

UNIVERSITÀ
DEGLI STUDI
DI PADOVA



DIPARTIMENTO
MATEMATICA

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA"

PROGETTO DI BASI DI DATI

DATABASE DI UN E-COMMERCE DI
STRUMENTI MUSICALI

realizzato da
Giacomo Nalotto (mat. 2067755)
e Marco Piro (mat. 2068075)

Indice:

1.	<u>Abstract</u>	3
2.	<u>Analisi dei Requisiti</u>	4
3.	<u>Progettazione concettuale</u>	5
4.	<u>Progettazione logica</u>	7
4.1.	<u>Analisi delle Ridondanze</u>	7
4.2.	<u>Eliminazione delle Generalizzazioni</u>	10
4.3.	<u>Schema Relazionale</u>	10
5.	<u>Implementazione in PostgreSQL e Definizione delle Query</u>	11
5.1.	<u>Definizione delle Query</u>	12
5.2.	<u>Creazione degli Indici</u>	14
6.	<u>Applicazione Software</u>	15
6.1.	<u>Compilazione</u>	15
6.2.	<u>Spiegazione</u>	15
6.3.	<u>Codice</u>	15

1. Abstract

Questo progetto ha lo scopo di modellare e simulare una base di dati per la gestione di uno store specializzato nella vendita di strumenti musicali. Il sistema è progettato per memorizzare e gestire in modo efficiente tutte le informazioni relative ai prodotti, agli ordini, ai clienti, ai fornitori e alle recensioni. La base di dati distingue tra diverse categorie di strumenti (a corda, a fiato, a percussione, digitali), gestendo attributi specifici per ciascuna. In particolare, gli strumenti a percussione e a corda sono ulteriormente classificati in acustici ed elettronici. Il sistema tiene traccia sia delle caratteristiche tecniche dei prodotti, sia della disponibilità in magazzino e delle modalità di approvvigionamento. L'architettura dati implementa una gestione differenziata dell'anagrafica clienti, distinguendo tra clienti privati e aziendali, ciascuno con attributi specifici. Per i clienti privati è prevista l'integrazione di un sistema di recensioni che arricchisce il valore informativo del catalogo prodotti attraverso feedback qualitativi sugli acquisti effettuati. La struttura relazionale supporta la gestione temporale degli ordini, memorizzando per ciascun prodotto le quantità richieste in diversi periodi. Parallelamente, il sistema implementa un monitoraggio dettagliato dei fornitori e degli approvvigionamenti, tracciando con precisione le quantità ordinate e ricevute per ogni transazione. Questo approccio consente un'analisi accurata delle performance dei fornitori e dell'efficienza della catena di approvvigionamento. L'obiettivo finale è pianificare una struttura informativa che ottimizzi la gestione del catalogo prodotti, faciliti l'elaborazione degli ordini e supporti l'analisi delle vendite, sia fisica che online.

2. Analisi dei requisiti

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

- Prodotti

Ogni prodotto è identificato da un codice SKU ed è caratterizzato dalle seguenti informazioni:

- **Codice SKU** che distingue in modo univoco ogni prodotto
- **Nome**
- **Prezzo**
- **Paese di produzione** dal quale proviene
- **Quantità** in magazzino per ogni prodotto
- **Marca**
- **Data di inserimento** nel catalogo

I prodotti vengono divisi in strumenti A PERCUSSIONE, A CORDA, A FIATO e DIGITALI; mantengono gli attributi di prodotto più altri specifici per ogni genere di strumento.

Sia gli strumenti a percussione che quelli a corda si dividono in due sottocategorie: ACUSTICI ed ELETTRICI.

- Percussione

- **Tipo pelle**
- **Numero dei fusti**
- **Tipo di strumento**
 - **Acustico**
 - **Materiale principale** dello strumento
 - **Elettrico**
 - **Numero Suoni**

- Corda

- **Materiale del corpo** dello strumento
- **Numero delle corde**
- **Scala dell' "intonazione"** dello strumento
- **Tipo di strumento**
 - **Acustico**
 - **Materiale delle Corde**

- **Elettrico**
- **Tipo di PickUp**
- **Fiato**
 - **Materiale principale** dello strumento
 - **Chiave**
 - **Tipo di strumento**
 - **Tipo di imboccatura**

- **Digitale**
 - **Requisiti dell'alimentazione**
 - **Connettività**
 - **Numero dei tasti**
 - **Tipo di strumento**

- **Dettaglio Ordine** (relazione)
 - **Quantità** per ogni prodotto

- **Ordine**

Ogni ordine è identificato da un numero d'ordine ed è caratterizzato dai seguenti attributi:

- **Data** dell'ordine
- **Totale dell'ordine** in €
- **Numero dell'ordine** chiave univoca per ogni ordine
- **Stato** (in consegna / consegnato)
- **Metodo Pagamento**

- **Cliente**

Ogni cliente viene identificato tramite codice ID e ed è caratterizzato dai seguenti attributi:

- **ID**
- **Indirizzo** (utile per la consegna)
- **Email**
- **Password**
- **Telefono**
- **Data di registrazione** al sito

Un cliente può essere PRIVATO o BUSINESS.

