**

POSIZIONE (IDPosizione, Strada, Longitudine, Latitudine, Timestamp, Numero, CodiceTragitto)

* IDPosizione → Strada, Longitudine, Latitudine, Timestamp, Numero, CodiceTragitto

Poiché la parte sinistra è superchiave, POSIZIONE è in BCNF.

**Tabella RELATIVA:**

RELATIVA (IDPosizione, Numero, IDStrada)

Poiché non esistono dipendenze funzionali, RELATIVA è in BCNF.

**Tabella CHILOMETRO:**

CHILOMETRO (IDStrada, Numero, Latitudine, Longitudine, LimiteVelocita, Pedaggio, TempoMedio)

* IDStrada, Numero → Latitudine, Longitudine, LimiteVelocita, Pedaggio, TempoMedio

Poiché la parte sinistra è superchiave, CHILOMETRO è in BCNF.

**Tabella INTERSEZIONE:**

INTERSEZIONE (Numero1, IDStrada1, Numero2, IDStrada2)

Poiché non esistono dipendenze funzionali, RELATIVA è in BCNF.

**Tabella STRADA:**

STRADA (IDStrada, Tipologia, Nome, TipoStrada, Lunghezza, NCarreggiata, CorsieCarreggiata, NSensiMarcia, CaratterizzazioniAggiuntive)

* IDStrada → Tipologia, Nome, TipoStrada, Lunghezza, NCarreggiata, CorsieCarreggiata, NSensiMarcia, CaratterizzazioniAggiuntive

Poiché la parte sinistra è superchiave, STRADA è in BCNF.

# ***Area Analytics***

L’ultima parte della progettazione consiste nell’andare ad implementare possibili funzionalità lato server che ci permettono di andare a migliorare i servizi offerti. Andremo a focalizzarci su due funzionalità, la prima relativa all’affidabilità di un utente, la seconda relativa ai tempi di percorrenza di una strada o parte di strada.

## *Affidabilità di un utente*

L’affidabilità di un utente si basa sui feedback ricevuti dagli utenti, sul rispetto degli orari prestabiliti e sul rispetto dei limiti di velocità.

Il grado di affidabilità di un utente viene salvato in una parte del database tramite l’implementazione di una materialized view ***MV\_Affidabilita***.

Per facilitare l’operazione abbiamo deciso di aggiungere due attributi entrambi chiamati *PartenzaEffettiva*, uno su Pool e uno su RideSharing. Entrambi gli attributi hanno la funzione di tenere traccia dell’effettivo orario di inizio dei rispettivi servizi. L’indice di puntualità è stato calcolato facendo la divisione tra il totale dei due servizi partiti puntuali e il totale dei due servizi erogati dal proponente. L’indice di puntualità è un valore compreso tra 0 e 1.

* AFFIDABILITA (CodFiscale, Affidabilita)

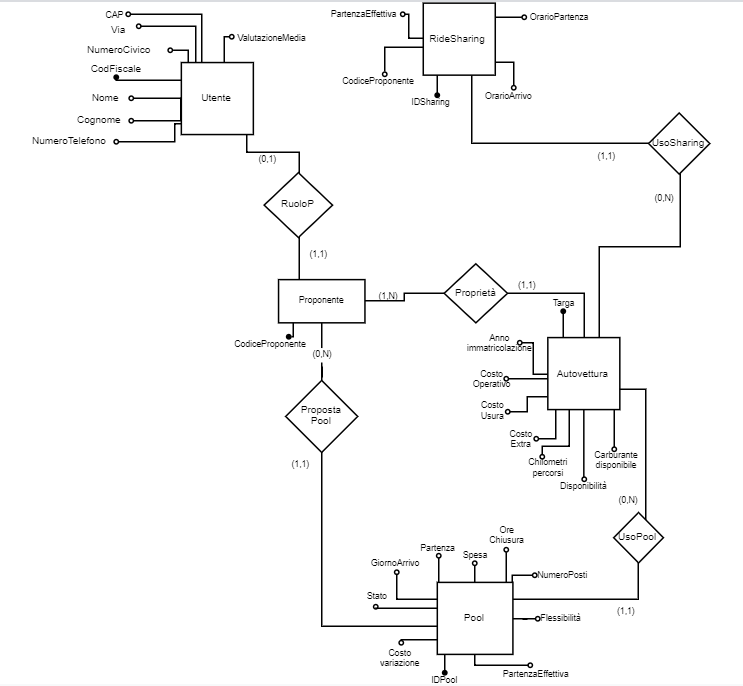
**Grado di affidabilità=** (ValutazioneMedia + IndicePuntualità × 5) /2.

**Input:** Utente

**Output:** Affidabilità di un utente (MV\_Affidabilita).

**Frequenza:** ogni 7 giorni.

**Parte di diagramma interessata:**

****

**Tavola dei volumi interessata:**

|  |  |  |
| --- | --- | --- |
| Utente | Entità | 50.000 |
| RideSharing | Entità | 18.000 |
| UsoSharing | Relazione | 18.000 |
| Autovettura | Entità | 22.000 |
| UsoPool | Relazione | 36.000 |
| Pool | Entità | 36.000 |
| PropostaPool | Relazione | 36.000 |
| Proprietà | Relazione | 22.000 |
| Proponente | Entità | 20.000 |
| RuoloP | Relazione | 20.000 |

**Tavola degli accessi interessata:**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome Costrutto | Numero Operazioni elementari | Tipo Operazione | Descrizione |
| Utente | 1 × 50.000 | L | Leggo la valutazione media per ogni utente |
| RuoloP | 1 × 20.000 | L | Accedo all’entità Proponente |
| Proponente | 1 × 20.000 | L | Leggo il codice proponente |
| PropostaPool | 1 × 20.000 | L | Accedo all’entità Pool |
| Pool | 1 × 2 × 20.000 | L | Conto il numero di pool per proponente e leggo l’orario di partenza prefissato ed effettivo di ogni pool effettuato |
| RideSharing | 1 × 20.000 | L | Conto il numero di ride sharing per proponente e leggo l’orario di partenza prefissato ed effettivo di ogni ride sharing effettuato |
| Affidabilità | 2 × 50.000 | S | Scrivo il valore di affidabilità per ogni utente |

**Tavola delle operazioni:**

|  |  |
| --- | --- |
| Totale operazioni elementari per singola operazione | 50.000+20.000+20.000+20.000+40.000+20.000+100.000=2.700.000 |
| Totale operazioni elementari | 2.700.000 × 1 = 2.700.000 (ogni 7 giorni) |

## *Tempi di percorrenza e rilevazione delle criticità*

L’analisi della viabilità su un determinato chilometro di una strada avviene grazie all’aggiornamento periodico dell’attributo *TempiMedi* su **Chilometro.** Infatti l’aumento dei tempi medi di percorrenza sono un segnale della possibile compromissione della viabilità.

Si è deciso di implementare questa funzione con l’utilizzo di una materialized view **MV\_Viabilita** in cui teniamo traccia del tempo medio di percorrenza regolare per un determinato chilometro di una strada. I tempi medi di percorrenza vengono aggiornati ogni ora, e sono calcolati grazie ad una funzione che calcola la media di tutti i tempi di percorrenza relativi all’ultima ora per ogni chilometro di una strada, registrati dai veicoli usati nei vari servizi. Nella materialized view l’attributo *Viabilità*, che indica appunto lo stato di percorrenza di un chilometro di una strada, viene modificato ogni volta che si aggiorna su ***Chilometro*** l’attributo *TempoMedio.* La viabilità sarà compromessa se i tempi medi di percorrenza raddoppiano rispetto ai tempi regolari.

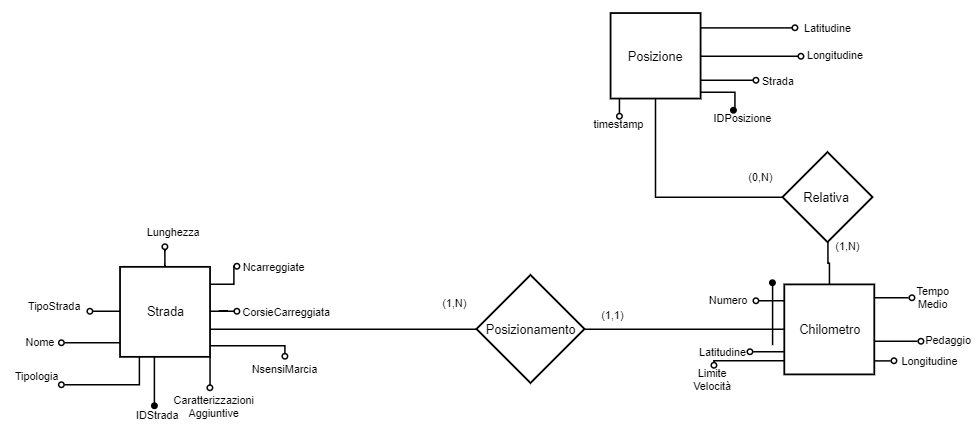
* VIABILITA (IDStrada, Numero, Viabilita, TempoMedioRegolare)

