

BikeStart

Abstract

L'azienda con il nome BikeStart vuole aggiornare il DB, perché aveva tanto successo negli ultimi anni e un'espansione era necessaria.

La compagnia si occupa di noleggi di biciclette per un trasporto più facile ed ecologico. Adesso con la nuova espansione sta aggiungendo anche biciclette elettriche e scooter. Il processo per usare i mezzi di trasporto è molto veloce e facile e con il fatto che gli studenti e giovani hanno uno sconto aggiuntivo per ogni uso, BikeStart spera per un futuro più GREEN e pulito.

L'azienda rende facile anche il processo per lavorare con loro dove ogni lavoratore ha una quota per completare per un mese e altri bonus che dipendono se fanno più che era richiesto.

Analisi dei requisiti

1-Descrizione Testuale

L'azienda si comporta con il noleggio di biciclette normale, scooter e biciclette elettriche, tutte e tre hanno una targa unica dove quelle ultime hanno anche la percentuale della batteria. La nuova data base deve essere in grado di memorizzare utenti normali e premium, i suoi dati come e-mail, nome e cognome, data di nascita e un numero di matricola. Ogni utente normale ha una rata fissa per ogni noleggio invece gli utenti premium hanno un'iscrizione da cui quella ha una rata mensile, data di iscrizione e data di scadenza dell'iscrizione. È necessario salvare anche le modalità di pagamento che un utente può usare come carta di credito, visa, PayPal, Google Pay, o Apple Pay. Un utente può avere fino a cinque metodi per pagare.

La data base deve salvare anche i impiegati e sui dati così composti: un numero di matricola, e-mail, numero di telefono, numero della patente se sono in possesso di una, nome, cognome, data di nascita e stipendio mensile. Il database deve avere un modo per memorizzare i noleggi passati, con la data ed ora di sblocco dell'oggetto consegnato e la posizione geografica di consegna, con attività, con la data e ora di noleggio. Tutti i due noleggi, attivi e passati, devono avere anche la posizione degli oggetti quando erano sbloccati. Dobbiamo memorizzare anche i dati di manutenzione degli oggetti, la data e costo, i caricamenti degli scooter e le bici elettriche, anche per questa serve salvare la data e la ricompensa; questi servono per sapere chi prende bonus per la quota passata.

Nel database sono salvate anche le sedi dell'azienda, che si individuano per mezzo della città, la spesa per l'anno, e il reddito dell'anno passato se non è nuovo, dove gli impiegati e i direttori si trovano. Ogni sede ha le sue proprietà nel possesso che includono oggetti di noleggio, uffici e i veicoli come furgoni che hanno le sue targhe, date di revisione e nomi del tipo etc. Inoltre, ogni sede ha le sue filiali con il suo indirizzo, piano e il prezzo d'affitto dell'ufficio.

2-Glossario dei termini

Termine	Descrizione	Collegamenti
Oggetto di Noleggio	Sono i vari modi di trasporti che il cliente può scegliere	Impiegato, Noleggio Passato, Noleggio Presente. Entità figlia di Proprietà in Possesso
Bicicletta Normale	Una delle varianti di mezzi di trasporto	Entità figlia di Oggetto di Noleggio
Bicicletta Elettrica	Sono come le biciclette normale ma hanno l'opzione di pedal assist	Entità figlia di Oggetto di Noleggio
Scooter	Una delle varianti elettriche di mezzi di trasporto	Entità figlia di Oggetto di Noleggio
Cliente	Utente che ha creato un account	Metodo di Pagamento, Noleggio Passato, Noleggio Presente
C. Premium	I clienti che hanno deciso di iscriversi all'abbonamento mensile	Entità figlia di Cliente
C. Normale	I clienti già registrati o quelli che non hanno fatto un abbonamento	Entità figlia di Cliente
Metodo di Pagamento	Un modo per memorizzare i diversi modi per pagare	Cliente
Carta	Uno tra i metodi di pagamento, includono Visa, MasterCard, carta prepagata, e American Express	Entità figlia di Metodo di Pagamento
Account	Un tra i metodi di pagamento, includono PayPal, Google Pay ed Apple Pay	Entità figlia di Metodo di Pagamento
Noleggio Passato	Una cronologia per i noleggi che un utente ha fatto nel passato	Oggetto di Noleggio, Cliente
Noleggio Presente	Un modo per visualizzare i mezzi di trasporto che sono in uso	Oggetto di Noleggio, Cliente
Impiegato	Una persona che lavora per	Oggetto di noleggio, Sede

