



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI MATEMATICA

LAUREA TRIENNALE IN INFORMATICA

PROGETTO DI BASI DI DATI

**Basi di Dati di un Sistema Informativo
per la Gestione di una Scuola Guida**

Andrea Difino, Davide Colabove

Indice

1 Abstract	1
2 Analisi dei Requisiti	1
3 Progettazione Concettuale	3
4 Progettazione Logica	5
4.1 Analisi delle ridondanze	7
4.2 Eliminazione delle generalizzazioni	8
4.3 Scelta di identificatori primari	9
4.4 Partizionamento/Accorpamento di Entità e Relationships	9
4.5 Schema Relazionale	10
5 Implementazione in PostgreSQL e Definizione delle Query	11
5.1 Definizione delle query	11
5.2 Creazione degli indici	14
6 Applicazione Software	15

1 Abstract

Questo lavoro sviluppa una base di dati progettata per gestire in modo strutturato e coerente le informazioni relative alle scuole guida, agli istruttori, agli iscritti ai corsi e agli esami sostenuti dagli allievi.

L'obiettivo principale è fornire un sistema che supporti le attività di pianificazione, monitoraggio e ottimizzazione delle risorse didattiche, garantendo un accesso organizzato ai dati e facilitando la loro analisi.

Nel contesto della formazione alla guida, la base di dati proposta distingue tra lezioni teoriche e pratiche, gestendo attributi specifici quali assegnazione di aule, veicoli disponibili e disponibilità degli istruttori. Gli iscritti sono tracciati in base ai percorsi di formazione intrapresi e ai risultati degli esami, con particolare attenzione alle performance nelle prove teoriche e pratiche. Il sistema registra anche le prenotazioni delle lezioni e i pagamenti associati ai servizi offerti.

Questa base di dati è progettata per garantire un'archiviazione efficiente e strutturata delle informazioni formative, migliorando il recupero e l'analisi dei dati. L'organizzazione sistematica delle informazioni contribuisce a un utilizzo più efficace delle risorse didattiche, ottimizzando la gestione delle scuole guida e supportando i processi decisionali.

2 Analisi dei Requisiti

Questa sezione riassume i requisiti a cui deve sottostare la base di dati.

Iscritto. Gli iscritti sono le persone che frequentano i corsi della scuola guida e sono identificati attraverso:

- Codice Fiscale (identificativo univoco).
- Nome.
- Cognome.
- Numero di telefono.
- Indirizzo email.
- Indirizzo di residenza.
- Città (CAP, Nome, Provincia).
- Categoria di patente richiesta (A, B, C, ecc.).
- Data di iscrizione.

Ogni iscritto può partecipare a più lezioni teoriche e pratiche, sostenere più esami, e può lasciare recensioni su lezioni o istruttori.

Abbiamo scelto di implementare l'attributo CAP perché esistono città distinte ma con lo stesso CAP (es. Terrassa Padovana, Pernumia, Albignasego, Pozzonovo che ricadono tutte sotto il CAP 35020) ma al contempo anche città (es. Padova, Roma, Milano) con più CAP al loro interno.

Istruttore. Gli istruttori sono i docenti responsabili delle lezioni e sono descritti tramite:

- Codice Fiscale (identificativo univoco).
- Nome.
- Cognome.
- Numero di telefono.
- Indirizzo email.
- Anni di esperienza professionale.

Un istruttore può tenere più lezioni sia teoriche che pratiche, ma non può essere assegnato contemporaneamente a due lezioni sovrapposte.

Patente. Le Patenti determinano per cosa gli istruttori sono abilitati ad insegnare:

- Tipo della patente.
- Descrizione.

Un istruttore può essere abilitato ad insegnare più patenti, ma una patente può non essere insegnata da nessun istruttore.

Veicolo. I veicoli sono utilizzati per le lezioni pratiche e per gli esami pratici. Per ogni veicolo vengono registrate le seguenti informazioni:

- Targa (identificativo univoco).
- Modello.
- Tipo di veicolo (Auto, Moto, ecc.).
- Anno di immatricolazione.
- Stato operativo (disponibile, in manutenzione, non disponibile).

Ogni veicolo può essere utilizzato in più lezioni pratiche o esami pratici. Tuttavia, ogni lezione pratica o esame pratico è associato a un solo veicolo alla volta.