**Input**: Chilometro

**Output**: MV\_Viabilita

**Frequenza giornaliera**: 24

**Parte di diagramma interessata:**

****

**Tavola dei volumi interessata:**

|  |  |  |
| --- | --- | --- |
| Chilometro | Entità | 750 |
| Posizionamento | Relazione | 750 |
| Strada | Entità | 750 |
| Relativa | Relazione | 9.600.000 |
| Posizione | Entità | 9.576.000 |

**Tavola degli accessi interessata:**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome Costrutto | Numero Operazioni elementari | Tipo Operazione | Descrizione |
| Chilometro | 1 × 750 | L | Leggo i chilometri relativi ad una strada |
| Relativa | 1 × 750 | L | Leggo un’occorrenza tra Chilometro e Posizione |
| Posizione | 1 × 750 × 2 | L | Leggo le posizioni registrate nell’ultima ora |
| Chilometro | 2 × 750 | S | Aggiorno i tempi medi di percorrenza |
| Viabilita | 2 | S | Aggiorno il valore dell’attributo Viabilita |

**Tavola delle operazioni:**

|  |  |
| --- | --- |
| Totale operazioni elementari per singola operazione | 750+750+ 1.500+ 1.500+ 2 = 4.502 |
| Totale operazioni elementari al giorno | 4.502 × 24 = 108.048 |

