|  |  |
| --- | --- |
|  |  |

Basi di Dati

Progetto A.A. 2023/2024

Sistema Informativo di azienda di Trasporto Pubblico

0334999

Andrea Sentina

**Indice**

[1. Descrizione del Minimondo 2](#_Toc161848079)

[2. Analisi dei Requisiti 4](#_Toc161848080)

[3. Progettazione concettuale 7](#_Toc161848081)

[4. Progettazione logica 12](#_Toc161848082)

[5. Progettazione fisica 19](#_Toc161848083)

# Descrizione del Minimondo

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | Si intende realizzare il sistema informativo di un’azienda di trasporto pubblico locale. L’azienda è dotata di un parco veicoli che permettono di coprire un determinato insieme di tratte. I veicoli sono caratterizzati da una matricola (codice univoco numerico di quattro cifre). Ogni veicolo è anche associato ad una data di acquisto. Ciascuna tratta ha un insieme di fermate identificate da latitudine e longitudine ed associata ad un codice numerico univoco di cinque cifre. La prima e l’ultima fermata sono contrassegnate come capilinea. Ciascuna tratta viene coperta da un numero predefinito di veicoli, la cui associazione viene gestita dai gestori del servizio. Ogni capolinea ha un orario di partenze prestabilito. Gli amministratori del servizio gestiscono anche i conducenti, caratterizzati da un codice fiscale, un nome, un cognome, una data di nascita ed un luogo di nascita. Di ogni conducente è di interesse conoscere anche il numero di patente e la data di scadenza della stessa. Gli utenti del sistema possono accedere al servizio per conoscere, dato il codice di una fermata, a quante fermate di distanza si trova un veicolo.  Quando salgono a bordo, gli utenti del servizio timbrano un biglietto elettronico o un abbonamento sul “validatore intelligente” installato sui veicoli. Nel caso di un biglietto elettronico, questo viene marcato come “utilizzato” all’interno del sistema. L’emissione di nuovi biglietti viene amministrata dai gestori del servizio. Nel caso di abbonamento, viene verificato se esso è ancora valido e viene memorizzata la data dell’ultimo utilizzo. Quando un autista si trova ad un capolinea, può interrogare il sistema per sapere qual è la prossima partenza prevista del veicolo che sta guidando. |

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | Si intende realizzare il sistema informativo di un’azienda di trasporto pubblico locale. L’azienda è dotata di un parco veicoli che permettono di coprire un determinato insieme di tratte. I veicoli sono caratterizzati da una matricola (codice univoco numerico di quattro cifre). Ogni veicolo è anche associato ad una data di acquisto. Ciascuna tratta ha un insieme di fermate identificate da latitudine e longitudine ed associata ad un codice numerico univoco di cinque cifre. La prima e l’ultima fermata sono contrassegnate come capilinea. Ciascuna tratta viene coperta da un numero predefinito di veicoli, la cui associazione viene gestita dai gestori del servizio. Ogni capolinea ha un orario di partenze prestabilito. Gli amministratori del servizio gestiscono anche i conducenti, caratterizzati da un codice fiscale, un nome, un cognome, una data di nascita ed un luogo di nascita. Di ogni conducente è di interesse conoscere anche il numero di patente e la data di scadenza della stessa. Gli utenti del sistema possono accedere al servizio per conoscere, dato il codice di una fermata, a quante fermate di distanza si trova un veicolo.  Quando salgono a bordo, gli utenti del servizio timbrano un biglietto elettronico o un abbonamento sul “validatore intelligente” installato sui veicoli. Nel caso di un biglietto elettronico, questo viene marcato come “utilizzato” all’interno del sistema. L’emissione di nuovi biglietti viene amministrata dai gestori del servizio. Nel caso di abbonamento, viene verificato se esso è ancora valido e viene memorizzata la data dell’ultimo utilizzo. Quando un autista si trova ad un capolinea, può interrogare il sistema per sapere qual è la prossima partenza prevista del veicolo che sta guidando. |

Si deduce che:

* Gli abbonamenti sono nominativi e validi su qualsiasi tratta fino alla scadenza;
* I biglietti elettronici possono essere utilizzati su una sola tratta;
* Il codice univoco citato in riga 6 è associato alla fermata;
* L’azienda copre una città di medie dimensioni e ha disposizione soltanto autobus;
* Ogni veicolo che copre una tratta ha un orario di partenze prestabilito;
* Qualsiasi tipologia di biglietto viene acquistata all’esterno del sistema.

# Analisi dei Requisiti

## Identificazione dei termini ambigui e correzioni possibili

|  |  |  |  |
| --- | --- | --- | --- |
| **Linea** | **Termine** | **Nuovo termine** | **Motivo correzione** |
| 9 | Capolinea | Veicolo in partenza dal Capolinea | L’orario è relativo alla partenza del veicolo da un capolinea che copre una determinata tratta e non dal capolinea. |
| 16 | Utenti del servizio | Viaggiatori | Confusione tra utenti del servizio e utenti del sistema, entrambe nel testo fanno riferimento alla stessa categoria. |
| 14 | Sevizio | Sistema | Gli utenti accedono al sistema. Il servizio viene rappresentato dagli autobus. |
| 21 | Autista | Conducente | Non c’è differenza semantica tra i due termini. |
| 10 | Amministratori | Gestori | Non c’è differenza semantica tra i due termini. |
| 6 | Associata ad | Ogni fermata identificata da | Il codice numero univoco di cinque cifre identifica la fermata e non la tratta. |
| 14 | Utenti del sistema | Viaggiatori | Sintetizzo che chi accede al sistema sono gli stessi viaggiatori oppure chi intende viaggiare. |
| 15 | Il Veicolo | Il veicolo in corsa più vicino | I viaggiatori intendono sapere quanto è distante il prossimo veicolo partito che passa dalla fermata e non un veicolo generico. |

