



BASE DI DATI PER LA GESTIONE DI UN'AZIENDA DI AUTOTRASPORTI

Materia: Base di dati
Docente: Pulvirenti Alfredo
Studente: Bertolami Carmelo
Matricola: 1000030407

Sommario:

Descrizione Del Database:	3
Analisi Dei Requisiti:	4
Glossario Dei Termini:	5
Progettazione Concettuale:	6
Cardinalità Delle Associazioni:	9
Raffinamento Schema Intermedio:	10
Schema Finale:	11
Dizionario Dei Dati:	12
Dizionario Delle Relazioni	13
Progettazione Logica:	14
Tabella delle frequenze e dei volumi:.....	15
Analisi Delle Ridondanze	16
Traduzione Della Base Di Dati.....	17
Implementazione Delle Tabelle.....	18
Implementazioni Operazioni:	24
Trigger	27
Grafico Uml:.....	30

Descrizione del database:

L'azienda di autotrasporti si occupa del trasporto di merci e beni su strada.

Il progetto che proponiamo prevede la creazione di un sistema di gestione efficiente e moderno, che permetta all'azienda di ottimizzare le operazioni di trasporto e di fornire un servizio di qualità ai propri clienti.

L'azienda per la quale creeremo il database è la: "[Celertrasporti](#)", con sede principale a Giammoro (ME). È un'azienda importante a livelli internazionali che opera nel settore degli autotrasporti sin dal 1980 fondata dal Sig. Lucchesi Giuseppe.

Nel progetto è richiesto l'utilizzo di tecnologie avanzate come sensori di localizzazione GPS e di monitoraggio dei veicoli, che permetteranno a chi di competenza di monitorare costantemente la posizione dei mezzi e di ottenere informazioni sulle prestazioni del personale e sull'utilizzo dei veicoli. Questo permetterà di prendere decisioni che migliorino l'efficienza su consumi e qualità del servizio.

IN BREVE:

Il funzionamento è molto semplice, vi è un ufficio che gestisce responsabili logistici autisti e rimorchi, ogni cliente che vuole acquistare si affida all'ufficio di competenza, i quali sono sparsi sul territorio italiano. La consegna della merce avviene tramite i camion che vengono guidati dagli autisti. Essi verranno pagati mensilmente e avranno un compenso aggiuntivo in base alla regione di partenza e di consegna della merce. Infine, in caso di problemi ai mezzi, vi saranno varie officine pronte a ripararli il prima possibile.

Analisi dei requisiti:

1. Dati sulle consegne: Il database dovrà contenere informazioni dettagliate sulle consegne:
data e ora di prenotazione, destinazione, tipo di merce trasportata, quantità, etc.
Questi dati dovranno essere facilmente accessibili e modificabili per supportare la pianificazione e l'organizzazione delle consegne.
2. Dati sulle risorse: dovrà contenere informazioni sulle risorse disponibili, come i veicoli, il personale, etc. Questi dati dovranno essere facilmente accessibili e modificabili per supportare la pianificazione e l'assegnazione delle risorse alle consegne.
3. Monitoraggio delle consegne: Il database dovrà fornire la possibilità di monitorare lo stato delle consegne in tempo reale, inclusi il tempo di consegna previsto e quello effettivo, il tracciamento della posizione dei veicoli, etc.
4. Reportistica: Il database dovrà generare report dettagliati sulle consegne, sull'utilizzo delle risorse e sulle prestazioni del personale, al fine di supportare la decisione e il miglioramento delle attività di trasporto.
5. Integrazione con altri sistemi: Il database dovrà essere in grado di integrarsi con altri sistemi aziendali, come il sistema di gestione delle fatture e delle entrate, per garantire una gestione completa e coerente delle informazioni.
6. Sicurezza dei dati: Il database dovrà essere progettato per garantire la sicurezza e la protezione dei dati, inclusi i meccanismi di autenticazione e autorizzazione, la crittografia dei dati, etc.
7. Facilità d'uso: Il database dovrà essere facile da usare e da navigare, con un'interfaccia intuitiva e user-friendly, per garantire una piena adozione del sistema da parte del personale.

Glossario dei Termini:

Termini	Descrizione	Sinonimi	Legami
<i>Autista</i>	Professionista che guida un veicolo per il trasporto di merci da un luogo a un altro. È responsabile della sicurezza del veicolo e della merce trasportata.	Camionista, Conducente, autotrasportatore, conducente di automezzi.	Ufficio, responsabile_della_logistica, carta(tachigrafo), camion, cliente, Busta-paga,
<i>Responsabili della logistica</i>	Si occupa di organizzare viaggi e percorsi della giornata che devono essere rispettati affinché orari di consegna combacino il più possibile.	Direttore della logistica, manager della logistica, coordinatore della logistica, capo della logistica, amministratore della logistica,	Ufficio, cliente, autista
<i>Ufficio</i>	Luogo locato nella sede principale dell'azienda. Si occupa della parte burocratica	sede, posto di lavoro, ambiente di lavoro	Responsabile della logistica, Busta paga
<i>Officina</i>	Luogo che si occupa della manutenzione dei mezzi.	officina di riparazione, officina di manutenzione,	camion, officina
<i>Carta(tachigrafo)</i>	Data al conducente che deve inserire nell'apposito lettore, che registra partenza e tutte le fermate, serve per tenere conto delle ore di guida e delle ore di sosta.	Scheda	Camion, autista
<i>Camion</i>	Mezzo che viene guidato dall'autista che permette di portare la merce in grande quantità a destinazione	autocarro, camioncino, autotreno, veicolo pesante, mezzo pesante, mezzo di trasporto, automezzo.	Autista
<i>Cliente</i>	Azienda o privato che richiede una materia prima o non.	consumatore, acquirente, destinatario	Responsabile della logistica
<i>Merce</i>	Tipo di prodotto che viene trasportato da un luogo ad un altro	prodotto, articolo, beni, materiale, fornitura,	camion, cliente
<i>stipendio</i>	Compenso mensile che viene fornito all'autista in base al suo livello/categoria, e giorni di lavoro	salario, retribuzione, compensazione	Ufficio, autista
<i>Rimorchio</i>	unità di carico che viene trainata da un camion, al fine di trasportare merci o altri oggetti.	semirimorchio, carrello trainato, rimorchio pesante, cassone rimorchio,	Merce, responsabile della logistica

Progettazione concettuale:

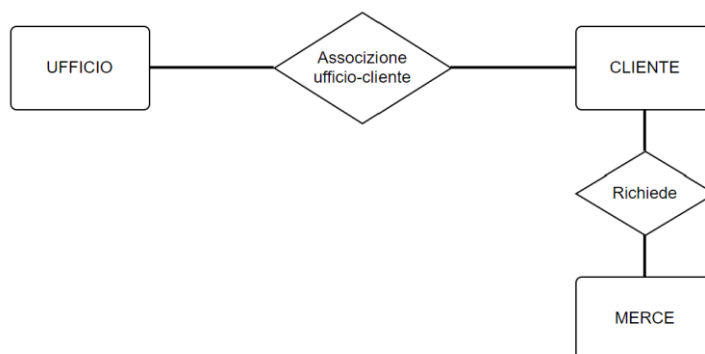
Strategia di progettazione: Si sceglie di creare il database sulla base di un approccio bottom-up. Di seguito verranno riepilogate tutte le sequenze, con i vari raffinamenti, che porteranno all'effettivo modello relazionale.

