

# ITALY EXPRESS RAIL

FEDERICO PERIN 1170747

a.a 2018/2019

## 1 ABSTRACT

La "Italy Express Rail" (abbreviato "IER") è una nuova società italiana recentemente fondata da un gruppo di imprenditori italiani e stranieri operanti in diversi settori dell'economia. Tali imprenditori di comune accordo hanno deciso di formare una nuova società ferroviaria per investire nel settore del trasporto passeggeri su rotaie in Italia. Recentemente la IER è riuscita ad ottenere la concessione di utilizzo della rete ferroviaria italiana, e quindi la possibilità di operare nel settore interessato.

Si necessita perciò di una realizzazione di una base di dati allo scopo di gestire le varie informazioni della società, le quali comprendono la gestione dei dati di dipendenti e clienti, la gestione dei servizi di trasporto ferroviario offerti e infine la gestione del parco treni a disposizione della società.

La base di dati inoltre deve fornire funzionalità di localizzazione dei treni in transito e per ogni stazione, fornire le informazioni riguardanti i treni in arrivo e in partenza per permettere di informare i pendolari che vogliono usufruire del servizio di trasporto offerto dalla società, in più per alcune particolari categorie di treni viene fornita la possibilità per i clienti di prenotare un posto nel treno.

## 2 ANALISI DEI REQUISITI

### DESCRIZIONE TESTUALE

Il progetto in esame vuole rappresentare una base di dati che permette di gestire i dati dei clienti e dipendenti della società ferroviaria, il parco treni a disposizione e la prenotazione di posti treno in alcune particolari tipologie di treno.

Ciascuna **persona** viene identificata dalle seguenti informazioni:

- Codice fiscale è univoco quindi identifica la persona
- Nome
- Cognome
- Dati nascita
  - Città di nascita
  - Data di nascita
- Recapito Telefonico
- Sesso
- E-mail
- Residenza
  - Città di residenza
  - Via di residenza

La base di dati della società per rappresentare correttamente le informazioni, distingue persone in due categorie:

- **Dipendenti:** persona che lavora all'interno della società, a cui viene attribuito un numero del Badge.
- **Cliente:** persona che usufruisce dei servizi offerti dalla società, che dispone di un attributo numero servizi (acquistati)

A sua volta ogni **dipendente** viene classificato in base alle sue mansioni, e quindi si avranno:

- **Operaio:** il dipendente che si occupa di eseguire la manutenzione del treno

- **Controllore:** il dipendente che si occupa di controllare i biglietti dei passeggeri presenti nel treno
- **Macchinista:** il dipendente che si occupa di guidare il treno, in particolare per lui viene registrata l'anno di scadenza della patente speciale per condurre i treni

Di fondamentale importanza è tenere le informazioni del **servizio** di trasporto acquistato dai clienti, quindi si memorizza:

- Codice, univoco e mi permette di identificare il servizio acquistato dal cliente
- Costo per chilometro
- Numero chilometri percorsi
- Prezzo

Ogni **servizio** viene distinto in due categorie:

- **Biglietto:** servizio che ha una validità di un giorno, quindi ne viene salvato il relativo giorno di validità
- **Abbonamento:** servizio che ha una durata maggiore di un giorno perciò viene salvato:
  - Data inizio: l'inizio della validità dell'abbonamento
  - Data fine: la fine della validità dell'abbonamento

Nella base di dati per gestire il parco treni a disposizione della società memorizza una rappresentazione logica di **treno** utilizzata per sapere che tratta deve seguire, sapere in che punto della tratta si trova e sapere da quali componenti (carrozze e motrici) è composto il **treno**. Come attributo proprio avrà un ID che lo identifica.

Per ogni **treno** come detto è formato da varie componenti ognuna identificata attraverso l'attributo ID, le **componenti treno** si distinguono principalmente in:

- **Motrice:** mezzo fisico che traina il treno, viene memorizzato:
  - Modello
  - Tipo motore, cioè se il motore è "elettrico" o a "diesel"
- **Carrozza:** mezzo fisico che trasporta i passeggeri, viene memorizzato:
  - Posti, il numero totale di posti che contiene la carrozza

L'attributo "Posti" è di interesse solo per una particolare **tipologia** di carrozze.

Le componenti del treno possono subire dei guasti e quindi prevista la memorizzazione della **manutenzione** eseguita su di esse. Per ogni **manutenzione** si ha:

- Data, attraverso essa e l'ID della componente su cui è stata eseguita la manutenzione si identifica la manutenzione
- Descrizione, la descrizione del lavoro che è stato svolto

La società prevede che i propri treni siano classificati in **tipologie** per differenziare la qualità dei servizi offerti, ogni tipologia viene identificata attraverso un ID e classificata una delle due categorie ammesse:

- **Regionale:** per offrire un trasporto a basso costo
- **Alta velocità:** per offrire un trasporto di qualità superiore ad un costo maggiore

Come accennato precedentemente anche **carrozza** prevede una classificazione simile a **treno**, nel caso in cui carrozza sia di tipologia alta velocità è possibile permettere al cliente di prenotare un posto e quindi di tenere conto dell'attributo "Posti occupati", nel caso sia di tipologia regionale allora questa funzione non è prevista.

È fondamentale sapere che percorso deve fare il treno perciò viene memorizzata la **tratta** che deve essere percorsa. Per ogni **tratta** si vuole sapere:

- ID, univoco che identifica la tratta
- Partenza prevista, contiene l'orario di partenza previsto dalla stazione di partenza della tratta, cioè la prima stazione della tratta
- Arrivo previsto, contiene l'orario d'arrivo previsto alla stazione di capolinea della tratta, cioè l'ultima stazione della tratta

- Binario di partenza, contiene il numero del binario della stazione di partenza della tratta, cioè la prima stazione della tratta
- Binario di arrivo, contiene il numero del binario della stazione di capolinea della tratta, cioè l'ultima stazione della tratta
- Straordinario, utilizzato per indicare che la tratta è stata aggiunta per un particolare evento ed è quindi presente per un determinato periodo
- Data inizio, utilizzato per indicare che la tratta è stata aggiunta per un particolare evento ed è quindi presente per un determinato periodo, perciò "data inizio" indica l'inizio del periodo di validità della tratta
- Data fine, utilizzato per indicare che la tratta è stata aggiunta per un particolare evento ed è quindi presente per un determinato periodo, perciò "data fine" indica la fine del periodo di validità della tratta

Per ogni **tratta** c'è bisogno di sapere in quali stazioni ne fanno parte, verranno perciò salvati i dati delle stazioni. Per ogni **stazione** si ha:

- Nome, univoco mi identifica la stazione
- Città
- Indirizzo
- Numero binari, viene indicato il numero totale di binari che ha una stazione

La base di dati prevede che ci sia una funzionalità dedicata alla localizzazione dei treni in movimento, in poche parole sapere in che punto della tratta si trovano, grazie a ciò si può stimare l'eventuale ritardo che il treno possa avere.

