

Progettazione Database: Azienda di Trasporti

Relazione di Progetto

Andrea Girlando

24 febbraio 2026

Indice

1	Progettazione Concettuale	2
1.1	Descrizione in linguaggio naturale del problema	2
1.2	Glossario dei termini	2
1.3	Analisi delle operazioni principali	3
1.4	Schema E-R	4
1.5	Descrizione in linguaggio naturale dello schema ER	5
1.6	Dizionario dei dati	6
1.7	Vincoli	8
2	Progettazione logica	9
2.1	Schema logico	9
2.2	Discussione dello schema logico	9
2.3	Volumi delle tabelle	10
2.4	Frequenza delle operazioni	10
3	Progettazione fisica	11
3.1	Analisi dei dati ridondanti	11
3.2	Analisi degli indici	12
3.3	Verifica delle forme normali	14
3.4	Conclusioni sulla progettazione fisica	18
4	Utilizzo degli strumenti AI	18

1 Progettazione Concettuale

In questa sezione verrà illustrato il modello concettuale del database, illustrando i ragionamenti e le scelte fatte per una stesura ottimale dello schema Entità-Relazione (E-R).

1.1 Descrizione in linguaggio naturale del problema

L'obiettivo di questo progetto è informatizzare e ottimizzare la gestione di un'azienda di trasporto pubblico, che opera sia a livello urbano che extraurbano. Si immagini una realtà complessa e dinamica: ogni giorno decine di mezzi, tra autobus e tram, percorrono tratte prestabilite, fermandosi in punti specifici suddivisi per zone tariffarie.

C'è bisogno di tenere traccia non degli orari pianificati, ma soprattutto degli orari effettivi, per poter analizzare la puntualità del servizio e storicizzare i ritardi. Inoltre, il sistema deve gestire tutte le risorse aziendali: il personale viaggiante (gli autisti, organizzati in turni lavorativi), l'intero parco mezzi (monitorando i chilometri percorsi, i rifornimenti di carburante e gli interventi di manutenzione) e, naturalmente, l'interazione con l'utenza. I passeggeri, infatti, acquistano biglietti o abbonamenti, li convalidano a bordo tramite sistemi elettronici e, all'occorrenza, devono poter presentare dei reclami legati a disservizi vissuti durante uno specifico viaggio.

1.2 Glossario dei termini

Per garantire una comprensione univoca del dominio aziendale e delle logiche modellate nello schema concettuale, si definiscono di seguito i termini chiave utilizzati nel progetto:

Linea: L'itinerario ufficiale e prestabilito servito dai mezzi aziendali, identificato solitamente da un numero o nome

Corsa pianificata: La singola istanza teorica di viaggio lungo una Linea, definita da un orario di partenza fissato a tabellone

Corsa Effettiva: Il viaggio reale che si verifica in una specifica data, associato a un veicolo fisico e a un autista. Permette di tracciare i ritardi.

Fermata: Luogo fisico designato per la salita e la discesa dei passeggeri.

Zona tariffaria: Area geografica in cui è suddivisa la rete di trasporti. Il passaggio tra zone diverse determina il costo del biglietto.

Titolo di viaggio: Il contratto di trasporto (biglietto singolo o abbonamento) che conferisce all'utente il diritto di viaggiare.

Validazione: L'atto di "timbrare" o registrare elettronicamente il proprio Titolo di Viaggio a bordo del mezzo o ai tornelli.

Deviazione: Modifica temporanea al normale percorso di una Linea, tipicamente causata da lavori in corso, incidenti o manifestazioni.

Turno: Il blocco di ore lavorative (o di riposo/ferie) assegnato a un Autista in una determinata giornata.

1.3 Analisi delle operazioni principali

Di seguito le operazioni critiche a database, saranno necessarie insieme ai volumi delle tabelle e alle frequenze per delle scelte progettuali che verranno fatte come la creazione di indici e l'analisi delle ridondanze

Cod.	Descrizione Operazione	Tipo
O1	Inserimento di una nuova Validazione a bordo.	Scrittura
O2	Emissione/Vendita di un nuovo Titolo di Viaggio (Biglietto o Abbonamento) per un Utente.	Scrittura
O3	Ricerca dell'orario di una Corsa Pianificata per una specifica Linea e Fermata.	Lettura
O4	Inserimento di una Corsa Effettiva alla partenza, con registrazione dell'orario reale e del Mezzo.	Scrittura
O5	Estrazione della sequenza ordinata delle Fermate appartenenti a una determinata Linea.	Lettura
O6	Calcolo statistico dell'affollamento (totale validazioni) raggruppato per singola Corsa Effettiva.	Lettura
O7	Inserimento di un Reclamo da parte di un Utente relativo a una specifica Corsa Effettiva.	Scrittura
O8	Report mensile sui ritardi medi accumulati dalle Corse Effettive rispetto al palinsesto, raggruppati per Linea.	Lettura

Tabella 1: Tabella delle principali operazioni a carico del database.

