

# Relazione di Basi di Dati

## Palestre QueryFlex

---

Università degli Studi di Padova  
Dipartimento di Matematica  
Laurea triennale in Informatica

Autori:

Riccardo Zaupa - 2034303

Sebastiano Lewental - 2043683

---

### Abstract

QueryFlex è una grande catena di palestre distribuite sul territorio italiano che ha bisogno di un database organizzato per permettere il supporto di tutti i suoi servizi.

QueryFlex comprende molte sedi, ognuna delle quali offre l'abbonamento alla propria **Area Fitness** completa di tutti gli attrezzi e a molti **Corsi** per tutte le necessità e livelli di allenamento.

Iscrivendosi è possibile accedere in qualunque palestra della catena, ideale per coloro che lavorano spostandosi; entrando viene controllata la data di iscrizione così da sapere quando è necessario rinnovarla e viene inoltre visualizzato l'abbonamento posseduto in modo da consentire l'accesso alle aree di allenamento.

Ogni iscritto all'Area Fitness ha diritto ad una o più schede personalizzate con un istruttore dedicato che lo aiuterà a fissare e raggiungere i propri obiettivi, la scheda digitale riporta gli esercizi da fare con una descrizione dei muscoli coinvolti, è possibile cambiare scheda all'occorrenza, le schede passate restano memorizzate così da tracciare i propri progressi e miglioramenti.

Le palestre istituiscono vari corsi durante l'anno, con istruttori professionali e preparati; gli iscritti possono abbonarsi e frequentare i corsi a cui sono interessati.

Grazie al database, QueryFlex sarà in grado di analizzare i propri servizi: osservando gli esercizi più adottati e i corsi più frequentati così da acquistare gli attrezzi per gli esercizi più richiesti e riproporre nelle altre sedi i corsi di successo, permettendo costante miglioramento e garanzia di un'ottima esperienza per i propri clienti e potendo così primeggiare sulla concorrenza.

### Analisi Dei Requisiti

#### Descrizione Testuale

Il database contiene i dati di tutti i **Clienti** che sono stati iscritti alla catena di palestre:

- Codice fiscale
- Nome
- Cognome
- Email
- Data di nascita
- Sesso

Ogni cliente ha attualmente una o più **Iscrizioni**, ogni iscrizione ha validità un'anno intero da quando viene effettuata, è possibile accedere alle palestre con un'iscrizione valida, è utile memorizzare:

- Scadenza

- Prezzo

I clienti possiedono anche almeno un **Abbonamento**, che può essere ad un **Corso** di una palestra o all'**Area Fitness**, ogni abbonamento ha:

- Prezzo
- Durata

Inoltre un Corso deve avere le seguenti informazioni:

- Nome
- Edizione
- Descrizione
- Orari
- Istruttore che lo conduce
- Numero massimo di partecipanti
- Palestra in cui è svolto

Mentre i prezzi dei corsi sono istituiti dalle singole palestre, i prezzi delle iscrizioni e degli abbonamenti all'Area Fitness sono gli stessi nella catena, e possono cambiare di anno in anno, perciò è necessario un **Listino Prezzi** che definisce:

- Prezzo iscrizione
- Prezzo mensile Area Fitness

La catena comprende molte Palestre in varie città, in ogni città ci possono anche essere più palestre, le palestre hanno queste caratteristiche:

- Indirizzo
- Giorni e orari di apertura
- Corsi che si svolgono in essa
- Istruttori affiliati ad essa

Nelle palestre, gli **Istruttori** tengono corsi e gestiscono le schede degli iscritti che le richiedono, i dati riguardanti gli istruttori sono:

- Codice fiscale
- Nome
- Cognome
- Data di nascita
- Sesso
- Palestra in cui lavora
- Corsi che conduce

Le **Schede** spesso hanno una certa durata nell'anno, perciò si salva:

- Durata
- Esercizi da svolgere
- Istruttore assegnato

Gli **Esercizi** che si svolgono allenano uno o più gruppi muscolari e il cliente deve sapere come si svolgono, perciò viene memorizzato:

- Descrizione
- Gruppi muscolari coinvolti

I **Gruppi Muscolari** appartengono ad una parte del corpo, perciò viene memorizzata anche una categoria oltre al loro nome identificativo.

