

Progetto Database

Sistema di gestione di una flotta di veicoli

Domenico G. Blanco

1000001201

25 giugno 2024

Analisi della realtà di riferimento	3
Glossario dei termini	4
Approccio alla progettazione concettuale	5
Individuazione delle entità	5
Creazione delle relazioni	12
Creazione dello schema E-R (intermedio)	17
Inserimento delle chiavi esterne	17
Creazione dello schema E-R finale	19
Dizionario delle entità	20
Dizionario delle relazioni	20
Progettazione logica	21
Specifiche sulle operazioni	21
Volume dei dati	23
Volume delle frequenze	23
Analisi delle ridondanze	24
Modello logico	26
Progettazione tabelle in SQL	27
Inserimento dati in SQL	29
Query in SQL	30
Eventi e Trigger	32
Utenti e permessi	41
Progettazione fisica	42
Schema fisico	42

Analisi della realtà di riferimento

Un'azienda locale vuole realizzare un sistema per la gestione di una flotta di veicoli aziendali guidabili dai dipendenti.

Ogni dipendente deve essere registrato all'assunzione, fornendo: matricola, nome, cognome ed eventuale possesso di una o più patenti.

Ogni veicolo deve essere registrato nel sistema nel momento in cui viene ricevuto dall'azienda. La registrazione consiste nell'inserire i seguenti dati: marca, modello, telaio, potenza del motore, tipologia del motore.

Un dipendente può compiere una trasferta per mezzo di un veicolo e deve includere, inoltre, ulteriori informazioni tra le quali: stato, destinazione, data d'inizio; al termine di una trasferta andranno aggiunti anche i KM percorsi e la data di fine.

Durante una trasferta, per i veicoli endotermici può essere eseguita l'operazione di rifornimento da un dipendente, che aggiunge carburante al serbatoio del veicolo. Per i veicoli elettrici la ricarica avviene in azienda, escludendoli quindi dall'operazione di rifornimento.

Una trasferta necessariamente comprende un luogo da raggiungere. Al fine di avere una stima su quelli che possono essere i costi accessori, si include l'ubicazione da raggiungere e una stima dei KM.

In caso di guasto, sinistro, scadenza revisione o raggiungimento dei KM necessari al tagliando successivo di un veicolo, deve essere programmata la sua manutenzione, specificando: telaio del veicolo, tipo di manutenzione, inizio e fine.

Il tipo di manutenzione indica se trattasi di guasto, tagliando, revisione o sinistro; in quest'ultimo caso è necessario specificare a chi sia stato concesso il veicolo al momento dell'incidente.

Il database serve quindi a tenere traccia dei veicoli presenti, della loro manutenzione, delle tratte percorse e dei costi derivati dal carburante, con attenzione a chi ha guidato ciascun veicolo e, al fine di limitare le spese, proporre solo veicoli elettrici nelle trasferte considerabili brevi.

Glossario dei termini

Termino	Descrizione	Sinonimo	Termini collegati
Dipendente	Persona assunta dall'azienda che può prendere parte a una trasferta con un veicolo ed effettuare operazioni di rifornimento	Autista	Trasferta, Manutenzione, Rifornimento
Veicolo	Un mezzo di trasporto presente nel parco veicoli dell'azienda, può essere guidato da un dipendente durante una trasferta, può essere soggetto a operazioni di manutenzione ordinaria e/o straordinaria e può essere soggetto a operazioni di rifornimento	Mezzo di trasporto	Trasferta, Manutenzione, Rifornimento
Trasferta	Resoconto dell'assegnazione di un veicolo a un dipendente per scopi lavorativi	Viaggio	Dipendente, Veicolo, Luogo

Termino	Descrizione	Sinonimo	Termini collegati
Luogo	Posizione da raggiungere per mezzo di una trasferta lavorativa o luogo in cui si è svolta un'operazione di rifornimento	Ubicazione, Sede, Destinazione, Località	Trasferta, Rifornimento
Manutenzione	Operazione atta al mantenimento o miglioramento delle condizioni attuali del veicolo	Riparazione	Dipendente, Veicolo
Rifornimento	L'operazione che ha lo scopo di rifornire di carburante un veicolo aziendale		Dipendente, Veicolo, Luogo

Approccio alla progettazione concettuale

Si sceglie di creare il database sulla base dell'approccio bottom-up, nel quale si specifica prima il carattere dell'entità, specificando gli attributi e le chiavi e successivamente la vera e propria struttura del database, al fine di non incorrere in strutture o operazioni poco efficienti.

Individuazione delle entità

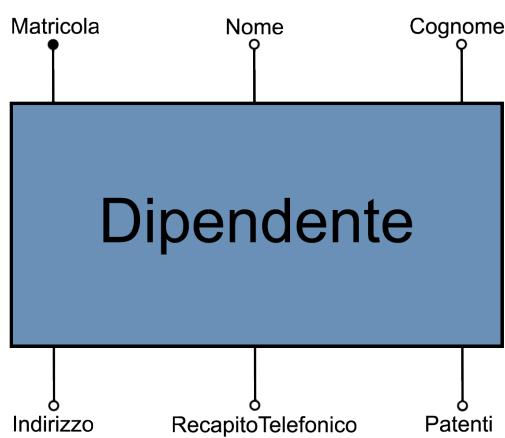
Entità Dipendente

Partendo dall'analisi del problema, si rende immediatamente necessario identificare i dipendenti dell'azienda attraverso l'entità **dipendente**.

Dipendente

L'entità ha come attributi tutti i dati necessari all'identificazione univoca del dipendente e alle sue eventuali idoneità di guida, rappresentate dal possesso di patenti.

- **Matricola:** chiave primaria dell'entità, rappresenta il numero di matricola assegnato al dipendente al momento dell'assunzione
- Nome: nome del dipendente
- Cognome: cognome del dipendente
- Indirizzo: indirizzo di residenza del dipendente
- RecapitoTelefonico: numero aziendale fornito al dipendente
- Patenti: lista di patenti fornite



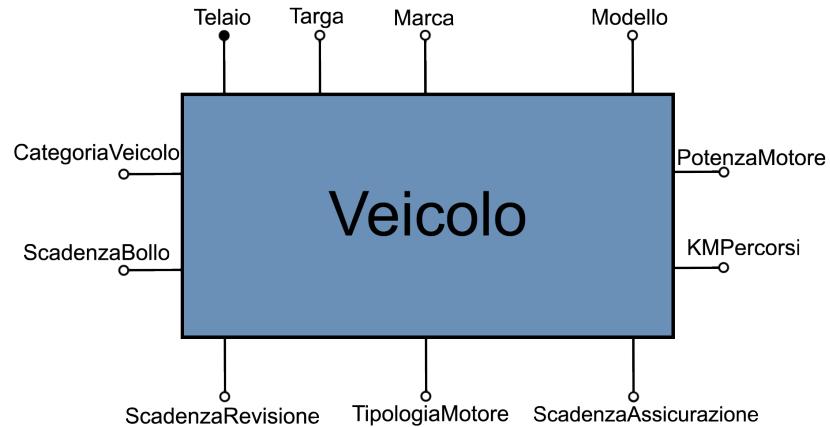
Entità Veicolo

Congiuntamente al dipendente, è necessario tenere traccia dei veicoli, identificando così l'entità **veicolo**.

Veicolo

Ciascun veicolo viene identificato univocamente per mezzo del proprio numero di telaio.

- **Telaio:** chiave primaria dell'entità, rappresentata dal numero di telaio del veicolo
- Marca: rappresenta il marchio del veicolo
- Modello: rappresenta il modello specifico del veicolo
- Targa: indica la targa associata al veicolo
- KMPercorsi: KM percorsi dal veicolo in totale
- PotenzaMotore: indica la potenza erogata dal motore in kW
- TipologiaMotore: differenzia i veicoli endotermici da veicoli elettrici
- CategoriaVeicolo: indica la macrocategoria di appartenenza del veicolo, permettendo di riconoscere la patente necessaria
- ScadenzaBollo: la data entro cui andrà rinnovato il pagamento del bollo del veicolo
- ScadenzaRevisione: la data entro cui andrà effettuata una nuova revisione del veicolo
- ScadenzaAssicurazione: la data entro cui andrà rinnovata l'assicurazione



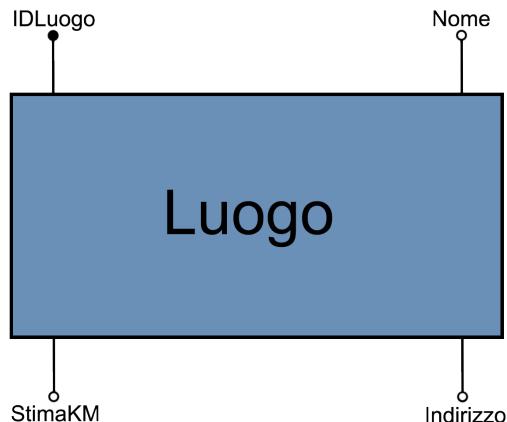
Entità Veicolo



Considerando le due entità precedenti, si rende necessario identificare le destinazioni raggiungibili per mezzo di un veicolo o in cui può essere effettuato il suo rifornimento, dando anche una stima dei KM che intercorrono tra l'azienda e la destinazione, portando così all'entità **luogo**.