1.4 Schema E-R

Di seguito lo schema ER

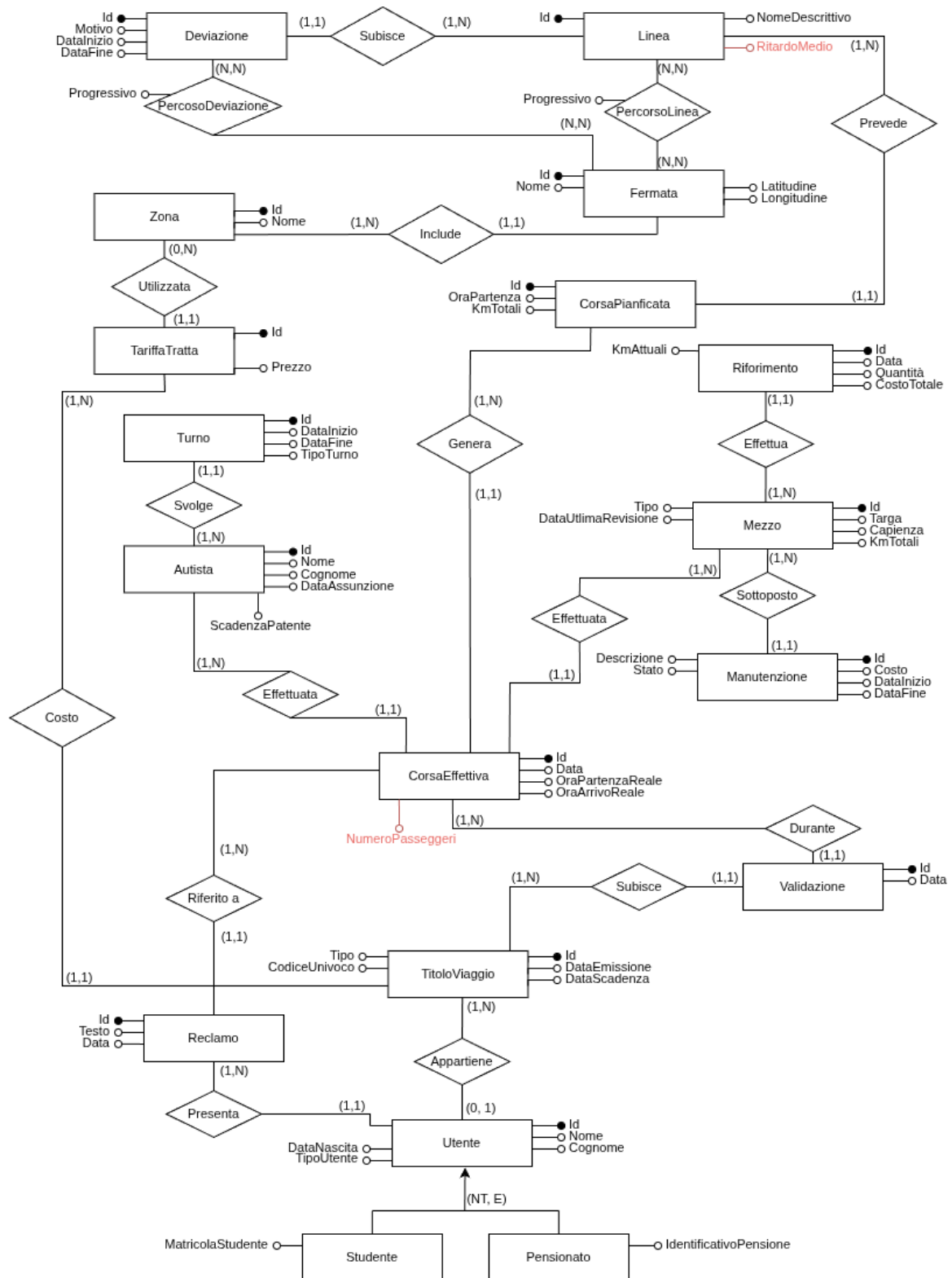


Figura 1: Schema Entità-Relazione (E-R)

