

Concessionario FastAuto

1. Abstract

FastAuto è una catena di concessionari di auto e moto e richiede una ristrutturazione del database.

Si dovrà tenere conto dei veicoli venduti dalle singole concessionarie, che possono essere auto nuove, auto usate e moto nuove. Ognuna sarà identificata dal numero seriale, e avrà anche come attributi costo, marca, modello, cilindrata, colore e motore. Per le moto si avrà anche quale livello di patente è necessaria per essere guidata. Delle auto usate si conoscono anche il chilometraggio, la targa e lo stato della macchina per segnalare eventuali danni o malfunzionamenti.

Le sedi inoltre sono identificate dalla loro città, poiché si ha una sola sede per città. Ciascuna di queste avrà anche il fatturato annuale, l'indirizzo e il numero di dipendenti.

I clienti del concessionario verranno identificati tramite codice fiscale e gli verranno richiesti altri dati come nome, cognome, e-mail e numero di telefono.

Le sedi richiedono alle case produttrici di rifornirle: di queste si conosce il nome, il tempo medio di consegna e il tipo di veicolo fabbricato.

Nel database verranno anche memorizzati tutti i dipendenti dell'azienda per ogni sede con l'assegnazione di un codice univoco, oltre al loro nome, cognome, salario e ruolo lavorativo.

I clienti possono eseguire un pagamento unico o a rate, del pagamento si conosce l'importo, IBAN, la data di emissione e lo si identifica con un codice univoco. L'importo può essere maggiore del costo del veicolo se vengono aggiunti degli optional dal cliente, e in caso di un pagamento a rate si salvano anche l'importo della rata e il numero di rate.

Infine, le concessionarie offrono 2 anni di garanzia su tutti i veicoli nuovi e 1 anno sulle auto usate, è possibile estendere la garanzia con pagamento ad un prezzo variabile in base al veicolo. Si terrà conto del tipo della garanzia, del costo e della durata, identificandole direttamente con il numero seriale del veicolo associato.

2. Analisi dei requisiti

2.1 Descrizione testuale

Nella base di dati sono presenti i dati della **Sede**, i quali sono:

- Città
- Indirizzo
- Numero Dipendenti
- Fatturato Annuale
- Nome Prod

Ogni sede si rifornisce da dei **Produttori** di cui si salvano i seguenti dati:

- Nome
- Tempo Consegna
- Tipo Veicolo
- Città Sede

Le sedi hanno anche dei **Dipendenti**:

- Codice
- Nome
- Cognome
- Ruolo
- Salario
- Città Sede

I **Veicoli** possono essere automobili o moto, e ha questi attributi:

- SN
- Costo
- Colore
- Cilindrata
- Marca
- Modello
- Motore

Le **Moto** hanno come attributo anche:

- Livello Patente

Le **Auto** invece si dividono fra **Nuove** ed **Usate**, queste ultime hanno come attributi:

- Stato
- KM
- Targa

Tutti i veicoli nuovi possiedono una **Garanzia** di due anni, mentre quelli usati un solo anno, in entrambi i casi è possibile estenderla:

- SN Veicolo
- Tipo
- Durata
- Costo Estensione

I veicoli verranno poi venduti a dei **Clienti** di cui si salveranno questi dati:

- CF
- Nome
- Cognome
- Telefono
- E-mail

Si tratterà anche il **Pagamento** accordato fra il cliente e la sede con questi dati:

- Codice
- Importo
- Data Emissione
- IBAN
- Rate
- Importo Rata
- Città Sede
- ID Cliente
- SN Veicolo

2.2 Glossario dei termini

Termine	Descrizione	Collegamenti
Cliente	Individuo che ha eseguito almeno un acquisto presso l'azienda	Pagamento
Pagamento	Acquisto di un veicolo effettuato nell'azienda	Cliente, Sede, Veicolo
Veicolo	Prodotto venduto dai concessionari dell'azienda	Pagamento, Garanzia, Entità padre di Automobile e Motociclo
Garanzia	Tutela al consumatore in caso di veicoli difettosi o danneggiati	Veicolo
Automobile	Veicolo a quattro ruote adibito al trasporto su strada	Entità figlia di Veicolo, Entità padre di Nuovo e Usato
Motociclo	Veicolo a due ruote adibito al trasporto su strada	Entità figlia di Veicolo
Nuovo	Automobile fabbricata da un produttore e venduta al cliente tramite una concessionaria	Entità figlia di Automobile
Usato	Automobile acquistata ad un privato da una concessionaria, per poi essere rivenduta	Entità figlia di Automobile

