

Progetto del corso Basi di Dati [SCP4065533]

Base di Dati per la gestione di una Palestra

Thomas Morettin [2111001]
Felician Mario Necsulescu [2111935]

1 Abstract

Il progetto si concentra sull'organizzazione di una base di dati per la gestione completa e dettagliata di una realtà sportiva, come una palestra. Il contesto è dunque quello di una moderna struttura sportiva che offre vari corsi svolti in diverse sale, accesso libero agli spazi, supporto di istruttori tramite i corsi e attrezzature per l'allenamento. Il modello si concentra in particolare sulla gestione dei clienti (con certificato medico valido), delle loro iscrizioni e dei relativi pagamenti, tracciando l'accesso alla struttura tramite check-in e le loro prenotazioni dei corsi. La base di dati quindi registra, tra le informazioni, gli accessi degli iscritti alla palestra. Ogni accesso viene rilevato al momento della scansione del codice QR da parte dell'iscritto e viene tracciato e memorizzato. In questo modo è possibile conservare una cronologia strutturata degli ingressi, utile sia a fini gestionali che statistici. Ogni corso è organizzato in lezioni distinte, assegnate a specifiche sale e condotte da istruttori dedicati, i quali devono essere in possesso di una certificazione sportiva valida. Le sale ospitano varie attrezzature, le quali possono essere soggette ad eventuali manutenzioni.

Il sistema è quindi progettato per supportare le operazioni quotidiane di una sede fitness e fornire una base solida per rendere possibile una consultazione chiara e una supervisione ottimizzata, offrendo quindi una migliore esperienza per gli iscritti e una panoramica più ordinata per i responsabili gestionali e operativi.

2 Analisi dei requisiti

Persone. Ogni persona all'interno della struttura è identificata mediante il codice fiscale e dispone delle seguenti informazioni associate:

- **Codice fiscale.**
- **Nome.**
- **Cognome.**
- **Email.**
- **Telefono.**

Queste persone possono distinguersi in: **clienti** e **istruttori**.

Istruttore. Oltre le informazioni personali di base, per gli istruttori viene tracciato un ulteriore dato:

- **Stipendio** (mensile).

Inoltre per ciascun istruttore vengono salvate le **certificazioni sportive** conseguite dal medesimo, contenenti i seguenti dettagli:

- **L'ente certificatore** che è responsabile del rilascio della abilitazione professionale.
- **Specializzazione**, che specifica l'ambito della certificazione.
- **Scadenza** della certificazione.

Cliente. Ogni individuo iscritto è definito, oltre che dalle informazioni principali della persona, anche da:

- **Data di iscrizione** alla palestra.
- **Numero di ingressi** a partire dal momento dell'iscrizione.

Per ogni cliente viene registrato anche un **certificato medico** che comprende:

- **Dottore** che rilascia il documento.
- **Data di rilascio** del documento.
- **Data di scadenza** del documento.

Ogni iscrizione effettuata associa la persona ad un **abbonamento** specifico, costituito dai seguenti attributi:

- **Codice**, che identifica l'abbonamento specifico del cliente.
- **Prezzo** dell'abbonamento.
- **Durata** (giornaliero, mensile, trimestrale, annuale).
- **Accesso libero**, che determina se l'accesso può essere fatto all'intera struttura senza restrizioni.
- **Accesso corsi**, che specifica invece se l'accesso è limitato solo ai corsi organizzati dal centro fitness.

Quindi ogni cliente è associato ad un abbonamento tramite una **iscrizione** che include:

- **Data di inizio** dell'iscrizione.
- **Data di fine** dell'iscrizione.
- Eventuale **sconto**.

Ogni iscrizione ha la possibilità di essere effettuata tramite più di un **pagamento** e quindi non necessariamente uno unico, ognuno di questi viene quindi tracciato per mezzo delle seguenti informazioni:

- **Numero di rata.**
- **Importo.**
- **Data** del pagamento.
- **Metodo** di pagamento.

Il cliente nel corso del tempo può eventualmente richiedere una o più **schede** personalizzate. Si tratta dunque di schede tarate in base al cliente, cioè in relazione delle sue prestazioni e obiettivi, pertanto sono uniche e attribuibili esclusivamente ad un cliente. Queste schede includono:

- **Codice**, per identificare la scheda dalle altre e per eventualmente modificarla in seguito.
- **Tipologia**, che differisce in considerazione di obiettivi e capacità del soggetto.
- **Data di creazione** della scheda.

La composizione di una scheda di allenamento avviene tramite l'inserimento di più esercizi, ognuno dei quali può essere personalizzato per la scheda in questione. Ogni **esercizio nella scheda** avrà i seguenti attributi:

- Numero di **serie** e relative **ripetizioni**.
- **Tempo di recupero**.
- **Note**, cioè eventuali annotazioni o commenti sull'esecuzione dell'esercizio in base alla scheda in cui inserito.

Gli **esercizi**, considerati nella loro definizione generale e indipendentemente dal contesto in cui vengono utilizzati all'interno delle schede, sono caratterizzati dai seguenti attributi:

- **Nome dell'esercizio**, che lo identifica dagli altri.
- **Steps di svolgimento**, cioè la sequenza ordinata di azioni o movimenti che compongono l'esecuzione dell'esercizio.

Attrezzatura. L'attrezzatura della palestra è formata da:

- **Codice**, cioè un identificativo univoco assegnato a ciascun attrezzo per distinguerlo dagli altri.
- **Tipologia**, indica la categoria a cui l'attrezzo appartiene (cardio, peso libero...).
- **Nome dell'attrezzo**.

Essendo dotata di un codice, è possibile determinare se un attrezzo specifico è stato soggetto ad una o più **manutenzioni** nel corso del tempo, che vengono identificate da:

- **Data di inizio** della manutenzione.
- **Descrizione** che contiene i motivi della manutenzione.
- **Data di fine** della manutenzione.