- **Cliente Privato**

I clienti privati mantengono gli stessi attributi di cliente con l'aggiunta di:

- **Nome**
- **Cognome**

- **Cliente Business**

Oltre le informazioni generali, per i clienti business vengono registrati ulteriori dati quali:

- **Partita IVA**
- **Ragione Sociale**
- **Indirizzo della sede legale**

- **Recensione**

I clienti privati hanno la possibilità di recensire un prodotto acquistato e ogni recensione è definita da:

- **Punteggio** classificabile tramite un voto
- **Commento** riguardo al singolo prodotto
- **Titolo** della recensione
- **Data** di rilascio della recensione

- **Fornitore**

Il fornitore è colui che approvvigiona l'e-commerce ed è caratterizzato dai seguenti attributi:

- **Nome** dell'azienda
- **PersonaContatto** (persona che fa da tramite per la comunicazione)
- **Email**
- **Telefono**
- **Partita IVA** chiave primaria

- **Ragione sociale**
- **Indirizzo della sede**

- Approvvigionamento

Ciò che il fornitore procura all'e-commerce, ogni approvvigionamento ha come attributi:

- **DataConsegnaPrevista**
- **DataOrdine**
- **Stato** della consegna

- Dettaglio Approvvigionamento (relazione)

Relazione che lega Prodotti e Approvvigionamento, utile per rappresentare generalità quali:

- **Quantità Ricevuta** dal fornitore
- **Quantità Ordinata** dal fornitore

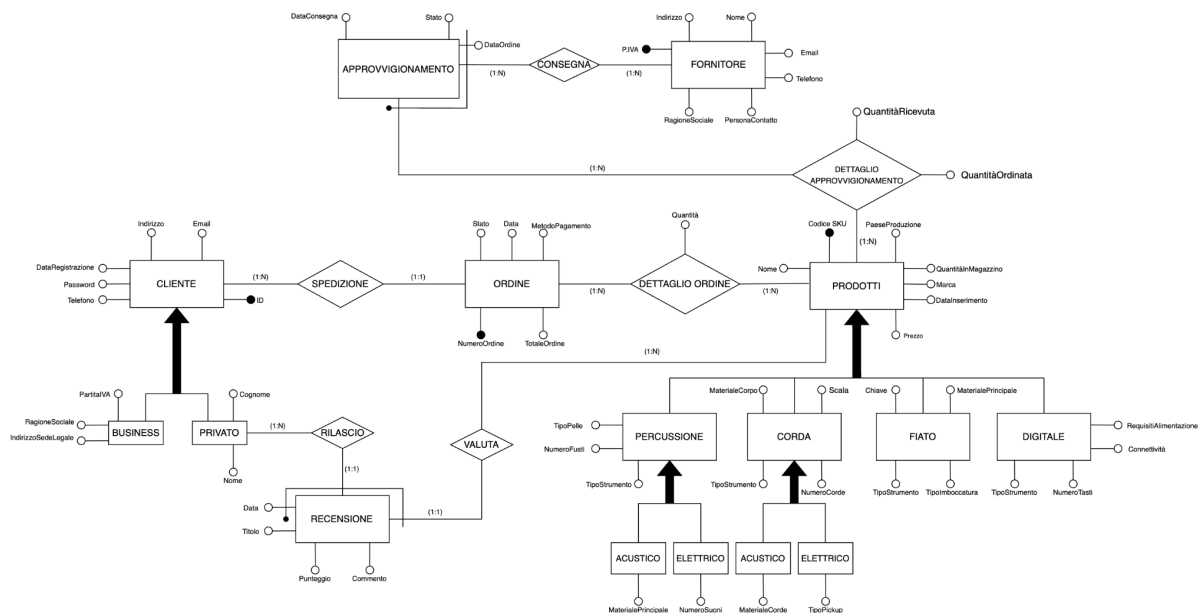


Figura 1: diagramma E-R della base di dati relativa alla gestione dell'e-commerce, riassume i requisiti sopra descritti

3. Progettazione concettuale

Ogni cliente registrato al sito di strumenti musicali ha la possibilità di fare un ordine comprendente uno o più prodotti, in quantità liberamente selezionabili. La spedizione viene gestita dal sistema che ne programma l'invio utilizzando le informazioni associate sia all'ordine sia al cliente, come ad esempio il numero dell'ordine, la data, ID del cliente e l'indirizzo di destinazione.