1.5 Descrizione in linguaggio naturale dello schema ER

Per creare lo schema ER il lavoro è stato diviso in delle macro-aree collegate tra di loro:

- **Gestione percorsi e linee:** L'entità **Linea** è costituita da un percorso ordinato di **Fermate**. Per gestire imprevisti come manutenzioni o chiusure stradali, è stata prevista l'entità **Deviazione**, che genera un percorso temporaneo composto anch'esso da **Fermate**.
- **Esecuzione e monitoraggio corse:** È stata mantenuta una distinzione netta tra **Corsa Pianificata** (orario teorico) e **Corsa Effettiva** (viaggio reale), utilizzata successivamente per il calcolo dei ritardi. Ad ogni corsa è associata:
 - Un **Mezzo** fisico e un **Autista** che lo guida.
 - La gestione dei **Turni** (riposi, ferie, malattie e straordinari).
 - Il monitoraggio dei costi tramite le relazioni di **Rifornimento** e **Manutenzione** del mezzo.
- **Profilazione utenza:** L'entità **Utente** è stata modellata tramite una gerarchia: partendo da un'anagrafica di base, sono state create le specializzazioni **Studente** e **Pensionato** per gestire gli attributi necessari alle agevolazioni tariffarie.
- **Sistema di bigliettazione e feedback:** Gli utenti acquistano un **Titolo di Viaggio**, il cui costo è calcolato in base alla **Tariffa Tratta** definita dalle **Zone** di partenza e arrivo. Il processo si conclude con la **Validazione** del titolo durante una **Corsa Effettiva**, con la possibilità per l'utente di inoltrare un **Reclamo** specifico per quel viaggio.
- **Dati ridondanti:** Gli attributi in rosso di **CorsaEffettiva** e **Linea** sono dei dati ridondanti la cui utilità verrà discussa durante la progettazione fisica

1.6 Dizionario dei dati

Per maggiore chiarezza, si riporta un riepilogo strutturato delle entità e delle relazioni principali che compongono lo schema.

Dizionario delle entità

Entità	Descrizione	Attributi
Linea	Tratta ufficiale gestita dall'azienda.	Id , NomeDescrittivo, RitardoMedio
Fermata	Luogo fisico di sosta per salita e discesa.	Id , Nome, Tipo, Latitudine, Longitudine
Zona	Area geografica per il calcolo tariffario.	Id , Nome
TariffaTratta	Prezzo calcolato in base alle zone attraversate.	Id , Prezzo
Deviazione	Variazione temporanea di un percorso di linea.	Id , Motivo, DataInizio, DataFine
CorsaPianificata	Orario teorico previsto dal palinsesto.	Id , OrarioPartenza
CorsaEffettiva	Istanza reale del viaggio in una determinata data.	Id , Data, OraPartenzaReale, OraArrivoReale, NumeroPasseggeri
Mezzo	Veicolo fisico impiegato per le corse.	Id , Targa, Capienza, KmTotali, Tipo, DataUltimaRevisione
Rifornimento	Registrazione dell'immissione di carburante/energia.	Id , Data, Quantità, CostoTotale
Manutenzione	Intervento tecnico effettuato su un mezzo.	Id , Costo, DataInizio, DataFine, Descrizione, Stato
Autista	Dipendente addetto alla conduzione del mezzo.	Id , Nome, Cognome, DataAssunzione, ScadenzaPatente
Turno	Frazione lavorativa assegnata al personale.	Id , DataInizio, DataFine, TipoTurno
TitoloViaggio	Biglietto o abbonamento acquistato.	Id , CodiceUnivoco, Tipo, DataEmissione, DataScadenza
Validazione	Timbratura o check-in elettronico a bordo.	Id , Data
Utente	Cliente fruitore del servizio (entità padre).	Id , Nome, Cognome, DataNascita, TipoUtente
Studente	Categoria speciale di Utente (sotto-entità).	MatricolaStudente
Pensionato	Categoria speciale di Utente (sotto-entità).	IdentificativoPensione
Reclamo	Segnalazione di disservizio da parte dell'utente.	Id , Testo, Data

Tabella 2: Dizionario completo delle entità del sistema.

