

VeicHome

Piattaforma di una Concessionaria.

Membri del gruppo

Michele del Mastro 08937 Armando Imbimbo 06867 Giuseppe Sabia 06468

Progetto di Basi di Dati

- Concessionario -

 Raccolta delle specifiche della realtà d'interesse Descrizione

Si vuole progettare una base di dati per la gestione di una piattaforma di una concessionaria.

L'acquisto di un veicolo avviene grazie alla registrazione alla piattaforma.

Di un cliente che si registra alla piattaforma si vuole conoscere: il nome, cognome, sesso, indirizzo, data di nascita, città e i numeri di telefono e il cf che lo identifica.

Inoltre di un cliente si vuole tenere traccia anche della carta della quale si vuole conoscere: numero di carta che la identifica e la scadenza.

Un cliente può acquistare un veicolo. Del veicolo si vuole tenere traccia delle seguenti informazioni: come targa che lo identifica, codice telaio, modello, kW, prezzo, marchio, colore e numero di riparazioni.

Inoltre, un veicolo si divide in auto e moto, delle auto si vuole tener conto dello sconto e del numero massimo di passeggeri, mentre delle moto se è custom, inoltre esse possono disporre di accessori di cui si vuole memorizzare: nome, nome produttore e sono identificate dal codice.

Le auto possono essere anche auto usate che sono caratterizzate da: numero km, anno di immatricolazione, esse hanno avuto dei proprietari di cui si vuole memorizzare nome, cognome e il codice fiscale che li identifica.

I veicoli vengono gestiti da un admin, di cui si vuole memorizzare: nome, cognome e codice fiscale che lo identifica.

Inoltre, un veicolo può essere soggetto a riparazione, la quale è caratterizzata da: data inizio, data fine è identificata da un codice. Una riparazione è effettuata da un'officina di cui si vuole conoscere il nome, il codice che la identifica ed il numero di telefono. La riparazione può coinvolgere dei ricambi di cui ci interessa sapere: nome, costo e un modello, inoltre il ricambio viene identificato dal modello e dalla riparazione.

SPECIFICA DELLA REALTÀ' D'INTERESSE

La realtà che andiamo a rappresentare riguarda la gestione di una piattaforma online (concessionaria online), sulla quale è possibile effettuare un acquisto di un veicolo che si può differenziare in auto e moto.

Con l'avanzare degli anni è diventata indispensabile la vendita online.

Tutto ciò rende difficile la gestione di chi si occupa delle vendite online, ossia nel tenere traccia dei veicoli venduti dalla concessionaria, oltre che ad un'organizzazione dei dati più efficiente.

Oggigiorno, con la nascita di numerosi siti web per la vendita di veicoli, i clienti hanno una vasta scelta di possibilità sulla piattaforma per effettuare l'acquisto.

Nella nostra realtà di interesse che rappresentiamo è necessaria la registrazione mediante l'inserimento di un cliente registrato all'interno del database.

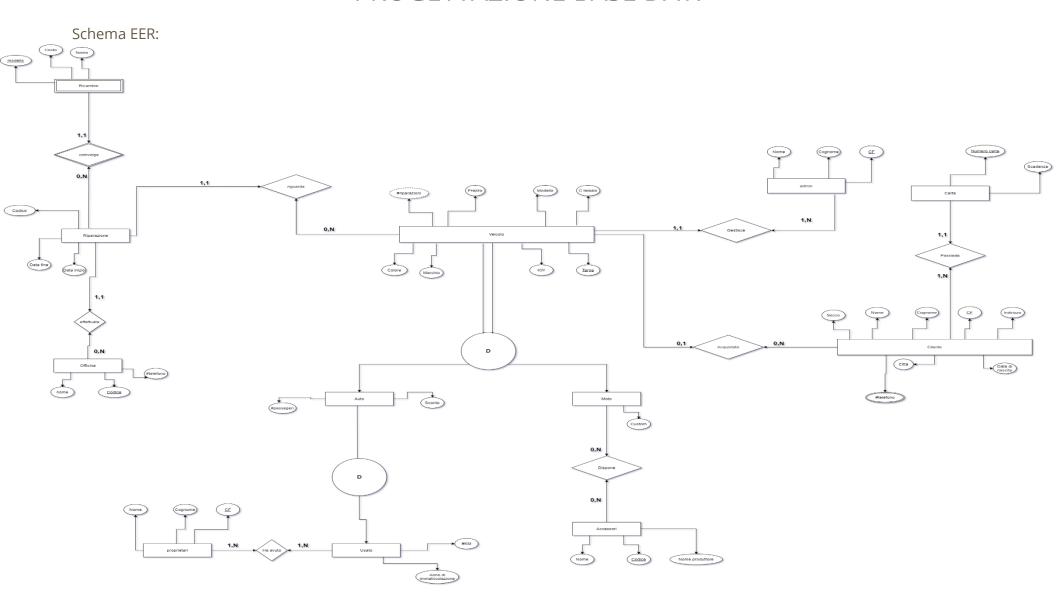
Il cliente ha la possibilità di scegliere e acquistare un veicolo disponibile sul database, oltre che ad inserire i dati della propria carta con la quale ha intenzione di procedere all'acquisto.

L'admin ha la possibilità invece di inserire nuovi veicoli, e di sottoporli a riparazione in caso di necessità.