Termine	Descrizione	Collegamenti
	l'azienda	
Sede	Rappresenta una città dove l'azienda opera	Impiegato, Proprietà in Possesso
Proprietà in Possesso	I beni che l'azienda ha in possesso in una sede	Sede
Ufficio	Il posto dove vengono controllate i oggetti di noleggio per una sede o una parte di una sede	Entità figlia di Proprietà in Possesso
Veicoli	Un modo per trasportare le biciclette e scooter per fare le manutenzioni o caricamento delle batterie	Entità figlia di Proprietà in Possesso
Manutenzione	Un modo per memorizzare le manutenzioni che verranno fatte per gli oggetti di noleggio	Impiegato, Oggetto di Noleggio
Caricamento	Un modo per memorizzare i caricamenti fatti per le biciclette elettriche e scooter	Impiegato, Oggetto di Noleggio

3-Operazioni Tipiche

Operazione	Tipo	Frequenza
Registrazione di un nuovo cliente	S	200 per Mese
Pagamento Manuale	S	50 per Mese
Pagamento Veloce	L	2000 per Mese
Stampa delle bici o scooter più usata	L	30 per Mese
Stampa delle bici o scooter in uso	L	10 per Mese
Stampa la percentuale delle batterie	L	60 per Mese
Registrazione di una nuova bici o scooter	S	50 per Anno
Registrazione di un nuovo Impiegato	S	100 per Anno

Le manutenzioni più costose	L	30 per Mese
I impiegati che ha caricato più batterie	L	30 per Mese
I uffici con l'affitto più alto	L	12 per Anno
I veicoli con la revisione più vicina	L	12 per Anno

Progettazione Concettuale

1-Lista entità

➤ Cliente:

- Matricola: Int Primary Key
- Nome: VarChar(45)
- Cognome: VarChar(45)
- Data di Nascita: Date
- Email: VarChar(80)

L'entità cliente si specializza in due sottocategorie con una generalizzazione totale:

➤ Premium:

- Iscrizione: Attributo composta
 - Rata Mensile: Decimal(5,2)
 - Data Iscrizione: Date
 - Data Fine Iscrizione: Date

➤ Base:

- Rata: Decimal(4,2)

➤ Metodo di Pagamento:

- ID: Int Primary Key

L'entità metodo di pagamento si specializza in due sottocategorie con una generalizzazione totale:

➤ Carta:

- Numero: Int(16)
- Nome Cognome: VarChar(100)
- Nr. di Sicurezza: Int(3)
- Data di Scadenza: Date

➤ Account:

- Email: VarChar(100)

➤ Noleggio Passato:

- ID: Int Primary Key
- Commento: VarChar(500)
- Data Ora Presso: Datetime
- Posizione Presa: VarChar(100)
- Data Ora Consegnata: Datetime
- Posizione Consegnata: VarChar(100)

➤ Noleggio Presente:

- ID: Int Primary Key

- Data Ora Presso: Datetime
- Posizione Presa: VarChar(100)
- **Oggetto di Noleggio:**
 - Targa: Char Primary Key

L'entità oggetto di noleggio si specializza in tre sottocategorie con una generalizzazione totale:

 - **Bicicletta Elettrica:**
 - Batteria: Int(3)
 - **Scooter:**
 - Batteria: Int(3)
 - **Bicicletta Normale**
- **Impiegato:**
 - Matricola: Int Primary Key
 - Email: VarChar(100)
 - Nome: VarChar(45)
 - Cognome: VarChar(45)
 - Stipendio: Decimal(8,2)
 - Numero di Telefono: Int
 - Numero di Patenta: Char può essere NULL
- **Sede:**
 - Città: VarChar(30) Primary Key
 - Reddito Anno Passato: Decimal(8,2) può essere NULL
 - Spesa Annuale: Decimal(8,2)
- **Proprietà in Possesso:**
 - Prezzo Iniziale: Decimal(8,2)
 - Data di Acquisizione : Date

