**­­­**

**Elaborato per il corso di Basi di dati**

**A.A 2021/2022**

***Progetto di una base di dati per la gestione di uno stabilimento balneare***

Muccioli Federico

federico.muccioli5@studio.unibo.it

0000971342

**Analisi dei requisiti**

[Intervista 3](#_Toc109824387)

[Estrazione dei concetti principali 4](#_Toc109824388)

[Progettazione concettuale 5](#_Toc109824389)

[Schema Scheletro 5](#_Toc109824390)

[Schema finale  10](#_Toc109824391)

[Progettazione logica 12](#_Toc109824392)

[Stima del volume dei dati 12](#_Toc109824393)

[Descrizione delle operazioni principali e stima della loro frequenza 13](#_Toc109824394)

[Schemi di navigazione e tabelle degli accessi 14](#_Toc109824395)

[Raffinamento dello schema 19](#_Toc109824396)

[Analisi delle ridondanze 20](#_Toc109824397)

[Traduzione di entità e associazioni in relazioni 22](#_Toc109824398)

[Schema relazionale finale 25](#_Toc109824399)

[Traduzione delle operazioni in query SQL 27](#_Toc109824400)

[Progettazione dell’applicazione 30](#_Toc109824401)

[Descrizione dell’architettura dell’applicazione realizzata 30](#_Toc109824402)

**Analisi dei requisiti**

Si vuole realizzare una base di dati per uno stabilimento balneare che gestisca i dati relativi ai dipendenti e clienti, i listini prezzi e la disposizione di ombrelloni e lettini sulla spiaggia.

## Intervista

Si vuole memorizzare la disposizione degli ombrelloni sulla spiaggia con la possibilità di aggiungere ed eliminare un ombrellone, questo in quanto il titolare paga una tassa per ogni ombrellone piantato e il numero di ombrelloni utilizzati sarà diverso a giugno rispetto che ad agosto.

Gli ombrelloni saranno disposti per file e colonne, questo per avere un bagnino ordinato e rendere più facile l’accesso al cliente.

Ogni ombrellone è raffigurato da un numero identificativo per essere riconosciuto dal cliente.

Ogni ombrellone comprende anche due lettini di default, ma che potranno essere aggiunti o eliminati a piacere del cliente.

Sarà possibile acquistare anche solo dei lettini senza ombrellone che verranno disposti a riva.

Esisteranno sedie e sdraio che potranno sostituire i lettini a preferenza del cliente.

Per questioni di spazio sarà possibile piantare al massimo 100 ombrelloni e disporre al massimo 20 lettini a riva e 4 sotto ad un ombrellone.

I clienti potranno affittare gli ombrelloni, lettini, sedie e sdraio per un certo periodo e il prezzo sarà dato a seconda di un listino che varia in base al mese (giugno sarà più conveniente rispetto che agosto), al destinatario (clienti del posto pagheranno meno rispetto a turisti) e al periodo di soggiorno (più il periodo sarà lungo e minore sarà il prezzo per giorno).

Verrà inoltre mantenuto uno storico dei listini.

Saranno anche registrati i dati dei dipendenti ovvero i bagnini: nome, cognome, data di nascita, indirizzo, numero di telefono e codice univoco.

Al momento dell’acquisto verranno registrati i dati dei clienti: nome, cognome, numero di telefono (opzionale), periodo di permanenza, ombrelloni e lettini affittati ed infine il bagnino che accompagnerà il cliente (quest’ultimo in caso i clienti vengano accompagnati all’ombrellone errato sarà possibile rintracciare il responsabile dell’equivoco).

## Rilevamento delle ambiguità e correzioni proposte

## Definizione delle specifiche in linguaggio naturale ed estrazione dei concetti principali

|  |  |  |
| --- | --- | --- |
| **Termine** | **Breve descrizione** | **Eventuali sinonimi** |
| Cliente | Colui che frequenta la palestra ed effettua allenamenti in base a quanto riportato sulla propria scheda | Iscritto |
| Istruttore | Colui che deve assistere i clienti durante i propri allenamenti e si occupa di assegnare nuove schede | Membro dello staff |
| Scheda | Oggetto che riporta gli esercizi da effettuare durante un allenamento organizzandoli in tabelle | Scheda di allenamento |
| Tabella | Componente fondamentale nella quale si dividono le schede | Routine di esercizi |
| Serie | Ripetizione in sequenza di uno stesso esercizio un determinato numero di volte |  |

Recupero Tempo indicato in minuti e secondi di pausa fra Pausa

una serie di esercizi e la successiva

A seguito della lettura e comprensione dei requisiti, si procede redigendo un testo che ne riassuma tutti i concetti e in particolare ne estragga quelli principali eliminando le ambiguità sopra rilevate:

Per ogni ***cliente*** della palestra vengono memorizzati nome, cognome, codice fiscale ed eventualmente un numero di telefono e una mail. Ogni cliente possiede un codice univoco fornitogli al momento dell’iscrizione. Un cliente pu entrare in palestra solo nel caso in cui disponga di un abbonamento valido, ogni ingresso viene memorizzato con relativa data e ora.

Ogni ***abbonamento*** ha una durata e un prezzo e al momento dell’acquisto si pu effettuare uno sconto per il cliente. I prezzi degli abbonamenti sono memorizzati in un ***listino*** di cui si mantiene uno storico delle versioni. L’abbonamento pu essere sospeso una sola volta per un determinato periodo per posticiparne la scadenza. Di ogni cliente si memorizzano le vecchie schede e la ***scheda*** attuale che perde di validità nel momento in cui ne viene assegnata una nuova.

Una scheda si compone di almeno una ***tabella***. Per ogni settimana della durata della scheda, gli ***esercizi*** riportati nelle tabelle sono caratterizzati da un numero di serie da effettuare, ciascuna con un numero di ripetizioni, un recupero e un eventuale peso da usare. Schede e esercizi posseggono ***obiettivi***.

Gli ***istruttori*** devono assegnare nuove schede ai clienti, compilarne di nuove e creare esercizi. Infine devono assistere quei clienti i quali richiedano sessioni di allenamento private (un istruttore pu assistere al massimo un cliente al giorno e un cliente pu prenotare al massimo un allenamento al giorno).