GLOSSARIO DEI TERMINI

Termine	Significato
Cliente	Identifica un utente registrato alla piattaforma dal quale si ha
	l'accesso.
Carta	Identifica il metodo di pagamento che ha il cliente.
Veicolo	Identifica il veicolo che è presente all'interno della piattaforma.
Auto	Veicolo che può essere acquistato dal cliente e inserito nella
	piattaforma dall'admin.
Moto	Veicolo che può essere acquistato dal cliente e inserito nella
	piattaforma dall'admin.
Usato	Veicolo che può essere acquistato dal cliente e inserito nella
	piattaforma dall'admin.
Admin	Rappresenta il super-user che gestisce i veicoli, riparazioni di essi.
Riparazione	Identifica la riparazione su un veicolo presente nella piattaforma.
Ricambio	Identifica i pezzi di ricambio al susseguirsi di una riparazione.
Proprietario	Identifica un proprietario di una auto usata
Officina	Identifica una officina che effettua riparazioni
Accessori	Identifica gli accessori che sono associati alle moto

PROGETTAZIONE BASE DATI



Dizionario delle entità

Entità	Descrizione	Attributi	Identificatore
Cliente	Identifica un utente registrato alla piattaforma	sesso nome cognome cf indirizzo citta data di nascita #telefono (multivalore)	CF
Carta	Identifica il metodo di pagamento che ha il cliente.	numero di carta scadenza	numero di carta
Veicolo	Identifica il veicolo che è presente all'interno della piattaforma.	prezzo modello codice telaio colore marchio KW targa #riparazioni (attributo ridondante)	targa
Auto	Veicolo che è presente all'interno della piattaforma.	#passeggeri sconto	/
Moto	Veicolo che è presente all'interno della piattaforma.	custom	/
Accessori	Accessori che sono associati alle moto	Nome Nome produttore Codice	Codice
Usato	Veicolo che è presente all'interno della piattaforma.	#Km Anno immatricolazione	/
Admin	Rappresenta il super- user che gestisce i	nome cognome	cf

	veicoli, riparazioni di essi.	cf	
Riparazione	Identifica la riparazione su un veicolo presente nella piattaforma.	codice data inizio data fine	codice
Ricambio	Identifica i pezzi di ricambio al susseguirsi di una riparazione.	modello costo nome	Modello (chiave candidata)
Proprietario	Identifica un proprietario di una auto usata	nome cognome cf	cf
Officina	ldentifica una officina che effettua riparazioni	nome codice #telefono	codice

Dizionario delle relazioni

Relazione	Descrizione	Entità coinvolte	Attributi
Possiede	Un cliente possiede una carta	Cliente (1,N) Carta (1,1)	/
Gestisce	Un admin gestisce i veicoli	Admin (1,N) Veicolo (1,1)	/
Acquistato	Un cliente può acquistare un veicolo	Cliente (0,N) Veicolo (0,1)	/
Dispone	Una moto dispone di accessori	Moto (0,N) Accessori (0,N)	/
Ha avuto	Una auto usata ha avuto dei proprietari	Usato (1,N) Proprietari (1,N)	/
Riguarda	Un veicolo può essere riparato	Veicolo (0,N) Riparazione (1,1)	/
Coinvolge	Una riparazione può coinvolgere dei ricambi	Riparazione (0,N) Ricambio (1,1)	/
Effettuata	Una riparazione viene effettuata da un'officina	Riparazione (1,1) Officina (0,N)	/

DEFINIZIONE DELLE PROCEDURE PER LA GESTIONE DELLA BASE DI DATI

Tavola dei volumi

CONCETTO	TIPO	CARICO APPLICATIVO
Veicolo	Е	20
Auto	Е	10
Moto	E	10
Usato	E	5
Cliente	E	13
Carte	Е	26
Admin	E	2
Ricambio	E	25
Riparazione	E	40
Officina	E	3
Proprietari	E	10
Accessori	E	20
Gestisce	R	20
Possiede	R	26
Acquistato	R	15
Dispone	R	30
Riguarda	R	40
Effettuata	R	40
Coinvolge	R	25
Ha avuto	R	20

CALCOLO CARICO APPLICATIVO

Ha avuto: In media un'auto usata ha 4 proprietari.

4 * 5 (Auto usate) = 20

Dispone: In media una moto dispone di 3 accessori.

3 * 10 (Numero di moto) = 30

TAVOLA DELLE OPERAZIONI

	OPERAZIONE	TIPO	FREQUENZA
1.	Inserisci una nuova riparazione	1	12/mm
2.	Selezionare il numero di riparazioni di un veicolo	I	16/mm
3.	Stampare il numero di pezzi di ricambio per una determinata riparazione	I	2/mm
4.	Inserimento una nuova auto		15/mm
<i>5.</i>	Inserimento una nuova moto	- 1	10/mm
6.	Inserimento di un'auto usata		3/mm
7.	Inserimento di uno nuovo cliente con la relativa carta	I	10/mm
8.	Stampare tutti i veicoli disponibili	В	15/mm
9.	Seleziona la targa dei veicoli riparati che hanno più di due ricambi	В	2/mm
10.	Selezionare i veicoli che hanno una riparazione conclusa dal "20/12/2022" al "27/02/2023"	В	5/mm
11.	Inserire un acquisto da parte del cliente		12/mm
12.	Un dato admin stampa i veicoli di colore rosso, che sono stati acquistati dai clienti di nome Marco.	I	2/mm
<i>13.</i>	Inserimento di un nuovo admin		1/mm
<i>14.</i>	Inserimento di una nuova officina	I	5/mm
15.	Stampare i clienti in ordine crescente in base al nome	В	5/mm
16.	Stampare le auto usate che hanno meno di 5 proprietari	В	4/mm
17.	Stampare le auto che hanno un numero maggiore di un dato numero di passeggeri	I	3/mm
18.	Stampare le auto che hanno uno sconto maggiore di un dato numero	I	5/mm