Schema scheletro iniziale:

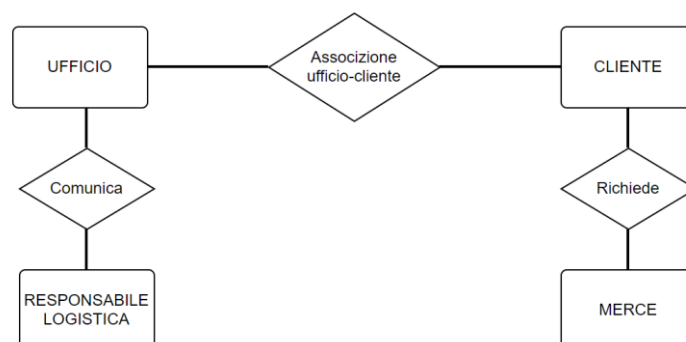
1ª sequenza: vi è la creazione del primo abbozzo di schema scheletro che individua due entità: "Ufficio" e "Cliente", collegate per mezzo della relazione ufficio-cliente.



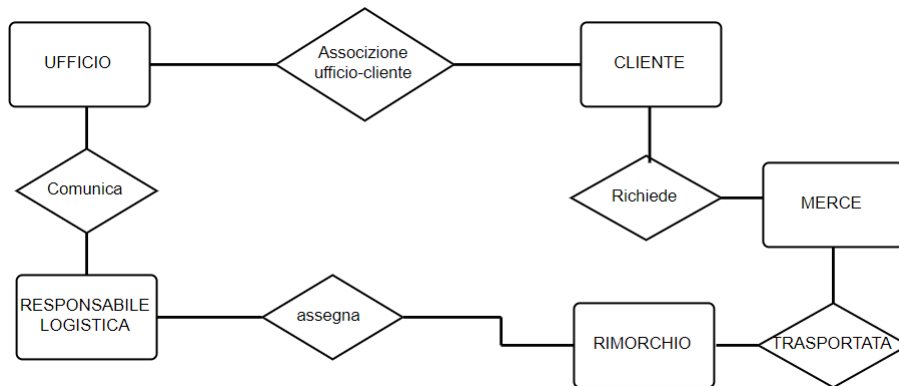
2ª sequenza: aggiungiamo un'altra entità "Merce" molto importante che verrà collegata al "Cliente". La relazione che vi è tra di esse è "richiesta", perché sarà il cliente a richiedere all'azienda una determinata categoria di merce, e di conseguenza sarà dopo competenza dell'azienda occuparsi del trasporto fino a destinazione.



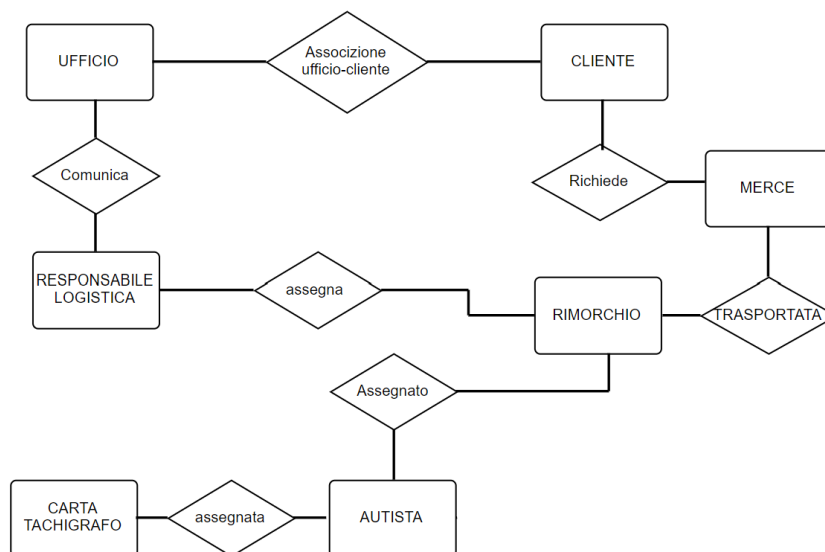
3ª sequenza: viene aggiunta un'altra entità: "Responsabile della logistica". Fondamentale senza la quale l'azienda non potrebbe funzionare, colui che assegna i viaggi a tutti gli autisti e schematizza tutti i viaggi per i giorni successivi, sarà legato per mezzo della relazione "Comunica".



4ª sequenza: giunzione dell'entità "Rimorchio", fondamentale perché tramite il rimorchio si può trasportare la merce fino alla sede del cliente. È collegata per mezzo di due relazioni "Trasporta" con "Merce" e "Assegna" con "Responsabile della logistica".

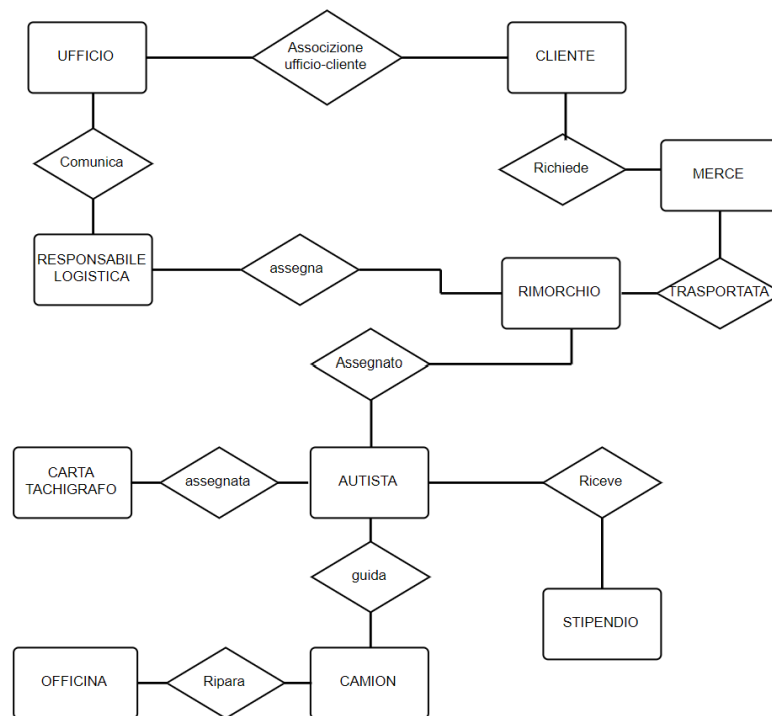


5ª sequenza: aggiunta delle entità "Autista" fulcro che fa da tramite tra l'azienda e il cliente. Mentre la seconda entità "Carta tachigrafo" permette all'autista di guidare secondo alcune regole, essa è indispensabile per ogni autista dal momento che può vedere le ore di lavoro giornaliere rimanenti e le pause che devono essere rispettate per regolarsi di conseguenza.



Si arriva così al primo affinamento di uno schema scheletro:

Dove sono state aggiunte le ultime due entità e relazioni: "Camion", "Officina" con le relative relazioni "guida" e "ripara".



Si passa ora ad elencare tutti vari attributi delle varie entità:

Ufficio: ID_ufficio, n_dipendenti, indirizzo, n_telefono, capo_ufficio

Cliente: ID_cliente, Nome, Cognome, nome_azienza, n_telefono, email

Merce: ID_merce, Tipologia, peso, volume, data_consegna, descrizione

Responsabile della logistica: ID_responsabile, nome, cognome, n_telefono,

Rimorchio: Targa_r, tipologia, anno, capacità_carico, pos_partenza, pos_arrivo

Autista: ID_autista, nome, cognome, età, n_telefono, giorni_lavoro

Carta(tachigrafo): ID_carta, ore_guida, ore_pausa, data_scadenza

Stipendio: ID_stipendio, importo_giornaliero, trasferta

Camion: Targa_c, cilindrata, modello, assicurazione, anno.

Officina: ID_officina, nome, numero_dipendenti, data, indirizzo

(Gli attributi sottolineati sono le "primary key" che identificano l'attributo univocamente)

Diamo uno sguardo al funzionamento di questa struttura:

Il cliente si rivolge all'ufficio per avere merce di diversa tipologia, l'ufficio comunica l'esigenza del cliente al responsabile della logistica. Esso si preoccupa di vedere la posizione del carico destinato al cliente e si rivolge all'autista più vicino al carico per assegnargli il rimorchio.

