

Autonoleggio Gianluca

ABSTRACT

L'autonoleggio Gianluca è un'impresa che gestisce la vendita e il noleggio di vetture che vanno dalle due alle quattro ruote, dalla bassa alla alta cilindrata, dal lusso all'utilitaria comune e molte altre opzioni.

Il cliente che è interessato ad intraprendere un viaggio con noi dovrà usufruire di un account con le sue generalità e la patente in possesso.

Il lavoratore si occuperà della registrazione dei noleggi e delle vetture vendute agli utenti mediante una apposita base di dati, nel quale si terrà traccia di chi noleggia o acquista un mezzo e quale lavoratore ha gestito la trattativa.

Ogni vettura possiede un posto fisso presso dei parcheggi dell'azienda, identificati con indirizzo e città.

I vari mezzi sono forniti dalle migliori aziende automobilistiche in circolazione.

Le vetture che vengono destinate al noleggio vengono naturalmente ricoperte da una polizza personalizzata.

ANALISI DEI REQUISITI

Nel database sono presenti i dati dei **Clienti** con i seguenti attributi:

- CF (codice Fiscale)
- Nome
- Cognome
- Sesso
- Telefono
- Email
- Inizio cliente (data registrazione);

i clienti si differiscono in due tipi, noleggio e concessionaria, la differenza è che i clienti noleggio devono inserire la patente per eseguire un noleggio.

Alcuni clienti possiedono dunque una **Patente**, la quale è caratterizzata da:

- Numero patente
- Tipo
- Data rilascio
- Data scadenza

Le **autovetture** richiedono diverse informazioni, quali:

- Targa
- Modello
- Numero posto parcheggio
- Cilindrata
- Lusso
- Prezzo giornaliero (per noleggio)
- Km (chilometri percorsi)
- Tipo cambio
- Prezzo (di vendita)
- Carburante
- Anno (di produzione)
- Tipo di veicolo (Auto, Moto o Furgone)
- Carico massimo (Furgone)
- Motociclo (Moto)
- Posti passeggeri (Auto)

I veicoli si differenziano in tre tipi: auto, moto e furgone, le auto devono avere l'attributo posti passeggeri compilato, le moto l'attributo motociclo e invece i furgoni il carico massimo.

I veicoli sono lasciati parcheggiati sui **parcheggi** riconosciuti tramite

- Città
- Indirizzo
- Numero di posti (totali)

le vetture possono essere provviste di **Polizza** assicurativa, se destinate al noleggio, caratterizzate da:

- Numero polizza
- Massimale
- Franchigia
- Tipo

Il **lavoratore** è caratterizzato da:

- ID (identificativo)

- Contratto
- Nome
- Cognome

GLOSSARIO DEI TERMINI

| Termine | descrizione | collegamenti |
|-------------------|---|-----------------------------|
| CF | Identificativo univoco del cliente | Acquista, patente, noleggia |
| ID | Identificativo univoco il lavoratore | Acquista, noleggia |
| Targa | Identificativo univoco la vettura | Acquista, noleggia, polizza |
| Indirizzo e città | Identificativo univoco del parcheggio | Autoveicolo |
| Nome fornitore | Identificativo univoco dell'azienda fornitrice di vetture | Autoveicolo |

3 Progettazione Concettuale

3.1 Lista entità

Cliente:

- CF VARCHAR(16) PRIMARY KEY
- nome VARCHAR(255) NOT NULL
- cognome VARCHAR(255) NOT NULL
- sesso VARCHAR(1) NOT NULL
- email VARCHAR(255)
- telefono VARCHAR(12) NOT NULL
- inizio_cliente DATE NOT NULL

L'entità cliente si specializza in una sottocategoria con una generalizzazione totale non esclusiva:

cliente noleggiatore e cliente concessionaria

Il campo email può non essere inserito in quanto il telefono è un attributo richiesto

patente:

- numpatente VARCHAR(10) PRIMARY KEY

- data_rilascio DATE NOT NULL
- data_scadenza DATE NOT NULL
- tipo VARCHAR(3) NOT NULL
- CF VARCHAR(16) NOT NULL