Segue un elenco delle principali azioni richieste:

1. Iscrivere un nuovo cliente
2. Aggiungere un nuovo istruttore
3. Registrare l’ingresso di un cliente
4. Registrare l’acquisto di un abbonamento da parte di un cliente
5. Cambiare il listino prezzi per l’anno corrente
6. Sospendere l’abbonamento di un cliente
7. Assegnare una scheda già esistente ad un cliente
8. Compilare una nuova scheda
9. Leggere tutte le tabelle della scheda attuale di un cliente
10. Data la tabella di una scheda visualizzare tutti i suoi esercizi
11. Visualizzare gli esercizi in base a particolari filtri (e.g. con uno specifico obiettivo o che riguardano uno specifico muscolo etc.)
12. Registrare la prenotazione di una sessione di allenamento individuale
13. Mostrare il prossimo allenamento di un cliente o un istruttore

# Progettazione concettuale

## Schema Scheletro

Le entità di **istruttore** e **cliente** sono la generalizzazione di una entità **persona**, identificata tramite un codice univoco (non si utilizza il codice fiscale per evitare problemi di omocodia). Dall’analisi del dominio si evince come un cliente non possa prenotare più di un allenamento in una giornata così come un istruttore non possa seguire più di un allenamento in una giornata. Perci si utilizzano come identificatori della entità **allenamento** sia la combinazione di istruttore e giorno che cliente e giorno tramite le associazioni **assiste** e **prenotazione**.

Genere

DataNascita

CodiceFiscale

Codice

Nome

Cognome

DataIscrizione

Telefono

(0-1)

Mail

(0-1)

CLIENTE

PERSONA

ISTRUTTORE

Giorno

ALLENAMENTO

ASSISTE

PRENOTAZIONE

(0-N)

(0-N)

(1-1)

(1-1)

(

)

t, e

OraInizio OraFine

*Fig. 1.1 Schema E/R con le principali entità per modellazione degli allenamenti*

Un cliente possiede più **abbonamenti** nel tempo, dunque non è sufficiente una associazione fra cliente e tipo abbonamento con un attributo *“data di acquisto”* perché non permetterebbe a un cliente di possedere più volte una stessa tipologia di abbonamento (e.g. un cliente non potrebbe comprare due volte l’abbonamento mensile nell’anno 2018). Per risolvere tale problema si ricorre alla reificazione dell’acquisto dell’abbonamento identificando tale entità tramite il codice del cliente e la data di acquisto.

Rimane il vincolo inespresso che un cliente non pu acquistare un abbonamento fino alla scadenza del precedente acquisito.

Poiché il **listino** degli abbonamenti pu cambiare al più fra un anno e il successivo viene identificato tramite l’anno nel quale è stato composto. Le tipologie di abbonamento di cui si compone devono avere tutte durata differente, questo viene modellato utilizzando come identificatore di tipologia abbonamento la sua durata e l'anno del listino nel quale compare.

Genere

DataNascita

CodiceFiscale

Codice

Nome

Cognome

DataIscrizione

DataSospensione

DataRiattivazione

Sconto

EFFETTUAZIONE

Telefono

(0-1)

Mail

(0-1)

CLIENTE

DataOra

INGRESSO

(0-N)

(1-1)

DataAcquisto

(0-N)

ACQUISIZIONE

(1-1)

Sospensione

(0-1)

Prezzo

Durata

TIPO

ABBONAMENTO

(1-1)

TIPOLOGIA

(0-N)

ABBONAMENTO

COMPOSIZIONE

Anno

LISTINO

(1-1)

(0-N)

PERSONA

*Fig. 1.2 Schema E/R rappresentante il sistema di abbonamenti e la loro storicizzazione*

Una stessa **scheda** pu essere assegnata a più clienti (e.g. scheda per principianti, scheda tonificazione…), dunque ho optato per utilizzare una seconda entità, **scheda posseduta**, per rappresentare il possedimento di un particolare tipo di scheda. Ogni scheda posseduta è posseduta da un solo cliente il quale dispone di una sola scheda attuale, tale possedimento è rappresentato dalla associazione **possedimento attuale**. Ho scelto di utilizzare tale distinzione nelle associazioni in quanto un cliente pu possedere molteplici schede ma sempre una sola scheda attuale che verrà visualizzata con alta frequenza, risulta quindi necessario poterla trovare velocemente (se non fosse presente questa seconda associazione, si dovrebbe scorrere tutte le schede possedute da un cliente e prendere quella con numero ordinale maggiore, il vantaggio di tale approccio è evidenziato nella Analisi delle Ridondanze).

Nome

Data

Genere

Durata

Codice

Ordinale

CLIENTE

ISTRUTTORE

POSSEDIMENTO

POSSEDIMENTO

ATTUALE

SCHEDA

POSSEDUTA

SCHEDA

COMPILAZIONE

PRODUCE

(0-1)

(0-N)

(0-1)

(1-1)

(1-1)

(1-1)

(0-N)

(0-N)

*Fig. 1.3 Schema E/R con la modellazione del possedimento delle schede*

Ogni **scheda** presenta una suddivisione in **tabelle** (identificate tramite la loro posizione nella scheda) a loro volta divise in sequenze di **esercizi** la cui posizione all’interno della tabella viene utilizzata per identificarli in maniera univoca. Così facendo viene rispettato il vincolo per cui, data una scheda, una sua tabella e un posizione in quest’ultima è univocamente determinato l'esercizio che vi compare.

Per modellare il concetto di ripetizione settimanale è stata creata una entità **settimana** che, in associazione con l’**esercizio in scheda** permette di memorizzare le ripetizioni per tale settimana. Rimane il vincolo inespresso per cui il numero di settimane associate a una scheda deve essere pari alla durata della scheda e ogni esercizio in scheda deve essere associato a tutte le settimane della scheda di cui fa parte.

Peso (0-1)

Nome

Descrizione

CodiceObiettivo

Nome

Nome

Descrizione

Serie

Ripetizione

Recupero

Ordinale

Posizione

Posizione

Genere

Durata

Codice

SCHEDA

SUDDIVISIONE

COMPOSIZIONE

SCHEDA

OBIETTIVO

SCHEDA

SETTIMANA

TABELLA

ESERCIZIO

IN TABELLA

ESERCIZIO

COMPOSIZIONE

TABELLA

TIPOLOGIA

OBIETTIVO

ESERCIZIO

RIPETIZIONE

RIGUARD

~~A~~

~~MU~~

SCOLO

OBIETTIVO

(1-N)

(1-1)

(1-N)

(1-N)

(1-N)

(1-1)

(1-N)

(1-1)

(1-1)

(0-N)

(0-N)

(0-N)

(0-N)

(0-N)

(0-N)

(0-N)

*Fig. 1.4 Schema E/R con la modellazione della composizione delle schede*

## Schema finale

DataIscrizione

DataSospensione

DataRiattivazione

Sconto

EFFETTUAZIONE

Telefono

(0-1)

