

CORSO DI LAUREA IN INGEGNERIA E SCIENZE INFORMATICHE  
A.A. 2024/25

## Smart City - Mobilità Integrata

Relazione per il corso di Basi di Dati



**Componenti del gruppo nr.2743:**  
Bartocetti Enrico, matr. 0001115097  
Benedetti Nicholas, matr. 0001114021  
Tazzieri Nicolas, matr. 0001114078

# Indice

<b>1 Analisi dei requisiti</b>	<b>3</b>
1.1 Intervista iniziale . . . . .	3
1.2 Prima fase di analisi . . . . .	4
1.3 Concetti principali . . . . .	5
<b>2 Progettazione Concettuale</b>	<b>7</b>
2.1 Schema Scheletro . . . . .	7
2.2 Raffinamenti Proposti . . . . .	8
2.2.1 Linee e Hub mobilità . . . . .	8
2.2.2 Manutenzioni . . . . .	8
2.2.3 Persone . . . . .	9
2.2.4 Orari Linee e mezzi . . . . .	10
2.2.5 Titolo di Viaggio . . . . .	11
2.3 Schema Concettuale Finale . . . . .	11
2.3.1 Vincoli inespressi . . . . .	13
<b>3 Progettazione Logica</b>	<b>14</b>
3.1 Stima del volume dei dati . . . . .	14
3.2 Descrizione operazioni . . . . .	15
3.3 Analisi delle operazioni . . . . .	16
3.4 Analisi delle ridondanze . . . . .	37
3.4.1 Analisi attributo <b>tempo percorrenza</b> . . . . .	37
3.4.2 Analisi attributo <b>attiva</b> . . . . .	37
3.5 Riepilogo operazioni . . . . .	37
3.6 Raffinamento dello schema . . . . .	38
3.6.1 Rimozione attributi multivалore . . . . .	38
3.6.2 Rimozione gerarchie . . . . .	39
3.6.3 Reificazione associazioni molti a molti . . . . .	41
3.6.4 Scelta degli identificatori principali . . . . .	41
3.7 Schema relazionale finale . . . . .	42
<b>4 Progettazione della Base di Dati</b>	<b>46</b>
4.1 Check . . . . .	46
4.2 Viste . . . . .	46
4.3 Stored Procedures . . . . .	47
4.4 Trigger . . . . .	48
4.5 Eventi . . . . .	49
4.6 Traduzione delle operazioni in query SQL . . . . .	49



# Capitolo 1

## Analisi dei requisiti

### 1.1 Intervista iniziale

Si vuole realizzare il software per la gestione del trasporto pubblico a lungo e corto raggio. A seguito dell'intervista è stata prodotta la seguente descrizione delle specifiche del sistema in linguaggio naturale:

*Le città di Vergneto, Urbania e San Giovanni in Marignano si sono accordate per avviare il progetto “Smart mobility 2030”, e richiedono un software per la gestione Smart e integrata del trasporto pubblico urbano ed extraurbano.*

*Il sistema dovrà gestire diversi mezzi di TPL (Trasporto Pubblico Locale), quali bus, tram, metropolitana e treni ma in futuro potrebbe sorgere la necessità di aggiungerne di nuovi (ad esempio in seguito alla creazione di linee per i filobus). Ogni mezzo potrà percorrere diverse linee: ciò significa che un autobus utilizzato oggi per la linea 15, domani potrà percorrere la linea 21.*

*Per ogni linea verranno effettuate delle fermate a orari prestabiliti. Potranno essere create anche linee straordinarie, ad esempio delle navette per eventi cittadini, di cui dovranno essere specificati periodo di validità, orari e fermate.*

*Inoltre, per ridurre l'inquinamento nelle nostre città, verranno creati degli spazi riservati per lo sharing di biciclette, scooter e biciclette, e verranno aggiunte delle colonnine per la ricarica elettrica in modo da incentivare sia l'acquisto di auto elettriche sia l'utilizzo di monopattini o bici elettriche.*

*Sarà disponibile una piattaforma online, nella quale si potranno visualizzare le diverse linee e le fermate che i mezzi effettuano con i relativi orari, nonché la disponibilità di posti liberi nelle aree di bikesharing e nelle colonnine elettriche. È anche prevista la registrazione di utenti “cittadini”, che potranno acquistare biglietti e abbonamenti.*

*Un unico biglietto vale per tutti i mezzi di trasporto, e si vuole memorizzarne la durata, la data di convalida, la data di acquisto ed il titolare nel caso venga acquistato online: infatti un biglietto potrà anche essere acquistato da un fornitore fisico (ad esempio le tabaccherie della città). Sono presenti diverse tipologie di biglietti che differiscono in base alla durata dalla convalida e al prezzo. Gli abbonamenti sono biglietti che hanno una durata superiore ad un giorno.*

*Si richiede anche la gestione dei dipendenti. In particolare, dovranno essere gestiti diversi autisti a cui verranno assegnati diversi mezzi in diversi slot orari. Sono inoltre presenti i controllori: in certi orari dovranno controllare diverse linee e saranno in grado di emettere multe. Gli orari e le linee da controllare vengono stabilite dal gestore delle*

*linee di trasporto: ad esempio verranno prese in particolare considerazione quelle dove non ci sono molti incassi, ovvero dove è ipotizzabile una maggior evasione tariffaria, generando ingenti perdite economiche. Infine, l'amministratore del sistema potrà registrarsi in un portale specifico nel quale potrà gestire autisti, controllori e le linee. Inoltre, avrà la possibilità di visualizzare diverse statistiche riguardanti il sistema di trasporto, con enfasi particolare nei ricavi dei biglietti e nel tasso di evasione.*

*Dovranno essere gestiti i lavori di manutenzione relativi a linee e mezzi per cui saranno specificati gli eventuali enti privati che li svolgeranno. Per le manutenzioni sarà richiesto di specificare una data di inizio e fine lavori, ed eventuali variazioni di servizio dovranno essere visualizzate al pubblico.*

## 1.2 Prima fase di analisi

Nell'analisi dell'intervista effettuata tempo addietro, abbiamo notato che il committente non è stato abbastanza esaustivo su alcuni argomenti. Si è quindi resa necessaria un'ulteriore intervista per comprendere meglio il funzionamento di:

- Sistema di acquisto e convalida di biglietti
- Gestione degli spazi speciali
- Gestione delle multe

Riportiamo le risposte R alle varie domande D.

**D:** “Come devono essere gestiti i biglietti, dal loro acquisto fino alla convalida?”

**R:** “I biglietti possono essere acquistati tramite il sito web, le biglietterie fisiche (sia dell'azienda sia da altri venditori autorizzati) e ticketmachine. Ogni mezzo sarà dotato di una obliteratrice che, comunicando con il sistema informativo, alla lettura di un biglietto (tramite codice a barre stampato sul biglietto fisico o letto dal proprio dispositivo) ne controlla la validità e in caso affermativo lo registra come convalidato.”

**D:** “Cosa deve essere mostrato riguardo agli spazi per la mobilità sostenibile?”

**R:** “L'idea è quella di far vedere giusto una panoramica riguardo agli spazi. In particolare, basta visualizzarne la posizione, l'indirizzo, un nome simbolico e quanti sono i posti rimanenti (quante biciclette rimanenti se si tratta di biciclette, quante colonnine se si tratta di colonnine elettriche, ecc.). Inoltre, uno spazio può essere legato ad una fermata per rendere agevole il collegamento con aree non raggiunte dalle linee del TPL (es. appena scendo dal bus mi trovo nei pressi di una zona per il bike sharing)”

**D:** “Come funziona il sistema delle multe?”

**R:** “Le multe possono essere emesse dal personale qualificato a seguito di controlli effettuati sulle linee. A ogni causale della multa corrisponde un certo importo base, e un importo massimo. Il controllore potrà assegnare un importo che sia compreso tra questi due valori”

## 1.3 Concetti principali

Per rendere meglio fruibile la descrizione, abbiamo deciso di rimuovere eventuali ambiguità, seguendo la tabella qua sotto.

Termine	Nuovo Termine
Mezzi di TPL	Mezzi di trasporto
Spazi riservati per il carsharing, ...	Hub mobilità
Biglietti, Abbonamenti	Titolo di Viaggio
Gestore delle linee	Amministratore di Sistema

A seguito dell'intervista aggiuntiva è stata redatta la seguente descrizione del dominio in linguaggio naturale. Ambiguità e informazioni ridondanti sono state eliminate per garantire una miglior fruizione della descrizione:

Le città di Vergineto, Urbania e San Giovanni in Marignano si sono accordate per avviare il progetto “Smart mobility 2030”, e richiedono un software per la gestione Smart e integrata del trasporto pubblico urbano ed extraurbano.

Il sistema dovrà gestire diversi mezzi di trasporto, quali bus, tram, metropolitana e treni lasciando la possibilità di aggiungerne altri in futuro. Ogni mezzo di trasporto potrà percorrere diverse linee.

Una linea può essere ordinaria o straordinaria. Ogni linea effettua fermate ad orari prestabiliti. Per una linea straordinaria dovrà essere specificato il periodo di validità della linea.

Dovranno essere gestiti degli hub mobilità, riservati per la ricarica elettrica e lo sharing di bici, monopattini e scooter. In particolare, si devono memorizzare la posizione, l'indirizzo, un nome simbolico e quanti sono i posti rimanenti. Inoltre, ad un hub mobilità può essere associato ad una fermata.

Sarà disponibile una piattaforma online, nella quale si potranno visualizzare le diverse linee e le fermate che i mezzi effettuano con i relativi orari, nonché la disponibilità di posti liberi negli hub mobilità. È anche prevista la registrazione di utenti cittadini, che potranno acquistare titoli di viaggio, che si dividono in biglietti e abbonamenti.

Un unico titolo di viaggio vale per tutti i mezzi di trasporto e vuole memorizzarne la durata, la data di convalida, la data di acquisto e l'utente cittadino associato nel caso sia un biglietto digitale. Sono presenti diverse tipologie di titoli di viaggio che differiscono in base alla durata dalla convalida e al prezzo. Gli abbonamenti sono biglietti che hanno una durata superiore ad un giorno. I biglietti possono essere acquistati anche in biglietterie fisiche e ticketmachine. Ogni mezzo di trasporto sarà dotato di una obliteratrice che, comunicando con il sistema informativo, alla lettura di un biglietto ne controlla la validità, e in caso affermativo, lo registra come convalidato.

Si richiede anche la gestione dei dipendenti. Essi si dividono in Autisti, Controllori e Amministratori di sistema. Agli Autisti a verranno assegnati diversi mezzi di trasporto in slot orari diversi. I controllori dovranno controllare diverse linee negli slot orari assegnati, e saranno in grado di emettere multe. A ogni causale della multa corrisponde un certo importo base e uno massimo. Il controllore potrà assegnare un importo alla multa tra questi due valori. Gli orari e le linee da controllare vengono stabilite dall'amministratore di sistema. Infine, l'amministratore del sistema potrà registrarsi in un portale specifico nel quale potrà gestire autisti, controllori e le linee. Inoltre, avrà

la possibilità di visualizzare diverse statistiche riguardanti il sistema di trasporto, con enfasi particolare nei ricavi dei titoli di viaggio e nel tasso di evasione.

Dovranno essere gestiti i lavori di manutenzione, che possono coinvolgere linee e mezzi di trasporto, per cui saranno specificati gli eventuali enti privati che li svolgeranno. Per le manutenzioni sarà richiesto di specificare una data di inizio e fine lavori, ed eventuali variazioni di servizio che coinvolgono linee dovranno essere visualizzate al pubblico.

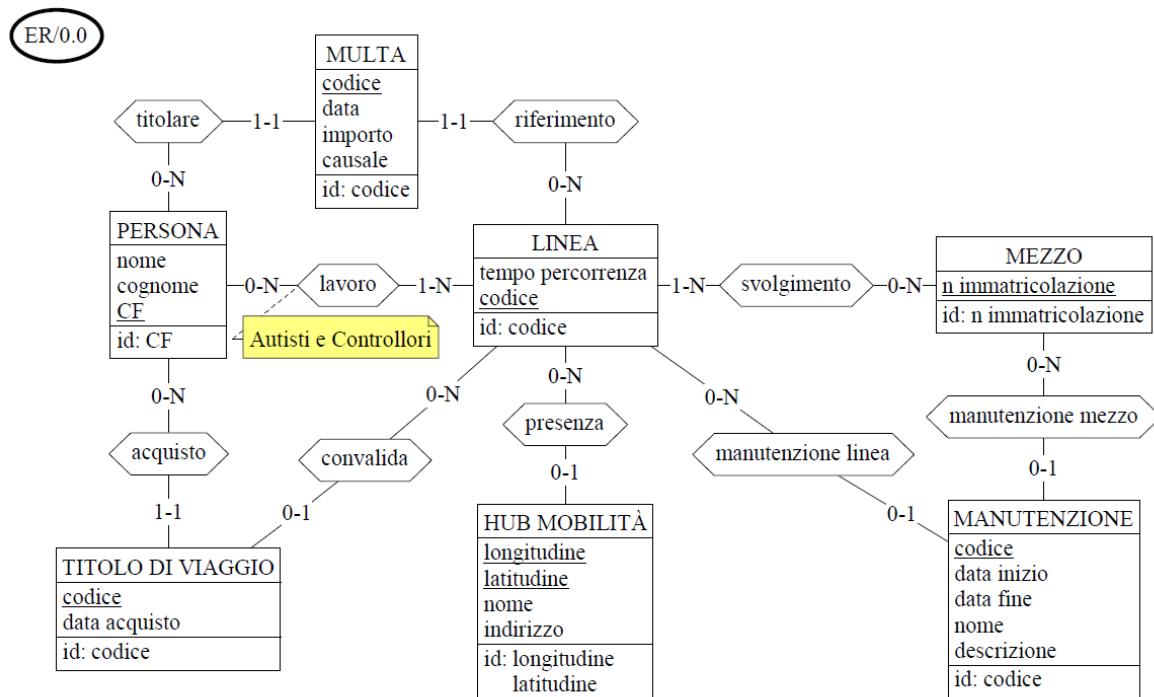
# Capitolo 2

## Progettazione Concettuale

In questo capitolo mostreremo la realizzazione dello schema ER. Partiamo da uno schema scheletro per poi, attraverso degli step di raffinamento, arrivare allo schema finale mantenendo un approccio top-down.

### 2.1 Schema Scheletro

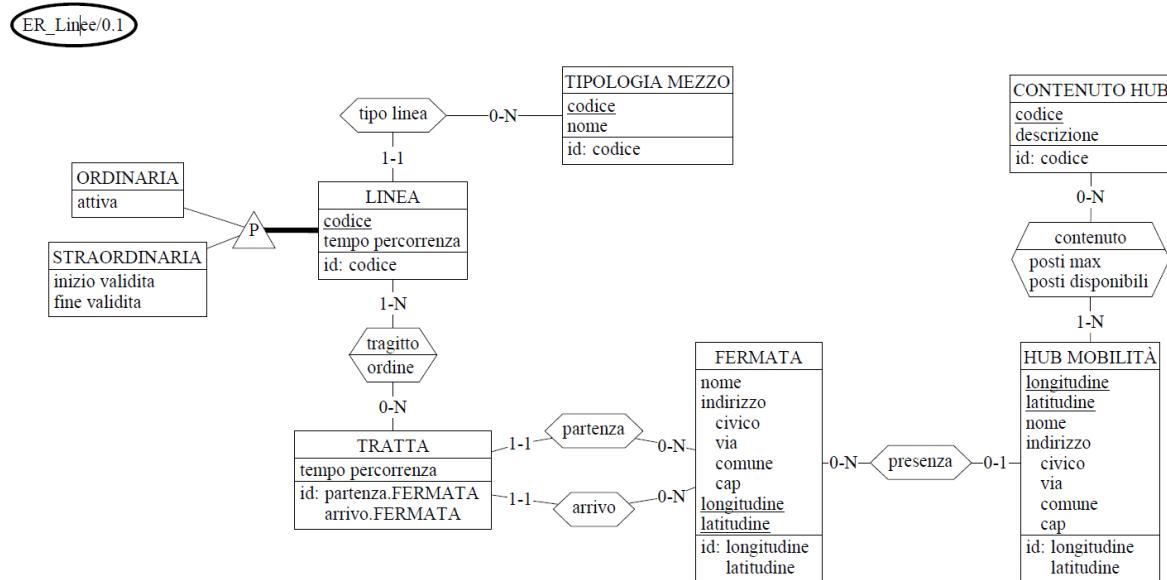
In seguito all'analisi del dominio abbiamo realizzato lo schema scheletro, contenente le principali entità e associazioni che andremo a raffinare nei prossimi step.