## Glossario Dei Termini

Termine	Descrizione	Collegamenti
Cliente	Persona che ha almeno un'iscrizione alla catena di palestre	Iscrizione, area fitness, scheda, corso
Catena di Palestre	Azienda che possiede molte palestre in varie sedi	-
Iscrizione	Inclusione alle attività delle palestre per un anno	Cliente, listino prezzi
Scadenza	Data in cui si smette di essere iscritti alla palestra, l'iscrizione ha validità per tutto l'anno precedente alla scadenza	Attributo di Iscrizione
Abbonamento	Permesso temporaneo a determinati servizi della palestra	Cliente
AreaFitness	Abbonamento alle classiche aree della palestra con tutti gli strumenti	Figlio di Abbonamento. Cliente, listino prezzi, scheda
Corso	Attività svolta in palestra che pratica una disciplina	Figlio di Abbonamento. Cliente, palestra, istruttore
Edizione	Numero che indica l'iterazione di svolgimento di un corso	Attributo di Corso
Istruttore	Persona che lavora in una palestra, svolge corsi e aiuta gli iscritti con schede e esercizi	Palestra, corso, scheda
Scheda	Raccolte di esercizi da svolgere nell'Area Fitness	Area Fitness, istruttore
Gruppo Muscolare	Muscolo o insieme di muscoli che vengono sollecitati e allenati da esercizi fisici	Esercizio
Categoria	Rappresenta un insieme di gruppi muscolari correlati	Attributo di Gruppo Muscolare

## Operazioni

Operazione	Tipo	Frequenza
Controllo validità iscrizione più recente (scadenza > oggi)	L	1000 al giorno
Aggiunta nuovo cliente	S	50 al mese
Aggiunta nuova iscrizione	S	100 al mese
Abbonamento ad Area Fitness	S	250 al mese
Visualizzazione corsi disponibili in una palestra tra due date	L	2000 al giorno
Abbonamento a Corso	S	250 al mese
Registrazione nuovo corso	S	1 al mese

Visualizzazione scheda	L	3000 al giorno
Registrazione nuova scheda	S	10 al giorno
Calcolo guadagno mensile per palestra	L	1 al mese

## Progettazione Concettuale

### Lista entità

- Palestra:
    - Indirizzo: attributo composto
      - Città: varchar(50) primary key
      - Indirizzo: varchar(50) primary key
    - Giorni di apertura: attributo composto
      - Apertura Feriali: time NOT NULL
      - Chiusura Feriali: time NOT NULL
      - Apertura Festivi: time NOT NULL
      - Chiusura Festivi: time NOT NULL
  - ListinoPrezzi:
    - ID: int primary key
    - PrezzoIscrizione: real NOT NULL
    - PrezzoAreaFitnessMensile: real NOT NULL
  - Cliente:
    - Cf: varchar(16) primary key
    - Nome: varchar(20) NOT NULL
    - Cognome: varchar(20) NOT NULL
    - Mail: varchar(50) NOT NULL
    - DataNascita: date NOT NULL
    - Sesso genere NOT NULL
  - Iscrizione:
    - ID: int primary key
    - Scadenza date NOT NULL
  - Abbonamento:
    - ID: int primary key
    - Prezzo: real NOT NULL
    - Durata: attributo composto
      - DataInizio: date NOT NULL
      - DataFine: date NOT NULL
- Abbonamento si specializza in due sottocategorie con una generalizzazione totale:
- Corso:
    - Nome: varchar(50) primary key
    - Edizione: int primary key
    - Indirizzo Palestra: attributo composto primary key
    - Descrizione: varchar(150) NOT NULL
    - Orario: varchar(50) NOT NULL
    - Max partecipanti: int NOT NULL
  - AreaFitness

- Istruttore:
  - CF: varchar(16) primary key
  - Nome: varchar(20) NOT NULL
  - Cognome: varchar(20) NOT NULL
  - DataNascita: date NOT NULL
  - Sesso: genere NOT NULL
- Scheda:
  - ID: int primary key
  - Durata: attributo composto
    - DataInizio: date NOT NULL
    - DataFine: date NOT NULL
- Esercizio:
  - Nome: varchar(50) primary key
  - Descrizione: varchar(200) NOT NULL
- GruppoMuscolare:
  - Nome: varchar(50) primary key
  - Categoria: varchar(20) NOT NULL

## Tabella delle relazioni

Relazione	Entità coinvolte	Descrizione
Possessione	Cliente (1,N) Abbonamento (1,N)	Un cliente può possedere più piani di abbonamenti e uno specifico abbonamento può essere posseduto da più clienti.
Possessione	Cliente (1,N) Iscrizione (1,1)	Un cliente può possedere più iscrizioni a QueryFlex in momenti differenti ma un'iscrizione specifica può essere posseduta da un solo cliente.
Riferimento	Iscrizione(1,1) ListinoPrezzi(0,N)	Un'iscrizione specifica può ricavare il prezzo da un solo listino mentre un listino prezzi può avere più riferimenti a più iscrizioni.
Riferimento	ListinoPrezzi(0,N) Areafitness (1,1)	Un listino prezzi può avere più riferimenti a più abbonamenti di tipo area fitness mentre un unico abbonamento area fitness si riferisce ad un unico listino prezzi.
Comprensione	Areafitness (1,N) Scheda (1,1)	Un abbonamento area fitness può comprendere da una o più schede di allenamento mentre un'unica scheda di allenamento può riferirsi ad un unico abbonamento area fitness.
Svolgimento	Corso (1,1) Palestra (0,N)	Un corso di una certa edizione può svolgersi in una sola palestra mentre in una palestra possono svolgersi più corsi.
Affiliazione	Palestra (1,N) Istruttore (1,1)	Una palestra può assumere più istruttori mentre un istruttore può lavorare in una sola palestra.
Assegnazione	Istruttore(0,N) Scheda (1,1)	Ad un istruttore possono essere assegnate più schede mentre una scheda può essere assegnata ad un solo istruttore.
Composizione	Esercizio (0,N) Scheda (1,N)	Un esercizio può essere inserito in più schede mentre una scheda contiene più esercizi.
Allenamento	Esercizio (1,N) GruppoMuscolare(0,N)	Un esercizio può allenare molti gruppi muscolari mentre un singolo gruppo muscolare può essere allenato da più esercizi