Mail

(0-1)

CLIENTE

DataOra

INGRESSO

(0-N)

(1-1)

DataAcquisto

(0-N)

ACQUISIZIONE

(1-1)

Sospensione

(0-1)

Prezzo

Durata

(1-1)

TIPOLOGIA

(0-N)

ABBONAMENTO

COMPOSIZIONE

Anno

LISTINO

(1-1)

(0-N)

POSSEDIMENTO

ATTUALE

(0-1)

(0-N)

POSSEDIMENTO

TIPO

ABBONAMENTO

Descrizione

CodiceObiettivo

Nome

Descrizione

Nome

ESERCIZIO

~~GUARDA~~

RI

~~MUSCOLO~~

(0-N)

(0-N)

OBIETTIVO

OBIETTIVO

ESERCIZIO

(0-N)

(0-N)

# Progettazione logica

## Stima del volume dei dati

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **Concetto** | **Costrutto** | **Volume** | | Cliente | E | 10.000 | | Ingresso | E | 1.000.000 | | Effettuazione | R | 1.000.000 | |  |  |  | | Abbonamento | E | 30.000 | | Acquisizione | R | 30.000 | | Tipo Abbonamento | E | 40 | | Tipologia | R | 30.000 | | Listino | E | 10 | | Composizione | R | 40 | |  |  |  | | Istruttore | E | 10 | | Allenamento | E | 15.000 | | Assiste | R | 15.000 | | Prenotazione | R | 15.000 | |  |  |  | | Scheda Posseduta | E | 100.000 | | Possedimento attuale | R | 10.000 | | Possedimento | R | 100.000 | | Scheda | E | 50.000 | | Compilazione | R | 50.000 | | Produce | R | 100.000 | | Tabella | E | 100.000 | | Composizione scheda | R | 100.000 | | Esercizio in tabella | E | 800.000 | | Tipologia Esercizio | R | 800.000 | | Composizione tabella | R | 800.000 | | |  |  |  | | --- | --- | --- | | **Concetto** | **Costrutto** | **Volume** | | Ripetizione | R | 24.000.000 | | Settimana | E | 300.000 | | Suddivisione | R | 300.000 | |  |  |  | | Esercizio | E | 500 | | Muscolo | E | 20 | | Riguarda | R | 200 | |  |  |  | | Obiettivo | E | 30 | | Obiettivo Scheda | R | 25.000 | | Obiettivo Esercizio | R | 1.000 | |

## Descrizione delle operazioni principali e stima della loro frequenza

Le operazioni da effettuare sono quelle già elencate nella fase di analisi. Segue una tabella riportante la loro descrizione e relativa frequenza:

|  |  |  |
| --- | --- | --- |
| **Codice** | **Operazione** | **Frequenza** |
| 1 | Iscrivere un nuovo cliente | 2 al giorno |
| 2 | Aggiungere un nuovo istruttore | 1 all’anno |
| 3 | Registrare l’ingresso di un cliente, controllando la validità dell’abbonamento | 100 al giorno |
| 4 | Registrare l’acquisto di un abbonamento da parte di un cliente | 5 al giorno |
| 5 | Cambiare il listino prezzi per l’anno corrente | 1 all’anno |
| 6 | Sospendere l’abbonamento di un cliente | 5 al mese |
| 7 | Assegnare una scheda già esistente ad un cliente | 10 al giorno |
| 8 | Compilare una nuova scheda | 1 al mese |
| 9 | Leggere tutte le tabelle della scheda attuale di un cliente | 100 al giorno |
| 10 | Data la tabella di una scheda visualizzare tutti i suoi esercizi e ripetizioni | 200 al giorno |
| 11 | Visualizzare gli esercizi in base a particolari filtri | 1 al mese |
| 12 | Registrare la prenotazione di una sessione di allenamento individuale | 2 al giorno |
| 13 | Mostrare il prossimo allenamento di un cliente o un istruttore | 40 al giorno |

## Schemi di navigazione e tabelle degli accessi

Sono riportate in seguito le tabelle degli accessi delle operazioni sopra riportate; inoltre, ove non risulti banale, sono stati inseriti i relativi schemi di navigazione. Al fine del calcolo degli costi, si considerano di peso doppio gli accessi in scrittura rispetto a quelli in lettura.

### OP 1 - Iscrivere un nuovo cliente

|  |  |  |
| --- | --- | --- |
| **Concetto Costrutto Accessi** | | **Tipo** |
| Cliente E 1 | | S |
| **Totale:** 1S → 2 al giorno | |  |
| **OP 2 - Aggiungere un nuovo istruttore** | |  |
| **Concetto** | **Costrutto Accessi** | **Tipo** |
| Istruttore | E 1 | S |
|  | **Totale:** 1S → 2 all’anno |  |

### OP 3 - Registrare l’ingresso di un cliente

Prima di permettere l’ingresso di un cliente bisogna verificare che questo disponga di un abbonamento valido, questo comporterà dover leggere gli abbonamenti del cliente che, in media, sono 3.

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Effettuazione | R | 1 | S |
| Ingresso | E | 1 | S |
| Acquisizione | R | 3 | L |
| Abbonamento | E | 3 | L |
|  |  | **Totale:** 2S + 6L → 1000 al giorno |  |

### OP 4 - Registrare l’acquisto di un abbonamento da parte di un cliente

Per poter portare a termine tale operazione, è necessario prima verificare che il cliente che sta acquisendo l’abbonamento non disponga di uno non ancora scaduto. Questo comporta ulteriori letture degli abbonamenti di un cliente.

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Acquisizione | R | 3 | L |
| Abbonamento | E | 3 | L |
| Acquisizione | R | 1 | S |
| Abbonamento | E | 1 | S |
| Tipologia | R | 1 | S |
|  |  | **Totale:** 3S + 6L → 60 al giorno |  |

### OP 5 - Cambiare il listino prezzi per l’anno corrente

Si considera che, al momento della aggiunta del nuovo listino, vengano aggiunte anche le tipologie di abbonamento di cui si compone che, in media, sono 4 per listino.

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Listino | E | 1 | S |
| Composizione | R | 4 | S |
| Tipo abbonamento | E | 4 | S |
|  |  | **Totale:** 9S → 18 all’anno |  |

### OP 6 - Sospendere l’abbonamento di un cliente

Per sospendere l’abbonamento di un cliente è necessario verificare che ve ne sia uno valido che possa essere sospeso, questo comporta la lettura degli abbonamenti del cliente.