Lavoratore:

- ID VARCHAR(3) PRIMARY KEY
- nome VARCHAR(255) NOT NULL
- cognome VARCHAR(255) NOT NULL
- contratto VARCHAR(255) NOT NULL

Fornitore:

- nome VARCHAR(255) PRIMARY KEY
- indirizzo VARCHAR(255) NOT NULL
- città VARCHAR(255) NOT NULL

Parcheggio:

- indirizzo VARCHAR(255) PRIMARY KEY
- città VARCHAR(255) PRIMARY KEY
- posti_totali INT NOT NULL

Autoveicolo:

- targa VARCHAR(7) PRIMARY KEY
- modello VARCHAR(255) NOT NULL
- anno INT NOT NULL
- km INT NOT NULL
- lusso BOOLEAN NOT NULL
- cambio VARCHAR(255) NOT NULL
- cilindrata INT NOT NULL
- carburante VARCHAR(255) NOT NULL
- indirizzo_parcheggio VARCHAR(255) NOT NULL
- città_parcheggio VARCHAR(255) NOT NULL
- posto_parcheggio INT NOT NULL
- marca VARCHAR(255) NOT NULL

I veicoli noleggiabili hanno l'attributo prezzo_giornaliero INT

I veicoli acquistabili hanno l'attributo prezzo INT

Inoltre, a seconda del tipo mezzo si avranno questi attributi compilati:

- motociclo BOOLEAN, -- se moto specifica se è motociclo o ciclomotore
- carico_massimo INT, -- se furgone specifica il carico massimo
- posti_passeggeri INT, -- se auto specifica i posti

Polizza:

- numero_polizza VARCHAR(12) PRIMARY KEY
- tipo VARCHAR(255) NOT NULL
- massimale INT NOT NULL
- franchigia INT NOT NULL
- targa VARCHAR(7) NOT NULL

Noleggio:

- codice VARCHAR(10) PRIMARY KEY
- data_inizio DATE NOT NULL
- data_fine DATE NOT NULL
- targa VARCHAR(7) NOT NULL
- CF VARCHAR(16) NOT NULL
- ID_lavoratore VARCHAR(3) NOT NULL

Acquisto:

- codice VARCHAR(10) PRIMARY KEY
- data_acquisto DATE NOT NULL
- targa VARCHAR(7) NOT NULL
- CF VARCHAR(16) NOT NULL
- ID_lavoratore VARCHAR(3) NOT NULL

I codici di noleggio e acquisto sono definiti come varchar ma per semplicità sono utilizzati come semplici numeri nel formato carattere

Tabella Relazioni

| Relazione | Entità coinvolte | Descrizione | Attributi |
|------------|---|--|---------------------------------|
| Acquista | Cliente concessionario (0,N) Autoveicolo(0,N) Lavoratore(0,N) | Un cliente può acquistare o meno un veicolo. Un lavoratore può amministrare più acquisti o anche nessuno Possono essere vendute più auto o anche nessuna | Data acquisto |
| Noleggio | Cliente Noleggio(0,N) Autoveicolo(0,N) Lavoratore(0,N) | Un cliente può noleggiare o meno un veicolo. Un lavoratore può amministrare più noleggi o anche nessuno Possono essere noleggiata più auto o anche nessuna | Data inizio noleggio, data fine |
| Possiede | Cliente Noleggio(1,N) Patente(1,1) | Un cliente può avere una o più patenti Una patente ha solo un cliente | nessuno |
| Fornisce | Fornitore(0,N) Autoveicolo(1,1) | Un fornitore fornisce nessuno o più veicoli Il veicolo viene fornito da un solo fornitore | nessuno |
| Assicurato | Autoveicolo(0,1) Polizza(1,1) | Un autoveicolo può essere assicurato oppure no Una polizza deve essere intestata ad un solo veicolo | nessuno |
| Situato | Parcheggio(0,N) Autoveicolo(1,1) | Un parcheggio può avere più veicoli parcheggiati o nessuno Un'auto ha un solo parcheggio | nessuno |

Vincoli non rappresentabili tramite schema E-R:

La data di registrazione dell'utente deve essere uguale o precedente a qualsiasi acquisto o noleggio effettuato da esso.