### Specifica disambiguate

|  |
| --- |
| Si intende realizzare il sistema informativo di un’azienda di trasporto pubblico locale. L’azienda è dotata di un parco veicoli che permettono di coprire un determinato insieme di tratte. I veicoli sono caratterizzati da una matricola (codice univoco numerico di quattro cifre). Ogni veicolo è anche associato ad una data di acquisto. Ciascuna tratta ha un insieme di fermate identificate da un codice numerico univoco di cinque cifre e da latitudine e longitudine. La prima e l’ultima fermata sono contrassegnate come capilinea.  Ciascuna tratta viene coperta da un numero predefinito di veicoli, la cui associazione viene gestita dai gestori del servizio. I veicoli in partenza da un capolinea hanno un orario di partenze prestabilito. I gestori del servizio gestiscono anche i conducenti, caratterizzati da un codice fiscale, un nome, un cognome, una data di nascita ed un luogo di nascita. Di ogni conducente è di interesse conoscere anche il numero di patente e la data di scadenza della stessa. I viaggiatori possono accedere al sistema per conoscere, dato il codice di una fermata, a quante fermate di distanza si trova il veicolo in corsa più vicino.  Quando salgono a bordo, i viaggiatori timbrano un biglietto elettronico o un abbonamento sul “validatore intelligente” installato sui veicoli. Nel caso di un biglietto elettronico, questo viene marcato come “utilizzato” all’interno del sistema. L’emissione di nuovi biglietti viene amministrata dai gestori del servizio. Nel caso di abbonamento, viene verificato se esso è ancora valido e viene memorizzata la data dell’ultimo utilizzo.  Quando un conducente si trova ad un capolinea, può interrogare il sistema per sapere qual è la prossima partenza prevista del veicolo che sta guidando. |

## Glossario dei Termini

|  |  |  |  |
| --- | --- | --- | --- |
| **Termine** | **Descrizione** | **Sinonimi** | **Collegamenti** |
| Veicolo | Rappresenta il collegamento fisico, in partenza da un capolinea per coprire una determinata tratta. | Tratta | Conducente, Tratta |
| Conducente | È colui che guida un veicolo. | Autista | Veicolo |
| Tratta | Descrive il collegamento tra due capolinea. Effettua un insieme di fermate intermedie. | Veicolo | Fermata, Capolinea |
| Capolinea | Identifica il fermata di partenza e di arrivo di una determinata tratta. | Fermata | Tratta, Fermata |
| Fermata | Identifica un punto di stop intermedio per un veicolo che copre una tratta dove i viaggiatori possono salire/scendere dal veicolo. | Capolinea | Tratta |
| Biglietto | Rappresenta il titolo di viaggio per usufruire del servizio di trasporto pubblico. | Biglietto elettronico, Abbonamento |  |
| Abbonamento | Titolo di viaggio nominativo. | Biglietto | Biglietto |

## Raggruppamento dei requisiti in insiemi omogenei

|  |
| --- |
| **Frasi relative a Veicolo** |
| I veicoli sono caratterizzati da una matricola (codice univoco numerico di quattro cifre). Ogni veicolo è anche associato ad una data di acquisto. Ogni veicolo in partenza da un capolinea ha un orario di partenze prestabilito. |

|  |
| --- |
| **Frasi relative a Conducente** |
| Gli gestori del servizio gestiscono anche i conducenti, caratterizzati da un codice fiscale, un nome, un cognome, una data di nascita ed un luogo di nascita. Di ogni conducente è di interesse conoscere anche il numero di patente e la data di scadenza della stessa.  Quando un conducente si trova ad un capolinea, può interrogare il sistema per sapere qual è la prossima partenza prevista del veicolo che sta guidando. |

|  |
| --- |
| **Frasi relative a Tratta** |
| Si intende realizzare il sistema informativo di un’azienda di trasporto pubblico locale. L’azienda è dotata di un parco veicoli che permettono di coprire un determinato insieme di tratte. Ciascuna tratta ha un insieme di fermate identificate da latitudine e longitudine ed associata ad un codice numerico univoco di cinque cifre. La prima e l’ultima fermata sono contrassegnate come capilinea. Ciascuna tratta viene coperta da un numero predefinito di veicoli, la cui associazione viene gestita dai gestori del servizio. Ciascuna tratta ha un insieme di fermate … viene coperta da un numero predefinito di veicoli. |

|  |
| --- |
| **Frasi relative a Fermate** |
| La tratta ha un insieme di fermate identificate da latitudine e longitudine ed associata da un codice numerico univoco di cinque cifre. La prima e l’ultima fermata sono contrassegnate come capilinea. |

|  |
| --- |
| **Frasi relative a Biglietto** |
| … i viaggiatori timbrano un biglietto elettronico o un abbonamento sul “validatore intelligente” installato sui veicoli. Nel caso di un biglietto elettronico, questo viene marcato come “utilizzato” all’interno del sistema. L’emissione di nuovi biglietti viene amministrata dai gestori del servizio. Nel caso di abbonamento, viene verificato se esso è ancora valido e viene memorizzata la data dell’ultimo utilizzo. |

# Progettazione concettuale

## Costruzione dello schema E-R

### Per la costruzione dello schema E-R finale ho utilizzato una strategia mista, combinando i vantaggi della strategia top-down con quelli della strategia bottom-up. Descrivo gli schemi parziali in ordine di come vengono citati nel testo.



### Nell’analizzare il testo, l’entità Conducente è in relazione con l’entità Veicolo in Corsa. Attribuisco come identificatore gli attributi *CF* (codice fiscale) e *Numero patente* poiché unici per singolo individuo, e come attributi semplici *Nome*, *Cognome*, *Data scadenza patente* e un attributo composto formato da *Data* e *Luogo di nascita*. Le cardinalità dell’associazione prendono valore: Conducente (1,N) poiché il conducente può guidare più di un veicolo e suppongo che ne guidi almeno uno, Veicolo in Corsa (1,1) poiché il servizio della corsa lo offre un solo conducente.