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Acquisizione | R | 3 | L |
| Abbonamento | E | 3 | L |
| Abbonamento | E | 1 | S |
|  |  | **Totale:** 1S + 6L → 40 al mese |  |

### OP 7 - Assegnare una scheda già esistente ad un cliente

Si assume di conoscere già il codice della scheda da assegnare e il codice del cliente a cui assegnarla. Nel momento in cui viene assegnata una scheda, si scrive sia nella relazione possedimento che in possedimento attuale (andando a sostituire una eventuale scheda presente in precedenza). Inoltre quando si inserisce una scheda, bisogna leggere quella che si sta andando a sostituire per scrivere il corretto valore nel campo ordinale della nuova scheda.

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| PossedimentoAttuale | R | 1 | L |
| Scheda posseduta | E | 1 | L |
| Scheda posseduta | E | 1 | S |
| Possedimento | R | 1 | S |
| Possedimento Attuale | R | 1 | S |
|  |  | **Totale:** 3S + 2L → 800 al giorno |  |

### OP 8 - Compilare una nuova scheda

Compilare una nuova scheda è un processo che si compone di diverse fasi: bisogna innanzitutto creare una scheda e tante entità settimana a essa associate pari alla sua durata; dopodiché è necessario creare le tabelle della scheda (in media sono 2 per scheda) e per ognuna inserirvi degli esercizi (in media 8 per tabella). Aggiunti gli esercizi, per ciascuno bisogna indicare una ripetizione per ogni settimana della tabella.

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Scheda | E | 1 | S |
| Suddivisione | R | 6 | S |
| Settimana | E | 6 | S |
| Composizione scheda | R | 2 | S |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Tabella | E | 2 | S |
| Composizione tabella | R | 16 | S |
| Esercizio in tabella | E | 16 | S |
| Tipologia | R | 16 | S |
| Ripetizione | R | 96 | S |
| Obbiettivo scheda | R | 1 | S |
|  |  | **Totale:** 162S → 324 al mese |  |

### OP 9 - Leggere tutte le tabelle della scheda attuale di un cliente

Innanzitutto è necessario reperire la scheda attuale di un cliente, questo richiederà una lettura in Possedimento Attuale, dopodiché si dovranno leggere le tabelle di tale scheda

Nome

Data

Genere

Durata

Codice

Ordinale

CLIENTE

ISTRUTTORE

POSSEDIMENTO

POSSEDIMENTO

ATTUALE

SCHEDA

POSSEDUTA

SCHEDA

COMPILAZIONE

PRODUCE

(0-1)

(0-N)

(0-1)

(1-1)

(1-1)

(1-1)

(0-N)

(0-N)

Posizione

COMPOSIZIONE

SCHEDA

TABELLA

(1-N)

(1-1)

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Possedimento Attuale | R | 1 | L |
| **Concetto Costrutto Accessi** | | | **Tipo** |
| Scheda Posseduta E 1 | | | L |
| Produce R 1 | | | L |
| Scheda E 1 | | | L |
| Composizione R 2 scheda | | | L |
| Tabella E 2 | | | L |
| **Totale:** 8L → 800 al giorno | | |  |
| **OP 10 - Data la tabella di una scheda visualizzare tutti i suoi esercizi** | | |  |
| **Concetto Costrutto Accessi** | | | **Tipo** |
| Composizione tabella R 8 | | | L |
| Esercizio in tabella E 8 | | | L |
| Ripetizione E 48 | | | L |
| **Totale:** 64L → 12.800 al giorno | | |  |

### OP 11 - Visualizzare gli esercizi in base a particolari filtri

Considero il caso generico di un filtro sul muscolo e uno sull’obiettivo, in tal caso gli accessi effettuati saranno:

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Riguarda | R | 10 | L |
| Esercizio | E | 10 | L |
| Obiettivo Esercizio | E | 2 | L |
| Esercizio | R | 2 | L |
|  |  | **Totale:** 24L → 24 al mese |  |

### OP 12 - Registrare la prenotazione di una sessione di allenamento individuale

Supponendo di Conoscere il codice del cliente e dell’istruttore da inserire nella tabella sarà sufficiente una scrittura dell’entità Allenamento.

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Allenamento | E | 1 | S |
| Assiste | R | 1 | S |
| Prenotazione | R | 1 | S |
|  |  | **Totale:** 3S → 12 al giorno |  |

### OP 13 - Mostrare il prossimo allenamento di un cliente o un istruttore

Si analizza il costo per una sola delle due query che risultano perfettamente simmetriche.

Genere

DataNascita

CodiceFiscale

Codice

Nome

Cognome

DataIscrizione

Telefono

(0-1)

Mail

(0-1)

CLIENTE

PERSONA

ISTRUTTORE

Giorno

ALLENAMENTO

OraInizio

OraFine

ASSISTE

PRENOTAZIONE

(0-N)

(0-N)

(1-1)

(1-1)

(

)

t, e

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Allenamento | E | 1.5 | L |
| Assiste | R | 1 | L |
| Istruttore | E | 1 | L |
|  |  | **Totale:** 3.5L → 140 al giorno |  |

## Raffinamento dello schema

### Eliminazione delle gerarchie

Per l’eliminazione della gerarchia persona si è scelto di adottare l’approccio del collasso verso il basso, replicando così gli attributi in istruttore e cliente. Si è adottata questa strategia in quanto si deve interagire con i clienti molto più spesso che con gli istruttori, e non si ha la necessità che l’identificatore per tali entità sia globalmente univoco.

### Eliminazione degli attributi compositi

Nello schema è presente un attributo composito nell’entità abbonamento che è stato diviso nelle sue sotto-componenti. Sarà poi necessario accertarsi, a livello di applicazione, che tali attributi siano sempre entrambi impostati a un valore coerente o entrambi null.

### Scelta delle chiavi primarie

Nello schema sono già evidenziate senza ambiguità tutte le chiavi primarie per la maggior parte delle entità; per quanto riguarda l’entità scheda, si sceglie di usare come chiave primaria il codice mentre per l’allenamento la chiave primaria sarà data dal giorno e l’associazione con cliente.

