

Elaborato per il corso di basi di dati

A.A. 2023/2024

Progettazione di una base di dati per il supporto di un'azienda di noleggio auto

Nicolò Morini

0001071241

nicolo.morini2@studio.unibo.it

Contents

1	Analisi dei requisiti	3
1.1	Intervista	3
1.2	Estrazione dei concetti principali	4
1.3	Elenco delle principali azioni richieste	4
2	Progettazione concettuale	5
2.1	Schema Scheletro	5
2.2	Schema Concettuale Finale	8
3	Progettazione Logica	10
3.1	Stima del volume dei dati	10
3.2	Descrizione delle operazioni principali e stima della loro frequenza	11
3.3	Schemi di navigazione e tabelle degli accessi	11
4	Raffinamento dello schema	15
4.1	Eliminazione delle gerarchie	15
4.2	Eliminazione degli attributi composti	15
4.3	Scelta delle chiavi primarie	15
4.4	Eliminazione degli identificatori esterni	15
4.5	Schema Raffinato	16
4.6	Analisi delle ridondanze	18
5	Traduzione di entità e associazioni in relazioni	20
5.1	Schema Relazionale Finale	21
6	Traduzione delle operazioni in query SQL	23
7	Progettazione dell'applicazione	28
7.1	Descrizione dell'architettura	28

1 Analisi dei requisiti

Si vuole realizzare un database di supporto per un'agenzia di noleggio automobilistico. La base di dati dovrà permettere quindi agli utilizzatori del sistema di gestire i dettagli di ogni veicolo noleggiabile, la registrazione di ogni cliente con annessa prenotazione e mantenerne uno storico dettagliato. Inoltre sarà possibile gestire la fatturazione dell'azienda, ossia inserire e visualizzare le fatture relative ad ogni noleggio, oltre a mantenerne uno storico a fini statistici.

1.1 Intervista

Si vuole registrare ogni veicolo con i relativi dettagli, memorizzandone modello, casa produttrice, anno di produzione, targa, chilometraggio, stato di manutenzione e disponibilità. I veicoli offerti dall'azienda di noleggio possono essere di diversi tipi (utilitarie, berline, luxury cars ecc.). L'utilizzatore del sistema deve poter cercare un determinato veicolo e conoscerne i soprastanti dettagli (durante la fase di prenotazione del veicolo se ne deve verificare la disponibilità). Riguardo ai clienti, si vogliono memorizzare i relativi dati anagrafici, ossia nome, cognome, codice fiscale, indirizzo di residenza, numero di telefono, eventuale e-mail, numero della patente di guida ed eventuale indirizzo di fatturazione diverso da quello di residenza. Dopo la registrazione del cliente, il sistema deve permettere la visualizzazione dello storico dei noleggi di quest'ultimo, con relativi dettagli come la data, il veicolo noleggiato e il costo. Inoltre, deve essere integrata una funzionalità che permette la visualizzazione dei clienti con il maggior numero di noleggi (ad esempio i primi 10 clienti, ordinati per numero di noleggi). Il sistema deve poi permettere la registrazione delle prenotazioni effettuate dai clienti, con annessi data di prenotazione, data di inizio e fine noleggio e identificativo del veicolo noleggiato, oltre che alle modifiche o cancellazioni di queste ultime (ad esempio, una prenotazione può essere modificata o cancellata entro 48 ore dall'inizio del noleggio). Una prenotazione di un noleggio per una determinata data può comprendere fino a 2 veicoli noleggiabili contemporaneamente (un cliente può noleggiare massimo 2 veicoli alla volta nella stessa prenotazione, quindi nello stesso periodo di tempo). A livello statistico deve essere possibile visualizzare il tasso di utilizzo di ogni veicolo e quelli più noleggiati nel complesso. Infine, per quanto riguarda la fatturazione dell'azienda e la gestione dei pagamenti, il sistema deve essere in grado di generare le fatture per ogni noleggio, con tutte le informazioni necessarie inclusi i dettagli sulle tariffe giornaliere ed eventuali sconti o costi aggiuntivi. Sarà necessario anche generare un eventuale sanzione nel caso ci siano ritardi di consegna del veicolo o danni, e quest'ultima sarà allegata e inclusa nella fattura. Ad ogni pagamento deve essere associato il metodo utilizzato (ad esempio contanti, carta di credito, bonifico bancario). L'utilizzatore del sistema deve poter inoltre visualizzare il fatturato totale di un particolare mese.

1.2 Estrazione dei concetti principali

TERMINE	BREVE DESCRIZIONE	EVENTUALI SINONIMI
Veicolo	Oggetto che viene noleggiato dall'azienda ad un determinato cliente	Mezzo
Cliente	Colui che effettua il noleggio di un determinato veicolo	
Noleggio	Servizio offerto dall'azienda con il quale affida un veicolo ad un determinato cliente per un periodo limitato di tempo	
Prenotazione	Azione con la quale un cliente richiede fino a 2 noleggi di veicoli in un determinato intervallo di tempo	
Fattura	Documento fiscale contenente i dettagli sul pagamento del noleggio di un veicolo	Fatturazione
Sanzione	Eventuale costo aggiuntivo dato da un ritardo di consegna del veicolo o da eventuali danni ad esso	

A seguito della comprensione dei requisiti del sistema, si redige un testo che ne contenga in modo sintetico gli aspetti principali, andando a rimuovere eventuali ambiguità semantiche:

Per ogni **Veicolo** vengono memorizzati modello, marca, anno, targa, chilometraggio, stato di manutenzione e disponibilità. L'azienda **Noleggia** i veicoli in seguito ad una **Prenotazione**, della quale si mantengono dettagli come la data di prenotazione, quella di inizio e fine noleggio e il veicolo interessato. Le prenotazioni possono essere modificate. Quest'ultime vengono effettuate dai **Clienti**, ossia coloro che vogliono noleggiare un veicolo, dei quali si mantengono nome, cognome, codice fiscale, indirizzo, numero di telefono, eventuale e-mail, numero della patente di guida e indirizzo di fatturazione. Ogni volta che viene completato un noleggio, l'azienda rilascia una **Fattura** che contiene il tipo di pagamento effettuato dal cliente e l'importo totale dell'acquisto.

1.3 Elenco delle principali azioni richieste

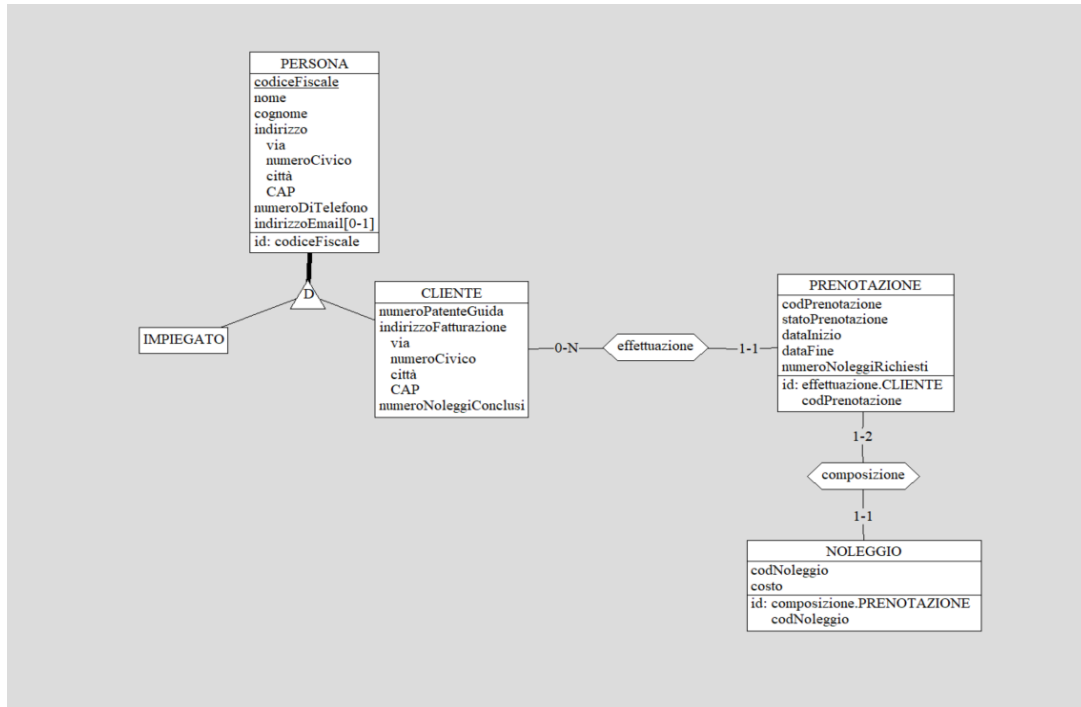
1. Registrare un nuovo veicolo
2. Visualizzare la disponibilità di un veicolo
3. Visualizzare la scheda tecnica di un veicolo
4. Registrare un nuovo cliente
5. Visualizzare lo storico dei noleggi di un cliente
6. Visualizzare i 10 clienti con più noleggi
7. Registrare una nuova prenotazione
8. Modificare una prenotazione
9. Cancellare una prenotazione
10. Avviare un noleggio