L'autista munito di carta per il tachigrafo deve rispettare gli orari concordati sia dal datore di

lavoro che dalla legge e la contrattazione collettiva. L'autista ha uno stipendio mensile, e anche con caduta settimanale o mensile viene controllato da un officina autorizzata, lo stato del mezzo.

Cardinalità delle associazioni:

CLIENTE <-> UFFICIO:

- un cliente può rivolgersi ad un ufficio per un carico (0,1) ,
- ad un ufficio si rivolgono più clienti per richiedere merce (1,N).

UFFICIO <-> R. LOGISTICA:

- più uffici si possono rivolgere ad un responsabile della logistica(1,N),
- un responsabile della logistica comunica con più uffici (1,N).

CLIENTE <-> MERCE:

- un cliente richiede più carichi di merce (1,N),
- un carico di merce viene richiesto da un solo cliente (1,1)

MERCE <-> RIMORCHIO:

- un carico di merce viene trasportata in uno solo rimorchio (1,1),
- un rimorchio può trasportare più carichi di merce (1,N).

R. LOGISTICA <-> RIMORCHIO:

- un responsabile assegna più rimorchi (1,N),
- un rimorchio è assegnato da un solo responsabile della logistica (1,1).

RIMORCHIO <-> AUTISTA:

- un rimorchio può essere assegnato a più autisti in una giornata (0,N),
- ad un autista vengono assegnati più rimorchi in una giornata (1,N).

AUTISTA <-> CARTA(TACHIGRAFO):

- ad un autista viene assegnata una e una sola carta fino alla sua scadenza (1,1),
- la carta viene assegnata ad un solo autista (1,1)

AUTISTA <-> STIPENDIO:

- un autista riceve un solo stipendio per la sua fascia (1,1),
- uno stipendio della stessa fascia è ricevuto da più autisti (1,N)

AUTISTA <-> CAMION:

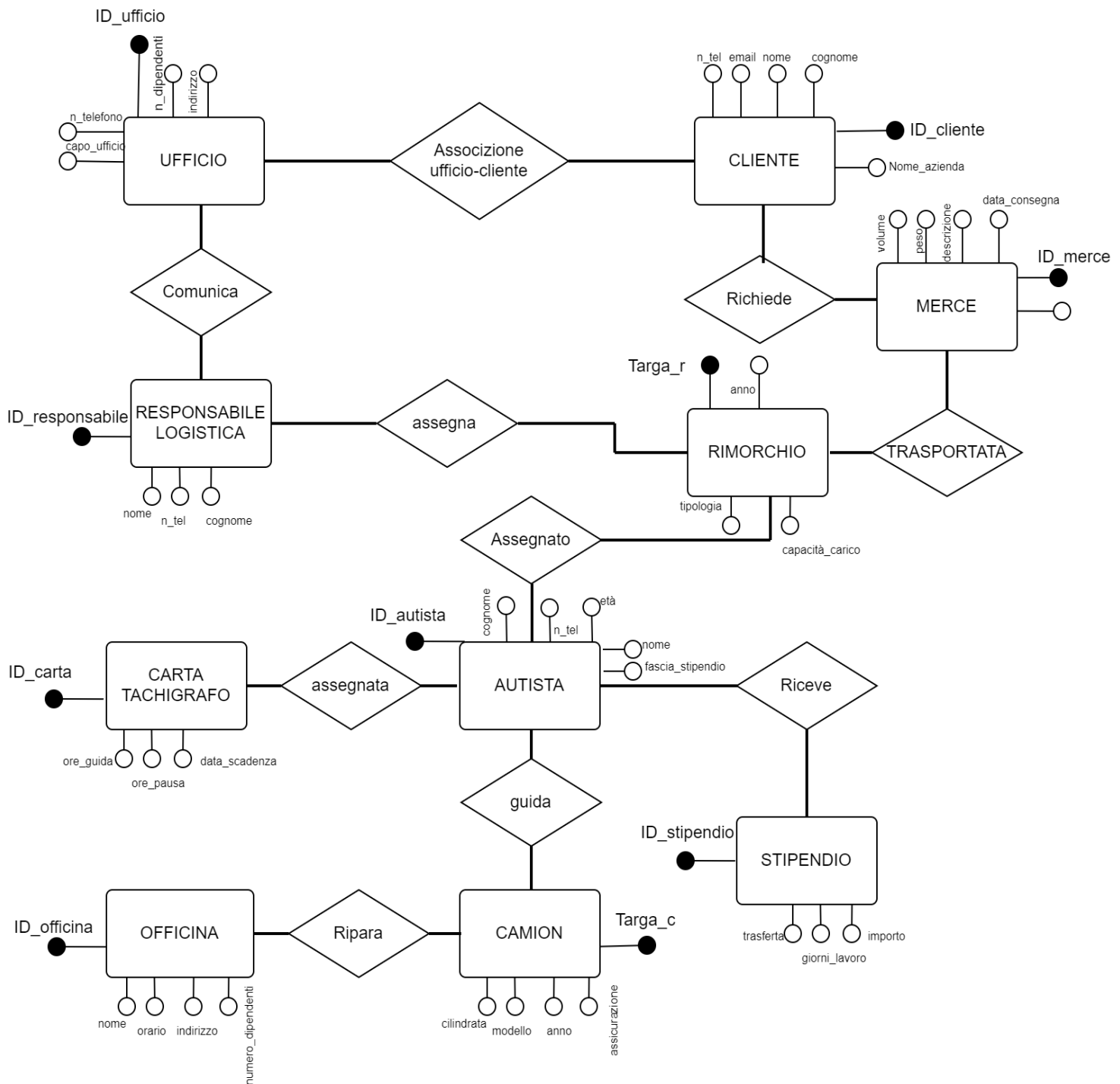
- un autista guida un solo camion (1,1),
- un camion viene guidato da un solo autista (1,1).

OFFICINA <-> CAMION:

- un officina ripara più camion (1,N),
- un camion può essere riparato da più officine(0,N).