### Eliminazione degli identificatori esterni

Nello schema E/R sono eliminate le seguenti relazioni:

* Prenotazione, importando codiceCliente in Allenamento
* Assiste, importando codiceIstruttore in Allenamento
* Effettuazione, importando codiceCliente in Ingresso
* Acquisizione, importando codiceCliente in Abbonamento
* ComposizioneListino, importando anno in TipoAbbonamento
* Tipologia, importando durata e anno in Abbonamento
* Produce, importando codiceScheda in SchedaPosseduta
* PossedimentoAttuale, importando codiceScheda e ordinale in Cliente
* Possedimento, importando codiceCliente in SchedaPosseduta
* Compilazione, importando codiceIstruttore e l’attributo data in Scheda
* Suddivisione, importando codiceScheda in Settimana
* ComposizioneScheda importando codiceScheda in Tabella
* ComposizioneTabella, importando posizioneTabella e codiceScheda in EsercizioInTabella
* Ripetizione, reificata importando codiceScheda, ordinaleSettimana da Settimane e posizioneTabella, posizioneEsercizio da EsercizioInTabella
* Tipologia, importando nomeEsercizio in EsercizioInTabella
* ObiettivoEsercizio, reificata importando nomeEsercizio da Esercizio e codiceObiettivo da Obiettivo
* ObiettivoScheda, reificata importando codiceScheda da Scheda e codiceObiettivo da Obiettivo
* Riguarda, reificata importando nomeEsercizio da Esercizio e nomeMuscolo da Muscolo

## Analisi delle ridondanze

È stata inserita una ridondanza tramite l’uso dell’associazione Possedimento Attuale, in quanto sarebbe sufficiente l’associazione Possedimento per risalire alla scheda attuale di un cliente, effettuando un ordinamento delle schede possedute per ordinale.

È riportata la valutazione del risparmio in termini di accessi dato dall’uso di questo approccio:

### OP 7 - Assegnare una scheda già esistente ad un cliente

L’avere una ridondanza in questo caso aiuta nell’inserimento: infatti, poiché la scheda attuale dispone di un ordinale, inserire il successivo nella nuova scheda pu essere fatto tramite una lettura.

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Possedimento Attuale | R | 1 | L |
| Scheda posseduta | E | 1 | L |
| Scheda posseduta | E | 1 | S |
| Possedimento | R | 1 | S |
| Possedimento Attuale | R | 1 | S |
|  |  | **Totale:** 3S + 2L → 800 al giorno |  |

Senza ridondanza sarebbe invece necessario leggere tutte le schede possedute dal cliente e individuare quella con ordinale maggiore per stabilire il prossimo numero da inserire:

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Possedimento | R | 10 | L |
| Scheda posseduta | E | 10 | L |
| Scheda posseduta | E | 1 | S |
| Possedimento | R | 1 | S |
| Possedimento Attuale | R | 1 | S |
| Scheda attuale | E | 1 | S |
|  |  | **Totale:** 4S + 20L → 2800 al giorno |  |

### OP 9 - Leggere tutte le tabelle della scheda attuale di un cliente

L’associazione ridondante Possedimento Attuale permette di ottenere con due sole letture il codice della scheda attuale del cliente. Senza ridondanza (seconda tabella) sarebbe necessario leggere tutte le schede del cliente per stabilire quale sia quella con ordinale maggiore.

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Possedimento Attuale | R | 1 | L |
| Scheda Posseduta | E | 1 | L |
| Produce | R | 1 | L |
| Scheda | E | 1 | L |
| Composizione scheda | R | 2 | L |
| Tabella | E | 2 | L |
|  |  | **Totale:** 8L → 800 al giorno |  |
|  |  |  |  |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Possedimento Attuale | R | 10 | L |
| Scheda Posseduta | E | 10 | L |
| Produce | R | 1 | L |
| Scheda | E | 1 | L |
| Composizione scheda | R | 2 | L |
| Tabella | E | 2 | L |
|  |  | **Totale:** 26L → 2600 al giorno |  |

Si pu vedere come in entrambi i caso la ridondanza garantisca un notevole vantaggio in termini di accessi risparmiati, per questo verrà mantenuta.

## Traduzione di entità e associazioni in relazioni

abbonamenti(codiceCliente, dataAcquisto, sconto, dataSospensione, dataRiattivazione, annoListino, durata)

allenamenti(codiceCliente: clienti, giorno, codiceIstruttore: istruttori, oraInizio, oraFine) UNIQUE(codiceIstruttore, giorno) clienti(codice, nome, cognome, CF, dataNascita, genere, ordinaleSchedaAttuale\*:

schede\_possedute.ordinale, codiceSchedaAttuale\*: schede\_possedute.codice, telefono\*, mail\*)

esercizi(nome, descrizione\*)

esercizi\_in\_tabella((codiceScheda, posizioneTabella): tabelle, posizioneEsercizio, nomeEsercizio: esercizi)

ingressi(codiceCliente: clienti, dataOra)

istruttori(codice, nome, cognome, CF, dataNascita, genere) listini(anno) muscoli(nome)

obiettivi(codiceObiettivo, descrizione)

obiettivi\_esercizi(nomeEsercizio: esercizi, codiceObiettivo: obiettivi) obiettivi\_schede(codiceScheda: schede, codiceObiettivo: obiettivi) riguarda(nomeEsercizio: esercizi, nomeMuscolo: muscoli)

ripetizioni((posizioneTabella, posizioneEsercizio, codiceScheda): esercizi\_in\_tabella, ordinale\_settimana: settimane, serie, ripetizione, peso\*, recupero)

schede(codiceScheda, nome, durata, genere, dataCompilazione, codiceIstruttore: istruttori)

UNIQUE(nome)

schede\_possedute(codiceCliente: clienti, ordinale, codiceScheda: schede) settimane(codiceScheda: schede, ordinale) tabelle(codiceScheda: schede, posizioneTabella) tipi\_abbonamento(annoListino: listini, durata, prezzo)

23

## Schema relazionale finale

**ISTRUTTORE**

CodiceIstruttore

Nome

Cognome

CF

Genere

DataNascita

id: CodiceIstruttore

**SCHEDA**

CodiceScheda

Nome

Durata

Genere

DataCompilazione

CodiceIstruttore

id: CodiceScheda

id’: Nome

ref: CodiceIstruttore

**TABELLA**

CodiceScheda

PosizioneTabella

id: CodiceScheda

PosizioneTabella

equ: CodiceScheda

**OBIETTIVO**

CodiceObiettivo

Descrizione

id: CodiceObiettivo

**ESERCIZIO-IN-TABELLA**

CodiceScheda

PosizioneTabella

PosizioneEsercizio

NomeEsercizio

id: CodiceScheda

PosizioneTabella

PosizioneEsercizio

equ: CodiceScheda

PosizioneTabella

ref: NomeEsercizio

**RIPETIZIONE**

PosizioneTabella

PosizioneEsercizio

CodiceScheda

OrdinaleSettimana

Serie

Ripetizione

Recupero

id: PosizioneTabella

PosizioneEsercizio

CodiceScheda

OrdinaleSettimana

equ: CodiceScheda

OrdinaleSettimana

equ: CodiceScheda

PosizioneTabella

PosizioneEsercizio

**MUSCOLO**

Nome

id: Nome

**RIGUARDA**

NomeMuscolo

NomeEsercizio

id: NomeEsercizio

NomeMuscolo

ref: NomeEsercizio

ref: NomeMuscolo

**ESERCIZIO**

Nome

Descrizione

[0-1]