Ogni attrezzatura essendo un'istanza unica e identificabile può essere allocata in una sola sala per volta, queste **sale** sono descritte da:

- **Nome della sala.**
- **Tipologia di sala** (spinning, cardio, pesi...).
- **Capienza** in persone della sala.

All'interno delle sale possono essere svolti più **corsi**, ovviamente in orari differenti, ciascuno dei quali possiede:

- **Nome**, che identifica il corso.
- **Descrizione**, che fornisce informazioni aggiuntive sul corso.
- La **durata** della lezione di un corso.
- Il **livello** associato al corso (principiante, intermedio, avanzato).

Questi corsi si articolano in una o più **lezioni** pianificate, che quindi associano un istruttore ad un corso specifico e includono:

- **Data e orario** di svolgimento della lezione.

Ogni cliente ha la possibilità di prenotare una o più lezioni e a ciascuna **prenotazione** è associata la seguente informazione:

- **Presenza**, che indica se l'utente ha effettivamente partecipato alla lezione.

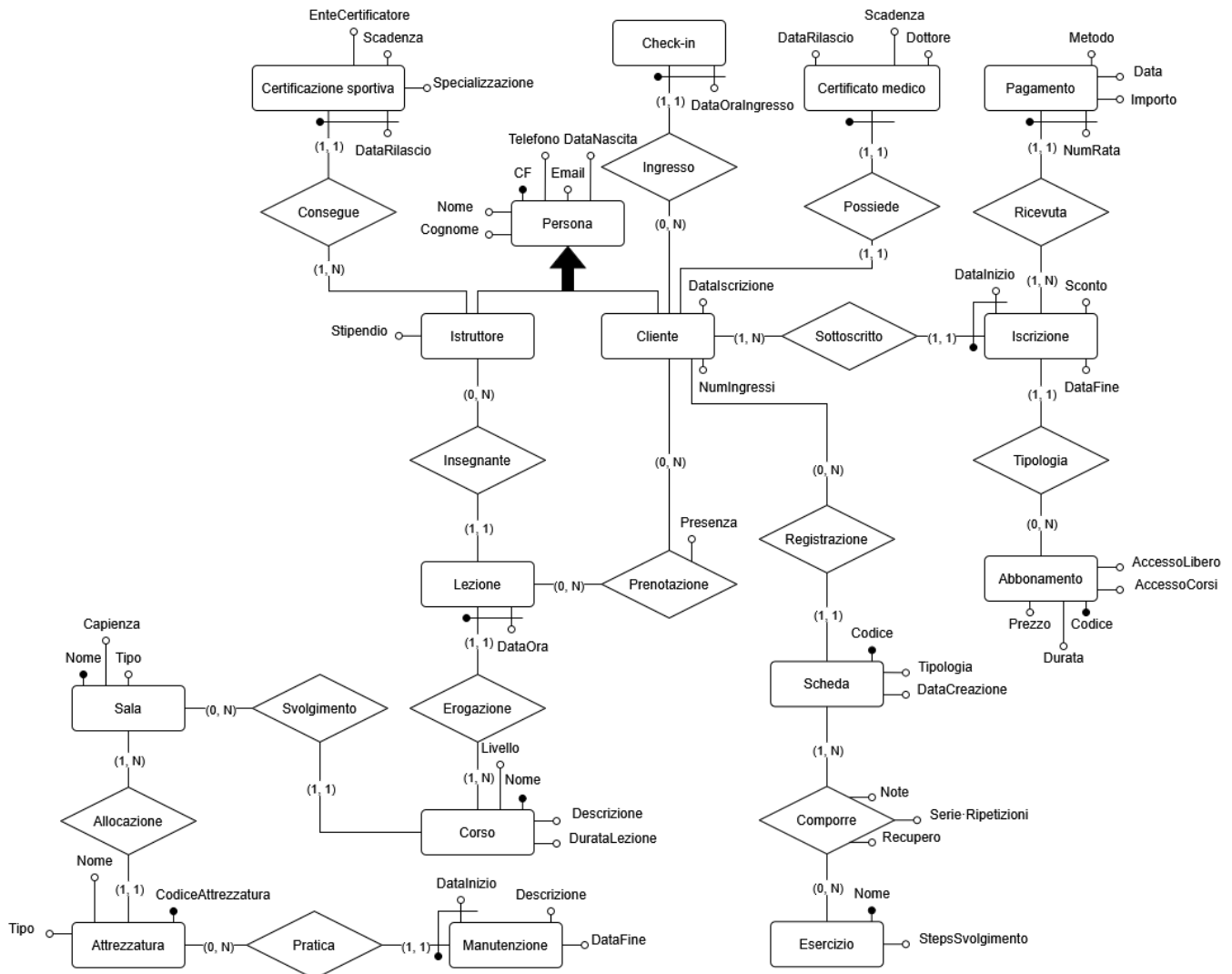
Check-in. Ogni cliente può accedere alla struttura in orari e giorni differenti, vengono quindi registrati i vari check-in associati al singolo cliente, salvando la seguente informazione:

- **Data e orario** dell'ingresso.

3 Progettazione concettuale

3.1 Schema concettuale (E-R)

La figura riportata rappresenta il diagramma Entità-Relazione (E-R) che riassume la struttura ed i requisiti riportati al paragrafo 2.