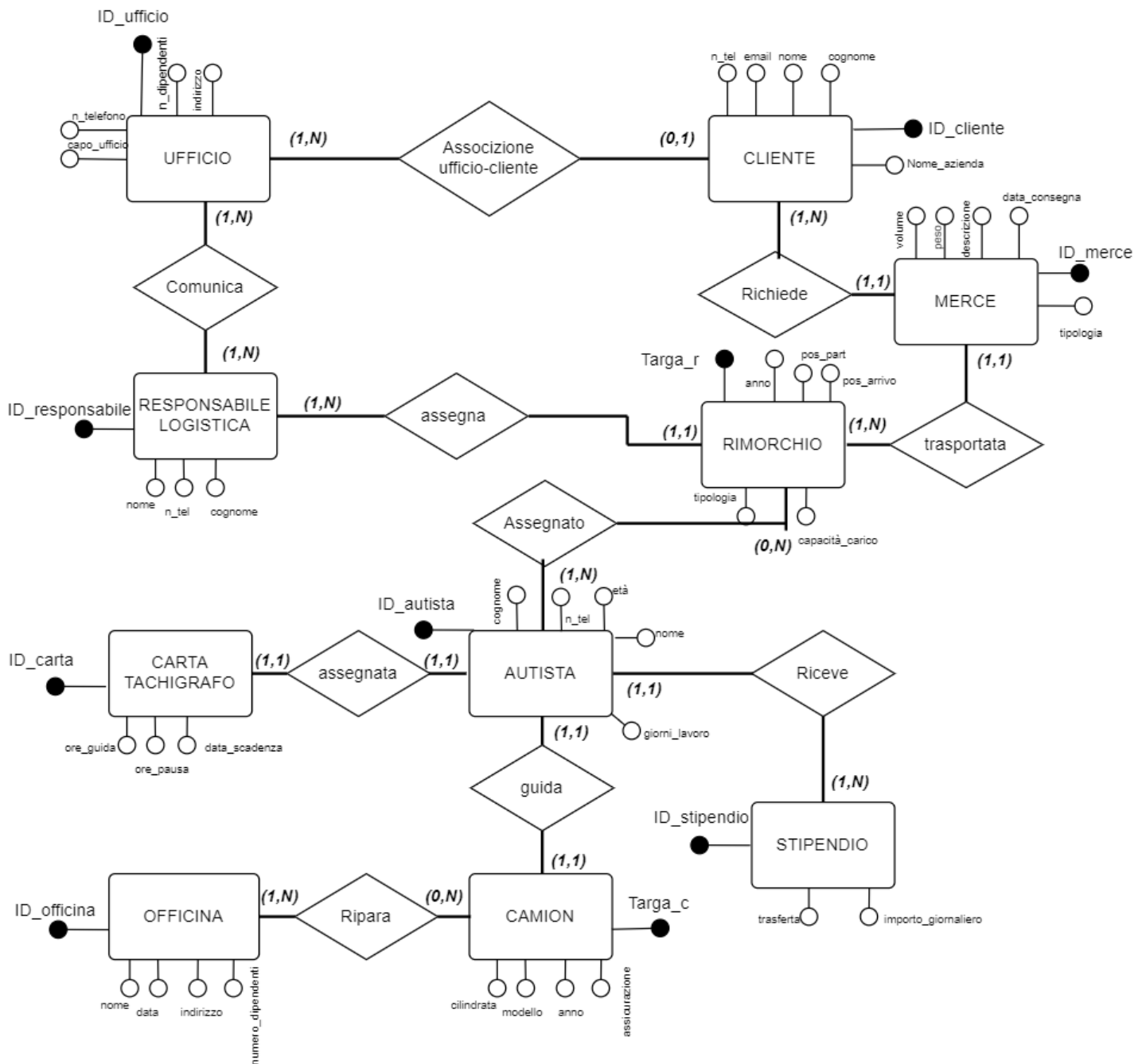
Raffinamento schema intermedio:

sono stati inseriti tutti gli attributi all'interno dello schema E/R:



SCHEMA FINALE:

A seguito della schematizzazione completa e delle rimozioni di alcuni attributi ridondanti del modello in questione, il sarà:



DIZIONARIO DEI DATI:

Entità	Descrizione	Attributi	identificatore
<i>Ufficio</i>	sede fisica che viene contattata da un cliente per la richiesta di una tipologia di merce	n_dipendenti, indirizzo, numero di telefono, capo_ufficio	ID_ufficio
<i>Cliente</i>	persona che richiede la merce per conto dell'azienda o privato	Nome, Cognome, nome_azienza, numero_t, e-mail	ID_cliente
<i>Merce</i>	Tipologia di materiale/materia che viene trasportata	Tipologia, peso, volume, data_consegna, descrizione	ID_merce
<i>Responsabile logistica</i>	Persona incaricata dall'azienda per gestire e far combaciare gli orari e il lavoro da assegnare agli autisti	nome, cognome, numero_t,	ID_responsabile
<i>Rimorchio</i>	Unità mobile che viene utilizzato per trasportare merci o materiali	tipologia, anno, capacità_carico, pos_part, pos_arrivo	Targa_R
<i>Carta(tachigrafo)</i>	Dispositivo che registra il tempo di guida e di riposo del conducente del mezzo	ore_guida, ore_pausa, data_scadenza	ID_carta
<i>Autista</i>	Persona che guida un veicolo sulla base di una licenza particolare	nome, cognome, età, numero_t, giorni_lavoro	ID_autista
<i>Stipendio</i>	Compenso mensile che viene dato all'autista per aver svolto il proprio lavoro	Importo_giornaliero, trasferta	ID_stipendio
<i>Camion</i>	Mezzo che permette di trasportare grandi carichi	cilindrata, modello, assicurazione, anno.	Targa_c
<i>Officina</i>	Luogo in cui è possibile riparare il mezzo danneggiato	nome, numero_dipendenti, data, indirizzo	ID_officina

DIZIONARIO DELLE RELAZIONI

Relazione	Descrizione	Componenti	attributi
<i>Associazione ufficio-cliente</i>	Cliente si rivolge all'ufficio	Ufficio (1,N) Cliente (0,1)	ID-ufficio, ID_cliente
<i>comunica</i>	L'ufficio consegna informazioni sul lavoro acquisito	Ufficio (1,N) Responsabile logistica(1,N)	ID_ufficio, ID_responsabile
<i>Richiede</i>	Cliente che chiama per avere un carico di materiale che desidera	Cliente (1,N) Merce (1,1)	ID_cliente, ID_merce
<i>Trasportata</i>	La merce viene caricata nell'unità mobile per essere consegnata	Merce (1,1) Rimorchio (1,N)	ID_merce, Targa_r
<i>Assegna</i>	Responsabile prende nota del rimorchio che deve essere consegnato	Responsabile logistica (1,N) Rimorchio (1,1)	ID_responsabile, Targa_r
<i>Assegnato</i>	Il rimorchio viene preso in carico da un autista che lo dovrà consegnare	Rimorchio (0,N) Autista (1,N)	Targa_r, ID_autista
<i>Assegnata</i>	La carta viene data ad inizio contratto all'autista che vuole lavorare in quell'azienda	Carta(tachigrafo) (1,1) Autista (1,1)	ID_autista, ID_carta
<i>Riceve</i>	Lo stipendio viene dato mensilmente all'autista per il lavoro svolto	Autista (1,1) Stipendio (1,N)	ID_autista, ID_stipendio
<i>Guida</i>	l'autista abilitato viene incaricato e svolge il lavoro principale	Autista (1,1) Camion(1,1)	ID_autista, Targa_c
<i>Ripara</i>	L'officina viene incaricata di sistemare parti danneggiate del mezzo	Camion(0,N) Officina(1,N)	Targa_c, ID_officina

Progettazione Logica

Dopo la stesura dello schema concettuale del progetto, si prosegue con una stima dei dati e delle operazioni da effettuare sul database.

Tabella volumi:

CONCETTO	TIPO	VOLUME
<i>Ufficio</i>	E	15
<i>Cliente</i>	E	1000
<i>Merce</i>	E	50000
<i>Responsabile logistica</i>	E	40
<i>Rimorchio</i>	E	20000
<i>Carta(tachigrafo)</i>	E	50
<i>Autista</i>	E	50
<i>Stipendio</i>	E	6
<i>Camion</i>	E	60
<i>Associazione ufficio-cliente</i>	R	700
<i>comunica</i>	R	200
<i>Richiede</i>	R	15/mese
<i>Trasportata</i>	R	2500
<i>Assegna</i>	R	200
<i>Assegnato</i>	R	800
<i>Assegnata</i>	R	50
<i>Riceve</i>	R	50
<i>Guida</i>	R	50
<i>Ripara</i>	R	15/mese

TABELLA DELLE FREQUENZE E DEI VOLUMI:

Operazione	Descrizione	Frequenza	Tipo
Lista consegna merce	Quanti carichi sono stati consegnati ad un cliente	1 /settimana	I
Registrazione nuovo cliente	Inserimento dati nuovo cliente	15/mese	I
Registrazione peso merce	Prima di ogni partenza si pesa un carico	35/giorno	I
Registrazione manutenzione camion "per targa"	numero di volte che il camion viene portato in officina	15/mese	I
Lista responsabili su ID responsabile	Viene stampato la lista dei responsabili che lavorano nell'ufficio	1/settimana	I
Stipendio mensile di tutti gli autisti	Viene stampato l'id degli autisti e il loro stipendio mensile	1/settimana	I
controllo giorni di lavoro di ogni dipendente	Numero di giorni effettivo di lavoro	1/mese	B
Controllo ore guida rimanenti tramite scheda tachigrafo	Viene visualizzato il numero di ore settimanale di guida del dipendente che ha superato le 50 ore (avranno diritto ad una pausa sosta del fine settimana più lunga)	1/settimana	I
Lista dipendenti "anziani"	Vengono stampati i dipendenti che hanno un'età maggiore di 60 (dovranno rinnovare la patente più spesso)	6/mese	I
Registrazione nuovo camion tramite targa	Viene acquistato un nuovo camion e deve essere inserito nei registri dell'azienda	3/anno	B
Lista clienti tramite ID	lista dei clienti che hanno cognome " Pulvirenti " e nome che inizia per 'A'	2/trimestre	B
Modifica dipendenti	Viene aggiunto un dipendente per ogni nuova assunzione	2/anno	B
lista camion	Viene stampata la lista dei camion che hanno per targa A___Z	4/anno	B
Conteggio Numero di consegne fatte da un autista	Si conta il numero di rimorchi che un'autista consegna a destinazione.	4/anno	B