L'entità proprietà in possesso si specializza in tre sottocategorie con una generalizzazione totale:

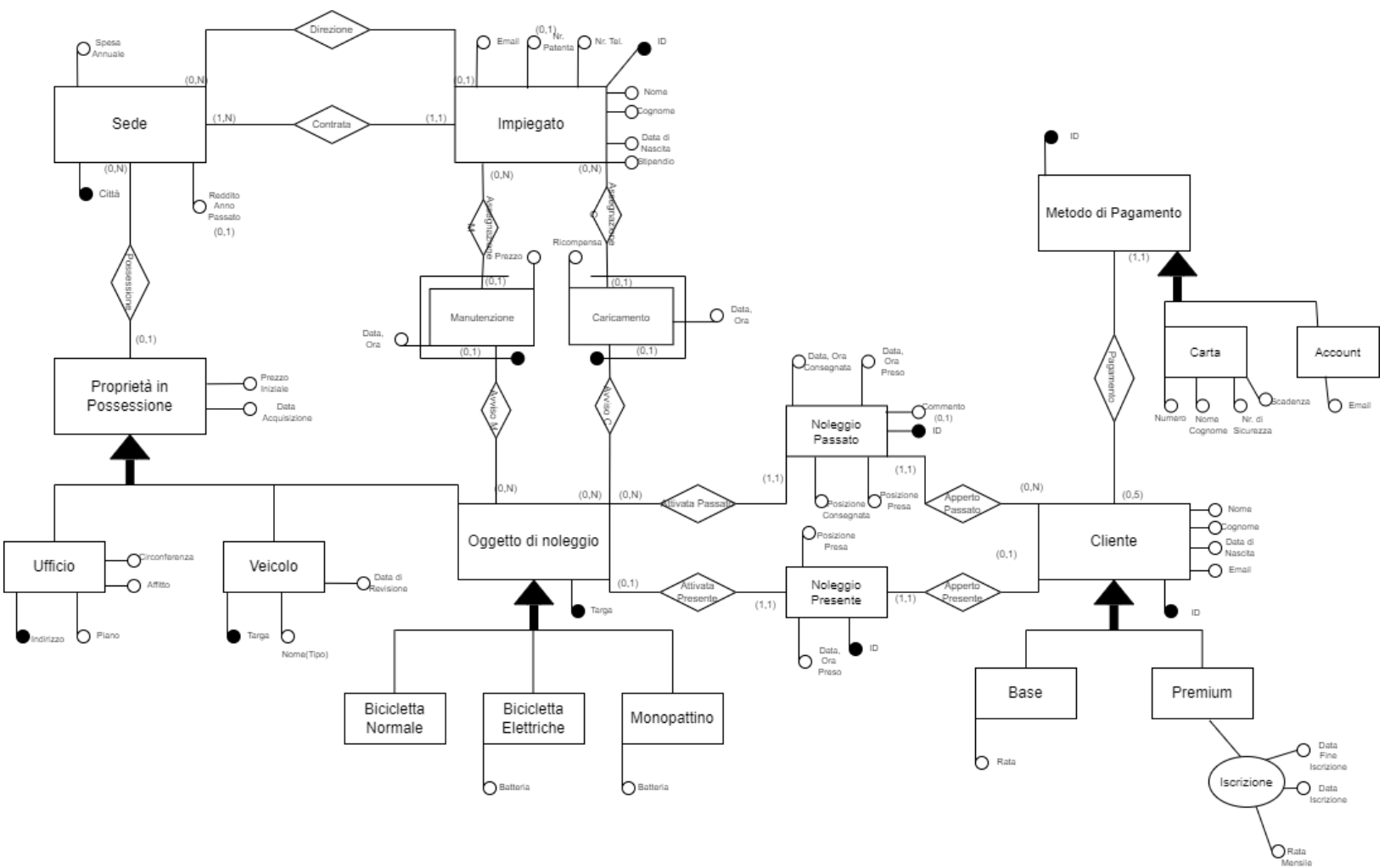
 - **Ufficio:**
 - Indirizzo: VarChar(40) Primary Key
 - Piano: Int
 - Circonferenza: Decimal(6,3)
 - Affitto: Decimal(7,2)
 - **Veicolo:**
 - Targa: Char(8) Primary Key
 - Nome(Tipo): VarChar(50)
 - Data di Revisione: Date
 - **Oggetto di Noleggio**
- **Manutenzione:**
 - Matricola: Int Primary Key
 - Data ora: Datetime Primary Key
 - Targa: Char(8) Primary Key
 - Prezzo: Decimal(5,2)
- **Caricamento:**
 - Matricola: Int Primary Key
 - Data ora: Datetime Primary Key
 - Targa: Char(8) Primary Key
 - Ricompensa: Decimal(5,2)