Sono previsti esattamente due tipi di clienti: Business e Privato. Un cliente Business può essere identificato in un qualsiasi negozio di strumenti musicali e in quanto tale viene individuato da una partita IVA e l'indirizzo della sede legale.

Il cliente Privato, invece, ha la possibilità di rilasciare una recensione per ciascun prodotto da lui ordinato; tali valutazioni sono pubblicamente visibili da tutti gli altri utenti registrati sul sito.

I prodotti disponibili sulla piattaforma appartengono a quattro macro categorie: percussione, corda, fiato e digitale. Le categorie percussione e corda sono ulteriormente suddivise nelle varianti elettrico ed acustico. Ogni strumento è caratterizzato da specifici attributi distintivi tra cui il materiale del corpo, la marca, il prezzo ed altri elementi distintivi che ne facilitano la selezione da parte dell'utente e che

garantiscono ampia scelta nell'acquisto del prodotto. L'intero catalogo di prodotti è rifornito da fornitori esterni. Le spedizioni di questi articoli viene rappresentata dall'entità 'Approvvigionamento' e la relativa relazione 'Dettaglio Approvvigionamento'. La logica che regola il processo di approvvigionamento rispecchia quella degli ordini effettuati dai clienti, mediante una struttura analoga basata sulla relazione Dettaglio Ordine. In entrambi i casi ogni singolo prodotto, ordinato o approvvigionato, viene rappresentato da una singola tupla nelle tabelle DettaglioOrdine e DettaglioApprovvigionamento.

Le seguenti tabelle riassumono tutte le entità e le relazioni individuate nell'analisi dei requisiti e rappresentate nel diagramma E-R.

ENTITÀ	DESCRIZIONE	ATTRIBUTI	IDENTIFICATORE
Prodotto	Insieme dei prodotti	CodiceSKU, Nome, Prezzo, QuantitàInMagazzino, DataInserimento, Marca, PaeseProduzione	CodiceSKU
Corda	Strumento a corda	TipoStrumento, MaterialeCorpo, NumeroCorde, Scala	
Acustico (Corda)	Strumento a corda acustico	MaterialeCorde	
Elettrico (Corda)	Strumento a corda elettrico	TipoPickup	
Percussione	Strumento a percussione	TipoStrumento, NumeroFusti, TipoPelle	
Acustico (Percussione)	Strumento a percussione acustico	MaterialePrincipale	
Elettrico (Percussione)	Strumento a percussione elettrico	NumeroSuoni	
Fiato	Strumento a fiato	TipoStrumento, MaterialePrincipale, Chiave, TipoImboccatura	
Digitale	Strumento digitale	TipoStrumento, NumeroTasti, Connettività, RequisitiAlimentazione	
Approvvigionamento	Articoli riforniti dai fornitori	DataConsegnaPrevista, DataOrdine, Stato	(Fornitore, Prodotto, DataOrdine)
Fornitore	Fornitore dei prodotti venduti nel sito	RagioneSociale, PartitaIVA, Nome, Indirizzo, Email, Telefono, PersonaContatto	PartitaIVA
Ordine	Ordine del cliente dal sito	NumeroOrdine, Data, Stato, MetodoPagamento, TotaleOrdine	NumeroOrdine
Cliente	Utente che ordina dal sito	ID, Email, Password, Indirizzo, Telefono, DataRegistrazione	ID
Cliente Privato	Cliente che ordina da privato	Nome, Cognome	

ENTITÀ	DESCRIZIONE	ATTRIBUTI	IDENTIFICATORE
Cliente Business	Negozio che ordina dal sito	PartitaIVA, RagioneSociale, IndirizzoSedeLegale	
Recensione	Giudizio lasciato dal cliente	Punteggio, Commento, Data, Titolo	(IDCliente, Prodotto, Data)