## 2.2 Raffinamenti Proposti

### 2.2.1 Linee e Hub mobilità

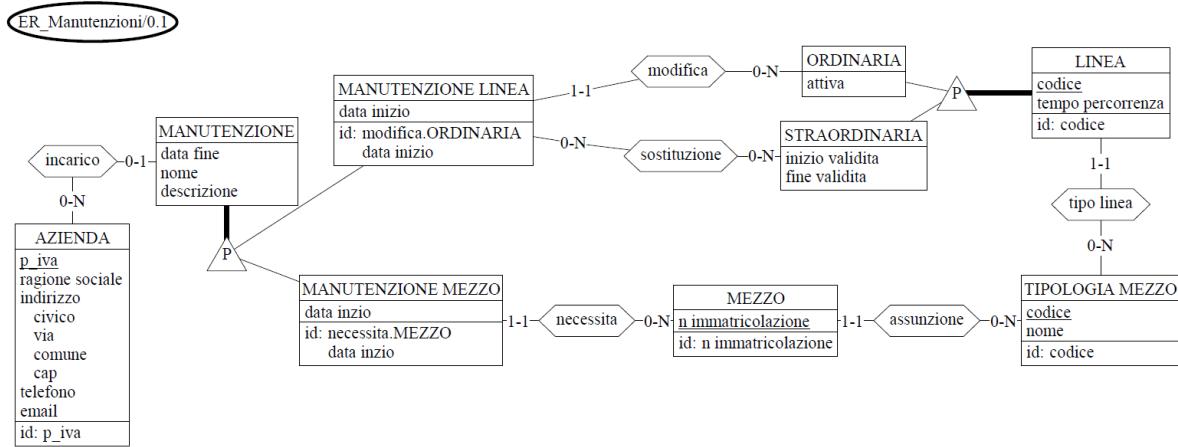
Per modellare le linee abbiamo creato una gerarchia totale ed esclusiva formata da due entità una rappresentante le linee ordinarie, una le linee straordinarie. Per modellare le varie tipologie di mezzi che possono operare in una linea abbiamo aggiunto l'entità TIPOLOGIA MEZZO. L'entità TRATTA modella l'esistenza di un percorso tra due fermate e ne memorizza il tempo di percorrenza. Una linea verrà quindi intesa come sequenza ordinata di tratte attraverso l'associazione TRAGITTO. Per gestire le varie tipologie di contenuti negli hub abbiamo creato l'entità CONTENUTO HUB: associata a uno specifico HUB MOBILITÀ ne memorizza il numero di posti massimi e quelli disponibili. Siccome non ci interessa storicizzare l'utilizzo degli hub mobilità, posti disponibili rimane un attributo dell'associazione contenuto che verrà incrementato / decrementato all'occorrenza (sempre rispettando il limite  $0 \leq \text{disponibili} \leq \text{massimi}$ ). Infine per modellare l'eventuale presenza di una fermata vicino ad un hub abbiamo creato l'associazione opzionale presenza. Con questa modellazione rimangono dei vincoli inespressi da gestire, la cui discussione viene effettuata in section 2.3.



### 2.2.2 Manutenzioni

Nel nostro dominio gli oggetti delle manutenzioni possono essere sia le LINEE che i MEZZI: per questo abbiamo raccolto gli attributi comuni nell'entità MANUTENZIONE e poi creato una gerarchia totale ed esclusiva per diversificare mezzi e linee. Visto che una manutenzione può essere svolta sia internamente sia affidandola a un'AZIENDA terza, abbiamo inserito un'associazione opzionale tra la manutenzione e l'azienda.

Per ridurre al minimo i disservizi, per ogni linea ORDINARIA posta in manutenzione si possono prevedere una serie di linee STRAORDINARIE che la rimpiazzano.



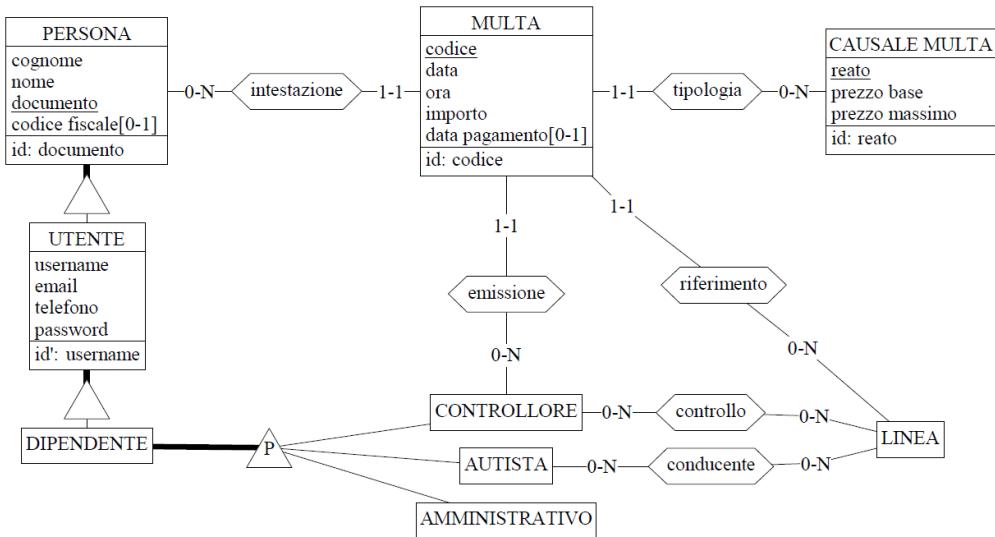
### 2.2.3 Persone

Per la modellazione delle persone abbiamo deciso di creare diversi livelli di gerarchia:

- Persone: Raccoglie i dati anagrafici di una persona
- Utenti: Subset di PERSONA registrate al sistema con username e password
- Dipendenti: Subset di UTENTE che lavorano al alla azienda TPL
- Controllori-Autisti-Amministrativi: Gerarchia totale ed esclusiva per specificare il ruolo del DIPENDENTE all'interno dell'azienda

Grazie a questa modellazione evidenziamo che il controllore potrà controllare alcune linee e un autista potrà condurre diverse linee. Questa modellazione non è ottimale poiché manca sia uno storico delle corse e controlli svolti, sia perché non è chiaro quando questi andranno effettuati: nei prossimi step andremo a risolvere il problema della storicizzazione, così da modellare anche un orario di lavoro. Per quanto riguarda le multe, è stata creata l'entità MULTA associata al controllore che la emette, alla linea nella quale è stata commessa l'infrazione e alla persona che ha commesso il reato. Per gestire l'importo della multa è stata creata l'entità CAUSALE MULTA che dato un tipo di infrazione viene determinato un prezzo base e prezzo massimo alla multa (il cui importo sarà deciso a discrezione del controllore).

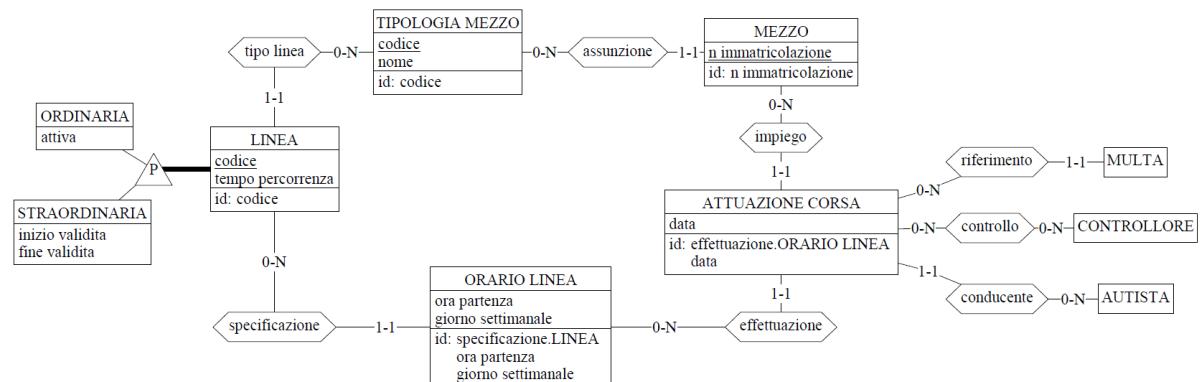
ER\_Persone/0.1



## 2.2.4 Orari Linee e mezzi

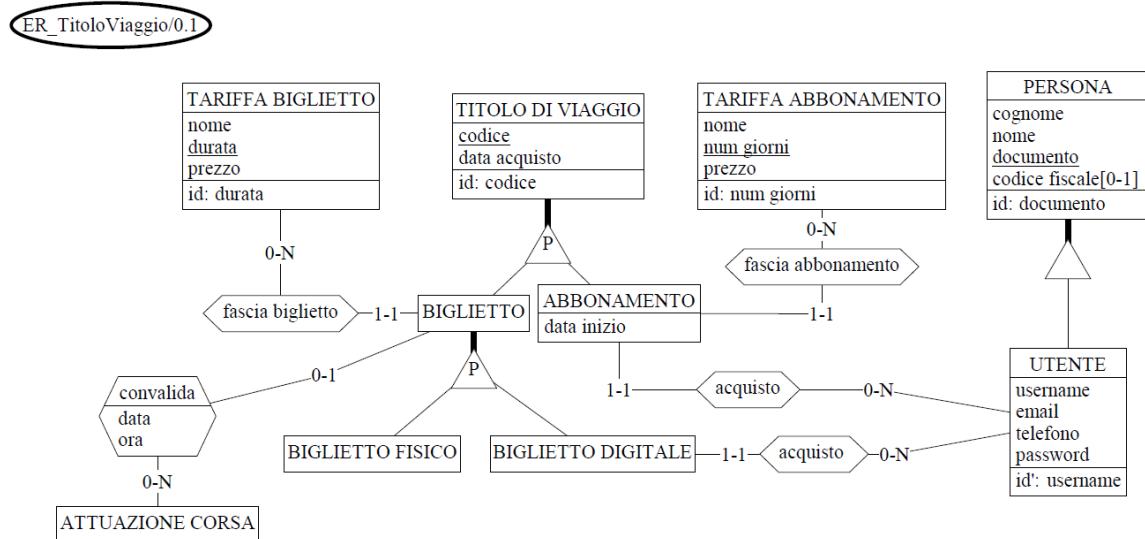
Per la gestione degli orari delle linee abbiamo aggiunto un'entità **ORARIO LINEA** i cui attributi (giorno ed ora) fanno tutti parte della chiave. In questo modo per ogni linea si può specificare un giorno settimanale e più orari di partenza per lo stesso giorno. Per storicizzare le corse abbiamo aggiunto **ATTUAZIONE CORSA** che fa riferimento ad un orario di una linea effettuato in una certa data, da un certo autista, tramite uno specifico mezzo e su cui possono essere assegnati vari controllori (le cui multe saranno associate alla corsa su cui sono state emesse). Va però controllato che il giorno settimanale della data sia lo stesso di quello associato a **ORARIO LINEA**. In questa maniera è stato possibile modellare l'orario di lavoro dei dipenenti, che farà riferimento ad una specifica **ATTUAZIONE CORSA**. Un altro vincolo inespresso da rispettare è che la tipologia del mezzo associato all'attuazione corsa deve essere la stessa associata alla linea.

ER\_AttuazioniLinee/0.1



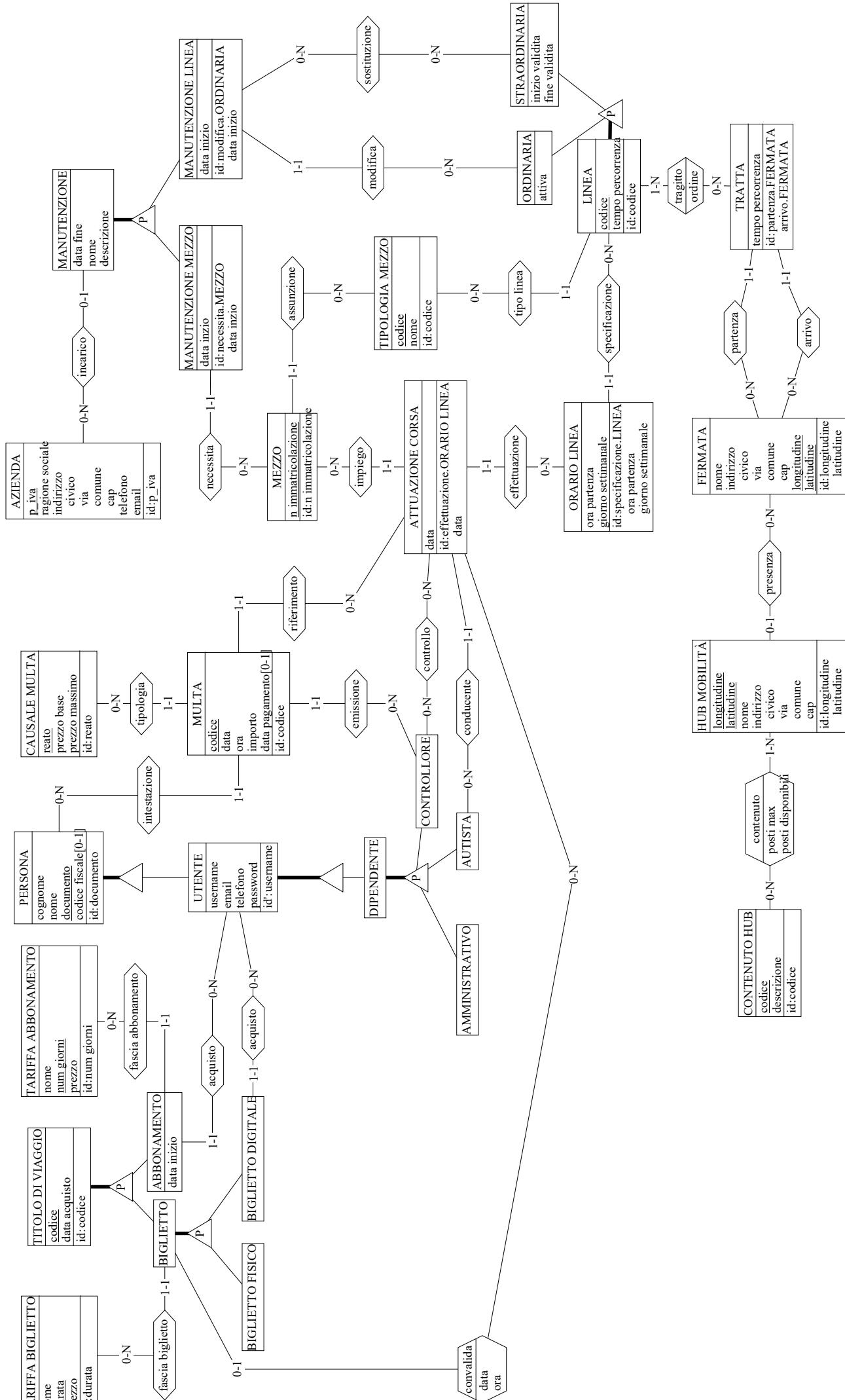
## 2.2.5 Titolo di Viaggio

Per quanto riguarda la gestione dei titoli di viaggio, come da requisiti, abbiamo generalizzato il concetto e creato due sottoentità BIGLIETTO e ABBONAMENTO. Un abbonamento è sempre associato all'utente proprietario del titolo. Per quanto riguarda i tariffari abbiamo creato due entità separate: per i biglietti la durata in minuti definisce il nome e il prezzo, mentre per gli abbonamenti è il numero di giorni di validità a determinarli. Anche il concetto di biglietto è stato generalizzato, creando le sottoentità BIGLIETTO FISICO e BIGLIETTO DIGITALE; quest'ultimo è sempre associato all'utente proprietario del biglietto. Per gestire le convalide abbiamo aggiunto l'associazione opzionale **convalida** con un'attuazione corsa che contiene la data e l'ora della convalida. In questa maniera sarà possibile tenere traccia delle linee in cui verranno effettuate più convalide.



## 2.3 Schema Concettuale Finale

Nella pagina seguente è presente lo schema concettuale finale completo.



### 2.3.1 Vincoli inespressi

In questa sezione discuteremo i vari vincoli inespressi dal nostro schema concettuale:

- In un **TRAGITTO** la tratta  $n$ -esima deve avere la fermata di partenza che coincide con la fermata di arrivo della tratta  $(n - 1)$ -esima
- In una **TRATTA** le fermate di partenza e arrivo devono essere diverse
- In **CONTENUTO**  $\text{posti\_max} \geq \text{posti\_disponibili}$
- In **CAUSALE\_MULTA**  $\text{prezzo\_base} \leq \text{prezzo\_massimo}$
- In **ATTUAZIONE\_CORSA** il giorno settimanale della data deve coincidere con il giorno settimanale in **ORARIO\_LINEA**
- Il mezzo assegnato a un'ATTUAZIONE\_CORSA deve avere la stessa **TIPOLOGIA\_MEZZO** assegnata alla **LINEA** di riferimento
- Nell'aggiunta di un **AUTISTA** a un'ATTUAZIONE\_CORSA bisogna controllare che durante tutta la durata della corsa l'autista non sia già stato assegnato ad altre linee (ovvero che non si sovrapponga l'orario di lavoro)

# Capitolo 3

## Progettazione Logica

### 3.1 Stima del volume dei dati

Per poter gestire al meglio il carico di lavoro della base di dati, insieme al committente si è creata una stima del volume dei dati per entità e associazioni che il database dovrà gestire, di cui se ne riportano i numeri nella Tabella 3.1. La stima è valida per un carico di lavoro di circa 6 mesi.