Aula. Le aule sono gli spazi in cui si svolgono le lezioni teoriche e sono caratterizzate da:

- Nome aula. (identificativo univoco)
- Numero di posti disponibili.
- Tipologia di attrezzatura didattica disponibile (proiettore, lavagna, ecc.).

Un'aula può ospitare più lezioni, ma non contemporaneamente.

Lezione. Le lezioni rappresentano le attività formative offerte dalla scuola guida e sono caratterizzate da:

- Orario di inizio lezione.
- Data.
- Argomento lezione.
- Numero di Partecipanti.

si suddivide in:

- Lezione teorica;
- Lezione pratica che aggiunge:
 - Durata.

Ogni lezione ha un solo istruttore assegnato. Le lezioni teoriche possono avere più iscritti, mentre quelle pratiche sono individuali. Le lezioni teoriche hanno sempre la stessa durata, mentre le lezioni pratiche possono avere durate diverse.

Prenotazione. Le prenotazioni associano un iscritto a una lezione:

- Data di prenotazione.
- Ora della prenotazione.

Un iscritto può prenotare più lezioni.

Esame. Gli esami valutano la preparazione degli iscritti e sono caratterizzati da:

- Data dell'esame.
- Esito (superato o non superato).
- Durata.

si suddivide in:

- Esame teorico:
 - Sede.
 - Punti.
- Esame pratico.

Un iscritto può sostenere più tentativi di esame.

Recensione. Gli iscritti possono lasciare recensioni relative a:

- Una lezione frequentata.
- Un istruttore con cui hanno svolto lezioni.

Per ogni recensione sono registrati:

- Data della recensione.
- Gradimento (da 1 a 5 stelle).
- Commento testuale.
- Oggetto

Un iscritto può recensire un istruttore o una lezione.

Pagamento. Gli iscritti devono effettuare i pagamenti relativi ai servizi formativi:

- Data del pagamento.
- Ora del pagamento.
- Importo versato.
- Metodo di pagamento (carta di credito, bonifico, contante).
- Stato del pagamento (pagato, da pagare).

Ogni pagamento è associato a un singolo iscritto.

3 Progettazione Concettuale

La [Figura 1](#) mostra il diagramma Entità-Relazione (E-R) che riassume i requisiti descritti in [Sezione 2](#).

Esistono esattamente due tipi di lezioni, teoriche e pratiche.

Ogni lezione pratica prevede l'assegnazione di un veicolo specifico e la partecipazione di un singolo iscritto per volta, mentre le lezioni teoriche si svolgono in un'aula e possono coinvolgere più iscritti contemporaneamente.

Gli istruttori sono responsabili della conduzione delle lezioni: ogni istruttore può tenere sia lezioni teoriche che pratiche, ma non può essere assegnato a più lezioni contemporaneamente. Gli iscritti sono gli allievi della scuola guida e possono prenotare lezioni, sostenere esami, e lasciare recensioni sugli istruttori o sulle lezioni frequentate.

L'analisi dei requisiti dà particolare enfasi agli esami pratici, i quali prevedono l'uso obbligatorio di un veicolo assegnato; per questo motivo, il concetto di esame pratico viene rappresentato come un'estensione dell'esame, caso particolare dell'esame generale. In particolare, si tiene traccia degli esami pratici svolti con uno specifico veicolo, registrando l'esito della prova.

Ogni prenotazione associa un iscritto a una lezione, garantendo la corretta gestione delle disponibilità di istruttori, aule e veicoli. Analogamente, ogni pagamento è riferito a un iscritto e rappresenta una transazione economica per il servizio formativo ricevuto.

Tabella 1 (descritta successivamente) riassume tutte le entità e **Tabella 2** (descritta successivamente) le relazioni individuate dall'analisi dei requisiti e rappresentate nel diagramma E-R, con i relativi attributi rilevanti. Per le entità principali viene anche fornito l'identificatore, che può includere riferimenti a relazioni nei casi di Lezione, Esame, Prenotazione e Pagamento.