# ***Script SQL***

1. **SET** NAMES latin1;
2. **DROP** **DATABASE** IF EXISTS `carsharing`;
3. **CREATE** **DATABASE**  IF NOT EXISTS `carsharing`;
4. USE `carsharing`;
5. **SET** FOREIGN\_KEY\_CHECKS = 1;
6. **SET** **GLOBAL** EVENT\_SCHEDULER = **ON**;
8. **DROP** **TABLE** IF EXISTS `utente`;
9. **CREATE** **TABLE** `utente` (
10. `CodFiscale` **char**(50) NOT NULL,
11. `Nome` **char**(50) NOT NULL,
12. `Cognome` **char**(50) NOT NULL,
13. `NumeroTelefono` **int**(11) unsigned NOT NULL,
14. `NumeroCivico` **int**(11) unsigned NOT NULL,
15. `Via` **char**(50) NOT NULL,
16. `CAP` **int**(11) unsigned NOT NULL,
17. `ValutazioneMedia` **float**(13, 2) unsigned **DEFAULT** NULL,
18. **PRIMARY** **KEY** (`CodFiscale`)
19. )ENGINE=InnoDB **DEFAULT** CHARSET=latin1;
21. **DROP** **TABLE** IF EXISTS `account`;
22. **CREATE** **TABLE** `account` (
23. `NomeUtente` **char**(50) NOT NULL,
24. `Data` **date** NOT NULL,
25. `CodFiscale` **char**(50) NOT NULL,
26. `**Password**` **char**(50) NOT NULL,
27. `DomandaRiserva` **char**(50) NOT NULL,
28. `RispostaRiserva` **char**(50) NOT NULL,
29. `Stato` **char**(50) NOT NULL **DEFAULT** 'inattivo',
30. **PRIMARY** **KEY** (`NomeUtente`),
31. **UNIQUE** (`CodFiscale`)
32. )ENGINE=InnoDB **DEFAULT** CHARSET=latin1;
34. **DROP** **TABLE** IF EXISTS `documento`;
35. **CREATE** **TABLE** `documento` (
36. `Numero` **int**(11) unsigned NOT NULL,
37. `CodFiscale` **char**(50) NOT NULL,
38. `Tipologia` **char**(50) NOT NULL,
39. `Scadenza` **date** NOT NULL,
40. `EnteRilascio` **char**(50) NOT NULL,
41. **PRIMARY** **KEY** (`Numero`)
42. )ENGINE=InnoDB **DEFAULT** CHARSET=latin1;
44. **DROP** **TABLE** IF EXISTS `fruitore`;
45. **CREATE** **TABLE** `fruitore` (
46. `CodiceFruitore` **char**(50) NOT NULL,
47. `CodFiscale` **char**(50) NOT NULL,
48. **PRIMARY** **KEY** (`CodiceFruitore`),
49. **UNIQUE** (`CodFiscale`)
50. )ENGINE=InnoDB **DEFAULT** CHARSET=latin1;
52. **DROP** **TABLE** IF EXISTS `valutazione`;
53. **CREATE** **TABLE** `valutazione` (
54. `Codice` **char**(50) NOT NULL,
55. `CodiceTragitto` **char**(50) NOT NULL,
56. `CodFiscale` **char**(50) NOT NULL,
57. `TestoValutazione` **char**(50) NOT NULL,
58. `Comportamento` **int**(11) unsigned NOT NULL,
59. `Serieta` **int**(11) unsigned NOT NULL,
60. `PiacereViaggio` **int**(11) unsigned NOT NULL,
61. `GiudizioPersona` **int**(11) unsigned NOT NULL,
62. `Ruolo` **char**(50) NOT NULL,
63. `Valutato` **char**(50) NOT NULL,
64. **PRIMARY** **KEY** (`Codice`)
65. )ENGINE=InnoDB **DEFAULT** CHARSET=latin1;
67. **DROP** **TABLE** IF EXISTS `optional`;
68. **CREATE** **TABLE** `optional`(
69. `Targa` **char**(50) NOT NULL,
70. `Connettivita` bool **DEFAULT** NULL,
71. `Tavolini` bool **DEFAULT** NULL,
72. `TettoVetro` bool **DEFAULT** NULL,
73. `DimensioneBagagliaio` **int**(11) unsigned NOT NULL, -- L
74. `RumoreAbitacolo` **int**(11) unsigned NOT NULL, -- dB
75. **PRIMARY** **KEY** (`Targa`)
76. )ENGINE=InnoDB **DEFAULT** CHARSET=latin1;
78. **DROP** **TABLE** IF EXISTS `prenotazionen`;
79. **CREATE** **TABLE** `prenotazionen`(
80. `CodiceFruitore` **char**(50) NOT NULL,
81. `IDNoleggio` **char**(50) NOT NULL,
82. `Accettata` **char**(50) **DEFAULT** 'pending' NOT NULL,
83. **PRIMARY** **KEY** (`CodiceFruitore`, `IDNoleggio`)
84. )ENGINE=InnoDB **DEFAULT** CHARSET=latin1;
86. **DROP** **TABLE** IF EXISTS `noleggio`;
87. **CREATE** **TABLE** `noleggio` (
88. `IDNoleggio` **char**(50) NOT NULL,
89. `CodiceFruitore` **char**(50) NOT NULL,
90. `Targa` **char**(50) NOT NULL,
91. `DataInizio` **date** NOT NULL,
92. `CarburanteFineNoleggio` **float**(13, 2) unsigned NOT NULL, -- L
93. `DataFine` **date** NOT NULL,
94. `CodiceTragitto` **char**(50) **DEFAULT** NULL,
95. **PRIMARY** **KEY** (`IDNoleggio`)
96. )ENGINE=InnoDB **DEFAULT** CHARSET=latin1;
98. **DROP** **TABLE** IF EXISTS `autocoinvolta`;
99. **CREATE** **TABLE** `autocoinvolta` (
100. `Targa` **char**(50) NOT NULL,
101. `IDNoleggio` **char**(50) NOT NULL,
102. `**Timestamp**` datetime NOT NULL,
103. `CasaAutomobilistica` **char**(50) NOT NULL,
104. `Modello` **char**(50) NOT NULL,
105. **PRIMARY** **KEY** (`Targa`, `IDNoleggio`, `**Timestamp**`)
106. )ENGINE=InnoDB **DEFAULT** CHARSET=latin1;
108. **DROP** **TABLE** IF EXISTS `ridesharing`;
109. **CREATE** **TABLE** `ridesharing` (
110. `IDSharing` **char**(50) NOT NULL,
111. `CodiceTragitto` **char**(50) NOT NULL,
112. `Targa` **char**(50) NOT NULL,
113. `OrarioArrivo` datetime NOT NULL,
114. `OrarioPartenza` datetime NOT NULL,
115. `PartenzaEffettiva` datetime **DEFAULT** NULL,
116. `CodiceProponente` **char**(50) NOT NULL,
117. **PRIMARY** **KEY** (`IDSharing`)
118. )ENGINE=InnoDB **DEFAULT** CHARSET=latin1;
120. **DROP** **TABLE** IF EXISTS `corsasharing`;
121. **CREATE** **TABLE** `corsasharing` (
122. `Codice` **char**(50) NOT NULL,
123. `IDSharing` **char**(50) NOT NULL,
124. **PRIMARY** **KEY** (`Codice`,`IDSharing`)
125. )ENGINE=InnoDB **DEFAULT** CHARSET=latin1;
127. **DROP** **TABLE** IF EXISTS `fasciaoraria`;
128. **CREATE** **TABLE** `fasciaoraria` (
129. `Targa` **char**(50) NOT NULL,
130. `OraInizio` **time** NOT NULL,
131. `OraFine` **time** NOT NULL,
132. `Giorno` **char**(50) NOT NULL,
133. **PRIMARY** **KEY** (`Targa`, `OraInizio`, `OraFine`, `Giorno`)
134. )ENGINE=InnoDB **DEFAULT** CHARSET=latin1;
136. **DROP** **TABLE** IF EXISTS `pool`;
137. **CREATE** **TABLE** `pool` (
138. `IDPool` **char**(50) NOT NULL,
139. `CodiceTragitto` **char**(50) NOT NULL,
140. `CodiceProponente` **char**(50) NOT NULL,
141. `Targa` **char**(50) NOT NULL,
142. `Flessibilita` **char**(50) NOT NULL,
143. `NumeroPosti` **int**(11) unsigned NOT NULL,
144. `OreChiusura` **int**(11) unsigned NOT NULL,
145. `Spesa` **float**(13, 2) unsigned NOT NULL, -- €
146. `Partenza` datetime NOT NULL,
147. `GiornoArrivo` datetime NOT NULL,
148. `Stato` **char**(50) NOT NULL,
149. `CostoVariazione` **float**(13, 2) unsigned **DEFAULT** 0, -- €
150. `PartenzaEffettiva` datetime,
151. **PRIMARY** **KEY** (`IDPool`)
152. )ENGINE=InnoDB **DEFAULT** CHARSET=latin1;
154. **DROP** **TABLE** IF EXISTS `variazione`;
155. **CREATE** **TABLE** `variazione` (
156. `IDPool` **char**(50) NOT NULL,
157. `IDFruitore` **char**(50) NOT NULL,
158. `Accettata` **char**(50) NOT NULL **DEFAULT** 'pending',
159. `Entita` **char**(50) NOT NULL,
160. `CodiceTragitto` **char**(50) NOT NULL,
161. **PRIMARY** **KEY** (`IDPool`, `IDFruitore`)
162. )ENGINE=InnoDB **DEFAULT** CHARSET=latin1;
164. **DROP** **TABLE** IF EXISTS `tragitto`;
165. **CREATE** **TABLE** `tragitto` (
166. `CodiceTragitto` **char**(50) NOT NULL,
167. `LunghezzaTragitto` **float**(13, 2) unsigned **DEFAULT** 0.0, -- km
168. `PosizioneArrivo` **char**(50) NOT NULL,
169. `PosizionePartenza` **char**(50) NOT NULL,