11. Terminare un noleggio
12. Visualizzare il tasso di utilizzo di un veicolo
13. Visualizzare i veicoli più noleggiati
14. Emettere una nuova fattura per una prenotazione
15. Generare una sanzione per una prenotazione
16. Visualizzare il fatturato mensile

2 Progettazione concettuale

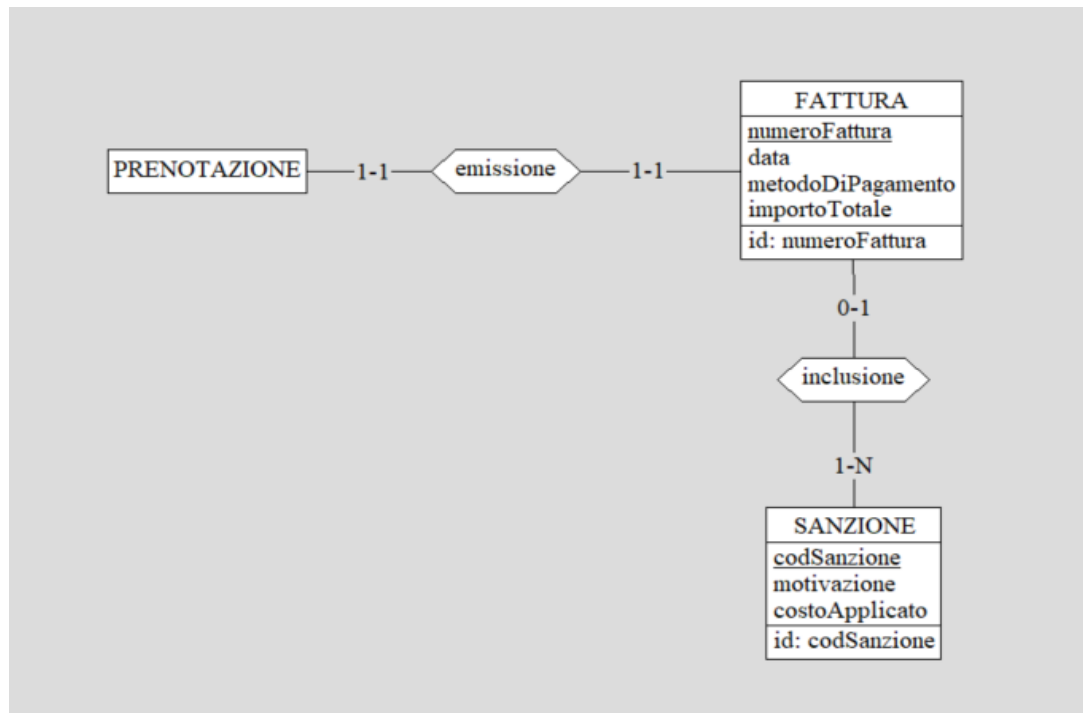
2.1 Schema Scheletro

All'intero dello schema concettuale ho introdotto l'entità Persona che funge da superclasse per le sotto-entità Impiegato e Cliente. All'interno di questo progetto verranno trattate solo casistiche riguardanti l'entità Cliente, ma viene posta come base anche l'entità Impiegato nel caso di volesse scalare l'applicazione con operazioni riguardanti quest'ultimo. La gerarchia è di tipo totale ed esclusiva, in quanto una Persona deve essere registrata obbligatoriamente o come Impiegato o come Cliente, e non esiste un'istanza in cui siano unite entrambe le tipologie. Un Cliente può effettuare una o più prenotazioni nell'arco del tempo, ognuna delle quali è rispettiva ad un solo Cliente. Una Prenotazione può riferirsi fino a 2 noleggi contemporanei, con il vincolo che debbano essere inclusi nello stesso intervallo di date. Un Noleggio si riferisce ad una e una sola prenotazione.

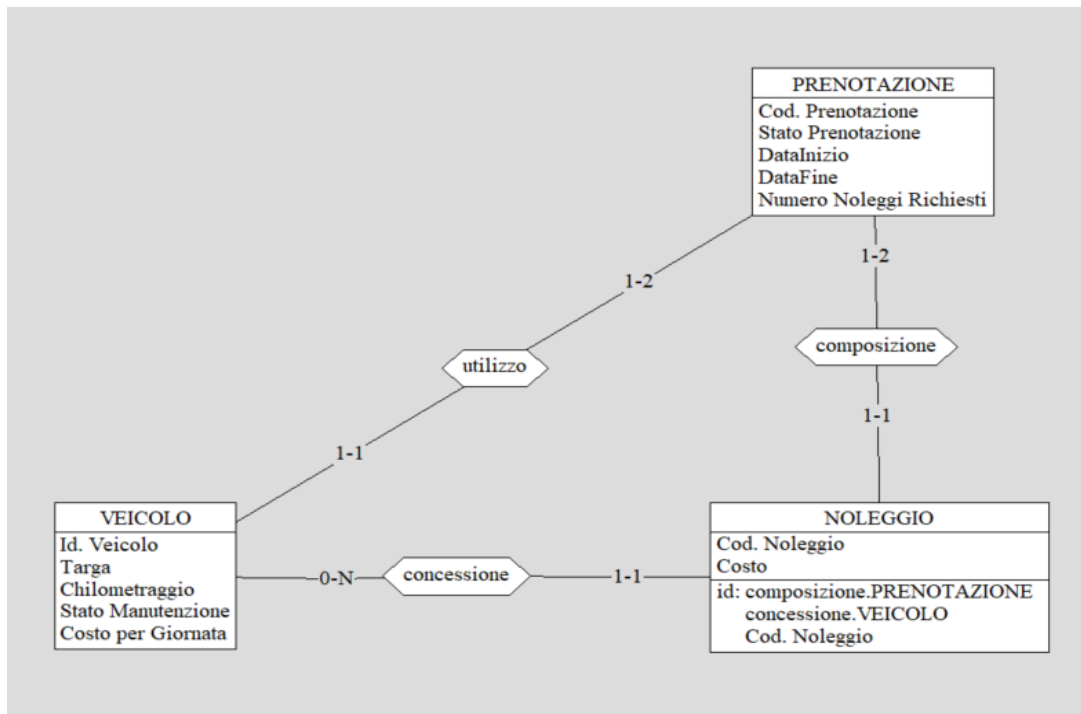


Quando una Prenotazione è conclusa, viene emessa una Fattura relativa ad essa contenente i dettagli del pagamento. Una Fattura si riferisce ad una e una sola Prenotazione ed è identificata da un `numeroFattura`. Ad ogni Fattura può poi essere inclusa un'eventuale Sanzione riguardante la Prenotazione, che comporta un costo aggiuntivo in Fattura. Una

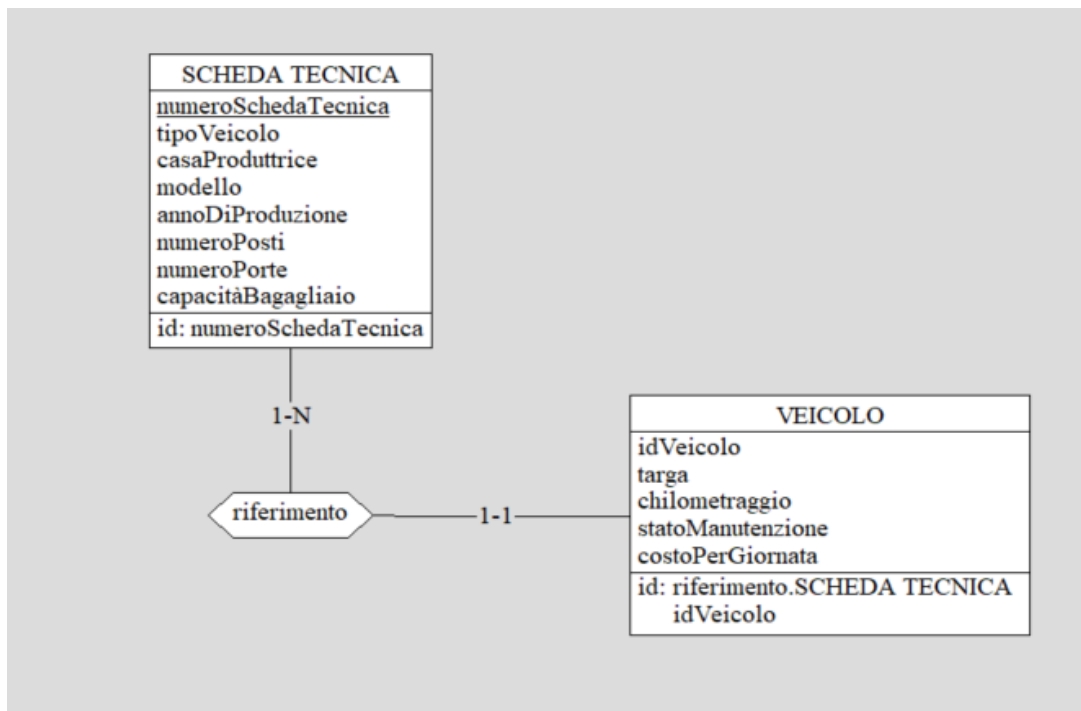
Sanzione può riferirsi anche a più fattura (ad esempio se scelgo di avere delle sanzioni predefinite con costi predefiniti es. "Ritardo nella consegna").