Si identificano i seguenti attributi:

- **IDLuogo**: chiave primaria dell'entità
- Nome: nome dell'ubicazione
- Indirizzo: indirizzo del luogo
- StimaKM: una stima dei KM che intercorrono tra la sede dell'azienda e la destinazione



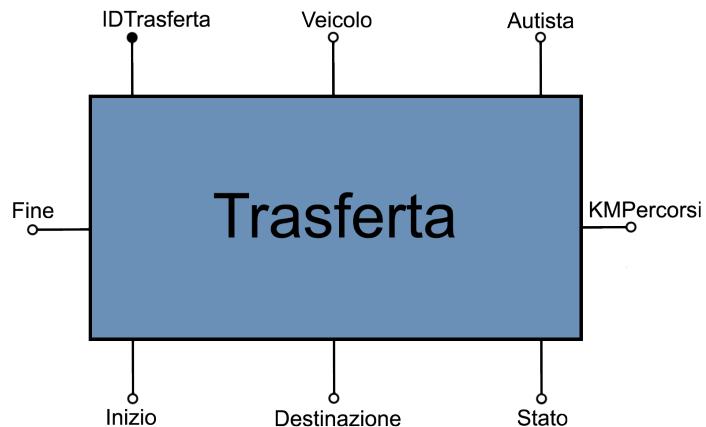
Entità Trasferta



Dopo aver identificato il dipendente e il veicolo, si ritiene opportuno identificare gli itinerari percorsi da un veicolo congiuntamente a un autista dalla sede aziendale verso una destinazione, pertanto si identifica l'entità **trasferta** con i seguenti attributi:

- **IDTrasferta**: chiave primaria dell'entità
- Veicolo: il telaio del veicolo assegnato per la trasferta in oggetto
- Autista: la matricola dell'autista per la trasferta in oggetto
- KMPercorsi: indica gli effettivi KM percorsi, questo attributo viene popolato al completamento della trasferta
- Stato: l'eventuale stato di completamento di una trasferta

- Destinazione: il luogo da raggiungere nella trasferta e che inciderà sull'offerta dei veicoli assegnabili.
- Inizio: la data e ora in cui inizia la trasferta
- Fine: la data e ora in cui si considera conclusa la trasferta



Le trasferte entro i 100KM vengono considerate brevi, perciò è preferibile assegnare un veicolo a propulsione elettrica.

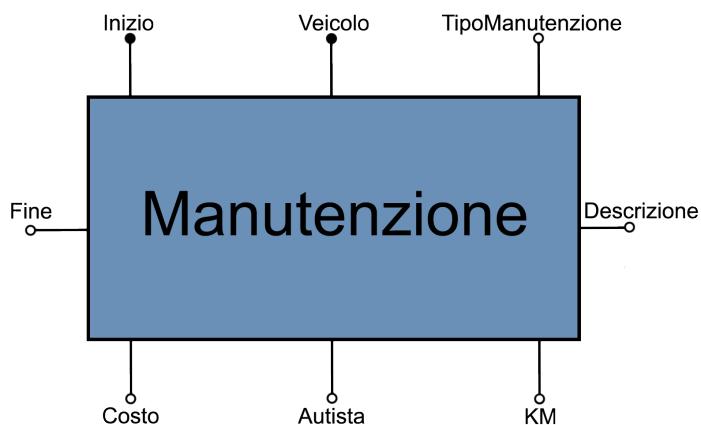
Entità Manutenzione



Monitorando il parco veicoli diventa necessario tenere traccia anche delle manutenzioni ordinarie e straordinarie di ciascun veicolo, individuando così l'entità **manutenzione** con i seguenti attributi:

- **Veicolo, Inizio**: chiavi primarie dell'entità, telaio rappresenta il numero di telaio del veicolo mentre inizio rappresenta la data e l'ora d'inizio della manutenzione

- TipoManutenzione: si differenzia in guasto, tagliando, revisione e sinistro.
Nel caso di sinistro deve essere specificato l'autista che ha causato il sinistro.
- Descrizione: campo preposto a contenere un breve sunto delle operazioni di manutenzione svolte
- Fine: data e ora di fine della manutenzione
- KM: KM registrati all'inizio e alla fine delle operazioni di manutenzione
- Autista: matricola dell'impiegato che ha causato il sinistro
- Costo: costo dell'attività manutentiva



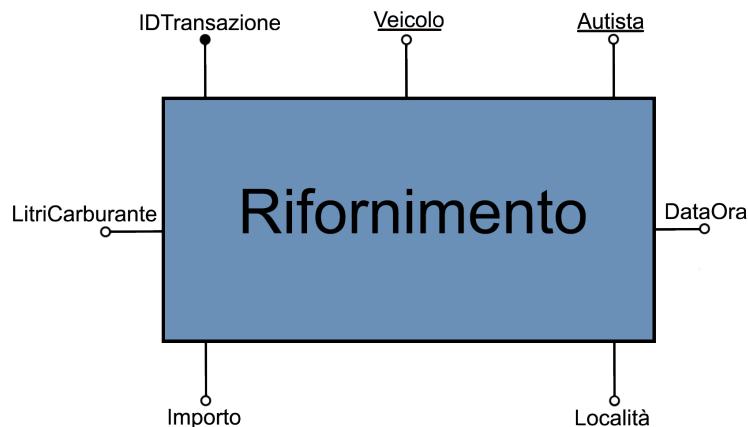
Considerando che alcune delle operazioni non possono essere imputabili all'azienda o all'autista, vedasi i casi di sinistro o riparazione in garanzia, si ammettono valori negativi e nulli per l'attributo Costo.

Entità rifornimento

Poiché ciascun veicolo a combustione endotermica può richiedere di effettuare un rifornimento durante una trasferta, si individua l'entità **rifornimento**.

Rifornimento

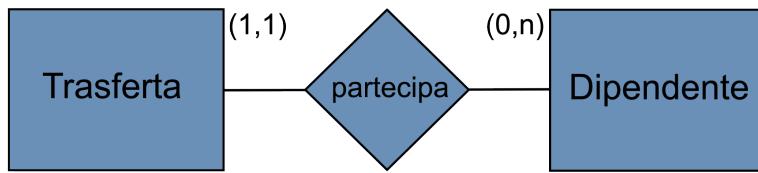
- **IDTransazione:** chiave primaria dell'entità, rappresenta l'identificativo della transazione riportata sulla ricevuta
- Veicolo: il telaio del veicolo su cui è stato effettuato rifornimento
- Autista: l'autista che ha effettuato il rifornimento
- DataOra: la data e l'ora riportati sulla ricevuta
- Località: il luogo presso cui è stato effettuato il rifornimento di carburante
- Importo: l'importo speso in carburante
- LitriCarburante: il totale in litri di carburante erogato



Creazione delle relazioni

Relazione "Partecipa"

L'entità Dipendente può essere messa in relazione con l'entità Trasferta attraverso la relazione **partecipa**.

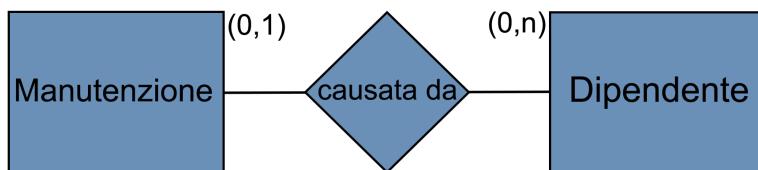


Regole di lettura:

- Un dipendente può prendere parte a nessuna o più trasferte
- A ciascuna trasferta partecipa un solo dipendente

Relazione “Causata da”

L'entità Dipendente può essere messa in relazione con l'entità Manutenzione per mezzo della relazione **causata da**.

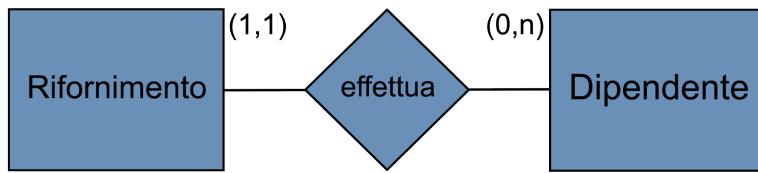


Regole di lettura:

- Un dipendente può causare nessuna o più operazioni di manutenzione
- A un'operazione di manutenzione può corrispondere un dipendente

Relazione “Effettua”

L'entità Dipendente può essere messa in relazione con l'entità Rifornimento attraverso la relazione **effettua**.

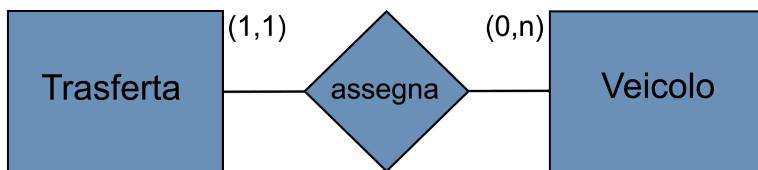


Regole di lettura:

- Un dipendente può effettuare nessuno o più rifornimenti
- Un'operazione di rifornimento è associata un dipendente

Relazione “Assegna”

L'entità Veicolo può essere messa in relazione con l'entità Trasferta per mezzo della relazione **assegna**.

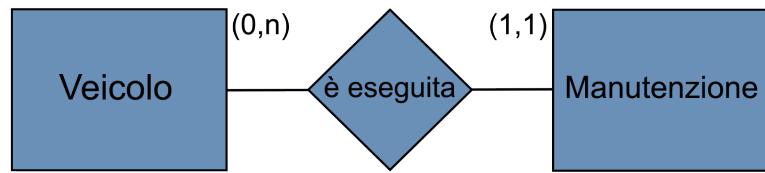


Regole di lettura:

- Un veicolo può essere assegnato a nessuna o più trasferte
- Ciascuna trasferta ha un veicolo assegnato

Relazione “È eseguita”

L'entità Veicolo può essere messa in relazione con l'entità Manutenzione attraverso la relazione **è eseguita**.

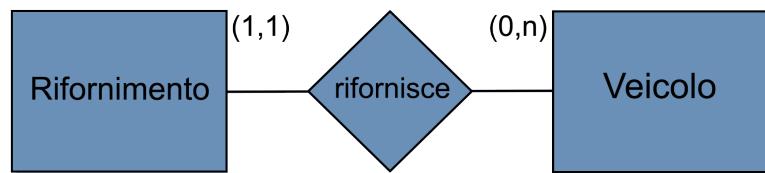


Regole di lettura:

- A un veicolo si possono effettuare nessuna o più operazioni di manutenzione
- A un'operazione di manutenzione corrisponde un veicolo

Relazione “Rifornisce”

L'entità Veicolo può essere messa in relazione con l'entità Rifornimento per mezzo della relazione **rifornisce**.