(a) Entità

RELAZIONE	DESCRIZIONE	COMPONENTE	ATTRIBUTI
Consegna	Recapito di una scorta all'e-commerce	Fornitore, Approvvigionamento	
Valuta	La recensione lasciata dal cliente valuta un singolo prodotto	Recensione, Prodotti	
Spedizione	L'ordine viene spedito al cliente	Ordine, Cliente	
Rilascio	Il cliente privato lascia una recensione su un singolo prodotto ordinato	Cliente privato, Recensione	
Dettaglio Ordine	Dettagli principali dell'ordine	Ordine, Prodotti	Quantità
Dettaglio Approvvigionamento	Dettagli principali sulle quantità di prodotti fornite all'e-commerce	Prodotti, Approvvigionamento	QuantitàOrdinata, QuantitàRicevuta

(b) Relazioni

4. Progettazione logica

La presente sezione descrive il processo di trasformazione dello schema concettuale in uno schema logico, con l'obiettivo di ottenere una rappresentazione dei dati che risulti chiara, coerente ed efficiente. Il processo ha inizio con un'attenta analisi del modello concettuale volta a individuare ed eliminare eventuali ridondanze, così da migliorarne la struttura complessiva e ottimizzarne l'organizzazione. A seguire, si procede con la rimozione delle due generalizzazioni presenti, al fine di semplificare ulteriormente il modello e garantire una maggiore aderenza alla logica del sistema. A conclusione dell'attività viene presentata la versione ristrutturata del diagramma, accompagnata da una spiegazione dettagliata delle modifiche effettuate e delle motivazioni che ne hanno guidato l'applicazione per poi concludere il tutto con lo schema relazionale.

4.1 Analisi delle ridondanze

L'attributo **TotaleOrdine** in **Ordine**, che memorizza il costo totale di ciascun ordine può essere ottenuto moltiplicando la **Quantità** di prodotti ordinati (il cui valore è contenuto nella relazione **DettaglioOrdine**) per il prezzo di ciascun prodotto facente parte dell'ordine. Questo attributo viene modificato ogni volta che viene effettuato un nuovo ordine da un cliente, ordine che verrà quindi inserito nel database e questo corrisponde all'**operazione 1**.

Mentre l'**operazione 2** consiste nella lettura del totale dell'ordine, cioè l'operazione con cui il sistema recupera l'importo complessivo da pagare per un ordine specifico, sommando il costo di tutti i prodotti inclusi.

Assumendo i seguenti volumi nella base di dati:

CONCETTO	COSTRUTTO	VOLUME	COMMENTO
Prodotto	Entità	500	Numero totale di strumenti musicali nel catalogo
DettaglioOrdine	Relazione	50.000	Ordini all'anno (~150 ordini al giorno)
Ordine	Entità	200.000	4 prodotti/ordine in media

La seguente analisi serve per stabilire se sia utile o meno tenere l'attributo ridondante TotaleOrdine in Ordine.

→ **CON RIDONDANZA** Analizziamo prima il costo totale con ridondanza

1. OPERAZIONE 1: INSERIMENTO ORDINE (150 volte/giorno)

CONCETTO	COSTRUTTO	ACCESSI	TIPO	QUANTITÀ
DettaglioOrdine	Relazione	1 per prodotto	scrittura	1 accesso x 4 prodotti x 150 ordini
DettaglioOrdine	Relazione	1 per prodotto	lettura	1 accesso x 4 prodotti x 150 ordini
Prodotto	Entità	1 per prodotto	lettura	1 accesso x 4 prezzi x 150 ordini
Ordine	Entità	1 per ordine	scrittura	x 150 ordini

Totale/ordine: 4 (inserimenti) + 4 + 4 + 1 = **13 accessi**

Giornaliero: 150 ordini × 13 = **1.950 accessi**

2. OPERAZIONE 2: LETTURA TOTALE

Stimiamo 3 letture/ordine: carrello, checkout, conferma → 150 ordini × 3 = 450 letture/giorno

CONCETTO	COSTRUTTO	ACCESSI	TIPO	QUANTITÀ
Ordine	Entità	1	lettura	x 450 letture

Giornaliero: 450 letture × 1 accesso = **450 accessi**

Assumendo costo doppio per gli accessi in scrittura

COSTO TOTALE GIORNALIERO CON RIDONDANZA:

Scritture: 1.950×2 (costo doppio) = 3.900

Lecture: 450

Totale: 4.350 accessi

→ **SENZA RIDONDANZA** Ora analizziamo il costo totale senza ridondanza

1. OPERAZIONE 1: INSERIMENTO ORDINE (150 volte/giorno)

CONCETTO	COSTRUTTO	ACCESSI	TIPO	QUANTITÀ
DettaglioOrdine	Relazione	1 per prodotto	scrittura	1 accesso x 4 prodotti x 150 ordini