170. **PRIMARY** **KEY** (`CodiceTragitto`)
171. )ENGINE=InnoDB **DEFAULT** CHARSET=latin1;
173. **DROP** **TABLE** IF EXISTS `posizione`;
174. **CREATE** **TABLE** `posizione` (
175. `IDPosizione` **char**(50) NOT NULL,
176. `Strada` **char**(50) NOT NULL,
177. `Latitudine` **float**(18, 7) NOT NULL,
178. `Longitudine` **float**(18, 7) NOT NULL,
179. `**Timestamp**` datetime,
180. `Numero` **int**(11) unsigned NOT NULL,
181. `CodiceTragitto` **char**(50) NOT NULL,
182. **PRIMARY** **KEY** (`IDPosizione`)
183. )ENGINE=InnoDB **DEFAULT** CHARSET=latin1;
185. **DROP** **TABLE** IF EXISTS `chilometro`;
186. **CREATE** **TABLE** `chilometro` (
187. `IDStrada` **char**(50) NOT NULL,
188. `Numero` **int**(11) unsigned NOT NULL,
189. `Latitudine` **float**(18, 7) NOT NULL,
190. `Longitudine` **float**(18, 7) NOT NULL,
191. `LimiteVelocita` **float**(13, 2) unsigned NOT NULL, -- km/h
192. `Pedaggio` **float**(13, 2) unsigned, -- €
193. `TempoMedio` **time**,
194. **PRIMARY** **KEY**(`IDStrada`, `Numero`)
195. )ENGINE=InnoDB **DEFAULT** CHARSET=latin1;
197. **DROP** **TABLE** IF EXISTS `strada`;
198. **CREATE** **TABLE** `strada` (
199. `IDStrada` **char**(50) NOT NULL,
200. `Tipologia` **char**(50) NOT NULL,
201. `Nome` **char**(50),
202. `TipoStrada` **char**(50) NOT NULL,
203. `Lunghezza` **float**(13, 2) unsigned NOT NULL,
204. `NCarreggiate` **int**(11) unsigned NOT NULL,
205. `CorsieCarreggiata`**int**(11) unsigned NOT NULL,
206. `NSensiMarcia`**int**(11) unsigned NOT NULL,
207. `CaratterizzazioniAggiuntive` **char**(50),
208. **PRIMARY** **KEY** (`IDStrada`)
209. )ENGINE=InnoDB **DEFAULT** CHARSET=latin1;
211. **DROP** **TABLE** IF EXISTS `proponente`;
212. **CREATE** **TABLE** `proponente` (
213. `CodiceProponente` **char**(50) NOT NULL,
214. `CodFiscale` **char**(50)  NOT NULL,
215. **PRIMARY** **KEY** (`CodiceProponente`),
216. **UNIQUE** (`CodFiscale`)
217. ) ENGINE=InnoDB **DEFAULT** CHARSET=latin1;
219. **DROP** **TABLE** IF EXISTS `autovettura`;
220. **CREATE** **TABLE** `autovettura` (
221. `Targa` **char**(50) NOT NULL,
222. `CodiceProponente` **char**(50)  NOT NULL,
223. `AnnoImmatricolazione` **int**(11)  NOT NULL,
224. `CostoOperativo` **float**(13, 2) unsigned NOT NULL, -- costo per kilometro
225. `CostoUsura` **float**(13, 2) unsigned NOT NULL, -- costo per kilometro
226. `CostoExtra` **float**(13, 2) unsigned NOT NULL, -- costo per kilometro
227. `ChilometriPercorsi` **float**(13, 2) unsigned NOT NULL,
228. `Disponibilita` **char**(50) NOT NULL,
229. `CarburanteDisponibile` **float**(13, 2) unsigned NOT NULL, -- L
230. `NomeModello` **char**(50) NOT NULL,
231. **PRIMARY** **KEY** (`Targa`)
232. ) ENGINE=InnoDB **DEFAULT** CHARSET=latin1;
234. **DROP** **TABLE** IF EXISTS `chiamatasharing`;
235. **CREATE** **TABLE** `chiamatasharing` (
236. `Codice` **char**(50) NOT NULL,
237. `CodiceFruitore` **char**(50)  NOT NULL,
238. `PosizioneFruitore` **char**(50) NOT NULL,
239. `Destinazione` **char**(50) NOT NULL,
240. `TimestampChiamata` datetime  NOT NULL,
241. `Stato` **char**(50) **DEFAULT** 'pending' NOT NULL,
242. `TimestampRisposta` datetime NOT NULL,
243. **PRIMARY** **KEY** (`Codice`)
244. ) ENGINE=InnoDB **DEFAULT** CHARSET=latin1;
246. **DROP** **TABLE** IF EXISTS `incidente`;
247. **CREATE** **TABLE** `incidente` (
248. `IDNoleggio` **char**(50) NOT NULL,
249. `**Timestamp**` datetime  NOT NULL,
250. `Dinamica` **char**(250)  NOT NULL,
251. **PRIMARY** **KEY** (`IDNoleggio`,`**Timestamp**`)
252. ) ENGINE=InnoDB **DEFAULT** CHARSET=latin1;
254. **DROP** **TABLE** IF EXISTS `prenotazionep`;
255. **CREATE** **TABLE** `prenotazionep` (
256. `CodiceFruitore` **char**(50) NOT NULL,
257. `IDPool` **char**(50)  NOT NULL,
258. `CodicePrenotazione` **char**(50) NOT NULL,
259. **PRIMARY** **KEY** (`CodiceFruitore`,`IDPool`,`CodicePrenotazione`)
260. ) ENGINE=InnoDB **DEFAULT** CHARSET=latin1;
262. **DROP** **TABLE** IF EXISTS `modello`;
263. **CREATE** **TABLE** `modello` (
264. `NomeModello` **char**(50) NOT NULL,
265. `ConsumoMisto` **float**(13, 2) unsigned  NOT NULL, -- litri per 100 kilometro
266. `ConsumoUrbano` **float**(13, 2) unsigned NOT NULL, -- litri per 100 kilometro
267. `ConsumoExtraurbano` **float**(13, 2) unsigned NOT NULL, - litri per 100 kilometro
268. `CasaProduttrice` **char**(50) NOT NULL,
269. `NPosti` **int**(11) unsigned NOT NULL,
270. `Alimentazione` **char**(50) NOT NULL,
271. `Comfort` **int** (11) unsigned NOT NULL, -- da 1 a 5
272. `Serbatoio` **int**(11) unsigned NOT NULL, -- dipende dal tipo di alimetazione
273. `Cilindrata` **int**(11) unsigned NOT NULL,
274. **PRIMARY** **KEY** (`NomeModello`)
275. ) ENGINE=InnoDB **DEFAULT** CHARSET=latin1;
277. **DROP** **TABLE** IF EXISTS `relativa`;
278. **CREATE** **TABLE** `relativa` (
279. `IDPosizione` **char**(50) NOT NULL,
280. `Numero` **int**(11) unsigned  NOT NULL,
281. `IDStrada` **char**(50) NOT NULL,
282. **PRIMARY** **KEY** (`IDPosizione`,`Numero`,`IDStrada`)
283. ) ENGINE=InnoDB **DEFAULT** CHARSET=latin1;
285. **DROP** **TABLE** IF EXISTS `intersezione`;
286. **CREATE** **TABLE** `intersezione` (
287. `Numero1` **int**(11) unsigned NOT NULL,
288. `IDStrada1` **char**(50)  NOT NULL,
289. `Numero2` **int**(11) unsigned NOT NULL,
290. `IDStrada2` **char**(50) NOT NULL,
291. **PRIMARY** **KEY** (`Numero1`,`IDStrada1`,`Numero2`,`IDStrada2`)
292. ) ENGINE=InnoDB **DEFAULT** CHARSET=latin1;
294. -– aggiornamento delle ridondanze
295. **DROP** **TRIGGER** IF EXISTS `aggiorna\_lunghezza\_tragitto`;
296. DELIMITER $$
297. **CREATE** **TRIGGER** `aggiorna\_lunghezza\_tragitto`
298. **AFTER** **INSERT** **ON** `posizione` **FOR** EACH ROW
299. **BEGIN**
300. **DECLARE** lat1 **float**(13, 2);
301. **DECLARE** lat2 **float**(13, 2);
302. **DECLARE** lon1 **float**(13, 2);
303. **DECLARE** lon2 **float**(13, 2);
304. **DECLARE** LunghezzaAggiunta **float**(18, 7);
305. IF (NEW.`Numero` > 1) **THEN**
306. **SET** lat1 = (**SELECT** `Latitudine`
307. **FROM** `posizione`
308. **WHERE** `CodiceTragitto` = NEW.`CodiceTragitto`
309. AND `Numero` = NEW.`Numero` - 1)\*PI()/180;
310. **SET** lon1 = (**SELECT** `Longitudine`
311. **FROM** `posizione`
312. **WHERE** `CodiceTragitto` = NEW.`CodiceTragitto`
313. AND `Numero` = NEW.`Numero` - 1)\*PI()/180;
314. **SET** lat2 = NEW.`Latitudine`\*PI()/180;
315. **SET** lon2 = NEW.`Longitudine`\*PI()/180;
316. **SET** LunghezzaAggiunta = ACOS(SIN(lat1)\*SIN(lat2) + COS(lat1)\*COS(lat2)\*COS(lon2-lon1))\*6371;
318. IF LunghezzaAggiunta **IS** NOT NULL **THEN**
319. **UPDATE** `tragitto`
320. **SET** `LunghezzaTragitto` = `LunghezzaTragitto` + LunghezzaAggiunta
321. **WHERE** `CodiceTragitto` = NEW.`CodiceTragitto`;
322. **END** IF;
323. **END** IF;
324. **END** $$
326. DELIMITER ;
328. **DROP** **TRIGGER** IF EXISTS `aggiornamento\_media\_valutazioni`;