Dizionario delle relazioni

Relazione	Entità Coinvolte	Descrizione e Attributi
PercorsoLinea	Linea, Fermata	Ordine delle fermate della linea (Attr: <i>Progressivo</i>).
PercorsoDeviazione	Deviazione, Fermata	Ordine delle fermate alternative (Attr: <i>Progressivo</i>).
Subisce	Linea, Deviazione	Associa una variazione di percorso a una Linea.
Include	Zona, Fermata	Indica a quale zona tariffaria appartiene una fermata.
Utilizzata	Zona, TariffaTratta	Lega le zone (partenza/arrivo) alla tariffa calcolata.
Costo	TariffaTratta, TitoloViaggio	Determina il costo del titolo in base alla tariffa.
Prevede	Linea, CorsaPianificata	Lega la linea ai suoi orari teorici a tabellone.
Genera	CorsaPianificata, CorsaEffettiva	Lega l'orario astratto ai viaggi fisicamente svolti.
Effettua	Mezzo, Rifornamento	Registra i rifornimenti di un mezzo (Attr: <i>KmAttuali</i>).
Sottoposto	Mezzo, Manutenzione	Associa gli interventi tecnici al rispettivo veicolo.
Effettuata (Mezzo)	Mezzo, CorsaEffettiva	Specifica quale veicolo ha svolto il viaggio.
Svolge	Turno, Autista	Associa gli autisti ai rispettivi turni lavorativi.
Effettuata (Autista)	Autista, CorsaEffettiva	Specifica quale autista ha guidato in un dato viaggio.
Appartiene	TitoloViaggio, Utente	Collega un biglietto o abbonamento al suo acquirente.
Subisce	TitoloViaggio, Validazione	Associa le timbrature fisiche al titolo acquistato.
Durante	Validazione, CorsaEffettiva	Contesta la validazione all'interno di una corsa specifica.
Riferito a	CorsaEffettiva, Reclamo	Collega un disservizio segnalato al viaggio esatto.
Presenta	Utente, Reclamo	Identifica l'utente che ha effettuato la segnalazione.

Tabella 3: Dizionario completo delle Relazioni del sistema.

1.7 Vincoli

Di seguito i vincoli non esprimibili tramite lo schema ER

- **Vincoli di dominio:** Molti attributi accettano solo un set predefinito di valori
 - **Utente:** L'attributo *TipoUtente* accetta solo i valori 'Studente', 'Pensionato', 'Standard'.
 - **TitoloViaggio:** L'attributo *Tipo* accetta solo i valori 'Abbonamento', 'Biglietto'.
 - **Mezzo:** L'attributo *Tipo* accetta solo i valori 'Bus', 'Tram'.
 - **Manutenzione:** L'attributo *Stato* accetta solo i valori 'Finita', 'Iniziata'.
 - **Turno:** L'attributo *TipoTurno* accetta solo i valori 'Lavoro', 'Riposo', 'Ferie', 'Malattia', 'Straordinario'.
- **Vincoli temporali:**
 - **Cronologia Eventi:** Nelle entità **Turno**, **Deviazione** e **Manutenzione**, la *DataFine* deve essere sempre maggiore o uguale alla *DataInizio*.
 - **Orari Corsa:** Nella **CorsaEffettiva**, l'*OraArrivoReale* deve essere strettamente successiva all'*OraPartenzaReale*.
 - **Data Validazione:** La *Data* di una **Validazione** deve rientrare nell'intervallo temporale definito da *DataEmissione* e *DataScadenza* del rispettivo **TitoloViaggio**.
 - **Turni:** Un **Autista** non può avere **Turni** di tipo 'Lavoro' o essere assegnato a una **CorsaEffettiva** in date successive alla propria *ScadenzaPatente*.
- **Vincoli Logici:**
 - **Positività:** Gli attributi di costo (*CostoTotale* e *Quantità* nel **Rifornimento**, *Costo* nella **Manutenzione**, *Prezzo* nella **Tariffa** e *Capienza* nel **Mezzo**) devono avere valori strettamente positivi (> 0).
 - **Consumi e Chilometraggio:** L'attributo *KmAttuali* registrato durante un **Rifornimento** deve essere coerente con i *KmTotali* del **Mezzo** in quel preciso istante.