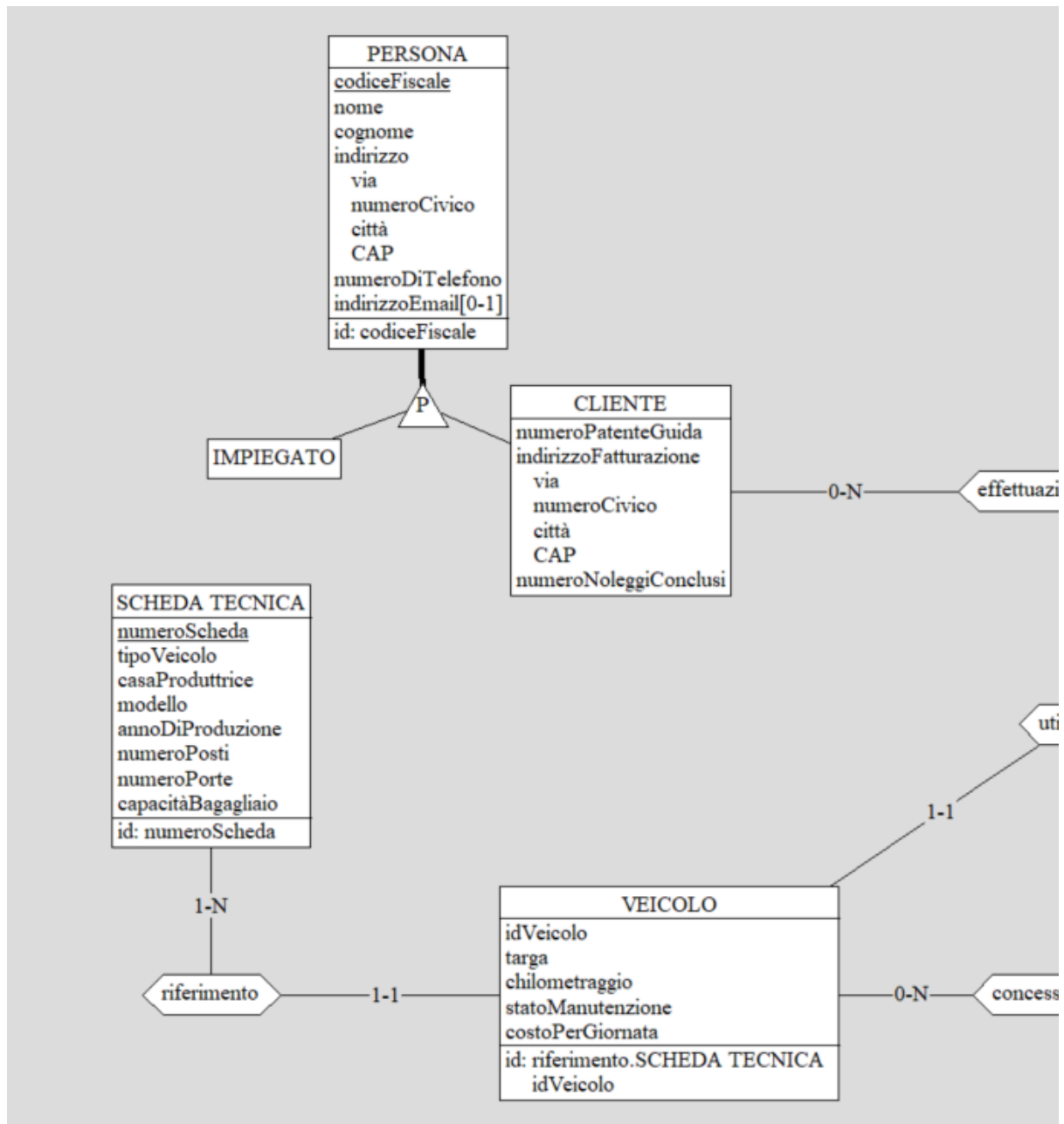
Ad ogni Prenotazione deve essere associato l'utilizzo di massimo 2 Veicoli contemporaneamente, come espresso dal vincolo (in modo che i veicoli in questione vengano prenotati e non si possa utilizzarli per altre prenotazioni). Un Noleggio si riferisce ad uno e un solo veicolo, mentre un veicolo può essere utilizzato per diversi noleggi. Nello schema concettuale un Noleggio viene identificato dal codice della prenotazione che compone, dall'id del veicolo che viene concesso per il noleggio e da un codice noleggio interno. Non essendo concesso il noleggio di uno stesso veicolo più volte per una stessa prenotazione, posso utilizzare come chiave primaria l'identificatore misto composto dai 3 attributi senza incorrere in errori.

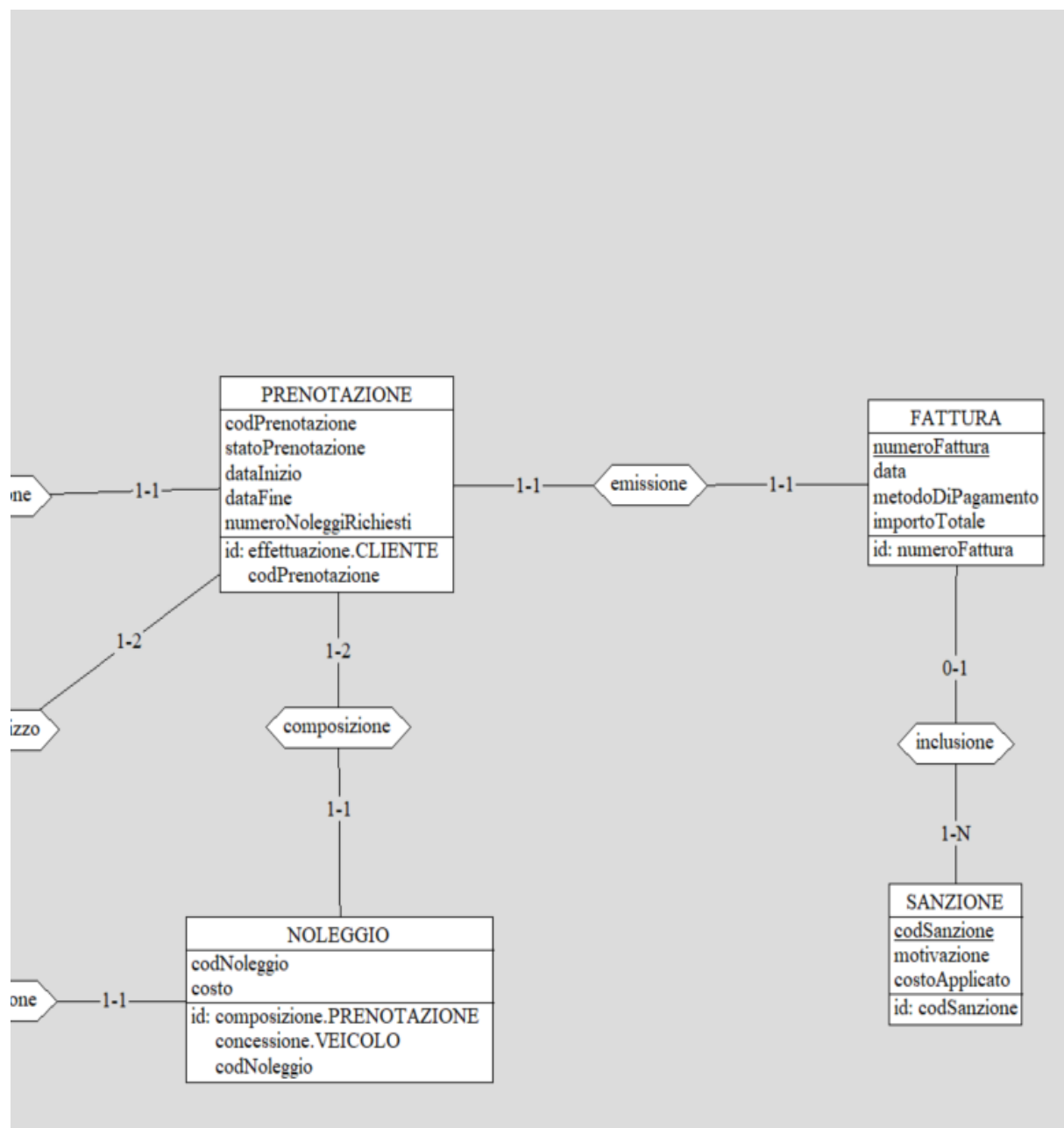


Infine, ogni Veicolo possiede una ed una sola Scheda Tecnica, quest'ultima però può essere associata a più veicoli (ad esempio se ho più auto dello stesso modello a disposizione). L'entità Veicolo è identificata in modo misto dal numero della Scheda Tecnica (esterno) e dall'id del veicolo (interno).