Tabella 3.1: Stima del volume dei dati

NOME	VOLUME STIMATO	E/A
ABBONAMENTO	30.000	E
AMMINISTRATIVO	10	E
ATTUAZIONE CORSA	250.000	E
AUMENTO	10	E
AUTISTA	130	E
AZIENDA	15	E
BIGLIETTO DIGITALE	150.000	E
BIGLIETTO FISICO	150.000	E
CAUSALE MULTA	10	E
CONTENUTO HUB	4	E
CONTROLLORE	40	E
DIPENDENTE	180	E
FERMATA	450	E
HUB MOBILITÀ	225	E
LINEA	75	E
MANUTENZIONE	50	E
MANUTENZIONE LINEA	10	E
MANUTENZIONE MEZZO	40	E
MEZZO	160	E
MULTA	10.000	E
ORARIO LINEA	1500	E
ORDINARIA	65	E
PERSONA	20.000	E
STRAORDINARIA	10	E
TARIFFA ABBONAMENTO	5	E

Tabella 3.1: Stima del volume dei dati

NOME	VOLUME STIMATO	E/A
TARIFFA BIGLIETTO	10	E
TIPOLOGIA MEZZO	4	E
TITOLO DI VIAGGIO	750.000	E
TRATTA	1.000	E
UTENTE	10.000	E
ACQUISTO ABBONAMENTO	30.000	A
ACQUISTO BIGLIETTO	150.000	A
ARRIVO	1.000	A
ASSUNZIONE	160	A
CONDUCENTE	250.000	A
CONTENUTO	450	A
CONTROLLO	50.000	A
CONVALIDA	200.000	A
EFFETTUAZIONE	250.000	A
EMISSIONE	10.000	A
FASCIA ABBONAMENTO	30.000	A
FASCIA BIGLIETTO	300.000	A
IMPIEGO	250.000	A
INCARICO	25	A
INCREMENTO	10	A
INTESTAZIONE	10.000	A
MODIFICA	10	A
NECESSITA	40	A
PARTENZA	1.000	A
PRESENZA	165	A
RIFERIMENTO	10.000	A
SOSTITUZIONE	10	A
SPECIFICAZIONE	1.500	A
TIPO LINEA	75	A
TIPOLOGIA	10.000	A
TRAGITTO	1.500	A

## 3.2 Descrizione operazioni

Riportiamo in Tabella 3.2 le principali operazioni che saranno svolte sulla base di dati, marcando con un asterisco (\*) le operazioni di cui è necessario calcolare i costi dati da attributi ridondanti.

Tabella 3.2: Numero stimato di operazioni per settimana,  
con tipo di utente che le effettua

#	Operazione	Op / 7gg	Tipo Utente
1*	Visualizzazione di tutte le linee attive	4.000	Tutti ****
2	Visualizzazione fermate e orari di una linea	3.500	Tutti ****

Tabella 3.2: Numero stimato di operazioni per settimana,  
con tipo di utente che le effettua

#	Operazione	Op / 7gg	Tipo Utente
3	Visualizzazione degli hub mobilità	500	Tutti
4*	Visualizzazione orario e mezzo assegnato	150	Autista *****
5*	Visualizzazione orario e linee assegnate	100	Controllore *****
6	Estrazione delle linee con più convalide nell'ultimo mese	3	Amministratore *****
7	Estrazione delle manutenzioni che coinvolgono un determinato mezzo	5	Amministratore
8	Estrazione delle manutenzioni ed eventuali linee sostitutive che coinvolgono una linea	5	Amministratore *****
9	Visualizzazione incassi dati dalle convalide per una linea	10	Amministratore ***
10	Estrazione degli incassi per tipo di titolo in periodo definito	5	Amministratore
11	Estrazione delle linee con più multe in periodo definito	2	Amministratore *****
12	Estrazione delle 5 linee con manutenzioni più gravose (in termini linee sostitutive e durata)	2	Amministratore *****
13	Estrazione delle linee con >5 controlli/giorno e <= 10 multe/giorno	3	Amministratore
14*	Visualizzazione delle linee con il maggior tempo di percorrenza	2	Amministratore *****
15	Estrazione della linea con più hub mobilità lungo il percorso	3	Amministratore *****
16	Media di soldi spesi in multe per persona	2	Amministratore
17	Visualizzazione delle aziende che non hanno effettuato nessuna manutenzione nell'ultimo mese	4	Amministratore
18	Visualizzazione delle fermate in cui è presente un hub mobilità contenente tutti i tipi di servizi green	1	Amministratore
19*	Inserimento di una variazione di servizio	1	Amministratore *****
20*	Aggiunta di una tratta a una linea esistente	1	Amministratore *****
21*	Creazione di una nuova linea	1	Amministratore *****

### 3.3 Analisi delle operazioni

Indichiamo con:

- $C_{tot}$  il costo totale dell'operazione.
- $O_{settimana}$  il numero di volte che l'operazione verrà eseguita in una settimana
- $A_{scrittura}$  il numero di accessi in scrittura effettuato da un'operazione
- $A_{lettura}$  il numero di accessi in lettura effettuato da un'operazione

Di seguito eseguiamo l'analisi delle operazioni presenti in Tabella 3.2, riportando lo schema di navigazione, la tabella degli accessi e il calcolo del costo della specifica operazione per ogni settimana.

### 1) Visualizzazione di tutte le linee attive

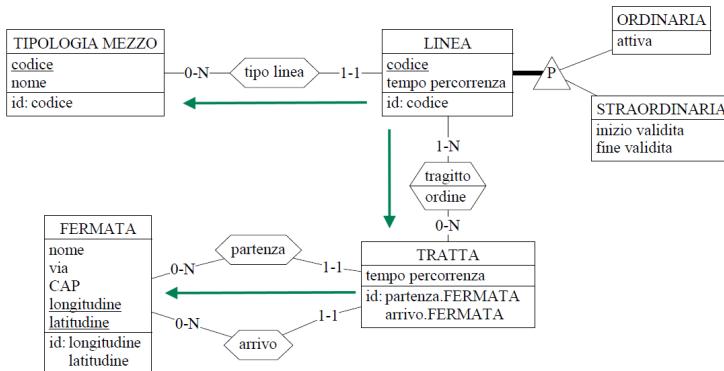
$$O_{settimana} = 4.000$$

Questa operazione serve per visualizzare tutte le linee che sono in funzione (ovvero non risultano in manutenzione) al momento attuale. Poiché è presente l'attributo ridondante **attiva** che potrebbe velocizzare lo svolgimento delle operazioni, viene eseguito il calcolo degli accessi sia tenendo conto della presenza dell'attributo, sia senza.

Si vuole quindi selezionare:

- Il codice della LINEA
- La prima e l'ultima FERMATA
- Il nome della TIPOLOGIA MEZZO utilizzato

#### • Analisi con attributo ridondante attiva su ORDINARIA



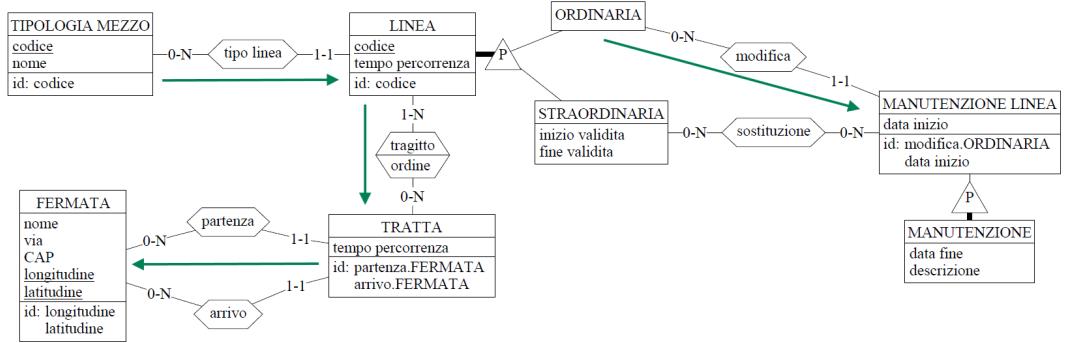
Nome	Tipo	Numero accessi	S/L
LINEA	E	75	L
TIPOLOGIA MEZZO	E	75	L
TRAGITTO	A	1.500	L
TRATTA	E	150	L
FERMATA (PARTENZA)	E	75	L
FERMATA (ARRIVO)	E	75	L

**Totalle**

$$A_{lettura} = 1.950, A_{scrittura} = 0$$

$$C_{tot} = O_{settimana} \cdot A_{lettura} = 7.800.000$$

#### • Analisi senza attributo ridondante attiva su ORDINARIA



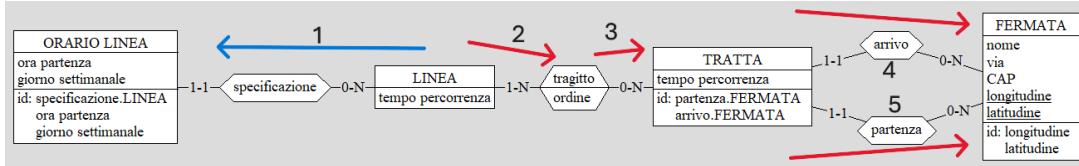
Nome	Tipo	Numero accessi	S/L
LINEA	E	75	L
MANUTENZIONE LINEA	E	10	L
TIPOLOGIA MEZZO	E	75	L
TRAGITTO	A	1.500	L
TRATTA	E	150	L
FERMATA (PARTENZA)	E	75	L
FERMATA (ARRIVO)	E	75	L
<b>Totalle</b>			
$A_{lettura} = 1.960, A_{scrittura} = 0$			

$$C_{tot} = O_{settimana} \cdot A_{lettura} = 7.840.000$$

## 2) Visualizzazione fermate e orari di una linea

$$O_{settimana} = 3.500$$

Dobbiamo visualizzare le linee, i suoi orari e le fermate fatte da essa per qualsiasi utente.



Nome	Tipo	Numero accessi	S/L
LINEA	E	1	L
ORARIO LINEA	E	20	L
TRAGITTO	A	20	L
TRATTA	E	20	L
FERMATA	E	40	L
<b>Totalle</b>			
$A_{lettura} = 71, A_{scrittura} = 0$			

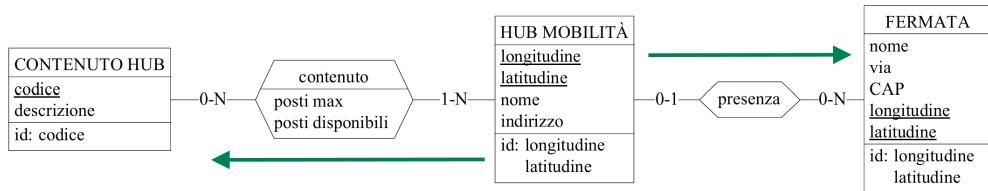
$$C_{tot} = O_{settimana} \cdot A_{lettura} = 248.500$$

### 3) Visualizzazione degli hub mobilità

$$O_{settimana} = 500$$

Questa operazione deve poter essere fatta da qualsiasi persona che accede all'applicativo. Si richiede per ogni HUB MOBILITÀ di visualizzare:

- Tutti gli attributi di HUB MOBILITÀ
- I tipi di CONTENUTO HUB per ogni HUB MOBILITÀ
- I posti rimanenti per ogni CONTENUTO HUB nell'HUB MOBILITÀ
- L'eventuale fermata a cui è collegato



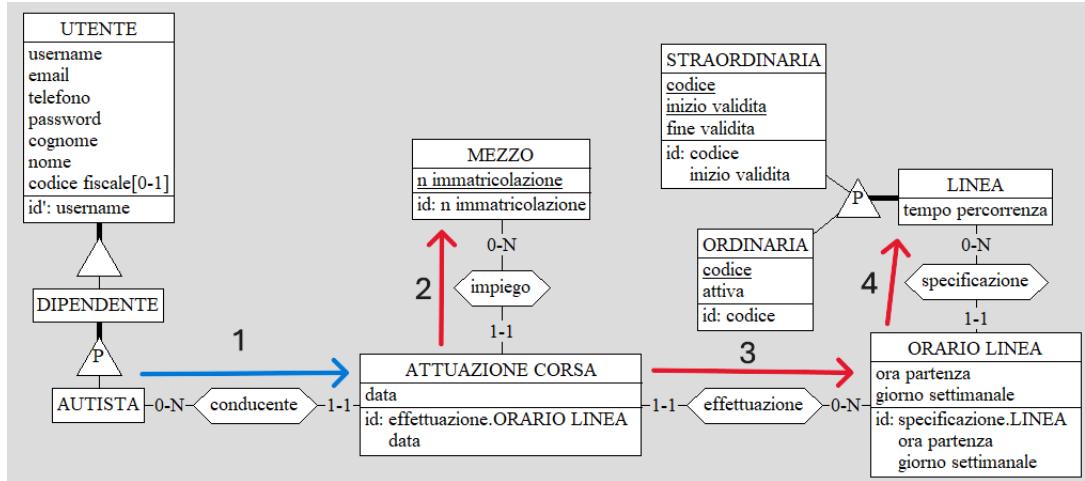
Nome	Tipo	Numero Accessi	S/L
HUB MOBILITÀ	E	225	L
CONTENUTO	A	$225 \times 2$	L
CONTENUTO HUB	E	$225 \times 2$	L
FERMATA	E	165	L
<b>Total</b>			
$A_{lettura} = 1290, A_{scrittura} = 0$			

$$C_{tot} = O_{settimana} \cdot A_{lettura} = 645.000$$

### 4) Visualizzazione orario e linee assegnate agli autisti

$$O_{settimana} = 150$$

- **Analisi con attributo ridondante tempo percorrenza su LINEA**  
Dobbiamo visualizzare l'orario di lavoro di un autista, mostrando la linea l'orario e il mezzo assegnato. La lettura partì da un autista; il numero di letture è stato calcolato dividendo il numero totale delle entità attuazione corsa (250.000) con il numero totale degli autisti (130), in modo da trovare la media di associazioni per ogni autista.

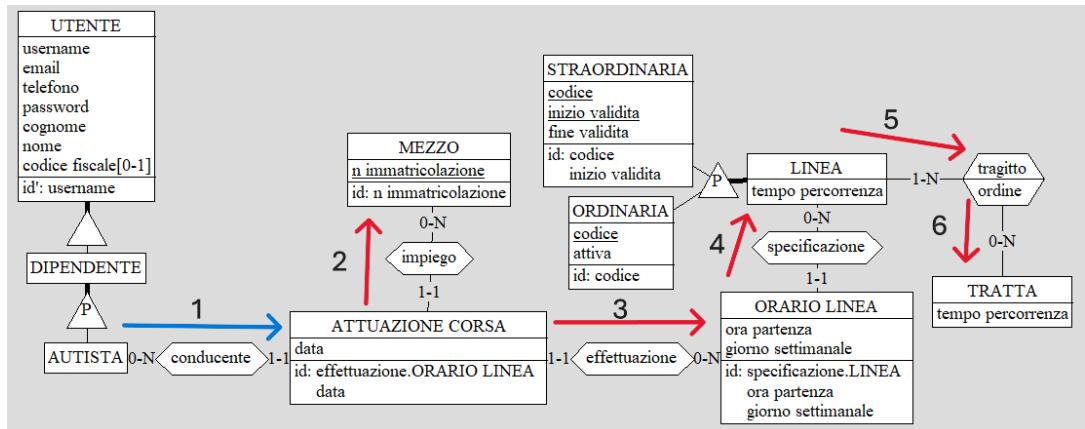


Nome	Tipo	Numero accessi	S/L
AUTISTA	E	1	L
ATTUAZIONE CORSA	E	1923	L
ORARIO LINEA	E	1923	L
LINEA	E	1923	L
MEZZO	E	1923	L

**Totalle**  
 $A_{lettura} = 7693, A_{scrittura} = 0$

$$C_{tot} = O_{settimana} \cdot A_{lettura} = 1.153.950$$

- Analisi senza attributo ridondante tempo percorrenza su LINEA**  
Dobbiamo visualizzare l'orario di lavoro di un autista, mostrando la linea l'orario e il mezzo assegnato. La lettura partirà da un autista, le letture sono state calcolate nello stesso modo.  
Visto che non abbiamo il tempo di percorrenza su **LINEA**, dobbiamo ricavarlo tramite le varie **TRATTE** che compongono la linea. Ogni **LINEA** ha in media 20 associazioni **TRAGITTO**.



Nome	Tipo	Numero accessi	S/L
AUTISTA	E	1	L
ATTUAZIONE CORSA	E	1.923	L
ORARIO LINEA	E	1.923	L
MEZZO	E	1.923	L
LINEA	E	1.923	L
TRAGITTO	A	38.460	L
TRATTA	E	38.460	L

**Totalle**

$$A_{lettura} = 84.613, A_{scrittura} = 0$$

$$C_{tot} = O_{settimana} \cdot A_{lettura} = 12.691.950$$

## 5) Visualizzazione orario e linee assegnate ai controllori

$$O_{settimana} = 100$$

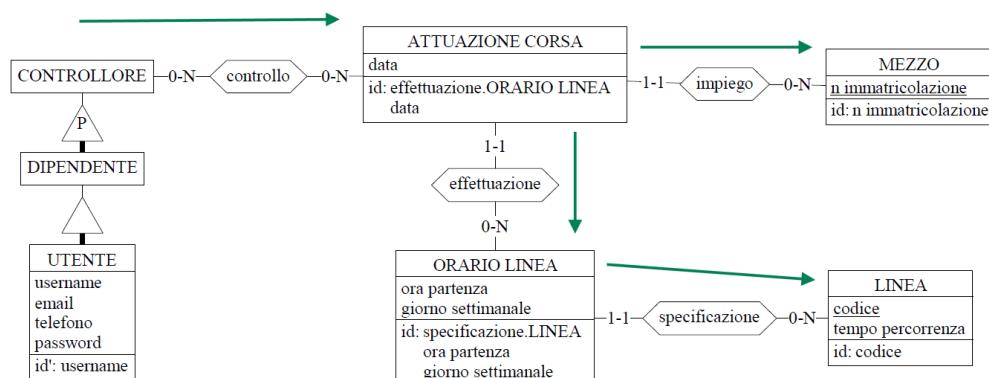
Questa operazione serve a un controllore per visualizzare il suo orario di lavoro e come dovrà essere svolto (ovvero le linee che dovrà controllare). Poiché è necessario calcolare l'orario di arrivo della linea, l'attributo **tempo percorrenza** su **LINEA** rende superfluo il calcolo del tempo tramite la somma dei tempi di tutte le **TRATTA**: è quindi necessario eseguire i calcoli dei costi con attributo ridondante e non.