Per garantire le funzionalità delle operazioni del sistema tra cui la tracciabilità del veicolo in tempo reale e per gestire gli orari di partenza dei veicoli dai capilinea, si decide di creare una nuova entità debole chiamata **Veicolo in Corsa** che prende come identificatori *Giorno* della settimana, *Orario* giornaliero di partenza e *Matricola* del **Veicolo** del servizio. *Partito* e *Ultima fermata* indicano se tale veicolo è già in corsa e l’indice corrispondente, della tratta associata, dell’ultima fermata effettuata se presente. Questa entità è in relazione con l’entità **Veicolo** e l’entità **Tratta**. Veicolo ha come identificatore *Matricola* e attributo semplice la *Data di acquisto* del mezzo. Tratta possiede come attributo *Numero*. *Numero* non è un identificatore per tratta poiché essa dipende dal capolinea di partenza e di arrivo, cui determina inoltre l’ordine e la sequenza delle eventuali fermate presenti.



Successivamente decido di fare una generalizzazione parziale e sovrapposta (esistono fermate che sono anche capolinea) con lo scopo di descrivere al meglio le seguenti entità: si ha un'entità padre che è **Fermata** e un’entità figlia **Capolinea**. Fermata possiede come identificatori: il *Codice* univoco di cinque cifre descritto nel testo e anche l’identificatore composto da *Latitudine* e *Longitudine* poiché suppongo che non possano essere presenti due diverse fermate in uno stesso punto. Capolinea rappresenta il luogo, fermata, di partenza per una tratta.

**Tratta** è un’entità debole che prende come identificatore *Numero* e *Capolinea* di partenza e di arrivo. È presente un’associazione tra Tratta e Fermate, che possiede un attributo *Indice* cui scopo è quello di dare una sequenza ordinata alle fermate da effettuare. Suppongo che ne sostenga almeno una, che è l’eventuale capolinea di arrivo.



Per l’entità **Biglietto** creo una generalizzazione totale ed esclusiva: le entità figlie sono rappresentate da **Biglietto elettronico** e **Abbonamento**. Creo un identificatore per Biglietto con il nome di *Numero* che serve a identificare univocamente qualsiasi tipologia di biglietto. Biglietto elettronico è composto da un attributo *Valido* che indica la validità del biglietto. Abbonamento ha come attributi semplici *Data Utilizzo*, poiché gli abbonamenti sono nominativi, e la data della *Scadenza*. Questa entità è in relazione con **Abbonato**, che possiede come attributi *Codice Fiscale*, *Nome* e *Cognome*. Suppongo che un abbonato non può possedere più di un abbonamento.



### Integrazione finale



## Regole aziendali

## Un viaggiatore può avere al più un abbonamento.

## Il numero di veicoli deve essere maggiore di zero.

Il numero della tratta può essere uguale per al più su due diverse tratte.

Gli orari in uno stesso giorno dello stesso veicolo devono avere una distanza temporale sufficiente che gli permetta di svolgere eventuali corse successive.

## Dizionario dei dati

|  |  |  |  |
| --- | --- | --- | --- |
| **Entità** | **Descrizione** | **Attributi** | **Identificatori** |
| Tratta | Descrive il collegamento tra due capolinea. | Numero | Numero, Capolinea\_Da, Capolinea\_A |
| Fermata | Luogo di temporanea sospensione della tratta per consentire la salita e la discesa dei viaggiatori. | Codice, Latitudine e Longitudine | Codice, Latitudine e Longitudine |
| Conducente | Colui che guida il veicolo. | Nome, Cognome, CF, Luogo e Data di Nascita, Numero e Data Scadenza Patente | CF, Numero di Patente |
| Biglietto | Titolo di viaggio singolo per usufruire del servizio di trasporto pubblico. | Numero | Numero |
| Abbonamento | Titolo di viaggio nominativo per usufruire del servizio di trasporto pubblico. | Data Utilizzo, Scadenza |  |
| Veicolo | Rappresenta il collegamento fisico, per coprire una determinata tratta. | Matricola, Data Acquisto | Matricola |
| Veicolo in Corsa | Rappresenta un’istanza di Veicolo, vale a dire la Corsa che copre una tratta in un certo orario di uno specifico giorno. | Giorno, Ora, Veicolo, Tratta Partito, Ultima Fermata | Giorno, Ora, Veicolo |
| Capolinea | Fermata terminale di un servizio di trasporto dalla quale ha inizio la corsa in senso contrario |  | Codice, Latitudine e Longitudine |
| Abbonato | Colui che possiede un abbonamento. | CF, Nome, Cognome | CF |

# Progettazione logica

## Volume dei dati

Nell’analisi si ipotizza che il sistema mantenga i dati relativi ai Biglietti, sia elettronici che abbonamenti, fino a 30 giorni dalla loro non validità, di conseguenza vengono eliminati anche quelli di Abbonato (on delete cascade). I dati relativi al Veicolo in Corsa vengono mantenuti così da poter mantenere le stesse istanze.

Si considera che vengono coperte ogni giorno tutte le possibili tratte. Inoltre, si suppone che mediamente gli abbonamenti coprono il 15% del totale dei biglietti in corso di validità. Si suppone che ogni tratta abbia un flusso di veicoli in partenza ogni 60 minuti e che non ci siano distinzioni di flusso dei Veicoli in Corsa tra giorni feriali e festivi. Inoltre, si assume che i conducenti svolgano un orario di lavoro di 48 ore settimanali. Gli orari di tutti i Veicoli in Corsa siano compresi nell’intervallo 05:30-21:30.