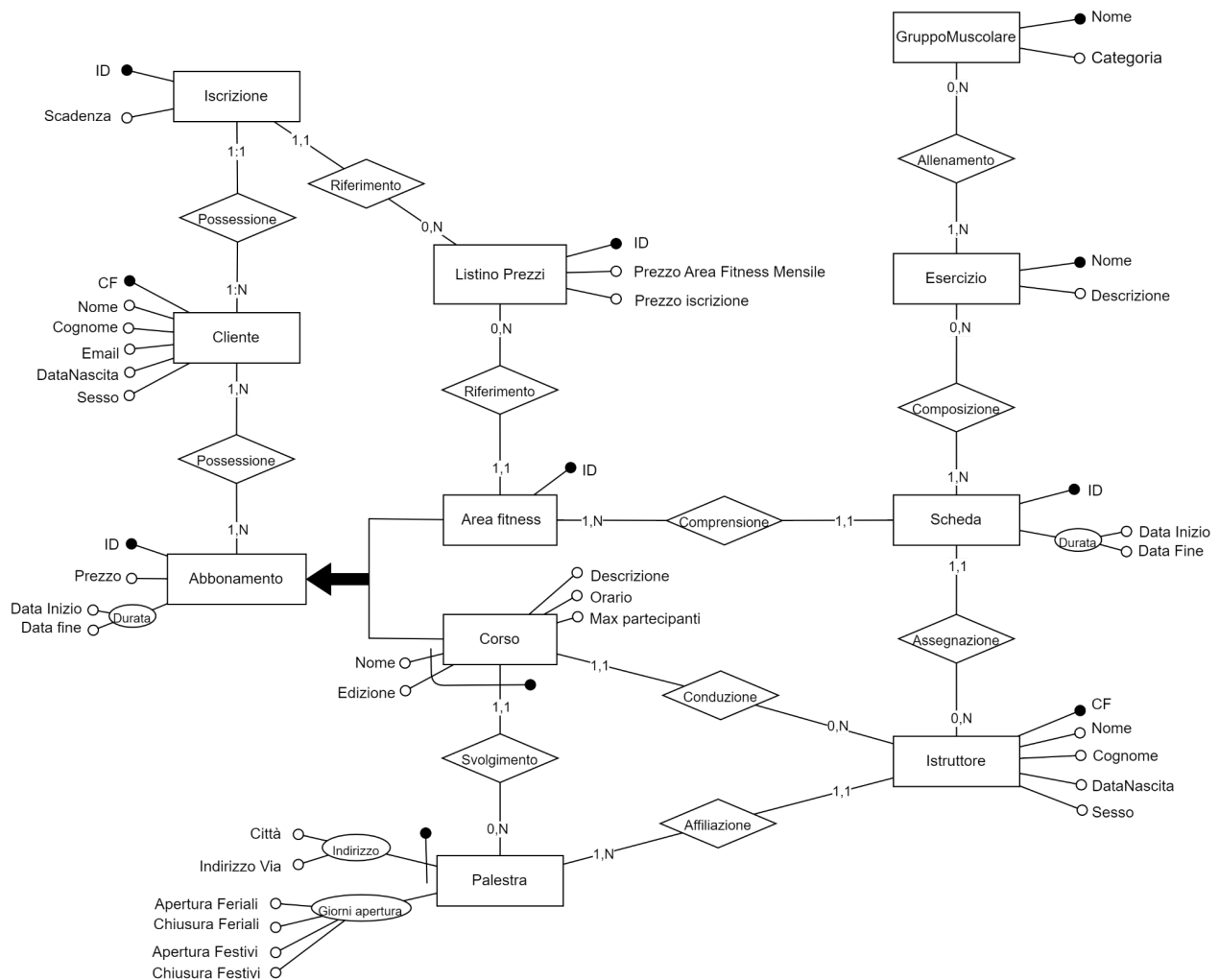
## Vincoli non rappresentabili tramite schema E-R