Giornaliero: 150 ordini \times 4 accessi = **600 accessi**

2. OPERAZIONE 2: LETTURA TOTALE (450 volte/giorno)

CONCETTO	COSTRUTTO	ACCESSI	TIPO	QUANTITÀ
DettaglioOrdine	Relazione	1 per prodotto	lettura	1 accesso x 4 prodotti x 450 letture
Prodotto	Entità	1 per prodotto	lettura	1 accesso x 4 prezzi x 450 letture

Totale/lettura: $4 + 4 = 8$ accessi

Giornaliero: 450 letture \times 8 accessi = **3600 accessi**

Assumendo costo doppio per gli accessi in scrittura

COSTO TOTALE GIORNALIERO SENZA RIDONDANZA:

Scritture: 600×2 (costo doppio) = 1200

Lecture: 3600

Totale: 4.800 accessi

→ L'analisi suggerisce quindi di mantenere l'attributo ridondante, ottimizzando così il numero di accessi. La differenza di accessi tra la soluzione con ridondanza e senza ridondanza è ridotta ma provando ad aumentare le letture (5 letture per ordine, quindi 750 al giorno) il vantaggio della ridondanza cresce: con la ridondanza le letture ammontano a 4650 mentre senza ridondanza si arriva a quota 7200.

4.2 Eliminazione delle generalizzazioni

Le generalizzazioni descritte nell'[Analisi dei requisiti](#) 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 nulli. Le modifiche vengono applicate come segue:

1. PRODOTTO

La generalizzazione totale di PRODOTTO viene rimossa includendo le relazioni IS-PERCUSSIONE, IS-CORDA, IS-FIATO e IS-DIGITALE.

Questa scelta riduce la presenza di valori nulli: se si mantenesse un'unica entità PRODOTTO con tutti gli attributi relativi a tutti i vari tipi di strumenti musicali, gli strumenti a percussione avrebbero campi nulli per informazioni come il materiale delle corde o il numero di tasti o, per fare un altro esempio, gli strumenti a fiato avrebbero campi nulli per materiale delle corde. Separando le informazioni con relazioni dedicate, si garantisce che ogni entità contenga solo gli attributi rilevanti. In questa soluzione, coerentemente con la metodologia vista a lezione, gli identificatori di Percussione, Corda, Fiato e Digitale coincidono con le rispettive relazioni con Prodotto. Sarebbe stato possibile rimuovere l'entità Prodotto, incorporandone gli attributi nelle entità figlie. La soluzione sarebbe stata del tutto legittima, ma avrebbe portato alla necessità di quadruplicare le relazioni DettaglioOrdine, Vende, Valuta e Approvvigionamento.

2. PERCUSSIONE/CORDA

La generalizzazione totale degli strumenti a PERCUSSIONE e a CORDA viene rimossa accorpendo le entità figlie (ACUSTICO ed ELETTRICO) alle entità padre. Le tabelle risultanti conterranno tutti gli attributi dei padri e delle figlie, con valori NULL per gli attributi non applicabili. Questa soluzione è preferibile in quanto le operazioni sul database coinvolgono contemporaneamente dati delle superclassi (padri) e delle sottoclassi (figlie): quando si accede all'entità Percussione, inevitabilmente vengono coinvolti i dati in base alla natura acustica o elettrica dello strumento, lo stesso vale per l'entità Corda.

3. CLIENTE

In maniera simile al caso 1, la generalizzazione totale di CLIENTE viene sostituita da due relazioni IS-PRIVATO e IS-BUSINESS. Seguendo un ragionamento simile a quello fatto per la prima generalizzazione, questo cambiamento permette la riduzione di valori nulli. Sempre sui passi del caso 1, sarebbe stato possibile rimuovere l'entità Cliente, incorporandone gli attributi nelle entità figlie ma questo avrebbe portato alla necessità di duplicare la relazione Spedizione.

4.3 Schema relazionale

Lo schema ristrutturato (rappresentato in [figura 2](#)) include esclusivamente costrutti che possono essere direttamente mappati in corrispondenti elementi dello schema relazionale, detto anche schema logico. Segue la rappresentazione dello schema logico, in cui un asterisco posto accanto al nome di un attributo indica che tale attributo può assumere valori nulli.