|  |  |  |
| --- | --- | --- |
| **Concetto nello schema** | **Tipo[[1]](#footnote-1)** | **Volume atteso** |
| Tratta | E | 24 |
| Veicolo | E | 40 |
| Conducente | E | 70 |
| Biglietto | E | 115.000 |
| Biglietto Elettronico | E | 97.000 |
| Abbonamento | E | 18.000 |
| Abbonato | E | 18.000 |
| Fermata | E | 132 |
| Capolinea | E | 40 |
| Veicolo in Corsa | E | 2.680 |
| Abbonato | E | 18.000 |
| Da | R | 24 |
| A | R | 24 |
| Copre | R | 2.680 |
| Guida | R | 2.680 |
| Effettua | R | 264 |
| Parte | R | 2.680 |
| Appartiene | R | 18.000 |

## Tavola delle operazioni

|  |  |  |
| --- | --- | --- |
| **Cod.** | **Descrizione** | **Frequenza attesa** |
| V1 | Lista dei veicoli in corsa passanti per fermata. | 10/giorno |
| V2 | Distanza veicolo dalla fermata. | 300/giorno |
| V3 | Timbro biglietto. | 9.580/mese |
| V4 | Lista giornaliera tratte passanti per la fermata. | 5/giorno |
| C1 | Prossimo orario partenza. | 384/giorno |
| C2 | Aggiorna dati patente. | 10/anno |
| C3 | Elenco orario lavorativo. | 10/settimana |
| G1 | Aggiungi orario partenza di un veicolo. | 250/mese |
| G2 | Aggiorna conducente a veicolo in corsa. | 250/settimana |
| G3 | Aggiungi fermata in una tratta. | 10/anno |
| G4 | Aggiungi tratta. | 5/anno |
| G5 | Registra conducente. | 5/anno |
| G6 | Elenco scadenze patente. | 2/mese |
| G7 | Lista tratte totali. | 20/mese |
| L1 | Login. | 70/giorno |

Sono presenti 5 stored procedure utilizzate dal sistema, presenti alla fine del file.

## Costo delle operazioni

**V1 – LISTA DEI VEICOLI IN CORSA**

* 1/LETTURA - FERMATA
* 5/LETTURA - EFFETTUA
* 5/LETTURA – VEICOLO IN CORSA

Costo totale: 11

Accesi/giorno: 110

**V2 – DISTANZA VEICOLO**

* 1/LETTURA - FERMATA
* 1/LETTURA - EFFETTUA
* 1/LETTURA – VEICOLO IN CORSA

Costo totale: 3

Accesi/giorno: 900

**V3 – TIBRO BIGLIETTO**

Decido successivamente di accorpare le entità Biglietto elettronico e Abbonamento con Biglietto, aggiungendo un attributo di nome Tipo.

* 1/LETTURA – BIGLIETTO
* 1/SCRITTURA - BIGLIETTO

Costo totale: 3

Accessi/mese: 28.740

**V4 – LISTA TRATTE**

* 1/LETTURA - FERMATA
* 5/LETTURA - EFFETTUA

Costo totale: 3

Accesi/giorno: 900

**C1 – PROSSIMO ORARIO PARTENZA**

* 1/LETTURA – VEICOLO IN CORSA

Costo totale: 1

Accessi/giorno: 384

**C2 – AGGIORNA DATI PATENTE**

* 1/LETTURA – CONDUCENTE
* 2/SCRITTURA – CONDUCENTE

Costo totale: 3

Accessi/anno: 30

**C3 – ELENCO ORARIO LAVORATIVO**

* 48/LETTURA – VEICOLO IN CORSA

Costo totale: 48

Accessi/settimana: 480

**G1 – AGGIUNGI ORARIO PARTENZA DI UN VEICOLO**

* 1/LETTURA – VEICOLO IN CORSA
* 7/SCRITTURA – VEICOLO IN CORSA

Costo totale: 15

Accessi/mese: 3.750

**G2 – AGGIORNA CONDUCENTE A VEICOLO IN CORSA**

* 1/LETTURA – VEICOLO IN CORSA
* 1/SCRITTURA – VEICOLO IN CORSA

Costo totale: 3

Accessi/settimana: 750

**G3 – AGGIUNGI FERMATA AD UNA TRATTA**

* 1/LETTURA – FERMATA
* 8/SCRITTURA – EFFETTUA

Costo totale: 17

Accessi/anno: 170

**G4 – AGGIUNGI TRATTA**

* 1/SCRITTURA – TRATTA

Costo totale: 2

Accessi/anno: 10

**G5 – REGISTRA CONDUCENTE**

* 1/SCRITTURA - UTENTI
* 8/SCRITTURA – CONDUCENTE

Costo totale: 18

Accessi/anno: 90

**G6 – ELENCO SCADENZE PATENTE**

* 50/LETTURA – CONDUCENTE

Costo totale: 50

Accessi/mese: 100

**G7 – LISTA TRATTE TOTALI**

* 24/LETTURA – TRATTA

Costo totale: 24

Accessi/mese: 480

**L1 – LOGIN**

Questa operazione richiede l’accesso in lettura ad un’entità Utente, che viene introdotta successivamente.

Costo totale: 1

Accessi/giorno: 70

## Ristrutturazione dello schema E-R

Si decide l’accorpamento delle entità **Biglietto Elettronico** e **Abbonamento** della generalizzazione dell’entità **Biglietto**. Dunque, le entità figlie vengono eliminate e le loro proprietà (attributi) vengono aggiunte nell’entità genitore. Questa scelta ci assicura un numero minori di accessi all’operazione V3. L’entità **Biglietto** oltre all’identificatore *Numero* assume altri due attributi *Tipo*, che serve a distinguere i due differenti tipi di biglietti, e *Valido*. L’associazione Appartiene lega adesso Biglietto e Abbonato, con cardinalità rispettivamente (0,1) e (1,1). Si mantiene l’entità Abbonato anche se di non significativa importanza ai fini delle funzionalità dell’applicazione, evita però la presenza di eventuali valori di attributi nulli, e mantiene i dati per il controllo dell’Abbonamento.

Per quanto riguarda la generalizzazione Capolinea, decido di rimuoverla totalmente senza l’aggiunta di un ulteriore attributo su Fermata.