Nei conti terremo conto del fatto che l'operazione è relativa a un solo giorno, ovvero sarà eseguito un filtro delle **ATTUAZIONE CORSA**.

Dati un controllore e una data, bisogna quindi visualizzare:

- L'orario di partenza, di arrivo e il codice della **LINEA** da controllare
- Il numero di immatricolazione del **MEZZO** che effettua la corsa

- **Analisi con attributo ridondante tempo percorrenza su LINEA**



Nome	Tipo	Numero accessi	S/L
CONTROLLORE	E	1	L
CONTROLLO	A	1.250	L
ATTUAZIONE CORSA	E	1.250	L
ORARIO LINEA	E	7	L
LINEA	E	7	L
MEZZO	E	7	L

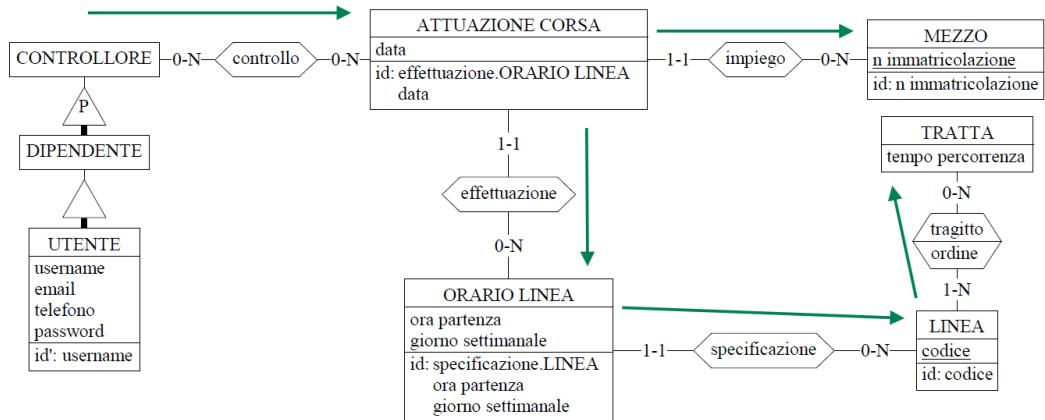
**Totalle**

$$A_{lettura} = 2.522, A_{scrittura} = 0$$

---


$$C_{tot} = O_{settimana} \cdot A_{lettura} = 252.200$$

- **Analisi senza attributo ridondante tempo percorrenza su LINEA**



Nome	Tipo	Numero accessi	S/L
CONTROLLORE	E	1	L
CONTROLLO	A	1.250	L
ATTUAZIONE CORSA	E	1.250	L
ORARIO LINEA	E	7	L
LINEA	E	7	L
TRAGITTO	A	140	L
TRATTA	E	140	L
MEZZO	E	7	L

**Totalle**

$$A_{lettura} = 2.802, A_{scrittura} = 0$$

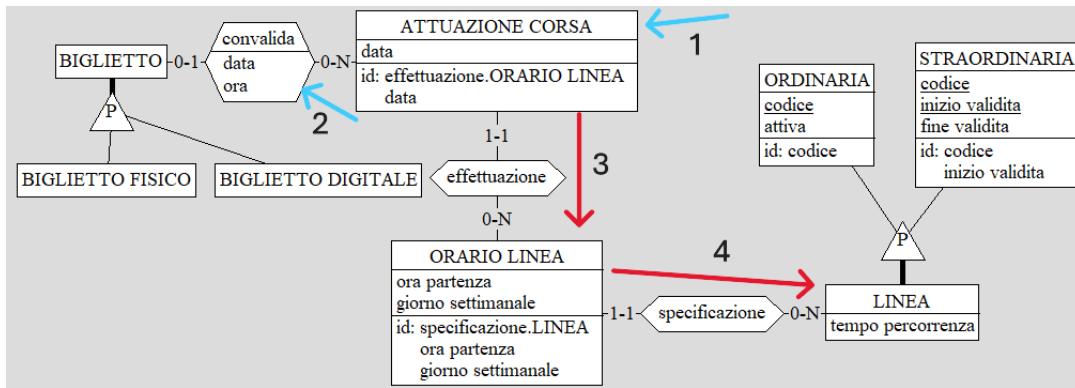
---


$$C_{tot} = O_{settimana} \cdot A_{lettura} = 280.200$$

## 6) Estrazione delle linee con più convalide nell'ultimo mese

$$O_{settimana} = 3$$

Dobbiamo estrarre le linee con più biglietti convalidati nell'ultimo mese. Il conteggio per la quantità di letture su CONVALIDA è stato fatto con una media tramite la tabella dei volumi.



Nome	Tipo	Numero accessi	S/L
ATTUAZIONE CORSA	E	250.000	L
CONVALIDA	E	33.333	L
ORARIO LINEA	E	1	L
LINEA	E	1	L

**Totalle**

$$A_{lettura} = 283.335, A_{scrittura} = 0$$


---

$$C_{tot} = O_{settimana} \cdot A_{lettura} = 850.005$$

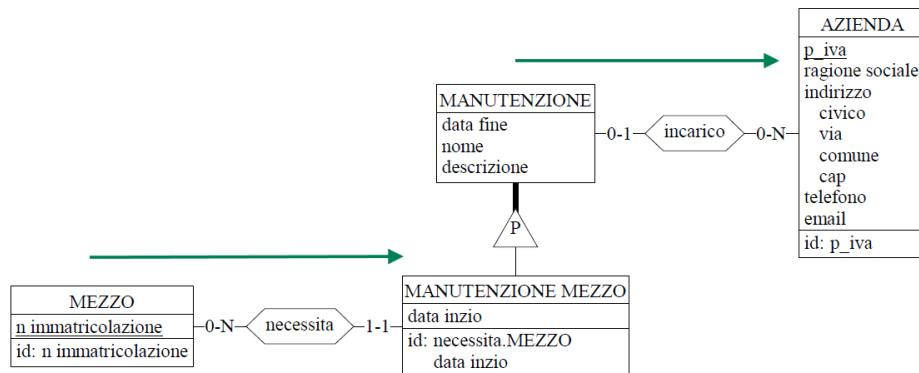
## 7) Estrazione delle manutenzioni che coinvolgono un determinato mezzo

$$O_{settimana} = 5$$

Questa operazione serve per visualizzare tutte le manutenzioni che sono state effettuate ad un mezzo.

Dato un mezzo bisogna quindi visualizzare:

- Tutti i dati della **MANUTENZIONE**
- La partita iva dell’eventuale **AZIENDA** esterna incaricata della manutenzione



Nome	Tipo	Numero accessi	S/L
MEZZO	E	1	L
MANUTENZIONE MEZZO	E	4	L
AZIENDA	E	2	L
<b>Totale</b>			
$A_{lettura} = 7, A_{scrittura} = 0$			

$$C_{tot} = O_{settimana} \cdot A_{lettura} = 35$$

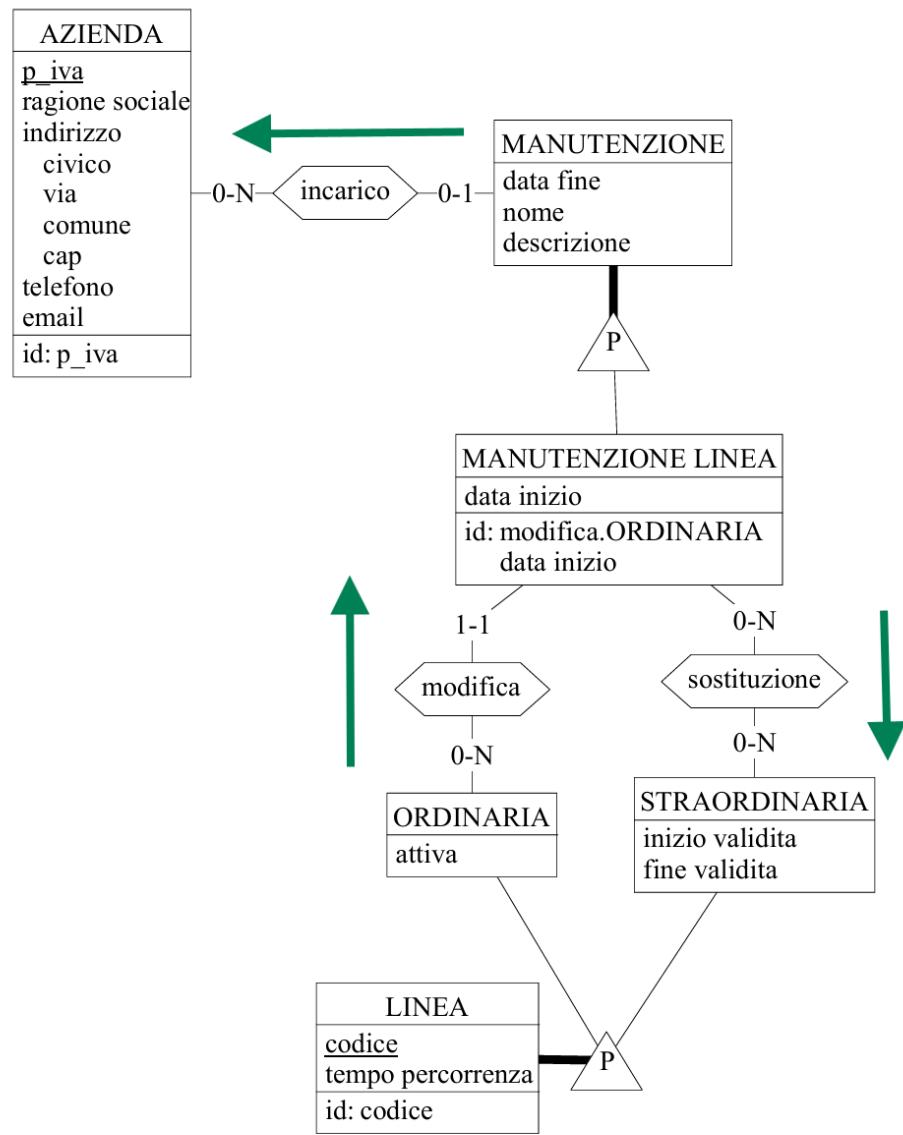
- 8) Estrazione delle manutenzioni ed eventuali linee sostitutive che coinvolgono una linea (**Variazioni di servizio**)

$$O_{settimana} = 5$$

Questa operazione deve poter essere effettuata da qualsiasi utente. Consente di visualizzare le variazioni di servizio di una specifica linea.

Specificata una linea, va estratto:

- Tutti gli attributi di **MANUNTEZIONE LINEA** e **MANUNTEZIONE**
- L’azienda che effettua la manutenzione
- Tutte le linee straordinarie (codice, inizioValidita, fineValidita, partenza, arrivo) con cui viene sostituita la linea



Nome	Tipo	Numero Accessi	S/L
ORDINARIA	E	1	L
MANUNtenzione LINEA	E	10	L
AZIENDA	E	10	L
SOSTITUZIONE	A	1 × 10	L
STRAORDINARIA	E	1 × 10	L

**Total**

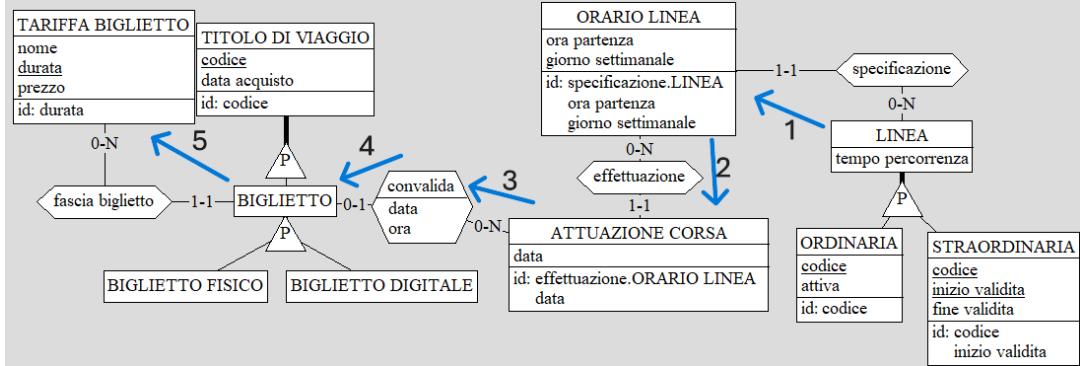
$$A_{lettura} = 41, A_{scrittura} = 0$$

$$C_{tot} = O_{settimana} \cdot A_{lettura} = 205$$

## 9) Visualizzazione incassi dati dalle convalide per una linea

$$O_{settimana} = 10$$

Partendo da una linea dobbiamo visualizzare gli incassi fatti da essa tramite i biglietti convalidati.



Nome	Tipo	Numero accessi	S/L
LINEA	E	1	L
ORARIO LINEA	E	20	L
ATTUAZIONE CORSA	E	167	L
CONVALIDA	A	209	L
BIGLIETTO	E	209	L
TARIFFA BIGLIETTO	E	209	L
<b>Total</b>			
$A_{lettura} = 815, A_{scrittura} = 0$			

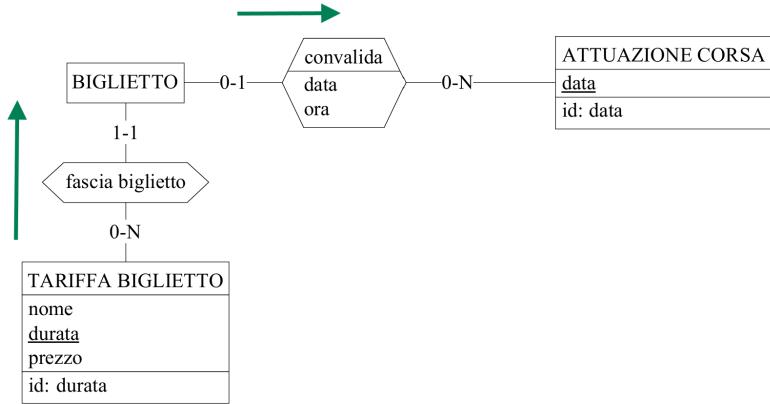
$$C_{tot} = O_{settimana} \cdot A_{lettura} = 8.150$$

## 10) Estrazione degli incassi per tipo di tariffa dei biglietti in periodo definito

$$O_{settimana} = 5$$

Vogliamo visualizzare gli incassi di una specifica tariffa dei biglietti in un certo periodo di tempo; per eseguire i calcoli considereremo come periodo una settimana. Per rendere la stima più realistica non terremo conto della data di acquisto dei biglietti, ma della data dell'effettiva convalida. Data una certa tariffa dei biglietti e l'intervallo di tempo vogliamo quindi visualizzare:

- Il numero di CONVALIDA nel dato periodo di tempo
- L'incasso totale, ovvero il prezzo della TARIFFA BIGLIETTO data per il numero di convalide



Nome	Tipo	Numero accessi	S/L
TARIFFA BIGLIETTO	E	1	L
BIGLIETTO	E	30.000	L
CONVALIDA	A	20.000	L
<b>Total</b>			
$A_{lettura} = 50.001, A_{scrittura} = 0$			

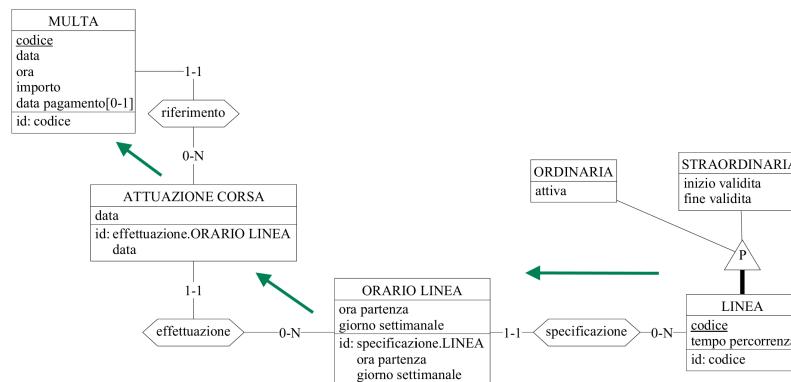
$$C_{tot} = O_{settimana} \cdot A_{lettura} = 250.005$$

## 11) Estrazione delle linee con più multe in periodo definito

$$O_{settimana} = 2$$

Questa operazione deve poter essere effettuata dall'amministratore di sistema.  
Per ogni linea va estratto:

- Il codice della linea
- Il numero di multe ottenute negli un periodo definito



Nome	Tipo	Numero Accessi	S/L
LINEA	E	75	L
ORARIO LINEA	E	75 × 20	L
ATTUAZIONE CORSA	E	3333 × 75	L
MULTA	E	0.04 × 75	L

**Totale**

$$A_{lettura} = 251.850, A_{scrittura} = 0$$


---

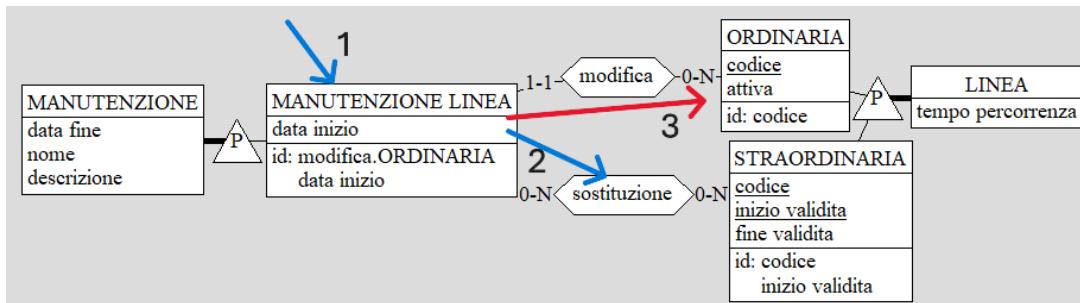
$$C_{tot} = O_{settimana} \cdot A_{lettura} = 503.700$$

- 12) Estrazione delle 5 linee con manutenzioni più gravose (in termini di linee sostitutive e durata)**

$$O_{settimana} = 2$$

Dobbiamo visualizzare le linee con le manutenzioni più gravose, a livello applicativo verrà calcolato un "punteggio" che indicherà quanto sia gravosa una manutenzione:

- +5 pti ogni linea straordinaria dovuta alla manutenzione
- +1 pto ogni 3 giorni di lavoro sulla linea.