Il sistema si concentra sull'esistenza di due entità principali interne alla palestra, ovvero l'istruttore e il cliente. Ciascun cliente, per essere considerato tale, deve sottoscrivere un'iscrizione al centro sportivo e deve possedere obbligatoriamente un certificato medico valido che deve essere registrato all'interno della base di dati. L'iscrizione permette al cliente di disporre di un abbonamento specifico (es. mensile, trimestrale, annuale, ...). Il costo dell'iscrizione di ciascun cliente può essere estinto tramite un pagamento immediato o rateizzandolo in più pagamenti. Inoltre il sistema permette di tenere traccia degli ingressi giornalieri effettuati dai clienti. La struttura non potrebbe funzionare senza la presenza di istruttori qualificati, i quali, se abilitati tramite una o più certificazioni sportive registrate all'interno della base di dati, possono prendere parte all'insegnamento in determinati corsi organizzati dal centro. All'interno di questa entità è racchiusa anche la figura di personal trainer. Ciascun corso della struttura si colloca, obbligatoriamente, all'interno di una delle sale apposite. Ogni giorno a diverse ore della giornata si possono svolgere le lezioni per i diversi corsi, alle quali possono prenotarsi i clienti abilitati alla fruizione di queste attività (grazie al proprio abbonamento). Per ciascun partecipante dovrà essere registrata la presenza. Per ciascuna sala è necessaria la presenza dell'attrezzatura, che può racchiudere macchinari o attrezzi per il corpo libero. Com'è possibile intuire la sala attrezzi è racchiusa nell'entità sala. Ciascun attrezzo della struttura subisce, per causa imprevista o programmata, una manutenzione da parte di uno staff esterno; questa manutenzione viene registrata per eventuali guasti/collaudi futuri. Inoltre la palestra offre la possibilità di creare delle schede di allenamento personalizzate per i clienti, composte da esercizi preparati dalla struttura.

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

- **Clienti:**
 - solamente i clienti che hanno abilitata un'iscrizione NON scaduta e un certificato medico valido possono accedere alla struttura;
 - solamente i clienti che hanno abilitata un'iscrizione ad un abbonamento con accesso ai corsi possono prenotarsi per le lezioni di qualsiasi corso;
 - solamente i clienti che hanno abilitata un'iscrizione ad un abbonamento con accesso libero possono farsi creare una scheda di allenamento personalizzata;
 - ciascun cliente può sottoscrivere solamente un'iscrizione contemporaneamente alla struttura.
- **Istruttore:**
 - solamente gli istruttori che hanno una certificazione sportiva valida possono lavorare presso la struttura sportiva;
 - solamente gli istruttori specializzati, tramite certificazione, come "Istruttore cardio" oppure "Istruttore mobilità" possono prendere parte all'insegnamento in determinati corsi.

3.2 Glossario di Entità e Relazioni

Si riporta di seguito la descrizione completa delle entità della base di dati (a meno che non sia specificato, nessun attributo sottostante assume valore NULL):

- **Persona:** rappresenta un membro della struttura sportiva
 - CF (Codice Fiscale): *string (max. 16)*
 - Nome: *string*
 - Cognome: *string*
 - Telefono: *string (max. 10)*
 - Email: *string*
 - DataNascita: *date*
- **Cliente:** qualsiasi cliente della struttura
 - DataIscrizione: *date*
 - NumIngressi: *int*
- **Istruttore:** qualsiasi istruttore e collaboratore della struttura
 - Stipendio (mensile): *float*
- **Iscrizione:** iscrizione alla struttura di un singolo cliente
 - DataInizio: *date + time* (relazione "Sottoscritto")
 - Sconto: *int*
 - DataFine: *date*
- **Pagamento:** rata del pagamento dell'iscrizione
 - NumRata: *int* (relazione "Ricevuta")
 - Data: *date + time*
 - Importo: *float*
 - Metodo: *enum* ("Contanti", "Elettronico")
- **Abbonamento:** tipologia di abbonamento in struttura
 - Codice: *int*
 - Prezzo: *float*
 - Durata (in giorni): *int*
 - AccessoLibero: *boolean*
 - AccessoCorsi: *boolean*
- **Certificato medico:** certificato medico non agonistico di un cliente
 - (relazione "Possiede")
 - DataRilascio: *date*
 - Medico: *string*
 - Scadenza: *date*
- **Check-in:** ingresso di un singolo cliente
 - DataOralIngresso: *date + time* (relazione "Ingresso")
- **Certificazione sportiva:** certificato abilitante per il lavoro di istruttore in una struttura sportiva
 - DataRilascio: *date* (relazione "Consegue")
 - Specializzazione: *enum* ("Personal trainer", "Istruttore cardio", "Istruttore mobilità")
 - EnteCertificatore: *string*
 - Scadenza: *date*
- **Corso:** corso fitness erogato dalla struttura
 - Nome: *string*
 - DurataLezione (in minuti): *int*
 - Descrizione: *string*
 - Livello: *enum* ("Base", "Intermedio", "Avanzato")

- **Lezione:** singola lezione per ciascun corso
 - DataOra: *date-time* (relazione “Erogazione”)
- **Sala:** singola sala della struttura sportiva
 - Nome: *string*
 - Capienza: *int*
 - Tipo: *enum* (“Sala attrezzi”, “Sala corsi”)
- **Attrezzatura:** attrezzo acquistato dalla struttura
 - CodiceAttrezzatura: *int*
 - Nome: *string*
 - Tipo: *enum* (“A carico”, “Isotonico”, “Cardio”, “Funzionale”)
- **Manutenzione:** attività di manutenzione per un attrezzo della struttura
 - DataInizio: *date* (relazione “Pratica”)
 - Descrizione: *string*
 - DataFine: *date* (eventualmente *NULL*)
- **Esercizio:** esercizio di sala attrezzi generico, per essere inserito nelle schede di allenamento
 - Nome: *string*
 - StepsSvolgimento: *string*
- **Scheda:** scheda di allenamento (per sala attrezzi) personalizzata per ciascun cliente
 - Codice: *int*
 - Tipologia: *enum* (“Massa”, “Tonificazione”)
 - DataCreazione: *date*