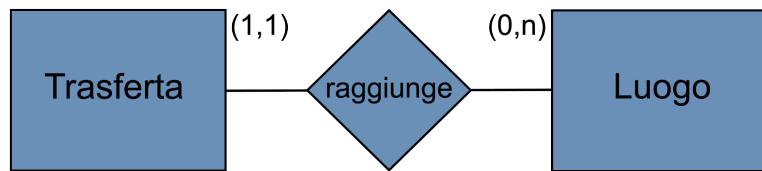


Regole di lettura:

- Un veicolo può essere rifornito nessuna o più volte
- Un'operazione di rifornimento rifornisce un veicolo

Relazione “Raggiunge”

L'entità Trasferta può essere messa in relazione con l'entità Luogo per mezzo della relazione **raggiunge**.

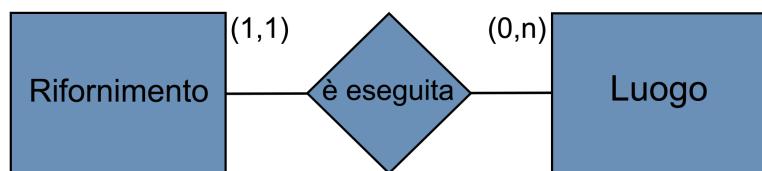


Regole di lettura:

- Un luogo può essere raggiunto da nessuna o più trasferte
- Una trasferta raggiunge un luogo

Relazione “È eseguita”

L'entità Rifornimento può essere messa in relazione con l'entità Luogo attraverso la relazione **è eseguita**.

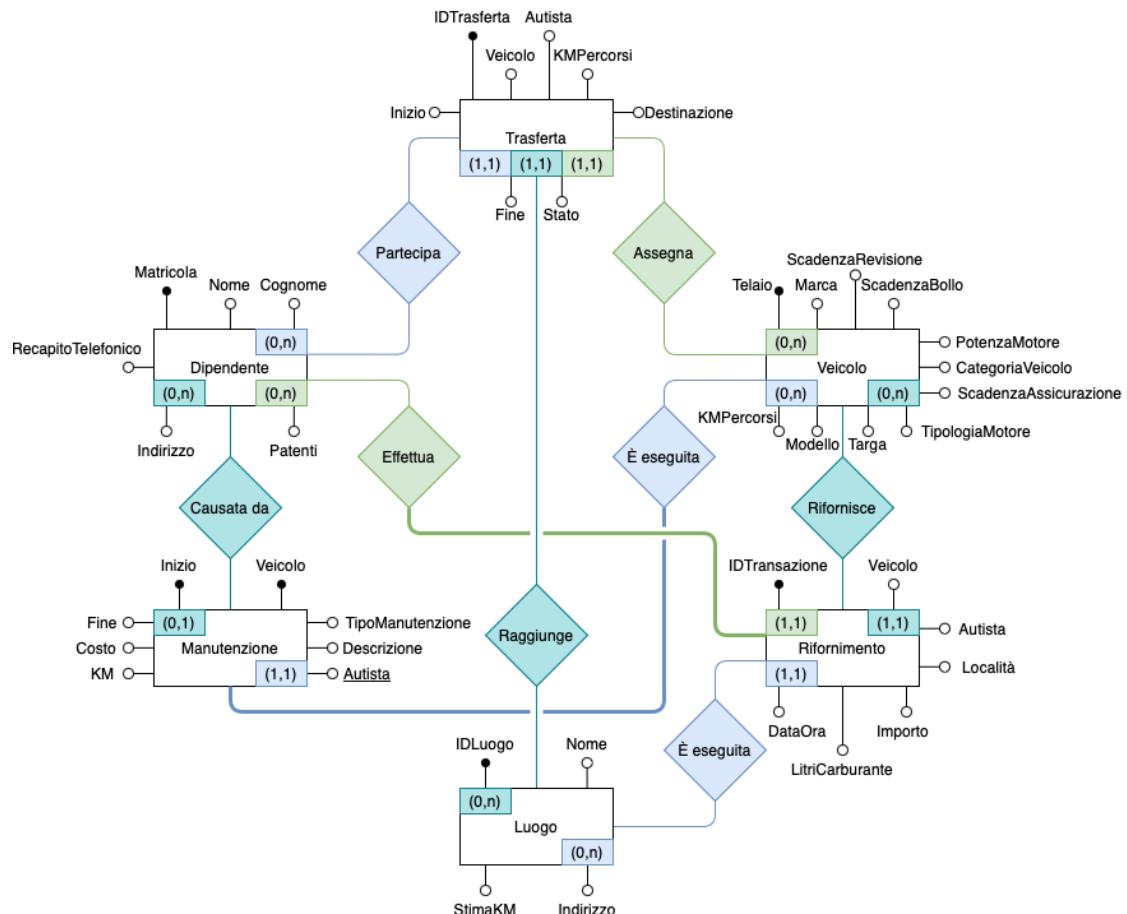


Regole di lettura:

- A un'operazione di rifornimento viene associato un luogo

- Un luogo può essere associato a nessuna o più operazioni di rifornimento

Creazione dello schema E-R (intermedio)

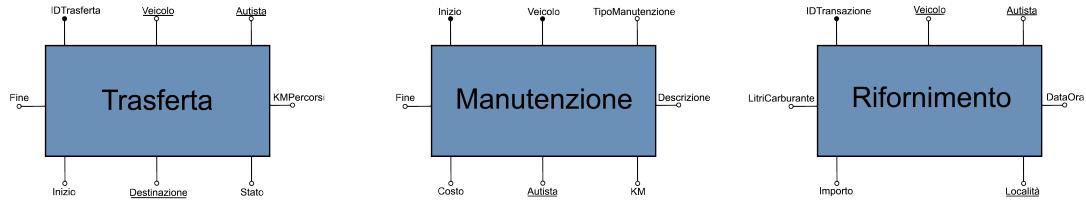


Inserimento delle chiavi esterne

Si rende quindi necessaria l'analisi degli attributi che verranno identificati come chiavi esterne per la creazione dello schema finale.

Chiave esterna: Veicolo

Questa chiave fa riferimento alla chiave primaria **Telaio** della tabella **Veicolo**. La chiave viene aggiunta alle tabelle Trasferta, Manutenzione e Rifornimento in modo da sapere a quale veicolo viene associata l'entità.



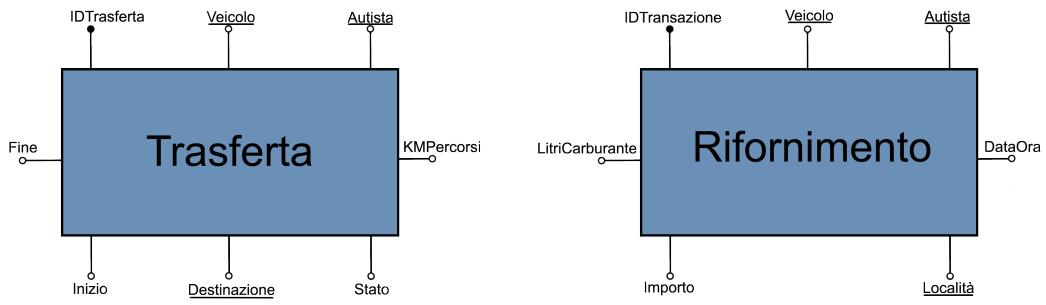
Chiave esterna: Autista

Questa chiave fa riferimento alla chiave primaria **Matricola** della tabella **Dipendente**. La chiave viene aggiunta alle tabelle Trasferta, Rifornimento e Manutenzione; in quest'ultima solo nel caso in cui il tipo di manutenzione corrisponda a un sinistro effettuato dall'autista.