Il presente schema E-R non permette di rappresentare direttamente i seguenti vincoli:

1. Un istruttore i non può tenere contemporaneamente più lezioni. Se un istruttore insegna due lezioni l_1 e l_2 , allora le date e orari devono essere diversi:

$$(l_1, i) \in Insegnamento \wedge (l_2, i) \in Insegnamento \Rightarrow (l_1.Data \neq l_2.Data \vee l_1.OraInizio \neq l_2.OraInizio)$$

2. Un iscritto s non può partecipare a due esami nello stesso momento. Un iscritto non può sostenere due esami e_1 ed e_2 nello stesso giorno e ora:

$$(e_1, s) \in Partecipazione \wedge (e_2, s) \in Partecipazione \Rightarrow (e_1.Data \neq e_2.Data)$$

3. Un iscritto può lasciare una recensione solo su una lezione a cui ha fatto prenotazione. Se un iscritto s scrive una recensione r relativa a una lezione l , allora s deve aver richiesto una prenotazione p per quella lezione:

$$(r, l) \in ValutazioneLezione \wedge (p, l) \in Appuntamento \Rightarrow \exists s((p, s) \in Richiesta \wedge (r, s) \in Feedback)$$

4. Un iscritto s non può partecipare lo stesso giorno sia ad un esame teorico e_1 e ad un esame pratico e_2 e la data dell'esame teorico deve obbligatoriamente precedere quella dell'esame pratico:

$$(s, e_1) \in EsameTeorico \wedge (s, e_2) \in EsamePratico \Rightarrow (e_1.Data \neq e_2.Data \wedge e_1.Data < e_2.Data)$$

5. La data di una prenotazione p deve essere antecedente alla data della lezione l prenotata.

$$(p, l) \in Appuntamento \Rightarrow (p.DataPrenotazione < l.Data)$$

6. Una recensione non può essere riferita contemporaneamente ad una lezione e ad un istruttore.

7. Se un veicolo v è in manutenzione o non disponibile allora non può essere usato per una lezione pratica.

$$(v, l) \in Utilizzo \rightarrow (v.Stato \neq \text{"In Manutenzione"} \wedge v.Stato \neq \text{"Non Disponibile"})$$

8. Il numero di partecipanti a una lezione teorica non può superare il numero di posti disponibili nell'aula assegnata:

$$(l, a) \in Assegnazione \wedge l.NumeroPartecipanti > a.Posti \Rightarrow \text{Vincolo Violato}$$

9. Nel caso di una lezione pratica, il numero di partecipanti è sempre 1.



Figura 1: Diagramma E-R della base di dati relativa alla gestione di una Scuola Guida

4 Progettazione Logica

In questa sezione viene presentato lo schema logico relazionale derivato dal modello concettuale;

Ciascuna entità concettuale è trasformata in una tabella, definendo attributi, chiavi primarie e chiavi esterne per rappresentare i legami.

Verrà effettuata una analisi degli attributi ridondanti per determinare la loro rimozione o mantenimento al fine per migliorare l'efficienza di archiviazione e accesso.

Le due generalizzazioni concettuali verranno risolte mediante l'introduzione di un attributo discriminante che distingue i tipi di istanza, con i campi specifici resi NULLABLE. L'attributo composto *Città* viene reso una tabella a se stante e suddiviso in tre attributi distinti: CAP, Provincia e Nome, per facilitare la gestione dei dati.

Tale modifica sarà rappresentata in [Figura 2](#).

Infine viene riportato il diagramma Entità-Relazione (E-R) ristrutturato, che riporta le modifiche apportate in questa sezione al fine di avere un modello logico completo e coerente.