Un veicolo destinato al noleggio deve possedere una polizza assicurativa.

Due noleggi dello stesso veicolo non possono avere date sovrapposte.

Un veicolo deve avere l'anno di produzione precedente o uguale all'anno corrente.

Un veicolo non può essere acquistato più di una volta.

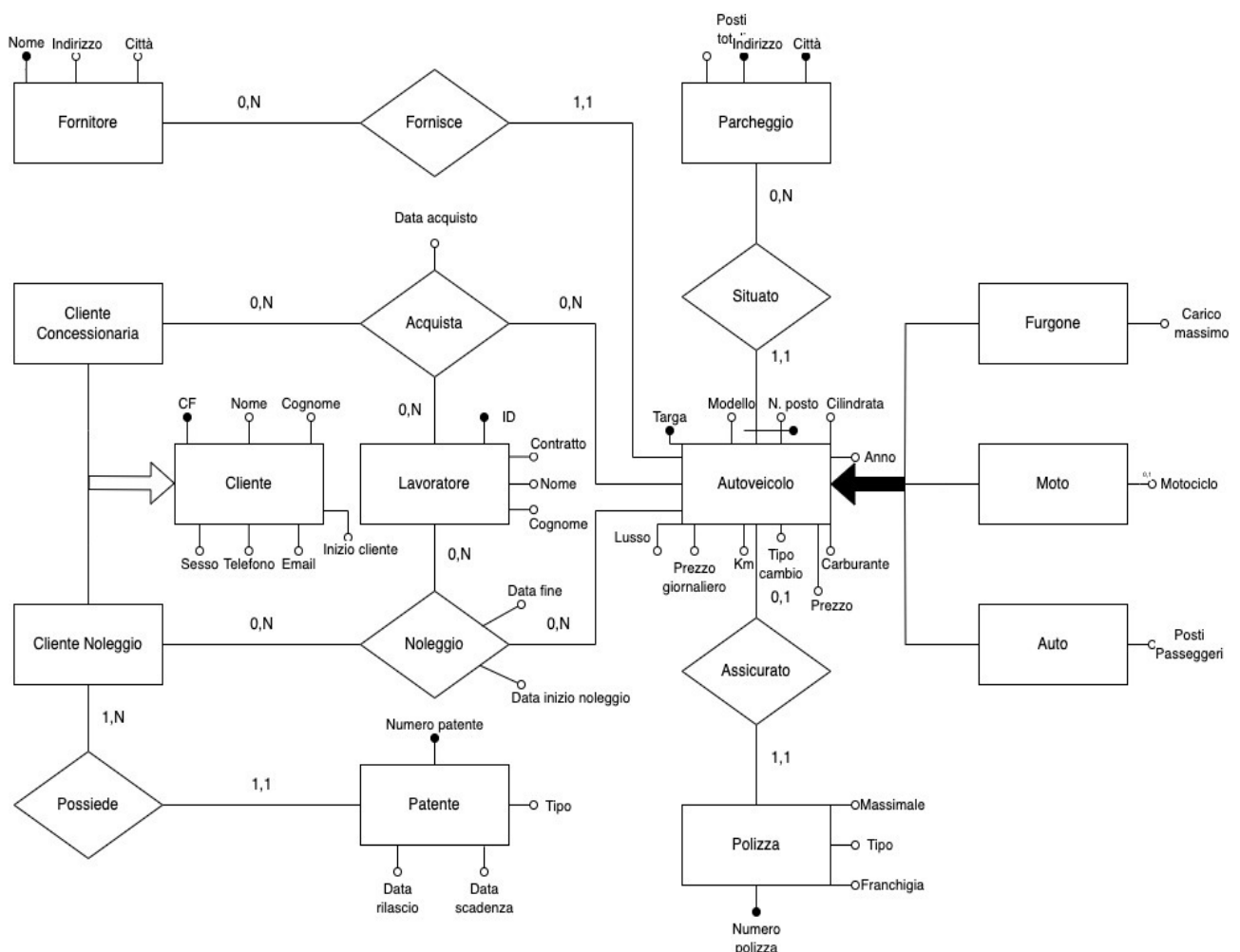
Un veicolo non può avere gli attributi 'pieni' di *motociclo*, *carico_massimo* e *posti_passeggeri* contemporaneamente.