19.	Selezionare i clienti che hanno acquistato	В	7/mm
	almeno 2 veicoli		
20.	Selezionare le auto usate di un dato marchio	I	1/mm
	che ha almeno 2 proprietari		
21.	Eliminare un numero telefono di un dato	1	1/mm
	cliente		

PROGETTAZIONE LOGICA

ANALISI DELLE RIDONDANZE

Il dato ridondante è l'attributo "#riparazioni" dell'entità Veicolo. Infatti, possiamo ottenere il numero di riparazioni tramite la relazione "Veicolo – Riguarda – Riparazione". Supponendo che l'attributo abbia un peso di 4 byte, essendo un normale intero, e considerando che il volume dell'entità veicolo è pari a 20, il dato andrà ad occupare lo spazio di circa 80 byte. Per decidere se bisogna o meno mantenere il dato ridondante, abbiamo bisogno di calcolare le operazioni che coinvolgono esso, in modo da calcolare la differenza del # di accessi con o senza il dato ridondante.

TAVOLA DEGLI ACCESSI

Operazione 1:

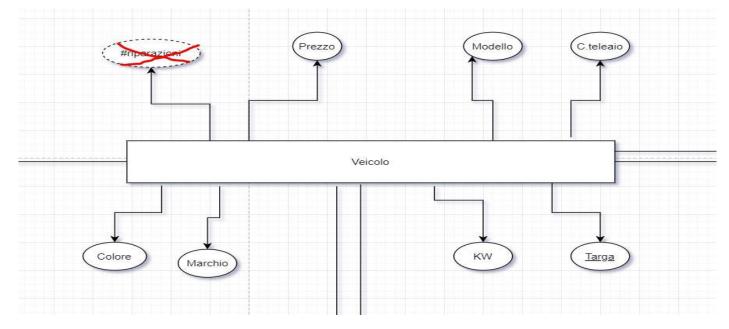
Calcolo con la ridondanza				Calc	olo ser	nza ridond	anza	
	Tabella	Tipo	Accessi	Tipo Accessi	Tabella	Tipo	Accessi	Tipo Accessi
	Riparazione	Е	1	S	Riparazione	Е	1	S
	Riguarda	R	1	S	Riguarda	R	1	S
	Veicolo	Е	1	L	Effettuata	R	1	S
	Veicolo	Е	1	S				
	Effettuata	R	1	S				
Totale	(4 * 2) + 1 = 9) * 12 =	: 108		(3*2) = 6 * 12	2 = 72		

Operazione 2:

Calcolo con la ridondanza				Cald	colo ser	nza ridond	anza	
	Tabella	Tipo	Accessi	Tipo Accessi	Tabella	Tipo	Accessi	Tipo Accessi
	Veicolo	Е	1	L	Veicolo	Е	1	L
					Riguarda	R	2	L
Totale	1*16 = 16				3*16 = 48			

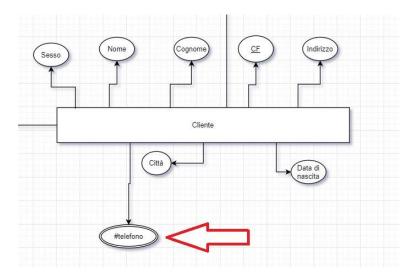
Totale accessi con ridondanza	Totale accessi senza ridondanza
108 + 16 = 124	72 + 48 = 120
80 byte	

Avendo il minor numero di accessi e lo spreco di spazio pari a 80byte, abbiamo deciso di eliminare il dato ridondante "#riparazioni" dell'entità veicolo.

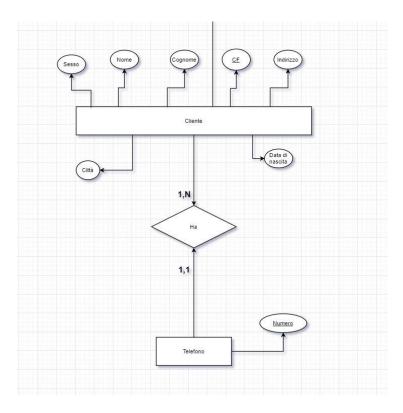


ELIMINAZIONE DELL'ATTRIBUTO MULTIVALORE

Nello schema inizialmente abbiamo l'attributo multivalore:

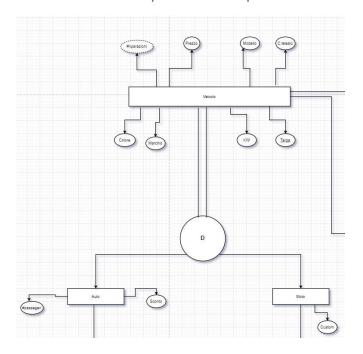


Tale attributo viene rappresentato in maniera differente nella progettazione logica. Abbiamo deciso di creare una nuova entità "Telefono" in relazione con l'entità "Cliente"