Entità	Descrizione	Attributi	Identificatore
Iscritto	Iscritto alla scuola guida	<i>CF, DataIscrizione, TipoPatente, Città(CAP, Provincia, Nome), Numero, Email, Indirizzo, Cognome, Nome</i>	<i>CF</i>
Istruttore	Istruttore pratico e teorico della scuola guida	<i>CF, Nome, Cognome, Numero, Email, AnniEsperienza</i>	<i>CF</i>
Patente	Patente per l'abilitazione	<i>TipoPatente, Descrizione</i>	<i>TipoPatente</i>
Veicolo	Veicolo utilizzato per esami pratici e lezioni pratiche	<i>Targa, Modello, Tipo, Stato, AnnoImmatricolazione</i>	<i>Targa</i>
Aula	Aula per le lezioni teoriche	<i>NomeAula, Attrezzatura, Posti</i>	<i>NomeAula</i>
Lezione	Lezione generica	<i>ArgomentoLezione, Data, OraInizio, NumeroPartecipanti</i>	<i>Data, ArgomentoLezione</i>
Lezione Teorica	Lezione teorica scuola guida		
Lezione Pratica	Lezione pratica scuola guida	<i>Durata</i>	
Prenotazione	Prenotazione lezioni	<i>Ora, DataPrenotazione</i>	<i>DataPrenotazione, relazione "Richiesta"</i>
Esame	Esame generico	<i>Data, Esito, Durata</i>	<i>Data, relazione "Partecipazione"</i>
Esame Teorico	Esame teorico con quiz	<i>Sede, Punti</i>	
Esame Pratico	Esame pratico con prova di guida		
Recensione	Recensione istruttori e lezioni	<i>Oggetto, Commento, Data, Gradimento</i>	<i>Oggetto, relazione "Feedback"</i>
Pagamento	Pagamenti associati ai servizi offerti	<i>Importo, Data, OraPagamento, Stato, Metodo</i>	<i>OraPagamento, Data, relazione "Quota"</i>

Tabella 1: Entità

Relazione	Descrizione	Componente	Attributi
Feedback	Feedback di un iscritto	Iscritto, Recensione	
Abilitazione	Abilitazione di un istruttore per una patente	Istruttore, Patente	
Insegnamento	Svolgimento della lezione da parte di un istruttore	Istruttore, Lezione	
Appuntamento	Prenotazione di una lezione	Prenotazione, Lezione	
Richiesta	Richiesta di un iscritto per una prenotazione	Iscritto, Prenotazione	<i>stato</i>
Partecipazione	Iscrizione ad un esame di un iscritto	Esame, Iscritto	

Quota	Versamento di una quota da parte di un iscritto	Iscritto, Pagamento	
Uso	Utilizzo di un veicolo per un esame pratico	Esame Pratico, Veicolo	
Utilizzo	Utilizzo di un veicolo per una lezione pratica	Veicolo, Lezione Pratica	
Assegnazione	Aula usata per una lezione teorica	Aula, Lezione Teorica	

Tabella 2: Relazioni

4.1 Analisi delle ridondanze

L'attributo *NumeroPartecipanti* presente in *Lezione*, che memorizza il conteggio dei partecipanti a una specifica data e ora, presenta una ridondanza. Questo valore può infatti essere ottenuto al volo tramite una semplice query di aggregazione sulla relazione *Prenotazione*.

Tale attributo viene modificato ogni volta che si inserisce o si elimina una prenotazione per la lezione (evento relativamente frequente), mentre viene letto molto spesso, ad esempio ogni volta che un utente o un sistema di report richiede il numero attuale di partecipanti alla lezione. Si stima che:

- **Operazione 1 (60 a settimana)** : vengono create o cancellate circa 60 prenotazioni a settimana;
- **Operazione 2 (200 a settimana)** : siano effettuate in media 200 letture a settimana del numero di partecipanti.

Assumendo i seguenti volumi nella base di dati:

Concetto	Costrutto	Volume
Lezione	E	800
Prenotazione	E	16000

la seguente analisi serve per stabilire se sia utile o meno tenere l'attributo ridondante *NumeroPartecipanti* in *Lezione*.

CON RIDONDANZA: Analizziamo prima il costo totale con ridondanza:

- **Operazione 1:**

Concetto	Costrutto	Accessi	Tipo	
Prenotazione	E	1	S	× 60
Lezione	E	1	S	× 60

- **Operazione 2:**

Concetto	Costrutto	Accessi	Tipo	
Lezione	E	1	L	× 200

Assumendo *costo doppio per gli accessi in scrittura*, il costo totale è:

$$\text{Costo Totale} = 60 \times 4 + 200 = \boxed{440}$$

SENZA RIDONDANZA: Analizziamo il costo totale senza ridondanza:

• **Operazione 1:**

Concetto	Costrutto	Accessi	Tipo
Prenotazione	E	1	S

× 60

Non c'è più il contatore in Lezione quindi si usa solo l'entità Prenotazione

• **Operazione 2:**

Concetto	Costrutto	Accessi	Tipo
Prenotazione	E	1	L
Lezione	E	1	L

× 200
× 200

Assumendo *costo doppio per gli accessi in scrittura*, il costo totale è:

$$\text{Costo Totale} = 60 \times 2 + 200 \times 2 = \boxed{520}$$

L'analisi suggerisce quindi di tenere l'attributo ridondante, ottimizzando così il numero di accessi.