id: Nome

**OBIETTIVO-ESERCIZIO**

NomeEsercizio

CodiceObiettivo

id: NomeEsercizio

CodiceObiettivo

ref: CodiceObiettivo

ref: NomeEsercizio

**OBIETTIVO-SCHEDA**

CodiceObiettivo

CodiceScheda

id: CodiceScheda

CodiceObiettivo

ref: CodiceScheda

ref: CodiceObiettivo

**ALLENAMENTO**

CodiceCliente

Giorno

CodiceIstruttore

OraInizio

OraFine

id: CodiceCliente

Giorno

id’: CodiceCliente

Istruttore

ref: CodiceCliente

ref: CodiceIstruttore

**CLIENTE**

CodiceCliente

Nome

Cognome

CF

Genere

DataIscrizione

CodiceSchedaAttuale [0-1]

OrdinaleSchedaAttuale [0-1]

Telefono [0-1

]

Mail [0-1]

id: CodiceCliente

**SCHEDA-POSSEDUTA**

CodiceCliente

Ordinale

CodiceScheda

id: CodiceCliente

Ordinale

ref: CodiceScheda

ref: CodiceCliente

**SETTIMANA**

CodiceScheda

Ordinale

id: CodiceScheda

Ordinale

equ: CodiceScheda

**ABBONAMENTO**

CodiceCliente

DataAcquisto

Sconto [0-1]

DataSospensione [0-1]

DataRiattivazione [0-1]

AnnoListino

Durata

id: CodiceCliente

DataAcquisto

ref: CodiceCliente

ref: AnnoListino

Durata

**TIPO-ABBONAMENTO**

AnnoListino

Durata

Prezzo

id: AnnoListino

Durata

ref: AnnoListino

**LISTINO**

Anno

id: Anno

**INGRESSO**

CodiceCliente

DataOra

id: CodiceCliente

DataOra

ref: CodiceCliente

## Traduzione delle operazioni in query SQL

### OP 1 - Iscrivere un nuovo cliente

**INSERT INTO** clienti (nome, cognome, CF, dataNascita, genere,

telefono, mail) **VALUES** (?, ?, ?, ?, ?, ?, ?)

### OP 2 - Aggiungere un nuovo istruttore

**INSERT INTO** istruttori (nome, cognome, CF, dataNascita, genere)

**VALUES** (?, ?, ?, ?, ?)

### OP 3 - Registrare l’ingresso di un cliente

Per verificare se il cliente dispone di un abbonamento valido la seguente query deve restituire un record, questo sarà l’unico abbonamento valido (si ha la certezza che ci sia un solo abbonamento valido per come viene gestita l’aggiunta di nuovi abbonamenti acquisiti).

**SELECT** \*

**FROM** abbonamenti **WHERE** codiceCliente = ?

**AND** DATEDIFF(NOW(), dataAcquisto) - **IF**(dataSospensione **IS** **NOT** **NULL**,

DATEDIFF(dataRiattivazione, dataSospensione), 0) < durata

**AND** **IF**(dataSospensione **IS** **NOT** **NULL**, (NOW() < dataSospensione **OR**

NOW() >= dataRiattivazione), true)

Se tale condizione risulta soddisfatta, si pu aggiungere il nuovo ingresso col codice del cliente che potrà entrare in palestra.

**INSERT INTO** ingressi (codiceCliente)

**VALUES** (?)

### OP 4 - Registrare l’acquisto di un abbonamento da parte di un cliente

Si pu registrare un abbonamento solo se il cliente non dispone già di abbonamenti validi, ovvero se la seguente query non ritorna alcun record:

**SELECT** \*

**FROM** abbonamenti **WHERE** codiceCliente = ?

**AND** DATEDIFF(NOW(), dataAcquisto) - **IF**(dataSospensione **IS** **NOT** **NULL**,

DATEDIFF(dataRiattivazione, dataSospensione), 0) <

DATEDIFF(DATE\_ADD(dataAcquisto, **INTERVAL** durata **MONTH**), dataAcquisto)

Appurata la possibilità di comprare un nuovo abbonamento si potrà aggiungere.

**INSERT INTO** abbonamenti (codiceCliente, sconto, durata, annoListino)

**VALUES** (?, ?, ?, **SELECT** anno

**FROM** listini

**ORDER BY** anno **DESC**

**LIMIT** 1)

Così facendo si forza l’acquisto di tipologie di abbonamento che sono presenti solo nel listino in uso, ovvero quello più recente.

**OP 5 - Cambiare il listino prezzi per l’anno corrente** L’operazione per aggiungere un listino è la seguente:

**INSERT INTO** listini (anno)

**VALUES** (YEAR(NOW()))

Per aggiungere tipologie di abbonamento a un listino:

**INSERT INTO** tipiAbbonamento

**VALUES** (YEAR(NOW()), ?, ?)

Si presuppone che tutti gli abbonamenti vengano aggiunti al momento della creazione del listino (o al più nel medesimo anno) e che, ovviamente, un listino si riferisca all’anno corrente nel quale viene creato.

### OP 6 - Sospendere l’abbonamento di un cliente

Il cliente deve disporre di un abbonamento valido che non sia già stato sospeso una volta (in quanto proibito sospenderlo per più di una volta), dunque sarà necessario eseguire la seguente query che dovrà restituire un solo risultato, ovvero il record relativo all’abbonamento da modificare:

**SELECT** \*

**FROM** abbonamenti **WHERE** codiceCliente = ?

**AND** DATEDIFF(NOW(), dataAcquisto) < DATEDIFF(DATE\_ADD(dataAcquisto,

**INTERVAL** durata **MONTH**), dataAcquisto)

**AND** dataSospensione **IS** **NULL**

**UPDATE** abbonamenti

**SET** dataSospensione = ?, dataRiattivazione = ? **WHERE** codiceCliente = ?

**AND** dataAcquisto = ?

### OP 7 - Assegnare una scheda già esistente ad un cliente

Le query per eseguire l’assegnazione sono le seguenti e devono essere eseguite nell’ordine riportato affinché i numeri ordinali delle schede risultino corretti.