Un veicolo deve possedere gli attributi corrispondenti (di cui sopra) in base al tipo di veicolo che è.

Un noleggio non può terminare prima di essere ancora iniziato.

Vincoli di derivazione: Nessuno.

Schema concettuale



4 Progettazione Logica

4.1 Ristrutturazione

4.1.1 Analisi delle ridondanze

Aggiunta dell'enum tipo per l'entità Autoveicoli per definire il tipo di autoveicolo (Auto, moto o furgone)

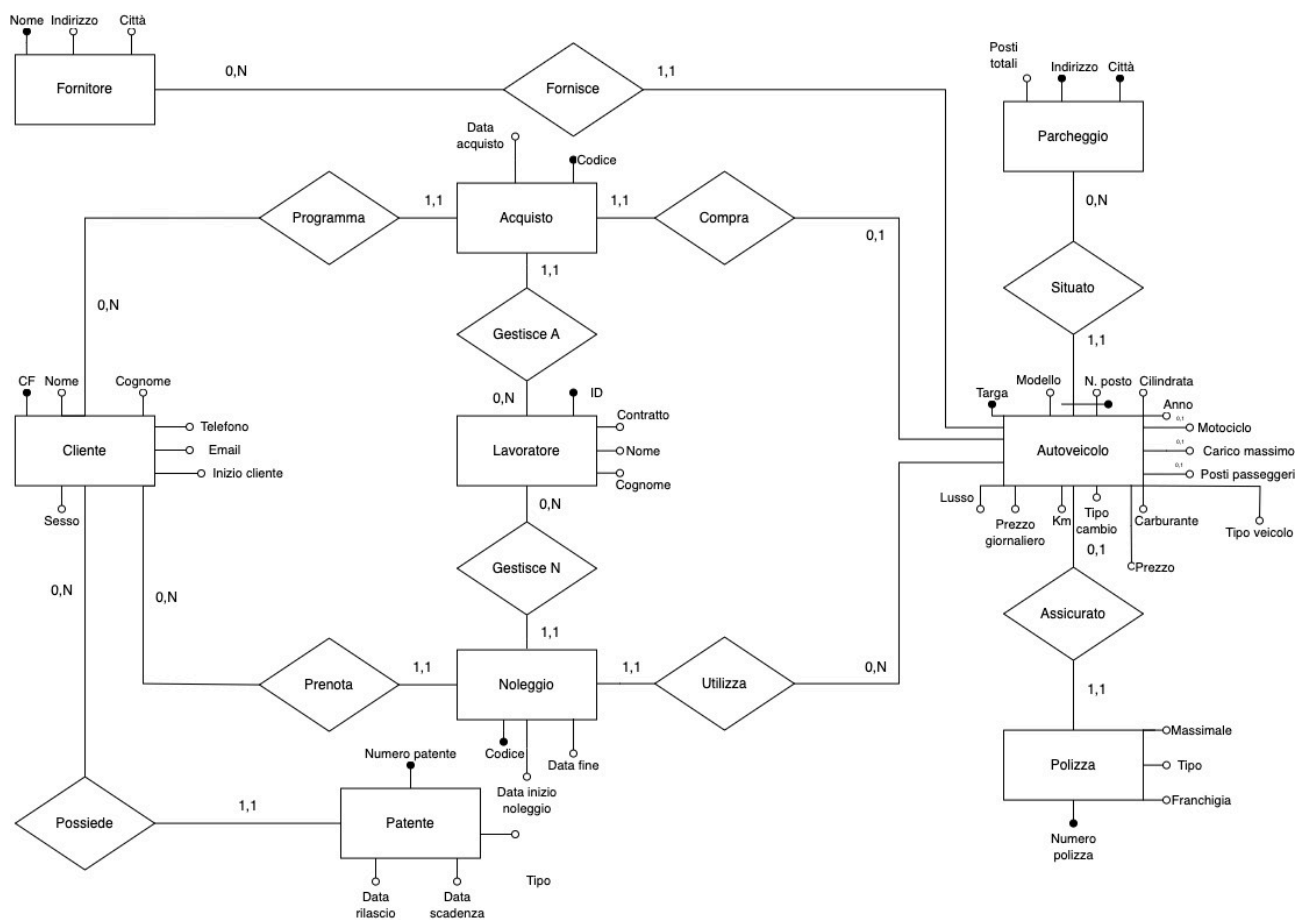
4.1.2 Eliminazione delle generalizzazioni