2-Tabella delle Relazioni

Relazione	Entità coinvolte	Descrizione	Attributi
Pagamento	Metodo di Pagamento(1,1) Cliente(0,5)	Un metodo di pagamento deve essere collegato con un cliente. Un cliente può avere fino a cinque modi diversi di pagamento	N/A
Aperto Passato	Cliente(0,N) Noleggio Passato(1,1)	Un cliente può avere tanti noleggi nel passato. Un noleggio nel passato deve essere di un cliente	N/A
Aperto Attivo	Cliente(0,1) Noleggio Presente(1,1)	Un cliente può sbloccare solo 1 bici o scooter alla volta. Un noleggio presente deve essere di un cliente	N/A
Attivata Passato	Oggetto di Noleggio(0,N) Noleggio Passato(1,1)	Una bici o scooter poteva essere usato tante volte. Un noleggio passato deve avere la sua bici o scooter corrispondente	N/A
Attivata Presente	Oggetto di Noleggio(0,1) Noleggio Presente(1,1)	Una bici o scooter può essere in uso o no. Un noleggio presente deve avere la sua bici o scooter corrispondente	N/A
Contrata	Impiegato(1,1) Sede(1,N)	Un impiegato deve essere impiegato solo a una sede. Una sede può avere più di un impiegato	<u>N/A</u>
Assegnazione M	Manutenzione(1,1) Impiegato(0,N)	Un Impiegato può avere fatto tanti aggiustamenti. Una manutenzione memorizzata deve essere fatta da solo un impiegato	N/A
Assegnazione C	Caricamento(1,1) Impiegato(0,N)	Un Impiegato può avere caricato tante batterie. Un caricamento memorizzata deve essere fatto da solo un impiegato	N/A
Avviso M	Manutenzione(1,1) Oggetto di Noleggio(0,N)	Un oggetto può essere aggiustato tante volte. Una manutenzione memorizzata deve essere associata con solo un oggetto di noleggio	N/A
Avviso C	Caricamento(1,1) Oggetto di Noleggio(0,N)	Una bicicletta elettrica o scooter può essere caricata tante volte. Un caricamento memorizzato deve essere associato con solo un oggetto	N/A

Relazione	Entità coinvolte	Descrizione	Attributi
		noleggiato.	
Direzione	Impiegato(0,1) Sede(0,N)	Un impiegato può essere un direttore o no. Una sede può avere tanti direttori.	N/A
Possessione	Sede(0,N) Proprietà in possesso(0,1)	Una sede può avere tanti beni in possesso. Un bene se non ha ancora una sede selezionata, deve essere in uso da una sola.	N/A

Schema Concettuale



Progettazione Logica

1- Analisi delle ridondanze

Una ridondanza è l'attributo Spesa Annuale che l'entità Sede ha. Questo attributo occupa spazio nel DB e la spesa annuale di una sede può essere trovata sommando l'affitto di uffici per un anno in una sede, lo stipendio degli impiegati di una sede e se esistono le ricompense e per i caricamenti e prezzo per le manutenzioni. Calcoliamo la ridondanza con dati e operazioni fatte in un anno.

Concetto	Tipo	Volume
Sede	E	20
Impiegato	E	2000
Contrata	R	2000
Assegnazione M	R	1000
Assegnazione C	R	10000
Manutenzione	E	1000
Caricamento	E	10000
Possessione	R	20
Ufficio	E	20

Con la ridondanza

Concetto	Costrutto	Accessi	Tipo
Sede	E	1	L

Senza la ridondanza

Concetto	Costrutto	Accessi	Tipo
Sede	E	1	L
Impiegato	E	1	L
Manutenzione	E	1	L
Caricamento	E	1	L

Ufficio	E	1	L
---------	---	---	---

Troviamo il costo solo per leggere la spesa annuale una volta per anno con l'attributo ridondante e poi senza l'attributo, per una sede.