Nome	Tipo	Numero accessi	S/L
MANUTENZIONE LINEA	E	10	L
SOSTITUZIONE	A	10	L
LINEA ORDINARIA	E	5	L

**Totale**

$$A_{lettura} = 25, A_{scrittura} = 0$$


---

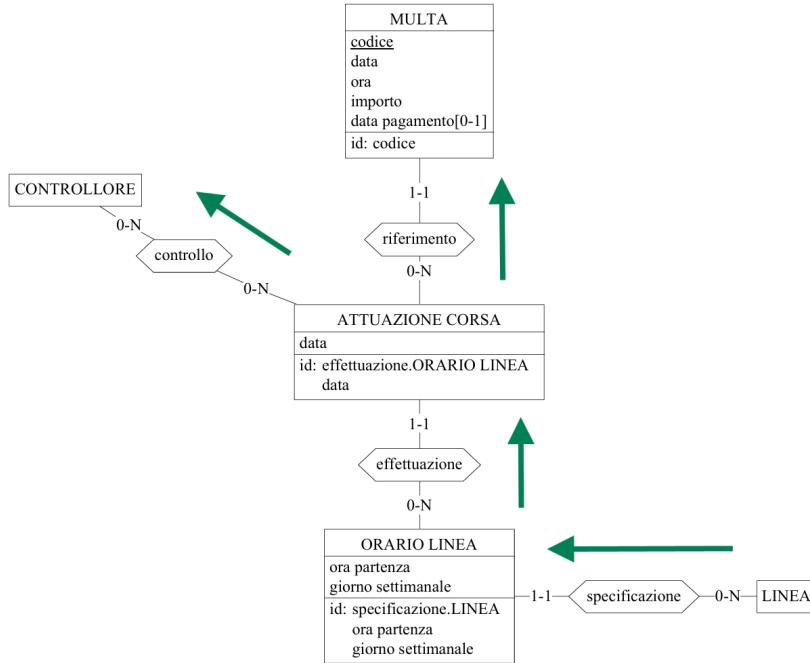
$$C_{tot} = O_{settimana} \cdot A_{lettura} = 100$$

- 13) Estrazione delle linee con > 5 controlli/giorno e ≤ 10 multe/giorno**

$$O_{settimana} = 3$$

Questa operazione deve poter essere effettuata dall'amministratore di sistema. Devono essere estratti i codici delle linee che rispettano la seguente condizione:

$$\text{controllo} > 5/\text{giorno} \cap \text{multe} \leq 10/\text{giorno}$$



Nome	Tipo	Numero Accessi	S/L
LINEA	E	75	L
ORARIO LINEA	E	$20 \times 75$	L
ATTUAZIONE CORSA	E	$3333 \times 75$	L
CONTROLLO	A	$0.2 \times 75$	L
<b>Totalle</b>			
$A_{lettura} = 251.700, A_{scrittura} = 0$			

$$C_{tot} = O_{settimana} \cdot A_{lettura} = 755.100$$

#### 14) Visualizzazione delle linee con il maggior tempo di percorrenza

$$O_{settimana} = 2$$

- **Analisi con attributo ridondante tempo percorrenza su LINEA**  
In questo caso basterà leggere il dato del tempo di percorrenza dalla relativa linea.

Nome	Tipo	Numero accessi	S/L
LINEA	E	75	L

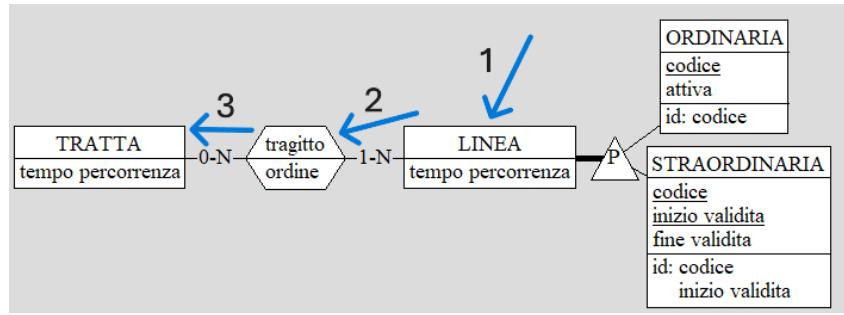
**Totalle**

---

$A_{lettura} = 75, A_{scrittura} = 0$

$$C_{tot} = O_{settimana} \cdot A_{lettura} = 150$$

- **Analisi senza attributo ridondante tempo percorrenza su LINEA**  
In questo caso invece dovremmo calcolare il tempo di percorrenza leggendo tutte le tratte di cui è composta la relativa linea.



Nome	Tipo	Numero accessi	S/L
LINEA	E	75	L
TRAGITTO	A	1.500	L
TRATTA	E	1.500	L
<b>Totalle</b>			
$A_{lettura} = 3.075, A_{scrittura} = 0$			

$$C_{tot} = O_{settimana} \cdot A_{lettura} = 6.150$$

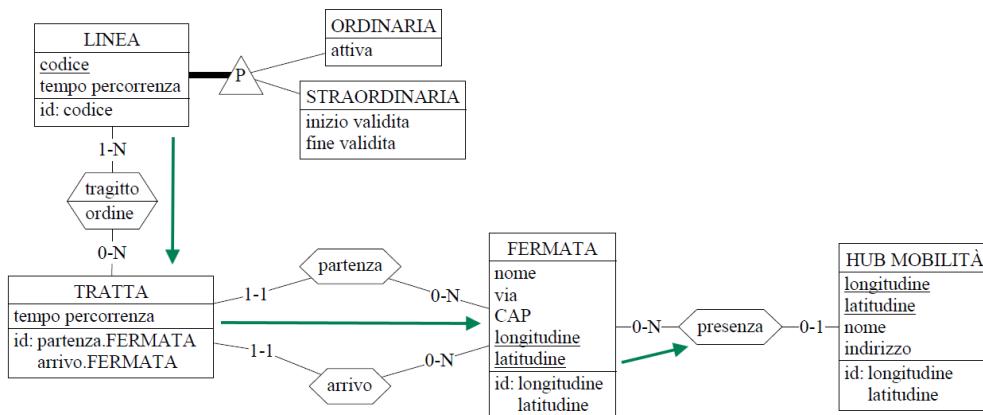
### 15) Estrazione della linea con più hub mobilità lungo il percorso

$$O_{settimana} = 3$$

A fini statistici si vuole scoprire qual è la linea che contiene più Hub Mobilità, dati dalla somma di tutte gli Hub che stanno nelle varie fermate.

Si vuole quindi estrarre:

- Il codice della **LINEA** che contiene il maggior numero di hub



Nome	Tipo	Numero accessi	S/L
LINEA	E	75	L
TRAGITTO	A	1.500	L
TRATTA	E	1.000	L
FERMATA	E	450	L
PRESENZA	A	165	L
<b>Totalle</b>			
$A_{lettura} = 3.190, A_{scrittura} = 0$			

$$C_{tot} = O_{settimana} \cdot A_{lettura} = 9.570$$

### 16) Media di soldi spesi in multe per persona

$$O_{settimana} = 1$$

Questa operazione deve poter essere effettuata dall'amministratore di sistema.  
Deve essere estratto un unico numero, ovvero la media del prezzo totale pagato per persona.

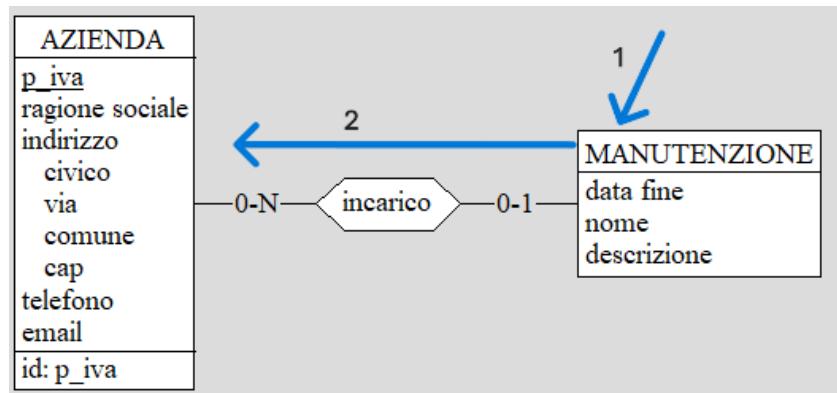
Nome	Tipo	Numero Accessi	S/L
MULTA	E	10000	L
CAUSALE MULTA	E	10000	L
<b>Totalle</b>			
$A_{lettura} = 20.000, A_{scrittura} = 0$			

$$C_{tot} = O_{settimana} \cdot A_{lettura} = 20.000$$

### 17) Visualizzazione delle aziende che non hanno effettuato nessuna manutenzione nell'ultimo mese

$$O_{settimana} = 4$$

Questa operazione serve a estrarre le aziende che non hanno effettuato nessuna manutenzione nell'ultimo mese.



Nome	Tipo	Numero accessi	S/L
MANUTENZIONE	E	50	L
AZIENDA	E	x	L
<b>Totalle</b>			
$A_{lettura} = 50 + x, A_{scrittura} = 0$			

$$C_{tot} = O_{settimana} \cdot A_{lettura} = 200 + 4x$$

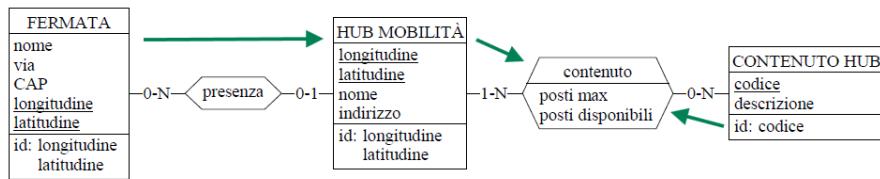
**18) Visualizzazione delle fermate in cui è presente almeno un hub mobilità contenente tutti i tipi di servizi green**

$$O_{settimana} = 1$$

Questa operazione punta a ricercare le fermate più eco-friendly, ovvero che fanno riferimento ad almeno un hub mobilità che contenga TUTTI i tipi di servizi possibili. Si tratta quindi di un'operazione di divisione.

Si vuole quindi visualizzare:

- L'elenco dei nomi di tutte le FERMATA che rispettano le specifiche



Nome	Tipo	Numero accessi	S/L
FERMATA	E	450	L
HUB MOBILITA	E	225	L
CONTENUTO	A	450	L
CONTENUTO HUB	E	4	L

**Totale**

$$A_{lettura} = 1.129, A_{scrittura} = 0$$


---

$$C_{tot} = O_{settimana} \cdot A_{lettura} = 1.129$$

**19) Inserimento di una variazione di servizio**

$$O_{settimana} = 1$$

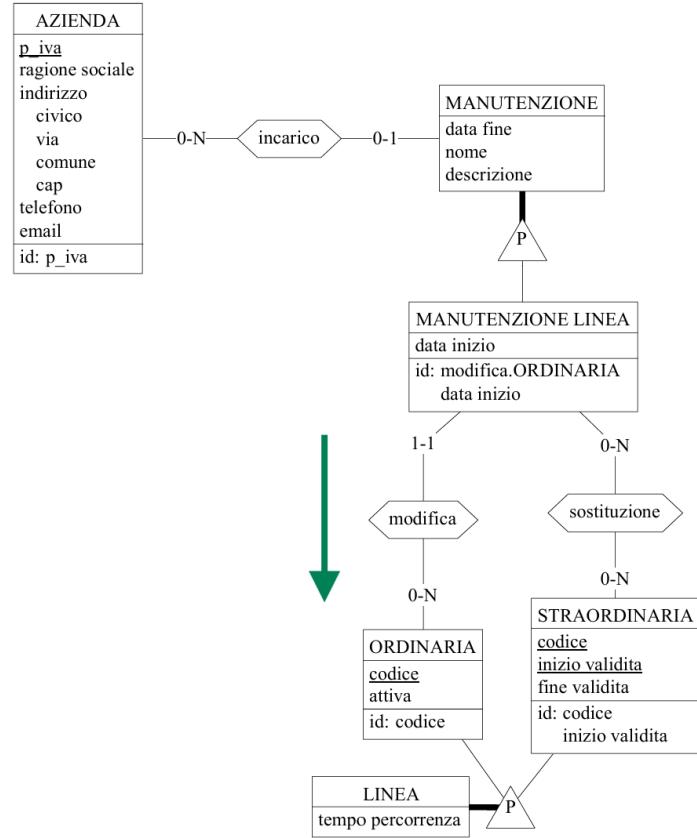
Questa operazione deve poter essere effettuata dall'amministratore di sistema. Supponendo che le linee straordinarie sostitutive siano già state inserite, si deve creare una nuova manutenzione.

Alla manutenzione va indicata la linea che viene modificata, e le eventuali linee che la sostituiscono.

Inoltre se l'inizio della manutenzione è immediato, allora la linea coinvolta va disattivata.

**OSS:** va calcolato l'impatto di avere o no l'attributo ridondante ATTIVA

- Analisi con attributo ridondante attiva su LINEA



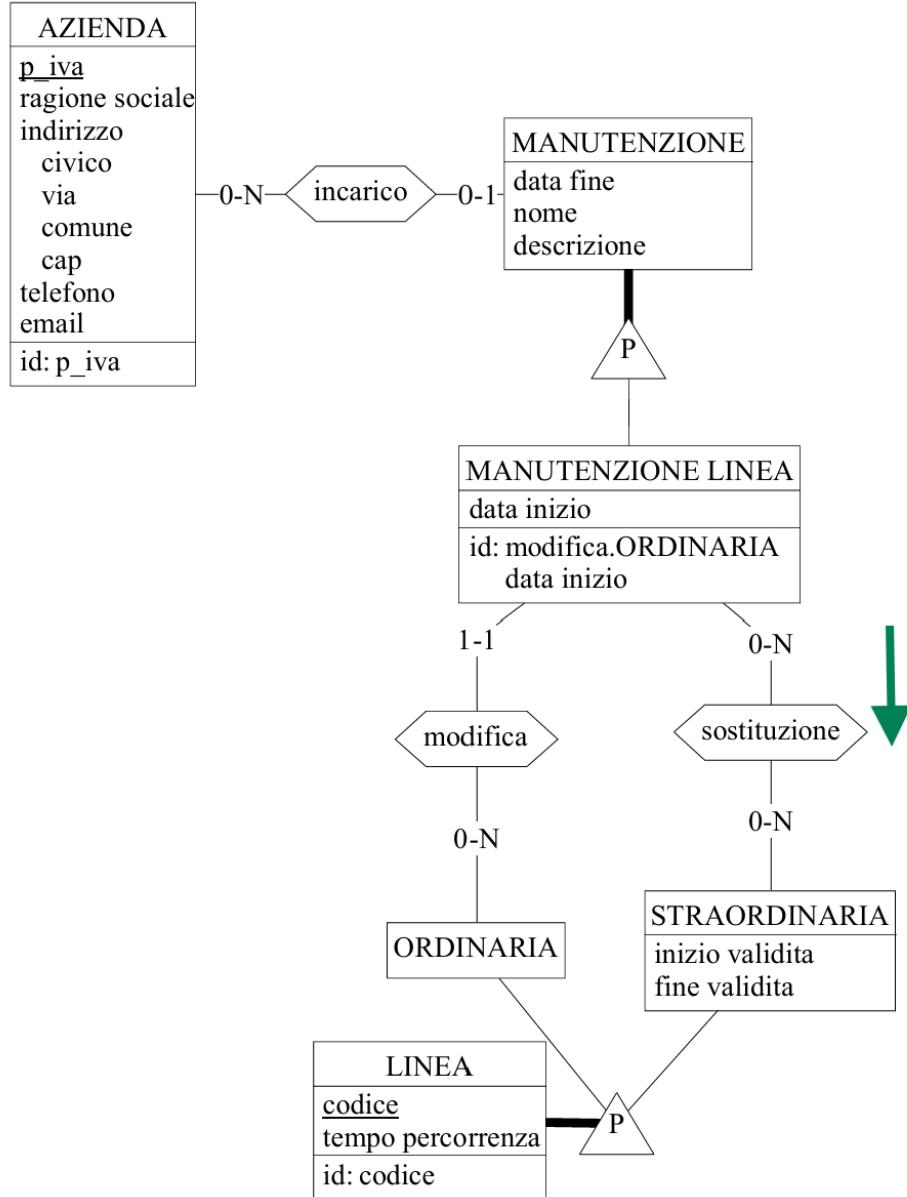
Nome	Tipo	Numero Accessi	S/L
MANUNTENZIONE LINEA	E	1	S
ORDINARIA	E	1	L
ORDINARIA	E	1	S
SOSTITUZIONE	A	1	S

**Totalle**  
 $A_{lettura} = 1, A_{scrittura} = 3$

---

$$C_{tot} = O_{settimana} \cdot (A_{lettura} + 2A_{scrittura}) = 7$$

- **Analisi senza attributo ridondante attiva su LINEA**



Nome	Tipo	Numero Accessi	S/L
MANUNtenzione LINEA	E	1	S
SOSTITUZIONE	A	1	S

**Total**

$$A_{lettura} = 0, A_{scrittura} = 2$$

$$C_{tot} = O_{settimana} \cdot (A_{lettura} + 2A_{scrittura}) = 4$$

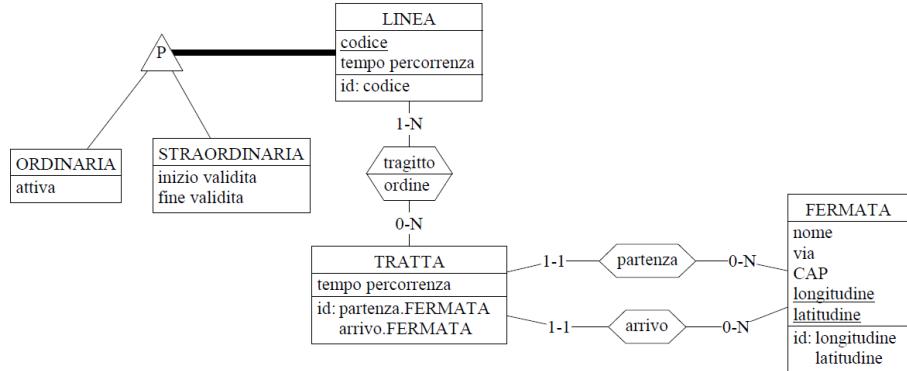
## 20) Aggiunta di una tratta a una linea esistente

$$O_{settimana} = 1$$

Questa operazione serve per aggiornare una linea, aggiungendo una fermata al suo tragitto. Nell'analisi terremo conto che la fermata va aggiunta dopo il capolinea, ovvero è necessario solo aggiungere la tratta per poi aggiungerla al tragitto. Visto

che il tempo di percorrenza di una linea è calcolabile anche come la somma dei tempi di percorrenza delle tratte, si rende necessaria l'analisi differenziata per la presenza dell'attributo **tempo percorrenza** su LINEA, poiché in caso di ridondanza sarà necessario rieseguire il calcolo del tempo totale per poi aggiornare l'attributo.