329. DELIMITER $$
330. **CREATE** **TRIGGER** `aggiornamento\_media\_valutazioni`
331. **AFTER** **INSERT** **ON** `valutazione` **FOR** EACH ROW
332. **BEGIN**
333. **DECLARE** media **float**(13, 2) **DEFAULT** NULL;
334. **SET** media = (**SELECT** AVG((`Comportamento`+`Serieta`+`PiacereViaggio`+`GiudizioPersona`)/4)
335. **FROM** `valutazione`
336. **WHERE** `Valutato` = NEW.`Valutato`
337. );
338. **UPDATE** `utente`
339. **SET** `ValutazioneMedia` = media
340. **WHERE** `CodFiscale` = NEW.`Valutato`;
341. **END** $$
343. DELIMITER ;
344. **SET** NAMES latin1;
345. USE `carsharing`;
346. **SET** FOREIGN\_KEY\_CHECKS = 1;
347. **SET** **GLOBAL** EVENT\_SCHEDULER = **ON**;
348. -– foreign keys
349. **ALTER** **TABLE** `utente`
350. **ADD** **FOREIGN** **KEY**(`CodFiscale`) **REFERENCES** `account`(`CodFiscale`)
351. **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**;
353. **ALTER** **TABLE** `documento`
354. **ADD** **FOREIGN** **KEY**(`CodFiscale`) **REFERENCES** `utente`(`CodFiscale`)
355. **ON** **DELETE** **CASCADE** **ON** **UPDATE** **NO** **ACTION**;
357. **ALTER** **TABLE** `fruitore`
358. **ADD** **FOREIGN** **KEY**(`CodFiscale`) **REFERENCES** `utente`(`CodFiscale`)
359. **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**;
361. **ALTER** **TABLE** `valutazione`
362. **ADD** **FOREIGN** **KEY**(`CodFiscale`) **REFERENCES** `utente`(`CodFiscale`)
363. **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,
364. **ADD** **FOREIGN** **KEY**(`CodiceTragitto`) **REFERENCES** `tragitto`(`CodiceTragitto`)
365. **ON** **DELETE** **NO** **ACTION** **ON** **UPDATE** **CASCADE**;
367. **ALTER** **TABLE** `optional`
368. **ADD** **FOREIGN** **KEY**(`Targa`) **REFERENCES** `autovettura`(`Targa`)
369. **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**;
371. **ALTER** **TABLE** `prenotazionen`
372. **ADD** **FOREIGN** **KEY**(`IDNoleggio`) **REFERENCES** `noleggio`(`IDNoleggio`)
373. **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,
374. **ADD** **FOREIGN** **KEY**(`CodiceFruitore`) **REFERENCES** `fruitore`(`CodiceFruitore`)
375. **ON** **DELETE** **NO** **ACTION** **ON** **UPDATE** **CASCADE**;
377. **ALTER** **TABLE** `noleggio`
378. **ADD** **FOREIGN** **KEY**(`Targa`) **REFERENCES** `autovettura`(`Targa`)
379. **ON** **DELETE** **NO** **ACTION** **ON** **UPDATE** **CASCADE**,
380. **ADD** **FOREIGN** **KEY**(`CodiceTragitto`) **REFERENCES** `Tragitto`(`CodiceTragitto`)
381. **ON** **DELETE** **NO** **ACTION** **ON** **UPDATE** **CASCADE**;
383. **ALTER** **TABLE** `autocoinvolta`
384. **ADD** **FOREIGN** **KEY**(`IDNoleggio`, `**Timestamp**`) **REFERENCES** `incidente`(`IDNoleggio`, `**Timestamp**`)
385. **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**;
387. **ALTER** **TABLE** `ridesharing`
388. **ADD** **FOREIGN** **KEY**(`CodiceTragitto`) **REFERENCES** `tragitto`(`CodiceTragitto`)
389. **ON** **DELETE** **NO** **ACTION** **ON** **UPDATE** **CASCADE**,
390. **ADD** **FOREIGN** **KEY**(`Targa`) **REFERENCES** `autovettura`(`Targa`)
391. **ON** **DELETE** **NO** **ACTION** **ON** **UPDATE** **CASCADE**;
393. **ALTER** **TABLE** `corsasharing`
394. **ADD** **FOREIGN** **KEY**(`Codice`) **REFERENCES** `chiamatasharing`(`Codice`)
395. **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,
396. **ADD** **FOREIGN** **KEY**(`IDSharing`) **REFERENCES** `ridesharing`(`IDSharing`)
397. **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**;
399. **ALTER** **TABLE** `fasciaoraria`
400. **ADD** **FOREIGN** **KEY**(`Targa`) **REFERENCES** `autovettura`(`Targa`)
401. **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**;
403. **ALTER** **TABLE** `pool`
404. **ADD** **FOREIGN** **KEY**(`CodiceProponente`) **REFERENCES** `proponente`(`CodiceProponente`)
405. **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,
406. **ADD** **FOREIGN** **KEY**(`CodiceTragitto`) **REFERENCES** `tragitto`(`CodiceTragitto`)
407. **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,
408. **ADD** **FOREIGN** **KEY**(`Targa`) **REFERENCES** `autovettura`(`Targa`)
409. **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**;
411. **ALTER** **TABLE** `variazione`
412. **ADD** **FOREIGN** **KEY**(`CodiceTragitto`) **REFERENCES** `tragitto`(`CodiceTragitto`)
413. **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,
414. **ADD** **FOREIGN** **KEY**(`IDPool`) **REFERENCES** `pool`(`IDPool`)
415. **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**;
417. **ALTER** **TABLE** `posizione`
418. **ADD** **FOREIGN** **KEY**(`CodiceTragitto`) **REFERENCES** `tragitto`(`CodiceTragitto`)
419. **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**;
421. **ALTER** **TABLE** `chilometro`
422. **ADD** **FOREIGN** **KEY**(`IDStrada`) **REFERENCES** `strada`(`IDStrada`)
423. **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**;
425. **ALTER** **TABLE** `proponente`
426. **ADD** **FOREIGN** **KEY**(`CodFiscale`) **REFERENCES** `utente`(`CodFiscale`)
427. **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**;
429. **ALTER** **TABLE** `autovettura`
430. **ADD** **FOREIGN** **KEY**(`CodiceProponente`) **REFERENCES** `proponente`(`CodiceProponente`)
431. **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,
432. **ADD** **FOREIGN** **KEY**(`NomeModello`) **REFERENCES** `modello`(`NomeModello`)
433. **ON** **DELETE** **NO** **ACTION** **ON** **UPDATE** **CASCADE**;
435. **ALTER** **TABLE** `chiamatasharing`
436. **ADD** **FOREIGN** **KEY**(`CodiceFruitore`) **REFERENCES** `fruitore`(`CodiceFruitore`)
437. **ON** **DELETE** **NO** **ACTION** **ON** **UPDATE** **CASCADE**;
439. **ALTER** **TABLE** `incidente`
440. **ADD** **FOREIGN** **KEY**(`IDNoleggio`) **REFERENCES** `noleggio`(`IDNoleggio`)
441. **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**;
443. **ALTER** **TABLE** `prenotazionep`
444. **ADD** **FOREIGN** **KEY**(`IDPool`) **REFERENCES** `pool`(`IDPool`)
445. **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,
446. **ADD** **FOREIGN** **KEY**(`CodiceFruitore`) **REFERENCES** `fruitore`(`CodiceFruitore`)
447. **ON** **DELETE** **NO** **ACTION** **ON** **UPDATE** **CASCADE**;
449. **ALTER** **TABLE** `relativa`
450. **ADD** **FOREIGN** **KEY**(`IDStrada`, `Numero`) **REFERENCES** `chilometro`(`IDStrada`, `Numero`)
451. **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,
452. **ADD** **FOREIGN** **KEY**(`IDPosizione`) **REFERENCES** `posizione`(`IDPosizione`)
453. **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**;
455. **ALTER** **TABLE** `intersezione`
456. **ADD** **FOREIGN** **KEY**(`IDStrada2`, `Numero2`) **REFERENCES** `chilometro`(`IDStrada`, `Numero`)
457. **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,