Se abbiamo la ridondanza il costo sarà 1 lettura per anno. Mentre se vogliamo calcolare la spesa senza la ridondanza servirà un po' di più. Con la tabella ci saranno 5 letture, che non sembra tanto ma se vogliamo sapere anche la spesa delle altre sedi il costo sarà moltiplicato per 20.

2- Eliminazione delle generalizzazioni

➤ Metodo di Pagamento

Accorpriamo i figli dell'entità dentro l'entità genitore, e poi mettiamo gli attributi dei figli tutti, con cardinalità (0,1), come attributi di Metodo di Pagamento. Poi aggiungiamo un altro attributo all'entità che descrive qual è il tipo di pagamento.

➤ Clienti

Anche per questa entità si è scelto di accorpare i suoi figli dentro il genitore. L'attributo dei clienti base, che è la rata per ogni noleggio, viene messo come attributo dei clienti con cardinalità (0,1). Invece quelli dei clienti premium ottengono la cardinalità (0,1) tranne la rata mensile che è uguale per tutti; i premium lo pagano e quelli di base, in mancanza di dati per l'iscrizione, non lo pagano.

➤ Oggetto di Noleggio

I figli di questa entità vengono tolti, dal fatto che non hanno tanti dati aggiuntivi. Mettiamo un altro attributo per Oggetto di Noleggio che sarà usato per identificare il tipo. L'attributo per la batteria delle biciclette elettriche e scooter avrà la cardinalità (0,1) quanto verrà messo dentro l'entità.