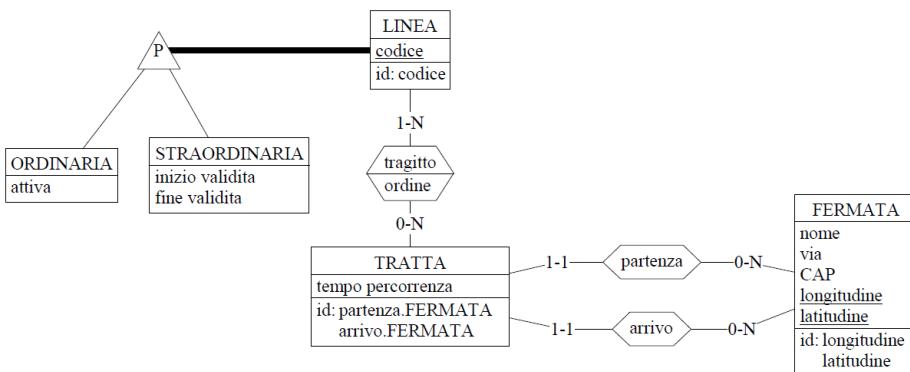
- **Analisi con attributo ridondante tempo percorrenza su LINEA**



Nome	Tipo	Numero accessi	S/L
TRATTA	E	1	S
TRAGITTO	A	1	S
TRATTA	A	21	L
LINEA	E	1	S + L
<b>Totalle</b>			
$A_{lettura} = 22, A_{scrittura} = 3$			

$$C_{tot} = O_{settimana} \cdot (A_{lettura} + 2A_{scrittura}) = 28$$

- **Analisi senza attributo ridondante tempo percorrenza su LINEA**



Nome	Tipo	Numero accessi	S/L
TRATTA	E	1	S
TRAGITTO	A	1	S
$A_{lettura} = 0, A_{scrittura} = 2$			

$$C_{tot} = O_{settimana} \cdot 2A_{scrittura} = 4$$

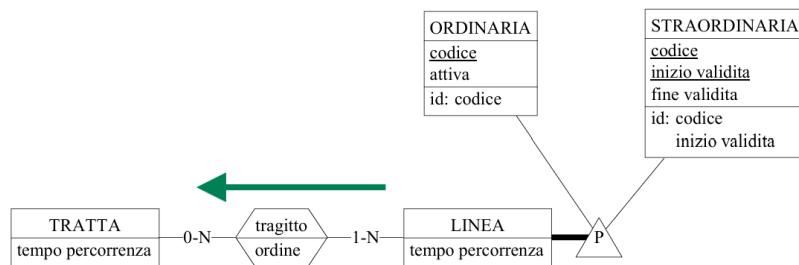
## 21) Creazione di una nuova linea

$$O_{settimana} = 1$$

Questa operazione deve poter essere fatta dall'amministratore di sistema.

Supponendo che le tratte siano già definite, per ogni linea va indicato l'insieme delle tratte che percorre e, per ogni tratta, specificare l'ordine con cui viene percorsa. Inoltre vanno sommati i tempi di percorrenza di ogni tratta per calcolare il campo tempo percorrenza

- **Analisi con attributo ridondante tempo percorrenza su LINEA**



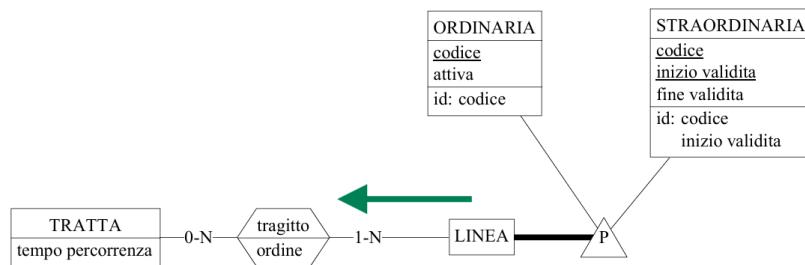
Nome	Tipo	Numero Accessi	S/L
LINEA	E	1	S
TRAGITTO	A	20	S
TRATTA	E	20	L

**Totalle**

$$A_{lettura} = 21, A_{scrittura} = 20$$

$$C_{tot} = O_{settimana} \cdot 2A_{scrittura} = 61$$

- **Analisi con attributo ridondante tempo percorrenza su LINEA**



Nome	Tipo	Numero Accessi	S/L
LINEA	E	1	S
TRAGITTO	A	20	S

**Totalle**

$$A_{lettura} = 1, A_{scrittura} = 20$$

$$C_{tot} = O_{settimana} \cdot 2A_{scrittura} = 41$$

## 3.4 Analisi delle ridondanze

In questa sezione andremo ad analizzare le due ridondanze presenti nel modello, ovvero:

- tempo percorrenza su LINEA
- attiva su LINEA

### 3.4.1 Analisi attributo tempo percorrenza

Num Op.	Con ridondanza	Senza ridondanza
4	1.153.950	12.691.950
5	252.200	280.200
14	150	6.150
20	28	4
21	61	41
<b>TOTALE</b>	<b>1.406.389</b>	<b>12.978.345</b>

A fronte di questa analisi, decidiamo di mantenere l'attributo **tempo percorrenza**.

### 3.4.2 Analisi attributo attiva

Num Op.	Con ridondanza	Senza ridondanza
1	1.950	7.800.000
19	7	4
<b>TOTALE</b>	<b>1.957</b>	<b>7.800.004</b>

A fronte di questa analisi, decidiamo di mantenere l'attributo **attiva**.

## 3.5 Riepilogo operazioni

Nella seguente tabella proponiamo un riepilogo delle operazioni analizzate, con il costo totale per ogni operazione.

Tabella 3.3: Costo totale delle operazioni stimato per 7 giorni

#	Operazione	Costo tot. / 7gg	Tipo Utente
1	Visualizzazione di tutte le linee attive	7.800.000	Tutti
2	Visualizzazione fermate e orari di una linea	248.500	Tutti
3	Visualizzazione degli hub mobilità	645.000	Tutti
4*	Visualizzazione orario e mezzo assegnato	1.153.950	Autista
5*	Visualizzazione orario e linee assegnate	252.200	Controllore
6	Estrazione delle linee con più convalide nell'ultimo mese	850.005	Amministratore

Tabella 3.3: Costo totale delle operazioni stimato per 7 giorni

#	Operazione	Costo tot. / 7gg	Tipo Utente
7	Estrazione delle manutenzioni che coinvolgono un determinato mezzo	35	Amministratore
8	Estrazione delle manutenzioni ed eventuali linee sostitutive che coinvolgono una linea	205	Amministratore
9	Visualizzazione incassi dati dalle convalide per una linea	8.150	Amministratore
10	Estrazione degli incassi per tipo di titolo in periodo definito	250.005	Amministratore
11	Estrazione delle linee con più multe in periodo definito	503.700	Amministratore
12	Estrazione delle 5 linee con manutenzioni più gravose (in termini linee sostitutive e durata)	100	Amministratore
13	Estrazione delle linee con >5 controlli/giorno e <= 10 multe/giorno	755.100	Amministratore
14*	Visualizzazione delle linee con il maggior tempo di percorrenza	150	Amministratore
15	Estrazione della linea con più hub mobilità lungo il percorso	9.570	Amministratore
16	Media di soldi spesi in multe per persona	20.000	Amministratore
17	Visualizzazione delle aziende che non hanno effettuato nessuna manutenzione nell'ultimo mese	4XXX	Amministratore
18	Visualizzazione delle fermate in cui è presente un hub mobilità contenente tutti i tipi di servizi green	1.129	Amministratore
19*	Inserimento di una variazione di servizio	7	Amministratore
20*	Aggiunta di una tratta a una linea esistente	28	Amministratore
21*	Creazione di una nuova linea	61	Amministratore

## 3.6 Raffinamento dello schema

Ora che abbiamo deciso quali attributi dovranno rimanere nella base di dati, non ci resta che apportare dei raffinamenti allo schema ER in modo da rielaborare i concetti non direttamente rappresentabili tramite il modello relazionale. Procediamo quindi per step successivi fino ad arrivare a uno schema logico equivalente allo schema ER.

### 3.6.1 Rimozione attributi multivaleore

L'unico attributo multivaleore nel nostro schema è presente nell'entità AZIENDA, nello specifico l'indirizzo. Aggiungiamo quindi il prefisso `indirizzo_` a ogni suo componente per poi decomporlo in attributi singoli.

### 3.6.2 Rimozione gerarchie

#### Biglietto

La gerarchia su biglietto è di tipo Totale ed Esclusiva. Poiché non ci sono attributi diversi nelle due entità figlie e **BIGLIETTO DIGITALE** ha l'associazione **acquisto** di cardinalità (1,1), scegliamo di effettuare un **collasso verso l'alto**. L'associazione **aquisto** diventa quindi a cardinalità (0, 1).

Sarà possibile riconoscere i biglietti digitali dai fisici perché questi ultimi non avranno l'associazione **acquisto** con l'entità **UTENTE**.

#### Titolo di Viaggio

Anche in questo caso la gerarchia è di tipo Totale ed Esclusiva. Dal momento che le entità figlie hanno associazioni e attributi diversi, scegliamo di effettuare un **collasso verso il basso** che ci permette di mantenere questi vincoli inalterati.

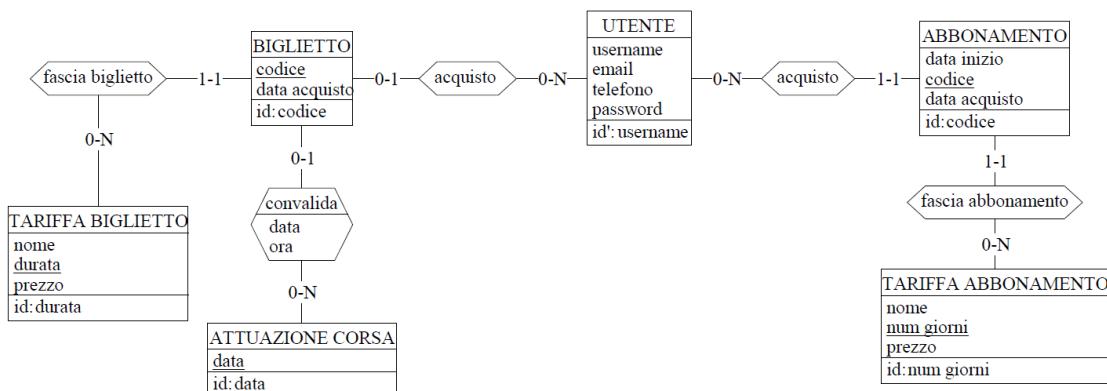


Figura 3.1: Entità Biglietto e Abbonamento dopo la rimozione delle gerarchie

#### Persona - Utente

Abbiamo che **UTENTE** è un subset di **PERSONA**. Se venisse effettuato un collasso avremmo vari attributi opzionali da gestire: scegliamo quindi di sostituire la gerarchia con un'**associazione**. Così facendo a una persona fisica può essere connessa un'utenza, e a un'utenza deve per forza essere connessa una persona.

#### Utente - Dipendente

Come per Persona e Utente il subset è stato sostituito dall'associazione **dipendenza**, visto che le entità **UTENTE** e **DIPENDENTE** hanno associazioni diverse.

#### Dipendente

La gerarchia sui vari tipi di dipendente (**AMMINISTRATIVO**, **AUTISTA**, **CONTROLLORE**) è di tipo Totale ed Esclusiva. Visto che le sottoentità non hanno attributi diversificati, scegliamo di fare un **collasso verso l'alto**. Introduciamo un selettore di tipo chiamato **ruolo** che potrà assumere solo i valori di Amministrativo, Autista o Controllore. Visto

che le associazioni specifiche per ogni ruolo ora fanno riferimento al generico dipendente sarà necessario porre attenzione ad eseguire dei controlli per verificare che si possano aggiungere delle associazioni solo se coerenti con il ruolo del dipendente coinvolto.

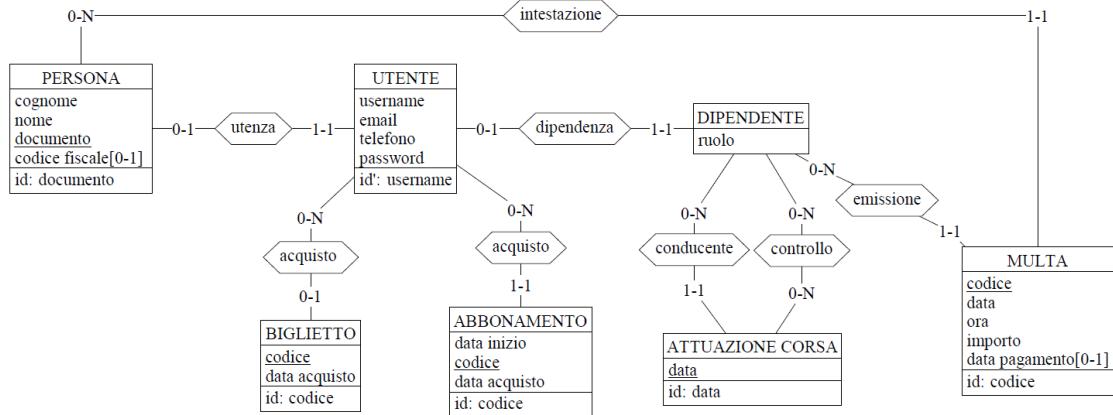


Figura 3.2: Entità Persona, Utente e Dipendente dopo la rimozione delle gerarchie

## Manutenzione

La gerarchia su MANUTENZIONE è di tipo Totale ed Esclusiva. Poiché le due entità figlie si riferiscono a due concetti differenti, quindi hanno anche associazioni diverse, scegliamo di effettuare un **collasso verso il basso** per mantenere i vincoli sulle associazioni. Importiamo quindi sui due figli gli attributi comuni e l'associazione comune **incarico**.

## Linea

Le linee di tipo ordinario e straordinario compongono una gerarchia Totale ed Esclusiva. Scegliamo di eseguire un **collasso verso l'alto**, così da avere tutte le linee in un'unica entità visto che sono oggetto di molte richieste comuni. Dal momento che hanno alcuni attributi diversi, questi diventeranno degli opzionali e ci serviranno per capire di che tipo è la linea.

