|  |  |
| --- | --- |
|  |  |

Basi di Dati

Progetto A.A. 2022/2023

TITOLO DEL PROGETTO

Matricola

Luca Martorelli

**Indice**

[1. Descrizione del Minimondo 3](#_Toc606296459)

[2. Analisi dei Requisiti 4](#_Toc1289394997)

[3. Progettazione concettuale 5](#_Toc2081466291)

[4. Progettazione logica 6](#_Toc2147004904)

[5. Progettazione fisica 8](#_Toc518560220)

[Appendice: Implementazione 9](#_Toc403811585)

Tutto il testo su sfondo grigio, all’interno di questo template, deve essere eliminato prima della consegna. Viene utilizzato per fornire informazioni sulla corretta compilazione del report di progetto.

Non modificare il formato del documento:

- Carattere: Times New Roman, 12pt

- Dimensione pagina: A4

- Margini: superiore/inferiore 2,5cm, sinistro/destro: 1,9cm

# Descrizione del Minimondo

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | Si vuole realizzare un’applicazione per consentire, ai personal trainer di una palestra, di gestire le schede degli esercizi dei propri clienti. Ciascun utente della palestra è associato ad un personal trainer di riferimento. Questi potrà redigere una scheda di allenamento personalizzata per ciascun utente, scegliendo gli esercizi e i macchinari da utilizzare in un insieme definito dal proprietario della palestra. Per ciascun esercizio, il personal trainer indicherà anche il numero di serie e ripetizioni dell’esercizio.  Periodicamente, quando viene redatta una nuova scheda di esercizi, la scheda precedente viene archiviata. Questa sarà ancora consultabile dagli utenti, ma essi potranno “interagire” solo con quella corrente. Gli utenti della palestra possono infatti effettuare il login nell’applicazione e mostrare quali sono gli esercizi che devono svolgere in una sessione di allenamento. Il sistema mostrerà l’esercizio e la serie corrente, permettendo all’utente di contrassegnare un dato esercizio come completato. L’atleta ha la possibilità di saltare un esercizio e passare al successivo.  I personal trainer possono generare un report che mostra, per tutti i clienti a loro assegnati, quanti sono gli allenamenti sostenuti in un intervallo temporale richiesto, qual è la percentuale di completamento delle schede di allenamento in ogni sessione di allenamento e quanto tempo è durato ciascun allenamento. |

# Analisi dei Requisiti

## Identificazione dei termini ambigui e correzioni possibili

Compilare la seguente tabella, facendo riferimento alla specifica del minimondo di riferimento precedentemente indicata. Individuare i termini ambigui nella specifica (indicando la linea in cui essi compaiono), indicare il nuovo termine che si intende adottare nella specifica, ed indicare il motivo del cambiamento che si propone.

|  |  |  |  |
| --- | --- | --- | --- |
| **Linea** | **Termine** | **Nuovo termine** | **Motivo correzione** |
| 3 | Questi | Il personal trainer | Termine più chiaro. |
| 4 | scegliendo gli esercizi e i macchinari da utilizzare | scegliendo gli esercizi da utilizzare da fare | I macchinari sono strumenti con i quali i clienti possono fare degli esercizi, quindi è ridondante andare a specificare sia esercizi che macchinari. |
| 8 | Questa | La scheda archiviata | Termine più chiaro. |
| 9 | quella | la scheda | Termine più chiaro. |
| 9 | infatti | // | Termine inutile nella sintassi della frase. |
| 10 | mostrare | vedere | Termine più chiaro. |
| 12 | L’atleta | Il cliente | Termine più chiaro. |

### Specifica disambiguata

|  |
| --- |
| Riportare in questo riquadro la specifica di progetto corretta, applicando le disambiguazioni proposte.  Si vuole realizzare un’applicazione per consentire, ai personal trainer di una palestra, di gestire le schede degli esercizi dei propri clienti. Ciascun utente della palestra è associato ad un personal trainer di riferimento. Il personal Trainer potrà redigere una scheda di allenamento personalizzata per ciascun utente, scegliendo gli esercizi da fare in un insieme definito dal proprietario della palestra. Per ciascun esercizio, il personal trainer indicherà anche il numero di serie e ripetizioni dell’esercizio.  Periodicamente, quando viene redatta una nuova scheda di esercizi, la scheda precedente viene archiviata. La scheda archiviata sarà ancora consultabile dagli utenti, ma essi potranno “interagire” solo con la scheda corrente. Gli utenti della palestra possono effettuare il login nell’applicazione e vedere quali sono gli esercizi che devono svolgere in una sessione di allenamento. Il sistema mostrerà l’esercizio e la serie corrente, permettendo all’utente di contrassegnare un dato esercizio come completato. L’atleta ha la possibilità di saltare un esercizio e passare al successivo.  I personal trainer possono generare un report che mostra, per tutti i clienti a loro assegnati, quanti sono gli allenamenti sostenuti in un intervallo temporale richiesto, qual è la percentuale di completamento delle schede di allenamento in ogni sessione di allenamento e quanto tempo è durato ciascun allenamento. |