2 Progettazione logica

In questa sezione verrà illustrato il modello logico del database, inoltre verranno fatte delle stime sui volumi delle tabelle e sulla frequenza delle operazioni

2.1 Schema logico

Di seguito è riportato lo schema relazionale. Le chiavi primarie sono indicate in **grassetto**, mentre le chiavi esterne sono sottolineate.

Utente(**Id**, Nome, Cognome, DataNascita, TipoUtente, MatricolaStudente, IdentificativoPensione)

Zona(**Id**, Nome)

TariffaTratta(**Id**, IdZonaPartenza, IdZonaArrivo, Prezzo)

TitoloViaggio(**Id**, IdUtente, IdTariffa, CodiceUnivoco, Tipo, DataEmissione, DataScadenza)

Linea(**Id**, **RitardoMedio**, NomeDescrittivo)

Fermata(**Id**, IdZona, Nome, Latitudine, Longitudine)

PercorsoLinea(**Id**, IdLinea, IdFermata, Progressivo)

Deviazione(**Id**, IdLinea, Motivo, DataInizio, DataFine)

PercorsoDeviazione(**Id**, IdDeviazione, IdFermata, Progressivo)

CorsaPianificata(**Id**, IdLinea, OraPartenza, KmTotali)

Autista(**Id**, Nome, Cognome, DataAssunzione, DataScadenzaPatente)

Turno(**Id**, IdAutista, DataInizio, DataFine, TipoTurno)

Mezzo(**Id**, Targa, Tipo, Capienza, KmTotali, DataUltimaRevisione)

CorsaEffettiva(**Id**, IdCorsaPianificata, IdAutista, **NumeroPasseggeri**, IdMezzo, Data, OraPartenzaReale, OraArrivoReale)

Manutenzione(**Id**, IdMezzo, Costo, DataInizio, DataFine, Descrizione, Stato)

Rifornimento(**Id**, IdMezzo, Data, Quantità, CostoTotale, KmAttuali)

Validazione(**Id**, IdTitoloViaggio, IdCorsaEffettiva, Data)

Reclamo(**Id**, IdCorsaEffettiva, IdUtente, Testo, Data)

2.2 Discussione dello schema logico

La transizione dallo schema E-R allo schema relazionale ha richiesto alcune scelte progettuali specifiche per garantire l'efficienza:

1. **Risoluzione della Gerarchia Utenti:** Per gestire l'entità padre *Utente* e le sue figlie *Studente* e *Pensionato*, si è optato per il collasso verso l'alto (Single Table Strategy). Tutte le informazioni sono racchiuse nella tabella **Utente**, utilizzando l'attributo *TipoUtente* come discriminatore. I campi specifici (*MatricolaStudente* e *IdentificativoPensione*) ammetteranno valori NULL qualora l'utente non appartenga a quella specifica categoria. Questa scelta evita pesanti e continui JOIN in fase di interrogazione.
2. **Associazioni Multi-a-Molti (N:M):** Le relazioni tra Linea/Fermata e Deviazione/Fermata sono state correttamente tradotte in tabelle associative (**PercorsoLinea** e **Percor-**

soDeviazione). Queste tabelle non solo risolvono la cardinalità, ma ospitano l'attributo fondamentale *Progressivo*, indispensabile per ricostruire l'ordine sequenziale delle fermate.

3. **Attributi ridondanti:** Gli attributi in rosso sono gli attributi ridondanti, il loro utilizzo verrà discusso successivamente

2.3 Volumi delle tabelle

Per le stime successive si avrà bisogno dei volumi delle tabelle che sono state appena definite tramite la progettazione logica, di seguito le stime per una piccola azienda di trasporti:

Tabella	Volume Stimato
Zona	3
TariffaTratta	6
Linea	2
Fermata	15
PercorsoLinea	25
Deviazione	2
PercorsoDeviazione	5
Mezzo	5
Autista	5
Utente	20
TitoloViaggio	30
CorsaPianificata	10
CorsaEffettiva	20
Validazione	60
Reclamo	2
Turno	10
Rifornimento	4
Manutenzione	2

Tabella 4: Volumi delle tabelle previsti per il popolamento del database

2.4 Frequenza delle operazioni

Oltre a stimare il volume dei dati memorizzati nel database, è essenziale quantificare il carico di lavoro quotidiano a cui il sistema sarà sottoposto. Di seguito si presenta la stima della frequenza giornaliera per le principali operazioni, precedentemente definite nella fase di progettazione concettuale.