ANALISI DELLE RIDONDANZE

Grazie all'approccio sopra utilizzato, ai numerosi raffinamenti e ad un'approfondita analisi iniziale di racconta dei dati necessari alla sua elaborazione e realizzazione, si è giunta alla produzione di una base di dati senza ridondanze. Tuttavia, si può provare ad introdurre una ridondanza al fine di ottimizzare alcune operazioni sopra proposte.

Aggiungeremo infatti l'attributo: numero_scheda su autista al fine di incrementare "teoricamente" l'operazione O15:

Analisi con ridondanza	Analisi senza ridondanza
O14: Conteggio Numero di consegne fatte da un autista. 1L in autista = $1L \times 4 = 4L$	O14: Conteggio Numero di consegne fatte da un autista. 400L autista = $400L \times 4 = 1600L$
4L annuali	1600L annuali

Di seguito a quanto appena analizzato, si sceglie quindi di non introdurre la ridondanza, anche se alleggerirebbe di molto il carico dell'operazione.

Traduzione Della base di dati

Si assuma che gli attributi evidenziati siano le chiavi primarie, invece quelli sottolineati, le chiavi esterne:

Ufficio (**ID_ufficio**, *n_dipendenti*, *indirizzo*, *n_telefono*, *capo_ufficio*)

Cliente (**ID_cliente**, *Nome*, *Cognome*, *nome_azienza*, *n_telefono*, *email*, ID_ufficio)

Merce (**ID_merce**, *Tipologia*, *peso*, *volume*, *data_consegna*, *descrizione*, ID_cliente, Targa_r)

Responsabile della logistica (**ID_responsabile**, *nome*, *cognome*, , *n_telefono*)

Rimorchio (**Targa_r**, *tipologia*, *anno*, *capacità_carico*, *pos_partenza*, *pos_arrivo*,
ID_responsabile)

Autista (**ID_autista**, *nome*, *cognome*, *età*, *n_telefono*, *giorni_lavoro*, ID_stipendio)

Carta(tachigrafo) (**ID_carta**, *ore_guida*, *ore_pausa*, *data_scadenza*, ID_autista)

Stipendio (**ID_stipendio**, *importo_giornaliero*, *trasferta*)

Camion (**Targa_c**, *cilindrata*, *modello*, *assicurazione*, *anno*, ID_autista)

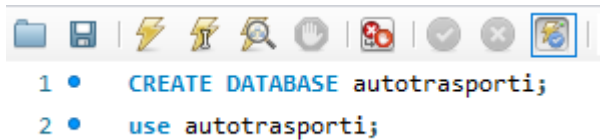
Officina (**ID_officina**, *nome*, *numero_dipendenti*, *data*, *indirizzo*)

Comunicazioni(Ufficio_ID_ufficio,responsabile_logistica_ID_responsabile)

Assegnazioni_rimorchi(rimorchio_Targa_R, Autista_ID_autista)

Manutenzione(Camion_Targa_c, officina_ID_officina)

IMPLEMENTAZIONE DELLE TABELLE



Creazione tramite comandi e sintassi di Mysql workbench del database: Autotrasporti.

Seguiranno le istruzioni che hanno permesso la creazione delle tabelle:

TABELLA UFFICIO:

```
-- Table `autotrasporti`.`Ufficio`
-----
DROP TABLE IF EXISTS `autotrasporti`.`Ufficio` ;

CREATE TABLE IF NOT EXISTS `autotrasporti`.`Ufficio` (
  `ID_ufficio` INT NOT NULL COMMENT 'tabella ufficio',
  `indirizzo` VARCHAR(45) NOT NULL,
  `n_dipendenti` INT NOT NULL,
  `n_telefono` INT NOT NULL,
  `capo_ufficio` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`ID_ufficio`),
  UNIQUE INDEX `ID_ufficio_UNIQUE` (`ID_ufficio` ASC))
ENGINE = InnoDB;
```

```
ALTER TABLE `autotrasporti`.`ufficio`
CHANGE COLUMN `n_telefono` `n_telefono` VARCHAR(255) NOT NULL ;
```

TABELLA CLIENTE:

```
-- Table `autotrasporti`.`Cliente`
-----
DROP TABLE IF EXISTS `autotrasporti`.`Cliente` ;

CREATE TABLE IF NOT EXISTS `autotrasporti`.`Cliente` (
  `ID_cliente` INT NOT NULL COMMENT 'tabella cliente',
  `nome_azienza` VARCHAR(45) NOT NULL COMMENT 'tabella cliente',
  `nome` VARCHAR(45) NOT NULL,
  `cognome` VARCHAR(45) NOT NULL,
  `email` VARCHAR(45) NOT NULL,
  `n_telefono` VARCHAR(255) NOT NULL,
  `Ufficio_ID_ufficio` INT NOT NULL,
  PRIMARY KEY (`ID_cliente`),
  UNIQUE INDEX `ID_cliente_UNIQUE` (`ID_cliente` ASC),
  INDEX `fk_Cliente_Ufficio_idx` (`Ufficio_ID_ufficio` ASC),
  CONSTRAINT `fk_Cliente_Ufficio`
    FOREIGN KEY (`Ufficio_ID_ufficio`)
    REFERENCES `autotrasporti`.`Ufficio` (`ID_ufficio`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

Tabella responsabile logistica:

```
-- -----  
-- Table `autotrasporti`.`responsabile_logistica`  
-- -----  
DROP TABLE IF EXISTS `autotrasporti`.`responsabile_logistica` ;  
  
CREATE TABLE IF NOT EXISTS `autotrasporti`.`responsabile_logistica` (  
  `ID_responsabile` INT NOT NULL COMMENT 'tabella responsabile logistica',  
  `nome` VARCHAR(45) NOT NULL,  
  `cognome` VARCHAR(45) NOT NULL,  
  `n_telefono` INT NOT NULL,  
  PRIMARY KEY (`ID_responsabile`),  
  UNIQUE INDEX `ID_responsabile_UNIQUE` (`ID_responsabile` ASC))  
ENGINE = InnoDB;
```

```
ALTER TABLE `autotrasporti`.`responsabile_logistica`  
CHANGE COLUMN `n_telefono` `n_telefono` VARCHAR(45) NOT NULL ;
```

Tabella rimorchio:

```
-- -----  
-- Table `autotrasporti`.`Rimorchio`  
-- -----  
DROP TABLE IF EXISTS `autotrasporti`.`Rimorchio` ;  
  
CREATE TABLE IF NOT EXISTS `autotrasporti`.`Rimorchio` (  
  `Targa_r` VARCHAR(255) NOT NULL COMMENT 'tabella rimorchio',  
  `anno` DATE NOT NULL,  
  `pos_partenza` VARCHAR(45) NOT NULL,  
  `pos_arrivo` VARCHAR(45) NOT NULL,  
  `capacità_carico` INT NOT NULL,  
  `tipologia` INT NOT NULL,  
  `responsabile_logistica_ID_responsabile` INT NOT NULL,  
  PRIMARY KEY (`Targa_r`),  
  UNIQUE INDEX `Targa_r_UNIQUE` (`Targa_r` ASC),  
  INDEX `fk_Rimorchio_responsabile_logistica1_idx` (`responsabile_logistica_ID_responsabile` ASC),  
  CONSTRAINT `fk_Rimorchio_responsabile_logistica1`  
    FOREIGN KEY (`responsabile_logistica_ID_responsabile`)  
    REFERENCES `autotrasporti`.`responsabile_logistica` (`ID_responsabile`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