- Un cliente è iscritto all'area fitness o ai corsi se possiede un'iscrizione attiva nel momento di svolgimento del corso e/o nel periodo di abbonamento all'area fitness.
- Il servizio scheda è attivo se è presente un abbonamento all'area fitness che lo permetta.
- La data di inizio nelle entità Areafitness, Scheda e Corso è minore della rispettiva data di fine
- La palestra in cui si svolge un corso è la stessa affiliata all'istruttore che lo tiene
- La data di scadenza di un'iscrizione di un cliente deve essere almeno a distanza di un anno dalla data di scadenza dell'iscrizione precedente
- Il listino che viene scelto per dare il prezzo a Iscrizione e AreaFitness è l'ultimo in ordine di ID

## Vincoli di derivazione

- Il prezzo di un Abbonamento ad Area fitness deve essere uguale al prezzo mensile sul Listino al momento dell'abbonamento relativo all'area fitness moltiplicato per i mesi di durata di abbonamento

## Schema concettuale



## Progettazione Logica

### Ristrutturazione

#### 1. Analisi delle ridondanze

Dall'analisi effettuata non risultano ridondanze sui dati.

#### 2. Eliminazione generalizzazioni

Abbonamento ← Area fitness, Corso:

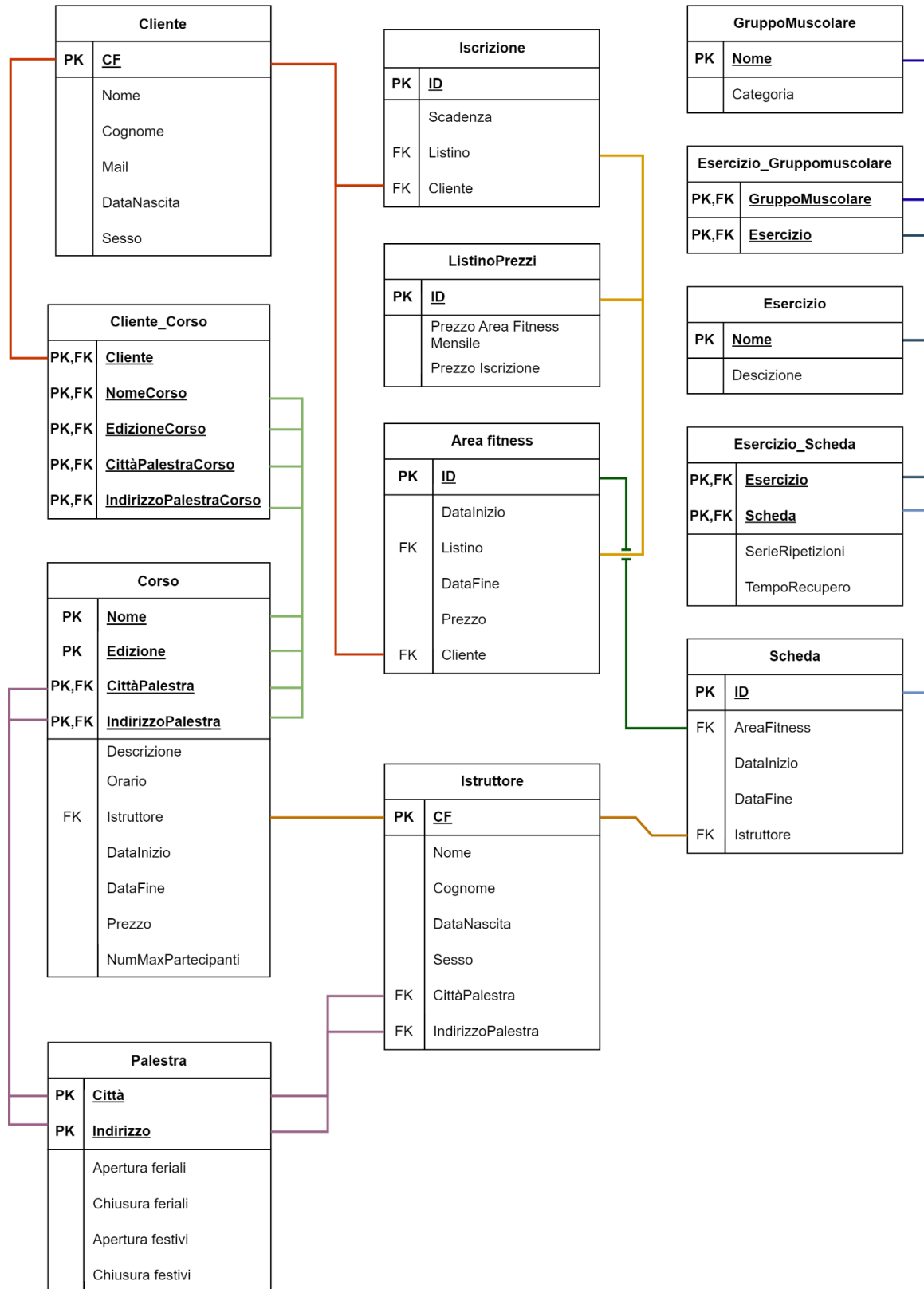
- Le entità Area fitness e Corso sono effettivamente due entità distinte con delle relazioni che le contraddistinguono. Per i corsi è necessario specificare la palestra dove vengono svolti e anche gli istruttori che li conducono.