Cod.	Frequenza (Esecuzioni / Giorno)
O1	30
O2	5
O3	15
O4	10
O5	10
O6	3
O7	1
O8	1

Tabella 5: Stima della frequenza giornaliera delle operazioni

3 Progettazione fisica

3.1 Analisi dei dati ridondanti

Come anticipato durante la progettazione concettuale, sono stati individuati due attributi derivabili e quindi potenzialmente ridondanti: **Ritardo Medio** nell'entità *Linea* e **Numero Passeggeri** nell'entità *Corsa Effettiva*. Per valutarne l'effettiva utilità prestazionale, si effettua di seguito un'analisi dei costi delle operazioni coinvolte, misurati in numero di accessi a disco. Si assume che

- un'operazione di Lettura equivale a $1L$
- un'operazione di Scrittura (S) equivale a $2L$.

Analisi attributo: Ritardo Medio (entità Linea)

Il ritardo medio è derivabile calcolando la differenza tra l'ora di partenza reale e l'ora di partenza pianificata. Le operazioni di carico principali che lo coinvolgono sono:

- **Operazione 04** (Scrittura, freq. 10/giorno): Inserimento di una Corsa Effettiva.
- **Operazione 08** (Lettura, freq. 1/giorno): Report sui ritardi medi raggruppati per Linea.

Caso 1: Mantenimento della ridondanza

- **Op. 04:** Inserimento del nuovo record (1S), lettura dell'idLinea nelle Corse Pianificate (1L), lettura della Linea (1L), modifica e aggiornamento del tempo medio (1S). Costo stimato a esecuzione: $1S + 1L + 1L + 1S = 6L$. Per 10 esecuzioni giornaliere = $60L$.
- **Op. 08:** Lettura diretta del dato dalle 2 Linee. Costo stimato a esecuzione: $1L \times 2 = 2L$. Per 1 esecuzione giornaliera = $2L$.
- **Costo Totale Giornaliero:** $60L + 2L = 62L$.

Caso 2: Assenza di ridondanza (Dato calcolato al volo)

- **Op. 04:** Solo inserimento del nuovo record. Costo stimato a esecuzione: $1S = 2L$. Per 10 esecuzioni giornaliere = $20L$.
- **Op. 08:** Lettura delle Linee (1L), lettura delle Corse Pianificate associate (1L), per ogni corsa pianificata ricerca delle Corse Effettive (media di $\frac{20}{10} = 2$ corse effettive per corsa pianificata, quindi $2L$). Costo per Linea: $(1L + 1L + 2L) = 4L$. Per 2 linee totali: $4L \times 2 = 8L$. Per 1 esecuzione giornaliera = $8L$.
- **Costo Totale Giornaliero:** $20L + 8L = 28L$.

Conclusione: Poiché $28L < 62L$, **non conviene** mantenere l'attributo ridondante *RitardoMedio* nello schema finale.

Analisi attributo: Numero Passeggeri

Il numero di passeggeri è calcolabile dinamicamente contando il numero di validazioni fisiche effettuate per quella specifica corsa. Le operazioni coinvolte sono:

- **Operazione 01** (Scrittura, freq. 30/giorno): Inserimento di una nuova Validazione a bordo.
- **Operazione 06** (Lettura, freq. 3/giorno): Calcolo statistico dell'affollamento per Corsa Effettiva.

Caso 1: Mantenimento della ridondanza

- **Op. 01:** Inserimento della nuova validazione (1S), lettura della Corsa Effettiva (1L), aggiornamento del numero passeggeri (1S). Costo stimato a esecuzione: $1S+1L+1S = 6L$. Per 30 esecuzioni giornaliere = $180L$.
- **Op. 06:** Lettura diretta dell'attributo dalle Corse Effettive (20 totali). Costo a esecuzione: $1L \times 20 = 20L$. Per 3 esecuzioni giornaliere = $60L$.
- **Costo Totale Giornaliero:** $180L + 60L = 240L$.

Caso 2: Assenza di ridondanza

- **Op. 01:** Solo inserimento della nuova validazione. Costo stimato a esecuzione: $1S = 2L$. Per 30 esecuzioni giornaliere = $60L$.
- **Op. 06:** Lettura delle Corse Effettive (1L). Per ognuna delle 20 corse, lettura delle relative validazioni (media di $\frac{60}{20} = 3$ validazioni per corsa, quindi $3L$). Costo a singola esecuzione: $(1L + 3L) \times 20 = 80L$. Per 3 esecuzioni giornaliere = $240L$.
- **Costo Totale Giornaliero:** $60L + 240L = 300L$.