Tale funzionalità si basa sul fatto che siano installati in punti strategici (nel caso nostro tra due stazioni della tratta) dei dispositivi che al passaggio del treno segnalino tale evento. Tutto ciò viene gestito dal database attraverso l'entità **checkpoint**, ogni **checkpoint** avrà:

- codice identificativo
- nome.

## GLOSSARIO DEI TERMINI

Termine	Descrizione	Sinonimi	Collegamenti
Cliente	Persona che usufruisce dei servizi offerti dalla società	Persona	Servizio
Dipendente	Persona che lavora all'interno della società che possono ricoprire il ruolo di operaio o di controllore o di macchinista, nel caso dei controllori e macchinisti operano per tutta la durata della tratta e possono essere assegnati a più tratte	Persona, Lavoratore	
Treno	entità astratta/logica che rappresenta il treno come insieme di carrozze e motrici avente una tratta da percorrere. In esso possono essere previsti servizi aggiuntivi al passeggero	Convoglio	Tratta, Macchinista, Controllore, Componenti treno, Tipologia, Checkpoint
Servizio	indica i dati relativi alla tariffa del servizio scelto		Cliente
Biglietto	Indica un tipo di servizio acquistato dal cliente, la validità vale solo per il giorno indicato e se è un biglietto per una carrozza alta velocità ne permette la prenotazione di un posto		Carrozza

Abbonamento	indica un tipo di servizio acquistato dal cliente, la validità vale solo per l'intervallo di tempo indicato in esso		
Checkpoint	Indica il dispositivo presente sui binari il quale fornisce le informazioni necessarie per localizzare i treni e saperne l'eventuale ritardo		Stazione, Treno
Localizzazione	Indica ritracciare il treno leggendo l'ultimo checkpoint attivato dal treno		Checkpoint, Treno

### 3 PROGETTAZIONE CONCETTUALE

#### DIZIONARIO DEI DATI (entità)

Entità	Descrizione	Attributi
Persona	Vengono memorizzati i dati delle persone coinvolte nella realtà rappresentata dal database	Codice Fiscale(PK), sesso, nome, cognome, data di nascita, città di nascita, città di residenza, via di residenza, recapito telefonico,
Cliente	Vengono salvati i dati di una persona che usufruisce dei servizi offerti dalla società	Numero servizi
Dipendente	Vengono salvati i dati di una persona che lavora all'interno della società, a cui viene attribuito un numero del Badge.	Badge
Operaio	Vengono salvati i dati di un dipendente che si occupa di eseguire la manutenzione del treno.	
Controllore	Vengono salvati i dati di un dipendente che si occupa di controllare i biglietti dei passeggeri presenti nel treno.	
Macchinista	Vengono salvati i dati di un dipendente che si occupa di guidare il treno, in particolare per lui viene registrata l'anno di scadenza della patente speciale per condurre i treni.	Anno scadenza patente
Manutenzione	Contiene i dati delle manutenzioni effettuate sulle componenti principali del treno	ID (PK), Data(PK), Descrizione,
Componenti treno	Contiene i dati delle principali componenti del treno	ID(PK)
Motrice	Mantiene i dati relativi alla motrice del treno	Modello, Tipo motore
Carrozza	Mantiene i dati relativi alla carrozza del treno, se è di tipologia "Alta velocità" permette la prenotazione di un posto al cliente	Posti
Tipologia	Contiene le informazioni relative al tipo di carrozza e treno	ID(PK)
Treno	entità astratta/logica del treno che rappresenta il treno come insieme di carrozze e motrici avente una tratta da percorrere. In esso possono essere previsti servizi aggiuntivi al passeggero	ID(PK)
Tratta	Contiene le informazioni relative al percorso che deve essere percorso dal treno	ID(PK), Partenza prevista, Arrivo previsto, Straordinario, Data inizio, Data fine, Binario partenza, Binario arrivo
Stazione	Contiene i dati delle stazioni coinvolte nelle tratte	Nome(PK), Città, Indirizzo, Numero Binari

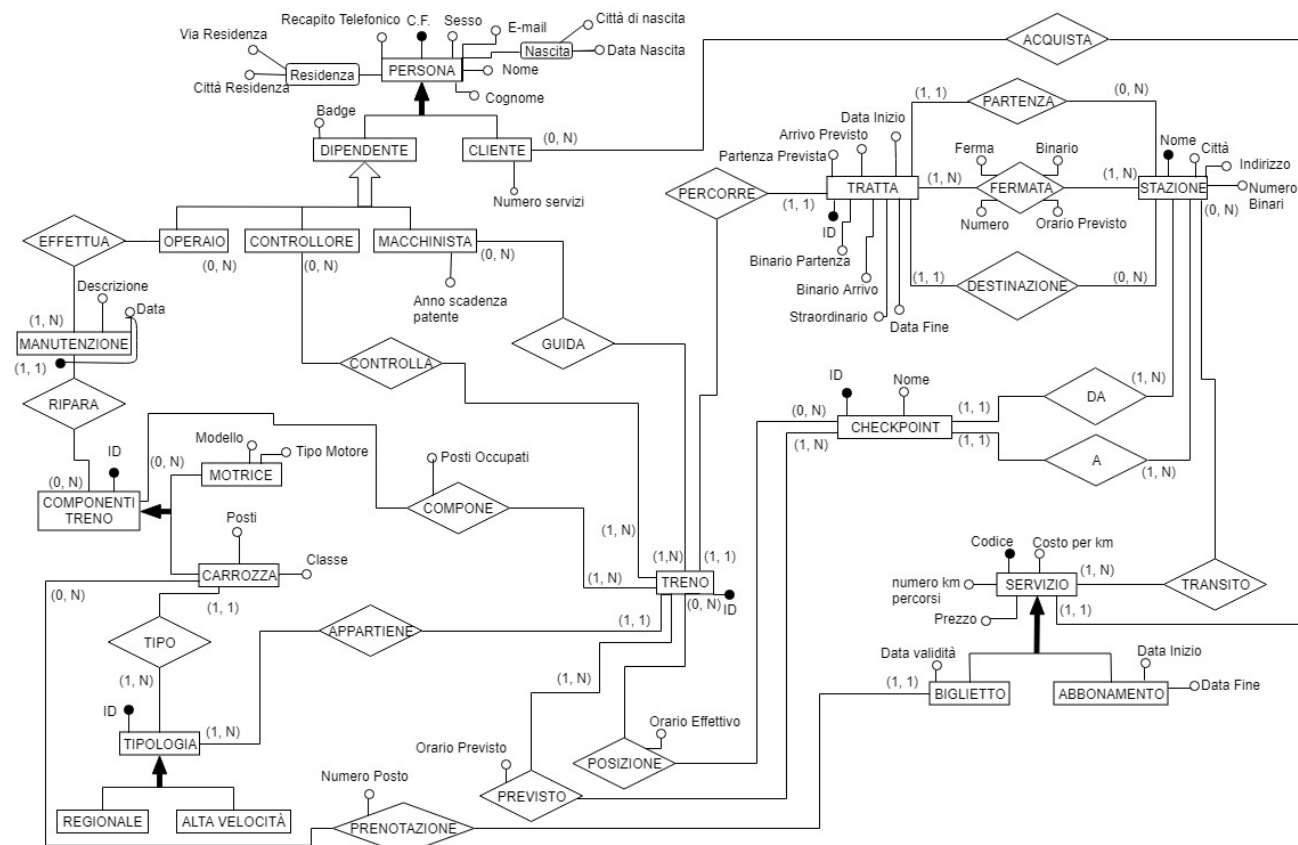
Checkpoint	Contiene i dati relativi a dispositivi di rilevazione del passaggio del treno	ID(PK), km
Servizio	Contiene i dati delle tariffe dei servizi di trasporto	Codice (PK), Costo per km, Numero km percorsi, prezzo
Biglietto	Contiene i dati di validità del biglietto	Data validità
Abbonamento	Contiene i dati di validità dell'abbonamento	Data inizio, Data fine