Si riporta di seguito la descrizione completa delle relazioni della base di dati (a meno che non sia specificato, nessun attributo sottostante assume valore *NULL*):

- **Iscrizione-Cliente: Sottoscritto** (1:N)
 - ogni iscrizione è sottoscritta da *uno* (p. obbligatoria) e *un solo* cliente → (1, 1);
 - ogni cliente sottoscrive *una* (p. obbligatoria) o *più* iscrizioni → (1, N).
- **Pagamento-Iscrizione: Ricevuta** (1:N)
 - ogni pagamento è effettuato per *una* (p. obbligatoria) e *una sola* iscrizione → (1, 1);
 - per ogni iscrizione sono effettuati *uno* (p. obbligatoria) o *più* pagamenti → (1, N).
- **Iscrizione-Abbonamento: Tipologia** (1:N)
 - ogni iscrizione fa riferimento ad *uno* (p. obbligatoria) e *un solo* abbonamento → (1, 1);
 - ogni abbonamento viene riferito da *nessuna* (p. opzionale) o *più* iscrizioni → (0, N).
- **Cliente-Certificato medico: Possiede** (1:1)
 - ogni cliente possiede *uno* (p. obbligatoria) e *un solo* certificato medico → (1, 1);
 - ogni certificato medico è a nome di *uno* (p. obbligatoria) e *un solo* cliente → (1, 1).
- **Check-in-Cliente: Ingresso** (1:N)
 - ogni check-in è effettuato da *uno* (p. obbligatoria) e *un solo* cliente → (1, 1);
 - ogni cliente effettua *nessuno* (p. opzionale) o *più* check-in → (0, N).
- **Certificazione sportiva-Istruttore: Consegue** (1:N)
 - ogni certificazione sportiva è conseguita da *uno* (p. obbligatoria) e *un solo* istruttore → (1, 1);
 - ogni istruttore consegue *una* (p. obbligatoria) o *più* certificazioni sportive → (1, N).
- **Lezione-Istruttore: Insegnante** (1:N)
 - ogni lezione è tenuta da *uno* (p. obbligatoria) e *un solo* istruttore → (1, 1);
 - un istruttore tiene *nessuno* (p. opzionale) o *più* corsi → (0, N).
- **Cliente-Lezione: Prenotazione** (N:M)
 - un cliente prenota *nessuna* (p. opzionale) o *più* lezioni → (0, N);
 - una lezione è prenotata da *nessun* (p. opzionale) o *più* clienti → (0, N);
 - attributi:
 - Presenza: *boolean*
- **Lezione-Corso: Erogazione** (1:N)
 - ogni lezione appartiene a *uno* (p. obbligatoria) e *un solo* corso → (1, 1);
 - ogni corso eroga *una* (p. obbligatoria) o *più* lezioni → (1, N).
- **Corso-Sala: Svolgimento** (1:N)
 - ogni corso si svolge in *una* (p. obbligatoria) e *una sola* sala → (1, 1);
 - in una sala si svolgono *nessuno* (p. opzionale) o *più* corsi → (1, N).
- **Attrezzatura-Sala: Allocazione** (1:N)
 - ogni attrezzo è allocato in *una* (p. obbligatoria) e *una sola* sala → (1, 1);
 - ogni sala ha allocati *uno* (p. obbligatoria) o *più* attrezzi → (1, N).
- **Manutenzione-Attrezzatura: Pratica** (1:N)
 - ogni manutenzione è praticata su *uno* (p. obbligatoria) e *un solo* attrezzo → (1, 1);
 - su un attrezzo è svolta *nessuna* (p. opzionale) o *più* manutenzioni → (0, N).

- **Esercizio-Attrezzatura: Utilizzo (1:N)**
 - ogni esercizio fa uso di *nessuno* (p. opzionale) o *un* attrezzo → (1, 1);
 - un attrezzo è utilizzato in *nessuno* (p. opzionale) o *più* esercizi → (0, N).
- **Esercizio-Scheda: Comporre (N:M)**
 - un esercizio è presente in *nessuna* (p. opzionale) o *più* schede → (0, N);
 - ogni scheda è composta da *uno* (p. obbligatoria) o *più* esercizi → (1, N);
 - attributi:
 - Note: *string*
 - Serie · Ripetizioni: *string* (eventualmente *NULL*)
 - Recupero (in secondi): *int*
- **Scheda-Cliente: Registrazione (1:N)**
 - ogni scheda è registrata a nome di *uno* (p. obbligatoria) e *un solo* cliente → (1, 1);
 - ad un cliente è registrata *nessuna* (p. opzionale) o *più* schede → (0, N).

L'unica generalizzazione presente all'interno dello schema E-R è **Persona**, di tipo *totale* di **Cliente** e **Istruttore**.

4 Progettazione logica

All'interno di questo paragrafo si vuole effettuare una "traduzione" del diagramma E-R e di tutti quelli che sono i requisiti individuati dal caso di studio.

I passi che seguirà questa traduzione sono i seguenti:

- eliminazione di possibili ridondanze all'interno della progettazione concettuale per ottimizzare la struttura della base di dati;
- eliminazione della generalizzazione totale;
- presentazione del diagramma ristrutturato (schema relazionale) con descrizione di possibili modifiche.

4.1 Analisi delle ridondanze

L'attributo *NumIngressi* per l'entità *Cliente*, che memorizza il numero di ingressi che il singolo cliente effettua all'interno della struttura a partire dalla sua iscrizione, presenta una ridondanza. Questo dato si potrebbe ottenere contando il numero di check-in attivi per il singolo cliente tramite la relazione *Ingresso*.

Successivamente a quanto descritto si decide di procedere con il seguente ragionamento: ogni qualvolta che il cliente scansiona il codice QR al suo ingresso (~20 ingressi mensilmente¹) l'attributo viene modificato; inoltre viene visualizzato, dallo stesso cliente, periodicamente (~8 volte mensilmente²).