Dipendente	Individuo che lavora presso FastAuto in qualsiasi posizione	Sede
Produttore	Fabbricatore di auto o moto che costruisce i veicoli che verranno venduti dall'azienda	Sede
Sede	Definisce una delle filiali dell'azienda, dov'è situata e che relazioni ha con clienti e produttori	Dipendente, Produttore, Pagamento
Rate	In caso di un pagamento a rate, definisce in quante parti è frazionato	Attributo di Pagamento
Importo	Costo totale dell'acquisto eseguito dal cliente, compreso del prezzo base del veicolo e degli eventuali optional	Attributo di Pagamento
Importo_Rata	In caso di un pagamento a rate, definisce l'importo della singola rata	Attributo di Pagamento
Cilindrata	Indica la capacità del motore del veicolo	Attributo di Veicolo
Motore	Indica il tipo di tecnologia motrice utilizzata per il veicolo (elettrico, ibrido, benzina, diesel)	Attributo di Veicolo
Costo_estensione	Costo complessivo dell'eventuale estensione della garanzia	Attributo di Garanzia
Tipo	Tipo di garanzia applicata al veicolo acquistato	Attributo di Garanzia
Durata	Longevità complessiva della copertura della garanzia	Attributo di Garanzia
Tempo_consegna	Tempo medio del produttore di consegna dei veicoli ordinati dall'azienda	Attributo di Produttore
Tipo_veicolo	Tipo di veicolo fabbricato da un determinato produttore: auto, moto o entrambi	Attributo di Produttore
Costo	Prezzo del modello di veicolo in questione	Attributo di Veicolo
Salario	Stipendio mensile del dipendente in questione	Attributo di Dipendente
Ruolo	Posizione assegnata al dipendente dell'azienda in base al compito che viene svolto	Attributo di Dipendente
Fatturato	Fatturato annuale di una delle sedi dell'azienda	Attributo di Sede

Liv_patente	Indica il livello di patente richiesto per poter guidare un modello di motociclo	Attributo di Motociclo
Stato	Condizione del veicolo usato, commento sulla sua usura ed eventuali danni	Attributo di Usato
Km	Chilometraggio dell'auto usata	Attributo di Usato
Targa	Identifica la targa già presente sull'auto usata	Attributo di Usato

2.3 Operazioni

OPERAZIONE	TIPO	FREQUENZA
Ricerca dipendenti con salario > 1600	L	20 al giorno
Aggiunta di un nuovo pagamento	S	40 al giorno
Controllo disponibilità veicoli con costo < 10.000	L	15 al giorno
Controllo produttori con tempo di consegna < 50	L	10 al giorno
Estensione della garanzia su un veicolo	S	15 al giorno
Controllo disponibilità auto usate con km < 80.000	L	10 al giorno
Aggiornamento dati personali del cliente	S	5 al giorno

3. Progettazione Concettuale

3.1 Lista Entità

(Se non specificato attributo NOT NULL)

Sede

- Città: varchar(50), primary key
- Indirizzo: varchar(100)
- Num_Dip: int
- Fatturato: decimal(10,2)

Produttore

- Nome: varchar(50), primary key
- Tempo Consegna: int
- Tipo Veicolo: varchar(50)

Dipendente

- Codice: int primary key
- Nome: varchar(50)
- Cognome: varchar(50)
- Ruolo: varchar(50)
- Salario: decimal(10,2)

Veicolo

- SN: int primary key
- Costo: decimal(10,2)
- Colore: varchar(50)
- Cilindrata: int, può essere NULL
- Marca: varchar(50)
- Modello: varchar(50)
- Motore: varchar(50)