Conclusione: Poiché $240L < 300L$, **conviene** mantenere l'attributo ridondante *NumeroPasseggeri* per ottimizzare i carichi di lettura.

3.2 Analisi degli indici

In questa sezione analizziamo la convenienza della creazione di un indice secondario sulla tabella **Validazione**. Come definito nell'analisi concettuale, questa tabella è soggetta principalmente a due operazioni:

- **Operazione 01 (Scrittura):** Inserimento di una nuova validazione (alta frequenza giornaliera).
- **Operazione 06 (Lettura):** Calcolo statistico dell'affollamento raggruppato per singola *Corsa Effettiva* (bassa frequenza, ma alta mole di record elaborati).

Per ottimizzare l'operazione 06, valutiamo un indice sull'attributo *IdCorsaEffettiva*. Poiché il file dati non è fisicamente ordinato in base a questo attributo (chiave secondaria), ci troviamo nel caso di un **indice denso**.

Per la stima, proiettiamo il sistema in un ambiente di produzione con $N = 500.000$ validazioni, assumendo i seguenti parametri fisici standard:

- Dimensione del blocco ($|B|$): 4096 byte
- Dimensione stimata del record Validazione ($|W|$): 20 byte
- Dimensione della chiave (IdCorsaEffettiva): 4 byte
- Dimensione del puntatore: 4 byte

Costo senza l'uso di indici

Calcoliamo il fattore di blocco dei dati (bfr) e il numero totale di blocchi dati (N_b):

$$bfr = \lfloor \frac{|B|}{|W|} \rfloor = \lfloor \frac{4096}{20} \rfloor = 204 \text{ record/blocco}$$

$$N_b = \lceil \frac{N}{bfr} \rceil = \lceil \frac{500000}{204} \rceil = 2451 \text{ blocchi dati}$$

Senza alcun indice, la ricerca richiederebbe una scansione sequenziale dell'intera tabella per recuperare i record desiderati, per un totale di **2451 accessi a disco**.

Costo con l'uso dell'indice denso

Dimensioniamo ora l'indice. La dimensione di un record indice (I) è data dalla somma della dimensione della chiave e del puntatore:

$$I = 4 + 4 = 8 \text{ byte}$$

Calcoliamo il fattore di blocco dell'indice ($bfr(I)$) e il numero di blocchi dell'indice ($N_b(I)$):

$$bfr(I) = \lfloor \frac{|B|}{I} \rfloor = \lfloor \frac{4096}{8} \rfloor = 512 \text{ record indice/blocco}$$

$$N_b(I) = \lceil \frac{N}{bfr(I)} \rceil = \lceil \frac{500000}{512} \rceil = 977 \text{ blocchi indice}$$

Eseguendo una ricerca binaria sull'indice denso, il numero di accessi necessari (incluso un accesso per recuperare il blocco dati effettivo) è:

$$\text{Accessi} = \lceil \log_2(N_b(I)) \rceil + 1$$

$$\text{Accessi} = \lceil \log_2(977) \rceil + 1 = 10 + 1 = 11 \text{ accessi a disco}$$

Conclusioni sull'indice

Confrontando i **2451 accessi** della scansione sequenziale contro gli **11 accessi** tramite indice, il risparmio prestazionale in fase di interrogazione è netto. Sebbene l'operazione 01 introduca un *overhead* per l'aggiornamento della struttura dell'indice ad ogni nuovo inserimento, i volumi stimati in produzione compensano ampiamente questo costo.

3.3 Verifica delle forme normali

Verifichiamo se lo schema è in **BCNF** (Boyce-Codd Normal Form) o almeno in **3NF**, analizzando le dipendenze funzionali di ciascuna relazione.