Le operazioni sono quindi le seguenti:

- **O1 (20 volte al mese)**: memorizzazione di un nuovo ingresso;
- **O2 (8 volte al mese)**: visualizzazione del numero di ingressi effettuati.

Si assumano i seguenti volumi per la base di dati:

Concetto	Costrutto	Volume
Cliente	Entità	500
Check-in	Entità	60.000 ³
Ingresso	Relazione	60.000

¹ 20 ingressi mensili si calcolano con una media di 5 ingressi alla settimana (corsi + sala attrezzi).

² 8 visualizzazioni mensili si calcolano con una media di 2 visualizzazioni alla settimana (apertura app).

³ 60.000 check-in si calcolano con una media di minimo 6 mensilità di frequentazione da parte dei clienti.

RIDONDANZA Si analizzi il costo totale:

- **O1:**

Concetto	Costrutto	Accesso	Tipo	
Check-in	Entità	1	Scrittura	× 20
Ingresso	Relazione	1	Scrittura	× 20
Cliente	Entità	1	Lettura	× 20
Cliente	Entità	1	Scrittura	× 20

Al termine dell'operazione si registra un costo di 60 operazioni in scrittura e 20 in lettura.

- **O2:**

Concetto	Costrutto	Accesso	Tipo	
Cliente	Entità	1	Lettura	× 8

Al termine dell'operazione si registra un costo di 8 operazioni in lettura.

Si assuma un costo ×2 per gli accessi in scrittura :

Costo Totale = $60 \times 2 + 20 + 8 = 148$

SENZA RIDONDANZA Si analizzi il costo totale:

- **O1:**

Concetto	Costrutto	Accesso	Tipo	
Check-in	Entità	1	Scrittura	× 20
Ingresso	Relazione	1	Scrittura	× 20

Al termine dell'operazione si registra un costo di 40 operazioni in scrittura.

- **O2:**

Concetto	Costrutto	Accesso	Tipo	
Cliente	Entità	1	Lettura	× 8
Ingresso	Relazione	20	Lettura	× 8

Al termine dell'operazione si registra un costo di 168 operazioni in lettura.

Si assuma un costo ×2 per gli accessi in scrittura (lettura dato + modifica dato):

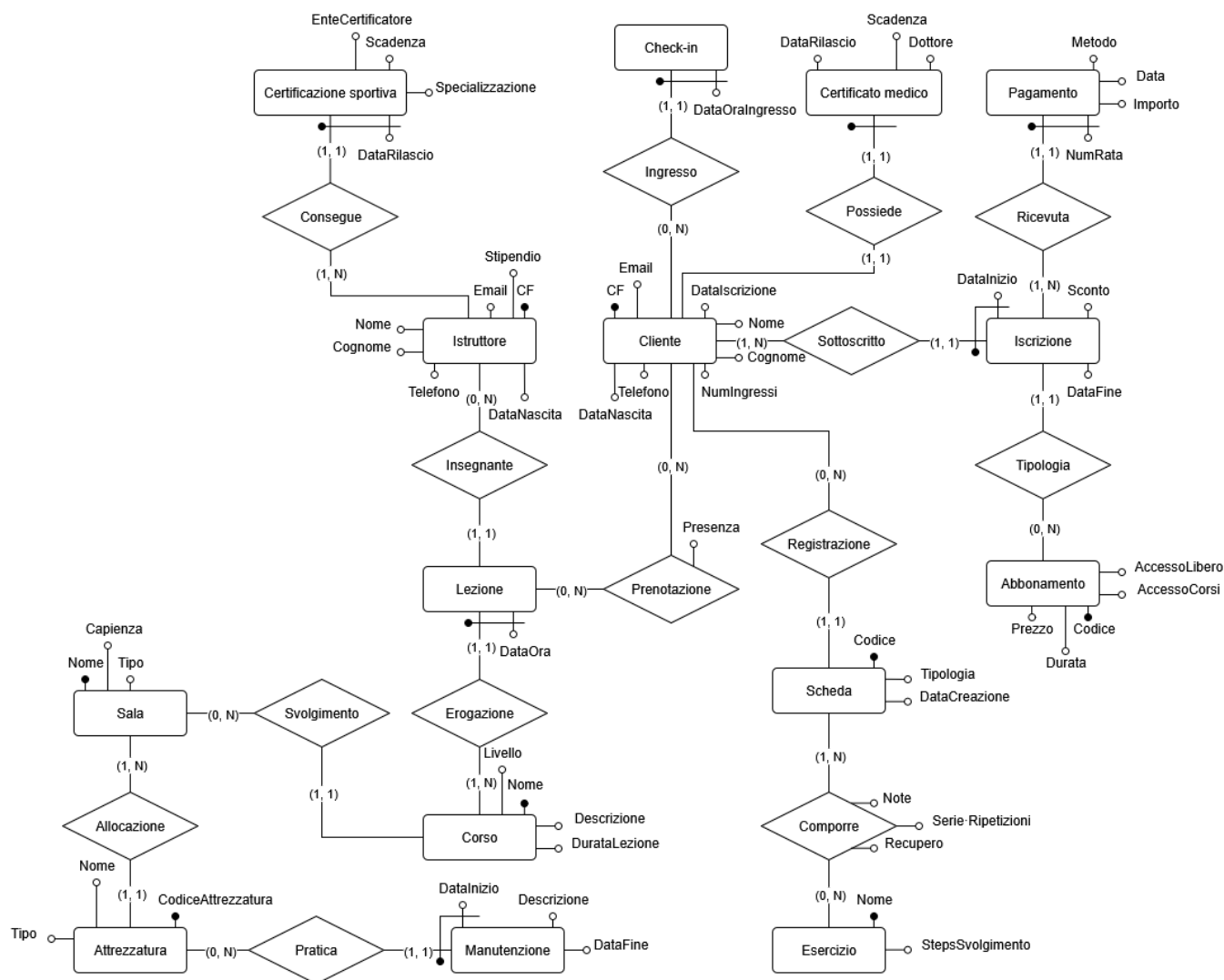
Costo Totale = $40 \times 2 + 168 = 240$

Al termine dell'analisi per la rimozione o meno dell'attributo ridondante si conclude che è opportuno **mantenere** l'attributo ridondante *NumIngressi*, ottimizzando così il numero di accessi alla base di dati. La modifica del valore dell'attributo viene fatta a livello applicazione ogni qualvolta viene registrato un nuovo ingresso per ciascun cliente.