Mentre per come è strutturato il suo abbonamento, Area fitness deve riferirsi a Listino Prezzi per permettere di calcolare il suo prezzo totale oltre al fatto che comprende la possibilità di ottenere una scheda personalizzata di allenamento. Si decide quindi di mantenere le entità separate.

#### 3. Scelta identificatori primari

Nelle tabelle Iscrizione, ListinoPrezzi, AreaFitness, Scheda viene usato come identificatore primario un ID per semplicità. Nella tabella corso vengono utilizzati come primary key gli attributi Nome, Edizione, CittàPalestra, IndirizzoPalestra, una soluzione più semplice sarebbe stata quella di un ID, ma si sarebbe perso il significato che questi attributi comunicano.

## Schema e-r ristrutturato





## Creazione delle tabelle

A -> B indica che B è chiave esterna di A

**Cliente** (Cf, Nome, Cognome, Mail, DataNascita, Sesso)

**Palestra** (Città, Indirizzo, Aperturaferiali, Chiusuraferiali, Aperturafestivi, Chiusurafestivi)

**Cliente\_corso** (Cliente -> Cliente.Cf, Nomecorso -> Corso.Nome, Edizionecorso -> Corso.Edizione, Cittapalestracorso -> Corso.Cittapalestra, Indirizzopalestracorso -> Corso.IndirizzoPalestra)

**Corso** (Nome, Edizione, Cittapalestra -> Palestra.Città, Indirizzopalestra -> Palestra.Indirizzo, Descrizione, Orario, Istruttore, Datainizio, DataFine, Prezzo, Numeromaxpartecipanti)

**Listinoprezzi** (Id, Prezzoareafitnessmensile, Prezzoiscrizione)

**Iscrizione** (Id, Scadenza, Listino -> Listinoprezzi.Id, Cliente -> Cliente.Id)

**Istruttore** (Cf, Nome, Cognome, DataNascita, Sesso, Cittapalestra -> Palestra.Città, IndirizzoPalestra -> Palestra.Indirizzo)

**Areafitness** (Id, Datainizio, Listino -> Listinoprezzi.Id, Datafine, Prezzo, Cliente -> Cliente.cf)

**Scheda** (Id, Areafitness -> Areafitness.Id, Datainizio, Datafine, Istruttore -> Istruttore.cf)

**Esercizio** (Nome, Descrizione)

**Esercizio\_scheda** (Esercizio -> Esercizio.Nome, Scheda -> Scheda.Id, SerieRipetizioni, TempoRecupero)

**Gruppomuscolare** (Nome, Categoria)

**Esercizio\_gruppomuscolare** (Gruppomuscolare -> Gruppomuscolare.nome, Esercizio -> Esercizio.Nome)

## Query e Indici

### Query

1. Trova tutti i nomi degli esercizi che allenano un categoria di muscoli a scelta

```
SELECT es.nome
FROM esercizio es
  JOIN esercizio_gruppomuscolare eg ON es.nome = eg.esercizio
  JOIN gruppomuscolare gm ON gm.nome = eg.gruppomuscolare
WHERE categoria = 'Braccia';
```

```
+-----+
| nome |
+-----+
| Dip alle parallele |
| Trazioni presa prona |
| Shoulder press |
| Curl in piedi cavo basso con sbarra |
| French press bilanciere panca piana |
| Curl 2 manubri in piedi |
| Lat pulldown |
| Bench press |
| Hammer curl |
| Military press |
| Bent-over rows |
| Flessioni tricipiti |
| Estensioni per tricipiti |
| Curl martello |
| Curl concentrato |
| Tritonatori tricipiti |
| Estensioni tricipiti al cavo |
| Estensioni per tricipiti al cavo alto |
| Curl con manubri in supinazione |
| French press con manubri |
+-----+
```