Motociclo

- Liv_Patente: varchar(2)

Usato

- Targa: varchar(7)
- Stato: varchar(100)
- KM: int

Cliente

- CF: varchar(16) primary key
- Nome: varchar(50)
- Cognome: varchar(50)
- Telefono: varchar(20)
- Email: varchar(50)

Pagamento

- Cod: int primary key
- Importo: decimal(10,2)
- Rate: int, può essere NULL
- Importo_Rata: decimal(10,2), può essere NULL
- Data Emissione: date
- IBAN: varchar(27)

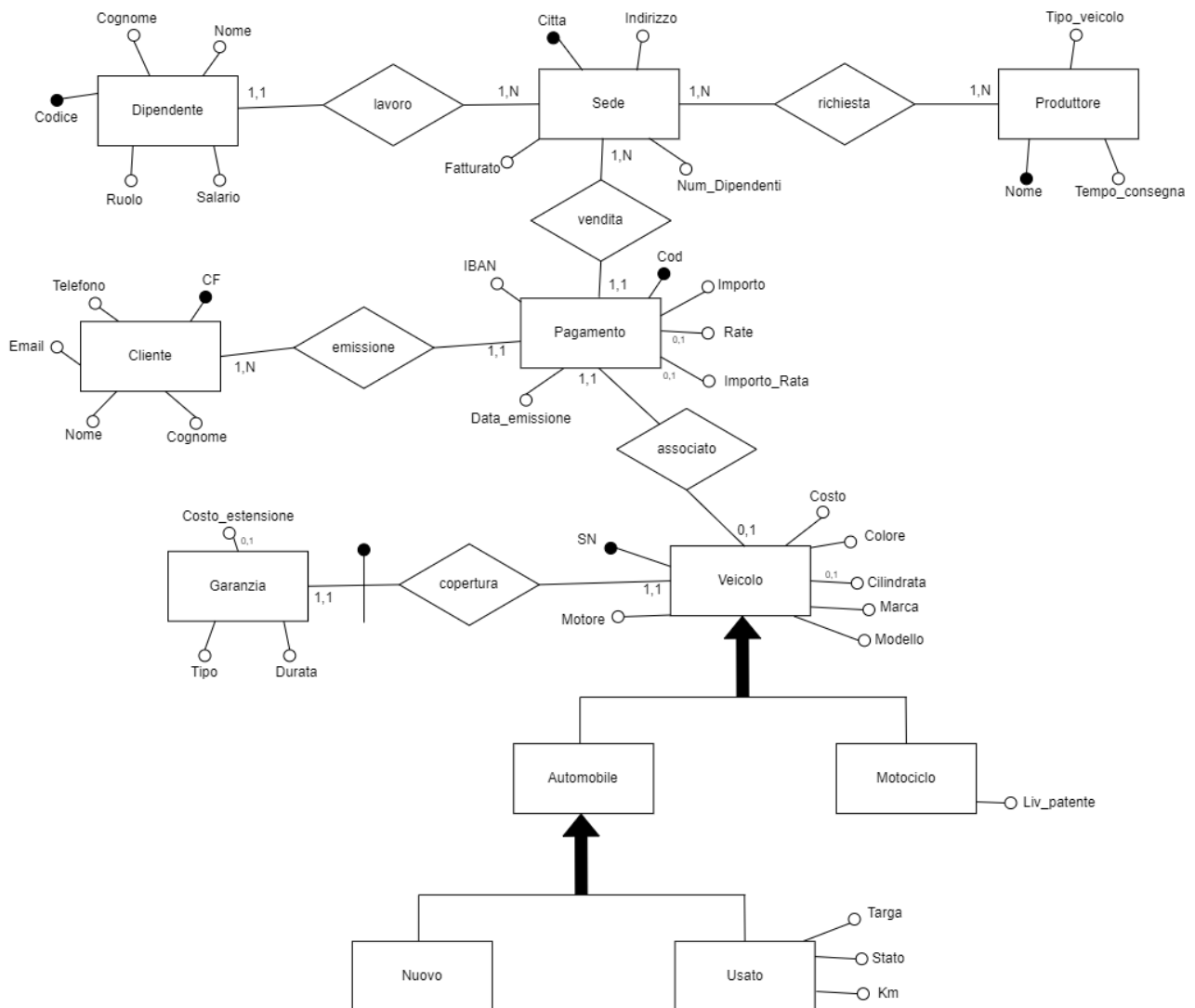
Garanzia

- Durata: int
- Tipo: varchar(50)
- Costo_Estensione: decimal(10,2), può essere NULL