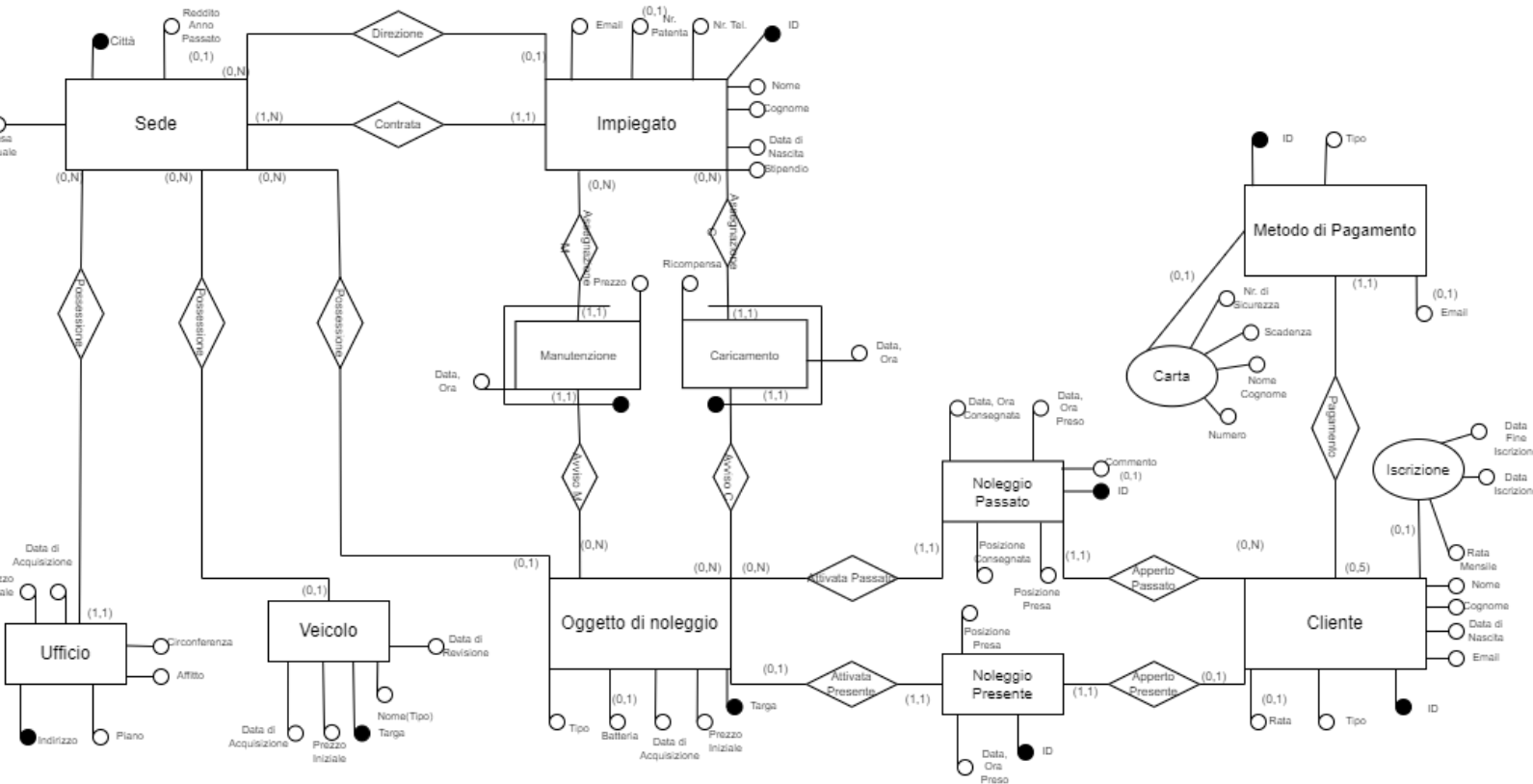
➤ Proprietà in Possesso

Questa entità sarà l'unica, tra quelle che avevano una generalizzazione, che venga cancellata. I suoi attributi vengono messi dentro le entità figli. La cardinalità della relazione tra i figli e le sedi non cambieranno, siccome una proprietà già comprata dall'azienda potrebbe non essere avere una sede associata.

3-Scelta degli identificatori principali

Tutte le entità hanno le loro identificatori apposite e non è necessario aggiungere altri indici.

Schema Logico



4-Creazione delle tabelle e le sue relazioni

L'attributo prima della " → " significa che dipende dall'attributo dopo.

Gli attributi sottolineati sono chiavi per la tabella. L'attributo con il " ° " simbolo significa che può accettare valori nulli.

Cliente(Matricola, Email, Nome, Cognome, Data_di_Nascita, Tipo, Rata°, Rata_Mensile°, Data_Iscrizione°, Data_Fine_Iscrizione°)

Metodo_di_Pagamento(ID, Tipo, Nr_di_Sicurezza°, Numero°, Nome_Cognome°, Scadenza°, Email°, Matricola → Matricola.Cliente)

Noleggio_Passato(ID, Data_Ora_Consegnata, Posizione_Consegnata, Data_Ora_Preso, Posizione_Presa, Commento°, Matricola → Matricola.Cliente, Targa → Oggetto di Noleggio.Targa)

Noleggio_Presente(ID, Data_Ora_Preso, Posizione_Presa, Matricola → Matricola.Cliente, Targa → Oggetto di Noleggio.Targa)

Oggetto_di_Noleggio(Targa, Tipo, Batteria°, Data_Acquisizione, Prezzo_Iniziale, Sede → Sede.Città)

Manutenzione(Data_Ora, Targa → Oggetto_di_noleggio.Targa, Matricola → Impiegato.Matricola, Prezzo)

Caricamento(Data_Ora, Targa → Oggetto_di_noleggio.Targa, Matricola → Impiegato.Matricola, Ricompensa)

Impiegato(Matricola, Nome, Cognome, Data_di_Nascita, Email, Stipendio, Nr_Tel, Nr_Patenta°, Sede → Sede.Città, Sede_Responsabilità° → Sede.Città)

Sede(Città, Spesa_Annuale, Reddito_Anno_Passato°)

Ufficio(Indirizzo, Affitto, Circonferenza, Piano, Data_Acquisizione, Prezzo_Iniziale, Sede → Sede.Città)

Veicolo(Targa, Nome, Data_Revisione, Data_Acquisizione, Prezzo_Iniziale, Sede → Sede.Città)

Query e Indici

1-Query

1.Dimostra tutti i impiegati con stipendio più basso della media.

```
SELECT matricola,stipendio, nome, cognome, email
FROM impiegato
WHERE stipendio <= (SELECT avg(stipendio)
                   FROM impiegato)
```