| Generalizzazione | Risoluzione |
|---|---|
| Cliente <= Cliente concessionaria, cliente noleggio | L'Entità Cliente è una gerarchia non esclusiva, quindi un cliente può essere sia un cliente concessionario che noleggio, solo che il cliente che noleggia deve possedere la patente |
| Autoveicolo <= Furgone, Moto, Auto | Le tre entità figlie vengono inglobate alla entità padre e gestite con un enum, e gli attributi vengono gestiti come possibilmente nulli |

4.1.3 Scelta degli identificatori primari

Sono state create le entità Noleggio e Acquisto con chiavi primarie: Codice.

Schema E-R ristrutturato



4.2 Creazione delle tabelle

Fornitore(Nome, Indirizzo, Città)

Cliente(CF, Nome, Cognome, Sesso, NumPatente)

Patente(NumeroPatente, DataRilascio, DataScadenza, tipo, CF -> Cliente.CF)

Lavoratore(ID, Contratto, Nome, Cognome)

Parcheggio(Città, Indirizzo, PostiTotali)

Autoveicoli(Targa, indirizzoP->Parcheggio.indirizzo, CittàP->Parcheggio.città,
Marca -> Fornitore.Nome, NPosto, Cilindrata, Motociclo, CaricoMassimo, PostiPasseggeri,
TipoVeicolo, Carburante, Prezzo, TipoCambio, Km, PrezzoGiornaliero, Lusso)

Polizza(numero_polizza, Massimale, Tipo, Franchigia, targa->Autoveicoli.Targa)

Acquisto(Codice, CF, Targa, IDLavoratore, Data_acquisto)

Noleggio(Codice, CF, Targa, IDLavoratore, data_inizio_noleggio, Data_fine)

5 Query e indici

5.1 Query

- 1) Mostra tutti i veicoli noleggiabili con una polizza assicurativa con massimale maggiore di 1.000.000

```
SELECT A.targa, A.marca, A.modelo,  
A.tipo_veicolo  
FROM autoveicolo AS A  
JOIN Polizza as P  
ON P.targa = A.targa  
WHERE P.massimale >= 1000000
```

| targa | marca | modelo | tipo_veicolo |
|---------|------------|---------------------|--------------|
| QG613BR | Volkswagen | SC | AUTO |
| BX345HC | Pontiac | Tribute | AUTO |
| BU100EL | Infiniti | Rodeo | AUTO |
| NR781TW | Infiniti | I | AUTO |
| LG070KV | Chevrolet | Suburban 2500 | AUTO |
| DE945DJ | Alfa Romeo | S-Series | AUTO |
| FA966IV | Infiniti | QX | AUTO |
| DY375WP | Porsche | 911 | AUTO |
| WK585IG | Pontiac | Vibe | AUTO |
| XX567XL | Eagle | Discovery Series II | MOTO |
| QD0390H | Ford | Thunderbird | MOTO |
| YP235LW | Dodge | Dakota Club | MOTO |
| LG619NQ | Volvo | Cooper | FURGONE |