3.2 Tabella delle relazioni

Relazione	Entità coinvolte	Descrizione	Attributi
Associato	Pagamento(1,1) Veicolo(0,1)	Un pagamento è associato ad un solo veicolo, mentre un veicolo può essere legato ad un pagamento o restare invenduto	Nessuno
Lavoro	Sede(1,N) Dipendente(1,1)	Ogni sede ha più dipendenti, un dipendente lavora solo per una sede	Nessuno
Richiesta	Sede(1,N) Produttore(1,N)	Ogni sede ha più di un produttore da cui ordina i veicoli e ogni produttore può vendere i mezzi a più di una sede	Nessuno
Vendita	Sede(1,N) Pagamento(1,1)	Per ogni sede avvengono molti acquisti di veicoli, ma ogni pagamento è associato ad una sola sede	Nessuno
Emissione	Pagamento(1,1) Cliente(1,N)	Ogni pagamento è associato ad un solo cliente, il quale però può compiere più di un acquisto	Nessuno
Copertura	Veicolo(1,1) Garanzia(1,1)	Ogni veicolo è coperto da una sola garanzia complessiva, così come ogni garanzia copre un solo veicolo	Nessuno

3.3 Schema E-R



4. Progettazione Logica

4.1 Ristrutturazione

4.1.1 Analisi delle ridondanze

Ci sono due ridondanze nello schema E-R, la prima si trova nella relazione **Sede** ed è "Num_Dipendenti", poiché è facilmente trovabile il numero di dipendenti tramite una query.

La seconda ridondanza si trova in **Pagamento**, "Importo_Rata" che è calcolabile eseguendo una operazione di divisione $\text{Importo} / \text{Rate}$. Nonostante vadano utilizzati più accessi per calcolare le rate, si risparmia molto in termini di volume impiegato nell'archiviazione dei dati.

Stato DB	Operazione	Attributi utilizzati	Tipo	Accessi
Non ottimizzato	Aggiornamento pagamento rata	Importo_Rata	L	60
Ottimizzato	Aggiornamento pagamento rata	Importo, Rate	L	30

Stato DB	Attributi presenti	Volume
Non ottimizzato	Importo, Rate, Importo_Rata	213
Ottimizzato	Importo, Rate	142

4.1.2 Eliminazione delle generalizzazioni

Tutte le entità figlie di **Veicolo** sono state eliminate e gli attributi portati all'interno di **Veicolo** mettendoli come 0,1. Lo stesso è successo per le entità **Nuovo** e **Usato**.

Ho seguito questa strada per l'eliminazione delle generalizzazioni poiché ci sono pochi attributi nelle relazioni figlie, alcune delle quali sono prive di attributi. Facendo così si semplifica il DB e si snellisce la ricerca dei veicoli.