2. Ordina i corsi che hanno avuto maggiori frequentazioni tra tutte le edizioni segnalando il numero di partecipanti, distinguendo per palestra e riportando il guadagno ricavato

```
SELECT c.nome, c.cittapalestra, c.indirizzopalestra, count(cc.cliente) AS tot_partecipanti, sum(c.prezzo) AS ricavi_corso
FROM corso AS c
  JOIN cliente_corso AS cc
  ON c.nome = cc.nomecorso
  AND c.edizione = cc.edizionecorso
  AND cc.cittapalestracorso = c.cittapalestra
  AND cc.indirizzopalestracorso = c.indirizzopalestra
GROUP BY c.nome, c.cittapalestra, c.indirizzopalestra
ORDER BY tot_partecipanti DESC;
```

nome	cittapalestra	indirizzopalestra	tot_partecipanti	ricavi_corso
Functional Training	Varese	Piazza Garibaldi, 22	12	379.88
Zumba	Varese	Piazza Garibaldi, 22	11	299.89
Spinning	Padova	Via Luzzatti 250	8	176
Zumba	Venezia	Località Martellago 130	6	167.94
Zumba	Pescara	Via Palmiro Togliatti	6	150
CrossFit	Verona	Via dell'Artigliere, 8	6	215
Total Body	Pescara	Via Palmiro Togliatti	6	150
Calisthenics	Verona	Via dell'Artigliere, 8	5	150
Aqua Fitness	Venezia	Località Martellago 130	5	110
Spinning	Varese	Piazza Garibaldi, 22	4	75.96
Pilates	Verona	Via dell'Artigliere, 8	4	80
Boxe Fit	Padova	Via Luzzatti 250	4	159.96
Functional Training	Padova	Via Luzzatti 250	4	120
Pilates	Varese	Piazza Garibaldi, 22	4	79.96
Calisthenics	Varese	Piazza Garibaldi, 22	4	180
Power Yoga	Pescara	Via Palmiro Togliatti	2	50
Ginnastica Posturale	Pescara	Via Palmiro Togliatti	2	60
Yoga	Varese	Piazza Garibaldi, 22	2	44
Zumba	Padova	Via Luzzatti 250	2	50
Stretching e Mobilità	Pescara	Via Palmiro Togliatti	2	32
Circuit Training	Venezia	Località Martellago 130	2	80
Yoga	Venezia	Località Martellago 130	1	25
Zumba	Verona	Via dell'Artigliere, 8	1	25

3. Trova e ordina per tempo di iscrizione all'area fitness i clienti riportando il tempo di iscrizione e soldi spesi nell'area fitness

```
SELECT c.nome, c.cognome, SUM(AGE(DataFine,DataInizio)) AS tempo_iscrizione, SUM(af.prezzo) AS tot_ricavi
FROM cliente AS c
  JOIN areafitness AS af ON af.cliente = c.cf
GROUP BY c.nome, c.cognome
ORDER BY tempo_iscrizione DESC;
```

nome	cognome	tempo_iscrizione	tot_ricavi
Pedro	Scala	3 years	1080
Alessia	Berti	2 years	660
Caterina	Santoro	1 year 6 mons	510
Alberto	Bianchi	1 year 6 mons	510
Elisa	Carullo	1 year	360
Francesco	Crescenzo	1 year	360
Giorgia	Tonolini	1 year	300
Giulio	Namari	1 year	420
Fabrizio	Colombo	1 year	360
Francesca	La Rosa	9 mons	285
Valentina	Caputo	9 mons	315
Giovanni	Capodanno	6 mons	180
Mario	Rossi	3 mons	105
Marco	Napolitano	3 mons	105
Miriam	Liberti	3 mons	105
Marta	Rosso	3 mons	105

## 4. Mostra in ordine i clienti che attualmente allenano piu' volte una categoria di muscoli a scelta

```

SELECT c.nome, c.cognome, COUNT(c.cf) AS num_volte
FROM cliente AS c
  JOIN areafitness AS af ON af.cliente = c.cf
  JOIN scheda AS s ON s.areafitness = af.id
  JOIN esercizio_scheda AS es ON es.scheda = s.id JOIN esercizio AS e ON e.nome = es.esercizio
  JOIN esercizio_gruppomuscolare AS eg ON eg.esercizio = e.nome
  JOIN gruppomuscolare AS gm ON gm.nome = eg.gruppomuscolare
  JOIN (SELECT c.cf AS cliente_last_scheda, MAX(s.DataFine) AS data_last_scheda
        FROM cliente AS c JOIN areafitness AS af ON af.cliente = c.cf
        JOIN scheda AS s ON s.areafitness = af.id
        GROUP BY c.cf) AS last_scheda
  ON c.cf = last_scheda.cliente_last_scheda AND s.DataFine = last_scheda.data_last_scheda
WHERE gm.categoria = 'Spalle'
GROUP BY c.cf ORDER BY num_volte DESC;

```

nome	cognome	num_volte
Valentina	Caputo	4
Elisa	Carullo	3
Fabrizio	Colombo	3
Pedro	Scala	3
Giorgia	Tonali	3
Miriam	Liberti	2
Marco	Napolitano	2
Giovanni	Capodanno	2
Giulio	Namari	1
Alessia	Berti	1
Caterina	Santoro	1
Francesca	La Rosa	1
Francesco	Crescenzo	1
Alberto	Bianchi	1
Mario	Rossi	1