- **Prodotti** (CodiceSKU, Nome, Prezzo, PaeseProduzione, QuantitàInMagazzino, Marca, DataInserimento)
- **Percussione** (Prodotto, TipoStrumento, NumeroFusti*, MaterialePrincipale, NumeroSuoni*, TipoPelle*)
 - Percussione.Prodotto → Prodotti.CodiceSKU
- **Corda** (Prodotto, TipoStrumento, MaterialeCorde*, Scala, MaterialeCorpo, NumeroCorde, TipoPickup*)
 - Corda.Prodotto → Prodotti.CodiceSKU

- **Fiato** (Prodotto, TipoStrumento, Chave, TipoImboccatura, MaterialePrincipale)
 - Fiato.Prodotto → Prodotti.CodiceSKU
- **Digitale** (Prodotto, TipoStrumento, RequisitiAlimentazione, Connettività, NumeroTasti)
 - Digitale.Prodotto → Prodotti.CodiceSKU
- **DettaglioOrdine** (Ordine, Prodotti, Quantità)
 - DettaglioOrdine.Ordine → Ordine.NumeroOrdine
 - DettaglioOrdine.Prodotti → Prodotti.CodiceSKU
- **Ordine** (NumeroOrdine, Cliente, Stato, Data, MetodoPagamento, TotaleOrdine)
 - Ordine.Cliente → Cliente.ID
- **Cliente** (ID, Indirizzo, Email, DataRegistrazione, Telefono, Password)
- **ClienteBusiness** (Cliente, PartitaIVA, RagioneSociale, IndirizzoSedeLegale)
 - ClienteBusiness.Cliente → Cliente.ID
- **ClientePrivato** (Cliente, Nome, Cognome)
 - ClientePrivato.Cliente → Cliente.ID
- **Recensione** (IDCliente, Prodotto, Data, Titolo, Punteggio, Commento)
 - Recensione.IDCliente → Cliente.ID
 - Recensione.Prodotto → Prodotti.CodiceSKU
- **DettaglioApprovvigionamento** (Prodotto, Fornitore, QuantitàOrdinata, QuantitàRicevuta)
 - DettaglioApprovvigionamento.Prodotto → Prodotti.CodiceSKU
 - DettaglioApprovvigionamento.Fornitore → Fornitore.PartitaIVA
- **Approvvigionamento** (Prodotto, Fornitore, DataOrdine, DataConsegna, Stato)
 - Approvvigionamento.Prodotto → Prodotti.CodiceSKU
 - Approvvigionamento.Fornitore → Fornitore.PartitaIVA
- **Fornitore** (PartitaIVA, Nome, Indirizzo, Email, Telefono, RagioneSociale, PersonaContatto)
- **Consegna** (Fornitore, DataOrdine, Prodotto)
 - Consegna.Fornitore → Fornitore.PartitaIVA
 - Consegna.Prodotto → Prodotti.CodiceSKU
 - Consegna.DataOrdine → Approvvigionamento.DataOrdine

Figura 2: diagramma E-R ristrutturato

5. Implementazione in PostgreSQL e Definizione delle Query

Il file E-Commerce.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: restituire i prodotti che hanno almeno 1 recensione con punteggio massimo (5 stelle)

```
SELECT p.CodiceSKU, p.Nome, COUNT(r.Prodotto) AS NumeroRecensioni5Stelle
FROM Prodotto p
JOIN Recensione r ON p.CodiceSKU = r.Prodotto
WHERE r.Punteggio = 5
GROUP BY p.CodiceSKU, p.Nome
HAVING COUNT(r.Prodotto) >= 1
ORDER BY NumeroRecensioni5Stelle DESC;
```

Di seguito un estratto dell'output della query:

	codicesku [PK] character varying (20)	nome character varying (100)	numerorecensioni5stelle bigint
1	CHT009	GS Mini Mahogany	3
2	DIG008	APC40 MKII	3
3	CHT001	Stratocaster Standard	3
4	CHT016	F310	1
5	CHT017	Ukulele Soprano	1
6	CHT024	Viola 15"	1
7	CHT025	Arpa Celtica 34 corde	1
8	CHT026	Les Paul Custom	1
Total rows: 40 Query complete 00:00:00.047			

- QUERY 2: restituire i totali (in €) delle vendite per ciascuna categoria di strumento in ordine decrescente

```
SELECT
CASE
WHEN sc.Prodotto IS NOT NULL THEN 'Strumenti a Corda'
WHEN sf.Prodotto IS NOT NULL THEN 'Strumenti a Fiato'
WHEN sp.Prodotto IS NOT NULL THEN 'Strumenti a Percussione'
WHEN sd.Prodotto IS NOT NULL THEN 'Strumenti Digitali'
END AS Categoria, SUM(o.TotaleOrdine) AS TotaleVenditeIn€
FROM Ordine o
JOIN DettaglioOrdine d_o ON o.NumeroOrdine = d_o.Ordine
JOIN Prodotto p ON d_o.Prodotto = p.CodiceSKU
LEFT JOIN StrumentoACorda sc ON p.CodiceSKU = sc.Prodotto
LEFT JOIN StrumentoAFiato sf ON p.CodiceSKU = sf.Prodotto
LEFT JOIN StrumentoAPercussione sp ON p.CodiceSKU = sp.Prodotto
LEFT JOIN StrumentoDigitale sd ON p.CodiceSKU = sd.Prodotto
GROUP BY Categoria
ORDER BY TotaleVenditeIn€ DESC;
```