2.2 Schema Concettuale Finale





3 Progettazione Logica

3.1 Stima del volume dei dati

CONCETTO	COSTRUTTO	VOLUME
Persona	E	5.020
Impiegato	E	20
Cliente	E	5.000
Effettuazione	R	15.000
Prenotazione	E	15.000
Utilizzo	R	15.000
Composizione	R	18.000
Noleggio	E	18.000
Concessione	R	18.000
Veicolo	E	200
Riferimento	R	200
Scheda Tecnica	E	150
Emissione	R	15.000
Fattura	E	15.000
Inclusione	R	750
Sanzione	E	750

3.2 Descrizione delle operazioni principali e stima della loro frequenza

CODICE	OPERAZIONE	FREQUENZA
1	Registrare un nuovo veicolo	2 al mese
2	Visualizzare la disponibilità di un veicolo	10 al giorno
3	Visualizzare la scheda tecnica di un veicolo	5 al giorno
4	Registrare un nuovo cliente	3 alla settimana
5	Visualizzare lo storico dei noleggi di un cliente	2 al giorno
6	Visualizzare i 10 clienti con più noleggi	1 al mese
7	Registrare una nuova prenotazione	5 al giorno
8	Modificare una prenotazione	1 al giorno
9	Cancellare una prenotazione	1 al giorno
10	Attivare un noleggio	5 al giorno
11	Terminare un noleggio	5 al giorno
12	Visualizzare il tasso di utilizzo di un veicolo	1 al mese
13	Visualizzare i veicoli più noleggiati	1 al mese
14	Emettere una nuova fattura per una prenotazione	5 al giorno
15	Generare una sanzione per una prenotazione	1 alla settimana
16	Visualizzare il fatturato mensile	1 al mese

3.3 Schemi di navigazione e tabelle degli accessi

Sono riportati solo gli schemi di navigazione dei casi più complessi.

OP 1 - Registrare un nuovo veicolo

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Veicolo	E	1	S
Scheda Tecnica	E	1	L
		Totale: 1S + 1L = 6 al mese	

OP 2 - Visualizzare la disponibilità di un veicolo

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Veicolo	E	1	L
Utilizzo	R	1	L
Prenotazione	E	1	L
		Totale: 3L = 10 al giorno	

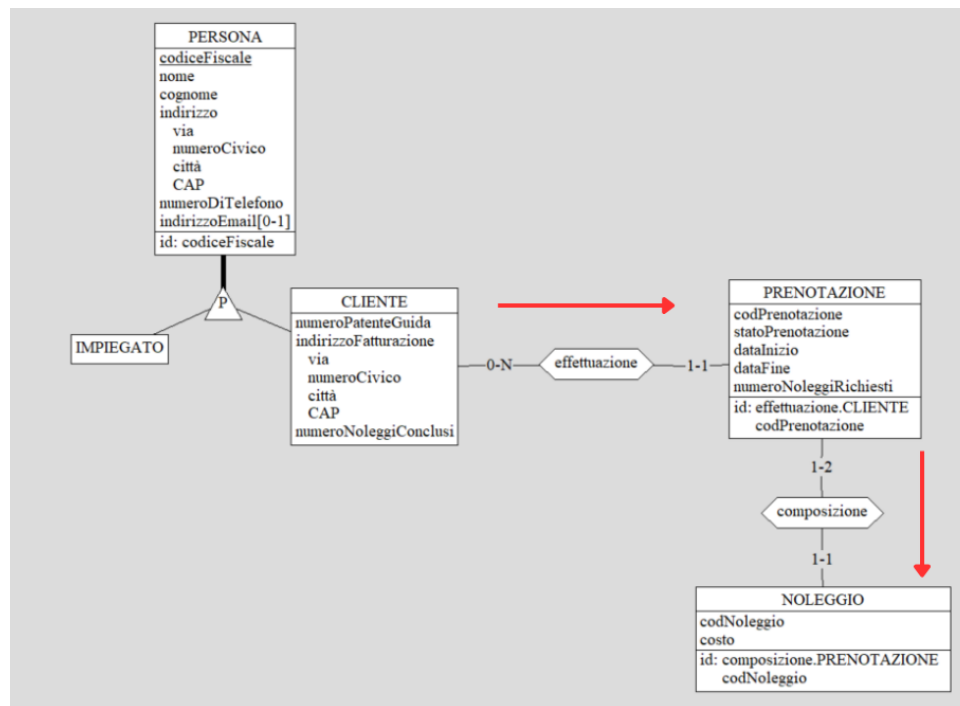
OP 3 - Visualizzare la scheda tecnica di un veicolo

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Veicolo	E	1	L
Scheda Tecnica	E	1	L
		Totale: 2L = 10 al giorno	

OP 4 - Registrare un nuovo cliente

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Persona	E	1	S
Cliente	E	1	S
		Totale: 2S = 12 alla settimana	

OP 5 - Visualizzare lo storico noleggi di un cliente



CONCETTO	COSTRUTTO	ACCESSI	TIPO
Cliente	E	1	L
Effettuazione	R	3	L
Prenotazione	E	3	L
Composizione	R	3,6	L
Noleggio	E	3,6	L
		Totale: 14,2L = 28,4 al giorno	

OP 6 - Visualizzare i 10 clienti con più noleggi

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Cliente	E	5.000	L
		Totale: 5.000L = 5.000 al mese	

OP 7 - Registrare una nuova prenotazione

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Cliente	E	1	L
Prenotazione	E	1	S
Utilizzo	R	1	S
		Totale: $2S + 1L = 25$ al giorno	

OP 8 - Modificare una prenotazione

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Prenotazione	E	1	L
Prenotazione	E	1	S
Cliente	E	1	L
Utilizzo	R	1	L
Utilizzo	R	1	S
		Totale: $2S + 3L = 7$ al giorno	

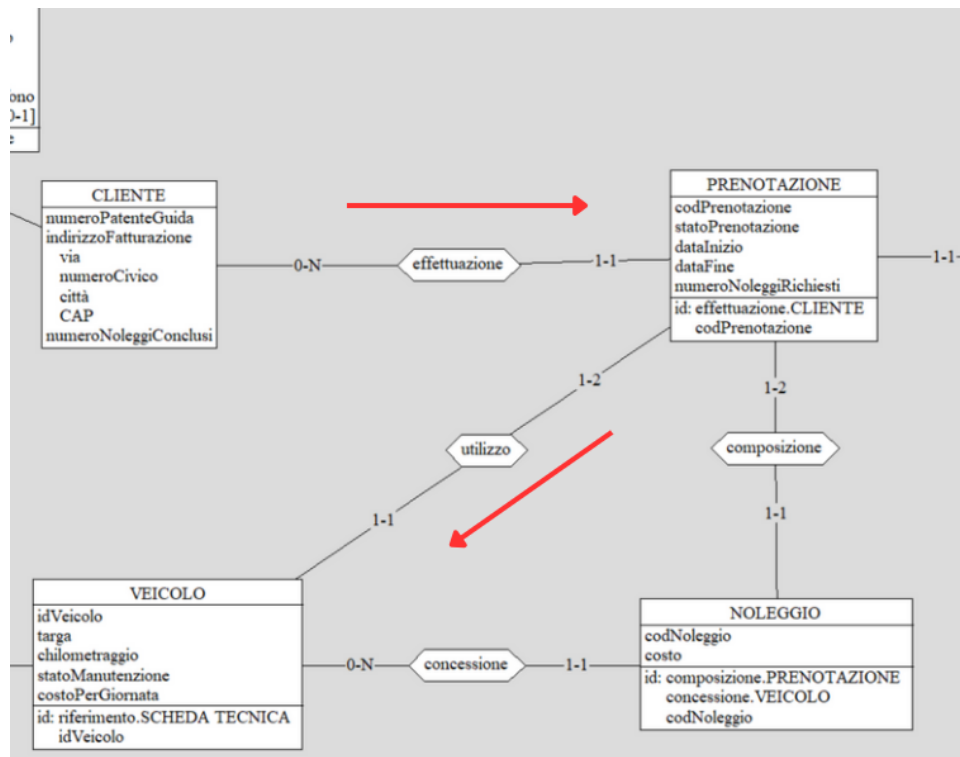
OP 9 - Cancellare una prenotazione

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Prenotazione	E	1	L
Prenotazione	E	1	S
Cliente	E	1	L
Utilizzo	R	1	L
Utilizzo	R	1	S
		Totale: $2S + 3L = 7$ al giorno	