Utenti

- **PK:** Id
- **Dipendenze funzionali:**

$$Id \rightarrow Nome, Cognome, DataNascita, TipoUtente, \\ MatricolaStudente, IdentificativoPensione$$

- **Esito:** BCNF
-

Zona

- **PK:** Id
- **Dipendenze funzionali:**

$$Id \rightarrow Nome$$

- **Esito:** BCNF
-

TariffaTratta

- **PK:** Id
- **Dipendenze funzionali:**

$$Id \rightarrow IdZonaPartenza, IdZonaArrivo, Prezzo$$

- **Esito:** BCNF
-

TitoloViaggio

- **PK:** Id
- **Chiave candidata:** CodiceUnivoco
- **Dipendenze funzionali:**

$$Id \rightarrow IdUtente, IdTariffa, CodiceUnivoco, Tipo, DataEmissione, DataScadenza$$

- **Esito:** BCNF
-

Linea

- PK: Id

- Dipendenze funzionali:

$$Id \rightarrow NomeDescrittivo$$

- Esito: BCNF
-

Fermata

- PK: Id

- Dipendenze funzionali:

$$Id \rightarrow IdZona, Nome, Tipo, Latitudine, Longitudine$$

- Esito: BCNF
-

PercorsoLinea

- PK: Id

- Dipendenze funzionali:

$$Id \rightarrow IdLinea, IdFermata, Progressivo$$

- Esito: BCNF
-

Deviazione

- PK: Id

- Dipendenze funzionali:

$$Id \rightarrow IdLinea, Motivo, DataInizio, DataFine$$

- Esito: BCNF
-

PercorsoDeviazione

- PK: Id

- Dipendenze funzionali:

$$Id \rightarrow IdDeviazione, IdFermata, Progressivo$$

- Esito: BCNF
-

CorsaPianificata

- PK: Id
- Dipendenze funzionali:

$$Id \rightarrow IdLinea, OraPartenza, KmTotali$$

- Esito: BCNF
-

Autista

- PK: Id
- Dipendenze funzionali:

$$Id \rightarrow Nome, Cognome, DataAssunzione, DataScadenzaPatente$$

- Esito: BCNF
-

Turno

- PK: Id
- Dipendenze funzionali:

$$Id \rightarrow IdAutista, DataInizio, DataFine, TipoTurno$$

- Esito: BCNF
-

Mezzo

- PK: Id
- Chiave candidata: Targa
- Dipendenze funzionali:

$$Id \rightarrow Targa, Tipo, Capienza, KmTotali, DataUltimaRevisione$$

- Esito: BCNF
-

CorsaEffettiva

- PK: Id
- Dipendenze funzionali (senza l'attributo derivato *NumeroPasseggeri*):

$$Id \rightarrow IdCorsaPianificata, IdAutista, IdMezzo, Data, OraPartenzaReale, OraArrivoReale$$

- **Osservazione:** l'attributo *NumeroPasseggeri* è ridondante, in quanto derivabile dal conteggio delle tuple nella relazione *Validazione*.
 - **Esito:** non rispetta la BCNF né la 3NF.
-

Manutenzione

- **PK:** Id
- **Dipendenze funzionali:**

$$Id \rightarrow IdMezzo, Costo, DataInizio, DataFine, Descrizione, Stato$$

- **Esito:** BCNF
-

Rifornimento

- **PK:** Id
- **Dipendenze funzionali:**

$$Id \rightarrow IdMezzo, Data, Quantità, CostoTotale, KmAttuali$$

- **Esito:** BCNF
-

Validazione

- **PK:** Id
- **Dipendenze funzionali:**

$$Id \rightarrow IdTitoloViaggio, IdCorsaEffettiva, Data$$

- **Esito:** BCNF
-

Reclamo

- **PK:** Id
- **Dipendenze funzionali:**

$$Id \rightarrow IdCorsaEffettiva, IdUtente, Testo, Data$$

- **Esito:** BCNF

3.4 Conclusioni sulla progettazione fisica

Dalle varie analisi fatte durante questa fase abbiamo capito che ha senso creare un indice secondario sull'attributo `IdCorsaEffettiva` e soprattutto dobbiamo mantenere solo l'attributo ridondante *NumeroPasseggeri* perché è l'unico utile tra quelli proposti. Da questo nasce l'implementazione tramite query SQL del vero e proprio database, in allegato a questo documento saranno presenti i seguenti script:

- **CreaDB.sql**: con al suo interno le query di creazione del db, inserimento dei dati e creazione dei trigger per far rispettare i vincoli
- **InserisciDati.sql**: con le operazioni considerate critiche durante la progettazione
- **Operazioni.sql**: con le operazioni considerate critiche durante la progettazione

4 Utilizzo degli strumenti AI

In questo progetto sono stati utilizzando degli strumenti di IA in specifico Gemini 3.1 PRO da interfaccia web per generare la grande quantità di insert necessarie per costruire il dump del db. Di seguito la chat utilizzata: chat con gemini