Decido inoltre di assegnare come identificatore unico *CF* all’entità **Conducente** e *Codice* all’entità **Fermata**, poiché un identificatore composto da un singolo attributo è da preferire a identificatore composti da più attributi (Longitudine e Latitudine). Si scompone anche l’attributo composto Nascita, rispettivamente in *Data di Nascita* e *Luogo di Nascita*.



## Trasformazione di attributi e identificatori

Si preferisce aggiungere un attributo *ID* all’entità **Tratta**, che funge da identificatore, eliminando così l’identificatore esterno. Analizzando l’entità Veicolo in Corsa, un suo eventuale partizionamento non comporta una riduzione dello spazio di memoria.

## Traduzione di entità e associazioni

* Si costruisce uno schema relazionale a partire dallo schema E-R ristrutturato a partire con la relativa traduzione dell’entità forti:

Conducente (CF, Nome, Cognome, DataNascita, LuogoNascita, NumeroPatente, ScadenzaPatente)

Veicolo (Matricola, DataAcquisto)

Biglietto (Numero, Valido, Tipo)

Abbonato (CodiceFiscale, Nome, Cognome, Biglietto, Data Scadenza, Ultimo Utilizzo)

Fermata (Codice, Latitudine, Longitudine)

Tratta (ID, Numero, Capolinea Partenza, Capolinea Arrivo)

Di cui i seguenti vincoli di integrità:

Tratta (Capolinea Partenza) ⊆ Fermata (Codice)

Tratta (Capolinea Arrivo) ⊆ Fermata (Codice)

Abbonato (Biglietto) ⊆ Biglietto (Numero)

* Si trasforma l’entità debole VeicoloinCorsa come segue:

VeicoloinCorsa (Veicolo, Giorno, Ora, Tratta, Partito, Conducente, Ultima Fermata\*)

Di cui:

VeicoloinCorsa (Tratta) ⊆ Tratta (ID)

VeicoloinCorsa (Conducente ) ⊆ Conducente(CF)

Veicoloincorsa (Veicolo) ⊆ Veicolo(Matricola)

* Per ultimo si trasformano le associazioni rimaste:

Effettua (Tratta, Fermata, Indice)

Di cui:

Effettua (Tratta) ⊆ Tratta(ID)

Effettua(Fermata) ⊆ Fermata(Codice)



## Normalizzazione del modello relazionale

Il modello relazionale precedente è in terza forma normale, 3BCNF, poiché ogni relazione soddisfa le proprietà del modello.

# Progettazione fisica

## Utenti e privilege

Si prevedono tre ruoli, per implementare il Principle of Least Privilege:

* Login:

Grant in esecuzione sull’operazione L1

* Viaggiatore:

Grant in esecuzione sulle operazione V1, V2, V3, V4

* Gestore del servizio

Grant in esecuzione sulle operazioni G1, G2, G3, G4, G5, G6, G7

* Conducente

Grant in esecuzione sull’operazione C1, C2, C3

Per identificare l’accesso al sistema si introduce una tabella Utenti. Essa è costituita dagli attributi *Username, Password* e *Ruolo.* Per permettere il recupero delle informazioni dei Conducenti che ha effettuato l’accesso si aggiunge l’attributo *Username* alla tabella Conducente con un vincolo di integrità referenziale alla tabella Utenti.



## Strutture di memorizzazione

|  |  |  |
| --- | --- | --- |
| **Tabella Conducenti** | | |
| **Colonna** | **Tipo di dato** | **Attributi[[2]](#footnote-2)** |
| CF | VARCHAR(20) | PK, NN |
| Nome | VARCHAR(45) | NN |
| Cognome | VARCHAR(45) | NN |
| NumeroPatente | VARCHAR(12) | NN, UQ |
| ScadenzaPatente | DATE | NN |
| LuogoNascita | VARCHAR(45) | NN |
| DataNascita | DATE | NN |
| Username | VARCHAR(45) | NN |

|  |  |  |
| --- | --- | --- |
| **Tabella Veicolo** | | |
| **Colonna** | **Tipo di dato** | **Attributi[[3]](#footnote-3)** |
| Matricola | SMALLINT(4) | PK, NN |
| DataAcquisto | DATE | NN |

|  |  |  |
| --- | --- | --- |
| **Tabella Fermata** | | |
| **Colonna** | **Tipo di dato** | **Attributi** |
| Codice | INT(5) | PK, NN |
| Longitudine | DECIMAL | NN |
| Latitudine | DECIMAL | NN |

|  |  |  |
| --- | --- | --- |
| **Tabella Tratta** | | |
| **Colonna** | **Tipo di dato** | **Attributi** |
| ID | TINYINT(2) | PK, NN, AI |
| Numero | SMALLINT(3) | NN |
| Da | INT(5) | NN |
| A | INT(5) | NN |

|  |  |  |
| --- | --- | --- |
| **Tabella VeicoloinCorsa** | | |
| **Colonna** | **Tipo di dato** | **Attributi** |
| Giorno | VARCHAR(45) | PK, NN |
| Ora | TIME | PK, NN |
| Veicolo | SMALLINT(4) | PK, NN |
| Tratta | TINYINT(2) | NN |
| Ultima Fermata | TINYINT(2) | NN |
| Partito | BOOLEAN | NN |
| Conducente | VARCHAR(16) | NN |

|  |  |  |
| --- | --- | --- |
| **Tabella Biglietto** | | |
| **Colonna** | **Tipo di dato** | **Attributi** |
| Numero | INT | PK, NN |
| Valido | BOOLEAN | NN |
| Tipo | VARCHAR(12) | NN |

|  |  |  |
| --- | --- | --- |
| **Tabella Abbonato** | | |
| **Colonna** | **Tipo di dato** | **Attributi** |
| CodiceFiscale | VARCHAR(45) | PK, NN |
| Nome | VARCHAR(45) | NN |
| Cognome | VARCHAR(45) | NN |
| Biglietto | INT | NN |
| DataScadenza | DATE | NN |
| UltimoUtilizzo | DATE |  |

|  |  |  |
| --- | --- | --- |
| **Tabella Utenti** | | |
| **Colonna** | **Tipo di dato** | **Attributi** |
| Username | VARCHAR(45) | PK, NN |
| Password | CHAR(32) | NN |
| Ruolo | Enum (‘Conducente’, ‘Gestore’, ‘Viaggiotore’) | NN |

|  |  |  |
| --- | --- | --- |
| **Tabella Effettua** | | |
| **Colonna** | **Tipo di dato** | **Attributi** |
| Fermata\_Codice | INT(5) | PK, NN |
| Tratta\_ID | TINYINT(2) | PK, NN |
| Indice | TINYINT(2) | NN |