4.1.3 Scelta degli identificatori primari

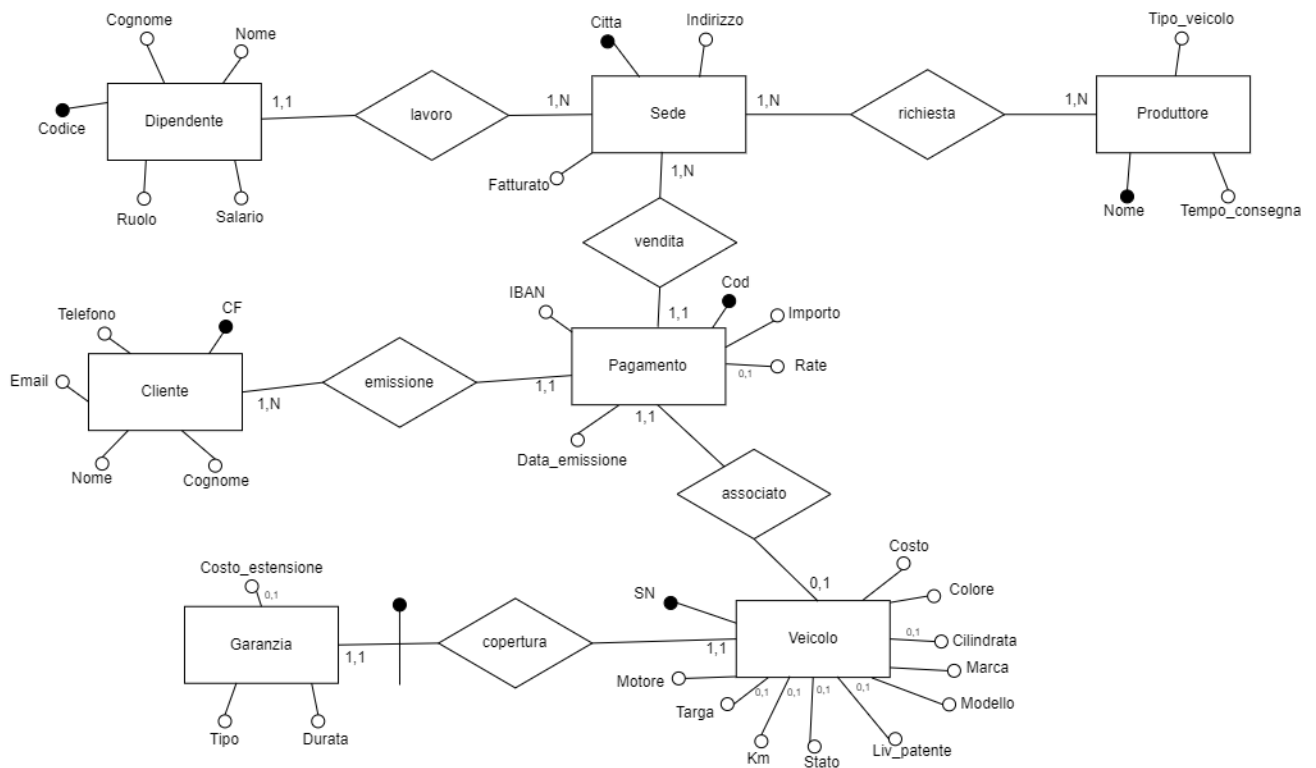
Si è scelto di usare come Chiave Primaria **Città** in **Sede** perché esiste solo una sede per città e quindi è facilmente individuabile. Anche per **Produttori** si è valutato che utilizzando un **Nome** come identificativo il DB sarebbe stato notevolmente semplice, visto che non esistono produttori di automobili e/o motocicli con lo stesso nome. Si identifica il **Cliente** tramite il suo **CF**.

Il **Dipendente** ha invece un **Codice** univoco che lo distingue dagli altri, così da poter ricercare facilmente ogni lavoratore per sede.

Per ogni **Veicolo** invece si tiene traccia dei loro Numeri Seriali, **SN**, visto che arrivano con codici univoci direttamente dalla fabbrica.

Per **Pagamento** invece si è pensato di usare un **Codice** così da poter tenere traccia di ogni pagamento eseguito.

4.2 Schema E-R Ristrutturato



4.3 Creazione delle tabelle

Per indicare una chiave esterna: A -> B, dove B è chiave esterna di A.