Tabella merce:

```
-- Table `autotrasporti`.`Merce`
--
DROP TABLE IF EXISTS `autotrasporti`.`Merce` ;

CREATE TABLE IF NOT EXISTS `autotrasporti`.`Merce` (
  `ID_merce` INT NOT NULL COMMENT 'tabella merce',
  `tipologia` VARCHAR(45) NOT NULL,
  `data_consegna` DATE NOT NULL,
  `descrizione` TEXT(255) NOT NULL,
  `peso` INT NOT NULL,
  `volume` INT NOT NULL,
  `Cliente_ID_cliente` INT NOT NULL,
  `Rimorchio_Targa_r` VARCHAR(255) NOT NULL,
  PRIMARY KEY (`ID_merce`),
  UNIQUE INDEX `ID_merce_UNIQUE` (`ID_merce` ASC),
  INDEX `fk_Merce_Cliente1_idx` (`Cliente_ID_cliente` ASC),
  INDEX `fk_Merce_Rimorchio1_idx` (`Rimorchio_Targa_r` ASC),
  CONSTRAINT `fk_Merce_Cliente1`
    FOREIGN KEY (`Cliente_ID_cliente`)
    REFERENCES `autotrasporti`.`Cliente` (`ID_cliente`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Merce_Rimorchio1`
    FOREIGN KEY (`Rimorchio_Targa_r`)
    REFERENCES `autotrasporti`.`Rimorchio` (`Targa_r`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

Tabella stipendio:

```
-- Table `autotrasporti`.`stipendio`
--
DROP TABLE IF EXISTS `autotrasporti`.`stipendio` ;

CREATE TABLE IF NOT EXISTS `autotrasporti`.`stipendio` (
  `ID_stipendio` INT(255) NOT NULL COMMENT 'tabella stipendio',
  `trasferta` INT NOT NULL,
  `giorni_lavoro` VARCHAR(255) NOT NULL,
  `importo` INT NOT NULL,
  PRIMARY KEY (`ID_stipendio`))
ENGINE = InnoDB;

ALTER TABLE `autotrasporti`.`stipendio`
DROP COLUMN `giorni_lavoro`,
CHANGE COLUMN `importo` `importo_giornaliero` INT NOT NULL ;
```

Tabella autista:

```
-- -----  
-- Table `autotrasporti`.`Autista`  
-- -----  
  
DROP TABLE IF EXISTS `autotrasporti`.`Autista` ;  
  
ALTER TABLE `autotrasporti`.`autista`  
ADD COLUMN `giorni_lavoro` INT NOT NULL AFTER `n_telefono`;  
  
  `cognome` VARCHAR(45) NOT NULL,  
  `età` INT NOT NULL,  
  `n_telefono` VARCHAR(45) NOT NULL,  
  `stipendio_ID_stipendio` INT(255) NOT NULL,  
PRIMARY KEY (`ID_autista`),  
UNIQUE INDEX `ID_autista_UNIQUE` (`ID_autista` ASC),  
INDEX `fk_Autista_stipendio1_idx` (`stipendio_ID_stipendio` ASC),  
CONSTRAINT `fk_Autista_stipendio1`  
  FOREIGN KEY (`stipendio_ID_stipendio`)  
  REFERENCES `autotrasporti`.`stipendio` (`ID_stipendio`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

Tabella carta tachigrafo:

```
-- -----  
-- Table `autotrasporti`.`carta_tachigrafo`  
-- -----  
  
DROP TABLE IF EXISTS `autotrasporti`.`carta_tachigrafo` ;  
  
CREATE TABLE IF NOT EXISTS `autotrasporti`.`carta_tachigrafo` (  
  `ID_carta` INT NOT NULL COMMENT 'tabella carta_tachigrafo',  
  `ore_guida` INT NOT NULL,  
  `ore_pausa` INT NOT NULL,  
  `data_scadenza` DATE NOT NULL,  
  `Autista_ID_autista` INT NOT NULL,  
PRIMARY KEY (`ID_carta`),  
UNIQUE INDEX `ID_carta_UNIQUE` (`ID_carta` ASC),  
INDEX `fk_carta_tachigrafo_Autista1_idx` (`Autista_ID_autista` ASC),  
CONSTRAINT `fk_carta_tachigrafo_Autista1`  
  FOREIGN KEY (`Autista_ID_autista`)  
  REFERENCES `autotrasporti`.`Autista` (`ID_autista`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

Tabella camion:

```
-- -----  
-- Table `autotrasporti`.`Camion`  
-- -----  
  
DROP TABLE IF EXISTS `autotrasporti`.`Camion` ;  
  
CREATE TABLE IF NOT EXISTS `autotrasporti`.`Camion` (  
  `targa_c` VARCHAR(255) NOT NULL COMMENT 'tabella camion',  
  `cilindrata` VARCHAR(45) NOT NULL,  
  `modello` VARCHAR(45) NOT NULL,  
  `anno` DATE NOT NULL,  
  `assicurazione` VARCHAR(45) NOT NULL,  
  `Autista_ID_autista` INT NOT NULL,  
  PRIMARY KEY (`targa_c`),  
  INDEX `fk_Camion_Autista1_idx` (`Autista_ID_autista` ASC),  
  CONSTRAINT `fk_Camion_Autista1`  
    FOREIGN KEY (`Autista_ID_autista`)  
    REFERENCES `autotrasporti`.`Autista` (`ID_autista`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
ALTER TABLE `autotrasporti`.`camion`  
CHANGE COLUMN `cilindrata` `cilindrata` INT NOT NULL ;
```

Tabella officina:

```
-- -----  
-- Table `autotrasporti`.`officina`  
-- -----  
  
DROP TABLE IF EXISTS `autotrasporti`.`officina` ;  
  
CREATE TABLE IF NOT EXISTS `autotrasporti`.`officina` (  
  `ID_officina` INT NOT NULL COMMENT 'tabella officina',  
  `nome` VARCHAR(45) NOT NULL,  
  `data` DATE NOT NULL,  
  `indirizzo` VARCHAR(45) NOT NULL,  
  `numero_dipendenti` INT NOT NULL,  
  PRIMARY KEY (`ID_officina`))  
ENGINE = InnoDB;
```

Tabella comunicazione:

```
-- -----  
-- Table `autotrasporti`.`comunicazioni`  
-- -----  
DROP TABLE IF EXISTS `autotrasporti`.`comunicazioni` ;  
  
CREATE TABLE IF NOT EXISTS `autotrasporti`.`comunicazioni` (  
  `responsabile_logistica_ID_responsabile` INT NOT NULL,  
  `Ufficio_ID_ufficio` INT NOT NULL,  
  PRIMARY KEY (`responsabile_logistica_ID_responsabile`, `Ufficio_ID_ufficio`),  
  INDEX `fk_responsabile_logistica_has_Ufficio_Ufficio1_idx` (`Ufficio_ID_ufficio` ASC),  
  INDEX `fk_responsabile_logistica_has_Ufficio_responsabile_logistica_idx` (`responsabile_logistica_ID_responsabile` ASC),  
  CONSTRAINT `fk_responsabile_logistica_has_Ufficio_responsabile_logistica1`  
    FOREIGN KEY (`responsabile_logistica_ID_responsabile`)  
    REFERENCES `autotrasporti`.`responsabile_logistica` (`ID_responsabile`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_responsabile_logistica_has_Ufficio_Ufficio1`  
    FOREIGN KEY (`Ufficio_ID_ufficio`)  
    REFERENCES `autotrasporti`.`Ufficio` (`ID_ufficio`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