OP 10 - Attivare un noleggio

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Noleggio	E	1	S
Prenotazione	E	1	L
Veicolo	E	1	L
		Totale: $1S + 2L = 20$ al giorno	

OP 11 - Terminare un noleggio



CONCETTO	COSTRUTTO	ACCESSI	TIPO
Cliente	E	1	L
Cliente	E	1	S
Effettuazione	R	3	L
Prenotazione	E	3	L
Prenotazione	E	3	S
Utilizzo	R	3	L
Utilizzo	R	3	S
Totale: 7S + 10L = 120 al giorno			

OP 12 - Visualizzare il tasso di utilizzo di un veicolo

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Veicolo	E	1	L
Utilizzo	R	75	L
Prenotazione	E	75	L
Totale: 151L = 302 al mese			

4 Raffinamento dello schema

4.1 Eliminazione delle gerarchie

Riguardo alla gerarchia presente in Persona, ho deciso di applicare una trasformazione delle sotto-entità in associazioni, ognuna delle quali sarà identificata esternamente dall'entità padre. Questa scelta è dovuta al fatto che la maggior parte degli accessi interessano solo l'entità "Cliente", perciò scegliendo un collasso verso l'alto avrei dovuto accedere ogni volta ad un'unica entità "Persona" contenente un grande numero di attributi, il che sarebbe risultato poco efficiente. In questo modo tengo quindi separati i due concetti potendo accedere solo alla cerchia ristretta di attributi che interessano la maggior parte degli accessi.

4.2 Eliminazione degli attributi composti

I due attributi composti presenti, ossia "indirizzo" in Persona e "indirizzoFatturazione" nell'entità Cliente sono stati divisi nelle rispettive componenti.

4.3 Scelta delle chiavi primarie

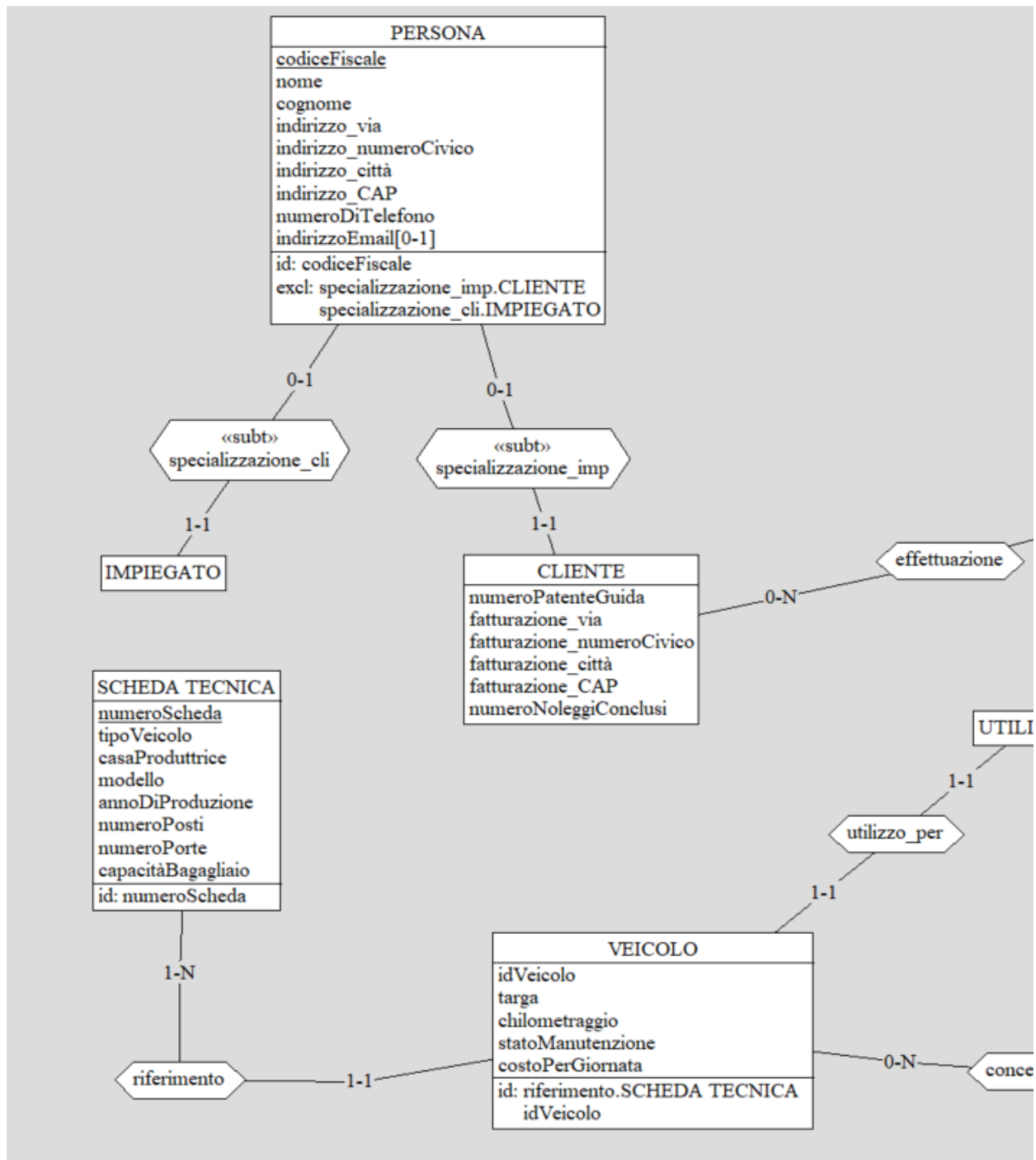
Entità	Chiave Primaria
PERSONA	codiceFiscale
IMPIEGATO	codiceFiscale:Persona
CLIENTE	codiceFiscale:Persona
PRENOTAZIONE	codPrenotazione, codiceFiscale
FATTURA	numeroFattura
SANZIONE	codSanzione
NOLEGGIO	codNoleggio, codPrenotazione, idVeicolo
VEICOLO	idVeicolo, numeroSchedaTecnica
SCHEDA TECNICA	numeroScheda

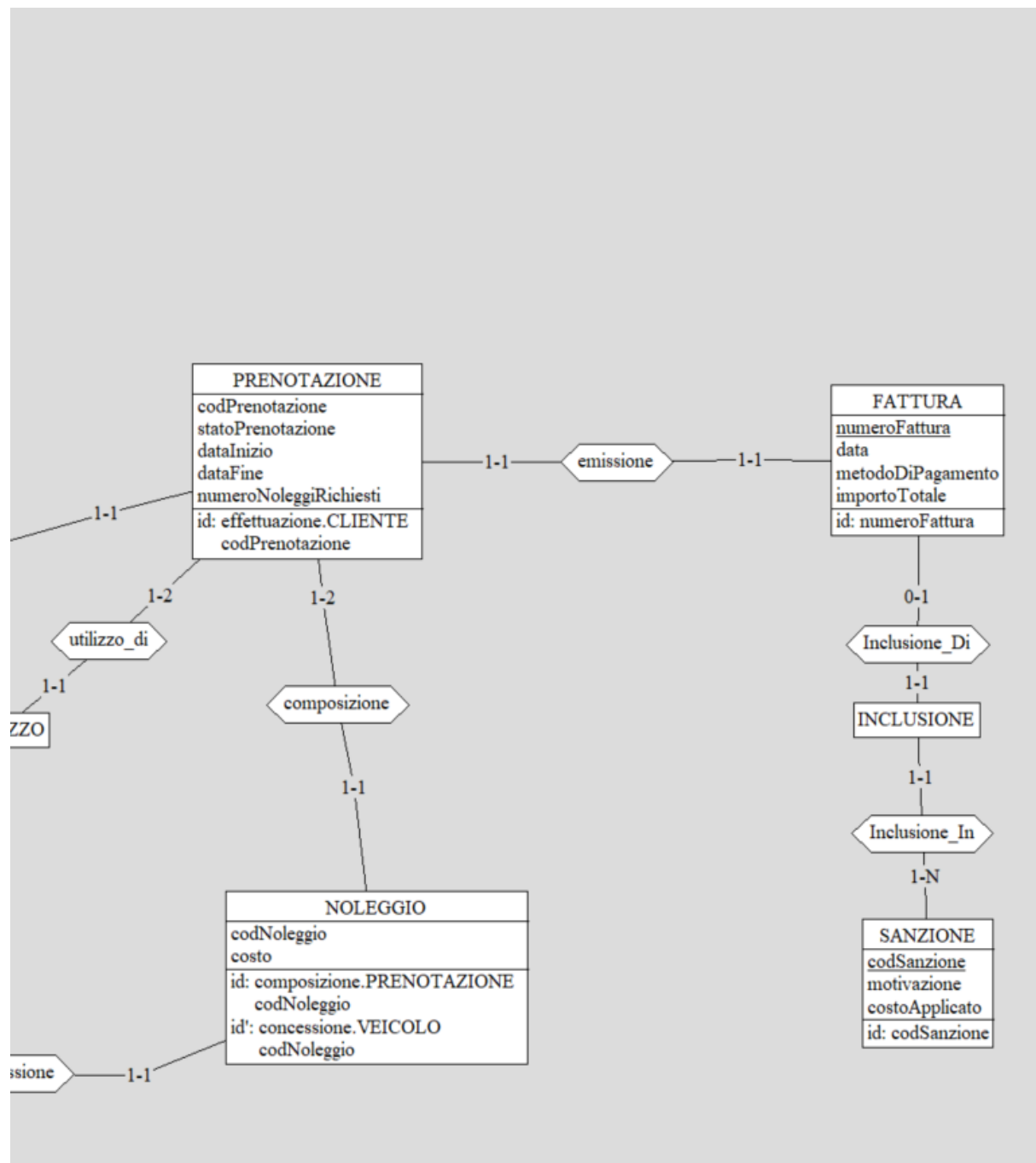
4.4 Eliminazione degli identificatori esterni