- 2) Mostra tutti i fornitori in ordine di autoveicoli forniti e il numero di elementi forniti

```
SELECT A.marca as fornitore, F.città,  
F.indirizzo, COUNT(*) AS numElementiForniti  
FROM autoveicolo AS A  
JOIN fornitore AS F  
ON A.marca = F.nome  
GROUP BY A.marca, F.città, F.indirizzo  
ORDER BY numElementiForniti DESC
```

| fornitore | città | indirizzo | numelementiforniti |
|---------------|-----------------------|----------------------------|--------------------|
| GMC | Sukakarya | 20223 Ridge Oak Drive | 5 |
| Infiniti | Luleå | 1012 Farmco Alley | 4 |
| Ford | Hengfan | 85 Columbus Center | 4 |
| Dodge | Tempe | 2604 Sutteridge Drive | 4 |
| Pontiac | Ardabil | 6 Erie Alley | 3 |
| Saab | Volodars'k-Volyns'kyy | 551 Jackson Circle | 3 |
| Chevrolet | Mulchen | 64 Pepper Wood Terrace | 3 |
| Honda | Pueblo Nuevo | 615 Gateway Terrace | 3 |
| Buick | Pho Thale | 9517 Raven Center | 3 |
| Porsche | Lenningen | 9052 Columbus Alley | 2 |
| Mercedes-Benz | Fengyi | 0 Brown Street | 2 |
| Mercury | Suresnes | 595 Atwood Way | 2 |
| Volvo | Sukakarya | 7 Mallory Pass | 2 |
| Audi | Minneapolis | 2 Graedel Terrace | 1 |
| Alfa Romeo | Lingbei | 28 Pierstorff Pass | 1 |
| Naserati | Fulong | 508 Westport Parkway | 1 |
| Toyota | Nancang | 9085 Russell Court | 1 |
| Jeep | Wirodayan | 12830 Ridge Oak Drive | 1 |
| Eagle | Sritanjung | 29223 Norway Maple Court | 1 |
| Lincoln | Fengyi | 462 North Court | 1 |
| Volkswagen | Lao Suea Kok | 82804 Autumn Leaf Junction | 1 |
| Cadillac | Nevel | 5 Cottonwood Lane | 1 |
| Oldsmobile | Wilamowice | 56 Arrowood Circle | 1 |

- 3) Mostra tutti i clienti con la patente scaduta (in data 18-06-2023)

```
SELECT DISTINCT C.cognome, C.nome, C.cf, C.email, C.telefono, C.email,  
C.inizio_cliente  
FROM cliente AS C
```

JOIN patente AS P
ON P.CF = C.CF
WHERE data_scadenza < '2023-06-18'
ORDER BY C.cognome, C.nome

| cognome | nome | cf | email | telefono | inizio_cliente |
|------------|-----------|------------------|-------------------------------|--------------|----------------|
| Aldington | Gordon | JJYVQV80F70J976H | galdingtonc@imdb.com | 728 296 1030 | 2011-04-13 |
| Clint | Franni | XZKXBT99E53M443W | fclintg@dailymail.co.uk | 776 967 9518 | 2004-10-31 |
| Gimblett | Free | NKIUML24J11Q923R | fgimblettl@blinklist.com | 261 380 0768 | 2009-05-16 |
| Glavis | Margalit | PWKMSJ23W68Q347T | | 243 832 1484 | 2022-09-15 |
| Gotter | Hanni | CFYDLL19I85M469F | | 854 790 0953 | 2006-01-31 |
| Hanselman | Frederich | QJGRHQ64U95J443W | fhanselman13@twitter.com | 551 929 4137 | 2011-05-22 |
| McTeer | Kerk | BSJGVG58P59M263N | | 126 760 6570 | 2022-04-07 |
| Morgans | Lita | TSJPHL97P35G831I | | 406 576 7923 | 2006-06-17 |
| Packham | Clayson | YULQKQ73P86X286L | cpackhamu@elegantthemes.com | 888 735 0431 | 2021-07-03 |
| Robey | Dredi | HZJTVS68X33K590B | drobey1@engadget.com | 843 876 2716 | 2003-08-06 |
| Taylor | Hortensia | VR5QHT76V59T281R | htaylorq@barnesandnoble.com | 359 766 3852 | 2009-07-27 |
| Yerrington | Emelda | EGSAFY87L48P287X | eyerringtonk@seattletimes.com | 505 296 8317 | 2000-01-30 |