4.2 Eliminazione della generalizzazione

La generalizzazione descritta all'interno del diagramma E-R presente al paragrafo 3 deve essere eliminata con una ristrutturazione dello stesso schema concettuale; ciò viene fatto per semplificare la successiva implementazione del modello relazionale.

PERSONA La generalizzazione totale di *Persona* viene rimossa, includendo così tutti gli attributi “comuni” all'interno delle entità figlie di *Istruttore* e *Cliente*. Ciò risulta una soluzione ottimale poiché l'entità *Persona* non viene riferita da nessuna relazione, pertanto non ci sarebbero duplicazioni di relazioni. Segue, dopo la spiegazione, la figura del diagramma E-R rimodulato.



4.3 Schema relazionale

Segue la rappresentazione dello schema logico; gli attributi contrassegnati da un asterisco consentono valori nulli.

- **Istruttore**(CF, Nome, Cognome, DataNascita, Telefono, Email, Stipendio)
- **Certificazione sportiva**(Istruttore, DataRilascio, Specializzazione, EnteCertificatore, Scadenza)
 - Certificazione sportiva.Istruttore → Istruttore.CF
- **Cliente**(CF, Nome, Cognome, DataNascita, Telefono, Email, DataIscrizione, NumIngressi)
- **Check-in**(Cliente, DataOralIngresso)
 - Check-in.Cliente → Cliente.CF
- **Certificato medico**(Cliente, DataRilascio, Scadenza, Dottore)
 - Certificato medico.Cliente → Cliente.CF

- **Iscrizione**(Cliente, DataInizio, Sconto, DataFine, Abbonamento)
 - Iscrizione.Abbonamento → Abbonamento.Codice
 - Iscrizione.Cliente → Cliente.CF
- **Abbonamento** (Codice, Prezzo, Durata, AccessoLibero, AccesoCorsi)
- **Pagamento**(Cliente, Iscrizione, NumRata, Metodo, Data, Importo)
 - Pagamento.(Cliente, Iscrizione) → Iscrizione.(Cliente, DataInizio)
- **Scheda**(Codice, Tipologia, DataCreazione, Cliente)
 - Scheda.Cliente → Cliente.CF
- **Comporre**(Scheda, Esercizio, Note, SerieRipetizioni, Recupero)
 - Comporre.Scheda → Scheda.Codice
 - Comporre.Esercizio → Esercizio.Nome
- **Esercizio**(Nome, StepsSvolgimento)
- **Attrezzatura**(CodiceAttrezzatura, Nome, Tipo, Sala)
 - Attrezzatura.Sala → Sala.Nome
- **Manutenzione**(Attrezzatura, DataInizio, DataFine*, Descrizione)
 - Manutenzione.Attrezzatura → Attrezzatura.CodiceAttrezzatura
- **Sala**(Nome, Tipo, Capienza)
- **Corso**(Nome, DurataLezione, Descrizione, Livello, Sala)
 - Corso.Sala → Sala.Nome
- **Lezione**(Corso, DataOra, Istruttore)
 - Lezione.Corso → Corso.Nome
 - Lezione.Istruttore → Istruttore.CF
- **Prenotazione**(Corso, Lezione, Cliente, Presenza)
 - Prenotazione.(Corso, Lezione) → Lezione.(Corso, DataOra)
 - Prenotazione.Cliente → Cliente.CF

5 Implementazione del DB e definizione delle Query

All'interno del file **palestra.sql** è presente il codice SQL prodotto dal gruppo per la creazione ed il popolamento della base di dati. Inoltre, all'interno del file sono incluse 5 query per interrogare il database ed estrapolare dati significativi insieme a 2 indici ideati per migliorare le prestazioni di due di queste interrogazioni.

5.1 Definizione delle Query

Q1 Stampare tutti i clienti che devono ancora pagare alla struttura almeno una certa percentuale del prezzo (ad esempio 50%) della propria quota di iscrizione attiva. Ordinare i clienti per “debito” più cospicuo.

```

1 SELECT c.CF, c.Nome, c.Cognome, i.DataInizio, i.DataFine,
2        SUM(p.Importo) AS TotalePagato,
3        ROUND(a.Prezzo * (1 - i.Sconto / 100.0), 2) AS PrezzoIscrizione,
4        (ROUND(a.Prezzo * (1 - i.Sconto / 100.0), 2) - SUM(p.Importo)) AS SaldoResiduo
5 FROM Cliente c
6 JOIN Iscrizione i ON c.CF = i.Cliente
7 JOIN Abbonamento a ON i.Abbonamento = a.Codice
8 JOIN Pagamento p ON i.Cliente = p.Cliente AND i.DataInizio = p.Iscrizione
9 WHERE i.DataFine >= CURRENT_DATE
10 GROUP BY c.CF, c.Nome, c.Cognome, i.DataInizio, i.DataFine, a.Prezzo, i.Sconto
11 HAVING ((a.Prezzo * (1 - i.Sconto / 100.0)) * (50.0 / 100)) <=
12         (ROUND(a.Prezzo * (1 - i.Sconto / 100.0), 2) - SUM(p.Importo))
13 ORDER BY SaldoResiduo DESC