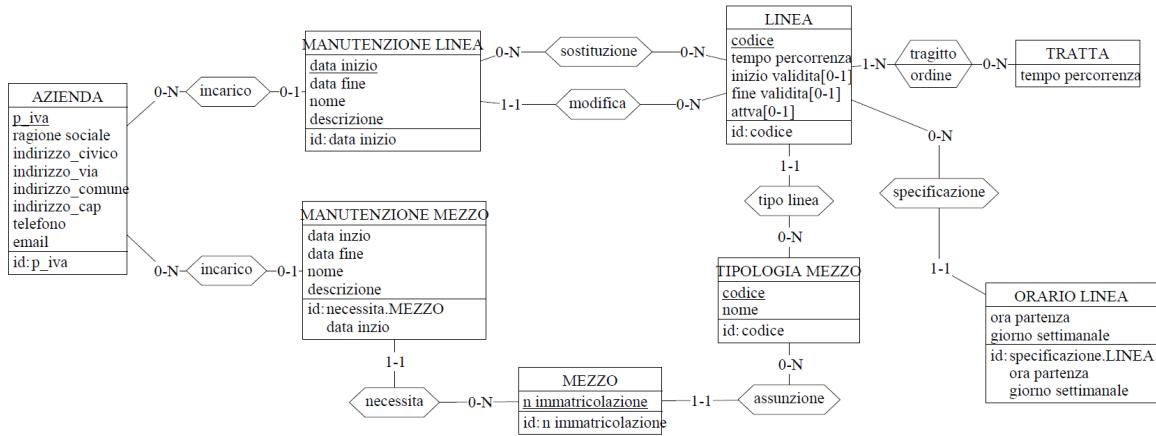


Figura 3.3: Entità Manutenzione e Linea dopo la rimozione delle gerarchie

### 3.6.3 Reificazione associazioni molti a molti

Le associazioni N-N non sono direttamente rappresentabili nei database relazionali, quindi per ovviare a questo problema, le associazioni subiscono un processo di **reificazione** che comporta la trasformazione dell'associazione N-N in un'entità che sarà connessa con due associazioni 1-1 alle entità che prima collegava.

Qui elencate troviamo le associazioni che sono state reificate:

- Contenuto
- Tragitto
- Sostituzione
- Controllo

### 3.6.4 Scelta degli identificatori principali

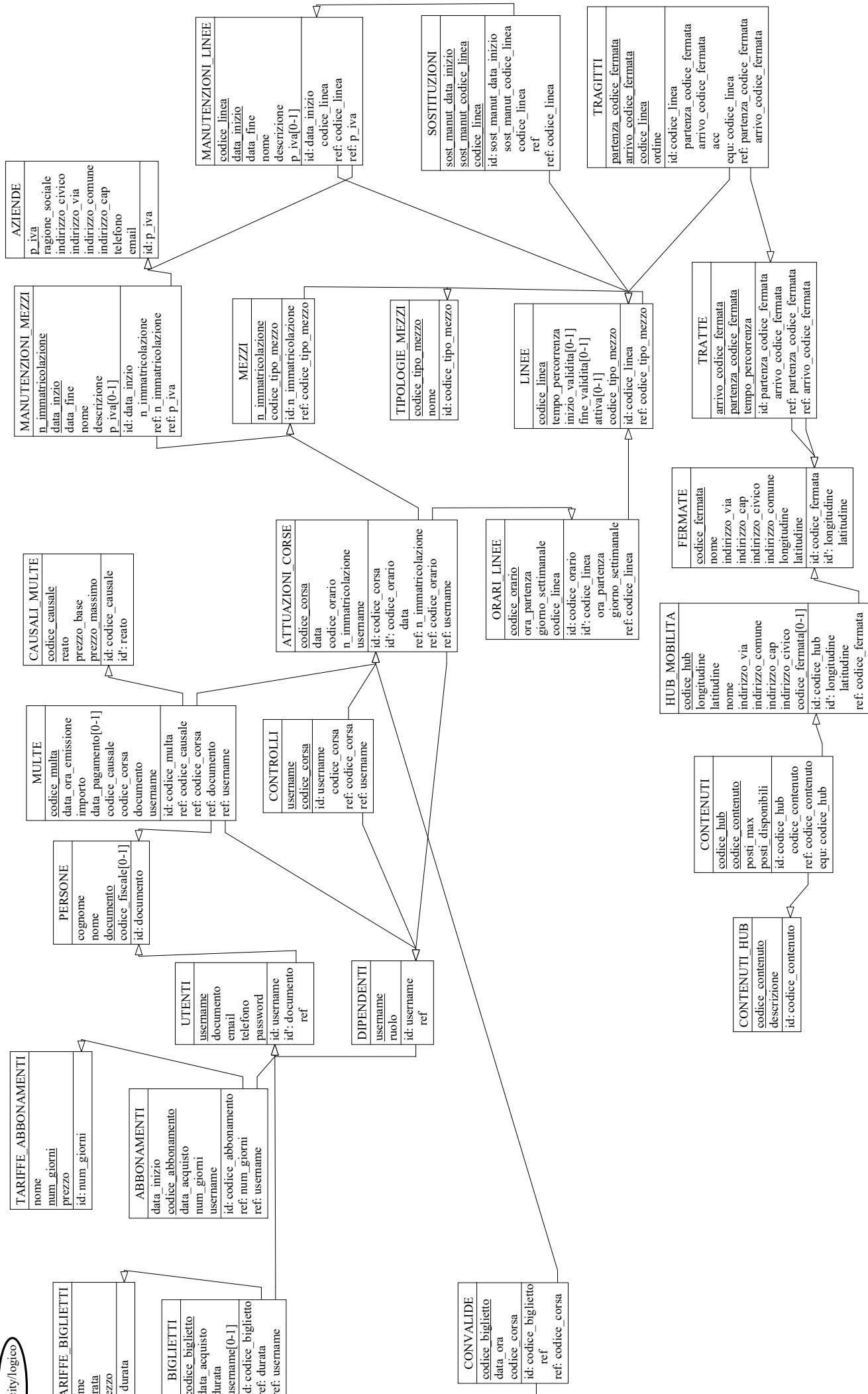
Tutte le entità del nostro dominio sono già dotate dei loro identificatori principali naturali. Per comodità ne sostituiamo alcuni con un semplice codice (lasciando comunque il vincolo di unicità su quelli esistenti) per rendere più semplici le query.

Riportiamo l'elenco delle entità in cui è stata effettuata quest'aggiunta:

- Fermata
- Hub mobilità
- Orario linea
- Causale multa
- Attuazione Corsa

### **3.7 Schema relazionale finale**

A questo punto la ristrutturazione è terminata, e il nostro schema è direttamente traducibile in delle relazioni. Riportiamo nelle pagine seguenti lo schema logico sia in formato grafico che testuale.



smart\_city/logico

- PERSONE(documento, cognome, nome, codice\_fiscale\*)
- UTENTI(username, documento: PERSONE, email, telefono, password)  
UNIQUE: (documento)
- DIPENDENTI(username: UTENTI, ruolo)
- FERMATE(codice\_fermata, nome, indirizzo\_via, indirizzo\_cap, indirizzo\_civico, indirizzo\_comune, longitudine, latitudine)  
UNIQUE: (longitudine, latitudine)
- HUB\_MOBILITA(codice\_hub, longitudine, latitudine, nome, indirizzo\_via, indirizzo\_comune, indirizzo\_cap, indirizzo\_civico, codice\_fermata\*: FERMATE)  
UNIQUE: (longitudine, latitudine)
- CONTENUTI\_HUB(codice\_contenuto, descrizione)
- CONTENUTI(codice\_hub: HUB\_MOBILITA, codice\_contenuto: CONTENUTI\_HUB, posti\_max, posti\_disponibili)
- TRATTE(partenza\_codice\_fermata: FERMATE, arrivo\_codice\_fermata: FERMATE, tempo\_percorrenza)
- TRAGITTI((partenza\_codice\_fermata, arrivo\_codice\_fermata): TRATTE, codice\_linea: LINEE, ordine)
- TIPOLOGIE\_MEZZI(codice\_tipo\_mezzo, nome)
- MEZZI(n\_immatricolazione, codice\_tipo\_mezzo: TIPOLOGIE\_MEZZI)
- LINEE(codice\_linea, tempo\_percorrenza, inizio\_validita\*, fine\_validita\*, attiva\*, codice\_tipo\_mezzo: TIPOLOGIE\_MEZZI)
- ORARI\_LINEE(codice\_orario, ora\_partenza, giorno\_settimanale, codice\_linea: LINEE)  
UNIQUE: (codice\_linea, ora\_partenza, giorno\_settimanale)
- ATTUAZIONI\_CORSE(codice\_corsa, data, codice\_orario: ORARI\_LINEE, n\_immatricolazione: MEZZI, username: DIPENDENTI)  
UNIQUE: (codice\_orario, data)
- CAUSALI\_MULTe(codice\_causale, reato, prezzo\_base, prezzo\_massimo)  
UNIQUE: (reato)
- MULTe(codice\_multa, data\_ora\_emissione, importo, data\_pagamento\*, codice\_causale: CAUSALI\_MULTe, codice\_corsa: ATTUAZIONI\_CORSE, documento: PERSONE, username: DIPENDENTI)
- CONTROLLI(username: DIPENDENTI, codice\_corsa: ATTUAZIONI\_CORSE)
- TARIFFE\_BIGLIETTI(durata, nome, prezzo)
- BIGLIETTI(codice\_biglietto, data\_acquisto, durata: TARIFFE\_BIGLIETTI, username\*: UTENTI)

- CONVALIDE(codice\_biglietto: BIGLIETTI, data\_ora,  
codice\_corsa: ATTUAZIONI\_CORSE)
- TARIFFE\_ABBONAMENTI(num\_giorni, nome, prezzo)
- ABBONAMENTI(codice\_abbonamento, num\_giorni: TARIFFE\_ABBONAMENTI,  
data\_inizio, data\_acquisto, username: UTENTI)
- AZIENDE(p\_iva, ragione\_sociale, indirizzo\_civico, indirizzo\_via,  
indirizzo\_comune, indirizzo\_cap, telefono, email)
- MANUTENZIONE\_MEZZI(n\_immatricolazione: MEZZI, data\_inizio, data\_fine,  
nome, descrizione, p\_iva\*: AZIENDE)
- MANUTENZIONI\_LINEE(codice\_linea: LINEE, data\_inizio, data\_fine, nome,  
descrizione, p\_iva\*: AZIENDE)
- SOSTITUZIONI(  
sost\_manut\_data\_inizio, sost\_manut\_codice\_linea): MANUNTENZIONI\_LINEE,  
codice\_linea: LINEE)

# Capitolo 4

## Progettazione della Base di Dati

Abbiamo creato la base di dati per il nostro applicativo mappando le relazioni (vedi section 3.7) in tabelle, utilizzando il DBMS MySQL. Per eseguire la traduzione abbiamo sfruttato lo strumento messo a disposizione da DB-Main.

Riportiamo in seguito il codice per eseguire ogni operazione indicata in Table 3.2, e le altre peculiarità relative alla creazione del database.

### 4.1 Check

Nella creazione delle tabelle abbiamo inserito vari check, per poter definire meglio il dominio dei dati inseribili nel database.

```
01 | create table TARIFFE_BIGLIETTI (
02 |     nome varchar(30) not null,
03 |     durata int not null,
04 |     prezzo decimal(6,2) not null,
05 |     check (durata > 0 and prezzo > 0),
06 |     constraint ID_TARIFFA_BIGLIETTO primary key (durata));
```

Listing 4.1: Esempio di check utilizzato per definire gli attributi prezzo e durata come numeri positivi.

### 4.2 Viste

Visto che l'operazione di visualizzazione delle linee attive nel giorno corrente ricopre un ruolo centrale nel nostro applicativo, abbiamo deciso di creare una vista che estrae tutte le linee attive (sia ordinarie che straordinarie) riportando anche la prima e l'ultima fermata del tragitto complessivo.

```
01 | CREATE VIEW VW_LINEE_ATTIVE_OGGI AS
02 | SELECT L.* , M.nome AS tipo_mezzo , F1.codice_fermata AS
    part_codice_fermata , F1.nome AS part_nome , F1.indirizzo_via AS
    part_via , F1.indirizzo_civico AS part_civico , F1.indirizzo_comune AS
    part_comune , F1.indirizzo_cap AS part_cap , F1.longitudine AS
    part_long , F1.latitudine AS part_lat , F2.codice_fermata AS
    arr_codice_fermata , F2.nome AS arr_nome , F2.indirizzo_via AS arr_via
    , F2.indirizzo_civico AS arr_civico , F2.indirizzo_comune AS
```

```

    arr_comune, F2.indirizzo_cap AS arr_cap, F2.longitudine AS arr_long,
    F2.latitudine AS arr_lat
03 | FROM LINEE L
04 | JOIN TIPOLOGIE_MEZZI M ON L.codice_tipo_mezzo = M.codice_tipo_mezzo
05 | JOIN TRAGITTI TRA1 on TRA1.codice_linea = L.codice_linea
06 | JOIN TRATTE TR1 ON (TRA1.partenza_codice_fermata = TR1.
    partenza_codice_fermata AND TRA1.arrivo_codice_fermata = TR1.
    arrivo_codice_fermata)
07 | JOIN FERMATE F1 ON F1.codice_fermata = TR1.partenza_codice_fermata
08 | JOIN TRAGITTI TRA2 on TRA2.codice_linea = L.codice_linea
09 | JOIN TRATTE TR2 ON (TRA2.partenza_codice_fermata = TR2.
    partenza_codice_fermata AND TRA2.arrivo_codice_fermata = TR2.
    arrivo_codice_fermata)
10 | JOIN FERMATE F2 ON F2.codice_fermata = TR2.arrivo_codice_fermata
11 | WHERE TRA1.ordine = 1
12 |     AND TRA2.ordine =
13 |         (SELECT MAX(T1.ordine)
14 |             FROM LINEE L1 JOIN TRAGITTI T1 ON L1.codice_linea = T1.
    codice_linea
15 |                 WHERE L1.codice_linea = L.codice_linea)
16 |                 AND (L.attiva IS TRUE OR (CURDATE() BETWEEN L.
    inizio_validita AND L.fine_validita)
17 |                     )
18 | ORDER BY L.codice_linea;

```

Listing 4.2: Codice SQL per la creazione della vista

## 4.3 Stored Procedures

L'introduzione della ridondanza sull'attributo **attiva** delle linee ordinarie ne rende necessaria la sua gestione (attivazione e disattivazione). Per rendere più semplice questa operazione, abbiamo creato due stored procedures:

- La prima prende il codice di una linea e controlla che non ci sia attualmente nessuna manutenzione per la linea data
- La seconda esegue l'operazione per tutte le linee ordinarie, sfruttando la procedura precedente

```

01 | DELIMITER //
02 |
03 | CREATE PROCEDURE aggiorna_attivazione_linea(IN cod_linea VARCHAR(30))
04 | BEGIN
05 |     UPDATE linee L
06 |     SET attiva = NOT EXISTS
07 |         (SELECT 1
08 |             FROM manutenzioni_linee
09 |                 WHERE codice_linea = cod_linea AND CURDATE() BETWEEN
    data_inizio AND data_fine)
10 |                 WHERE codice_linea = cod_linea;
11 | END //
12 |
13 | DELIMITER ;

```

```

01 | DELIMITER //
02 |
03 | CREATE PROCEDURE aggiorna_attivazione_linee()
04 | BEGIN
05 |     DECLARE done INT DEFAULT FALSE;
06 |     DECLARE cur_cod_linea VARCHAR(30);
07 |
08 |     DECLARE cur CURSOR FOR
09 |         SELECT codice_linea FROM linee WHERE attiva IS NOT NULL;
10 |
11 |     DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
12 |
13 |     OPEN cur;
14 |
15 |     read_loop: LOOP
16 |         FETCH cur INTO cur_cod_linea;
17 |         IF done THEN
18 |             LEAVE read_loop;
19 |         END IF;
20 |         CALL aggiorna_attivazione_linea(cur_cod_linea);
21 |     END LOOP;
22 |
23 |     CLOSE cur;
24 | END //
25 |
26 | DELIMITER ;

```

## 4.4 Trigger

Per rendere automatica la gestione della ridondanza dell’attributo `attiva`, ogni volta che viene effettuata un’operazione che modifica la tabella `MANUTENZIONI LINEE` viene richiamata la stored procedure per aggiornare l’attributo della linea modificata sfruttando dei trigger. Questo permette agli utilizzatori del database di gestire le manutenzioni senza preoccuparsi di tenere aggiornato l’attributo `attiva`.

```

01 | DELIMITER //
02 |
03 | CREATE TRIGGER dopo_insert_manut_linea
04 | AFTER INSERT ON manutenzioni_linee
05 | FOR EACH ROW
06 | BEGIN
07 |     CALL aggiorna_attivazione_linea(NEW.codice_linea);
08 | END// 
09 |
10 | CREATE TRIGGER dopo_update_manut_linea
11 | AFTER UPDATE ON manutenzioni_linee
12 | FOR EACH ROW
13 | BEGIN
14 |     CALL aggiorna_attivazione_linea(NEW.codice_linea);
15 | END// 
16 |
17 | CREATE TRIGGER dopo_delete_manut_linea
18 | AFTER DELETE ON manutenzioni_linee
19 | FOR EACH ROW
20 | BEGIN

```

```

21 |     CALL aggiorna_attivazione_linea(OLD.codice_linea);
22 | END//;
23 |
24 | DELIMITER ;

```

## 4.5 Eventi

Le manutenzioni delle linee possono terminare in una data qualiasi registrata al momento dell'inserimento della manutenzione stessa: ad esempio oggi potrebbe essere terminata la manutenzione della linea 2, ma nessuno potrebbe essersi ricordato di riattivare la linea. Diventa quindi necessario eseguire ogni giorno il "refresh" di tutte le linee attive nel giorno corrente. Questa operazione viene eseguita sfruttando un evento MySQL, che invoca una volta al giorno la stored procedure adibita all'aggiornamento dell'attributo attiva su tutte le linee ordinarie.

```

01 | CREATE EVENT aggiorna_linee_event
02 | ON SCHEDULE EVERY 1 DAY
03 | DO
04 |     CALL aggiorna_attivazione_linee();

```

## 4.6 Traduzione delle operazioni in query SQL

Nelle query potrebbero apparire dei simboli ?: questo perché abbiamo utilizzato i prepared statement per poter adattare le clausole where ai filtri imposti dagli utenti.