## 5. Mostra in ordine le palestre che guadagnano di piu' dai corsi in un anno a scelta

```

SELECT guadagno_corsi.cittapalestra, guadagno_corsi.indirizzopalestra, sum(guadagno_corso) AS guadagno_tot
FROM (SELECT c.nome, c.edizione, c.cittapalestra, c.indirizzopalestra, c.datainizio, count(cc.cliente)*c.prezzo AS guadagno_corso
      FROM corso c
        JOIN cliente_corso cc ON cc.cittapalestracorso = c.cittapalestra
                                AND cc.edizionecorso = c.edizione
                                AND cc.indirizzopalestracorso = c.indirizzopalestra
                                AND cc.nomecorso = c.nome
      GROUP BY c.nome, c.edizione, c.cittapalestra, c.indirizzopalestra) AS guadagno_corsi
GROUP BY guadagno_corsi.cittapalestra, guadagno_corsi.indirizzopalestra, EXTRACT(YEAR FROM guadagno_corsi.datainizio)
HAVING EXTRACT(YEAR FROM guadagno_corsi.datainizio)=2022
ORDER BY guadagno_tot DESC;

```

cittapalestra	indirizzopalestra	guadagno_tot
Varese	Piazza Garibaldi, 22	354.86999702453613
Pescara	Via Palmiro Togliatti	342
Padova	Via Luzzatti 250	313.9600067138672
Verona	Via dell'Artigliere, 8	204
Venezia	Località Martellago 130	140.97999954223633

## 6. Mostra in ordine i clienti che hanno speso di piu' in totale riportando l'importo

```

SELECT c.nome, c.cognome, iscrizione_tot + COALESCE(areafitness_tot, 0) + COALESCE(corso_tot, 0) AS pagamento_tot
FROM cliente c
  JOIN (SELECT c.cf, c.nome, c.cognome, SUM(lp.prezzoiscrizione) AS iscrizione_tot
        FROM cliente c
        JOIN iscrizione i      ON i.cliente = c.cf
        JOIN listinoprezzi lp  ON lp.id = i.listino
        GROUP BY c.cf) AS iscr
  ON c.cf=iscr.cf
  LEFT JOIN (SELECT c.cf, c.nome, c.cognome, SUM(a.prezzo) AS areafitness_tot
             FROM cliente c
             JOIN areafitness a ON a.cliente = c.cf
             GROUP BY c.cf) AS ar
  ON iscr.cf=ar.cf
  LEFT JOIN (SELECT cl.cf, cl.nome, cl.cognome, sum(c.prezzo) AS corso_tot
             FROM cliente cl
             JOIN cliente_corso cc ON cc.cliente = cl.cf
             JOIN corso c          ON c.indirizzopalestra = cc.indirizzopalestracorso
                                AND c.cittapalestra = cc.cittapalestracorso
                                AND c.nome = cc.nomecorso
                                AND c.edizione = cc.edizionecorso
             GROUP BY cl.cf) AS co
  ON iscr.cf=co.cf
ORDER BY pagamento_tot DESC;

```

nome	cognome	pagamento_tot
Pedro	Scala	1210
Alberto	Bianchi	880.49
Alessia	Berti	856
Caterina	Santoro	779.95
Fabrizio	Colombo	486.99
Francesco	Crescenzo	475.97998
Giulio	Namari	470
Luca	Barbara	437.90002
Giorgia	Tonalini	431.5
Valentina	Caputo	415
Elisa	Carullo	400
Francesca	La Rosa	335
Fabio	Cocci	330.98
Mariella	Mannino	285
Nicole	Marchesi	257.93
Gianluca	Marconi	240.98
Patrizia	Rosselli	236
Giovanni	Capodanno	220
Romina	Talenti	218.99
Simone	Marchetti	206
Alberto	Borgese	169.96
Mario	Rossi	155
Marco	Napolitano	155
Miriam	Liberti	155
Marta	Rosso	155
Francesco	Camilli	139.98
Carmen	Monaldi	136
Giuliana	Nicoletti	132.98999
Elena	Carotti	116.979996
Veronica	Carli	102.99
Ogan	Valverde	102
Elodia	Raviolo	75

7. Mostra in ordine gli istruttori che si occupano attualmente di piu' clienti nell'area fitness con relativo numero di clienti

```
SELECT i.nome,i.cognome,count(af.cliente) AS clienti
FROM istruttore i
  JOIN scheda s ON s.istruttore = i.cf
  JOIN areafitness af ON af.id = s.areafitness
  JOIN (SELECT c.cf AS cliente_last_scheda, MAX(s.DataFine) AS data_last_scheda
        FROM cliente AS c
        JOIN areafitness AS af ON af.cliente = c.cf
        JOIN scheda AS s ON s.areafitness = af.id
        GROUP BY c.cf) AS last_scheda
  ON af.cliente = last_scheda.cliente_last_scheda AND s.DataFine = last_scheda.data_last_scheda
WHERE s.DataFine>CURRENT_DATE
GROUP BY i.cf
ORDER BY clienti DESC;
```

nome	cognome	clienti
Francesca	La Rosa	2
Alberto	Bianchi	1
Alessandro	Berti	1
Marco	Napolitano	1
Roberto	Rossi	1
Valeria	Caputo	1