2.Dimostra i veicoli, dove si trovano e il suo nome, e da quanti impiegati possono essere usati.

```
DROP view if exists carsede;
CREATE view carsede AS
SELECT v.targa, v.nome, s.citta, v.sede
FROM veicolo V, sede s
WHERE v.sede = s.citta
;

SELECT count(i.nome) as nr_impiegati, c.targa,c.citta,c.nome
FROM impiegato I , carsede c
WHERE i.nr_patenta IS NOT NULL AND c.citta = i.sede
GROUP BY c.targa, c.citta, c.nome
```

3.Dimostra i dipendenti delle sede e anche quelle sede che non hanno hanno un dipendente.

```
SELECT citta, nome, cognome, email
FROM sede LEFT JOIN impiegato ON citta = sede_responsabilita
```

4.Dimostra chi dai impiegati ha fatto almeno 2 caricamenti.

```
SELECT I.matricola, I.nome, I.cognome, I.stipendio, I.email, count(C.matricola) as Numero_di_caricamenti
FROM impiegato I, caricamento C
WHERE I.matricola = C.matricola
GROUP BY I.matricola
HAVING count(C.matricola) >= 2
ORDER BY Numero_di_caricamenti DESC
```

5.Dimostra quante volte hanno usato una bici o scooter i clienti attivi.

```
SELECT count(np.id), c.matricola
FROM noleggio_passato NP, cliente C
WHERE c.matricola = np.matricola
GROUP BY c.matricola
```

6. Dimostra gli utenti con numero di matricola, che tipo sono, base o premium e quanti metodi di pagamento usano.

```
SELECT count(M.id), C.matricola, c.tipo
FROM cliente C, metodo_di_pagamento M
WHERE c.matricola = m.matricola
GROUP BY c.matricola
```

7. Dimostra quante volte i clienti hanno noleggiato nella sede attuale dove si trovano i mezzi di trasporti per noleggiare.

```
Drop view if exists tottal;
DROP VIEW IF EXISTS oggettopas;
DROP VIEW if exists oggettopres;
```

```
CREATE VIEW OggettoPas AS
SELECT o.targa, np.matricola, o.sede
FROM noleggio_passato NP FULL JOIN oggetto_di_noleggio O
ON o.targa = np.targa ;
```

```
CREATE VIEW OggettoPres AS
SELECT o.targa, ns.matricola, o.sede
FROM noleggio_presente Ns FULL JOIN oggetto_di_noleggio O
ON o.targa = ns.targa;
```

```
CREATE VIEW Tottal AS
select s.citta, o.targa, o.matricola from oggettopas O Right join sede S on o.sede = s.citta
union all
select s.citta, o.targa, o.matricola from oggettopres O right join sede S on o.sede = s.citta;

SELECT s.citta, count(t.matricola)
FROM sede S, tottal T
WHERE s.citta = t.citta
GROUP BY s.citta
```

2-Indici

Le tabelle che subisce tante READ saranno i Noleggio_Passato e Noleggio_Presente. Da questo fatto è scelto di creare due indici per facilitare e velocizzare il processo di READ. Il read sarebbe necessario per indicare dove sono gli oggetti di noleggio per essere in grado l'utente di trovare quello più vicino possibile e chi sono attive.