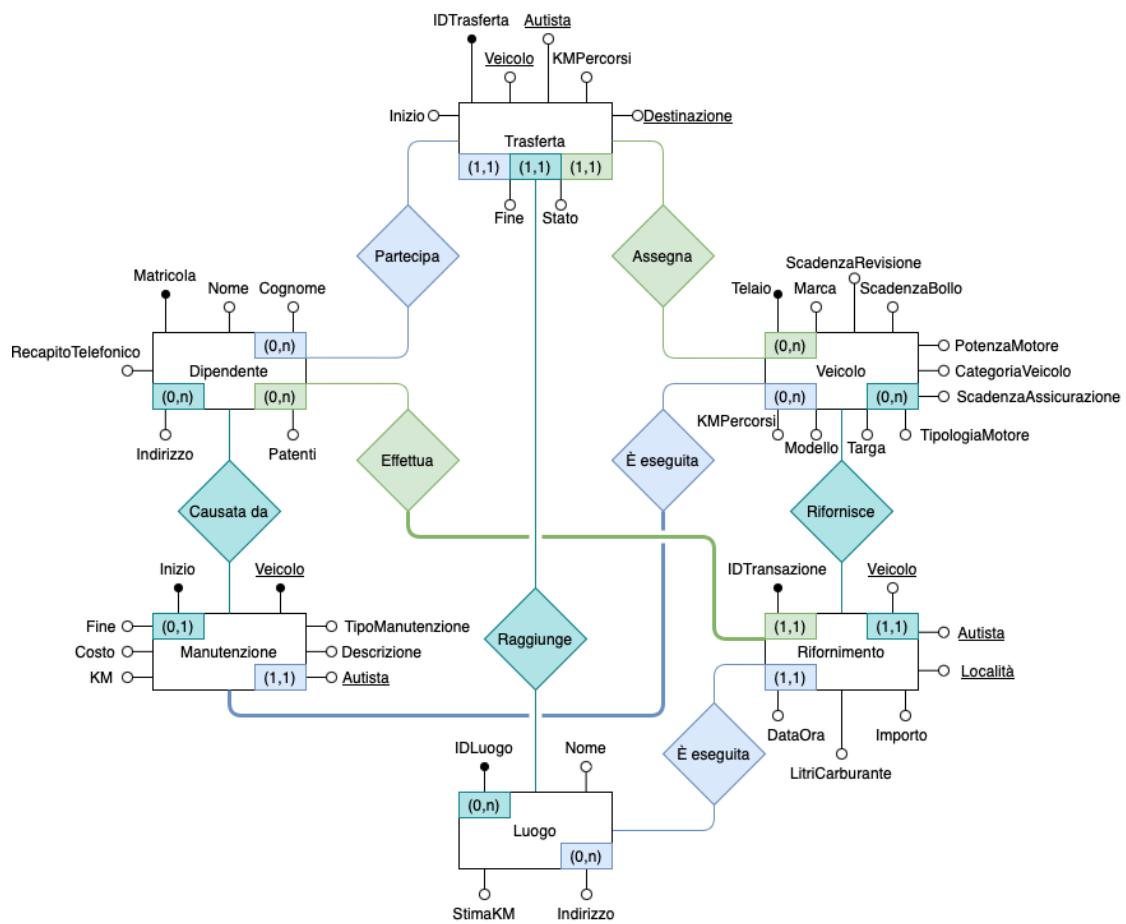
Chiave esterna: Località/Destinazione

Queste chiavi fanno riferimento alla chiave primaria **IDLuogo** della tabella **Luogo**. La chiave viene aggiunta alle tabelle Trasferta e Rifornimento per identificare i luoghi in cui un dipendente è tenuto a fermarsi.



Creazione dello schema E-R finale

Dopo aver inserito le chiavi esterne nelle entità e non essendoci relazioni di tipo molti a molti da decomporre, lo schema E-R diventa lo schema finale:



Dizionario delle entità

Entità	Descrizione	Attributi	Chiave/i
Dipendente	Elenco di tutti i dipendenti	Matricola, Nome, Cognome, Indirizzo, RecapitoTelefonico, Patenti	Matricola
Veicolo	Elenco del parco veicoli aziendale	Telaio, Marca, Modello, Targa, KMPercorsi, PotenzaMotore, TipologiaMotore, CategoriaVeicolo, ScadenzaBollo, ScadenzaRevisione, ScadenzaAssicurazione	Telaio
Luogo	Elenco dei luoghi raggiungibili per mezzo di una trasferta	IDLuogo, Nome, Indirizzo, StimaKM	IDLuogo
Trasferta	Storico delle trasferte aziendali	IDTrasferta, Veicolo, Autista, KMPercorsi, Stato, Destinazione, Inizio, Fine	IDTrasferta
Manutenzione	Storico delle manutenzioni effettuate sui veicoli	Veicolo, TipoManutenzione, Descrizione, Inizio, Fine, KM, Autista, Costo	Veicolo, Inizio
Rifornimento	Storico delle transazioni atte al rifornimento di un veicolo	IDTransazione, Veicolo, Autista, DataOra, Localita, Importo, LitriCarburante	IDTransazione

Dizionario delle relazioni

Relazione	Entità partecipanti	Descrizione	Attributi
Partecipa	Dipendente - Trasferta	Rappresenta la partecipazione di un dipendente a una trasferta	/

Relazione	Entità partecipanti	Descrizione	Attributi
Causata da	Dipendente - Manutenzione	Rappresenta un'operazione di manutenzione causata da un sinistro mentre alla guida vi era un dipendente	/
Effettua	Dipendente - Rifornimento	Rappresenta un dipendente che effettua un'operazione di rifornimento	/
Assegna	Veicolo - Trasferta	Rappresenta un veicolo assegnato per una trasferta	/
È eseguita	Veicolo - Manutenzione	L'operazione di manutenzione che è eseguita su un veicolo	/
Rifornisce	Veicolo - Rifornimento	Rappresenta l'operazione di rifornire un veicolo presso un luogo di rifornimento	/
Raggiunge	Trasferta - Luogo	Rappresenta un luogo da raggiungere in una trasferta	/
È eseguita	Rifornimento - Luogo	Associa un'operazione di rifornimento che è eseguita presso un luogo	/

Progettazione logica

Specifiche sulle operazioni

Si prevede che nel database vengano effettuate le seguenti operazioni:

Operazione	Descrizione	Frequenza
O1	Inserimento di un nuovo dipendente	1 al mese

Operazione	Descrizione	Frequenza
O2	Inserimento di un nuovo veicolo	1 al mese
O3	Inserimento di un nuovo luogo	2 al mese
O4	Programmazione di un'operazione di manutenzione	5 al mese
O5	Aggiunta di un'operazione di rifornimento	3 al giorno
O6	Programmazione di una trasferta	3 al giorno
O7	Ricerca dei veicoli che hanno subito più di 3 guasti durante l'anno	1 al mese
O8	Ricerca dei dipendenti che hanno causato più sinistri durante l'anno	1 all'anno
O9	Aggiornamento anagrafica dipendente	3 all'anno
O10	Ricerca dei veicoli a cui scadrà il bollo durante il mese corrente	1 al mese
O11	Ricerca dei veicoli a cui scadrà la revisione durante il mese corrente	1 al mese
O12	Ricerca dei veicoli a cui scadrà l'assicurazione durante il mese corrente	1 al mese
O13	Calcolo dei consumi di carburante per ciascun veicolo durante i 30 giorni precedenti	2 al mese
O14	Ricerca dei dipendenti che hanno impiegato oltre il 20% della stima dei KM in una trasferta durante i 30 giorni precedenti	1 al mese

Volume dei dati

Di seguito viene riportata una stima del database dopo un anno di utilizzo:

Concetto	Tipo	Volume
Dipendente	Entità	40
Veicolo	Entità	20
Trasferta	Entità	1095
Luogo	Entità	50
Manutenzione	Entità	60
Rifornimento	Entità	1100

Volume delle frequenze

Operazione	Descrizione	Frequenza	Tipo
O1	Inserimento di un nuovo dipendente	1 al mese	Interattiva
O2	Inserimento di un nuovo veicolo	1 al mese	Interattiva
O3	Inserimento di un nuovo luogo	2 al mese	Interattiva
O4	Programmazione di un'operazione di manutenzione	5 al mese	Interattiva
O5	Aggiunta di un'operazione di rifornimento	3 al giorno	Interattiva
O6	Programmazione di una trasferta	3 al giorno	Interattiva
O7	Ricerca dei veicoli che hanno subito più di 3 guasti durante l'anno	1 al mese	Batch
O8	Ricerca dei dipendenti che hanno causato più sinistri durante l'anno	1 all'anno	Batch

Operazione	Descrizione	Frequenza	Tipo
O9	Aggiornamento anagrafica dipendente	3 all'anno	Interattiva
O10	Ricerca dei veicoli a cui scadrà il bollo durante il mese corrente	1 al mese	Batch
O11	Ricerca dei veicoli a cui scadrà la revisione durante il mese corrente	1 al mese	Batch
O12	Ricerca dei veicoli a cui scadrà l'assicurazione durante il mese corrente	1 al mese	Batch
O13	Calcolo dei consumi di carburante per ciascun veicolo durante i 30 giorni precedenti	2 al mese	Batch
O14	Ricerca dei dipendenti che hanno impiegato oltre il 20% della stima dei KM in una trasferta durante i 30 giorni precedenti	1 al mese	Batch

Analisi delle ridondanze

All'interno del database non sono presenti ridondanze. Tuttavia si potrebbe valutare l'inserimento di un attributo che indichi il numero di guasti riscontrati in un veicolo nell'arco di un anno, utile per l'operazione **O7**, e un attributo che conti il numero di sinistri effettuati da un dipendente in un anno, utile per l'operazione **O8**; ambedue strettamente legate anche all'operazione **O4**.

Per le seguenti analisi assumeremo i volumi previsti dopo un anno di utilizzo.

Analisi O4-O7

Al momento della programmazione di un'operazione di manutenzione (**O4**) è necessario impostare il tipo di operazione da effettuare.

Nel caso di ridondanza, dobbiamo ciascuna volta inserire il nuovo record e poi incrementare di un'unità l'eventuale attributo che conta il numero di guasti di un veicolo, se trattasi di guasto. Nel caso in cui non vi sia ridondanza avremmo invece il mero inserimento del record.

La ricerca dei veicoli che hanno subito più guasti durante l'anno (**O7**) invece mantiene i medesimi costi, poiché comunque dovremo controllare tutto il parco veicoli.

In sintesi:

Operazione	Con ridondanza	Senza ridondanza
O4	$(1L + 2S) * 5/\text{mese} = 25 L/\text{mese}$	$1S * 5/\text{mese} = 10 L/\text{mese}$
O7	$20 \text{ veicoli} * 1L * 1/\text{mese} = 20 L$	$20 \text{ veicoli} * 1L * 1/\text{mese} = 20 L/\text{mese}$
Totale	45 L/mese	30 L/mese

Dato che senza ridondanza eseguiamo un terzo in meno delle letture rispetto al caso in cui sia presente la ridondanza, si predilige la sua assenza.

Analisi O4-O8

Al momento della programmazione di un'operazione di manutenzione (**O4**) è necessario impostare il tipo di operazione da effettuare.

Nel caso di ridondanza, dobbiamo ciascuna volta inserire il nuovo record e poi incrementare di un'unità l'eventuale attributo che conta il numero di sinistri affrontati da un dipendente. Nel caso in cui non vi sia ridondanza avremmo invece il mero inserimento del record.

La ricerca dei dipendenti che hanno causato più sinistri durante l'anno (**O8**) invece mantiene i medesimi costi, poiché basterebbe controllare la tabella Manutenzione.