4.2 Eliminazione delle generalizzazioni

Le generalizzazioni descritte in [Sezione 3](#) vengono eliminate attraverso una ristrutturazione dello schema concettuale, con l'obiettivo di semplificare la successiva implementazione del modello relazionale e ridurre la presenza di valori NULL.

Le modifiche vengono applicate come segue:

• **Lezione**

La generalizzazione totale ed esclusiva di LEZIONE viene rimossa accorpando le entità figlie Lezione Teorica e Lezione Pratica all'interno dell'entità padre, come mostrato in [Figura 2](#). Questa scelta semplifica la struttura del modello e permette di distinguere i due tipi di lezione attraverso un apposito attributo discriminante Tipo, che ne indica la natura teorica o pratica.

Inoltre, l'attributo proprio *Durata*, specifico delle sole lezioni pratiche, viene incluso direttamente in Lezione come campo NULLABLE, evitando così la creazione di entità separate.

Se si fosse mantenuta un'unica entità Lezione con tutti gli attributi relativi a entrambe le sottoclassi senza distinguere i casi, le lezioni teoriche avrebbero presentato campi nulli come *Durata*, mentre le lezioni pratiche non avrebbero utilizzato altri attributi eventualmente specifici delle lezioni teoriche (come *Aula*).

Separando logicamente i due casi tramite l'attributo *TipoLezione* e rendendo *Durata* facoltativo, si garantisce che ogni istanza contenga solo le informazioni rilevanti. In questa soluzione, coerentemente con la metodologia vista a lezione, non sono presenti entità figlie autonome e l'identificazione avviene direttamente tramite i campi della superclasse.

Sarebbe stato possibile mantenere le entità figlie, spostando in esse gli attributi specifici, ma ciò avrebbe comportato la duplicazione delle relazioni comuni come Prenotazione e Recensione verso entrambe le sottoclassi.

• **Esame**

In maniera simile, la generalizzazione totale ed esclusiva di ESAME viene sostituita accorpando le entità figlie all'interno dell'entità padre, come mostrato in [Figura 2](#).

Anche in questo caso, seguendo un ragionamento simile a quello fatto per la generalizzazione precedente, questo cambiamento consente la riduzione di valori nulli. Essendo la generalizzazione totale (ogni esame è necessariamente teorico o pratico), la soluzione di eliminare l'entità padre Esame non sarebbe corretta.

L'integrazione avviene mediante l'inclusione degli attributi propri delle sottoclassi, come ad esempio *Punti* e *Sede* dell'esame teorico, direttamente in *Esame* come campi NULLABLE, distinguendo i due casi attraverso un attributo discriminante *TipoEsame*.

Questo consente di rappresentare le informazioni in modo compatto e coerente, evitando la duplicazione di relazioni comuni come quella con *Iscritto*.

4.3 Scelta di identificatori primari

Città : alla nuova entità *Città* viene aggiunto l'attributo *CodiceCatastale* anche noto come "codice Belfiore" che identifica univocamente ogni comune nei registri catastali.

Pagamento : assumiamo che ogni *Iscritto* possa effettuare più di un pagamento in un giorno per cui, al fine di minimizzare le chiavi primarie ID abbiamo deciso di identificare ogni pagamento mediante una chiave primaria composta dall'attributo *Data* e dall'attributo *OraPagamento*. In questo modo evitiamo che ci possano essere due tuple uguali nel caso in cui un iscritto effettui più pagamenti nella stessa giornata.

4.4 Partizionamento/Accorpamento di Entità e Relationships

In questa sezione si analizzano le entità e le relazioni per determinare se sia opportuno partizionare o accorpate alcune di esse:

Città.

Creiamo quindi una ulteriore entità *Città* partendo dall'attributo composto *Città* che contiene gli attributi: *CAP*, *Provincia* e *NomeCittà*.



Figura 2: Diagramma E-R Ristrutturato

4.5 Schema Relazionale

Lo schema ristrutturato in [Figura 2](#) contiene solamente costrutti mappabili in corrispettivi dello schema relazionale - detto anche schema logico. Lo schema logico è rappresentato a seguire, dove l'asterisco dopo il nome degli attributi indica quelli che ammettono valori nulli.