Nello schema E/R sono state eliminate le seguenti relazioni:

- SpecializzazioneImp, importando CodiceFiscale da Persona
- SpecializzazioneCli, importando CodiceFiscale da Persona
- Emissione, importando codPrenotazione in Fattura
- Effettuazione, importando CodiceFiscale in Prenotazione
- Utilizzo, reificata importando codPrenotazione da Prenotazione e idVeicolo da Veicolo
- Composizione, importando CodicePrenotazione in Noleggio
- Concessione, importando IdVeicolo in Noleggio
- Riferimento, importando NumeroScheda in Veicolo
- Inclusione, reificata importando NumeroFattura da Fattura e CodiceSanzione da Sanzione

4.5 Schema Raffinato





4.6 Analisi delle ridondanze

OP 6 - Visualizzare i 10 clienti con più noleggi

E' stata inserita una ridondanza per quanto riguarda il numero di noleggi effettuati da un cliente. Infatti si potrebbe risalire a questo dato ripercorrendo le prenotazioni da lui effettuate e a sua volta contare il numero di noleggi, per poi stilare una "classifica" utile al fine dell'operazione di visualizzare i clienti con più noleggi. Questo però sarebbe estremamente più dispendioso in termini di accessi, perché significherebbe ripercorrere per ogni cliente le sue prenotazioni e i corrispondenti noleggi:

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Cliente	E	5.000	L
Effettuazione	R	15.000	L
Prenotazione	E	15.000	L
Composizione	R	18.000	L
Noleggio	E	18.000	L
		Totale: 71.000L = 71.000 al mese	

Andando ad inserire la ridondanza, aumento gli accessi necessari ogni volta che eseguo l'operazione OP 11 - TERMINARE UN NOLEGGIO , in quanto non mi basta accedere alle entità noleggio e prenotazione, ma devo anche aggiornare il valore contenuto nell'entità cliente. Comparando però il numero di accessi che "guadagno" rispetto a quelli che "perdo", è conveniente mantenere questa ridondanza, in quanto ora l'operazione 6 avrà questo numero di accessi:

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Cliente	E	5.000	L
		Totale: 5.000L = 5000 al mese	

OP 13 - Emettere una nuova fattura

Una seconda ridondanza individuata in fase di progettazione consiste nell'attributo "Costo" all'interno dell'entità Noleggio. Quest'ultimo infatti potrebbe essere calcolato accedendo all'entità Veicolo e moltiplicando il "Costo per giornata". Questa scelta però comprometterebbe il numero di accessi da effettuare quando voglio emettere una nuova fattura. Infatti, avendo bisogno di conoscere il costo del singolo (o dei due) noleggi, dovrei arrivare ad accedere all'entità veicolo per fare il calcolo. Questo comporterebbe i seguenti accessi: SENZA RIDONDANZA ("Costo" in "Noleggio"):

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Fattura	E	1	S
Emissione	R	1	L
Prenotazione	E	1	L
Composizione	R	1	L
Noleggio	E	1	L
Veicolo	E	1	L
		Totale: $1S + 5L = 35$ al giorno	

CON RIDONDANZA:

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Fattura	E	1	S
Emissione	R	1	L
Prenotazione	E	1	L
Composizione	R	1	L
Noleggio	E	1	L
		Totale: $1S + 4L = 30$ al giorno	

Gli accessi quindi aumentano rimuovendo la ridondanza (seppur non in modo eccessivo). La scelta di mantenerla è anche data dal fatto che non va a compromettere il numero di accessi di nessun'altra operazione riguardante le entità Noleggio e Veicolo.

5 Traduzione di entità e associazioni in relazioni

PERSONE (codiceFiscale, ruolo, nome, cognome, indirizzo_via,
indirizzo_numeroCivico, indirizzo_città, indirizzo_CAP, numeroDiTelefono,
indirizzoEmail*)

IMPIEGATI (CFImpiegato:Persone)

CLIENTI (CFCliente:Persone, numeroPatenteGuida,
indirizzoFatturazione_via, indirizzoFatturazione_numeroCivico,
indirizzoFatturazione_città, indirizzoFatturazione_CAP,
numeroNoleggiConclusi)

PRENOTAZIONI (codPrenotazione, cliente:Clienti, statoPrenotazione,
dataInizio, dataFine, numeroNoleggiRichiesti)

FATTURE (numeroFattura, codPrenotazione:Prenotazioni,
data, metodoDiPagamento, importoTotale)

INCLUSIONI (fattura:Fatture, codSanzione:Sanzioni)

SANZIONI (codSanzioni, motivazione, costoApplicato)

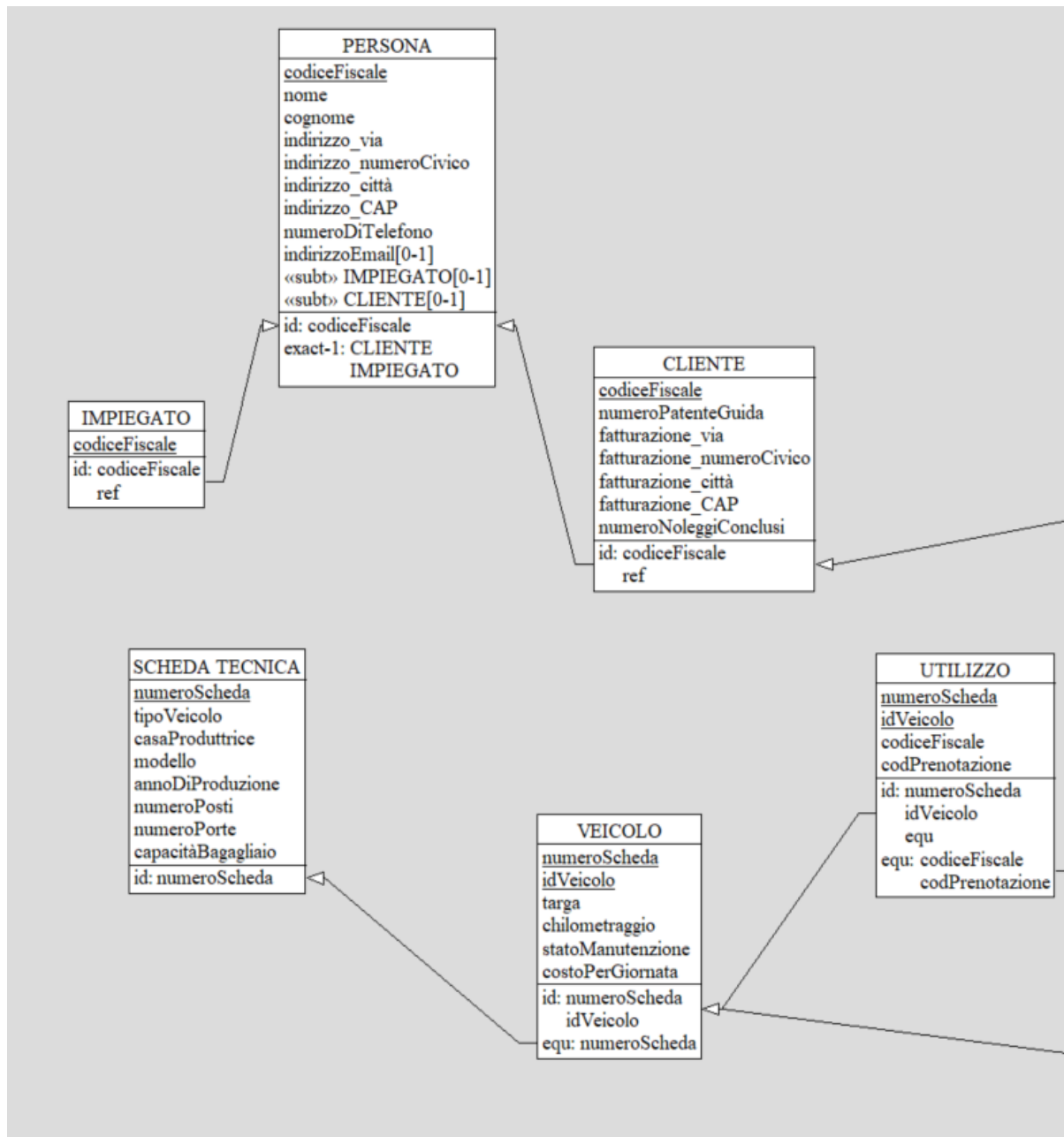
NOLEGGI (codNoleggio, codPrenotazione:Prenotazioni,
veicolo:Veicoli, costo)