In sintesi:

Operazione	Con ridondanza	Senza ridondanza
O4	$(1L + 2S) * 5/\text{mese} = 25 \text{ L/mese}$	$1S * 5/\text{mese} = 10 \text{ L/mese}$
O8	$20 \text{ veicoli} * 1L * 1/\text{mese} = 20 \text{ L}$	$20 \text{ veicoli} * 1L * 1/\text{mese} = 20 \text{ L/mese}$
Totale	45 L/mese	30 L/mese

Dato che senza ridondanza eseguiamo un terzo in meno delle letture rispetto al caso in cui sia presente la ridondanza, si predilige la sua assenza.

Modello logico

Di seguito viene riportata la legenda per la corretta interpretazione dello schema: **Entità**, chiave primaria, chiave esterna

Dipendente (Matricola, Nome, Cognome, Indirizzo, RecapitoTelefonico, Patenti)

Veicolo (Telaio, Marca, Modello, Targa, KMPercorsi, PotenzaMotore, TipologiaMotore, CategoriaVeicolo, ScadenzaBollo, ScadenzaRevisione, ScadenzaAssicurazione)

Luogo (IDLuogo, Nome, Indirizzo, StimaKM)

Trasferta (IDTrasferta, Veicolo, Autista, KMPercorsi, Stato, Destinazione, Inizio, Fine)

Manutenzione (Veicolo, TipoManutenzione, Descrizione, Inizio, Fine, KM, Autista, Costo)

Rifornimento (IDTransazione, Veicolo, Autista, DataOra, Localita, Importo, LitriCarburante)

Progettazione tabelle in SQL

Entità Dipendente

```
CREATE TABLE Dipendente (
    Matricola      INT UNSIGNED NOT NULL AUTO_INCREMENT,
    Nome           VARCHAR(30) NOT NULL,
    Cognome        VARCHAR(30) NOT NULL,
    Indirizzo      VARCHAR(50),
    RecapitoTelefonico VARCHAR(10),
    Patenti         SET('AM', 'A1', 'A2', 'A', 'B', 'C') DEFAULT NULL,
    PRIMARY KEY(Matricola)
);
```

Entità Veicolo

```
CREATE TABLE Veicolo (
    Telaio          VARCHAR(30) NOT NULL,
    Marca           VARCHAR(15) NOT NULL,
    Modello         VARCHAR(20) NOT NULL,
    Targa           VARCHAR(7) NOT NULL,
    KMPercorsi     DECIMAL(10, 2) UNSIGNED NOT NULL DEFAULT 0,
    PotenzaMotore   INT UNSIGNED NOT NULL DEFAULT 1000,
    TipologiaMotore ENUM('Endotermico', 'Elettrico') NOT NULL DEFAULT 'Endotermico',
    CategoriaVeicolo ENUM('Ciclomotore', 'Motociclo', 'Automobile', 'Camion') NOT NULL,
    ScadenzaBollo    DATE,
    ScadenzaRevisione DATE,
    ScadenzaAssicurazione DATE NOT NULL,
    PRIMARY KEY(Telaio)
);
```

Entità Luogo

```
CREATE TABLE Luogo (
    IDLuogo      INT UNSIGNED NOT NULL AUTO_INCREMENT,
    Nome          VARCHAR(20),
    Indirizzo     VARCHAR(50),
    StimaKM       DECIMAL(10, 2) UNSIGNED NOT NULL,
    PRIMARY KEY(IDLuogo)
);
```

Entità Trasferta

```
CREATE TABLE Trasferta (
    IDTrasferta INT UNSIGNED NOT NULL AUTO_INCREMENT,
    Veicolo VARCHAR(30) NOT NULL,
    Autista INT UNSIGNED NOT NULL,
    KMPercorsi DECIMAL(10, 2) UNSIGNED,
    Stato ENUM('Conclusa', 'Non conclusa') DEFAULT 'Non conclusa',
    Destinazione INT UNSIGNED NOT NULL,
    Inizio DATETIME NOT NULL,
    Fine DATETIME,
    PRIMARY KEY(IDTrasferta),
    FOREIGN KEY(Veicolo) REFERENCES Veicolo(Telaio),
    FOREIGN KEY(Autista) REFERENCES Dipendente(Matricola),
    FOREIGN KEY(Destinazione) REFERENCES Luogo(IDLuogo),
    CONSTRAINT controllo_date_trasferta CHECK (Fine IS NULL OR Fine > Inizio),
    CONSTRAINT controllo_conclusa CHECK ((Stato = 'Non conclusa' AND Fine IS NULL) OR (Stato = 'Conclusa' AND Fine IS NOT NULL))
);
```

Entità Manutenzione

```
CREATE TABLE Manutenzione (
    Veicolo VARCHAR(30) NOT NULL,
    TipoManutenzione ENUM('Guasto', 'Tagliando', 'Revisione', 'Sinistro') NOT NULL,
    Descrizione VARCHAR(200),
    Inizio DATETIME NOT NULL,
    Fine DATETIME,
    KM DECIMAL(10, 2) UNSIGNED NOT NULL,
    Autista INT UNSIGNED DEFAULT NULL,
    Costo DECIMAL(10, 2) NOT NULL,
    PRIMARY KEY(Veicolo, Inizio),
    FOREIGN KEY(Autista) REFERENCES Dipendente(Matricola),
    FOREIGN KEY(Veicolo) REFERENCES Veicolo(Telaio),
    CONSTRAINT controllo_date_manutenzione CHECK (Fine IS NULL OR Fine > Inizio),
    CONSTRAINT controllo_inserimento CHECK ((TipoManutenzione => 'Sinistro' AND Autista IS NULL) OR (TipoManutenzione = 'Sinistro' AND Autista IS NOT NULL))
);
```

Entità Rifornimento

```
CREATE TABLE Rifornimento (
    IDTransazione INT UNSIGNED NOT NULL,
    Veicolo VARCHAR(30) NOT NULL,
    Autista INT UNSIGNED NOT NULL,
    DataOra DATETIME NOT NULL,
    Localita INT UNSIGNED NOT NULL,
    Importo DECIMAL(6, 3) UNSIGNED NOT NULL,
    LitriCarburante DECIMAL(5, 2) UNSIGNED NOT NULL,
    PRIMARY KEY(IDTransazione),
    FOREIGN KEY(Veicolo) REFERENCES Veicolo(Telaio),
    FOREIGN KEY(Autista) REFERENCES Dipendente(Matricola),
    FOREIGN KEY(Localita) REFERENCES Luogo(IDLuogo)
);
```

Inserimento dati in SQL

Di seguito vengono riportati degli esempi per l'inserimento di ciascuna operazione.

O1 - Inserimento di un nuovo dipendente

```
● ● ●

INSERT INTO Dipendente (Nome, Cognome, Indirizzo, RecapitoTelefonico, Patenti) VALUES
('Asimov', 'Isaac', 'Piazza dei Cicli, 3', '0987612345', 'A1,B');
```

O2 - Inserimento di un nuovo veicolo

```
● ● ●

INSERT INTO Veicolo (Telaio, Marca, Modello, Targa, KMPercorsi, PotenzaMotore, TipologiaMotore, CategoriaVeicolo, ScadenzaBollo,
ScadenzaRevisione, ScadenzaAssicurazione) VALUES
('TX1234ABCD', 'Citroen', 'C1', 'AB123CD', 5000.00, 53, 'Endotermico', 'Automobile', '2024-12-31', '2024-07-30', '2024-07-30'),
('TX5678EFGH', 'Tesla', 'Model S', 'EF456GH', 12000.50, 760, 'Elettrico', 'Automobile', '2024-11-30', '2025-07-31', '2024-09-01');
```

O3 - Inserimento di un nuovo luogo

```
● ● ●

INSERT INTO Luogo (Nome, Indirizzo, StimaKM) VALUES (
    'Terminus', 'Viale della Fondazioni, 1', 75.50
);
```

O4 - Programmazione di un'operazione di manutenzione

```
● ● ●

INSERT INTO Manutenzione (Veicolo, TipoManutenzione, Descrizione, Inizio, Fine, KM, Autista, Costo) VALUES
('TX1234ABCD', 'Guasto', 'Sostituzione batteria', '2024-06-01 08:00:00', '2024-06-01 12:00:00', 5300.53, NULL, 250.00),
('TX1213MNOP', 'Sinistro', 'Incidente lieve', '2024-06-04 11:00:00', '2024-06-04 18:00:00', 6150.50, 4, 300.00);
```

O5 - Aggiunta di un'operazione di rifornimento

```
● ● ●

INSERT INTO Rifornimento (IDTransazione, Veicolo, Autista, DataOra, Localita, Importo, LitriCarburante) VALUES (
    1, 'TX1234ABCD', 1, '2024-07-01 08:00:00', 10, 50.000, 40.50
);
```

O6 - Programmazione di una trasferta

```
INSERT INTO Trasferta (Veicolo, Autista, KMPercorsi, Stato, Destinazione, Inizio, Fine) VALUES (
    'TX1234ABCD', 2, 450.53, 'Conclusa', 1, '2024-07-10 08:00:00', '2024-07-10 18:00:00'
);
```

Query in SQL

O7 - Ricerca dei veicoli che hanno subito più di 3 guasti

```
SELECT Telaio, Marca, Modello, COUNT(*) 'Numero guasti'
FROM Veicolo v, Manutenzione m
WHERE v.Telaio = m.Veicolo AND m.TipoManutenzione = 'Guasto'
GROUP BY Telaio
HAVING `Numero guasti` > 3;
```