- **Lezione**(ArgomentoLezione, Data, Orainizio, Tipo, NumeroPartecipanti, CFIstruttore, Durata*, IDAula*, VeicoloUsato*)
 - Lezione.CFIstruttore → Istruttore.CF
 - Lezione.IDAula → Aula.Nome

- Lezione.VeicoloUsato → Veicolo.Targa
- **Iscritto**(CF, DataIscrizione, TipoPatente, Numero, Email, Indirizzo, Cognome, Nome, IDCitta)
 - Iscritto.IDCitta → Città.CodiceCatastale
- **Città**(CodiceCatastale, NomeCittà, Provincia, CAP)
- **Istruttore**(CF, Nome, Cognome, AnniEsperienza, Numero, Email)
- **Patente**(TipoPatente, Descrizione)
- **Veicolo**(Targa, Modello, Tipo, Stato, AnnoImmatricolazione)
- **Aula**(Nome, Attrezzatura, Posti)
- **Prenotazione**(DataPrenotazione, Iscritto, Email, Stato, Data, ArgomentoLezione)
 - Prenotazione.Iscritto → Iscritto.CF
 - Prenotazione.(Data, ArgomentoLezione) → Lezione.(Data, ArgomentoLezione)
- **Esame**(Data, Iscritto, Esito, Durata, TipoEsame, Sede*, Punti*)
 - Esame.Iscritto → Iscritto.CF
- **Recensione**(Oggetto, Iscritto, Commento, Data, Gradimento, CFIstruttore*, DataLezione*, ArgomentoLezione*)
 - Recensione.Iscritto → Iscritto.CF
 - Recensione.CFIstruttore → Istruttore.CF
 - Recensione.(DataLezione, ArgomentoLezione) → Lezione.(Data, ArgomentoLezione)
- **Pagamento**(OraPagamento, Data, Iscritto, Stato, Metodo, Importo)
 - Pagamento.Iscritto → Iscritto.CF
- **Partecipazione**(PersonaCF, DataEsame)
 - Partecipazione.PersonaCF → Iscritto.CF
 - Partecipazione.DataEsame → Esame.Data
- **Abilitazione**(IstruttoreCF, TipoPatente)
 - Abilitazione.IstruttoreCF → Istruttore.CF
 - Abilitazione.TipoPatente → Patente.TipoPatente

5 Implementazione in PostgreSQL e Definizione delle Query

Il file ScuolaGuida.sql contiene il codice SQL necessario per la creazione e il popolamento delle tabelle del database. Questo file include inoltre una serie di query per l'estrazione dei dati e un indice creato specificamente per migliorare le prestazioni di una di queste interrogazioni.

5.1 Definizione delle query

Di seguito vengono presentate e descritte le query con i relativi output generati e viene motivato l'utilizzo dell'indice proposto.

Query 1 Calcolare, per ogni tipo di patente, il numero di esami teorici sostenuti, la media dei punti ottenuti e i promossi, considerando solo chi ha ottenuto in media più di 20 punti. I risultati sono ordinati per media punti in modo decrescente.

```

1 SELECT
2   i.TipoPatente,
3   COUNT(*) as NumeroEsami,
4   AVG(e.Punti) as MediaPunti,
5   COUNT(e.Esito) as NumeroPromossi
6 FROM Iscritto i
7 JOIN Esame e ON i.CodiceFiscale = e.CodiceFiscale
8 WHERE e.TipoEsame = 'Teorico'
9 GROUP BY i.TipoPatente
10 HAVING AVG(e.Punti) >= 20
11 ORDER BY MediaPunti DESC;

```

Un estratto dell'output:

	tipopatente text	numeroesami bigint	mediapunti numeric	numeropromossi bigint
1	C1	1	30.0000000000000000	1
2	A2	2	27.5000000000000000	2
3	A1	1	25.0000000000000000	1
4	B	5	25.0000000000000000	5