## Indici

Si introduce l’indice orario\_veicolo per migliorare le prestazioni dell’operazione C1 e l’indice ora\_UNIQUE per assicurare che venga assegnata una sola corsa al veicolo in un orario preciso.

|  |  |
| --- | --- |
| **Tabella VeicoloinCorsa** | |
| **Indice orario\_veicolo** | **Tipo[[4]](#footnote-4):** |
| Conducente, Giorno, Ora | IDX |
| **Indice ora\_UNIQUE** | **Tipo:** |
| Veicolo, Giorno, Ora | UQ |

Si introduce l’indice coordinate\_UNIQUE poiché in uno stesso punto si suppone non possono esserci due diverse fermate.

|  |  |
| --- | --- |
| **Tabella Fermata** | |
| **Indice coordinate\_UNIQUE** | **Tipo:** |
| Longitudine, Latitudine | UQ |

## Trigger

* Si introduce il seguente trigger prima di un inserimento su Fermata per un check sull’inserimento del codice della fermata che deve soddisfare il formato di 5 cifre

CREATE DEFINER = CURRENT\_USER TRIGGER `mydb`.`Fermata\_BEFORE\_INSERT` BEFORE INSERT ON `Fermata` FOR EACH ROW

BEGIN  
  
IF NOT (char\_length(new.codice) = 5 and new.codice REGEXP '^[0-9]+$') THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Formato codice fermata non valido.';

END IF;

END

* Si introduce un trigger su tratta per un check prima di un inserimento che il numero della tratta non viene usato su al più due tratte e un ulteriore controllo sui capilinea, ovvero è possibile avere un numero di tratta uguale solo se è la stessa tratta ma con i capolinea invertiti.

CREATE DEFINER = CURRENT\_USER TRIGGER `mydb`.`Tratta\_BEFORE\_INSERT` BEFORE INSERT ON `Tratta` FOR EACH ROW

BEGIN

declare numero\_tratte int;

declare capolinea\_da1 smallint(5);

declare capolinea\_a1 smallint(5);

select count(\*)

into numero\_tratta

from Numero

where Numero = New.Numero;

if numero\_tratte = 2 then

signal sqlstate '45000'

set message\_text = 'Non è possibile inserire la tratta. Numero tratta già utilizzato';

END IF;

select Capolinea\_Da, Capolinea\_A

into capolinea\_da1, capolinea\_a1

from Tratta

where Numero = new.Numero;

if capolinea\_da1 is not null and capolinea\_a1 is not null then

if new.Capolinea\_Da != capolinea\_a1 or new.Capolinea\_A != capolinea\_da1 then

signal sqlstate '45000’

set message\_text = 'Non è possibile inserire la tratta. I capolinea non corrispondono al numero della tratta.'

END IF;

END IF;

END

* Si introducono i seguenti due trigger con lo scopo di controllare prima di un inserimento/aggiornamento su conducente il formato del Codice Fiscale e del Numero Patente.

CREATE DEFINER = CURRENT\_USER TRIGGER `mydb`.`Conducente\_BEFORE\_INSERT` BEFORE INSERT ON `Conducente` FOR EACH ROW

BEGIN  
  
IF NOT [NEW.CF](http://NEW.CF) REGEXP '^[A-Z]{6}[0-9]{2}[A-Z][0-9]{2}[A-Z][0-9]{3}[A-Z]$' THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Codice Fiscale non valido.';

END IF;

END

CREATE DEFINER = CURRENT\_USER TRIGGER `mydb`.`Conducente\_BEFORE\_UPDATE`

BEFORE UPDATE ON `Conducente` FOR EACH ROW

BEGIN  
  
IF NOT NEW.NumeroPatente REGEXP '^[A-Za-z0-9]{5,10}$' THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Formato del numero di patente non valido.';

END IF;

END

## Eventi

* RIMOZIONI BIGLIETTI NON VALIDI OGNI 30 GIORNI

NOTA: ABBONATO VIENE ELIMINATO/AGGIORNATO DI CONSCEGUENZA (DELETE ON CASCADE, UPDATE ON CASCADE);

CREATE EVENT IF NOT EXISTS biglietti\_non \_validi

ON SCHEDULE EVERY 30 DAY

CURRENT TIMESTAMP + INTERVAL 30 DAY

ON COMPLETION PRESERVE

DO

BEGIN

DELETE FROM BIGLIETTO

WHERE VALIDO = 0;

END

* CONTROLLO GIORNALIERO DATA SCADENZA DEGLI ABBONAMENTI CON SETTAGGIO DELLA VALIDITà =0

CREATE EVENT IF NOT EXISTS controllo\_abbonamento

ON SCHEDULE EVERY 1 DAY

STARTS ‘2024-06-12 00:01:00’

ON COMPLETION PRESERVE

DO

BEGIN

UPDATE BIGLIETTO

JOIN ABBONATO ON NUMERO = BIGLIETTO\_NUMERO

SET VALIDO = 0

WHERE DATASCADENZA < CURDATE();

END

## Viste

Non sono state utilizzate viste.