O8 - Ricerca dei dipendenti che hanno causato più sinistri

```
SELECT Matricola, Nome, Cognome, Sin 'Numero sinistri'
FROM (
    SELECT d.Matricola, d.Nome, d.Cognome, COUNT(*) 'Sin'
    FROM Dipendente d, Manutenzione m
    WHERE d.Matricola = m.Autista AND m.TipoManutenzione = 'Sinistro' AND m.Inizio BETWEEN DATE_SUB(CURRENT_DATE, INTERVAL 1 YEAR) AND CURRENT_DATE
    GROUP BY d.Matricola, d.Nome, d.Cognome
) Sinistri,
(
    SELECT COUNT(*) 'Si'
    FROM Dipendente d, Manutenzione m
    WHERE d.Matricola = m.Autista AND m.TipoManutenzione = 'Sinistro' AND m.Inizio BETWEEN DATE_SUB(CURRENT_DATE, INTERVAL 1 YEAR) AND CURRENT_DATE
    GROUP BY d.Matricola
    ORDER BY Si DESC
    LIMIT 1
) Si
WHERE Sinistri.Sin = Si.Si;
```

O9 - Aggiornamento anagrafica dipendente

```
● ● ●  
UPDATE Dipendente SET  
Indirizzo = 'Viale Andrea Doria, 6',  
RecapitoTelefonico = '1123456789',  
Patenti = 'A,B'  
WHERE Matricola = 1;
```

O10 - Ricerca dei veicoli a cui scadrà il bollo durante il mese corrente

```
● ● ●  
SELECT Telaio, Marca, Modello  
FROM Veicolo  
WHERE YEAR(ScadenzaBollo) = YEAR(CURRENT_DATE) AND  
MONTH(ScadenzaBollo) = MONTH(CURRENT_DATE);
```

O11 - Ricerca dei veicoli a cui scadrà la revisione durante il mese corrente

```
● ● ●  
SELECT Telaio, Marca, Modello  
FROM Veicolo  
WHERE YEAR(ScadenzaRevisione) = YEAR(CURRENT_DATE) AND  
MONTH(ScadenzaRevisione) = MONTH(CURRENT_DATE);
```

O12 - Ricerca dei veicoli a cui scadrà l'assicurazione durante il mese corrente

```
SELECT Telaio, Marca, Modello
FROM Veicolo
WHERE YEAR(ScadenzaAssicurazione) = YEAR(CURRENT_DATE) AND
MONTH(ScadenzaAssicurazione) = MONTH(CURRENT_DATE);
```

O13 - Calcolo dei consumi di carburante per ciascun veicolo durante i 30 giorni precedenti

```
SELECT *, `Litri riforniti` / `KM percorsi` 'Stima KM/L'
FROM (
    SELECT v.Telaio, v.Marca, v.Modello, IFNULL(SUM(r.Importo), 0) 'Costo carburante',
           IFNULL(SUM(r.LitriCarburante), 0) 'Litri riforniti', COUNT(DISTINCT t.IDTrasferta) 'Trasferte effettuate',
           SUM(t.KMPercorsi) 'KM percorsi'
    FROM Veicolo v
    LEFT JOIN Rifornimento r ON v.Telaio = r.Veicolo AND
        r.DataOra BETWEEN DATE_SUB(CURRENT_DATE, INTERVAL 31 DAY) AND CURRENT_DATE
    LEFT JOIN Trasferta t ON v.Telaio = t.Veicolo AND t.Inizio >= DATE_SUB(CURRENT_DATE, INTERVAL 31 DAY)
        AND t.Fine <= CURRENT_DATE
    WHERE v.TipologiaMotore = 'Endotermico'
    GROUP BY v.Telaio
) AnalisiConsumi
WHERE `KM percorsi` IS NOT NULL;
```

O14 - Ricerca dei dipendenti che hanno impiegato oltre il 20% della stima dei KM in una trasferta durante i 30 giorni precedenti

```
SELECT Matricola, d.Nome, d.Cognome, t.IDTrasferta 'Trasferta', l.StimaKM*2 'Stima KM trasferta A/R', KMPercorsi 'KM effettivi trasferta A/R'
FROM Dipendente d, Trasferta t, Luogo l
WHERE d.Matricola = t.Autista AND t.Stato = 'Conclusa'
    AND l.IDLuogo = t.Destinazione AND l.StimaKM*2 + l.StimaKM*2*0.2 < KMPercorsi
    AND t.Inizio >= DATE_SUB(CURRENT_DATE, INTERVAL 31 DAY) AND t.Fine <= DATE_SUB(CURRENT_DATE, INTERVAL 1 DAY)
GROUP BY Matricola, StimaKM, KMPercorsi, t.IDTrasferta;
```

Eventi e Trigger

Di seguito vengono illustrati gli eventi e i trigger presenti.

Spesso verrà usato come delimitatore finale la combinazione di caratteri \$\$.

Viene omessa l'istruzione di cambio delimitatore nella rappresentazione di ciascuna query per motivi di brevità.

E1 - Aggiornamento data bollo ogni giorno della scadenza

Ammettendo l'esistenza di uno scheduling automatico dei pagamenti del bollo al giorno della scadenza, viene definito l'evento E1 che si occupa di aggiornare la data di scadenza del bollo a 12 mesi dalla data odierna, nel giorno della scadenza.

```
CREATE EVENT AggiornamentoScadenzaBollo  
ON SCHEDULE EVERY 1 DAY DO  
UPDATE Veicolo  
SET ScadenzaBollo = DATE_ADD(CURRENT_DATE, INTERVAL 12 MONTH)  
WHERE ScadenzaBollo = CURRENT_DATE;
```

E2 - Rinnovo assicurazione

Con la medesima premessa dell'evento precedente, si implementa anche uno schedule per il rinnovo automatico il giorno della scadenza.

```
CREATE EVENT AggiornamentoScadenzaAssicurazione  
ON SCHEDULE EVERY 1 DAY DO  
UPDATE Veicolo  
SET ScadenzaAssicurazione = DATE_ADD(CURRENT_DATE, INTERVAL 12 MONTH)  
WHERE ScadenzaAssicurazione = CURRENT_DATE;
```

E3 - Prenotazione revisione una settimana prima della scadenza

La prenotazione avviene attraverso un semplice evento che però richiama una ben più complessa stored procedure.

```
CREATE EVENT PrenotazioneRevisione  
ON SCHEDULE EVERY 1 DAY DO  
CALL PrenotazioneRevisione;
```

```

CREATE PROCEDURE PrenotazioneRevisione()
BEGIN
    DECLARE fine_veicoli BOOLEAN DEFAULT FALSE;
    DECLARE tel VARCHAR(30);
    DECLARE km_percorsi DECIMAL(10, 2) UNSIGNED;
    DECLARE domani DATE DEFAULT DATE_ADD(NOW(), INTERVAL 1 DAY);
    DECLARE settimana_prossima DATE DEFAULT DATE_ADD(CURRENT_DATE, INTERVAL 7 DAY);
    DECLARE revisione_ln_scadenza CURSOR FOR
        SELECT Telaio
        FROM Veicolo LEFT JOIN Manutenzione ON Telaio = Veicolo
        WHERE ScadenzaRevisione = settimana_prossima AND
            (TipoManutenzione IS NULL OR NOT (TipoManutenzione = 'Revisione' AND Inizio NOT BETWEEN CURRENT_DATE AND settimana_prossima));
    DECLARE km_revisione_ln_scadenza CURSOR FOR
        SELECT KMPercorsi
        FROM Veicolo LEFT JOIN Manutenzione ON Telaio = Veicolo
        WHERE ScadenzaRevisione = settimana_prossima AND
            (TipoManutenzione IS NULL OR NOT (TipoManutenzione = 'Revisione' AND Inizio NOT BETWEEN CURRENT_DATE AND settimana_prossima));
    DECLARE CONTINUE HANDLER FOR
        NOT FOUND SET fine_veicoli = TRUE;

    OPEN revisione_ln_scadenza;
    OPEN km_revisione_ln_scadenza;

    read_loop: LOOP
        FETCH revisione_ln_scadenza INTO tel;
        FETCH km_revisione_ln_scadenza INTO km_percorsi;

        IF fine_veicoli THEN
            LEAVE read_loop;
        END IF;

        INSERT INTO Manutenzione (Veicolo, TipoManutenzione, Descrizione, Inizio, KM, Costo)
        VALUES (tel, 'Revisione', 'Revisione periodica', domani, km_percorsi, 80);
    END LOOP;

    CLOSE revisione_ln_scadenza;
    CLOSE km_revisione_ln_scadenza;
END$$

```

T1 - Inizializzazione scadenza revisione e bollo

Al momento dell'inserimento o aggiornamento nella tabella Veicolo, è possibile indicare la scadenza del bollo e della revisione successiva, mentre è necessario inserire quello della scadenza dell'assicurazione. Nel caso in cui non venga fornito nessun valore di default, il trigger T1 si occuperà di impostare una data di default a dodici mesi dalla data in cui viene aggiunto per la scadenza del bollo successivo e di quattro anni per la revisione successiva nel caso di inserimento, mentre controlla che i valori non vengano valorizzati a NULL nel caso di update. Infine, oltre a verificare che i valori appena inseriti non siano passati, esegue il medesimo controllo anche per il campo ScadenzaAssicurazione.

```

CREATE TRIGGER AggiuntaNuovoVeicolo
BEFORE INSERT ON Veicolo
FOR EACH ROW
BEGIN
    IF NEW.ScadenzaBollo IS NULL THEN
        SET NEW.ScadenzaBollo = DATE_ADD(CURRENT_DATE, INTERVAL 12 MONTH);
    ELSEIF NEW.ScadenzaBollo < CURRENT_DATE THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Il bollo non può essere già scaduto!';
    END IF;

    IF NEW.ScadenzaRevisione IS NULL THEN
        SET NEW.ScadenzaRevisione = DATE_ADD(CURRENT_DATE, INTERVAL 4 YEAR);
    ELSEIF NEW.ScadenzaRevisione < CURRENT_DATE THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'La revisione non può essere già scaduta!';
    END IF;

    IF NEW.ScadenzaAssicurazione < CURRENT_DATE THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "L'assicurazione non può essere già scaduta!";
    END IF;
END$$

CREATE TRIGGER AggiuntaNuovoVeicoloUpd
BEFORE UPDATE ON Veicolo
FOR EACH ROW
BEGIN
    IF NEW.ScadenzaBollo IS NULL THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Il bollo deve essere valorizzato!';
    ELSEIF NEW.ScadenzaBollo < CURRENT_DATE THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Il bollo non può essere già scaduto!';
    END IF;

    IF NEW.ScadenzaRevisione IS NULL THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'La revisione deve essere valorizzata!';
    ELSEIF NEW.ScadenzaRevisione < CURRENT_DATE THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'La revisione non può essere già scaduta!';
    END IF;

    IF NEW.ScadenzaAssicurazione < CURRENT_DATE THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "L'assicurazione non può essere già scaduta!";
    END IF;
END$$