4) Mostra il lavoratore che ha venduto più veicoli

SELECT id_lavoratore, nome, cognome, count(*) as numVendite
FROM acquisto
JOIN lavoratore
ON acquisto.id_lavoratore = lavoratore.id
GROUP BY id_lavoratore, nome, cognome
ORDER BY numVendite DESC
LIMIT 1

| id_lavoratore | nome | cognome | numvendite |
|---------------|--------|---------|------------|
| 3 | Kermie | Smaling | 3 |

5) Mostra tutti i parcheggi in cui c'è parcheggiato almeno un veicolo

SELECT indirizzo, citta, count(*) AS
numVeicoli
FROM parcheggio AS P
LEFT JOIN autoveicolo AS A
ON P.indirizzo = A.indirizzo_parcheggio
AND P.citta = A.citta_parcheggio
GROUP BY indirizzo, citta
HAVING count(*) > 0
ORDER BY numVeicoli DESC

| indirizzo | citta | numveicoli |
|-------------------------|--------------|------------|
| 6 Northwestern Circle | Wang Nam Yen | 4 |
| 80 Texas Road | Cochabamba | 4 |
| 231 Oakridge Alley | Jiazhuoyuan | 4 |
| 03531 Eastlawn Lane | Kyprinos | 4 |
| 9 Northfield Parkway | Zougang | 4 |
| 50758 La Follette Alley | Saint Hubert | 3 |
| 0342 Kensington Hill | Limoges | 3 |
| 21284 Mifflin Road | Lidköping | 3 |
| 171 Del Mar Pass | Bagay | 3 |
| 23 Bultman Pass | Messina | 3 |
| 1 Tennessee Avenue | Mlonggo | 3 |
| 1889 3rd Alley | Valleymount | 3 |
| 7 Dakota Point | Anolaima | 3 |
| 536 Thierer Terrace | Waihi Beach | 3 |
| 8 Vera Place | Funaishikawa | 3 |

6) Mostra i clienti fedeli, ovvero quelli che sono clienti da prima del 2010 e hanno noleggiato almeno due veicoli o comprato uno

SELECT DISTINCT *
FROM

((SELECT DISTINCT C.nome, C.cognome, C.cf, C.telefono, C.inizio_cliente
FROM cliente AS C
JOIN acquisto AS A
ON C.cf = A.cf)

UNION

(SELECT C.nome, C.cognome, C.cf, C.telefono, C.inizio_cliente
FROM cliente AS C
JOIN noleggio AS N

```

ON C.cf = N.cf
GROUP BY C.cf
HAVING count(*) >= 2) as fedeli

```

```

WHERE inizio_cliente <= '2010-12-31'

```

| nome | cognome | cf | telefono | inizio_cliente |
|-----------|-----------|------------------|--------------|----------------|
| Brook | Wanklin | OMPWJW04L87M229C | 848 590 8507 | 2006-02-20 |
| Cassy | McCormick | HRKUEB99Z37H176V | 130 622 6576 | 2007-08-15 |
| Dredi | Robey | HZJTVS68X33K590B | 843 876 2716 | 2003-08-06 |
| Ferrell | Jackman | COGIFU01E96P143P | 874 169 2665 | 2001-11-28 |
| Hortensia | Taylor | VRSQHT76V59T281R | 359 766 3852 | 2009-07-27 |
| Kale | Grzes | JVPISK44Y40F049B | 918 788 2462 | 2009-08-31 |
| Lita | Morgans | TSJPHL97P35G831I | 406 576 7923 | 2006-06-17 |
| Roxanne | Bartley | VSOUYZ76G32H046N | 126 480 1921 | 2009-08-18 |

5.2 Query parametriche

7) Mostra i clienti che hanno noleggiato un veicolo di lusso attraverso un lavoratore

```

SELECT C.nome, C.cognome, C.cf, C.sesso, C.email, C.telefono
FROM Cliente AS C
JOIN Noleggio AS N ON N.CF = C.CF
JOIN Autoveicolo AS A ON N.targa = A.targa
JOIN Lavoratore AS L ON L.ID = N.ID_lavoratore
WHERE A.lusso=true AND N.ID_lavoratore = '1'