## DIZIONARIO DEI DATI (Relazioni)

Relazione	Descrizione	Entità coinvolte	Attributi
Acquista	Associa ad un cliente i servizi acquistati	Cliente (0, N) Servizio (1, 1)	
Effettua	Associa gli operai che hanno effettuato le manutenzioni	Operaio (0, N) Manutenzione (1, N)	
Ripara	Associa le manutenzioni fatte alle componenti su cui sono state eseguite	Manutenzione (1, 1) Componenti treno (0, N)	
Compone	Associa le componenti principali di un treno ai treni in momenti diversi	Componenti treno (0, N) Treno (1, N)	Posti occupati
Controlla	Associa più controllori a più treni in orari diversi	Controllore (0, N) Treno (1, N)	
Guida	Associa i macchinisti ai treni che guidano in orari diversi	Macchinista (0, N) Treno (1, N)	
Appartiene	Associa a più treni una tipologia	Treno (1, 1) Tipologia (1, N)	
Tipo	Associa un tipo a più carrozze	Carrozza (1, 1) Tipologia (1, N)	
Prenotazione	Associa una prenotazione a un posto in una carrozza se questa è di tipologia "Alta velocità"	Carrozza (0, N) Biglietto (1, 1)	Numero Posto
Transito	Associa a un servizio acquistato da un cliente le stazioni che può raggiungere	Servizio (1, N) Stazione (0, N)	
Partenza	Associa la stazione di partenza alla tratta cioè la prima stazione della tratta	Tratta (1, 1) Stazione (0, N)	
Destinazione	Associa la stazione di capolinea alla tratta cioè l'ultima stazione della tratta	Tratta (1, 1) Stazione (0, N)	
Fermata	Associa le stazioni coinvolte nella tratta, sia nelle stazioni dove il treno è previsto che si fermi sia dove non si fermi, indicando ciò nell'attributo ferma. Vengono indicati l'orario di transito, il binario, e per tenere una corretta lista di percorrenza della il numero di ordine d'arrivo per ogni stazione	Tratta (1, N) Stazione (1, N)	Ferma, Binario, Numero, Orario previsto
Percorre	Associa una tratta a un treno, la seguente associazione è una uno a uno perché l'entità treno è una rappresentazione logica del convoglio che ha un ID che lo identifica e vale solo per una tratta con cui è legato, quando la "stessa" entità treno (stesse carrozze e stesse motrici) percorrere un'altra tratta (in orario diversi ovviamente) a questa viene assegnato un nuovo ID. Inoltre, ogni tratta è unica perché ogni tratta ha orari diversi perciò solo un treno può percorrere una tratta	Tratta (1, 1) Treno (1, 1)	
Posizione	Associa una posizione di un treno indicando in quale checkpoint si trova, indicando l'ora effettiva del passaggio sopra al checkpoint	Treno (0, N) Checkpoint (0, N)	Ora effettivo
Previsto	Associa la posizione prevista (non effettiva) di un treno su un checkpoint, indicando l'ora prevista del passaggio	Treno (1, N) Checkpoint (1, N)	Ora prevista

Da	Associa il tratto che va dalla stazione di partenza del treno al checkpoint	Checkpoint (1, 1)	
a	Associa il tratto che va dal checkpoint alla stazione di arrivo del treno	Checkpoint (1, 1)	

## SCHEMA E/R



## VINCOLI DI INTEGRITA'

- Un Controllore non può essere assegnato contemporaneamente in più di un treno nello stesso orario ma in orario diversi che comunque non devono sovrapporsi.
- Un Macchinista non può essere assegnato contemporaneamente in più di un treno nello stesso orario ma in orario diversi che comunque non devono sovrapporsi.
- Una Motrice non può essere assegnata contemporaneamente in più di un treno nello stesso orario ma in orario diversi che comunque non devono sovrapporsi.
- Una Carrozza non può essere assegnata contemporaneamente in più di un treno nello stesso orario ma in orario diversi che comunque non devono sovrapporsi.
- L'attributo "Posti occupati" di Compone viene valorizzato se Carrozza fa parte della tipologia "Alta Velocità".
- Viene associata una prenotazione a Carrozza se vale la condizione "Posti > Posti occupati" e che la tipologia sia "Alta Velocità".
- "Data inizio" e "Data Fine" vengono valorizzati se l'attributo "Straordinario" contiene "TRUE" in Tratta ha come dominio di valori "TRUE" e "FALSE".

## 4 PROGETTAZIONE LOGICA

### ANALISI DI RIDONDANZA

Una ridondanza presente nello schema ER è possibile individuarla nella entità **Cliente** nella quale il valore dell'attributo "numero servizi" si può calcolare visitando la relazione Cliente-Servizio (Acquista).

CONCETTO	TIPO	VOLUME
CLIENTI	ENTITA'	20000
SERVIZIO	ENTITA'	100000
ACQUISTA	RELAZIONE	100000

- Operazione 1: Memorizzare un nuovo servizio con il relativo cliente che lo ha acquistato (operazione eseguita 300 volte al giorno)
- Operazione 2: Stampare i dati di tutti i clienti (con il relativo numero di servizi acquistati) (operazione eseguita 3 volte al giorno)

#### Presenza di ridondanza

L'attributo numero servizio indica il numero di servizi acquistati da un cliente, anche quelli non più validi.