ELIMINAZIONE DELLE GERARCHIE

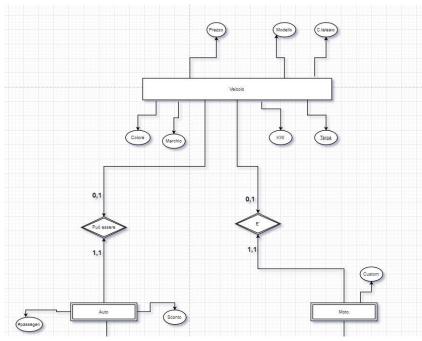
Nello schema inizialmente è presente la specializzazione dell'entità Veicolo:

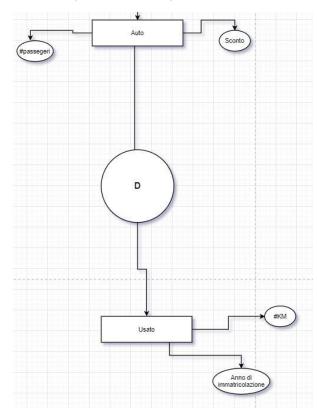


Successivamente in fase di progettazione logica, la nostra scelta è quella di rendere le figlie entità deboli, così da preservare la separazione dei due concetti, così dà

evitare attributi null.

Ristrutturazione:



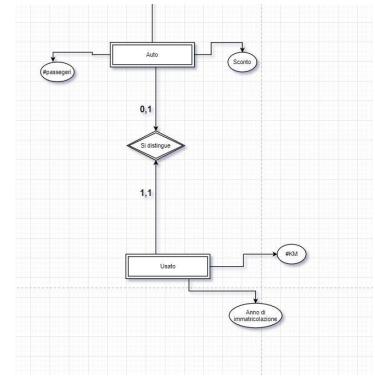


Nello schema inizialmente è presente la specializzazione dell'entità Auto:

Successivamente in fase di progettazione logica, la nostra scelta è quella di rendere le figlie entità deboli, così da preservare la separazione dei concetti, così dà evitare

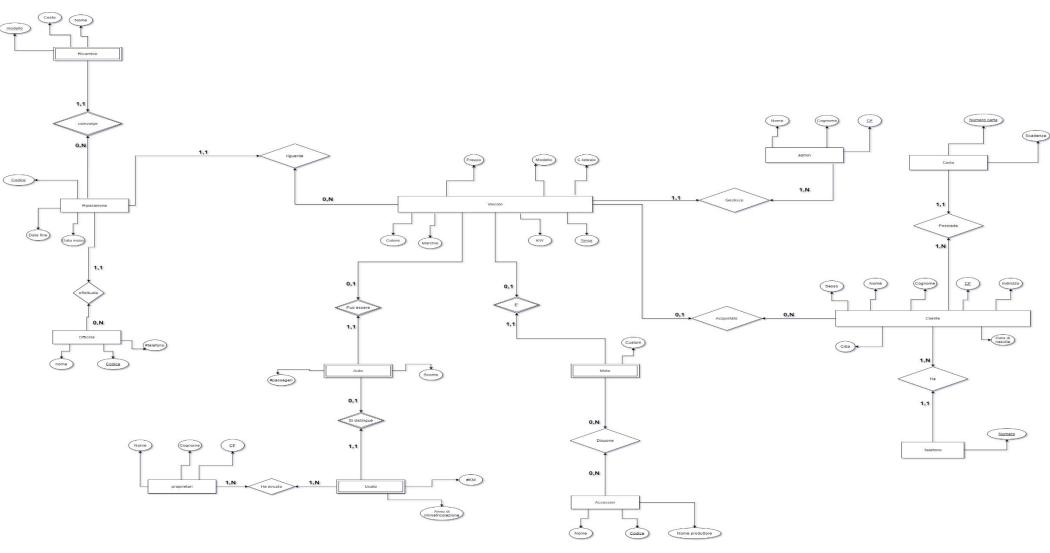
attributi null:

Ristrutturazione:



SCHEMA EER RISTRUTTURATO

Schema EER:



SCHEMA RELAZIONALE

Mapping.

Cliente: (CF, Sesso, Nome, Cognome, Indirizzo, Città, Data di nascita)

Telefono: (Numero, CLIENTE.CF[↑])

Carta: (NUMERO CARTA, Scadenza, CLIENTE.CF[↑])

Veicolo: (TARGA, Prezzo, Modello, Codice telaio, Kw, Colore, Marchio,

ADMIN.CF[↑])

Acquistato: (CLIENTE.CF↑, VECIOLO.TARGA↑)

Admin: (<u>CF</u>, Nome, Cognome)

Moto: (Custom, <u>VEICOLO.TARGA</u>↑)

Auto: (Sconto, #Passegeri, <u>VEICOLO.TARGA</u>↑)

Accessori: (Nome produttore, Nome, <u>CODICE</u>)

Dispone: (MOTO.VEICOLO.TARGA↑, ACCESSORI CODICE↑)

Usato: (#Km, Anno di immatricolazione, <u>AUTO.VECIOLO.TARGA</u>↑)

Ha avuto: (PROPRIETARIO.CF↑, USATO.AUTO.VEICOLO.TARGA↑)

Proprietario: (CF, Nome, Cognome)

Riparazione: (CODICE, Data fine, Data inizio, VEICOLO.TARGA[↑],

OFFICINA.CODICE 1)

Ricambio: (RIPARAZIONE.CODICE↑, MODELLO, Costo, Nome)

Officina: (#Telefono, Nome, CODICE)

NORMALIZZAZIONE

Il database si presenta in parte già normalizzato.

Gli attributi sono atomici dopo la ristrutturazione, visto che è stato eliminato l'attributo numeri di telefono presente nell'entità cliente.