```

8) Mostra franchigia e massimale della polizza di un noleggio dato l'id del noleggio

```

SELECT P.numero_polizza, A.targa, P.massimale, P.franchigia
FROM polizza AS P
JOIN autoveicolo AS A
ON P.targa = A.targa
JOIN noleggio AS N
ON A.targa = N.targa
WHERE codice = '1'

```

9) Mostra quanto deve pagare un cliente per il suo noleggio dato il codice del noleggio

```

SELECT N.codice as codiceNoleggio, A.targa, A.marca, A.modello, DATE_PART('day',
AGE(N.data_fine, N.data_inizio)) * A.prezzo_giornaliero AS prezzoGiornaliero
FROM noleggio AS N
JOIN autoveicolo AS A
ON N.targa = A.targa
WHERE codice = '7'

```

10) Mostra tutti i veicoli del tipo inserito disponibili nella data inserita al noleggio

```

SELECT targa, marca, modello
FROM autoveicolo
WHERE tipo_veicolo = 'AUTO' AND prezzo IS null

```

EXCEPT

```
(SELECT A.targa, A.marca, A.modello  
FROM autoveicolo AS A  
JOIN noleggio as N  
ON A.targa = N.targa  
WHERE N.data_inizio < '2018-03-01' AND N.data_fine > '2018-03-01')
```

5.3 Indici

La lettura di un veicolo è molto frequente mentre la sua aggiunta lo è meno. Ipotizzando l'espansione del noleggio su scala nazionale questo determinerebbe un buon punto per utilizzare un indice e rendere le ricerche più veloci; inoltre una aggiunta di un veicolo verrebbe gestita da un lavoratore del noleggio mentre la lettura dei veicolo da tutti gli utenti.

```
CREATE INDEX indexTipoVeicolo ON autoveicolo(modello, marca, tipo_veicolo)
```

6 Codice C++

6.1 Restore del database

Per ripristinare il database sarà sufficiente creare un database vuoto su pgAdmin ed eseguire il seguente comando su terminale:

```
psql -U <username> -d <database_name> -f backup.sql
```

6.2 Descrizione dell'utilizzo del codice

Il codice c++ consiste in un solo file *main.cpp* che si compila con il comando:

```
c++ main.cpp -L dependencies/lib -lpq -std=c++11 -o main
```

Per la compilazione saranno necessarie alcune librerie per la connessione con Postgres e PgAdmin.

Prima di compilare è necessario inserire il nome del database e la password per accederci nelle apposite define PG_DB e PG_PASS definite all'inizio del file.

Una volta fatto questo il programma riuscirà a connettersi al database.

Successivamente il programma mostrerà due opzioni: la visualizzazione delle query disponibili (inserendo 1) e la possibilità di terminare il programma (inserendo 2).

Se si decide di visualizzare le query verranno mostrate a schermo tutte le query disponibili, divise tra query normali e query parametriche. Inserire il numero corrispondente alla query che si vuole visualizzare o inserire 0 per tornare al menù principale.

Per quanto riguarda le query parametriche forniranno automaticamente i dati da inserire e le modalità in cui dovranno essere inseriti. In caso di inserimento errato il programma terminerà immediatamente segnalando il problema avvenuto.

6.3 Documentazione del codice

Il codice presenta un vettore di stringhe in cui sono contenute tutte le query in ordine come sono state elencate sopra. Per semplicità il vettore è una variabile globale.

Il programma inizia subito accertandosi che la connessione sia andata a buon fine; in caso di fallimento il programma terminerà immediatamente segnalando l'errore.

void showMenu() mostra il menù principale.

void showQuery() si occupa invece di mostrare le query disponibili.

void selectQuery(PGconn *conn) si occupa di gestire la query che viene selezionata e chiamare la corrispondente funzione che eseguirà l'interrogazione al database.

void queries1To6(PGconn *conn, int k); void queries7To9(PGconn *conn, int k); void query10(PGconn *conn) si occupano di interrogare il database delle rispettive query.

void checkResults(PGresult *res, const PGconn *conn) controlla che i risultati delle interrogazioni siano corretti.

void printResults(PGresult *res) stampa i risultati su console.