Di seguito un estratto dell'output della query:

	categoria text	totale vendite in € numeric
1	Strumenti a Corda	196556.51
2	Strumenti a Fiato	162756.45
3	Strumenti Digitali	159512.19
4	Strumenti a Percussione	117343.44
Total rows: 4 Query complete 00:00:00.039		

- QUERY 3: ordinare in ordine decrescente i fornitori in termini di velocità nel rifornire i prodotti (considerando la differenza tra la data di consegna prevista e la data dell'ordine)

```
SELECT f.RagioneSociale, ROUND(AVG(a.DataConsegnaPrevista - a.DataOrdine), 1) AS  
TempoMedioConsegna, COUNT(*) AS NumeroOrdini  
FROM Fornitore f  
JOIN Approvvigionamento a ON f.PartitaIVA = a.Fornitore  
WHERE a.DataConsegnaPrevista IS NOT NULL  
GROUP BY f.PartitaIVA, f.RagioneSociale  
HAVING COUNT(*) >= 2  
ORDER BY TempoMedioConsegna ASC  
LIMIT 10;
```

Di seguito un estratto dell'output della query:

	ragionesociale character varying (100)	tempomedioconsegna numeric	numeroordini bigint
1	Kawai Europe GmbH Srl	5.7	3
2	Shure Inc. Srl	7.0	2
3	Nord Keyboards Italy Srl	7.5	2
4	M-Audio Italy Srl	8.5	2
5	Martin Guitar Italy Srl	9.0	3
6	Gibson USA Srl	9.0	7
7	Meinl Percussion GmbH	9.3	18
8	Arturia Italy Srl	9.3	4
Total rows: 10 Query complete 00:00:00.041			

- QUERY 4: restituire in ordine decrescente i clienti (id di ogni singolo cliente, la sua email e la somma dei totali di tutti i suoi ordini registrati) che hanno speso più di €5000 nei vari ordini

```
SELECT c.ID, c.Email, SUM(o.TotaleOrdine) AS SpesaTotale  
FROM Cliente c  
JOIN Ordine o ON c.ID = o.Cliente  
GROUP BY c.ID, c.Email  
HAVING SUM(o.TotaleOrdine) > 5000  
ORDER BY SpesaTotale DESC;
```

Di seguito un estratto dell'output della query:

	id [PK] character varying (20)	email character varying (100)	spesatotale numeric
1	HTN19765382	contatti@rockfoundation.it	12199.98
2	XQJ76123459	luca.martini@email.it	11999.98
3	TKB31469852	veronica.bianchi@email.it	7999.99
4	GYQ39182574	stefano.gentile@email.it	7999.98
5	MRF29384756	marco.rossi@email.it	6799.96
6	BPW13629587	francesca.monti@email.it	6299.97
7	JQW76539284	ilaria.colombo@email.it	6049.96
8	BND89276134	federica.colombo@email...	5699.98
Total rows: 9 Query complete 00:00:00.044			

- **QUERY 5: restituire i clienti che nel 2025 hanno ordinato almeno uno strumento a percussione e uno strumento a corda**

```
CREATE OR REPLACE VIEW OrdiniPerCategoria AS
SELECT o.Cliente, o.NumeroOrdine,
CASE
WHEN sc.Prodotto IS NOT NULL THEN 'Corda'
WHEN sp.Prodotto IS NOT NULL THEN 'Percussione'
END AS Categoria
FROM Ordine o
JOIN DettaglioOrdine d ON o.NumeroOrdine = d.Ordine
JOIN Prodotto p ON d.Prodotto = p.CodiceSKU
LEFT JOIN StrumentoACorda sc ON p.CodiceSKU = sc.Prodotto
LEFT JOIN StrumentoAPercussione sp ON p.CodiceSKU = sp.Prodotto
WHERE EXTRACT(YEAR FROM o.Data) = 2025;

SELECT DISTINCT c.ID, c.Email, c.Telefono
FROM Cliente c
WHERE c.ID IN (
SELECT Cliente
FROM OrdiniPerCategoria
WHERE Categoria = 'Corda'
)
AND c.ID IN (
SELECT Cliente
FROM OrdiniPerCategoria
WHERE Categoria = 'Percussione'
);
```