## Stored Procedures e transazioni

OPERAZIONE V1

CREATE PROCEDURE `lista\_veicoli` (in var\_fermata int(5))

BEGIN

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level repeatable read;

set transaction read only;

start transaction;

select Tratta, Veicolo, (Indice-UltimaFermata) as Distanza

from VeicoloinCorsa as V join Effettua as E on Tratta = Tratta\_ID

where E.Fermata\_Codice = var\_fermata and V.Giorno = dayname(now()) and Partito = 1

and V.UltimaFermata < E.Indice

and E.Indice != (select count(\*)

from Tratta as T join Effettua as Ef on T.ID = Ef.Tratta\_ID

where T.ID = V.Tratta);

commit;

END

OPERAZIONE V2

CREATE PROCEDURE `distanza\_veicolo` (in var\_fermata int(5), in var\_idTratta tinyint(3))

BEGIN

declare var\_index int;

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level read committed;

set transaction read only;

start transaction;

select count(\*)

into var\_index

from Effettua

where Tratta\_ID = var\_idTratta;

select Veicolo, (Indice - UltimaFermata) as Distanza

from VeicoloinCorsa join Effettua on Tratta = Tratta\_ID

where Fermata\_Codice = var\_fermata and Giorno = dayname(now())

and Tratta = var\_idTratta and Indice != (var\_index) and Partito = 1;

commit;

END

OPERAZIONE V3

CREATE PROCEDURE `convalida\_biglietto` (in var\_numero int)

BEGIN

declare var\_tipo varchar(12);

declare var\_count INT;

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level serializable;

start transaction;

select Tipo, count(\*)

into var\_tipo, var\_count

FROM Biglietto

WHERE Numero = var\_numero AND Valido = 1

group by Tipo;

if var\_tipo = 'Standard' then

update Biglietto

set Valido = 0

where Numero = var\_numero;

elseif var\_tipo = 'Abbonamento' then

update Abbonato

join Biglietto on Numero = Biglietto\_Numero

set UltimoUtilizzo = current\_date()

where Biglietto\_Numero = var\_numero;

else

signal sqlstate '45000'

set message\_text = 'Biglietto non valido o non esistente.';

end if;

COMMIT;

END

OPERAZIONE V4

CREATE PROCEDURE `lista\_tratte` (in var\_fermata int(5))

BEGIN

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level read committed;

set transaction read only;

start transaction;

select Numero, Capolinea\_Da, Capolinea\_A, Ora

from VeicoloinCorsa join Effettua on Tratta = Tratta\_ID join Tratta on Tratta\_ID = ID

where Fermata\_Codice = var\_fermata and Giorno = dayname(now())

and Partito = 0 and Ora >= current\_time()

order by Numero;

commit;

END

OPERAZIONE C1

CREATE PROCEDURE `prossima\_partenza` (in var\_conducente varchar(20))

BEGIN

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level read uncommitted;

set transaction read only;

start transaction;

select Giorno, Ora

from VeicoloinCorsa

where Conducente = var\_conducente and Giorno = dayname(now())

and Ora >= current\_time();

commit;

END

OPERAZIONE C2

CREATE PROCEDURE `aggiorna\_patente` (in var\_conducente varchar(20), in var\_numero varchar(12), in var\_scadenza date)

BEGIN

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level read uncommitted;

start transaction;

update Conducente

set NumeroPatente = var\_numero, ScadenzaPatente = var\_scadenza

where CF = var\_conducente;

commit;

END

OPERAZIONE C3

CREATE PROCEDURE `elenco\_orari` (in var\_conducente varchar(20))

BEGIN

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level read committed;

set transaction read only;

start transaction;

select Veicolo, Giorno, Ora

from VeicoloinCorsa

where Conducente = var\_conducente and Partito = 0

order by Giorno, Ora;

commit;

END

OPERAZIONE G1

CREATE PROCEDURE `nuova\_partenza` (in var\_veicolo smallint(4), in var\_tratta tinyint(2), in var\_giorno varchar(45), in var\_orario time, in var\_CF varchar(20))

BEGIN

declare var\_count int;

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level serializable;

start transaction;

select count(\*)

into var\_count

from VeicoloinCorsa as V

where V.Veicolo = var\_veicolo and V.Giorno = var\_giorno and V.Partito = 1

and V.Ora = (var\_orario - interval 30 minute)

and V.UltimaFermata != (select count(\*)

from Effettua

where V.Tratta = Tratta\_ID);

if var\_count > 0 then

signal sqlstate '45000'

set message\_text = 'Il veicolo è in corsa su una tratta. Non è possibile inserire orario';

else

DELETE FROM VeicoloinCorsa

WHERE Veicolo = var\_veicolo and Giorno = var\_giorno and Ora between var\_orario - interval 30 minute and var\_orario + interval 30 minute;

insert into VeicoloinCorsa(Giorno, Ora, Veicolo, Tratta, Partito, UltimaFermata, Conducente) values (var\_giorno, var\_orario, var\_veicolo, var\_tratta, 0, 0, var\_CF);

end if;

commit;

END

OPERAZIONE G2

CREATE PROCEDURE `assegna\_conducente\_corsa` (in var\_CF varchar(20), in var\_veicolo smallint(4), in var\_giorno varchar(45), in var\_ora time)

BEGIN

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level repeatable read;

start transaction;

update VeicoloinCorsa

set Conducente = var\_CF

where Veicolo = var\_veicolo and Giorno = var\_giorno and Ora = var\_ora;

commit;

END

OPERAZIONE G3

CREATE PROCEDURE `aggiungi\_fermata\_tratta` (in var\_fermata int(5), in var\_indice int, in var\_tratta tinyint(2))

BEGIN

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level repeatable read;

start transaction;

update Effettua

set indice = indice +1

where Tratta\_ID = var\_tratta and indice >= var\_indice;

insert into Effettua(Indice, Fermata\_Codice, Tratta\_ID) values (var\_indice, var\_fermata, var\_tratta);

commit;