Si presenta in 2NF, oltre che ad essere in 1NF, inoltre quando è presente una chiave primaria composta da più attributi le dipendenze funzionali sono piene e non parziali.

Inoltre sono in 3NF essendo già in 2NF, visto che non sono presente nelle tabelle dipendenze transitive tra attributi non chiave e chiavi primarie.

L'unica tabella che non risulta in 2NF è:

Ricambio

Dipendenza funzionale parziale tra modello, costo e tra modello, nome.

Non è in 2NF:

Ricambio: (MODELLO, Costo, Nome)

Ricambio_info: (RICAMBIO.MODELLO↑, RIPARAZIONE.CODICE↑)

La tabella **VEICOLO** risulta in 3NF perché un modello può avere lo

stesso nome(modello) con marchio diverso.

REALIZZAZIONE DELLA BASE DI DATI CON MySQL

Istruzioni MySQL per la creazione della base di dati:

```
○ CREATE TABLE `concessionario`. `cliente` (
         'CF' CHAR(20) NOT NULL,
3
         'sesso' CHAR(20) NOT NULL,
        'nome' CHAR(20) NOT NULL,
4
5
        'cognome' CHAR(20) NOT NULL,
6
        'indirizzo' CHAR(40) NOT NULL,
        'città' CHAR(20) NOT NULL,
7
8
         'data_di_nascita' DATE NOT NULL,
        PRIMARY KEY ('CF'));
9
10
     ○ CREATE TABLE 'concessionario'. 'telefono' (
1
2
         'Numero' BIGINT(20) NOT NULL,
3
         'cliente_CF' CHAR(20) NOT NULL,
4
          FOREIGN KEY (cliente_CF) references cliente (CF) ON UPDATE cascade
5
          ON DELETE cascade,
6
         PRIMARY KEY ('Numero'));
     ○ CREATE TABLE `concessionario`.`carta` (
 1
 2
          'Numero_carta' BIGINT(20) NOT NULL,
 3
          'scadenza' DATE NOT NULL,
 4
          'carta_cliente_CF' CHAR(20) NOT NULL,
           FOREIGN KEY (carta_cliente_CF) references cliente (CF) ON UPDATE cascade
 5
 6
           ON DELETE cascade,
 7
          PRIMARY KEY ('Numero_carta'));
     ○ CREATE TABLE `concessionario`.`admin` (
 1
 2
         'CF' CHAR(20) NOT NULL,
         'nome' CHAR(20) NOT NULL,
 3
         'cognome' CHAR(20) NOT NULL,
 4
       PRIMARY KEY ('CF'));
 5
 6
```

```
→ CREATE TABLE `concessionario`.`veicolo` (
1
 2
         'Targa' CHAR(10) NOT NULL,
         'prezzo' FLOAT NOT NULL,
 3
 4
         'modello' CHAR(20) NOT NULL,
         'codice_telaio' CHAR(20) NOT NULL,
5
 6
         'kw' INT NOT NULL,
 7
         'colore' CHAR(10) NOT NULL,
         'marchio' CHAR(20) NOT NULL,
8
9
         'admin_CF' CHAR(20) NOT NULL,
       FOREIGN KEY (admin_CF) references admin (CF) ON UPDATE cascade
10
          ON DELETE cascade,
11
12
      PRIMARY KEY ('Targa'));
13
    ○ CREATE TABLE 'concessionario', 'acquistato' (
1
2
         'acquistato_cliente_CF' CHAR(20) NOT NULL,
3
         'acquistato_veicolo_TARGA' CHAR(20) NOT NULL,
       FOREIGN KEY (acquistato_cliente_CF) references cliente (CF) ON UPDATE cascade
4
5
          ON DELETE cascade,
6
       FOREIGN KEY (acquistato_veicolo_TARGA) references veicolo (Targa) ON UPDATE cascade
7
          ON DELETE cascade,
        PRIMARY KEY ('acquistato_cliente_CF', 'acquistato_veicolo_TARGA'));
8
9
     ○ CREATE TABLE `concessionario`.`moto` (
 2
         'custom' CHAR(20) NOT NULL,
 3
         'moto_veicolo_TARGA' CHAR(10) NOT NULL,
        FOREIGN KEY (moto_veicolo_TARGA) references veicolo (Targa) ON UPDATE cascade
 4
           ON DELETE cascade,
 5
        PRIMARY KEY ('moto_veicolo_TARGA'));
 6
 7

		○ CREATE TABLE `concessionario`.`auto` (

 2
         'sconto' INT(5) NOT NULL,
 3
         'numero_passegeri' INT(5) NOT NULL,
         'auto_veicolo_TARGA' VARCHAR(45) NOT NULL,
 5
        FOREIGN KEY (auto_veicolo_TARGA) references veicolo (Targa) ON UPDATE cascade
          ON DELETE cascade,
          PRIMARY KEY ('auto_veicolo_TARGA'));
 7
```

```
○ CREATE TABLE `concessionario`.`accessori` (
 2
          'Codice' INT(20) NOT NULL,
 3
          'nome' CHAR(10) NOT NULL,
 4
          'nome_produttore' CHAR(20) NOT NULL,
 5
          PRIMARY KEY ('Codice'));
 6
    ○ CREATE TABLE 'concessionario'. 'dispone' (
2
        'dispone_moto_veicolo_TARGA' CHAR(10) NOT NULL,
        'dispone_accessori_CODICE' INT(20) NOT NULL,
3
4
       FOREIGN KEY (dispone_moto_veicolo_TARGA) references moto (moto_veicolo_TARGA) ON UPDATE cascade
         ON DELETE cascade,
5
6
      FOREIGN KEY (dispone_accessori_CODICE) references accessori (Codice) ON UPDATE cascade
7
         ON DELETE cascade,
8
        PRIMARY KEY ('dispone_moto_veicolo_TARGA', 'dispone_accessori_CODICE'));
9
    ○ CREATE TABLE 'concessionario'.'usato' (
1
2
         'numero_km' INT NOT NULL,
3
        'anno_di_immatricolazione' DATE NOT NULL,
4
        'usato_auto_veicolo_TARGA' VARCHAR(45) NOT NULL,
       FOREIGN KEY (usato_auto_veicolo_TARGA) references auto (auto_veicolo_TARGA) ON UPDATE cascade
5
6
          ON DELETE cascade,
7
        PRIMARY KEY ('usato_auto_veicolo_TARGA'));
8
    ○ CREATE TABLE 'concessionario'. 'proprietario' (
2
         'CF' CHAR(20) NOT NULL,
         'nome' CHAR(20) NOT NULL,
3
4
         'cognome' CHAR(20) NOT NULL,
        PRIMARY KEY ('CF'));
5
6
   O CREATE TABLE 'concessionario'.'ha_avuto' (
1
       'proprietario_CF' CHAR(20) NOT NULL,
       'usato_auto_veicolo_TARGA' VARCHAR(45) NOT NULL,
3
4
      FOREIGN KEY (usato_auto_veicolo_TARGA) references usato (usato_auto_veicolo_TARGA) ON UPDATE cascade
5
        ON DELETE cascade,
       PRIMARY KEY ('proprietario_CF', 'usato_auto_veicolo_TARGA'));
6
```

```
○ CREATE TABLE 'concessionario'.'riparazione' (
2
         'Codice' INT NOT NULL,
         'data_fine' DATE NOT NULL,
3
         'data_inizio' DATE NOT NULL,
4
5
         'veicolo_TARGA' VARCHAR(45) NOT NULL,
6
         'officina CODICE' INT NOT NULL,
7
        FOREIGN KEY (veicolo_TARGA) references veicolo (Targa) ON UPDATE cascade
          ON DELETE cascade,
8
9
       FOREIGN KEY (officina_CODICE) references officna (Codice) ON UPDATE cascade
          ON DELETE cascade,
10
       PRIMARY KEY ('Codice'));
11
12
    ○ CREATE TABLE `concessionario`.`ricambio` (
         'Modello' CHAR(30) NOT NULL,
2
         'costo' FLOAT NOT NULL,
3
4
         'nome' CHAR(20) NOT NULL,
        PRIMARY KEY ('Modello'));
5
6
     ○ CREATE TABLE `concessionario`.`ricambio_info` (
1
2
         'ricambio_MODELLO' CHAR(30) NOT NULL,
3
         'riparazione_CODICE' INT NOT NULL,
        FOREIGN KEY (ricambio_MODELLO) references ricambio (Modello) ON UPDATE cascade
4
5
          ON DELETE cascade,
6
        FOREIGN KEY (riparazione_CODICE) references riparazione (Codice) ON UPDATE cascade
7
          ON DELETE cascade,
         PRIMARY KEY ('ricambio_MODELLO', 'riparazione_CODICE'));
8
 1 

CREATE TABLE `concessionario`.` officina` (
 2
       'numero_telefono' BIGINT(20) NOT NULL,
      'nome' CHAR(20) NOT NULL,
 3
    'Codice' INT NOT NULL,
      PRIMARY KEY ('Codice'));
```