458. **ADD** **FOREIGN** **KEY**(`IDStrada1`, `Numero1`) **REFERENCES** `chilometro`(`IDStrada`, `Numero`)
459. **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**;
461. -– triggers
462. **SET** NAMES latin1;
463. USE `carsharing`;
464. **SET** FOREIGN\_KEY\_CHECKS = 1;
465. **SET** **GLOBAL** EVENT\_SCHEDULER = **ON**;
467. **DROP** **TRIGGER** IF EXISTS `controllo\_scadenza\_documento`;
468. DELIMITER $$
469. **CREATE** **TRIGGER** `controllo\_scadenza\_documento`
470. BEFORE **INSERT** **ON** `documento` **FOR** EACH ROW
471. **BEGIN**
472. IF NEW.`Scadenza` < **current\_date** **THEN**
473. SIGNAL SQLSTATE '45000'
474. **SET** MESSAGE\_TEXT = 'Documento scaduto';
475. **END** IF;
476. **END** $$
478. DELIMITER ;
480. **DROP** **TRIGGER** IF EXISTS `controllo\_entita\_variazione`;
481. DELIMITER $$
482. **CREATE** **TRIGGER** `controllo\_entita\_variazione`
483. BEFORE **INSERT** **ON** `variazione` **FOR** EACH ROW
484. **BEGIN**
485. **SET** @flessibilita = (**SELECT** `Flessibilita`
486. **FROM** `pool`
487. **WHERE** `IDPool` = NEW.`IDPool`);
488. IF (NEW.`Entita` = 'media') **THEN**
489. IF (flessibilita = 'bassa') **THEN**
490. SIGNAL SQLSTATE '45000'
491. **SET** MESSAGE\_TEXT = 'flessibilità non compatibile con la variazione';
492. **END** IF;
493. ELSEIF (NEW.`Entita` = 'alta') **THEN**
494. IF (flessibilita = 'bassa') **THEN**
495. SIGNAL SQLSTATE '45000'
496. **SET** MESSAGE\_TEXT = 'flessibilità non compatibile con la variazione';
497. ELSEIF (flessibilita = 'media') **THEN**
498. SIGNAL SQLSTATE '45000'
499. **SET** MESSAGE\_TEXT = 'flessibilità non compatibile con la variazione';
500. **END** IF;
501. **END** IF;
502. **END** $$
504. DELIMITER ;
506. **DROP** **TRIGGER** IF EXISTS `controllo\_stato\_pool`;
507. DELIMITER $$
508. **CREATE** **TRIGGER** `controllo\_stato\_pool`
509. BEFORE **INSERT** **ON** `prenotazionep` **FOR** EACH ROW
510. **BEGIN**
511. IF (**SELECT** `Stato`
512. **FROM** `pool`
513. **WHERE** `IDPool` = NEW.`IDPool`
514. ) = 'chiuso' **THEN**
515. SIGNAL SQLSTATE '45000'
516. **SET** MESSAGE\_TEXT = 'impossibile prenotare un pool chiuso';
517. **END** IF;
518. **END** $$
520. DELIMITER ;
522. **DROP** **TRIGGER** IF EXISTS `controllo\_stato\_pool2`;
523. DELIMITER $$
524. **CREATE** **TRIGGER** `controllo\_stato\_pool2`
525. BEFORE **INSERT** **ON** `variazione` **FOR** EACH ROW
526. **BEGIN**
527. IF (**SELECT** `Stato`
528. **FROM** `pool`
529. **WHERE** `IDPool` = NEW.`IDPool`
530. ) = 'chiuso' **THEN**
531. SIGNAL SQLSTATE '45000'
532. **SET** MESSAGE\_TEXT = 'impossibile inserire una variazione per un pool chiuso';
533. **END** IF;
534. **END** $$
536. DELIMITER ;
538. **DROP** **TRIGGER** IF EXISTS `controllo\_variazione`;
539. DELIMITER $$
540. **CREATE** **TRIGGER** `controllo\_variazione`
541. BEFORE **INSERT** **ON** `variazione` **FOR** EACH ROW
542. **BEGIN**
543. IF (**SELECT** `PosizionePartenza`
544. **FROM** `tragitto`
545. **WHERE** NEW.`CodiceTragitto` = `CodiceTragitto`) =
546. (**SELECT** `PosizioneArrivo`
547. **FROM** `tragitto`
548. **WHERE** NEW.`CodiceTragitto` = `CodiceTragitto`) **THEN**
549. SIGNAL SQLSTATE '45000'
550. **SET** MESSAGE\_TEXT = 'impossibile inserire una variazione per un pool chiuso';
551. **END** IF;
552. **END** $$
554. DELIMITER ;
556. **DROP** **TRIGGER** IF EXISTS `aggiornamento\_tempi\_medi`;
557. DELIMITER $$
558. **CREATE** **TRIGGER** `aggiornamento\_tempi\_medi`
559. **AFTER** **UPDATE** **ON** `chilometro` **FOR** EACH ROW
560. **BEGIN**
561. **DECLARE** `TMR` **time**;
562. **SET** `TMR` = (**SELECT** `TempoMedioRegolare`
563. **FROM** `MV\_Viabilita`
564. **WHERE** NEW.`Numero` = `Numero` AND NEW.`IDStrada` = `IDStrada`
565. );
566. IF (NEW.`TempoMedio` > 2\*`TMR`) **THEN**
567. **UPDATE** `MV\_Viabilita`
568. **SET** `Viabilita` = 'compromessa';
569. **WHERE** NEW.`Numero` = `Numero` AND NEW.`IDStrada` = `IDStrada`;
570. **ELSE**
571. **UPDATE** `MV\_Viabilita`
572. **SET** `Viabilita` = 'regolare';
573. **WHERE** NEW.`Numero` = `Numero` AND NEW.`IDStrada` = `IDStrada`;
574. **END** IF;
575. **END** $$
577. DELIMITER ;
578. -- Operazioni
579. **SET** NAMES latin1;
580. USE `carsharing`;
581. **SET** FOREIGN\_KEY\_CHECKS = 1;
582. **SET** **GLOBAL** EVENT\_SCHEDULER = **ON**;
584. -- Operazione 1: Registrazione Utente
586. **DROP** **PROCEDURE** IF EXISTS RegistrazioneUtente;
587. DELIMITER $$
588. **CREATE** **PROCEDURE** RegistrazioneUtente(IN `CodFiscale` **char**(50),
589. IN `Nome` **char**(50),
590. IN `Cognome` **char**(50),
591. IN `NumeroTelefono` **int**(11) unsigned,
592. IN `NumeroCivico` **int**(11) unsigned,
593. IN `Via` **char**(50),
594. IN `CAP` **int**(11) unsigned,
595. IN `**Password**` **char**(50),
596. IN `NomeUtente` **char**(50),
597. IN `DomandaRiserva` **char**(50),
598. IN `RispostaRiserva` **char**(50),
599. IN `Numero` **int**(11) unsigned,
600. IN `Tipologia` **char**(50),
601. IN `Scadenza` **date**,
602. IN `EnteRilascio` **char**(50)
603. )
604. **BEGIN**
605. **INSERT** **INTO** `Account`
606. **VALUES**(`NomeUtente`, **current\_date**(), `CodFiscale`, `**Password**`, `DomandaRiserva`, `RispostaRiserva`, 'inattivo');
608. **INSERT** **INTO** `Utente`
609. **VALUES**(`CodFiscale`, `Nome`, `Cognome`, `NumeroTelefono`, `NumeroCivico`, `Via`, `CAP`, NULL);
611. **INSERT** **INTO** `Documento`
612. **VALUES**(`Numero`, `CodFiscale`, `Tipologia`, `Scadenza`, `EnteRilascio`);
613. **END** $$
615. DELIMITER ;
617. -- Operazione 2: Registrazione Autovettura
619. **DROP** **PROCEDURE** IF EXISTS RegistrazioneAutovettura;
620. DELIMITER $$
621. **CREATE** **PROCEDURE** RegistrazioneAutovettura(IN `Targa` **char**(50),
622. IN `CodiceProponente` **char**(50),
623. IN `CodFiscale` **char**(50),
624. IN `AnnoImmatricolazione` **int**(11),
625. IN `CostoOperativo` **int**(11) unsigned,
626. IN `CostoUsura` **int**(11) unsigned,
627. IN `CostoExtra` **int**(11) unsigned,
628. IN `ChilometriPercorsi` **int**(11) unsigned,
629. IN `Disponibilita` **char**(50),
630. IN `CarburanteDisponibile` **float**(13,2) unsigned,
631. IN `NomeModello` **char**(50),
632. IN `Connettivita` bool,
633. IN `Tavolini` bool,
634. IN `TettoVetro` bool,
635. IN `DimensioneBagagliaio` **int**(11) unsigned,
636. IN `RumoreAbitacolo` **int**(11) unsigned
637. )
638. **BEGIN**
639. IF `CodiceProponente` NOT IN (**SELECT** `CodiceProponente`
640. **FROM** `proponente` P
641. **WHERE** P.`CodiceProponente` = `CodiceProponente`) **THEN**
642. **INSERT** **INTO** `proponente`
643. **VALUES**(`CodiceProponente`, `CodFiscale`);
644. **END** IF;
646. **INSERT** **INTO** `autovettura`
647. **VALUES**(`Targa`,
648. `CodiceProponente`,
649. `AnnoImmatricolazione`,
650. `CostoOperativo`,
651. `CostoUsura`,
652. `CostoExtra`,
653. `ChilometriPercorsi`,
654. `Disponibilita`,