**UPDATE** clienti

**SET** ordinaleSchedaAttuale = ordinaleSchedaAttuale + 1 **WHERE** codiceCliente = ?

**INSERT** **INTO** schedePossedute (codiceCliente, codiceScheda, ordinale)

**VALUES** (?, ?, **SELECT** ordinaleSchedaAttuale

**FROM** clienti

**WHERE** codiceCliente = ?)

**UPDATE** clienti **SET** codiceSchedaAttuale = ?

**WHERE** codiceCliente = ?

### OP 8 - Compilare una nuova scheda

**INSERT INTO** schede (nome, durata, genere, codiceIstruttore)

**VALUES**(?, ?, ?, ?)

Dopodiché, vanno inserite tante settimane per quanto indicato dalla durata, quindi bisognerà eseguire a livello applicativo la seguente query più volte, incrementando di volta in volta l’ordinale della settimana:

**INSERT INTO** settimane

**VALUES** (LAST\_INSERT\_ID(), ?)

Infine si aggiungerà alla scheda tante tabelle quanto desiderato come segue:

**INSERT INTO** tabelle

**VALUES** (LAST\_INSERT\_ID(), ?)

E a ogni tabella si aggiungerà quanti esercizi si vuole aggiungendo anche una ripetizione per ogni settimana.

**INSERT INTO** esercizi\_in\_tabella

**VALUES** (LAST\_INSERT\_ID(), ?, ?, ?)

**INSERT INTO** ripetizioni

**VALUES** (?, ?, LAST\_INSERT\_ID(), ?, ?, ?, ?)

### OP 9 - Leggere tutte le tabelle della scheda attuale di un cliente

**SELECT** \*

**FROM** tabelle **WHERE** codiceScheda = ?

### OP 10 - Data la tabella di una scheda visualizzare tutti i suoi esercizi

**SELECT** posizioneEsercizio, nomeEsercizio

**FROM** esercizi\_in\_tabella **WHERE** codiceScheda = ?

**AND** posizioneTabella = ?

**ORDER BY** posizioneEsercizio **ASC**

### OP 11 - Visualizzare gli esercizi in base a particolari filtri

In generale è richiesta la possibilità di filtrare gli esercizi in base ai loro obiettivi o ai muscoli che riguardano, dunque una generica query di filtraggio potrebbe essere:

**SELECT** E.nome, E.descrizione

**FROM** esercizi E, riguarda R, obiettivi\_esercizi O

**WHERE** R.nomeEsercizio = E.nome

**AND** O.nomeEsercizio = E.nome **AND** R.nomeMuscolo = ?

**OR** codiceObiettivo = ?

### OP 12 - Registrare la prenotazione di una sessione di allenamento individuale

**INSERT** **INTO** allenamenti

**VALUES** (?, ?, ?, ?, ?)

**OP 13 - Mostrare il prossimo allenamento di un cliente o un istruttore** Query per gli istruttori che permette di visualizzare il nome del cliente:

**SELECT** C.nome, C.cognome, A.\*

**FROM** allenamenti A, clienti C

**WHERE** C.codiceCliente = A.codiceCliente **AND** codiceIstruttore = ?

**AND** giorno > NOW()

**ORDER** **BY** giorno

**LIMIT** 1

Query per i clienti che permette di visualizzare il nome dell’istruttore:

**SELECT** I.nome, I.cognome, A.\*

**FROM** allenamenti A, istruttori I

**WHERE** I.codiceIstruttore = A.codiceIstruttore **AND** codiceCliente = ?

**AND** giorno > NOW()

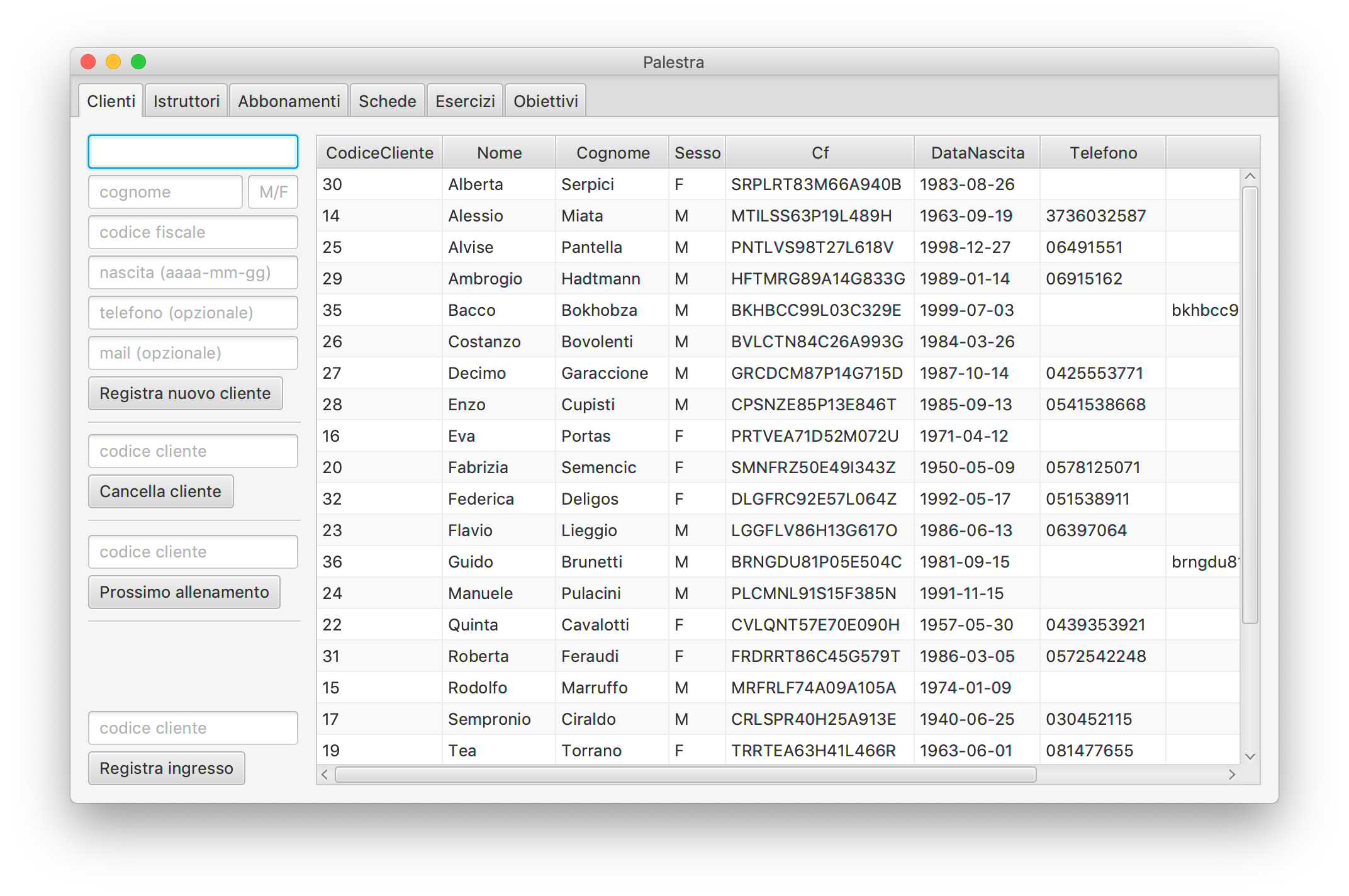
**ORDER** **BY** giorno

**LIMIT** 1

# Progettazione dell’applicazione

## Descrizione dell’architettura dell’applicazione realizzata

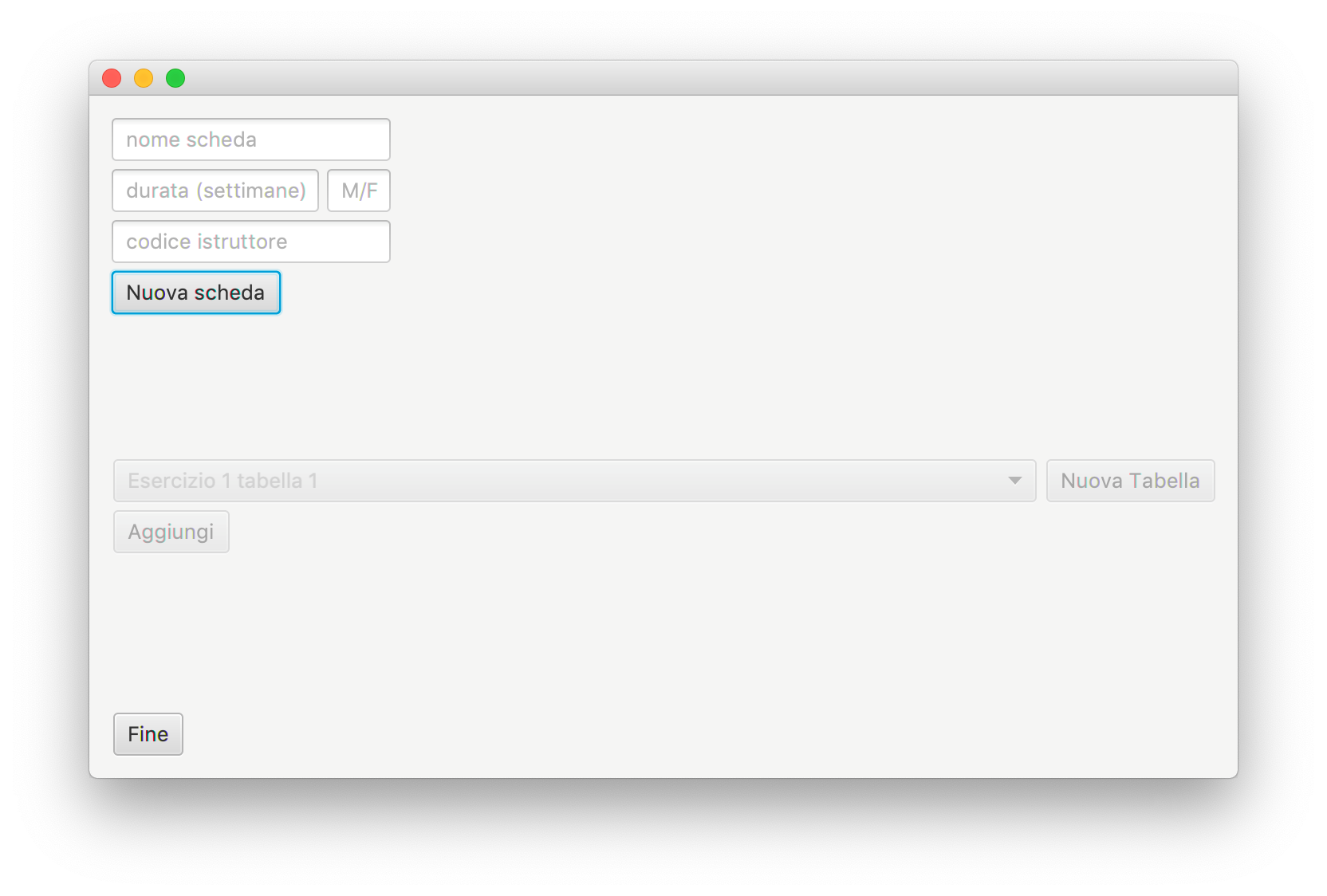
L’applicazione per interfacciarsi al database è stata realizzata in Java, sfruttando lo strumento di ORM Hibernate; il database risiede in locale e il DBMS usato è mySQL. L’applicazione è una semplice JavaFX application che fa uso di file FXML (uno per ogni finestra indipendente) associati a un proprio controller il quale ha il compito di portare a termine le query appoggiandosi alla classe DatabaseConnection.



*Fig 1.5 Schermata principale dell’applicazione per aggiungere clienti*

Per la maggior parte delle query, la correttezza dei dati inseriti viene verificata dal DBMS sfruttando opportuni check, così facendo si è cercato di ridurre al minimo i controlli che l’applicazione deve effettuare.

Nella scheda relativa ai clienti è stato inserito, per completezza, un form per poter registrare nuovi ingressi da parte dei clienti; tuttavia questa operazione dovrebbe essere effettuata automaticamente tramite la connessione a un dispositivo di lettura di codici a barre (nelle schede dei clienti).



*Fig 1.6 Schermata per la creazione di schede*

L’inserimento di una scheda è un processo complesso diviso in più passaggi, per questo si ricorre all’uso di una seconda finestra pop-up (con un proprio controller distinto) grazie alla quale è possibile effettuare il rollback delle query di inserimento qualora, in un qualunque passaggio, si dovesse verificare un errore. Inoltre impedisce all’utente di creare schede mal formate dove per esempio una tabella non possiede esercizi.

L’applicazione fornisce nel complesso le funzionalità richieste evidenziate nella fase di progettazione, includendo ulteriormente alcune banali operazioni non elencate come la cancellazione di un cliente o di un istruttore.