##### Operazione 1

CONCETTO	COSTRUTTO	ACCESSI	TIPO
SERVIZIO	ENTITA'	1	S
ACQUISTA	RELAZIONE	1	S
CLIENTI	ENTITA'	1	L
CLIENTI	ENTITA'	1	S

##### Operazione 2

CONCETTO	COSTRUTTO	ACCESSI	TIPO
CLIENTE	ENTITA'	20000	L

Numero accessi:  $3 \cdot 300 \cdot 2 + 300 + 20000 \cdot 3 = 2100 + 60000 = 62100$  accessi al giorno con la ridondanza.

#### Assenza di ridondanza

In questo caso per risalire al numero di servizi acquistati da un cliente bisogna necessariamente fare riferimento alla relazione

##### Operazione 1

CONCETTO	COSTRUTTO	ACCESSI	TIPO
SERVIZIO	ENTITA'	1	S
ACQUISTA	RELAZIONE	1	S

##### Operazione 2

CONCETTO	COSTRUTTO	ACCESSI	TIPO
CLIENTE	ENTITA'	20000	L
ACQUISTA	RELAZIONE	100000	L

Numero accessi:  $2 \cdot 2 \cdot 300 + 20000 \cdot 3 + 100000 \cdot 3 = 1200 + 360000 = 361200$  accessi al giorno senza la ridondanza.

**Conclusione:** La presenza di ridondanza si mostra utile nei casi in cui si necessita sapere quanti servizi ha acquistato ogni cliente. Si è quindi deciso di tenere questo attributo.



## ELIMINAZIONE GENERALIZZAZIONI

La generalizzazione **Persona** ha come entità figlie **Cliente** e **Dipendente**. Gli attributi dell'entità padre sono comuni alle due entità figlie le quali hanno attributi propri e associazioni proprie. La generalizzazione è totale, si è scelto di togliere l'entità padre e di inserire tutti gli attributi del padre nelle entità figlie riducendo il numero di accessi.

La generalizzazione **Dipendente** ha come entità figlie **Operaio**, **Controllore**, **Macchinista**. Gli attributi e le associazioni dell'entità padre sono comuni alle due entità figlie le quali hanno attributi propri e associazioni proprie. La generalizzazione è parziale, c'è bisogno di avere la possibilità di riferirsi sia alla entità padre sia alle entità figlie perciò si è deciso di lasciare le tre entità collegando il padre con le figlie attraverso la creazione di una relazione per ogni figlio.

La generalizzazione **Componenti treno** ha come entità figlie **Motrice**, **Carrozza**. L'entità padre ha come unico attributo la chiave mentre le entità figlie fanno attributi propri. La generalizzazione è totale è c'è bisogno solo di riferirsi alle entità figlie perciò viene tolta l'entità padre e le associazioni legate ad esso vengo legate con le entità figlie, riducendo così il numero di accessi.

La generalizzazione **Tipologia** ha come entità figlie **Regionale** e **Altavelocità**. L'entità padre ha come unico attributo la chiave e le figlie nessun attributo, La generalizzazione è totale perciò si è scelto di tenere solo l'entità padre con l'attributo "Tipo".

La generalizzazione **Servizio** ha come entità figlie **Biglietto** e **Abbonamento**. Gli attributi e le associazioni dell'entità padre sono comuni alle due entità figlie le quali hanno attributi propri e **Biglietto** ha un'associazione propria. La generalizzazione è totale c'è bisogno di avere la possibilità di riferirsi sia alla entità padre sia alle entità figlie perciò si è deciso di lasciare le tre entità collegando il padre con le figlie attraverso la creazione di una relazione per ogni figlio.

## PARTIZIONAMENTO/ACCORPAMENTO DI ENTITA' E RELATIONSHIP

**Tipologia:** l'entità tipologia ha come attributi la chiave più tipo, appare inutile tenere un'entità con così pochi attributi, perciò per risparmiare numero di accessi e memoria si decide di inserire l'attributo tipo nelle entità **Treno** e **Carrozza** le quali erano in relazione con Tipologia.

**Attributi composti Residenza e Nascita:** Gli Attributi composti vengono inseriti nelle due entità Dipendenti e Clienti (gli attributi vengo ereditati dal padre che era stato tolto) , non si è deciso di creare una entità a se e poi metterla in relazione perché si vuole evitare di effettuare troppi accessi.

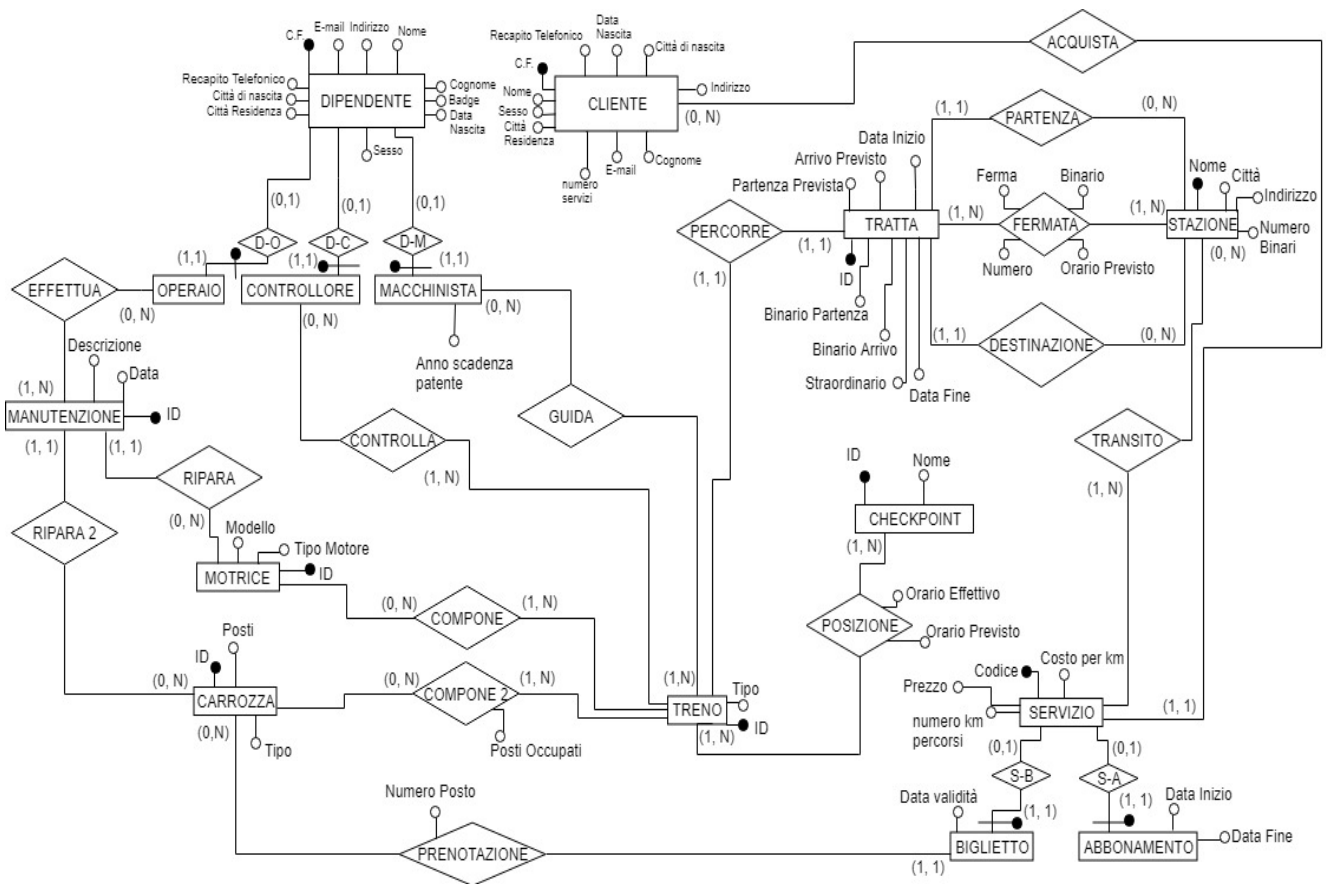
**Previsto:** La relazione Previsto viene accorpata nella relazione posizione riducendo il numero di accessi necessari per calcolare il ritardo di un treno.