IMPLEMENTAZIONE QUERY SQL

OPERAZIONE 1 - Inserisci una nuova riparazione -

INSERT INTO riparazione(Codice, data_inizio, data_fine, veicolo_TARGA, officina_CODICE)

VALUES(?,?,?,?,?);

OPERAZIONE 2 - Selezionare il numero di riparazioni di un veicolo -

SELECT Count(r.Codice)

FROM riparazione as r,veicolo as v

WHERE r.veicolo_Targa=v.Targa and v.Targa=?

OPERAZIONE 3 - Stampare il numero di pezzi di ricambio per una determinata riparazione -

SELECT Count(i.riparazione_CODICE)

FROM riparazione as r,ricambio_info as i

WHERE r.Codice=i.riparazione_CODICE and r.Codice=?

OPERAZIONE 4 - L'inserimento di un'auto prevede anche l'inserimento di un veicolo

INSERT INTO veicolo(Targa, prezzo, modello, codice_telaio, kw, colore, marchio, admin_CF)

VALUES(?,?,?,?,?,?,?);

INSERT INTO auto(sconto, numero_passeggeri, auto_veicolo)

VALUES(?,?,?);

OPERAZIONE 5 - L'inserimento di una moto prevede anche l'inserimento di un veicolo -

INSERT INTO veicolo(Targa, prezzo, modello, codice_telaio, kw, colore, marchio, admin_CF)

VALUES(?,?,?,?,?,?,?);

INSERT INTO moto(custom, moto veicolo)

VALUES(?,?);

OPERAZIONE 6 - L'inserimento di un'auto usata prevede l'inserimento di un'auto e di un veicolo. -

INSERT INTO veicolo(Targa, prezzo, modello, codice_telaio, kw, colore, marchio, admin_CF)