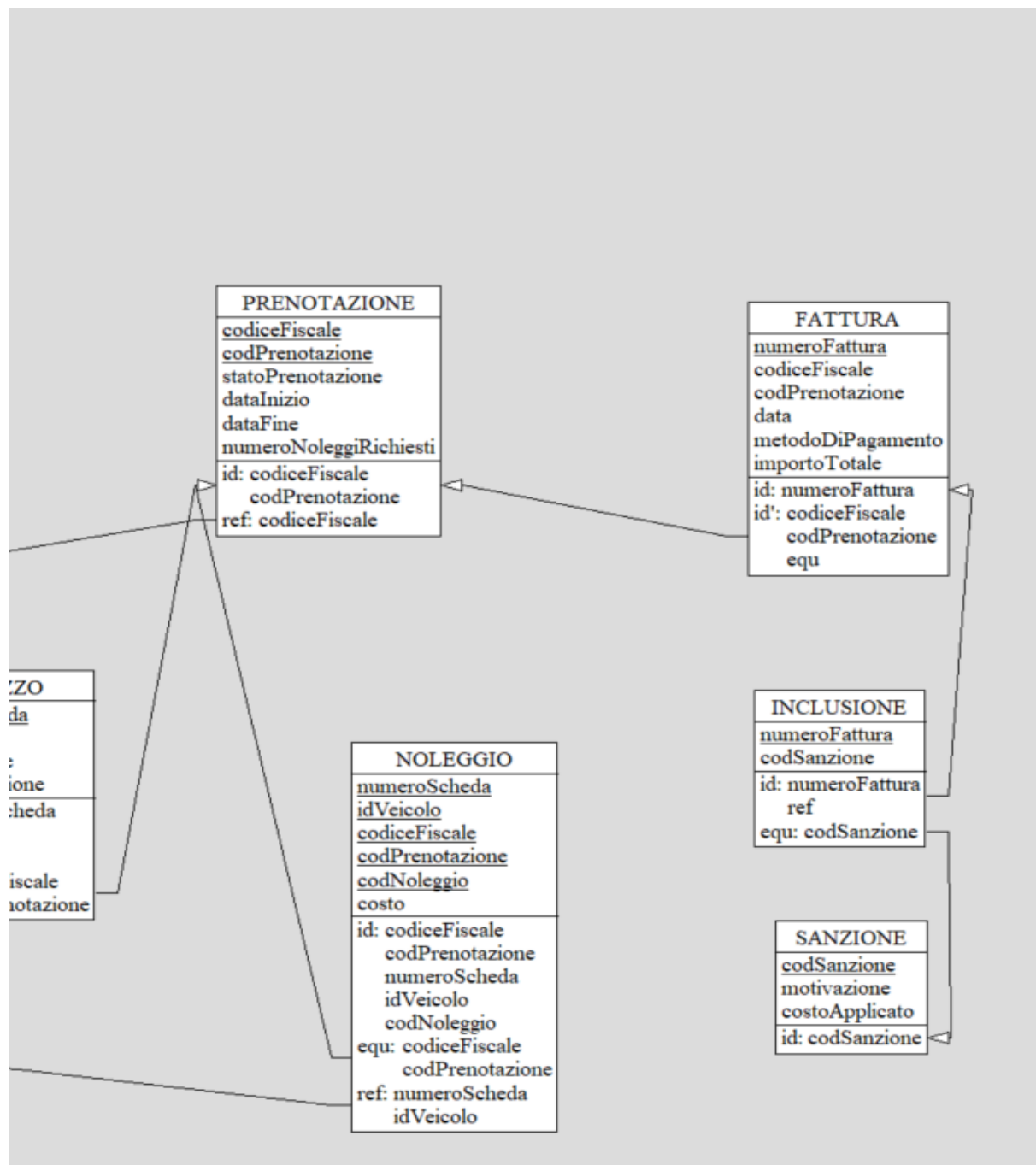
UTILIZZI (idPrenotazione:Prenotazioni, numVeicolo:Veicoli)

VEICOLI (idVeicolo, schedaTecnica:SchedeTecniche, targa,
chilometraggio, statoManutenzione, costoPerGiornata)
UNIQUE(targa)

SCHEDETECNICHE (numeroScheda, tipoVeicolo, casaProduttrice,
modello, annoDiProduzione, numeroPosti, numeroPorte,
capacitàBagagliaio)

5.1 Schema Relazionale Finale





6 Traduzione delle operazioni in query SQL

Tutti i valori di input sono adeguatamente controllati all'interno del codice dell'applicazione per l'interfaccia utente. Ogni codice/numero/Id utilizzati per identificare le varie entità sono stati impostati come valori auto-incrementali nella piattaforma di MySQL, perciò non verranno inseriti dall'utente. Il controllo di unicità dei valori che la richiedono è ugualmente impostato sulla piattaforma di MySQL.

OP1 - Registrare un nuovo veicolo

```
INSERT INTO veicoli (schedaTecnica, targa, chilometraggio, costoPerGiornata)
VALUES (?, ?, ?, ?)
```

OP2 - Visualizzare la disponibilità di un veicolo

Per visualizzare la disponibilità di un veicolo devo controllare che non esista nessuna prenotazione che lo riguardi tra una determinata data di inizio e una di fine.

```
SELECT CASE WHEN EXISTS
(SELECT *
FROM prenotazioni p JOIN utilizzi u ON p.codPrenotazione = u.idPrenotazione
WHERE u.numVeicolo = ? AND ? <= p.dataFine AND ? <= p.dataInizio)
THEN
'Esiste già una prenotazione in questo intervallo di date per questo veicolo'
ELSE
'La prenotazione per questo veicolo può essere effettuata'
END AS risultato
```

OP3 - Visualizzare la scheda tecnica di un veicolo

```
SELECT *
FROM schedeTecniche
WHERE numeroScheda = (SELECT schedaTecnica
FROM veicoli
WHERE idVeicolo = ?)
```

OP4 - Registrare un nuovo cliente

Essendo la quasi totalità di inserimenti quelli di un cliente è stata impostato il "ruolo" delle entità Persona come "Cliente" di default. Nel caso si volesse implementare la registrazione di un impiegato, basta inserire la stringa "Impiegato" nel valore dell'attributo "ruolo".

```
INSERT INTO persone (codiceFiscale, nome, cognome, indirizzo_via, indirizzo_numeroCivico, indirizzo_citta, indirizzo_CAP, numeroDiTelefono, indirizzoEmail)
VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)
```

```
INSERT INTO clienti (CFCliente, numeroPatenteGuida, indirizzoFatturazione_via,  
indirizzoFatturazione_numeroCivico, indirizzoFatturazione_citta, indirizzoFatturazione_-  
CAP)  
VALUES (?, ?, ?, ?, ?, ?)
```

OP5 - Visualizzare lo storico noleggi di un cliente

Visualizza tutti i dati rilevanti riguardanti i noleggi effettuati dal cliente.

```
SELECT  
n.codNoleggio,  
n.codPrenotazione,  
n.veicolo,  
n.costi,  
p.dataInizio,  
p.dataFine,  
p.statoPrenotazione  
FROM  
noleggi n  
JOIN  
prenotazioni p ON n.codPrenotazione = p.codPrenotazione  
JOIN  
clienti c ON p.cliente = c.CFCliente  
WHERE  
c.CFCliente = ?  
ORDER BY  
p.dataInizio DESC
```

OP6 - Visualizzare i 10 clienti con più noleggi

```
SELECT CFCliente  
FROM clienti  
ORDER BY numeroNoleggiConclusi DESC  
LIMIT 10
```

OP7 - Registrare una nuova prenotazione

All'interno della tabella relativa alle prenotazioni sono stati inseriti i seguenti valori di default: "statoPrenotazione" = "Da effettuare" "numeroNoleggiRichiesti" = 1 In quanto lo stato di una prenotazione appena registrata sarà sempre quello sopraindicato e la maggior parte delle prenotazioni riguarderà, come da stima, un solo noleggio.