### 1) Visualizzazione di tutte le linee attive

Quest'operazione corrisponde alla visualizzazione delle informazioni estratte dalla vista indicata in Listing 4.2. Sfruttiamo quindi la vista senza dover riscrivere la query.

```

01 | SELECT *
02 | FROM VW_LINEE_ATTIVE_OGGI;

```

### 2) Visualizzazione fermate e orari di una linea

Questa operazione è stata divisa in 2 parti dato che sulla UI non mostreremo mai tutte le fermate con i relativi orari di ogni linea; ma mostreremo tutte le linee poi l'utente potrà visualizzare le fermate e gli orari della linea desiderata.

```

01 | SELECT t.codice_linea, t.arrivo_codice_fermata, t.
02 |     partenza_codice_fermata, t.ordine, f.nome, f.indirizzo_via, f.
03 |     indirizzo_civico, f.indirizzo_comune, f.indirizzo_cap, tr.
04 |     tempo_percorrenza
05 | FROM TRAGITTI t
06 | JOIN TRATTE tr ON t.partenza_codice_fermata = tr.
07 |     partenza_codice_fermata
08 | AND t.arrivo_codice_fermata = tr.arrivo_codice_fermata
09 | JOIN FERMATE f ON t.arrivo_codice_fermata = f.codice_fermata
10 | WHERE t.codice_linea = ?

```

Listing 4.3: Query for Fermate Details by Linea

```

01 | SELECT ol.codice_orario, ol.codice_linea, ol.orario_partenza, ol.
     giorno_settimanale, ac.data
02 | FROM ORARI_LINEE ol
03 | JOIN ATTUAZIONI_CORSE ac ON ol.codice_orario = ac.codice_orario
04 | WHERE ol.codice_linea = ?

```

Listing 4.4: Query for Orari Linee with Attuazioni Corse

### 3) Visualizzazione degli hub mobilità

```

01 | SELECT h.codice_hub, h.nome nome_hub, h.indirizzo_via, h.
     indirizzo_civico, h.indirizzo_comune, h.indirizzo_cap, h.
     longitudine, h.latitudine, h.codice_fermata, f.nome
     nome_fermata, ch.descrizione tipo_contenuto, c.
     posti_disponibili
02 | FROM HUB_MOBILITA h LEFT JOIN FERMATE f on (h.codice_fermata = f.
     codice_fermata)
03 | JOIN CONTENUTI c ON (c.codice_hub = h.codice_hub)
04 | JOIN CONTENUTI_HUB ch ON (c.codice_contenuto = ch.codice_contenuto
     )
05 | ORDER BY h.codice_hub;

```

Listing 4.5: Query for Hub Mobilità with Contenuti and Fermate

### 4) Visualizzazione orario e linee assegnate agli autisti

Con questa operazione andiamo a visualizzare l'orario di lavoro di un autista e il mezzo che gli è stato assegnato per le linee che dovrà percorrere. Le ore effettive di lavoro potranno essere visualizzate tramite il tempo di percorrenza delle linee che dovrà percorrere.

```

01 | SELECT ol.codice_linea, ol.ora_partenza, ol.giorno_settimanale, ol.
     .codice_orario, ac.data, l.tempo_percorrenza, ac.
     n_immatricolazione
02 | FROM ORARI_LINEE ol
03 | JOIN ATTUAZIONI_CORSE ac ON ol.codice_orario = ac.codice_orario
04 | JOIN LINEE l ON ol.codice_linea = l.codice_linea
05 | WHERE ac.username = ?
06 | ORDER BY ac.data, ol.ora_partenza;

```

Listing 4.6: Query for Corse Attuate by User

### 5) Visualizzazione orario e linee assegnate ai controllori

```

01 | SELECT A.data, O.codice_orario, O.ora_partenza, O.
     giorno_settimanale, O.codice_linea, L.tempo_percorrenza, A.
     n_immatricolazione
02 | FROM CONTROLLI C
03 | JOIN ATTUAZIONI_CORSE A ON C.codice_corsa = A.codice_corsa
04 | JOIN ORARI_LINEE O ON A.codice_orario = O.codice_orario
05 | JOIN LINEE L ON O.codice_linea = L.codice_linea
06 | WHERE C.username = ?
07 | ORDER BY A.data, O.ora_partenza;

```

### 6) Estrazione delle linee con più convalide nell'ultimo mese

Con questa operazione andremo ad estrarre il numero di convalide per linea in ordine decrescente.

```

01 | SELECT COUNT(*) AS numero_convalide
02 | FROM LINEE l
03 | JOIN ORARI_LINEE ol ON l.codice_linea = ol.codice_linea
04 | JOIN ATTUAZIONI_CORSE ac ON ol.codice_orario = ac.codice_orario
05 | JOIN CONVALIDE c ON ac.codice_corsa = c.codice_corsa
06 | GROUP BY codice_linea
07 | ORDER BY COUNT(*) DESC

```

Listing 4.7: Query for Number of Validations per Linea

## 7) Estrazione delle manutenzioni che coinvolgono un determinato mezzo

```

01 | SELECT *
02 | FROM MANUTENZIONI_MEZZI mm
03 | WHERE mm.n_immatricolazione = ?

```

## 8) Estrazione delle manutenzioni ed eventuali linee sostitutive che coinvolgono una linea (Variazioni di servizio)\*

```

01 | SELECT l.codice_linea codice_linea_in_manutenzione, m.data_inizio,
      m.data_fine, m.nome, m.descrizione, a.p_iva, a.email, a.
      telefono, a.ragione_sociale, ls.codice_linea
      codice_linea_sostituta
02 | FROM LINEE l JOIN MANUTENZIONI_LINEE m ON (m.codice_linea = l.
      codice_linea)
03 | LEFT JOIN AZIENDE a ON (m.p_iva = a.p_iva)
04 | LEFT JOIN SOSTITUZIONI s ON (m.codice_linea = s.
      sost_manut_codice_linea AND m.data_inizio = s.
      sost_manut_data_inizio)
05 | JOIN LINEE ls ON (ls.codice_linea = s.codice_linea)
06 | WHERE l.codice_linea = ?;

```

Listing 4.8: Query for Linee in Manutenzione and Their Sostituzioni

## 9) Visualizzazione incassi dati dalle convalide per una linea

```

01 | SELECT sum(tb.prezzo) AS incassi
02 | FROM ORARI_LINEE ol
03 | JOIN ATTUAZIONI_CORSE ac ON ol.codice_linea = ac.codice_linea
04 | JOIN CONVALIDE c ON ac.codice_corsa = c.codice_corsa
05 | JOIN BIGLIETTI b ON c.codice_biglietto = b.codice_biglietto
06 | JOIN TARIFFE_BIGLIETTI tb ON b.durata = tb.durata
07 | WHERE ol.codice_linea = ?

```

Listing 4.9: Query for Total Revenue (Incassi) by Linea

## 10) Estrazione degli incassi per tipo di tariffa dei biglietti in periodo definito

Utilizziamo la funzione DATE() per trasformare l'attributo data\_ora da tipo DATETIME a tipo DATE.

```

01 | SELECT SUM(prezzo) as incasso
02 | FROM CONVALIDE C
03 | JOIN BIGLIETTI B ON C.codice_biglietto = B.codice_biglietto
04 | JOIN TARIFFE_BIGLIETTI T ON B.durata = T.durata
05 | WHERE B.durata = ? AND (DATE(C.data_ora) BETWEEN ? AND ?);

```

## 11) Estrazione delle linee con più multe in periodo definito

```
01 | SELECT l.codice_linea, COUNT(*) numero_multe
02 | FROM LINEE l, ORARI_LINEE ol, ATTUAZIONI_CORSE ac, MULTES m
03 | WHERE l.codice_linea = ol.codice_linea
04 | AND ol.codice_orario = ac.codice_orario
05 | AND ac.codice_corsa = m.codice_corsa
06 | AND ac.data BETWEEN ? AND ?
07 | GROUP BY l.codice_linea
08 | ORDER BY numero_multe DESC;
```

Listing 4.10: Query for Number of Fines per Linea Between Two Dates

## 12) Estrazione delle 5 linee con manutenzioni più gravose (in termini di linee sostitutive e durata)

Dobbiamo visualizzare le linee con le manutenzioni più gravose; la gravità verrà calcolata tramite un punteggio che indicherà quanto siano "pesanti" le manutenzioni:

- +5 pti ogni linea straordinaria dovuta alla manutenzione
- +1 pto ogni 3 giorni di lavoro sulla linea.

```
01 | SELECT ml.codice_linea, ml.nome, DATEDIFF(ml.data_fine, ml.
02 | data_inizio) AS durata_lavoro, COUNT(*) AS
03 | num_linee_sostitutive
04 | FROM MANUTENZIONI_LINEE ml
05 | JOIN SOSTITUZIONI s ON ml.codice_linea = s.sost_manut_codice_linea
06 | GROUP BY ml.codice_linea, ml.data_inizio, ml.nome
```

Listing 4.11: Query for Duration of Maintenance and Substitute Lines Count

## 13) Estrazione delle linee con >5 controlli/giorno e <= 10 multe/giorno

Lo scopo è quello di estrarre le linee che in tutti i giorni in cui sono state effettuate attuazioni corsa, sono stati svolti 5 controlli e 10 multe. La query è stata divisa in 2 subquery:

- Nella query più interna vengono selezionati i giorni in cui la condizione *controlli > 5*  $\cap$  *multe  $\leq$  10* è rispettata
- Nella query di mezzo vengono contate quante sono le attuazioni corsa effettuate nei giorni che rispettano la condizione

Alla fine vengono contate le attuazioni corsa totali di una linea e vengono confrontate con il numero di attuazioni corsa contate dalla subquery di mezzo. Se sono lo stesso numero (ovvero tutte le attuazioni corsa rispettano la condizione) allora la linea viene visualizzata.

```
01 | SELECT DISTINCT l.codice_linea
02 | FROM LINEE l JOIN ORARI_LINEE ol ON (l.codice_linea = ol.
03 | codice_linea)
04 | JOIN ATTUAZIONI_CORSE ac ON (ol.codice_orario = ac.codice_orario)
05 | GROUP BY l.codice_linea
06 | HAVING COUNT(DISTINCT ac.codice_corsa) = (SELECT COUNT(DISTINCT
07 | ac1.codice_corsa))
```

```

06 |     FROM LINEE l1 JOIN ORARI_LINEE o11 ON (l1.codice_linea = o11.
07 |         codice_linea)
08 |         JOIN ATTUAZIONI_CORSE ac1 ON (o11.codice_orario = ac1.
09 |             codice_orario)
10 |             WHERE l1.codice_linea = l.codice_linea
11 |             AND ac1.data IN (SELECT ac2.data
12 |                 FROM attuazioni_corse ac2
13 |                 JOIN controlli c ON (c.codice_corso = ac2.codice_corso)
14 |                 JOIN multe m ON (m.codice_corso = ac2.codice_corso)
15 |                     GROUP BY ac2.data
16 |                     HAVING COUNT(DISTINCT m.codice_multa) <= 10 AND COUNT(
17 |                         DISTINCT c.codice_corso) > 5);

```

Listing 4.12: Query to Find Lines Matching Fine/Control Conditions on All Trips

#### 14) Visualizzazione delle linee con il maggior tempo di percorrenza

```

01 | SELECT codice_linea, tempo_percorrenza
02 | FROM LINEE
03 | ORDER BY tempo_percorrenza DESC
04 | LIMIT 1

```

Listing 4.13: Query for Line with Maximum Tempo Percorrenza

#### 15) Estrazione della linea con più hub mobilità lungo il percorso

```

01 | SELECT COUNT(DISTINCT H.codice_hub) AS num_hub, TRA.codice_linea
02 | FROM HUB_MOBILITA H
03 | JOIN FERMATE F ON H.codice_fermata = F.codice_fermata
04 | JOIN TRAGITTI TRA ON (TRA.arrivo_codice_fermata = F.codice_fermata
05 | ) OR (TRA.partenza_codice_fermata = F.codice_fermata)
06 | WHERE F.codice_fermata IN
07 |     (SELECT partenza_codice_fermata
08 |      FROM TRAGITTI
09 |      WHERE codice_linea = TRA.codice_linea)
10 | OR F.codice_fermata IN
11 |     (SELECT arrivo_codice_fermata
12 |      FROM TRAGITTI
13 |      WHERE codice_linea = TRA.codice_linea)
14 | GROUP BY TRA.codice_linea
15 | ORDER BY num_hub DESC
16 | LIMIT 1;

```

#### 16) Media di soldi spesi in multe per persona

In questa operazione vengono considerate tutte le persone presenti nel sistema.

```

01 | SELECT AVG(COALESCE(m.importo, 0)) AS media_soldi
02 |     FROM PERSONE p
03 |     LEFT JOIN MULTe m ON p.documento = m.documento

```

Listing 4.14: Query for Average Fine Amount Grouped by Documento

#### 17) Visualizzazione delle aziende che non hanno effettuato nessuna manutenzione nell'ultimo mese

```

01 | SELECT a.*  

02 | FROM AZIENDE a  

03 | WHERE NOT EXISTS (  

04 |     SELECT 1  

05 |     FROM MANUTENZIONI_MEZZI mm  

06 |     WHERE mm.p_iva = a.p_iva  

07 |     AND mm.data_inizio >= CURRENT_DATE - INTERVAL 1 MONTH  

08 | )  

09 | AND NOT EXISTS (  

10 |     SELECT 1  

11 |     FROM MANUTENZIONI_LINEE ml  

12 |     WHERE ml.p_iva = a.p_iva  

13 |     AND ml.data_inizio >= CURRENT_DATE - INTERVAL 1 MONTH  

14 | );

```

Listing 4.15: Query for Companies Without Maintenance in Last Month

### 18) Visualizzazione delle fermate in cui è presente almeno un hub mobilità contenente tutti i tipi di servizi green

Quest'operazione corrisponde a una divisione: prendiamo gli hub che hanno un numero di servizi green pari al totale dei servizi green esistenti.

Una volta estratti gli hub, scegliamo le fermate che sono connesse ad almeno uno di questi.

```

01 | SELECT DISTINCT *  

02 | FROM FERMATE F  

03 | JOIN HUB_MOBILITA H ON F.codice_fermata = H.codice_fermata  

04 | WHERE H.codice_hub IN  

05 |     (SELECT H.codice_hub  

06 |      FROM HUB_MOBILITA H  

07 |      JOIN CONTENUTI C ON H.codice_hub = C.codice_hub  

08 |      GROUP BY H.codice_hub  

09 |      HAVING COUNT(DISTINCT C.codice_contenuto) =  

10 |          (SELECT COUNT(DISTINCT codice_contenuto) FROM  

11 |             CONTENUTI_HUB)

```

### 19) Inserimento di una variazione di servizio

Avendo l'attributo ridondante **attiva**, dopo aver inserito la variazione di servizio, questo va modificato. Avendo creato la stored procedure apposita useremo questa.

```

01 | INSERT INTO MANUTENZIONI_LINEE (codice_linea, data_inizio,  

02 |     data_fine, nome, descrizione, p_iva)  

03 | VALUES (?, ?, ?, ?, ?, ?);  

04 | INSERT INTO SOSTITUZIONI (sost_manut_data_inizio,  

05 |     sost_manut_codice_linea, codice_linea)  

06 | VALUES (?, ?, ?);  

07 | aggiorna_attivazione_linee();

```

Listing 4.16: Insert into SOSTITUZIONI and Update LINEE status

### 20) Aggiunta di una tratta a una linea esistente

Data la presenza dell'attributo ridondante **tempo\_percorrenza** su LINEE, una volta inserito il tragitto è necessario aggiornare il tempo di percorrenza totale della linea (leggendo quello vecchio e aumentandolo). Sarebbe ottimale eseguire le tre operazioni tramite una **transazione**, così da essere certi che il tempo di percorrenza

rimane coerente con quello reale (ad esempio in caso di errore nell'aggiornamento del tempo, la transazione fallisce e il tragitto non viene aggiunto).

```
01 | INSERT INTO TRAGITTI (partenza_codice_fermata,
    arrivo_codice_fermata, codice_linea, ordine)
02 | VALUES (?, ?, ?, ?);
```

Listing 4.17: Inserimento del tragitto

```
01 | SELECT tempo_percorrenza
02 | FROM LINEE
03 | WHERE codice_linea = ?;
```

Listing 4.18: Lettura vecchio tempo di percorrenza

```
01 | UPDATE LINEE
02 | SET tempo_percorrenza = ?
03 | WHERE codice_linea = ?;
```

Listing 4.19: Aggiornamento del tempo di percorrenza

## 21) Creazione di una nuova linea

Per lo stesso motivo dell'operazione 20 va inizializzato il tempo di percorrenza una volta aggiunte le tratte alla linea. Anche in questo caso sarebbe ottimale eseguire le operazioni in una **transazione**.

```
01 | INSERT INTO LINEE (codice_linea, inizio_validita, fine_validita,
    attiva, codice_tipo_mezzo)
02 | VALUES (?, ?, ?, ?, ?, ?);
```

Listing 4.20: Inserimento linea

```
01 | INSERT INTO TRAGITTI (partenza_codice_fermata,
    arrivo_codice_fermata, codice_linea, ordine)
02 | VALUES (?, ?, ?, ?);
```

Listing 4.21: Inserimento tragitti *Da ripetere n volte, con n = num\_tragitti*

```
01 | UPDATE LINEE
02 | SET tempo_percorrenza = ?
03 | WHERE codice_linea = ?;
```

Listing 4.22: Update tempo di percorrenza della linea

# **Capitolo 5**

## **Progettazione dell'Applicazione**