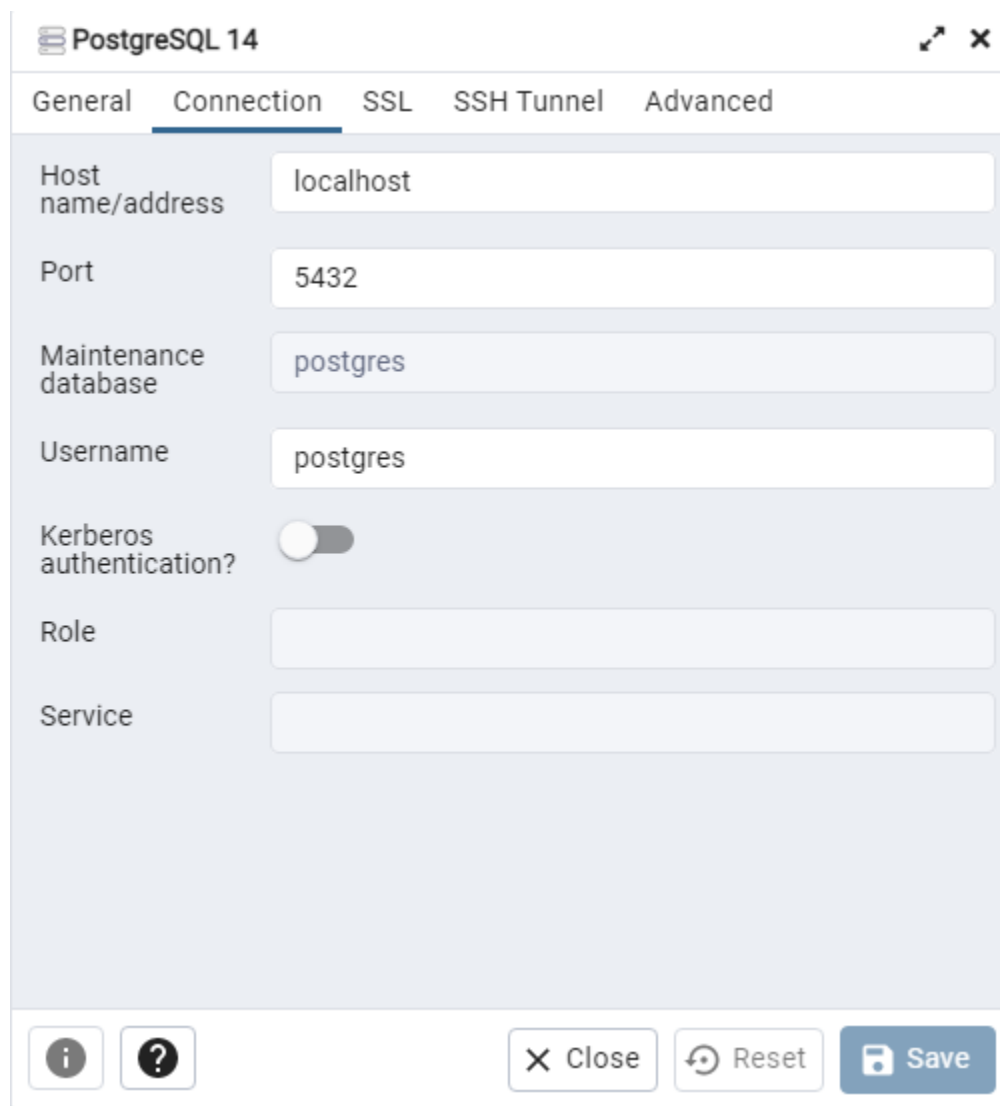
```
CREATE INDEX Posizione_Presente ON noleggio_presente(posizione_presa);
CREATE INDEX Posizione_Lasciato ON noleggio_passato(posizione_consegnata);
```

Codice C++

1-Descrizione

```
#define PG_HOST "127.0.0.1"
#define PG_USER "postgres"
#define PG_DB "postgres"
#define PG_PASS "password"
#define PG_PORT 5432
```

Il codice per dimostrare le tabelle sarà chiamato codice_sql.cpp. Prima di compilare sarebbe necessario modificare il file. Le modificazioni devono essere il nome del host name/address → PG_HOST, nome dell'utente → PG_USER, il password del database → PG_PASS e il numero del port usato → PG_PORT. Queste si trovano nelle proprietà del PostgreSQL 14. Anche il nome del database sarà necessario a cambiare → PG_DB



The screenshot shows the 'PostgreSQL 14' connection configuration window. The 'Connection' tab is selected, showing fields for Host name/address (localhost), Port (5432), Maintenance database (postgres), Username (postgres), Kerberos authentication? (disabled), Role, and Service. The bottom of the window has buttons for 'Close', 'Reset', and 'Save'.

Field	Value
Host name/address	localhost
Port	5432
Maintenance database	postgres
Username	postgres
Kerberos authentication?	<input type="checkbox"/>
Role	
Service	

Per compilare il codice e necessario copiare il codice nel terminale `g++ codice_sql.cpp -Ldependencies\lib -lpq -o [Nome Casuale]`. Dopo avere creato il database, le tabelle che si trovano nel file code.sql , e aver fatto l'inserimenti che si trovano nella file insert.sql potete aprire la file compilata [Nome Casuale].exe.