Di seguito un estratto dell'output della query:

	id [PK] character varying (20)	email character varying (100)	telefono character varying (20)
1	BGT72649318	elena.costa@email.it	3241234636
2	BHC39674125	giulia.riva@email.it	3441234656
3	CPZ82461357	elisa.battaglia@email.it	3881234660
4	DRT59184726	alice.de.santis@email.it	3341234646
5	EFW97632154	contatti@electronicdownbeat.it	0117654277
6	GMW74352961	contatti@sinfonia.it	0512345679
7	ILC35741269	contatti@jazzmania.it	0119876543
8	ILZ28476539	info@symphonystore.it	0298765432
Total rows: 22 Query complete 00:00:00.088			

5.2 Creazione degli indici

Si vuole ottimizzare la **query n.5** la quale filtra gli ordini per:

- cliente specifico → WHERE c.ID IN (...)
- anno 2025 → WHERE EXTRACT(YEAR FROM o.Data) = 2025

Attualmente senza l'ausilio di indici la base di dati deve svolgere numerose operazioni quali:

- scansionare tutte le tuple della tabella **Ordini**
- per ogni ordine estrarre l'anno dalla **Data**
- filtrare manualmente gli ordini del cliente corretto e dell'anno 2025

L'indice proposto in seguito ha quindi il fine di organizzare i dati in ordine crescente per **Cliente** e **Data** di ciascun **ordine** permettendo quindi al database di saltare direttamente agli ordini del 2025 per ogni cliente senza dover scansionare l'intera tabella.

```
CREATE INDEX idx_ordine_cliente_data ON Ordine(Cliente, EXTRACT(YEAR FROM Data));
```

Il tipo di indice proposto è organizzato come un B+ Tree con chiavi composte (default in PostgreSQL), perfetto per range di date e ricerche per cliente.

1. **Primo livello:** Cliente (ID del cliente): i nodi sono ordinati in modo lessicografico. Il B+ Tree salta direttamente al nodo del cliente in **tempo O(log N)**.
2. **Secondo livello:** EXTRACT(YEAR FROM Data) (anno dell'ordine): sotto ogni Cliente, gli anni sono ordinati in modo crescente. Dopo aver trovato il cliente, l'indice **scorre solo gli anni rilevanti** (2025). Dopo aver trovato il primo ordine del 2025, il database legge gli altri in **O(1)**.

Questo indice si adatta perfettamente alla query in quanto la complessità passa da $O(N)$ (caso senza indice) a $O(\log N)$ (caso con indice).

6. Applicazione Software

6.1 Compilazione

Comandi per compilare da terminale:

- Su macOS:
**gcc thomann_software.c -o thomann **
**-Idependencies/include **
**-Ldependencies/lib **
-lpq
Poi: **./thomann**
- Su Linux:
**gcc thomann_software.c -o thomann **
**-Idependencies/include **
**-Ldependencies/lib **
-lpq
Poi: **./thomann**
- Su Windows:
**gcc thomann_software.c -o thomann.exe **
**-Idependencies/include **
**-Ldependencies/lib **
-lpq
Poi: **thomann.exe**

6.2 Spiegazione

Il file `thomann_software.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 menù con la seguente lista di interrogazioni disponibili:

1. Prodotti con ≥ 5 recensioni a 5 stelle
2. Totale vendite per categoria di strumento
3. Fornitori più affidabili per consegne
4. Clienti con spesa $> 5000\text{€}$
5. Clienti che hanno ordinato almeno uno strumento a percussione e uno a corda nel 2025
0. Esci

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

Le query 1, 4 e 5 sono state rese parametriche, questo implica che il programma richiede l'inserimento di alcuni dati sui quali si basano le query.

6.3 Codice

All'interno del main vengono richiamate le seguenti funzioni:

- **connect_db()** si connette al database di PostgreSQL, usa i dati di accesso e prova a stabilire la connessione, in caso di errore viene stampato un messaggio e chiuso il programma

- **print_result(PGresult* res)** stampa a schermo i risultati di una query mostrando prima i nomi delle colonne e poi ogni riga con i valori trovati

- **execute_query(Pgconn* conn, int choice)** esegue una query diversa in base al numero scelto dall'utente; dopo la scelta del numero da parte dell'utente viene costruito il comando SQL corrispondente ed in seguito viene inviato al database. Se la query funziona, mostra i risultati con **print_result()**, se c'è un errore stampa un messaggio e non fa nulla.