Query 2 Analizzare i veicoli per tipo, conteggiando il numero totale, quelli disponibili, i tipi di patente necessari e le prenotazioni effettuate. I dati sono raggruppati per tipo di veicolo e ordinati per numero di veicoli decrescente.

```

1 SELECT
2   v.TipoVeicolo,
3   COUNT(*) as NumeroVeicoli,
4   COUNT(CASE WHEN v.Stato = 'Disponibile' THEN 1 END) as VeicoliDisponibili,
5   COUNT(DISTINCT i.TipoPatente) as TipiPatentiNecessarie,
6   COUNT(DISTINCT p.DataPrenotazione) as NumeroPrenotazioni
7 FROM Veicolo v
8 JOIN Iscritto i ON (
9   (v.TipoVeicolo = 'Auto' AND i.TipoPatente = 'B') OR
10  (v.TipoVeicolo = 'Moto' AND i.TipoPatente = 'A') OR
11  (v.TipoVeicolo = 'Camion' AND i.TipoPatente = 'C')
12 )
13 LEFT JOIN Prenotazione p ON i.CodiceFiscale = p.CodiceFiscale
14 GROUP BY v.TipoVeicolo
15 ORDER BY NumeroVeicoli DESC;

```

Un estratto dell'output:

	tipoveicolo text	numeroveicoli bigint	veicolidisponibili bigint	tipipatentinecessarie bigint	numeroprenotazioni bigint
1	Auto	77	55	1	11
2	Moto	14	12	1	2
3	Camion	12	6	1	2

Query 3 Visualizzare le prenotazioni accettate per veicoli disponibili, mostrando data, ora, modello, stato del veicolo e dati dell'iscritto. I risultati sono ordinati cronologicamente per data e ora della prenotazione.

```

1  SELECT
2      v.TipoVeicolo,
3      v.Modello,
4      v.Stato,
5      p.DataPrenotazione,
6      p.Ora,
7      i.Nome,
8      i.Cognome
9  FROM Veicolo v
10 JOIN Prenotazione p ON v.Stato = 'Disponibile'
11 JOIN Iscritto i ON p.CodiceFiscale = i.CodiceFiscale
12 WHERE v.Stato = 'Disponibile'
13        AND p.Stato = 'Accettata'
14 ORDER BY p.DataPrenotazione, p.Ora;

```

Un estratto dell'output:

	tipoveicolo text	modello character varying (30)	stato text	dataprenotazione date	ora time without time zone	nome character varying (20)	cognome character varying (20)
1	Auto	Fiat Panda	Disponibile	2025-05-01	09:00:00	Mario	Rossi
2	Moto	Honda CBR	Disponibile	2025-05-01	09:00:00	Mario	Rossi
3	Auto	Toyota Yaris	Disponibile	2025-05-01	09:00:00	Mario	Rossi
4	Camion	Mercedes Actros	Disponibile	2025-05-01	09:00:00	Mario	Rossi
5	Auto	Volkswagen Golf	Disponibile	2025-05-01	09:00:00	Mario	Rossi
6	Moto	Ducati Monster	Disponibile	2025-05-01	09:00:00	Mario	Rossi
7	Auto	Ford Fiesta	Disponibile	2025-05-01	09:00:00	Mario	Rossi
Total rows: 140		Query complete 00:00:00.144					

Query 4 Restituisce un elenco degli iscritti che hanno effettuato almeno un pagamento, mostrando per ciascuno quanto hanno versato e per che tipo di patente stanno studiando:

```

1  SELECT I.Nome, P.CodiceFiscale, I.TipoPatente, SUM(P.Importo) AS TotalePagato
2  FROM Pagamento P
3  JOIN Iscritto I ON P.CodiceFiscale = I.CodiceFiscale
4  GROUP BY I.Nome, P.CodiceFiscale, I.TipoPatente
5  HAVING SUM(P.Importo) > 0;

```

Un estratto dell'output:

	nome character varying (20)	codicefiscale character (16)	tipopatente text	totalepagato numeric
1	Davide	CNTDVD97U18H50...	B	600.00
2	Maria	RSSMRA86G07H50...	C1	550.00
3	Sara	FRRSRA98V19H501V	A2	550.00
4	Elisa	VLNLSA95S16H501S	A1	500.00
5	Anna	NRENN83D04H50...	B	500.00
6	Luca	CNTLCA87H08H50...	B	500.00