Sede (Citta, Indirizzo, Fatturato,)

Produttore (Nome, Tempo_Consegna, Tipo_Veicolo)

Richiesta (Citta_Sede -> Sede.Citta, Nome_Prod -> Produttore.Nome)

Dipendente (Codice, Nome, Cognome, Ruolo, Salario, Citta_Sede -> Sede.Citta)

Veicolo (Sn, Costo, Colore, Cilindrata, Marca, Modello, Liv_Patente, Stato, KM, Targa, Motore)

Cliente (CF, Nome, Cognome, Telefono, Email)

Pagamento (Cod, Importo, Rate, Data_Emissione, IBAN, Citta_Sede -> Sede.Citta, CF_Cliente -> Cliente.CF, SN_Veicolo -> Veicolo.SN)

Garanzia (SN_Veicolo -> Veicolo.SN, Tipo, Durata, Costo_Est)

5. Query e Indici

5.1 Query

1. Contare il numero di clienti con garanzia maggiore o uguale a 3 anni

```
SELECT COUNT(Cl.cf) AS NumGarEstese, Gar.durata
FROM Cliente AS Cl, Pagamento AS Pag, Veicolo AS Vei, Garanzia AS Gar
WHERE Cl.cf = Pag.cf_cliente AND Pag.sn_veicolo = Vei.sn AND Vei.sn = Gar.sn_veicolo AND Gar.durata >= 3
GROUP BY(Gar.durata);
```

Output

	numgarestese bigint	durata integer
1	1	3
2	1	5
3	4	4

2. Selezionare i modelli di veicoli venduti con un importo maggiore di 35000 euro ordinandole in ordine decrescente

```
SELECT DISTINCT Vei.modello, Pag.importo
FROM Veicolo AS Vei, Pagamento AS Pag
WHERE Vei.sn = Pag.sn_veicolo AND Pag.importo >= 35000
ORDER BY(Pag.importo) DESC;
```

Output

	modello character varying (50)	importo numeric (10,2)
1	E-tron	92860.00
2	M3	68450.00
3	Stelvio	58230.00
4	Giulia	50000.00
5	Megane E-tech	40260.00
6	308	38000.00
7	Civic	37240.00
8	Sportage	35000.00

3. Selezionare le marche e i modelli dei veicoli vendute dalla sede di Milano

```
SELECT Vei.marca, Vei.modello, Se.citta
FROM Sede AS Se, Pagamento AS Pag, Veicolo AS Vei
WHERE Se.citta = Pag.citta_sede AND Pag.sn_veicolo = Vei.sn AND Se.citta = 'Milano';
```

Output

	marca character varying (50)	modello character varying (50)	citta character varying (50)
1	Opel	Corsa	Milano
2	Audi	E-tron	Milano

4. Calcolare l'importo totale delle vendite effettuate dalle sedi con un numero di dipendenti maggiore di 5

```
SELECT SUM(Pag.importo) AS SommaTot, Se.citta, COUNT(Dip.codice) AS NumDip
FROM Pagamento AS Pag, Sede AS Se, Dipendente AS Dip
WHERE Se.citta = Pag.citta_sede AND Se.citta = Dip.citta_sede
GROUP BY(Se.citta)
HAVING (COUNT(Dip.codice) >= 5);
```

Output

	sommatot numeric	citta [PK] character varying (50)	numdip bigint
1	313040.00	Berlino	8
2	207800.00	Roma	8
3	446680.00	Milano	8
4	137610.00	Parigi	6
5	247000.00	Padova	10
6	159000.00	Napoli	6

5. Selezionare i clienti che hanno acquistato un'auto usata con più di 70000 km, raggruppando per marca, ordinandoli per i km in modo decrescente

```
SELECT Cl.nome, Cl.cognome, Vei.km, Vei.marca
FROM Cliente AS Cl, Pagamento AS Pag, Veicolo AS Vei
WHERE Cl.cf = Pag.cf_cliente AND Pag.sn_veicolo = Vei.sn AND Vei.km > 70000
GROUP BY(Cl.cf, Vei.km,Vei.marca)
ORDER BY (Vei.km) DESC;
```

Output

	nome character varying (50)	cognome character varying (50)	km integer	marca character varying (50)
1	Michele	Ultimo	91000	Nissan
2	Elisa	Magò	89000	Alfa Romeo
3	Matteo	Fiori	77000	Volkswagen