Tabella assegnazioni:

```
-- -----  
-- Table `autotrasporti`.`assegnazioni`  
-- -----  
DROP TABLE IF EXISTS `autotrasporti`.`assegnazioni` ;  
  
CREATE TABLE IF NOT EXISTS `autotrasporti`.`assegnazioni` (  
  `Autista_ID_autista` INT NOT NULL,  
  `Rimorchio_Targa_r` VARCHAR(255) NOT NULL,  
  PRIMARY KEY (`Autista_ID_autista`, `Rimorchio_Targa_r`),  
  INDEX `fk_Autista_has_Rimorchio_Rimorchio1_idx` (`Rimorchio_Targa_r` ASC),  
  INDEX `fk_Autista_has_Rimorchio_Autista1_idx` (`Autista_ID_autista` ASC),  
  CONSTRAINT `fk_Autista_has_Rimorchio_Autista1`  
    FOREIGN KEY (`Autista_ID_autista`)  
    REFERENCES `autotrasporti`.`Autista` (`ID_autista`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_Autista_has_Rimorchio_Rimorchio1`  
    FOREIGN KEY (`Rimorchio_Targa_r`)  
    REFERENCES `autotrasporti`.`Rimorchio` (`Targa_r`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

Tabella manutenzione:

```
-- -----  
-- Table `autotrasporti`.`Manutenzione`  
-- -----  
  
DROP TABLE IF EXISTS `autotrasporti`.`Manutenzione` ;  
  
CREATE TABLE IF NOT EXISTS `autotrasporti`.`Manutenzione` (  
  `officina_ID_officina` INT NOT NULL,  
  `Camion_targa_c` VARCHAR(255) NOT NULL,  
  PRIMARY KEY (`officina_ID_officina`, `Camion_targa_c`),  
  INDEX `fk_officina_has_Camion_Camion1_idx` (`Camion_targa_c` ASC),  
  INDEX `fk_officina_has_Camion_officina1_idx` (`officina_ID_officina` ASC),  
  CONSTRAINT `fk_officina_has_Camion_officina1`  
    FOREIGN KEY (`officina_ID_officina`)  
    REFERENCES `autotrasporti`.`officina` (`ID_officina`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_officina_has_Camion_Camion1`  
    FOREIGN KEY (`Camion_targa_c`)  
    REFERENCES `autotrasporti`.`Camion` (`targa_c`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

Implementazioni operazioni:

Op 1: Lista consegna merce

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `lista_consegna_merce`()  
BEGIN  
  select Cliente_ID_cliente, count(*) as numero_consegne  
  from merce  
  group by Cliente_ID_cliente;  
END
```

Op 2: registrazione nuovo cliente

```
INSERT INTO Cliente (ID_cliente, Nome, Cognome, nome_azienza, n_telefono, email, ID_ufficio)  
VALUES ('100', 'Simon', 'Riley', 'Ghost', '3270776175', 'simonghost@unict.it',  
      '132');
```

Op 3: Registrazione peso merce

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `registrazione_peso`()  
BEGIN  
  UPDATE Merce  
  SET peso = 10000  
  WHERE ID_merce = 0002;  
END
```


Op 4: registrazione manutenzione camion "per targa"

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `numero_manutenzioni`()
BEGIN
select targa_c, count(*) AS numero_manutenzioni
from camion LEFT join manutenzione on camion.targa_c=manutenzione.Camion_targa_c
group by targa_c;
END
```

Op 5: lista responsabili su "ID_responsabili" (Si assume che ogni responsabile abbia ricevuto almeno una comunicazione)

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `lista responsabili`()
BEGIN
select *
from comunicazioni;
END
```

Op 6: stipendio mensile per ogni autista "su stipendio"

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `Stipendio mensile di tutti gli autisti`()
BEGIN
select ID_autista,nome,cognome, (giorni_lavoro * importo_giornaliero) as stipendio_mensile
from autista NATURAL join stipendio ;
END
```

Op 7: controllo giorni di lavoro di ogni dipendente

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `giorni_lavoro`()
BEGIN
select ID_autista, giorni_lavoro
from autista;
END
```

Op 8: controllo ore guida rimanenti "tramite scheda tachigrafo"

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `controllo ore guida rimanenti`()
BEGIN
select Autista_ID_autista, ore_guida
from carta_tachigrafo
where ore_guida>50;
END
```

Op 9: lista dipendenti "anziani"

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `lista dipendenti anziani`(  
BEGIN  
select ID_autista, nome, età  
from autista  
where età>60 ;  
END
```

Op 10: registrazione nuovo camion tramite targa

```
INSERT TO camion(Targa_c, cilindrata, modello, assicurazione, anno, ID_autista)  
VALUES ('valore_Targa_c','cilindrata','modello','assicurazione','anno','ID_autista')
```

Op 11: lista clienti

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `lista clienti`()  
BEGIN  
select ID_cliente  
from cliente  
where cognome='Pulvirenti' AND nome like 'A%';  
END
```

Op 12: modifica dipendenti

```
INSERT INTO autista (ID_autista, nome, cognome, età, numero_t, giorni_lavoro, ID_stipendio)  
VALUES ('valore ID_autista','valore nome','valore cognome','valore età','valore numero_t','valore giorni lavoro'  
| | | | 'valore ID_stipendio')
```

Op 13: lista camion

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `lista camion`()  
BEGIN  
select*  
from camion  
where targa_c LIKE 'A%Z';  
END
```

Op 14: conteggio numero di consegne fatte da un autista

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `lista numero di consegne`()  
BEGIN  
select count(*) AS numero_rimorchi , ID_autista  
from assegnazioni_rimorchio  
group by ID_autista;  
END
```

TRIGGER

Per una migliore qualità del database saranno inseriti di seguito, i trigger che permettono un corretto funzionamento e la creazione di file di log e delle variazioni apportate al database a scopo di controllo.

Ne verrà assegnato uno per ogni tabella:

UFFICIO:

```
CREATE DEFINER='root'@'localhost' TRIGGER `ufficio_AFTER_INSERT` AFTER INSERT ON `ufficio` FOR EACH ROW BEGIN
IF (NEW.n_dipendenti <= 0 OR NEW.n_dipendenti >= 20) THEN
    -- messaggio di errore
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Il numero di dipendenti deve essere compreso tra 1 e 9.';
END IF;
END
```

CLIENTE:

```
CREATE DEFINER='root'@'localhost' TRIGGER `cliente_BEFORE_INSERT` BEFORE INSERT ON `cliente` FOR EACH ROW BEGIN
-- controllo del numero_t
IF (LENGTH(NEW.n_telefono) != 12) THEN
    -- mostrare un messaggio di errore
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Il numero_t deve essere uguale a 10 cifre numeriche.';
END IF;
END
```

MERCE:

```
CREATE DEFINER='root'@'localhost' TRIGGER `merce_BEFORE_INSERT` BEFORE INSERT ON `merce` FOR EACH ROW BEGIN
-- controllo della data di consegna
IF (NEW.data_consegna < CURRENT_DATE) THEN
    -- mostrare un messaggio di errore
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La data di consegna deve essere maggiore della data attuale.';
END IF;
END
```

RESPONSABILE_LOGISTICA

```
CREATE DEFINER='root'@'localhost' TRIGGER `responsabile_logistica_BEFORE_INSERT` BEFORE INSERT ON `responsabile_logistica` FOR EACH ROW BEGIN
-- controllo del numero_t
IF (length(NEW.n_telefono) != 12) THEN
    -- mostrare un messaggio di errore
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Il numero_t deve essere uguale a 10 cifre numeriche.';
END IF;
END
```

RIMORCHIO:

```
CREATE DEFINER='root'@'localhost' TRIGGER `rimorchio_BEFORE_INSERT` BEFORE INSERT ON `rimorchio` FOR EACH ROW BEGIN
IF (LENGTH(NEW.Targa_r) < 7 OR NOT NEW.Targa_r REGEXP '^XA.*$') THEN
    -- mostrare un messaggio di errore
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La targa del rimorchio deve iniziare con XA.';
END IF;

-- controllo dell'anno
IF (NEW.anno < '2000-01-01') THEN
    -- mostrare un messaggio di errore
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'bisogna segnalare ad un superiore questo rimorchio perchè troppo vecchio.';
END IF;

-- controllo della capacità di carico per il modello autoarticolato
IF (NEW.capacità_carico > 40000) THEN
    -- mostrare un messaggio di errore
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La capacità di carico non può superare 40000 per il modello autoarticolato.';
END IF;
END
```

AUTISTA:

```
CREATE DEFINER='root'@'localhost' TRIGGER `autista_BEFORE_INSERT` BEFORE INSERT ON `autista` FOR EACH ROW BEGIN
IF (LENGTH(NEW.n_telefono) <> 12) THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Il numero di telefono deve avere 10 cifre';
END IF;

IF (NEW.giorni_lavoro > 26) THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Il numero massimo di giorni di lavoro è 25';
END IF;

IF (NEW.età <= 21) THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'L'età deve essere maggiore di 21';
END IF;
END
```

CARTA_TACHIGRAFO;

```
CREATE DEFINER='root'@'localhost' TRIGGER `carta_tachigrafo_BEFORE_INSERT` BEFORE INSERT ON `carta_tachigrafo`  
FOR EACH ROW BEGIN  
    -- controllo della data di scadenza  
    IF (NEW.data_scadenza < CURRENT_DATE) THEN  
        -- mostrare un messaggio di errore  
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La data di scadenza deve essere maggiore della data attuale.';  
    END IF;  
END
```

STIPENDIO:

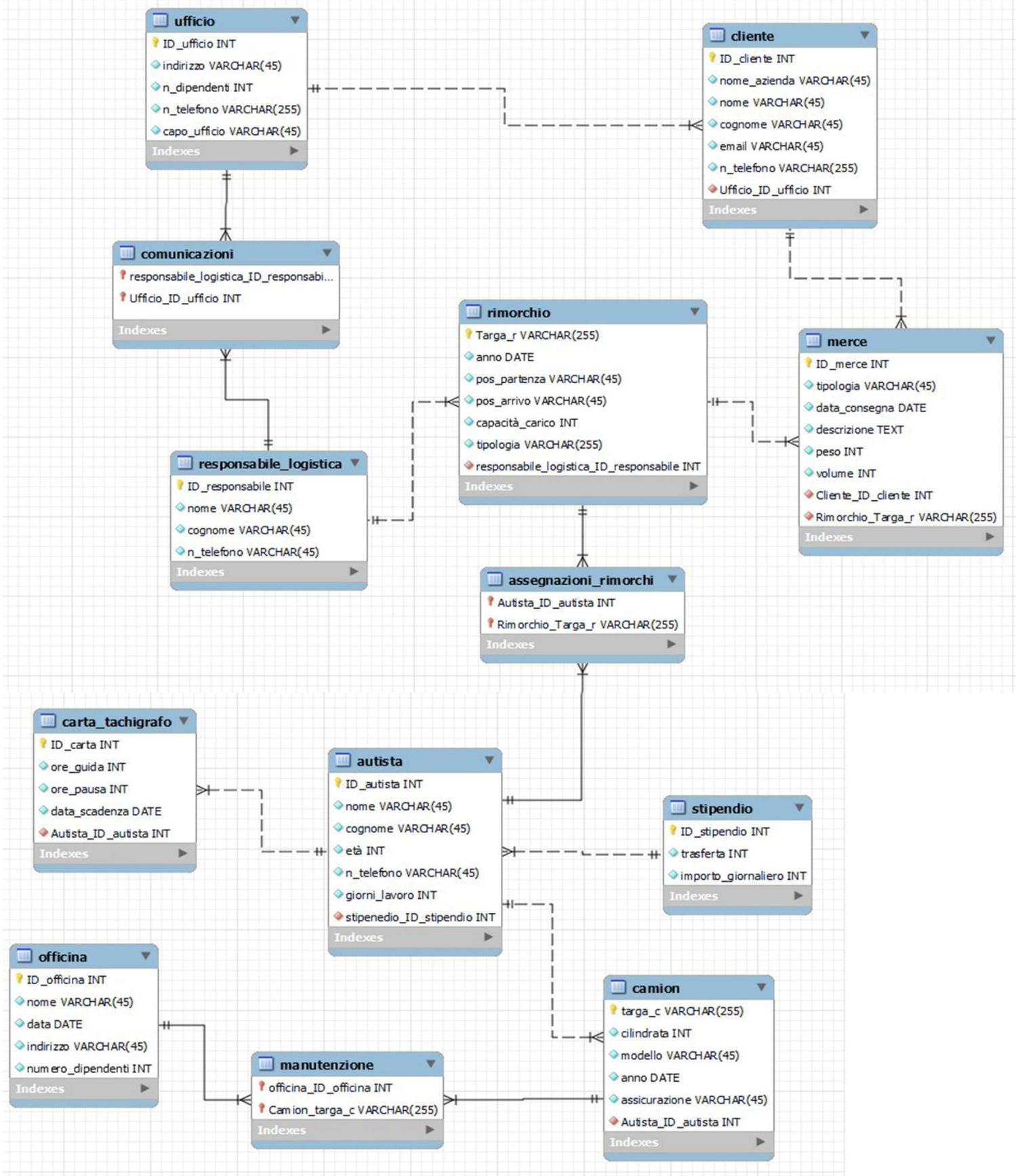
```
CREATE DEFINER='root'@'localhost' TRIGGER `stipendio_AFTER_INSERT` AFTER INSERT ON `stipendio` FOR EACH ROW BEGIN  
IF (NEW.importo_giornaliero > 80) THEN  
    -- mostrare un messaggio di errore  
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'L'importo giornaliero non può superare 80€.';  
END IF;  
END
```

CAMION:

```
CREATE DEFINER='root'@'localhost' TRIGGER `camion_BEFORE_INSERT` BEFORE INSERT ON `camion` FOR EACH ROW BEGIN  
IF (LENGTH(NEW.Targa_c) < 7 OR NOT NEW.Targa_c REGEXP '^BF.*$') THEN  
    -- mostrare un messaggio di errore  
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La targa del trattore deve essere formata da 4 lettere e 3 numeri.';  
END IF;  
-- controllo della compagnia di assicurazione  
IF (NEW.assicurazione != 'Allianz') THEN  
    -- mostrare un messaggio di errore  
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La compagnia di assicurazione deve essere Allianz.';  
END IF;  
  
-- controllo della cilindrata  
IF (NEW.cilindrata < 7000 OR NEW.cilindrata > 10000) THEN  
    -- mostrare un messaggio di errore  
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La cilindrata deve essere compresa tra 7000 e 10000.';  
END IF;  
END
```

Su officina non verrà effettuato nessun controllo.

GRAFICO UML:



Come Provare il database su mysql workbench:

1. aprire il file zip contenuto nella cartella: "Bertolami_carmelo_1000030407"
2. estrarre il dump e aprirlo con mysql workbench
3. sulla barra dei menù cliccare: database -> connect to database -> ok lasciando le impostazioni di default
4. incollare il codice del dump in una nuova tab di mysql e cliccare l'icona a forma di fulmine per eseguirlo.
5. Nel quadrato a sinistra "SCHEMAS" cliccare le frecce per il refresh dei database salvati.
6. Usare il comando **use "nomedatabase"**

Utilizzate il database a vostro piacimento.

Per assistenza tecnica contattare:

Email: carmelo.bertolami29@gmail.com

Fax: /