**Da e A:** Le relazioni Da e A vengo tolte perché si viene a formare il ciclo **Checkpoint-Posizione-Treno-Percorre-Tratta-Partenza/Fermata/Destinazione-Stazione**, ciò che importa al database è di tenere traccia su quali **Checkpoint Treno** deve passare e quindi **Checkpoint** deve essere legato a **Tratta**. Questa informazione la si può ricavare sia passando per **Stazione** sia passando per **Treno** per ciò ho una ridondanza che va perciò tolta.

**Partenza, Fermata, e Destinazione:** Le relazioni formano un ciclo che però viene mantenuto perché ogni relazione da un'informazione diversa.



## SCHEMA CONCETTUALE RISTRUTTURATO



**Dipendente** (Codice Fiscale, Nome, Cognome, Indirizzo, E-mail, Badge, Data Nascita, Sesso, Città di nascita, Città residenza, Recapito telefonico)

**Clienti** (Codice Fiscale, Nome, Cognome, Indirizzo, E-mail, Numero servizi, Data Nascita, Sesso, Città di nascita, Città residenza, Recapito telefonico)

**Operaio** (Codice Fiscale)

**Controllore** (Codice Fiscale)

**Macchinista** (Codice Fiscale, Anno scadenza patente)

**Effettua** (Operaio, Manutenzione)

**Manutenzione** (ID, Data, Descrizione, Carrozza, Motrice)

**Motrice** (ID, Tipo motore, Modello)

**Carrozza** (ID, Posti, Tipo)

**Compone** (Motrice, Treno) **Compone 2** (Carrozza, Treno, Posti Occupati)

**Controlla** (Controllore, Treno)

**Guida** (Macchinista, Treno)

**Treno** (ID, Tipo, Percorso)

**Posizione** (Treno, Checkpoint, Orario Effettivo, Orario Previsto)

**Checkpoint** (ID, Nome)

**Tratta** (ID, Partenza Prevista, Arrivo Previsto, Data Inizio, Data Fine, Straordinario, Binario partenza, Binario arrivo, Stazione di partenza, Stazione di arrivo)

**Fermata** (Tratta, Stazione, Ferma, Binario, Numero, Orario previsto)

**Stazione** (Nome, Città, Indirizzo, Numero binari)

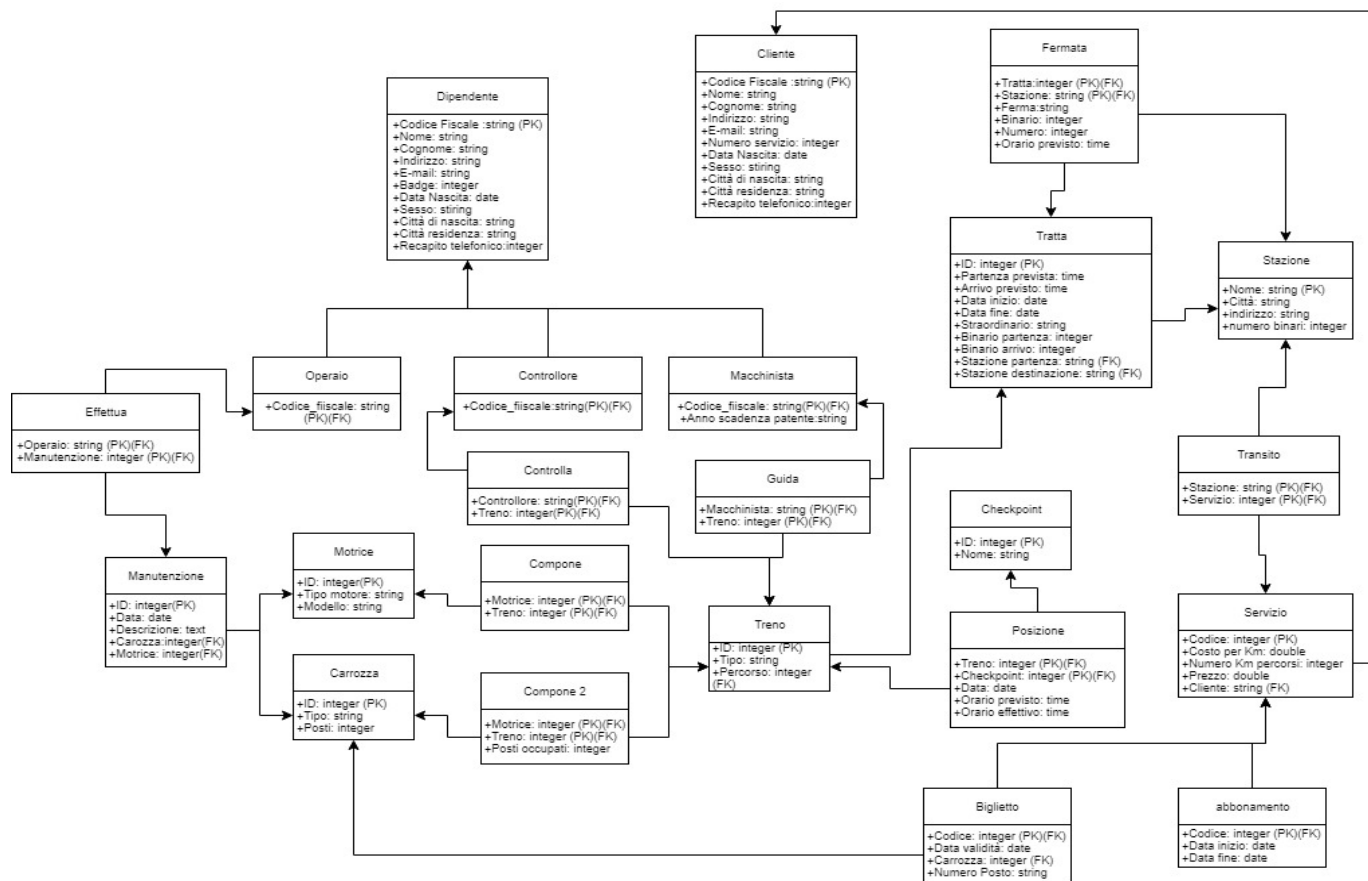
**Transito** (Stazione, Servizio)