VALUES(?,?,?,?,?,?,?);

INSERT INTO auto(sconto, numero_passeggeri, auto_veicolo)

VALUES(?,?,?);

INSERT INTO usato(numero_km, anno_di_immatricolazione, usato_auto_veicolo_TARGA)

VALUES(?,?,?);

```
OPERAZIONE 7 - Inserimento di uno nuovo cliente con la relativa carta -
```

INSERT INTO cliente(CF, sesso, nome, cognome, indirizzo, città, data_di_nascita)

VALUES(?,?,?,?,?,?,?);

INSERT INTO carta(Numero_carta, scadenza, carta_cliente_CF)

VALUES(?,?,?);

INSERT INTO telefono(Numero, cliente CF)

VALUES(?,?);

OPERAZIONE 8 - Stampare tutti i veicoli disponibili -

SELECT v.Targa, v.prezzo, v.modello, v.codice_telaio, v.kw, v.colore, v.marchio, v.admin_CF

FROM veicolo as v

WHERE (v.Targa) NOT IN(SELECT v1.Targa

FROM veicolo as v1, acquistato as a

WHERE v1.Targa = a.acquistato_veicolo_TARGA)

GROUP BY(v.Targa);

OPERAZIONE 9 - Seleziona la targa dei veicoli riparati che hanno più di due ricambi -

SELECT r.veicolo_TARGA

FROM riparazione as r, ricambio_info as ri

WHERE r.Codice = ri.riparazione CODICE

GROUP BY r.veicolo_TARGA

HAVING COUNT(ri.riparazione_CODICE)> 2;

OPERAZIONE 10 - Selezionare i veicoli che hanno una riparazione conclusa dal "20/12/2022" al "27/02/2023" -

SELECT v.Targa,v.modello,v.marchio,v.colore,v.codice_telaio

FROM veicolo as v,riparazione as r

WHERE v.Targa=r.veicolo_TARGA and (r.data_fine BETWEEN '2022-12-20' and '2023-02-27');

OPERAZIONE 11- Inserire un acquisto da parte del cliente -

INSERT INTO acquistato(acquistato_cliente_CF, acquistato_veicolo_TARGA)
VALUES(?,?);

OPERAZIONE 12 - Un dato admin stampa i veicoli di colore rosso, che sono stati acquistati dai clienti di nome Marco -

SELECT v.Targa,v.modello,v.marchio,v.colore,v.codice_telaio

FROM veicolo as v, admin as a,acquistato as ac,cliente as c

WHERE a.CF="BXTTQV56D24A896H" and a.CF=v.admin_CF and v.colore='rosso' and ac.acquistato_cliente_CF=c.CF and c.nome='Marco'

AND ac.acquistato_veicolo_TARGA = v.Targa

OPERAZIONE 13 - L'inserimento di un admin prevede anche l'inserimento di un veicolo, in questo caso si è scelto di inserire un auto -

INSERT INTO admin(CF, nome, cognome)

VALUES(?,?,?);

INSERT INTO veicolo(Targa, prezzo, modello, codice_telaio, kw, colore, marchio, admin_CF)

VALUES(?,?,?,?,?,?,?);

INSERT INTO auto(sconto, numero_passeggeri, auto_veicolo)

VALUES(?,?,?);

OPERAZIONE 14 - Inserimento di una nuova officina -

INSERT INTO officina(numero_telefono, nome, Codice)

VALUES(?,?,?);

OPERAZIONE 15 - Stampare i clienti in ordine crescente in base al nome -

SELECT c.CF, c.sesso, c.nome, c.cognome, c.indirizzo, c.città, c.data_di_nascita

FROM cliente as c

ORDER BY c.nome ASC;

OPERAZIONE 16 - Stampare le auto usate che hanno meno di 5 proprietari -

SELECT v.Targa, v.prezzo, v.modello, v.codice_telaio, v.kw, v.colore, v.marchio, v.admin_CF

FROM veicolo as v, auto as a, usato as u, ha_avuto as h

WHERE a.auto_veicolo_TARGA = u.usato_auto_veicolo_TARGA AND v.Targa = a.auto_veicolo_TARGA AND u.usato_auto_veicolo_TARGA = h.usato_auto_veicolo_TARGA

GROUP BY v.Targa

HAVING COUNT(h.usato auto veicolo TARGA) < 5;

OPERAZIONE 17 - Stampare le auto che hanno un numero maggiore di un dato numero di passeggeri -

SELECT v.Targa, v.prezzo, v.modello, v.codice_telaio, v.kw, v.colore, v.marchio, v.admin_CF

FROM veicolo as v, auto as a

WHERE a.numero_passegeri > ? AND v.Targa = a.auto_veicolo_TARGA;

OPERAZIONE 18 - Stampare le auto che hanno uno sconto maggiore di un dato numero -

SELECT v.Targa, v.prezzo, v.modello, v.codice_telaio, v.kw, v.colore, v.marchio, v.admin CF

FROM veicolo as v, auto as a

WHERE a.sconto > ? AND v.Targa = a.auto_veicolo_TARGA;

OPERAZIONE 19 - Selezionare i clienti che hanno acquistato almeno 2 veicoli -

SELECT c.CF, c.nome, c.cognome, c.sesso, c.indirizzo, c.città, c.data_di_nascita

FROM cliente as c, acquistato as a

WHERE c.CF = a.acquistato_cliente_CF

GROUP BY c.CF

HAVING COUNT(a.acquistato cliente CF) >= 2;

OPERAZIONE 20 - Selezionare le auto usate di un dato marchio che ha almeno 2 proprietari -

SELECT v.Targa, v.prezzo, v.modello, v.codice_telaio, v.kw, v.colore, v.marchio, v.admin_CF

FROM veicolo as v, auto as a, usato as u, ha_avuto as h

WHERE a.auto_veicolo_TARGA = u.usato_auto_veicolo_TARGA AND v.Targa = a.auto_veicolo_TARGA

AND u.usato_auto_veicolo_TARGA = h.usato_auto_veicolo_TARGA AND a.auto_veicolo_TARGA = v.Targa

AND v.marchio = ?