```

	cf character (16)	nome character varying (20)	cognome character varying (20)	datainizio date	datafine date	totalepagato numeric	prezzoiscrizione numeric	saldoresiduo numeric
1	GQN9DF0R68JETGQN	Enzo	Armellini	2024-12-22	2025-12-17	100.00	400.00	300.00
2	EC39NXSHA1TBFJ3S	Romana	Vergassola	2024-11-01	2025-10-27	100.00	400.00	300.00
3	BQD2URT9UDJM7S83	Benedetto	Brugnaro	2024-10-05	2025-09-30	100.00	400.00	300.00
4	KXPZ19G9C5CLCD4J	Baccio	Valguarnera	2025-05-15	2026-05-10	100.00	400.00	300.00
5	9WU6TA3LF42T49FI	Vittorio	Fuseli	2024-12-03	2025-11-28	100.00	400.00	300.00
6	7FK2S14FBVJ5WE52	Valerio	Rosmini	2025-03-28	2026-03-23	100.00	400.00	300.00
7	6GK5U1QL4EFCG4D...	Atenuif	Giannone	2024-08-21	2025-08-16	100.00	400.00	300.00
8	MNEIBZCN3CQOQS...	Lorenzo	Antonello	2024-09-09	2025-09-04	95.00	380.00	285.00
9	4OW9UW89DQ6FY0...	Salvi	Schiavo	2024-07-30	2025-07-25	95.00	380.00	285.00
10	4AZ2WA5F859MW8...	Serafina	Bonomo	2024-10-27	2025-10-22	95.00	380.00	285.00
11	Y74Y3BINKN8EW34X	Adriana	Littizzetto	2025-03-10	2026-03-05	95.00	380.00	285.00
12	3DVTXW25C2VRP2...	Annamaria	Comboni	2025-04-16	2026-04-11	95.00	380.00	285.00
13	NI3JRLW41288HTL3	Silvia	Caffarelli	2024-06-19	2025-06-14	95.00	380.00	285.00
14	FL9HF3IH6SYU3EUM	Gianpietro	Valentino	2025-01-25	2026-01-20	95.00	380.00	285.00
15	ROO2FVW363UVEX...	Adriana	Casaleggio	2025-05-09	2026-05-04	95.00	380.00	285.00
16	7NOO7T7U7R36HZTY	Liana	Fagiani	2024-11-05	2025-10-31	95.00	380.00	285.00
17	3XOMVW25NIGNHC...	Tatiana	Golgi	2024-08-24	2025-08-19	90.00	360.00	270.00
18	CQRPNB6WPCBML...	Eleanora	Anguissola	2024-12-30	2025-12-25	90.00	360.00	270.00
19	QAALND3AKTAS3Z1O	Marcella	Errigo	2024-08-12	2025-08-07	90.00	360.00	270.00
20	E4M5KRBHEEE3CW...	Gino	Tedesco	2024-08-29	2025-08-24	90.00	360.00	270.00
Total rows: 81		Query complete 00:00:00.185						

Q2 Stampare i clienti con almeno un certo numero di assenze (ad esempio 5) ai corsi ai quali si sono prenotati.

```

1 SELECT c.CF, c.Nome, c.Cognome, COUNT(*) AS Assenze
2 FROM Cliente c
3 JOIN Prenotazione p ON c.CF = p.Cliente
4 WHERE Presenza = FALSE
5 GROUP BY c.CF, c.Nome, c.Cognome
6 HAVING COUNT(*) >= 5

```

	cf [PK] character (16)	nome character varying (20)	cognome character varying (20)	assenze bigint
1	CQRPNB6WPCBML...	Eleanora	Anguissola	5
2	XOMGCS8RZ7A4G7...	Alessio	Finetti	5
3	ER88XJIRAUKEGF8K	Lucio	Leonetti	5
4	BH6DZ96WBMBSX...	Pierluigi	Faranda	5
5	OUYFXC0XM1A900IU	Liberto	Bacosi	5
6	YVNBFM911CWCP...	Guarino	Mascagni	5
7	PYPH1PUQRL8GAM...	Fabrizia	Sonnino	5
8	H7KA62CA2UCVL5LF	Rocco	Boslo	5
9	GSCI2PM3VIMN002...	Sonia	Bolardo	5
10	KG98UO448GZV4R...	Ruggero	Benussi	5
11	IWSO8L2DOAVV5U...	Eleanora	Finotto	5
12	Q3NR6D7LQ9PEW0...	Margherita	Cimini	5
13	AOLF87JE12UG8GXF	Giuseppe	Jovinelli	6
14	3PA13PH570F15ILA	Fabrizia	Mancini	5
15	U8EPF9FBT12JME0H	Oreste	Giradello	5
Total rows: 15		Query complete 00:00:00.534		

Q3 Stampare per ciascun abbonamento, offerto dalla palestra, il numero di clienti che lo hanno sottoscritto (anche in più iscrizioni). Ordinare i risultati in ordine decrescente.

```

1 SELECT a.*, COUNT(DISTINCT i.Clienti) AS Clienti
2 FROM Abbonamento a
3 LEFT JOIN Iscrizione i ON a.Codice = i.Abbonamento
4 GROUP BY a.Codice
5 ORDER BY Clienti DESC

```

	codice [PK] integer	prezzo numeric (6,2)	durata integer	accessolibero boolean	accessocorsi boolean	clienti bigint
1	6	350.00	6	false	true	132
2	8	450.00	12	false	true	101
3	7	400.00	12	true	false	99
4	3	60.00	1	true	false	59
5	5	250.00	6	true	false	52
6	4	100.00	1	false	true	51
7	1	10.00	1	true	false	44
8	2	15.00	1	false	true	44
Total rows: 8		Query complete 00:00:00.474				