**Servizio** (Codice, Prezzo, Costo per km, numero km percorsi, Cliente)

**Biglietto** (Codice, Data validità, Carrozza, Numero Posto)

**Abbonamento** (Codice, Data inizio, Data fine)

## SCHEMA LOGICO



## LISTA DEI VINCOLI DI INTEGRITA' REFERENZIALE

Relazione	Vincoli
Effettua	Operario → Operario.Codice Fiscale Manutenzione → Manutenzione.ID
Operaio	Codice Fiscale → Dipendente.Codice Fiscale
Controllore	Codice Fiscale → Controllore.Codice Fiscale
Macchinista	Codice Fiscale → Macchinista.Codice Fiscale
Manutenzione	Carrozza → Carrozza.ID Motrice → Motrice.ID
Compone	Motrice → Motrice.ID Treno → Treno.ID
Compone 2	Carrozza → Carrozza.ID Treno → Treno.ID
Controlla	Controllore → Controllore.Codice Fiscale Treno → Treno.ID
Guida	Macchinista → Macchinista.Codice Fiscale Treno → Treno.ID
Treno	Percorso → Tratta.ID
Posizione	Treno → Treno.ID Checkpoint → Checkpoint.ID
Tratta	Stazione partenza → Stazione.Nome Stazione arrivo → Stazione.Nome
Fermata	Tratta → Tratta.ID Stazione → Stazione.Nome
Transito	Stazione → Stazione.Nome Servizio → Servizio.Codice
Servizio	Cliente → Cliente.Codice Fiscale
Abbonamento	Codice → Servizio.Codice
Biglietto	Codice → Servizio.Codice Carrozza → Carrozza.ID

## 5 REALIZZAZIONE

Nelle seguenti pagine saranno riportate le viste, le procedure, le funzioni e i trigger (di quest'ultimi i più rilevanti)

### VISTE

- Creare una vista **GrandiStazioni** dove si vuole visualizzare tutte le stazioni che sono inizio di una tratta per 2 o più tratte e che sono capolinea di 3 o più tratte, visualizzare quindi Nome, Città e indirizzo delle stazioni.

```
CREATE VIEW GrandiStazioni AS
SELECT stazione.Nome, stazione.Citta, stazione.Indirizzo
FROM stazione
WHERE stazione.Nome IN(SELECT stazione.Nome
                        FROM stazione JOIN tratta ON stazione.Nome=tratta.Stazione_Partenza
                        GROUP BY stazione.Nome
                        HAVING COUNT(*)>=2)
AND stazione.Nome IN(SELECT stazione.Nome
                      FROM stazione JOIN tratta ON stazione.Nome=tratta.Stazione_Arrivo
                      GROUP BY stazione.Nome
                      HAVING COUNT(*)>=3)
```

Nome	Citta	Indirizzo
Venezia S. Lucia	Venezia	Viale Stazione

- Creare una vista **ElencoOperai** dove si visualizza un elenco ordinato di operai (si mostri codice fiscale, nome, cognome recapito telefonico e e-mail) che hanno effettuato una manutenzione di tipo "Saldatura motore" sulle motrici che sono operative sulle tratte che comprendo una o più stazioni di Milano (per esempio Milano Centrale, Milano Porta Garibaldi ecc.), si visualizzi in oltre il codice della tratta dove operano queste motrici.

```
CREATE VIEW ElencoOperai AS
SELECT dipendente.Codice_Fiscale, dipendente.Nome, dipendente.Cognome, dipendente.Recapito_Telefonico, dipendente.Email, Tratta.ID AS Tratta
FROM (((((dipendente JOIN operaio ON dipendente.Codice_Fiscale=operaio.Codice_Fiscale) JOIN effettua ON operaio.Codice_Fiscale=effettua.Operaio) JOIN
manutenzione ON effettua.Manutenzione=manutenzione.ID) JOIN motrice ON manutenzione.Motrice=motrice.ID) JOIN componel ON motrice.ID=componel.Motrice) JOIN
treno ON componel.Treno=treno.ID) JOIN tratta ON treno.Percorso=tratta.ID
WHERE manutenzione.Descrizione='Saldatura motore' AND (tratta.ID IN (SELECT tratta.ID FROM tratta WHERE tratta.Stazione_Partenza LIKE 'Milano%') OR
tratta.ID IN (SELECT tratta.ID FROM tratta WHERE tratta.Stazione_Arrivo LIKE 'Milano%') OR
tratta.ID IN (SELECT tratta.ID FROM tratta JOIN fermata ON tratta.ID=fermata.Tratta WHERE fermata.Stazione LIKE 'Milano%'))
ORDER BY dipendente.Cognome, dipendente.Nome
```

Codice_Fiscale	Nome	Cognome	Recapito_Telefonico	Email	Tratta
BJ99655077	Rosaria	Milanesi	3247957547	RosariaMilanesi@rhyta.com	40004890
BJ99655077	Rosaria	Milanesi	3247957547	RosariaMilanesi@rhyta.com	40115801
WO33356240	Aloisia	Russo	3942178251	AloisiaRusso@outlook.it	40004890
WO33356240	Aloisia	Russo	3942178251	AloisiaRusso@outlook.it	40115801