655. `CarburanteDisponibile`,
656. `NomeModello`);
658. **INSERT** **INTO** `optional`
659. **VALUES**(`Targa`, `Connettivita`, `Tavolini`, `TettoVetro`, `DimensioneBagagliaio`, `RumoreAbitacolo`);
661. **END** $$
663. DELIMITER ;
665. -- Operazione 3: Prenotazione Car Sharing
667. **DROP** **PROCEDURE** IF EXISTS PrenotazioneCarsharing;
668. DELIMITER $$
669. **CREATE** **PROCEDURE** PrenotazioneCarSharing(IN `Targa` **char**(50),
670. IN `CodiceFruitore` **char**(50),
671. IN `IDNoleggio` **char**(50),
672. IN `CodFiscale` **char**(50),
673. IN `DataInizio` **date**,
674. IN `DataFine` **date**
675. )
676. **BEGIN**
677. IF `CodFiscale` NOT IN (**SELECT** `CodFiscale`
678. **FROM** `fruitore` F
679. **WHERE** F.`CodFiscale` = `CodFiscale`) **THEN**
680. **INSERT** **INTO** `fruitore`
681. **VALUES**(`CodiceFruitore`, `CodFiscale`);
682. **END** IF;
684. **INSERT** **INTO** `noleggio`
685. **VALUES**(`IDNoleggio`, `CodiceFruitore`, `Targa`, `DataInizio`, NULL, `DataFine`, NULL);
687. **INSERT** **INTO** `prenotazionen`(`CodiceFruitore`, `IDNoleggio`)
688. **VALUES**(`CodiceFruitore`, `IDNoleggio`);
689. **END** $$
691. DELIMITER ;
693. -- Operazione 4: Restituzione Noleggio
695. **DROP** **PROCEDURE** IF EXISTS RestituzioneNoleggio;
696. DELIMITER $$
697. **CREATE** **PROCEDURE** RestituzioneNoleggio(IN `IDNoleggio` **char**(50),
698. IN `Targa` **char**(50),
699. IN `ChilometriPercorsi` **char**(50),
700. IN `CarburanteFinenoleggio` **char**(50)
701. )
702. **BEGIN**
703. IF `CarburanteFineNoleggio` >= 0.95\*(**SELECT** `CarburanteDisponibile`
704. **FROM** `autovettura` A
705. **WHERE** A.`Targa` = `Targa`
706. ) **THEN**
707. **UPDATE** `noleggio` N
708. **SET** N.`CarburanteFineNoleggio` = `CarburanteFineNoleggio`
709. **WHERE** N.`IDNoleggio` = `IDNoleggio`;
711. **UPDATE** `autovettura` A
712. **SET** A.`ChilometriPercorsi` = `ChilometriPercorsi`
713. **WHERE** A.`Targa` = `Targa`;
714. **END** IF;
715. **END** $$
717. DELIMITER ;
719. -- Operazione 5: Registrazione Sinistro
721. **DROP** **PROCEDURE** IF EXISTS RegistrazioneSinistro;
722. DELIMITER $$
723. **CREATE** **PROCEDURE** RegistrazioneSinistro(IN `IDNoleggio` **char**(50),
724. IN `**Timestamp**` datetime,
725. IN `Dinamica` **char**(250),
726. IN `CasaAutomobilistica` **char**(50),
727. IN `Modello` **char**(50),
728. IN `Targa` **char**(50)
729. )
730. **BEGIN**
731. **INSERT** **INTO** `incidente`
732. **VALUES**(`IDNoleggio`, `**Timestamp**`, `Dinamica`);
734. **INSERT** **INTO** `autocoinvolta`
735. **VALUES**(`Targa`, `IDNoleggio`, `**Timestamp**`, `CasaAutomobilistica`, `Modello`);
736. **END** $$
738. DELIMITER ;
740. -- Operazione 6: Creazione Pool
742. **DROP** **PROCEDURE** IF EXISTS CreazionePool;
743. DELIMITER $$
744. **CREATE** **PROCEDURE** CreazionePool(IN `CodiceProponente` **char**(50),
745. IN `IDPool` **char**(50),
746. IN `CodiceTragitto` **char**(50),
747. IN `Targa` **char**(50),
748. IN `Flessibilita` **char**(50),
749. IN `NumeroPosti` **int**(11),
750. IN `OreChiusura` **int**(11),
751. IN `Spesa` **float**(13, 2),
752. IN `Partenza` datetime,
753. IN `GiornoArrivo` datetime
754. )
755. **BEGIN**
756. **INSERT** **INTO** `pool`
757. **VALUES**(`IDPool`,
758. `CodiceTragitto`,
759. `CodiceProponente`,
760. `Targa`,
761. `Flessibilita`,
762. `NumeroPosti`,
763. `OreChiusura`,
764. `Spesa`,
765. `Partenza`,
766. `GiornoArrivo`,
767. 'aperto',
768. 0,
769. NULL);
770. **END** $$
772. DELIMITER ;
774. -- Operazione 7: Proposta Variazione Pool
776. **DROP** **PROCEDURE** IF EXISTS PropostaVariazionePool;
777. DELIMITER $$
778. **CREATE** **PROCEDURE** PropostaVariazionePool(IN `IDPool` **char**(50),
779. IN `CodiceTragitto` **char**(50),
780. IN `IDFruitore` **char**(50)
781. )
782. **BEGIN**
783. **DECLARE** Entita **char**(50);
784. **DECLARE** Lunghezza **int**(11) unsigned;
785. **SET** Lunghezza = (**SELECT** `LunghezzaTragitto`
786. **FROM** `tragitto` T
787. **WHERE** T.`CodiceTragitto` = `CodiceTragitto`
788. );
789. CASE
790. **WHEN** Lunghezza <= 2 **THEN**
791. **SET** Entita = 'bassa';
792. **WHEN** (Lunghezza > 2 AND Lunghezza <= 5) **THEN**
793. **SET** Entita = 'media';
794. **WHEN** (Lunghezza > 5 AND Lunghezza <= 10) **THEN**
795. **SET** Entita = 'alta';
796. **END** CASE;
798. **INSERT** **INTO** `variazione`(`IDPool`, `IDFruitore`, `Entita`, `CodiceTragitto`)
799. **VALUES**(`IDPool`,
800. `IDFruitore`,
801. Entita,
802. `CodiceTragitto`
803. );
804. **END** $$
806. DELIMITER ;
808. -- Operazione 8: Chiamata Ride Sharing
810. **DROP** **PROCEDURE** IF EXISTS ChiamataRideSharing;
811. DELIMITER $$
812. **CREATE** **PROCEDURE** ChiamataRideSharing(IN `CodFiscale` **char**(50),
813. IN `CodiceFruitore` **char**(50),
814. IN `PosizioneFruitore` **char**(50),
815. IN `Destinazione` **char**(50),
816. IN `TimestampChiamata` datetime,
817. IN `Codice` **char**(50),
818. IN `TimestampRisposta` datetime
819. )
820. **BEGIN**
821. IF `CodFiscale` NOT IN (**SELECT** `CodFiscale`
822. **FROM** `fruitore` F
823. **WHERE** F.`CodFiscale` = `CodFiscale`) **THEN**
824. **INSERT** **INTO** `fruitore`
825. **VALUES**(`CodiceFruitore`, `CodFiscale`);
826. **END** IF;
828. **INSERT** **INTO** `chiamatasharing`(`Codice`, `CodiceFruitore`, `PosizioneFruitore`, `Destinazione`, `TimestampChiamata`, `TimestampRisposta`)
829. **VALUES**(`Codice`, `CodiceFruitore`, `PosizioneFruitore`, `Destinazione`, `TimestampChiamata`, `TimestampRisposta`);
830. **END** $$
832. DELIMITER ;
834. -- Operazione 9: Inserimento di una recensione
836. **DROP** **PROCEDURE** IF EXISTS InserimentoRecensione;
837. DELIMITER $$
838. **CREATE** **PROCEDURE** InserimentoRecensione(IN `TestoValutazione` **char**(50),
839. IN `CodiceTragitto` **char**(50),
840. IN `CodFiscale` **char**(50),
841. IN `Valutato` **char**(50),
842. IN `Codice` **char**(50),
843. IN `Serieta` **int**(11) unsigned,
844. IN `Comportamento` **int**(11) unsigned,
845. IN `PiacereViaggio` **int**(11) unsigned,
846. IN `GiudizioPersona` **int**(11) unsigned,
847. IN `Ruolo` **char**(50)
848. )
849. **BEGIN**
850. **INSERT** **INTO** `valutazione`
851. **VALUES**(`Codice`, `CodiceTragitto`, `CodFiscale`, `TestoValutazione`, `Comportamento`, `Serieta`, `PiacereViaggio`, `GiudizioPersona`, `Ruolo`, `Valutato`);
852. **END** $$
854. DELIMITER ;
856. -- Operazione 10: Visualizzazione della valutazione media di un utente
858. **DROP** **PROCEDURE** IF EXISTS VisualizzaValutazioneMedia;
859. DELIMITER $$
860. **CREATE** **PROCEDURE** VisualizzaValutazioneMedia(IN `CodFiscale` **char**(50), **OUT** `MediaValutazioni` **float**)
861. **BEGIN**
862. **SELECT** `ValutazioneMedia` **into** `MediaValutazioni`
863. **FROM** `Utente` U
864. **WHERE** U.`CodFiscale` = `CodFiscale`;
865. **END** $$
867. DELIMITER ;