END

OPERAZIONE G4

CREATE PROCEDURE `aggiungi\_tratta` (in var\_ID tinyint(2), in var\_num smallint(3), in var\_partenza int(5), in var\_destinazione int(5))

BEGIN

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level read uncommitted;

start transaction;

insert into Tratta(ID, Numero, Capolinea\_Da, Capolinea\_A) values (var\_ID, var\_num, var\_partenza, var\_destinazione);

commit;

END

OPERAZIONE G5

CREATE PROCEDURE `registra\_conducente` (in var\_CF varchar(20), in var\_nome varchar(45), in var\_cognome varchar(45), in var\_patente varchar(10), in var\_scadenza DATE, in var\_luogo varchar(45), in var\_data DATE, in var\_username varchar(45), in var\_pass char(32))

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level read uncommitted;

start transaction;

if (var\_username not in (select Username from Utenti)) then

insert into Utenti(Username, Password, Ruolo) values (var\_username, md5(var\_pass), 'Conducente');

insert into Conducente(CF, Nome, Cognome, NumeroPatente, ScadenzaPatente, LuogoNascita, DataNascita, Username) values (var\_CF, var\_nome, var\_cognome, var\_patente, var\_scadenza, var\_luogo, var\_data, var\_username);

else

signal sqlstate '45000'

set message\_text = 'Username già esistente.';

end if;

commit;

END

OPERAZIONE G6

CREATE PROCEDURE `elenco\_scadenze\_patente` ()

BEGIN

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level read uncommitted;

set transaction read only;

start transaction;

select CF, NumeroPatente, ScadenzaPatente

from Conducente;

commit;

END

OPERAZIONE G7

CREATE PROCEDURE `lista\_tratte\_totali` ()

BEGIN

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level read committed;

set transaction read only;

start transaction;

select \*

from Tratta;

commit;

END

OPERAZIONE L1

CREATE PROCEDURE `login` (in var\_username varchar(45), in var\_pass varchar(45), out var\_role INT, out var\_CF varchar(20))

BEGIN

declare var\_user\_role ENUM('Conducente','Gestore','Viaggiatore');

select Ruolo

from Utenti

where Username = var\_username and Password = md5(var\_pass)

into var\_user\_role;

if var\_user\_role = 'Conducente' then

set var\_role = 1;

select CF

from Conducente

where Username = var\_username

into var\_CF;

elseif var\_user\_role = 'Gestore' then

set var\_role = 2;

elseif var\_user\_role = 'Viaggiatore' then

set var\_role = 3;

else

set var\_role = 0;

end if;

END

* Sono presenti 5 stored procedure utilizzate dal sistema per la gestione dei GPS nei veicoli e la sincronizzazione della loro posizione. La stored procedure attiva\_gps mi “attiva” il GPS delle corse odierne, ovvero prelevo le corse e le istanzio sull’applicazione. Si decide di implementarlo così per semplicità per testare il codice, ma l’idea è quella di “attivare” il GPS dei veicoli una volta raggiunto il loro orario di partenza presente nella base di dati. Successivamente, una volta raggiunto l’orario, si confrontano le coordinate del veicolo con le coordinate del capolinea di partenza. Questo perché è possibile che il veicolo stia ancora effettuando una corsa precedente oppure è in ritardo. Appena si verifica, si aggiorna Partito presente nell’entità Veicolo in Corsa. Infine, prelevo la sequenza ordinata di fermate da effettuare: mi serve a tenere traccia del veicolo e modificare l’Ultima Fermata fatta per quella corsa. Una volta raggiunto il Capolinea di destinazione, il GPS per quella corsa smette di funzionare.

OPERAZIONE S1

CREATE PROCEDURE `attiva\_gps` ()

BEGIN

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level read repeatable read;

set transaction read only;

start transaction;

select Giorno, Ora, Veicolo, Tratta, Conducente

from VeicoloinCorsa

where Giorno = dayname(now());

commit;

END

OPERAZIONE S2

REATE PROCEDURE `preleva\_fermata` (in var\_idtratta tinyint(3))

BEGIN

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level serializable;

set transaction read only;

start transaction;

select Codice, Longitudine, Latitudine

from Effettua join Fermata on Fermata\_Codice = Codice

where Tratta\_ID = var\_idtratta

order by Indice;

commit;

END

OPERAZIONE S3

CREATE PROCEDURE `preleva\_capolinea` (in var\_idtratta tinyint(3))

BEGIN

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level read uncommitted;

set transaction read only;

start transaction;

select Codice, Longitudine, Latitudine

from Tratta join Fermata on Capolinea\_Da = Codice

where ID = var\_idtratta;

commit;

END

OPERAZIONE S4

CREATE PROCEDURE `aggiorna\_partito` (in var\_giorno varchar(45), in var\_ora time, in var\_veicolo int)

BEGIN

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level read uncommitted;

start transaction;

update VeicoloinCorsa

set Partito = 1

where Giorno = var\_giorno and Ora = var\_ora and Veicolo = var\_veicolo;

commit;

END

OPERAZIONE S5

CREATE PROCEDURE `aggiorna\_indice` (in var\_giorno varchar(45), in var\_ora time, in var\_veicolo int, in var\_index int)

BEGIN

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level read uncommitted;

start transaction;

update VeicoloinCorsa

set UltimeFermata = var\_index

where Giorno = var\_giorno and Ora = var\_ora and Veicolo = var\_veicolo and Partito = 1;

commit;

END

1. Indicare con E le entità, con R le relazioni [↑](#footnote-ref-1)
2. [↑](#footnote-ref-2)
3. PK = primary key, NN = not null, UQ = unique, UN = unsigned, AI = auto increment. È ovviamente possibile specificare più di un attributo per ciascuna colonna. [↑](#footnote-ref-3)
4. IDX = index, UQ = unique, FT = full text, PR = primary. [↑](#footnote-ref-4)