- Creare una vista **StraordinarioT** dove si vuole visualizzare il percorso delle tratte straordinarie, l'ID della tratta, il numero di checkpoint che fanno parte della tratta e la data di inizio e fine della tratta straordinaria, si utilizzino, per calcolare il numero di check point una vista ausiliaria(App1) e una funzione per stampare il percorso della tratta (la funzione verrà riportata nelle pagine successive).

```
CREATE VIEW App1 AS
```

```
SELECT tratta.ID AS TID, COUNT(posizione.Check_Point) AS Numero_di_checkpoint
FROM (tratta JOIN treno ON tratta.ID=treno.Percorso) JOIN posizione ON treno.ID=posizione.Treno
GROUP BY TID
```

```
CREATE VIEW StraordinarioT AS
```

```
SELECT tratta.ID, tratta.Data_Inizio, tratta.Data_Fine, appl.Numero_di_checkpoint, tratta.Stazione_Partenza, tratta.Partenza_Previsto,
PercorsoTratta(1, tratta.ID) AS Percorso, tratta.Stazione_Arrivo, tratta.Arrivo_Previsto
FROM tratta JOIN appl ON tratta.ID = appl.TID
WHERE tratta.Straordinario=True'
```

ID	Data_Inizio	Data_Fine	Numero_di_checkpoint	Stazione_Partenza	Partenza_Previsto	Percorso	Stazione_Arrivo	Arrivo_Previsto
30000000	2019-05-01	2019-07-31	2	Padova	13:02:00	Venezia-Mestre 13:20:00	Venezia S. Lucia	13:28:00

## PROCEDURE

- Creare una procedura **Tabellone** che dato il nome di una stazione e 2 estremi di un intervallo temporale, mostri un tabellone ferroviario di tutti i treni in arrivo nella stazione indicata in input e nell'intervallo temporale scelto, si mostri quindi ID e tipo del treno, il binario dove è in arrivo, la destinazione, il ritardo espresso in minuti e il percorso rimanente della tratta (nella procedura per il calcolo del ritardo e del percorso rimanente vengono usate funzioni che saranno presentate in seguito).

*Call Tabellone('Venezia-Mestre', '07:00:00', '08:00:00');*

```
CREATE PROCEDURE `Tabellone`(`NomeS` VARCHAR(100), `Tfiltro1` TIME, `Tfiltro2` TIME)
```

```
BEGIN
```

```
SELECT treno.Tipo, treno.ID, tratta.Stazione_Arrivo, CalRitardoTreno(treno.ID) as ritardo, fermata.Binario,
PercorsoTratta(fermata.Numero, fermata.Tratta) as Percorso
```

```
FROM (fermata JOIN tratta ON fermata.Tratta=tratta.ID) JOIN treno ON tratta.ID=treno.Percorso
```

```
WHERE fermata.Stazione=NomeS AND fermata.Ferma=True' AND fermata.Orario_previsto BETWEEN Tfiltro1 AND Tfiltro2;
```

```
END
```

Tipo	ID	Stazione_Arrivo	ritardo	Binario	Percorso
Regionale	172466360	Udine	5	8	Venezia-Mestre 07:14:00 Mogliano Veneto 07:25:00 Treviso Centrale 07:36:00 Spresiano 07:48:00 Cone
AltaVelocita	595563421	Venezia S. Lucia	11	4	Venezia-Mestre 07:49:00

- Creare una procedura **InfoTreni** dove si vuole visualizzare per ogni treno di un tipo scelto in Input, il tipo, il suo ID il numero di carrozze che lo compone e il numero di manutenzioni che hanno avuto le carrozze e le motrici.

*Call InfoTreni('Regionale');*

```
CREATE PROCEDURE `InfoTreni`(`tipo` VARCHAR(250))
BEGIN

    CREATE TEMPORARY TABLE IF NOT EXISTS NCarr AS /*calcolo quante carrozze ha ogni treno*/
    SELECT compone2.Treno AS C2T, COUNT(compone2.Carrozze) AS NCarrozze
    FROM compone2 JOIN carrozza ON compone2.Carrozze=carrozze.ID
    GROUP BY C2T;

    CREATE TEMPORARY TABLE IF NOT EXISTS NManC AS /*calcolo quante manutenzioni hanno subito le carrozze di ogni treno*/
    SELECT compone2.Treno AS IDC, COUNT(manutenzione.ID) AS NManCarr
    FROM (compone2 JOIN carrozza ON compone2.Carrozze= carrozza.ID )JOIN manutenzione ON carrozza.ID=manutenzione.Carrozze
    GROUP BY IDC;

    CREATE TEMPORARY TABLE IF NOT EXISTS NManM AS /*calcolo quante manutenzioni hanno subito le motrici di ogni treno*/
    SELECT compone1.Treno AS IDM, COUNT(manutenzione.ID) AS NManMotr
    FROM (compone1 JOIN motrice ON compone1.Motrice= motrice.ID )JOIN manutenzione ON motrice.ID=manutenzione.Motrice
    GROUP BY IDM;

    SELECT treno.ID, treno.Tipo, NManC.NManCarr, NManM.NManMotr, NCarr.NCarrozze /* unisco tutto quello che mi sono calcolato in precedenza e lo raggruppo per ogni treno */
    FROM ((treno JOIN NManC ON treno.ID=NManC.IDC) JOIN NCarr ON treno.ID=NCarr.C2T ) JOIN NManM ON treno.ID=NManM.IDM
    WHERE treno.Tipo = tipo /*tipo scelto Input*/
    GROUP BY treno.ID, treno.Tipo, NManC.NManCarr, NManM.NManMotr, NCarr.NCarrozze;

    DROP TABLE NCarr;
    DROP TABLE NManC;
    DROP TABLE NManM;

END
```

ID	Tipo	NManCarr	NManMotr	NCarrozze
271652385	Regionale	1	1	4
371547191	Regionale	2	1	4
686340101	Regionale	1	1	4

- Creare una procedura **ServiziAcquistati** dove si vuole sapere il codice fiscale, nome, cognome, recapito telefonico e E-mail dei clienti che hanno acquistato un numero di servizi maggiore di un valore indicato in input e con un importo totale superiore ad un valore indicato in input, si considerano solo i servizi biglietti validi per il mese di agosto.