Nel caso debba registrare una prenotazione con un singolo noleggio richiesto:

```
INSERT INTO prenotazioni (cliente, dataInizio, dataFine)  
VALUES (?, ?, ?);
```



```
INSERT INTO utilizzi (idPrenotazione, numVeicolo)
VALUES (LAST_INSERT_ID(), ?)
```

Nel caso debba registrare una prenotazione con 2 noleggi richiesti:

```
INSERT INTO prenotazioni (cliente, dataInizio, dataFine, numeroNoleggiRichiesti)
VALUES (?, ?, ?, 2);
```

```
INSERT INTO utilizzi (idPrenotazione, numVeicolo)
VALUES (LAST_INSERT_ID(), ?)
```

OP8 - Modificare una prenotazione

```
UPDATE prenotazioni
SET dataInizio = ?, dataFine = ?
WHERE codPrenotazione = ?
```

```
UPDATE utilizzo
SET numVeicolo = ?
WHERE idPrenotazione = ?
```

OP9 - Cancellare una prenotazione

```
DELETE FROM prenotazioni
WHERE codPrenotazione = ?
```

```
DELETE FROM utilizzo
WHERE idPrenotazione = ?
```

OP10 - Attivare un noleggio

Per calcolare il costo effettuo il conteggio dei giorni per i quali la prenotazione del veicolo è stata richiesta e li moltiplico per il "costoPerGiornata" del veicolo in questione.

```
INSERT INTO noleggi (codPrenotazione, veicolo, costo)
SELECT
u.idPrenotazione,
u.numVeicolo,
(DATEDIFF(p.dataFine, p.dataInizio) + 1) * v.costoPerGiornata AS costo
FROM
utilizzi u
JOIN
prenotazioni p ON u.idPrenotazione = p.codPrenotazione
JOIN
veicoli v ON u.numVeicolo = v.idVeicolo
WHERE
p.codPrenotazione = ?
```

OP11 - Terminare un noleggio

Quando termino un noleggio devo segnare la prenotazione relativa come conclusa, rimuovere il veicolo interessato dall'entità "Utilizzo" e cancellare la tupla relativa al noleggio terminato.

```
UPDATE prenotazioni AS p
JOIN clienti AS c ON c.CFCliente = p.cliente
SET p.statoPrenotazione = 'Conclusa',
c.numeroNoleggiConclusi = c.numeroNoleggiConclusi + p.numeroNoleggiRichiesti
WHERE p.codPrenotazione = (SELECT codPrenotazione FROM noleggi WHERE codNoleggio = ?);
```

```
DELETE FROM utilizzi
WHERE idPrenotazione = (SELECT codPrenotazione FROM noleggi WHERE codNoleggio = ?);
```

OP12 - Visualizzare il tasso di utilizzo di un veicolo

Per visualizzare il tasso di utilizzo mensile di un veicolo, si sommano i giorni in cui è stato utilizzato il veicolo per ogni mese, andando poi a calcolare la percentuale. In particolare vengono contati i giorni anche considerando date che si accavallano tra un mese e l'altro, con apposite funzioni che estraggono solo i giorni fino alla fine del mese.

```
SELECT
v.idVeicolo,
MONTH(p.dataInizio) AS mese,
YEAR(p.dataInizio) AS anno,
(SUM(DATEDIFF(LEAST(LAST_DAY(p.dataInizio), p.dataFine),
GREATEST(DATE_FORMAT(p.dataInizio, '%Y-%m-01'), p.dataInizio)) + 1) /
DAY(LAST_DAY(p.dataInizio))) * 100 AS tasso_utilizzo_mensile
FROM
utilizzi u
JOIN
prenotazioni p ON u.idPrenotazione = p.codPrenotazione
JOIN
veicoli v ON u.numVeicolo = v.idVeicolo
WHERE
v.idVeicolo = ? AND p.dataInizio <= LAST_DAY(p.dataInizio)
GROUP BY
v.idVeicolo,
mese,
anno
```

OP13 - Visualizzare i veicoli più noleggiati

Per visualizzare i veicoli più noleggiati vado a calcolarne il tasso di utilizzo per ognuno, per poi ordinarli in modo decrescente formato una "classifica".

```

SELECT
v.idVeicolo,
CONCAT(YEAR(p.dataInizio), '-', MONTH(p.dataInizio)) AS mese_anno,
(SUM(DATEDIFF(LEAST(LAST_DAY(p.dataInizio), p.dataFine),
GREATEST(DATE_FORMAT(p.dataInizio, '%Y-%m-01'), p.dataInizio)) + 1) /
DAY(LAST_DAY(p.dataInizio))) * 100 AS tasso_utilizzo_mensile
FROM
utilizzi u
JOIN
prenotazioni p ON u.idPrenotazione = p.codPrenotazione
JOIN
veicoli v ON u.numVeicolo = v.idVeicolo
GROUP BY
v.idVeicolo,
YEAR(p.dataInizio),
MONTH(p.dataInizio)
ORDER BY
tasso_utilizzo_mensile DESC

```

OP14 - Emettere una nuova fattura per una prenotazione

Per calcolare l'importoTotale di una fattura vado a selezionare il costo del singolo o doppio noleggio della prenotazione a cui si riferisce.

```

INSERT INTO fatture (prenotazione, data, metodoDiPagamento, importoTotale)
VALUES (?, CURRENT_DATE(), ?, (SELECT SUM(n.costo) AS costoTotale
FROM noleggi n
JOIN prenotazioni p ON n.codPrenotazione = p.codPrenotazione
WHERE p.codPrenotazione = ?))

```

OP15 - Generare una sanzione per una prenotazione

Quando genero una sanzione devo aggiornare anche l'importoTotale della fattura alla quale la includo.

```

INSERT INTO sanzioni (motivazione, costoApplicato)
VALUES (?, ?);

```

```

INSERT INTO inclusioni (fattura, codSanzione)
VALUES ((SELECT numeroFattura
FROM fatture
WHERE prenotazione = ?), LAST_INSERT_ID());

```

```

UPDATE fatture
SET importoTotale = importoTotale + (SELECT SUM(s.costoApplicato)
FROM inclusioni i
JOIN sanzioni s ON i.codSanzione = s.codSanzione)

```

```
WHERE i.fattura = (SELECT fattura
FROM inclusioni
WHERE codSanzione = LAST_INSERT_ID()))
WHERE prenotazione = ?
```

OP16 - Visualizzare il fatturato mensile

Per visualizzare il fatturato di ogni mese si raggruppano per mese e anno le somme di tutti gli importi delle fatture emesse.

```
SELECT
DATE_FORMAT(data, '%Y-%m') AS mese_anno,
SUM(importoTotale) AS fatturato_mensile
FROM
fatture
GROUP BY
DATE_FORMAT(data, '%Y-%m')
ORDER BY mese_anno;
```

7 Progettazione dell'applicazione

7.1 Descrizione dell'architettura

L'applicazione per l'interfaccia utente è stata realizzata in Java, utilizzando la libreria JDBC per la connessione al database, il quale risiede in locale. Il DBMS usato è MySQL. La parte di controller del progetto contiene una classe che connette l'applicazione con il DBMS, quest'ultima utilizzata in tutte le altre classi che contengono metodi per ogni operazione e rispettiva query da eseguire. La GUI è stata impostata in modalità schermo intero all'apertura e per ogni sezione che viene visitata, per garantire un migliore adattamento dei vari campi di testo e tabelle di output, ma può essere ridotta a finestra tramite apposito pulsante "ESC" come indicato anche all'interno dell'applicazione. All'apertura è stata inserita una schermata di autenticazione, che al momento contiene una logica molto basica, ma facilmente scalabile.

Nome utente:

admin

Password:

.....

Accedi

Una volta effettuato l'accesso, sarà presente una menù superiore con il quale visitare le relative sezioni, ognuna contenente pulsanti che apriranno a loro volta una scena per ogni operazione, contenente i vari campi di testo ed altri strumenti per inserire i dati di input e visualizzare quelli di output.

Gestione Clienti | Gestione Veicoli | Gestione Prenotazioni | Gestione Noleggi | Gestione Fatture | Gestione Sanzioni

Registra qui un nuovo cliente

Visualizza lo storico noleggi di un cliente

Visualizza i 10 migliori clienti

Registra qui un nuovo cliente

Inserisci il codice fiscale

Inserisci il nome

Inserisci il cognome

Inserisci la via di residenza

Inserisci il numero civico di residenza

Inserisci la città di residenza

Inserisci il CAP di residenza

Inserisci il numero di telefono

Inserisci indirizzo email (facoltativo)

Inserisci il numero della patente di guida

☐ Usa l'indirizzo di residenza per la fatturazione

Inserisci la via dell'indirizzo di fatturazione

Inserisci il numero civico dell'indirizzo di fatturazione

Inserisci la città dell'indirizzo di fatturazione

Inserisci il CAP dell'indirizzo di fatturazione

Aggiungi Nuovo Cliente

Response

Come specificato in precedenza, i vari controlli sui dati di input sono stati implementati direttamente all'interno del codice dell'applicazione in modo esaustivo, con rispettivi handling di eventuali errori.