```

T2 - Aggiornamento KM totali veicolo dopo ogni trasferta

Dopo l'inserimento o l'aggiornamento di una trasferta conclusa, viene aggiornato il valore dei KM totali percorsi dal veicolo: ciò avviene per mezzo di due trigger che controllano lo stato di una trasferta ed eventualmente aggiornano il dato.

```

CREATE TRIGGER AggiornamentoKMPPostTrasferta
AFTER UPDATE ON Trasferta
FOR EACH ROW
BEGIN
    IF NEW.Stato = 'Conclusa' THEN
        UPDATE Veicolo
        SET Veicolo.KMPercorsi = Veicolo.KMPercorsi + NEW.KMPercorsi
        WHERE Telaio = NEW.Veicolo;
    END IF;
END$$

CREATE TRIGGER AggiornamentoKMTrasfertaPassata
AFTER INSERT ON Trasferta
FOR EACH ROW
BEGIN
    IF NEW.Stato = 'Conclusa' THEN
        UPDATE Veicolo
        SET Veicolo.KMPercorsi = Veicolo.KMPercorsi + NEW.KMPercorsi
        WHERE Telaio = NEW.Veicolo;
    END IF;
END$$

```

T3 - Aggiornamento KM totali veicolo dopo ogni manutenzione

Dopo ciascuna operazione di manutenzione, vengono aggiornati i KM totali del veicolo, al fine di mantenere una maggiore granularità dei KM percorsi dal veicolo e maggiore integrità tra i dati conservati.

```

CREATE TRIGGER AggiornamentoKMPPostManutenzione
AFTER INSERT ON Manutenzione
FOR EACH ROW
BEGIN
    DECLARE km_veicolo DECIMAL(10, 2) UNSIGNED;
    DECLARE trasferta_recente DATETIME;

    SELECT KMPercorsi INTO km_veicolo
    FROM Veicolo
    WHERE Telaio = NEW.Veicolo;

    SELECT Fine INTO trasferta_recente
    FROM Trasferta
    WHERE Veicolo = NEW.Veicolo
    ORDER BY Fine DESC
    LIMIT 1;

    IF NEW.Fine IS NOT NULL THEN
        IF NEW.KM >= km_veicolo THEN
            UPDATE Veicolo
            SET Veicolo.KMPercorsi = NEW.KM
            WHERE Telaio = NEW.Veicolo;
        ELSEIF NEW.Fine > IFNULL(trasferta_recente, '1970-01-01 08:00:00') THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Il veicolo non può avere meno km!';
        END IF;
    END IF;
END$$

```

T4 - Aggiornamento scadenza revisione

Alla fine di un'operazione di manutenzione categorizzata come 'Revisione', viene aggiornata la scadenza della revisione del veicolo a due anni dalla data di completamento.

```
CREATE TRIGGER AggiornamentoScadenzaRevisione
AFTER INSERT ON Manutenzione
FOR EACH ROW
BEGIN
    IF NEW.TipoManutenzione = 'Revisione' AND NEW.Fine IS NOT NULL THEN
        UPDATE Veicolo
        SET Veicolo.ScadenzaRevisione = DATE_ADD(NEW.Fine, INTERVAL 2 YEAR)
        WHERE Telaio = NEW.Veicolo;
    END IF;
END$$

CREATE TRIGGER AggiornamentoScadenzaRevisioneUpd
AFTER UPDATE ON Manutenzione
FOR EACH ROW
BEGIN
    IF NEW.TipoManutenzione = 'Revisione' AND NEW.Fine IS NOT NULL THEN
        UPDATE Veicolo
        SET Veicolo.ScadenzaRevisione = DATE_ADD(NEW.Fine, INTERVAL 2 YEAR)
        WHERE Telaio = NEW.Veicolo;
    END IF;
END$$
```

T5 - Prenotazione tagliando il giorno successivo al superamento dei KM
Dopo ogni aggiornamento nella tabella Veicolo, viene controllato se è necessario eseguire il tagliando al veicolo.

```
CREATE TRIGGER ProgrammazioneTagliando
AFTER UPDATE ON Veicolo
FOR EACH ROW
BEGIN
    DECLARE km_veicolo DECIMAL(10, 2) UNSIGNED;
    DECLARE is_scheduled BOOLEAN;

    SELECT IFNULL(MAX(KM), 0) INTO km_veicolo
    FROM Manutenzione m
    WHERE m.Veicolo = NEW.Telaio;

    SELECT CASE WHEN COUNT(Veicolo) > 0 THEN 1 ELSE 0 END INTO is_scheduled
    FROM Manutenzione
    WHERE Veicolo = NEW.Telaio AND TipoManutenzione = 'Tagliando' AND Fine IS NULL;

    IF NEW.KMPercorsi > km_veicolo + 10000 AND NOT is_scheduled THEN
        INSERT INTO Manutenzione (Veicolo, TipoManutenzione, Descrizione, Inizio, KM, Costo) VALUES
        (NEW.Telaio, 'Tagliando', 'Prenotazione tagliando temporanea', DATE_ADD(CURRENT_DATE, INTERVAL 1 DAY),
        NEW.KMPercorsi, 0);
    END IF;
END$$
```

T6 - Selezione veicolo adeguato

Per il requisito aziendale di favorire un veicolo endotermico per tratte superiori ai 100KM, si implementa il seguente trigger che automaticamente

modifica il veicolo, in caso sia possibile.

Per le tratte più lunghe, viene automaticamente segnalato l'errore.

```
CREATE TRIGGER PrediligeVeicoloAdeguato
BEFORE INSERT ON Trasferta
FOR EACH ROW
BEGIN
    DECLARE stima_km INT;
    DECLARE tipologia_motore VARCHAR(11);
    DECLARE veicolo_alternativo VARCHAR(30);

    SELECT StimaKM INTO stima_km
    FROM Luogo
    WHERE IDLuogo = NEW.Destinazione;

    SELECT TipologiaMotore INTO tipologia_motore
    FROM Veicolo
    WHERE Telaio = NEW.Veicolo;

    IF stima_km <= 100 AND tipologia_motore = 'Endotermico' THEN
        SELECT Telaio INTO veicolo_alternativo
        FROM Veicolo
        WHERE TipologiaMotore = 'Elettrico' AND Telaio NOT IN (
            SELECT Veicolo
            FROM Trasferta t LEFT JOIN Manutenzione m ON t.Veicolo = m.Veicolo
            WHERE (NEW.Inizio BETWEEN t.Inizio AND IFNULL(t.Fine, NOW()))
            OR (NEW.Fine BETWEEN t.Inizio AND IFNULL(t.Fine, NOW()))
            OR (t.Inizio BETWEEN NEW.Inizio AND NEW.Fine)
            OR (NEW.Inizio BETWEEN m.Inizio AND IFNULL(m.Fine, NOW()))
            OR (NEW.Fine BETWEEN m.Inizio AND IFNULL(m.Fine, NOW()))
            OR (m.Inizio BETWEEN NEW.Inizio AND NEW.Fine)
        ) LIMIT 1;

        IF veicolo_alternativo IS NOT NULL THEN
            SET NEW.Veicolo = veicolo_alternativo;
        END IF;

    ELSEIF stima_km > 100 AND tipologia_motore = 'Elettrico' THEN
        SELECT Telaio INTO veicolo_alternativo
        FROM Veicolo
        WHERE TipologiaMotore = 'Endotermico' AND Telaio NOT IN (
            SELECT Veicolo
            FROM Trasferta t LEFT JOIN Manutenzione m ON t.Veicolo = m.Veicolo
            WHERE (NEW.Inizio BETWEEN t.Inizio AND IFNULL(t.Fine, NOW()))
            OR (NEW.Fine BETWEEN t.Inizio AND IFNULL(t.Fine, NOW()))
            OR (t.Inizio BETWEEN NEW.Inizio AND NEW.Fine)
            OR (NEW.Inizio BETWEEN m.Inizio AND IFNULL(m.Fine, NOW()))
            OR (NEW.Fine BETWEEN m.Inizio AND IFNULL(m.Fine, NOW()))
            OR (m.Inizio BETWEEN NEW.Inizio AND NEW.Fine)
        ) LIMIT 1;

        IF veicolo_alternativo IS NOT NULL THEN
            SET NEW.Veicolo = veicolo_alternativo;
        ELSE
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Non ci sono veicoli endotermici disponibili';
        END IF;
    END IF;
END$$
```