*Call ServiziAcquistati (1,12);*

```
CREATE PROCEDURE ServiziAcquistati(Nserv INT(10), Imp FLOAT)
BEGIN

    CREATE TEMPORARY TABLE IF NOT EXISTS VistaApp1 AS
    SELECT cliente.Codice_Fiscale AS CF, SUM(servizio.Prezzo) AS importo
    FROM (cliente JOIN servizio ON cliente.Codice_Fiscale=servizio.Cliente)JOIN biglietto ON servizio.Codice=biglietto.Codice
    WHERE MONTH(biglietto.Data_Validita)='08'
    GROUP BY cliente.Codice_Fiscale
    HAVING importo>Imp;

    SELECT cliente.Codice_Fiscale, cliente.Nome, cliente.Cognome, cliente.Recapito_Telefonico, cliente.Email
    FROM cliente JOIN servizio ON cliente.Codice_Fiscale=servizio.Cliente
    WHERE cliente.Numero_Servizi>Nserv AND cliente.Codice_Fiscale IN (SELECT CF FROM VistaApp1)
    GROUP BY cliente.Codice_Fiscale, cliente.Nome, cliente.Cognome, cliente.Recapito_Telefonico, cliente.Email
    HAVING COUNT(*)>Nserv;

    DROP TABLE VistaApp1;

END
```

Codice_Fiscale	Nome	Cognome	Recapito_Telefonico	Email
JY72772979	Gisella	Pisani	3393222831	GisellaPisani@jourrapide.com
WO33356241	Gavino	Udinesi	3766783708	GavinoUdinesi@rhyta.com

## FUNZIONI

- Creare la funzione **CalRitardoTreno** che dato in input l'ID di un treno si calcoli il suo ritardo in minuti.

```
CREATE FUNCTION `CalRitardoTreno`(`idtr` INT(11)) RETURNS INT(100)
BEGIN
    DECLARE diff INT(100);
    DECLARE UltOE TIME;
    DECLARE UltOP TIME;
    SET diff = 0;

    SELECT MAX(posizione.Orario_Effettivo) into UltOE /*seleziona l'ultimo orario effettivo registrato*/
    FROM posizione
    WHERE idtr=posizione.Treno;

    SELECT posizione.Orario_Previsto into UltOP /*seleziona il corrispondente orario previsto rispetto all'ultimo orario effettivo*/
    FROM posizione
    WHERE idtr=posizione.Treno AND UltOE=posizione.Orario_Effettivo;

    IF(UltOE>UltOP)
    THEN
        SET diff = TIMESTAMPDIFF(MINUTE,UltOP,UltOE); /*calcola il ritardo*/
    END IF;

    RETURN diff;
END
```

- Creare la funzione **PercorsoTratta** che dato il numero progressivo di una stazione all'interno di una tratta e l'ID della tratta, calcoli il percorso rimanente della tratta indicata in input.

```
CREATE FUNCTION `PercorsoTratta`(`Npart` INT(100), `IDTratta` INT) RETURNS text CHARSET latin1
BEGIN
    DECLARE Nferm INT(100);
    DECLARE str varchar(100);
    DECLARE staz varchar(100);
    DECLARE orario TIME;
    DECLARE i INT(100);

    SET str="";
    SET i=Npart;

    SELECT COUNT(fermata.Stazione) INTO Nferm /*calcola il numero di fermate che ha la tratta*/
    FROM fermata
    WHERE fermata.Tratta=IDTratta;

    WHILE i<Nferm+1 DO
        SELECT fermata.Stazione, fermata.Orario_previsto INTO staz, orario
        FROM fermata
        WHERE fermata.Tratta = IDTratta AND fermata.Numero = i;
        SET i = i + 1;
        SET str = CONCAT(str, ",",staz, ",", orario); /*concatena il nome e l'orario della fermata*/
    END WHILE;

    RETURN str;
END
```



## TRIGGER

- Creare un trigger che controlli che il controllore si liberi nell'intervallo di tempo dato dalla durata della tratta, quindi si deve controllare che il controllore non sia assegnato su due treni i cui tempi di durata si sovrappongono (il controllore non deve essere in due treni o più nello stesso momento).

```
CREATE TRIGGER `OrarioLibero` BEFORE INSERT ON `controlla` FOR EACH ROW BEGIN

    DECLARE pp TIME;
    DECLARE ap TIME;
    DECLARE flag INT(100);
    DECLARE ErrorCode CONDITION FOR SQLSTATE '45000';
    SET @err_mess = 'inserimento non valido orario già assegnato';
    SET flag = 0;

    SELECT tratta.Partenza_Previsto, tratta.Arrivo_Previsto INTO pp, ap /*estraggo l'intervallo di tempo dove il controllore dovrebbe lavorare*/
    FROM (controlla JOIN treno ON NEW.Treno=treno.ID) JOIN tratta ON treno.Percorso=tratta.ID LIMIT 1;

    SELECT COUNT(controlla.Treno) AS n INTO flag /*controllo se il controllore lavora già nel periodo temporale ricavato in precedenza*/
    FROM (controlla JOIN treno ON controlla.Treno=treno.ID) JOIN tratta ON treno.Percorso=tratta.ID
    WHERE controlla.Controllore=NEW.Controllore AND
    (pp BETWEEN tratta.Partenza_Previsto AND tratta.Arrivo_Previsto OR ap BETWEEN tratta.Partenza_Previsto AND tratta.Arrivo_Previsto);

    IF(flag>0)
    THEN
        SIGNAL ErrorCode SET MESSAGE_TEXT=@err_mess;
    END IF;
END
```

- Creare un trigger che controlli se è possibile prenotare un posto cioè la carrozza e di tipo Alta Velocità e ci sono dei posti liberi. Se la prenotazione va a buon fine si aggiorna il numero di posti occupati dato che si è aggiunta una nuova prenotazione.

```
CREATE TRIGGER `ControlloCreaPrenotazione` BEFORE INSERT ON `biglietto` FOR EACH ROW BEGIN
    DECLARE type varchar(100);
    DECLARE P INT(100);
    DECLARE PO INT(100);
    DECLARE ErrorCode CONDITION FOR SQLSTATE '45000';
    SET @err_mess = 'ERRORE RENOTAZIONE NON VALIDA';
    SET @err_mess2 = 'ERRORE PRENOTAZIONE NON VALIDA POSTI ESAURITI';

    IF(NEW.Carrozza IS NOT NULL)
    THEN
        SELECT carrozza.Tipo INTO type
        FROM carrozza
        WHERE carrozza.ID=new.Carrozza;
        IF(type<>'AltaVelocita')
        THEN
            SIGNAL ErrorCode SET MESSAGE_TEXT=@err_mess; /*controllo se è permesso prenotare*/
            SET NEW.Carrozza=NULL;
        ELSE
            SELECT carrozza.Tipo, carrozza.Posti, compone2.Posti_Occupati INTO type, P, PO /*controllo se ci sono posti liberi*/
            FROM (biglietto JOIN carrozza ON NEW.Carrozza=carrozza.ID) JOIN compone2 ON carrozza.ID=compone2.Carrozza
            WHERE NEW.Codice=biglietto.Codice;

            IF(P=PO)
            THEN
                SIGNAL ErrorCode SET MESSAGE_TEXT=@err_mess2;
                SET NEW.Carrozza=NULL;
            ELSE
                UPDATE compone2 SET compone2.Posti_Occupati=compone2.Posti_Occupati+1/*aggiorno il numero dei posti occupati*/
                WHERE compone2.Carrozza=NEW.Carrozza;
            END IF;
        END IF;
    END IF;
END
```