GROUP BY v.Targa

HAVING COUNT(h.usato auto veicolo TARGA) >= 2;

OPERAZIONE 21- Eliminare un numero telefono di un dato cliente -

DELETE

FROM telefono AS t

WHERE t.Numero = ? AND t.cliente_CF = ?

TEST APLICAZIONE JAVA

All'avvio dell'applicazione viene mostrato un menù con le varie operazioni che possono essere eseguite.

```
Connected to the database!
******** SCEGLI OPERAZIONE *******
OPERAZIONE 1
OPERAZIONE 2
OPERAZIONE 3
OPERAZIONE 4
OPERAZIONE 5
OPERAZIONE 6
OPERAZIONE 7
OPERAZIONE 9
OPERAZIONE 10
OPERAZIONE 11
OPERAZIONE 12
OPERAZIONE 13
OPERAZIONE 14
OPERAZIONE 15
OPERAZIONE 16
OPERAZIONE 17
OPERAZIONE 18
OPERAZIONE 19
OPERAZIONE 20
INSERISCI SCELTA:
```

Le operazioni di inserimento mostreranno dei campi da compilare da parte dell utente.

L'inserimento andrà a buon fine e verrà visualizzato il messaggio "OEPRAZIONE EFFETTUATA".

```
INSERISCI SCELTA:
4
INSERISCI TARGA:
fj725ds
INSERISCI Prezzo:
12000
INSERISCI MODELLO:
m5
INSERISCI CODICE TELAIO:
7207308h840cs
INSERISCI KW:
90
INSERISCI COLORE:
bianco
INSERISCI COLORE:
bianco
INSERISCI MARCHIO:
bmw
INSERISCI CODICE FISCALE ADMIN:
BXTTQV56D24A896H
INSERISCI SCONTO:
3
INSERISCI NUMERO PASSEGERI:
4
DPERAZIONE EFFETTUATA
```

Di seguito verranno riportati alcuni screen che mostrano il risultato di alcunne query.

SCREEN OPERAZIONE 16

INSERISCI SCELTA:

TARGA= d345fh PREZZO= 13500.0 MODELLO= tipo CODICE TELAIO= 789088h780fg KW= 59 COLORE= giallo MARCHIO= fiat CODICE FISCALE ADMIN= BXTTQV56D24A896H

TARGA= ff321hj PREZZO= 49500.0 MODELLO= stelvio CODICE TELAIO= 7987878h780vb KW= 89 COLORE= rosso MARCHIO= alfa_romeo CODICE FISCALE ADMIN= BXTTQV56D24A896H

TARGA= gh678cx PREZZO= 11500.0 MODELLO= multipla CODICE TELAIO= 7456788h780yu KW= 59 COLORE= rosso MARCHIO= fiat CODICE FISCALE ADMIN= BXTTQV56D24A896H

TARGA= jd791ds PREZZO= 25500.0 MODELLO= m3 CODICE TELAIO= 4560788h780df KW= 70 COLORE= verde MARCHIO= bmw CODICE FISCALE ADMIN= BXTTQV56D24A896H

TARGA= tr234sd PREZZO= 14500.0 MODELLO= cinquecento CODICE TELAIO= 8987788h780gh KW= 49 COLORE= bianco MARCHIO= fiat CODICE FISCALE ADMIN= BXTTQV56D24A896H

SCREEN OPERAZIONE 17

INSERISCI SCELTA:

INSERISCI NUMERO PASSEGERI:

4 | TARGA= d345fh PREZZO= 13500.0 MODELLO= tipo CODICE TELAIO= 789088h780fg KW= 59 COLORE= giallo MARCHIO= fiat CODICE FISCALE ADMIN= BXTTQV56D24A896H

TARGA= ds123fs PREZZO= 38500.0 MODELLO= x5 CODICE TELAIO= 7898078h780jj KW= 78 COLORE= giallo MARCHIO= bmw CODICE FISCALE ADMIN= BXTTQV56D24A896H

TARGA= sa802hk PREZZO= 22500.0 MODELLO= c3_picasso CODICE TELAIO= 7887888h780ft KW= 59 COLORE= nero MARCHIO= citroen CODICE FISCALE ADMIN= BXTTQV56D24A896H

SCREEN OPERAZIONE 18

INSERISCI SCELTA:

1Ω

INSERISCI SCONTO:

TARGA= ds123fs PREZZO= 38500.0 MODELLO= x5 CODICE TELAIO= 7898078h780jj KW= 78 COLORE= giallo MARCHIO= bmw CODICE FISCALE ADMIN= BXTTQV56D24A896H

TARGA= jd791ds PREZZO= 25500.0 MODELLO= m3 CODICE TELAIO= 4560788h780df KW= 70 COLORE= verde MARCHIO= bmw CODICE FISCALE ADMIN= BXTTQV56D24A896H