T7 - Controllo che il veicolo aggiunto in trasferta sia disponibile

Prima di poter inserire una nuova trasferta o modificarla, viene controllato se il veicolo sia disponibile in quel range di date; in caso non sia disponibile, viene impedito l'inserimento e fornito un messaggio di errore.

```
CREATE TRIGGER VerificaVeicoloDisponibile
BEFORE INSERT ON Trasferta
FOR EACH ROW
BEGIN
    DECLARE veicolo_occupato BOOLEAN;

    SELECT CASE WHEN COUNT(*) > 0 THEN 1 ELSE 0 END INTO veicolo_occupato
    FROM Trasferta t LEFT JOIN Manutenzione m ON t.Veicolo = NEW.Veicolo AND m.Veicolo = t.Veicolo
    WHERE NEW.Inizio < IFNULL(t.Fine, NOW()) AND IFNULL(NEW.Fine, NOW()) > t.Inizio
        AND NEW.Inizio < IFNULL(m.Fine, NOW()) AND IFNULL(NEW.Fine, NOW()) > m.Inizio
        AND t.IDTrasferta <> NEW.IDTrasferta;

    IF veicolo_occupato THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Il veicolo è già occupato in questo periodo.';
    END IF;
END$$

CREATE TRIGGER VerificaVeicoloDisponibileUpd
BEFORE UPDATE ON Trasferta
FOR EACH ROW
BEGIN
    DECLARE veicolo_occupato BOOLEAN;

    SELECT CASE WHEN COUNT(*) > 0 THEN 1 ELSE 0 END INTO veicolo_occupato
    FROM Trasferta t LEFT JOIN Manutenzione m ON t.Veicolo = NEW.Veicolo AND m.Veicolo = t.Veicolo
    WHERE NEW.Inizio < IFNULL(t.Fine, NOW()) AND IFNULL(NEW.Fine, NOW()) > t.Inizio
        AND NEW.Inizio < IFNULL(m.Fine, NOW()) AND IFNULL(NEW.Fine, NOW()) > m.Inizio
        AND t.IDTrasferta <> NEW.IDTrasferta;

    IF veicolo_occupato THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Il veicolo è già occupato in questo periodo.';
    END IF;
END$$
```

T8 - Controllo che l'autista aggiunto in trasferta sia disponibile

Prima di poter inserire una nuova trasferta o modificarla, viene controllato se l'autista sia disponibile in quel range di date; in caso non sia disponibile, viene impedito l'inserimento e fornito un messaggio di errore.

```
CREATE TRIGGER VerificaAutistaDisponibile
BEFORE INSERT ON Trasferta
FOR EACH ROW
BEGIN
    DECLARE autista_occupato BOOLEAN;

    SELECT CASE WHEN COUNT(*) > 0 THEN 1 ELSE 0 END INTO autista_occupato
    FROM Trasferta
    WHERE Autista = NEW.Autista AND NEW.Inizio < IFNULL(Fine, NOW()) AND IFNULL(NEW.Fine, NOW()) > Inizio
        AND IDTrasferta <> NEW.IDTrasferta;

    IF autista_occupato THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "L'autista è già occupato in questo periodo.";
    END IF;
END$$

CREATE TRIGGER VerificaAutistaDisponibileUpd
BEFORE UPDATE ON Trasferta
FOR EACH ROW
BEGIN
    DECLARE autista_occupato BOOLEAN;

    SELECT CASE WHEN COUNT(*) > 0 THEN 1 ELSE 0 END INTO autista_occupato
    FROM Trasferta
    WHERE Autista = NEW.Autista AND NEW.Inizio < IFNULL(Fine, NOW()) AND IFNULL(NEW.Fine, NOW()) > Inizio
        AND IDTrasferta <> NEW.IDTrasferta;

    IF autista_occupato THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "L'autista è già occupato in questo periodo.";
    END IF;
END$$
```

T9 - Controllo che venga effettuato il rifornimento su un veicolo endotermico

Poiché le operazioni di ricarica vengono effettuate esclusivamente in azienda, si decide di implementare un controllo tale per cui le auto elettriche non possano essere rifornite presso sistemi terzi, limitando quindi l'operazione ai soli veicoli endotermici.

```
CREATE TRIGGER VerificaRifornimento
BEFORE INSERT ON Rifornimento
FOR EACH ROW
BEGIN
    DECLARE tipologia_motore VARCHAR(30);

    SELECT TipologiaMotore INTO tipologia_motore
    FROM Veicolo
    WHERE Telaio = NEW.Veicolo;

    IF tipologia_motore = 'Elettrico' THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Non si può effettuare un'operazione di rifornimento su un veicolo elettrico!";
    END IF;
END$$

CREATE TRIGGER VerificaRifornimentoUpd
BEFORE UPDATE ON Rifornimento
FOR EACH ROW
BEGIN
    DECLARE tipologia_motore VARCHAR(30);

    SELECT TipologiaMotore INTO tipologia_motore
    FROM Veicolo
    WHERE Telaio = NEW.Veicolo;

    IF tipologia_motore = 'Elettrico' THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Non si può effettuare un'operazione di rifornimento su un veicolo elettrico!";
    END IF;
END$$
```

T10 - Controllo che l'autista possa guidare quel determinato veicolo
Ciascun autista può essere in possesso o meno di determinate patenti, perciò si ritiene necessario effettuare un controllo se il veicolo assegnato a un autista per una trasferta possa essere guidato dal dipendente.

```
CREATE TRIGGER VerificaPatente
BEFORE INSERT ON Trasferta
FOR EACH ROW
BEGIN
    DECLARE pat VARCHAR(5);
    DECLARE categoria_veicolo VARCHAR(20);
    DECLARE potenza INT UNSIGNED;

    SELECT Patenti INTO pat
    FROM Dipendente
    WHERE Matricola = NEW.Autista;

    SELECT CategoriaVeicolo, PotenzaMotore
    INTO categoria_veicolo, potenza
    FROM Veicolo
    WHERE Telaio = NEW.Veicolo;

    IF pat IS NULL THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Il dipendente non è in possesso di patenti.";
    ELSEIF (FIND_IN_SET('Ciclomotore', pat) OR
            FIND_IN_SET('A1', pat) OR
            FIND_IN_SET('A2', pat) OR
            FIND_IN_SET('A', pat) OR
            FIND_IN_SET('B', pat)) AND NOT (
            FIND_IN_SET('Motociclo', categoria_veicolo) AND potenza <= 11) AND NOT (
            FIND_IN_SET('A1', pat) OR
            FIND_IN_SET('A2', pat) OR
            FIND_IN_SET('A', pat) OR
            FIND_IN_SET('B', pat))
        ) OR (FIND_IN_SET('Motociclo', categoria_veicolo) AND potenza <= 11) AND NOT (
            FIND_IN_SET('A1', pat) OR
            FIND_IN_SET('A2', pat) OR
            FIND_IN_SET('A', pat) OR
            FIND_IN_SET('B', pat))
        ) OR (FIND_IN_SET('Motociclo', categoria_veicolo) AND potenza > 15 AND NOT FIND_IN_SET('A', pat))
        OR (FIND_IN_SET('Automobile', categoria_veicolo) AND NOT FIND_IN_SET('B', pat))
        OR (FIND_IN_SET('Camion', categoria_veicolo) AND NOT FIND_IN_SET('C', pat))
    THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "L'autista non ha la patente idonea a guidare il veicolo selezionato!";
    END IF;
END$$
```

Utenti e permessi

Al fine di garantire una corretta separazione dei ruoli all'interno dell'azienda, si sceglie di suddividere l'accesso alle tabelle in ruoli, in modo tale da evitare errori dettati da un incauto uso.

HR

Il reparto risorse umane ha il compito di assumere nuovi dipendenti e poter gestire le ricevute fiscali ottenute dai dipendenti in trasferta.

Pertanto si decide di affidare loro un utente con possibilità di inserimento e aggiornamento sulle tabelle **Dipendente, Rifornimento e Luogo**.

La necessità di inserire un nuovo luogo è data dal fatto che al momento della registrazione della ricevuta fiscale, va registrata anche l'ubicazione.

```
CREATE USER 'hr'@'localhost' IDENTIFIED BY 'ZTWGNmYr3SsQzMCjIIFBjalWgx9lu4';
GRANT INSERT, UPDATE ON Progetto_Blanco.Dipendente TO 'hr'@'localhost';
GRANT INSERT, UPDATE ON Progetto_Blanco.Luogo TO 'hr'@'localhost';
GRANT INSERT, UPDATE ON Progetto_Blanco.Rifornimento TO 'hr'@'localhost';
```

Amministrazione

Il reparto amministrazione ha la possibilità di acquistare e vendere (o rottamare) veicoli e di poter eseguire gestire il personale. Pertanto si decide di garantire permessi in inserimento, aggiornamento e cancellazione su **Dipendente e Veicolo**.

```
CREATE USER 'amministrazione'@'localhost' IDENTIFIED BY 'P8A2gl3i5hsJTLHiF80bkxzv80mCpS7';
GRANT INSERT, UPDATE, DELETE ON Progetto_Blanco.Dipendente TO 'amministrazione'@'localhost';
GRANT INSERT, UPDATE, DELETE ON Progetto_Blanco.Veicolo TO 'amministrazione'@'localhost';
```

Pianificazione

Il reparto pianificazione si occupa di programmare le trasferte dei dipendenti, pertanto avrà accesso in inserimento, aggiornamento e cancellazione su **Trasferte e Luogo**, e di programmare le operazioni di manutenzione

(ordinaria o straordinaria) su ciascun mezzo, pertanto avrà la possibilità di inserire e aggiornare la tabella **Manutenzione**.

```
CREATE USER 'pianificazione'@'localhost' IDENTIFIED BY 'VUPFg3j0awunuD0d9EQPz83mHKzwmag';
GRANT INSERT, UPDATE, DELETE ON Progetto_Blanco.Trasferta TO 'pianificazione'@'localhost';
GRANT INSERT, UPDATE, DELETE ON Progetto_Blanco.Luogo TO 'pianificazione'@'localhost';
GRANT INSERT, UPDATE ON Progetto_Blanco.Manutenzione TO 'pianificazione'@'localhost';
```

Tutti i privilegi non forniti agli utenti aziendali sono comunque disponibili per il DBA.

Progettazione fisica

Schema fisico