## Indici

All'ingresso di un cliente nella palestra viene effettuato il controllo della validità dell'iscrizione che deve essere immediato, specialmente nelle ore di punta; in un caso su larga scala, avendo un gran numero di clienti è necessario rendere questa operazione più fulminea possibile, perciò è ideale indicizzare gli attributi Cliente e Scadenza in indice\_iscrizione.

```
create index indice_iscrizione on Iscrizione(Cliente,Scadenza)
```

## Codice C++

### Descrizione dell'utilizzo del codice

Il codice C++ per eseguire, attraverso l'accesso al DB PostgreSQL, e visualizzare le query presentate in precedenza si struttura in un singolo file c++ che viene compilato con la seguente comando:

```
g++ -static-libstdc++ query.cpp -L dependencies\lib -lpq -o query.
```

Prima di eseguire questo comando è necessario inserire nella cartella "dependencies/include" i file libpq-fe.h, pg\_consigt\_ext.h e postgres\_ext.h e nella cartella "dependencies/lib" i file libpq.dll e libpq.lib tenendo presente di posizionare la cartella dependencies nella stessa directory contenente il file .cpp da eseguire.

Per far partire il programma basta eseguire il file query.exe. Una volta partito il programma chiederà nome del database in cui si vuole accedere (dipende da come l'utente ha scelto di chiamarlo) e password associata.

Se la connessione è andata buon fine verrà mostrato un menu di opzioni contenente una lista delle query da poter eseguire, numerate da 1 a 7. Se si vuole far terminare il programma basta digitare '0' (facendo comparire a schermo la stringa "Fine programma") altrimenti si può inserire il numero della query scelta. Alcune query come la 1,4 e la 5 richiedono da parte dell'utente l'inserimento di alcuni parametri:

1. Viene mostrata la lista delle possibili categorie di muscoli da inserire per ottenere gli esercizi ad essa associati. Nota: categorie deve essere riportate dall'utente così come sono state stampate a schermo altrimenti potrebbero esserci errori.
4. Viene mostrata la lista delle possibili categorie di muscoli da inserire per ottenere la lista dei clienti che allenano gli esercizi ad esse associati. Nota: categorie deve essere riportate dall'utente così come sono state stampate a schermo altrimenti potrebbero esserci errori
6. Viene mostrata la lista degli anni che si possono inserire per mostrare i relativi guadagni per ogni palestra. Nota: altri anni inseriti non saranno validi perché potranno essere futuri all'anno corrente o antecedenti all'apertura della catena QueryFlex.

## Documentazione del codice

Nel codice sono state utilizzate le seguenti funzioni:

```
PGconn *connect(const char *host, const char *user, const char *db, const char *pass, const char *port);
```

La funzione "connect" permette di stabilire una connessione al database PostgreSQL tramite la libreria di connessione 'libpq'. Una volta stabilita la connessione, la funzione controlla se la connessione è stata effettuata correttamente o se si sono verificati errori tramite l'implementazione della funzione "checkConnection". Il valore restituito è una connessione attiva pronta per eseguire query ed interagire con il database stesso.

```
PGresult *executePQ(PGconn *conn, const char *query);
```

La funzione "executePQ" permette di eseguire una query sul database PostgreSQL tramite una connessione esistente. Il valore ritornato conterrà le informazioni restituite dalla query come le righe selezionate o lo stato dell'operazione non prima però di aver effettuato un controllo tramite la funzione "checkResults".

```
void checkConnection(PGconn *conn);
```

La funzione "checkConnection" si occupa di verificare lo stato di una connessione al database PostgreSQL e di gestire eventuali errori di connessione terminando il programma in caso di errore o stampando un messaggio di successo se la connessione è stata stabilita correttamente.

```
void checkResults(PGresult *res, const PGconn *conn);
```

La funzione "checkResults" è responsabile di verificare lo stato di un risultato di query ottenuto dalla libreria 'libpq' e di gestire eventuali errori relativi ai risultati.

```
void printResults(PGconn *conn, PGresult *res);
```

La funzione "printResults" ha lo scopo di stampare i risultati di una query PostgreSQL in formato tabellare con un aspetto formattato e ordinato creando un output facilmente leggibile.

La funzione "printLine" ha lo scopo di stampare una riga di separazione in uno stile di cornice orizzontale.

```
void printLine(const vector<int> &maxChar);
```

E' utilizzata nella funzione "printResult" per formattare output tabellari.

Progetto a cura di

Riccardo Zaupa - 2034303

Sebastiano Lewental - 2043683