870. -- Operazione 11: Ricerca Pool ordinati per lunghezza del tragitto
872. **DROP** **PROCEDURE** IF EXISTS RicercaPoolOrdineLunghezza;
873. DELIMITER $$
874. **CREATE** **PROCEDURE** RicercaPoolOrdineLunghezza(IN `PosizionePartenza` **char**(50), IN `PosizioneArrivo` **char**(50))
875. **BEGIN**
876. **SELECT** `IDPool`, `LunghezzaTragitto`
877. **FROM** `pool` NATURAL JOIN `tragitto` T
878. **WHERE** `Stato` = 'aperto' AND
879. T.`PosizionePartenza` = `PosizionePartenza` AND
880. T.`PosizioneArrivo` = `PosizioneArrivo`
881. **ORDER** **BY** `LunghezzaTragitto`, `Partenza` **DESC**;
882. **END** $$
884. -- Analytics
885. DELIMITER ;
886. **SET** NAMES latin1;
887. USE `carsharing`;
888. **SET** FOREIGN\_KEY\_CHECKS = 1;
889. **SET** **GLOBAL** EVENT\_SCHEDULER = **ON**;
891. **DROP** **TABLE** IF EXISTS `MV\_Affidabilita`;
892. **CREATE** **TABLE** `MV\_Affidabilita`(
893. `CodFiscale` **char**(50) NOT NULL,
894. `Affidabilita` **float**(13, 2) unsigned **DEFAULT** NULL,
895. **PRIMARY** **KEY** (`CodFiscale`)
896. )ENGINE=InnoDB **DEFAULT** CHARSET=latin1;
898. **INSERT** **INTO** `MV\_Affidabilita`
899. **SELECT** U.`CodFiscale`, NULL
900. **FROM** `Utente` U;
902. **DROP** **PROCEDURE** IF EXISTS scan\_usr;
903. DELIMITER $$
904. **CREATE** **PROCEDURE** scan\_usr()
905. **BEGIN**
906. **DECLARE** finished **integer**(11) **DEFAULT** 0;
908. **DECLARE** `curr\_codfiscale` **varchar**(50);
910. **DECLARE** scan **CURSOR** **FOR**
911. **SELECT** `CodFiscale`
912. **FROM** `utente`;
914. **DECLARE** **CONTINUE** HANDLER **FOR** NOT FOUND
915. **SET** finished = 1;
917. **TRUNCATE** `MV\_Affidabilita`;
919. **INSERT** **INTO** `MV\_Affidabilita`
920. **SELECT** `CodFiscale`, NULL
921. **FROM** `utente`;
923. **OPEN** scan;
924. scan : LOOP
925. **FETCH** scan **INTO** `curr\_codfiscale`;
926. IF finished = 1 **THEN**
927. LEAVE scan;
928. **END** IF;
930. CALL aggiorna\_affidabilita\_utente(`curr\_codfiscale`);
931. **END** LOOP;
932. **CLOSE** scan;
933. **END** $$
934. DELIMITER ;
936. **DROP** **PROCEDURE** IF EXISTS aggiorna\_affidabilita\_utente;
937. DELIMITER $$
938. **CREATE** **PROCEDURE** aggiorna\_affidabilita\_utente(IN `CodiceFiscale` **char**(50))
939. **BEGIN**
940. **DECLARE** `ValutazioneMedia` **float**(13, 2) **DEFAULT** 0.0;
941. **DECLARE** `PartenzePoolRispettate` **int**(11) **DEFAULT** NULL;
942. **DECLARE** `PartenzeRideSharingRispettate` **int**(11) **DEFAULT** NULL;
943. **DECLARE** `IndicePuntualita` **float**(13, 2) **DEFAULT** 1.0;
944. **DECLARE** `CodiceProponente` **char**(50) **DEFAULT** NULL;
945. **DECLARE** `PoolTotali` **int**(11) **DEFAULT** NULL;
946. **DECLARE** `RidesharingTotali` **int**(11) **DEFAULT** NULL;
948. **SET** `CodiceProponente` = (**SELECT** P.`CodiceProponente`
949. **FROM** `Proponente` P
950. **WHERE** `CodiceFiscale` = P.`CodFiscale`
951. );
953. CALL VisualizzaValutazioneMedia(`CodiceFiscale`, `ValutazioneMedia`);
954. **SET** `PartenzePoolRispettate` = (**SELECT** count(\*)
955. **FROM** `pool` P
956. **WHERE** P.`PartenzaEffettiva` **IS** NOT NULL AND
957. P.`PartenzaEffettiva` < DATE\_ADD(`Partenza`, INTERVAL 5 **MINUTE**) AND
958. `CodiceProponente` = P.`CodiceProponente`
959. );
961. IF `PartenzePoolRispettate` **IS** NULL **THEN**
962. **SET** `PartenzePoolRispettate` = 0;
963. **END** IF;
965. **SET** `PartenzeRideSharingRispettate` = (**SELECT** count(\*)
966. **FROM** `ridesharing` R
967. **WHERE** R.`PartenzaEffettiva` **IS** NOT NULL AND
968. R.`PartenzaEffettiva` < DATE\_ADD(`OrarioPartenza`, INTERVAL 5 **MINUTE**) AND
969. `CodiceProponente` = R.`CodiceProponente`
970. );
972. IF `PartenzeRideSharingRispettate` **IS** NULL **THEN**
973. **SET** `PartenzeRideSharingRispettate` = 0;
974. **END** IF;
976. **SET** `PoolTotali` = (**SELECT** count(\*)
977. **FROM** `pool` P
978. **WHERE** P.`PartenzaEffettiva` **IS** NOT NULL AND
979. `CodiceProponente` = P.`CodiceProponente`
980. );
982. IF `PoolTotali` **IS** NULL **THEN**
983. **SET** `PoolTotali` = 0;
984. **END** IF;
986. **SET** `RideSharingTotali` = (**SELECT** count(\*)
987. **FROM** `ridesharing` R
988. **WHERE** R.`PartenzaEffettiva` **IS** NOT NULL AND
989. `CodiceProponente` = R.`CodiceProponente`
990. );
992. IF `RideSharingTotali` **IS** NULL **THEN**
993. **SET** `RideSharingTotali` = 0;
994. **END** IF;
996. IF (`PoolTotali` <> 0 OR `RideSharingTotali` <> 0) **THEN**
997. **SET** `IndicePuntualita` = (`PartenzePoolRispettate` + `PartenzeRideSharingRispettate`) / (`PoolTotali` + `RidesharingTotali`);
998. **END** IF;
1000. **UPDATE** `MV\_Affidabilita`
1001. **SET** `Affidabilita` = (`ValutazioneMedia` + `IndicePuntualita`\*5 ) / 2
1002. **WHERE** `CodiceFiscale` = `CodFiscale`;
1003. **END** $$
1005. DELIMITER ;
1007. **DROP** EVENT IF EXISTS aggiorna\_affidabilita;
1008. DELIMITER $$
1010. **CREATE** EVENT aggiorna\_affidabilita
1011. **ON** SCHEDULE EVERY 7 DAY
1012. DO
1013. **BEGIN**
1014. CALL scan\_usr();
1015. **END** $$
1016. DELIMITER ;
1017. **DROP** **TABLE** IF EXISTS `MV\_Viabilita`;
1018. **CREATE** **TABLE** `MV\_Viabilita`(
1019. `IDStrada` **char**(50) NOT NULL,
1020. `Numero` **int**(11) unsigned NOT NULL,
1021. `Viabilita` **char**(50) **DEFAULT** 'regolare',
1022. `TempoMedioRegolare` **time** NOT NULL,
1023. **PRIMARY** **KEY** (`IDStrada`, `Numero`)
1024. )ENGINE=InnoDB **DEFAULT** CHARSET=latin1;
1026. **INSERT** **INTO** `MV\_Viabilita`
1027. **SELECT** C.`IDStrada`, C.`Numero`, 'regolare', C.`TempoMedio`
1028. **FROM** `chilometro` C;
1030. **DROP** **PROCEDURE** IF EXISTS scan\_numero;
1031. DELIMITER $$
1032. **CREATE** **PROCEDURE** scan\_numero(IN `curr\_strada` **char**(50))
1033. **BEGIN**
1034. **DECLARE** finished **integer**(11) **DEFAULT** 0;
1036. **DECLARE** `curr\_num` **varchar**(50);
1038. **DECLARE** scan **CURSOR** **FOR**
1039. **SELECT** `Numero`
1040. **FROM** `chilometro`;
1042. **DECLARE** **CONTINUE** HANDLER **FOR** NOT FOUND
1043. **SET** finished = 1;
1045. **OPEN** scan;
1046. scan : LOOP
1047. **FETCH** scan **INTO** `curr\_num`;
1048. IF finished = 1 **THEN**
1049. LEAVE scan;
1050. **END** IF;
1052. CALL aggiorna\_tempi\_medi(`curr\_strada`, `curr\_num`);
1053. **END** LOOP;
1054. **CLOSE** scan;
1055. **END** $$
1056. DELIMITER ;
1058. **DROP** **PROCEDURE** IF EXISTS scan\_strada;
1059. DELIMITER $$
1060. **CREATE** **PROCEDURE** scan\_strada()
1061. **BEGIN**
1062. **DECLARE** finished **integer**(11) **DEFAULT** 0;
1064. **DECLARE** `curr\_strada` **varchar**(50);
1066. **DECLARE** scan **CURSOR** **FOR**
1067. **SELECT** `IDSTrada`
1068. **FROM** `chilometro`;
1070. **DECLARE** **CONTINUE** HANDLER **FOR** NOT FOUND
1071. **SET** finished = 1;
1073. **OPEN** scan;
1074. scan : LOOP
1075. **FETCH** scan **INTO** `curr\_strada`;
1076. IF finished = 1 **THEN**
1077. LEAVE scan;
1078. **END** IF;
1080. CALL scan\_numero(`curr\_strada`);
1081. **END** LOOP;
1082. **CLOSE** scan;
1083. **END** $$
1084. DELIMITER ;
1086. **DROP** **PROCEDURE** IF EXISTS aggiorna\_tempi\_medi;
1087. DELIMITER $$
1088. **CREATE** **PROCEDURE** aggiorna\_tempi\_medi(IN `IDSTrada` **char**(50), IN `Numero` **int**(11))
1089. **BEGIN**
1090. **DECLARE** DiffTempi **float**(13, 2);
1091. **DECLARE** DiffSpazio **float**(13, 2);
1092. **DECLARE** TempoMedioSecondi **int**(11);
1094. **SET** DiffTempi = (**SELECT** AVG(time\_to\_sec(timediff(P2.`**Timestamp**`, P1.`**Timestamp**`)))
1095. **FROM** `posizione` P1 **INNER** JOIN `posizione` P2 **ON** P1.`CodiceTragitto` = P2.`CodiceTragitto`
1096. **WHERE** P1.`**Timestamp**`< P2.`**Timestamp**` AND current\_timestamp < DATE\_ADD(P1.`**Timestamp**`, INTERVAL 1 **HOUR**) AND
1097. P2.`Numero` = P1.`Numero` + 1 AND
1098. P1.`IDPosizione` IN(
1099. **SELECT** `IDPosizione`
1100. **FROM** `relativa` R
1101. **WHERE** R.`IDSTrada` = `IDSTrada` AND
1102. R.`Numero` = `Numero`
1103. ) AND
1104. P2.`IDPosizione` IN(
1105. **SELECT** `IDPosizione`
1106. **FROM** `relativa` R
1107. **WHERE** R.`IDSTrada` = `IDSTrada` AND
1108. R.`Numero` = `Numero`
1109. )
1110. );
1112. **SET** DiffSpazio = (**SELECT** AVG(ACOS(SIN(P1.`Latitudine`\*PI()/180)\*SIN(P2.`Latitudine`\*PI()/180) + COS(P1.`Latitudine`\*PI()/180)\*COS(P2.`Latitudine`\*PI()/180)\*COS(P2.`Longitudine`\*PI()/180-P1.`Longitudine`\*PI()/180))\*6371)
1113. **FROM** `posizione` P1 **INNER** JOIN `posizione` P2 **ON** P1.`CodiceTragitto` = P2.`CodiceTragitto`
1114. **WHERE** P1.`**Timestamp**`< P2.`**Timestamp**` AND current\_timestamp < DATE\_ADD(P1.`**Timestamp**`, INTERVAL 1 **HOUR**) AND
1115. P2.`Numero` = P1.`Numero` + 1 AND
1116. P1.`IDPosizione` IN(
1117. **SELECT** `IDPosizione`
1118. **FROM** `relativa` R
1119. **WHERE** R.`IDSTrada` = `IDSTrada` AND
1120. R.`Numero` = `Numero`
1121. ) AND
1122. P2.`IDPosizione` IN(
1123. **SELECT** `IDPosizione`
1124. **FROM** `relativa` R
1125. **WHERE** R.`IDSTrada` = `IDSTrada` AND
1126. R.`Numero` = `Numero`
1127. )
1128. );
1130. **SET** TempoMedioSecondi = 1/(DiffSpazio/DiffTempi);
1131. IF TempoMedioSecondi **IS** NOT NULL **THEN**
1132. **UPDATE** `chilometro` C
1133. **SET** C.`TempoMedio` = sec\_to\_time(TempoMedioSecondi)
1134. **WHERE** C.`IDStrada` = `IDSTrada` AND C.`Numero` = `Numero`;
1135. **END** IF;
1137. **END** $$
1139. DELIMITER ;
1141. **DROP** EVENT IF EXISTS aggiorna\_tempi\_medi\_percorrenza;
1142. DELIMITER $$
1144. **CREATE** EVENT aggiorna\_tempi\_medi\_percorrenza
1145. **ON** SCHEDULE EVERY 1 **HOUR**
1146. DO
1147. **BEGIN**
1148. CALL scan\_strada();
1149. **END** $$
1151. DELIMITER ;
1152. **DROP** **TRIGGER** IF EXISTS `aggiornamento\_tempi\_medi`;
1153. DELIMITER $$
1154. **CREATE** **TRIGGER** `aggiornamento\_tempi\_medi`
1155. **AFTER** **UPDATE** **ON** `chilometro` **FOR** EACH ROW
1156. **BEGIN**
1157. **DECLARE** `TMR` **time**;
1158. **SET** `TMR` = (**SELECT** `TempoMedioRegolare`
1159. **FROM** `MV\_Viabilita`
1160. **WHERE** NEW.`Numero` = `Numero` AND NEW.`IDStrada` = `IDStrada`
1161. );
1162. IF (time\_to\_sec(NEW.`TempoMedio`) > 2\*time\_to\_sec(`TMR`)) **THEN**
1163. **UPDATE** `MV\_Viabilita`
1164. **SET** `Viabilita` = 'compromessa'
1165. **WHERE** NEW.`Numero` = `Numero` AND NEW.`IDStrada` = `IDStrada`;
1166. **ELSE**
1167. **UPDATE** `MV\_Viabilita`
1168. **SET** `Viabilita` = 'regolare'
1169. **WHERE** NEW.`Numero` = `Numero` AND NEW.`IDStrada` = `IDStrada`;
1170. **END** IF;
1171. **END** $$
1173. DELIMITER ;