Query 5 Calcolare, per ogni istruttore, la media di gradimento ricevuta e il numero di recensioni scritte da iscritti. Vengono mostrati solo gli istruttori con una media di gradimento superiore a 4.

```

1  SELECT
2      I.Nome,
3      I.Cognome,
4      AVG(R.Gradimento) AS MediaGradimento,
5      COUNT(*) AS NumeroRecensioni
6  FROM Istruttore I
7  JOIN Recensione R ON I.CodiceFiscale = R.CFIstruttore
8  JOIN Iscritto S ON R.CodiceFiscale = S.CodiceFiscale
9  GROUP BY I.CodiceFiscale, I.Nome, I.Cognome
10 HAVING AVG(R.Gradimento) > 4.0
11 ORDER BY MediaGradimento DESC;

```

Un estratto dell'output:

	nome character varying (20)	cognome character varying (20)	mediagradimento numeric	numerorecensioni bigint
1	Mario	Bianchi	5.0000000000000000	1
2	Maria	Rossi	5.0000000000000000	1
3	Luca	Verdi	5.0000000000000000	1
4	Paolo	Verdi	4.5000000000000000	2

5.2 Creazione degli indici

Si suppone di voler ottimizzare la query 3, per la quale occorre considerare:

- Condizione di Join tra le tabelle Veicolo, Prenotazione e Iscritto, in particolare:
 - p.CodiceFiscale = i.CodiceFiscale
 - v.Stato = 'Disponibile'
- Condizioni di filtro su:
 - v.Stato = 'Disponibile'
 - p.Stato = 'Accettata'
- Ordinamento finale sui campi DataPrenotazione e Ora.

Per ottimizzare il punto 1 e 2, è opportuno creare due indici:

```

1  CREATE INDEX idx_veicolo_stato ON Veicolo USING HASH(Stato);
2
3  CREATE INDEX idx_prenotazione_cf_stato_data_ora ON Prenotazione(CodiceFiscale, Stato, DataPrenotazione, Ora);

```

L'indice **idx_veicolo_stato** permette di cercare rapidamente tutti i veicoli con stato Disponibile, utilizzando una struttura hash, particolarmente efficiente per confronti di uguaglianza.

L'indice **idx_prenotazione_cf_stato_data_ora**, invece, migliora l'accesso alla tabella Prenotazione in fase di join con la tabella Iscritto (tramite CodiceFiscale) e consente un accesso ordinato sui campi DataPrenotazione e Ora, riducendo i costi di ordinamento nel risultato finale.

6 Applicazione Software

Il file ScuolaGuidaQuery.c contiene il codice C necessario per connettersi al database e visualizzare a schermo i risultati delle query presentate nella [Sezione 5](#).

Una volta compilato ed eseguito il programma, verrà mostrato all'utente un menu con la seguente lista di interrogazioni disponibili:

- 1. Numero di esami teorici sostenuti per ogni tipo di patente, media punti e promossi (media >20 punti).
- 2. Analisi dei veicoli per tipo, conteggiando il numero totale, quelli disponibili, i tipi di patente necessari e le prenotazioni effettuate.
- 3. Visualizzazione delle prenotazioni accettate per veicoli disponibili, mostrando data, ora, modello, stato del veicolo e dati dell'iscritto.
- 4. Elenco degli iscritti che hanno effettuato almeno un pagamento mostrando per ciascuno quanto hanno versato e la patente per cui stanno studiando.
- 5. Calcolo della media di gradimento ricevuta da un istruttore e del numero di recensioni scritte da iscritti per ogni istruttore.

L'utente potrà selezionare la query da eseguire digitando il numero corrispondente.

A seguito un esempio di esecuzione del programma, in cui l'utente ha selezionato la query 1:

```

----- MENU QUERIES -----
1. Media punti e numero promossi per tipo patente (parametrico)
2. Info veicoli e tipi patenti necessarie
3. Prenotazioni accettate per veicoli disponibili
4. Pagamenti per Iscritto (parametrico)
5. Gradimento istruttori (parametrico)
7. Uscire
Seleziona un'opzione: 1
Inserisci la soglia minima di punti per filtrare i risultati (es. 20): 25

```

TIPOPATENTE	NUMEROESAMI	MEDIAPUNTI	NUMEROPROMOSI
C1	1	30.0000000000000000	1
A2	2	27.5000000000000000	2
A1	1	25.0000000000000000	1
B	5	25.0000000000000000	5