Q4 Trovare l'istruttore (o gli istruttori) che ha tenuto la lezione con il maggior numero di presenze effettive.

```

1 SELECT DISTINCT CF, Nome, Cognome, Num_prenotazioni
2 FROM (
3     SELECT I.CF, I.Nome, I.Cognome, COUNT(*) AS Num_prenotazioni
4     FROM Istruttore I
5     JOIN Lezione L ON I.CF = L.Istruttore
6     JOIN Prenotazione P ON P.Lezione = L.DataOra AND P.Corso = L.Corso
7     WHERE P.Presenza = TRUE
8     GROUP BY I.CF, I.Nome, I.Cognome, L.Corso, L.DataOra
9     HAVING COUNT(*) = (
10         SELECT MAX(Num_prenotazioni) FROM (
11             SELECT COUNT(*) AS Num_prenotazioni
12             FROM Lezione L
13             JOIN Prenotazione P ON P.Lezione = L.DataOra AND P.Corso = L.Corso
14             WHERE P.Presenza = TRUE
15             GROUP BY L.Corso, L.DataOra
16         ) AS Prenotazioni
17     )
18 ) AS IstruttoriTop

```

	cf [PK] character (16)	nome character varying (20)	cognome character varying (20)	num_prenotazioni bigint
1	AA8AWEHWKSS97...	Alfredo	Gianetti	13
2	CJM2JOPOU6I8RS2V	Geronimo	Vasari	13
3	HYF2FOA1BZLR8QDF	Caterina	Anguillara	13
Total rows: 3		Query complete 00:00:00.450		

Q5 Calcolare il numero di manutenzioni attive per ciascuna sala, ordinandole dalla più "problematica" alla meno.

```

1 SELECT S.Nome, COUNT (*) AS Manutenzioni_Atтиве
2 FROM Sala S
3 JOIN Attrezzatura A ON A.Sala = S.Nome
4 JOIN Manutenzione M ON M.Attrezzatura = A.CodiceAttrezzatura
5 WHERE M.DataFine IS NULL
6 GROUP BY S.Nome
7 ORDER BY Manutenzioni_Atтиве DESC

```

	nome [PK] character varying (16)	manutenzioni_attive bigint
1	Sala B	14
2	Sala E	8
3	Sala C	7
4	Sala A	7
5	Sala D	5
Total rows: 5		Query complete 00:00:00.524

5.2 Creazione degli indici

I1 Con la creazione del primo indice si punta ad ottimizzare la **Q1**, all'interno della quale si distingue l'operazione:

- condizione di WHERE: $\sigma_{Iscrizione.DataFine \geq CURRENT_DATE}$

Analizzando la condizione di WHERE della stessa **Q1** è opportuna la creazione di un indice B+ Tree, poiché permetterebbe di facilitare la ricerca dei dati con il criterio specificato (aumento dell'efficienza) e, successivamente, a differenza di un indice hash può essere utilizzato per confronti diversi dall'uguaglianza:

```
1 CREATE INDEX idx_datafine_iscr ON Iscrizione(DataFine);
```

I2 Con la creazione dei seguenti indici si punta ad ottimizzare la **Q4**, all'interno della quale si distingue l'operazione:

- condizioni di JOIN:
 - $Istruttore \bowtie_{(Istruttore.CF = Lezione.Istruttore)} Lezione;$
 - $\bowtie_{(Lezione.Corso = Prenotazione.Corso \wedge Lezione.DataOra = Prenotazione.Lezione)} Prenotazione;$

Analizzando la seconda condizione di JOIN della stessa **Q4** si nota che questa viene eseguita sulla coppia di attributi (*Corso*, *DataOra*), pertanto si ritiene opportuna la creazione di un indice B+ Tree su entrambi gli attributi che permette di identificare i record della tabella *Lezione* corrispondenti a ciascuna *Prenotazione*, riducendo il numero di tuple lette durante la condizione di JOIN. L'indice può essere utilizzato in seguito nella sottoquery per la condizione di JOIN (simile a quella descritta precedentemente) e GROUP BY, evitando una fase di sorting costosa (aumento dell'efficienza)..

```
1 CREATE INDEX idx_lezione ON Lezione(Corso, DataOra);
```

È importante sottolineare che il DBMS PostgreSQL, utilizzato per la definizione di questa base di dati, si occupa automaticamente di creare questo indice per ciascuna chiave primaria delle tabelle (in questo caso gli attributi *Corso* e *DataOra* formano chiave primaria per la tabella *Lezione*) pertanto questo indice viene già definito implicitamente.

Si precisa che gli indici precedentemente creati risultano particolarmente utili con un elevato volume di dati, migliorando significativamente le prestazioni delle operazioni di accesso ed interrogazione.

6 Codice C per query

È stato creato il file *palestra.c*, il quale contiene il codice in linguaggio C per la connessione alla base di dati e visualizzazione su prompt dei risultati delle query precedentemente definite.

Il programma permette di visualizzare un menù che mostra le query eseguibili, con le seguenti opzioni:

1. Prezzo di iscrizione e totale pagato per ciascun cliente.
2. Clienti con minimo num. di assenze ai corsi.
3. Classifica degli abbonamenti con numero di sottoscrizioni.
4. Istruttori con più partecipanti presenti.
5. Manutenzioni attive per ciascuna sala.

Si evidenzia che l'**interrogazione 1** è stata progettata in modalità parametrica: al momento della selezione, il sistema richiede all'utente l'inserimento di una percentuale di soglia minima che il cliente deve ancora pagare alla struttura, che costituisce il criterio per filtrare i clienti da includere nel risultato.

Inoltre, anche l'**interrogazione 2** è stata progettata in modalità parametrica: al momento della selezione, il sistema richiede all'utente l'inserimento del numero minimo di assenze, che costituisce il criterio per filtrare i clienti da includere nel risultato.