6. Selezionare i meccanici nelle sedi in cui sono state vendute veicoli per una somma maggiore di 70000 euro, ordinandoli per il loro salario

```
SELECT Dip.nome, Dip.cognome, Dip.salario, SUM(Pag.importo) AS SommaTot, Se.citta
FROM Pagamento AS Pag, Sede AS Se, Dipendente AS Dip
WHERE Se.citta = Pag.citta_sede AND Se.citta = Dip.citta_sede
GROUP BY(Dip.nome, Dip.cognome, Dip.salario,Se.citta)
HAVING (SUM(Pag.importo) > 70000)
ORDER BY Dip.salario DESC;
```

Output

	nome character varying (50)	cognome character varying (50)	salario numeric (10,2)	sommatot numeric	citta character varying (50)
1	Mattia	Moreti	3100.00	78260.00	Berlino
2	Ugo	Valente	2900.00	111670.00	Milano
3	Alvise	Olivia	2800.00	111670.00	Milano
4	Alessandro	Sorrentino	2400.00	78260.00	Berlino
5	Edoardo	Mancini	2100.00	71000.00	Palermo
6	Ivano	Bassi	1860.00	78260.00	Berlino
7	Roberto	Barbieri	1600.00	111670.00	Milano
8	Giulia	Esposito	1600.00	71000.00	Palermo
9	Carlo	Marini	1400.00	111670.00	Milano
10	Enzo	Alti	600.00	78260.00	Berlino

7. Selezionare i clienti che hanno scelto il pagamento a rate, calcolando l'importo delle rate arrotondato e ordinandole in modo decrescente

```
SELECT Cl.nome, Cl.cognome, ROUND(Pag.importo/Pag.rate) AS Importo_Rate
FROM Cliente AS Cl, Pagamento AS Pag
WHERE Cl.cf = Pag.cf_cliente AND Pag.rate is not null
ORDER BY (Pag.importo/Pag.rate) DESC;
```

Output

	nome character varying (50)	cognome character varying (50)	importo_rate numeric
1	Andrea	Ferrari	2852
2	Francesca	Monti	2617
3	Mario	Rossi	2579
4	Antonio	Blu	2183
5	Sara	Ruberti	1678
6	Michele	Ultimo	1500
7	Sabrina	Canada	1053
8	Luigi	Verdi	875
9	Dario	Ducati	832

5.2 Indici

Vista la mole di veicoli che si immettono e si modificano, si è scelto di creare un indice per il numero seriale di Veicolo così da velocizzare la ricerca e la modifica di quel dato.

Codice

```
CREATE INDEX Indice_Veicolo ON Veicolo(sn);
```

6. Codice C++

6.1 Descrizione del codice

Il progetto dovrà chiamarsi “Progetto_FastAuto” e come password chiederà “FastAuto”.

Il codice C++ per eseguire le query è tutto all'interno di un unico file .cpp, nominato “Codice_C++.cpp”, esso va compilato tramite comando:

```
g++ Codice_C++.cpp -o codice -lpq
```

È necessario avere la cartella dependencies nello stesso percorso del file .cpp (come fornito nella cartella .zip consegnata).

6.2 Documentazione Codice

Queste sono alcune funzioni utilizzate nel codice C++:

```
sprintf(ConnectionInfo, " user =%s password =%s dbname =%s hostaddr =%s port =%d", PG_USER, PG_PASS, PG_DB, PG_HOST, PG_PORT);
PGconn* conn;
```

Ritorna una connessione al database, usando i parametri passati nelle credenziali di accesso; in caso di non connessione ritornerà un messaggio di errore.

```
// Query 2
cout << "Query 2. Selezionare i modelli di veicoli venduti con un importo maggiore di 35000 euro ordinandole in ordine decrescente " << endl;
res = PQexec(conn, "SELECT DISTINCT Vei.modello, Pag.importo FROM Veicolo AS Vei, Pagamento AS Pag WHERE Vei.sn = Pag.sn_veicolo AND Pag.importo >= 35000 ORDER BY(Pag.importo) DESC; ");
checkResults(res, conn);
printQuery(res, PQnfields(res), PQntuples(res));
PQclear(res);
```

Eseguirà una query passata come stringa, stampandone il risultato a schermo.