## Glossario dei Termini

|  |  |  |  |
| --- | --- | --- | --- |
| **Termine** | **Descrizione** | **Sinonimi** | **Collegamenti** |
| Personal Trainer | Persone che gestiscono schede e report |  | Clienti, Schede |
| Clienti | Persone che vogliono usufruire dei servizi della palestra. | Utenti | Personal Trainer, Scheda, Scheda Attiva, Sessione di Allenamento |
| Scheda | Elenco di esercizi che un Personal Trainer organizza per un cliente. |  | Esercizi di Scheda, Personal Trainer, Clienti |
| Schede Attive | Elenco di esercizi che un cliente può contrassegnare come eseguiti o saltati. |  | Esercizi di Scheda, Personal Trainer, Clienti |
| Schede Archiviate | Elenco di esercizi che un cliente può solo vedere. |  | Esercizi di Scheda, Personal Trainer, Clienti |
| Esercizi | Attività motorie che possono essere presenti in una scheda e svolte da un cliente. |  | Esercizi di Scheda |
| Esercizi di Scheda | Attività motorie presenti sulla scheda di un cliente. |  | Esercizi di sessione, Esercizi, Scheda |
| Sessione di allenamento | Intervallo temporale nel quale un cliente esegue, o salta, gli esercizi presenti sulla scheda. |  | Esercizi di Sessione, Clienti. |
| Esercizi di Sessione | Attività motorie che se presenti sulla scheda di un cliente, quest’ultimo può saltare o eseguire. |  | Esercizi di Scheda, Sessione di allenamento. |

## Raggruppamento dei requisiti in insiemi omogenei

|  |
| --- |
| **Frasi relative a PERSONAL TRAINER** |
| I Personal Trainer sono identificati univocamente da un ID. I Personal Trainer gestiscono le schede dei clienti. |
| **Frasi relative a CLIENTI** |
| I clienti sono identificati univocamente da un Codice Fiscale. È inoltre noto il nome e cognome. Ogni cliente è associato a un Personal Trainer. I clienti possono, inoltre, sia interagire con la scheda attiva, sia visualizzare le possibili schede archiviate. |
| **Frasi relative a SCHEDA** |
| Ciascuna scheda è identificata da una data di inizio e dal cliente a cui la scheda è associata. Essa può essere attiva o archiviata. |
| **Frasi relative a SCHEDA ATTIVA** |
| Nel caso di scheda attiva, l’utente può interagirci, scegliendo se eseguire o saltare uno o più esercizi. |
| **Frasi relative a SCHEDE ARCHIVIATE** |
| Nel caso di scheda archiviata, l’utente può solo visualizzare la scheda. Per queste schede sono note anche le date di fine attività (la data in cui la scheda passa da attiva ad archiviata). |
| **Frasi relative a ESERCIZI** |
| Gli esercizi sono identificati dal loro nome. |
| **Frasi relative a SESSIONE DI ALLENAMENTO** |
| Si vuole tener traccia di tutte le sessioni di allenamento fatte da un cliente, in quale data e quanto tempo è durata. |

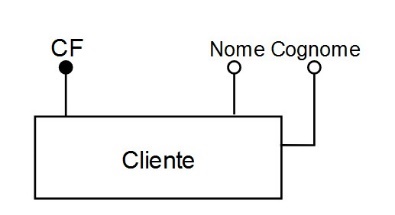
# Progettazione concettuale

## Costruzione dello schema E-R

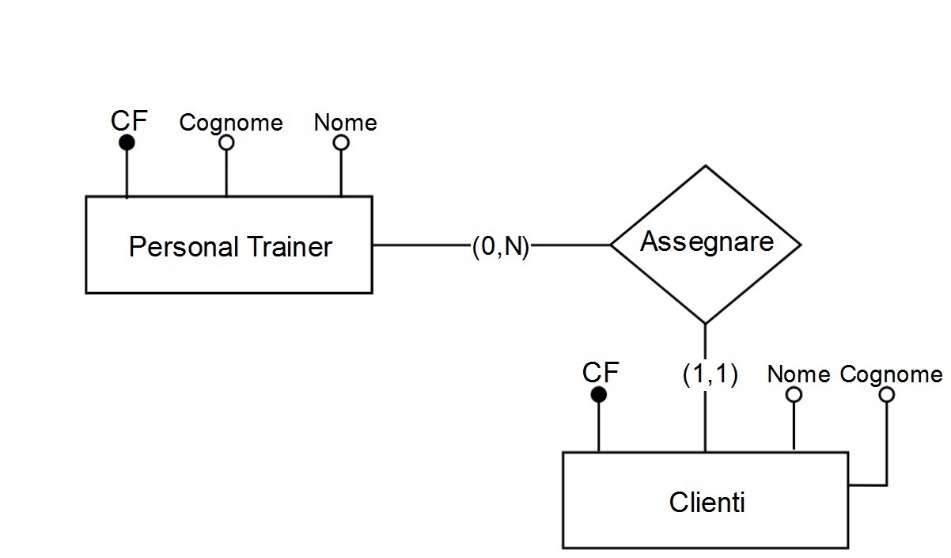
In questa sezione è necessario riportare tutti passi seguiti per la costruzione dello schema E-R finale, a partire dalle specifiche raccolte ed organizzate nel capitolo precedente. Non è richiesto un procedimento specifico: si può adottare una strategia top-down, bottom-up, a macchia d’olio o mista. L’importante è descrivere e commentare tutti i passi della costruzione, andando anche ad inserire “schemi parziali” utilizzati nel processo.

Organizziamo la progettazione concettuale della nostra base dati adottando la strategia inside-out, che consiste nello sviluppare prima i concetti più importanti (primari) in bottom-up. Gli altri concetti secondari verranno successivamente sviluppati per collegamento ai concetti primari.

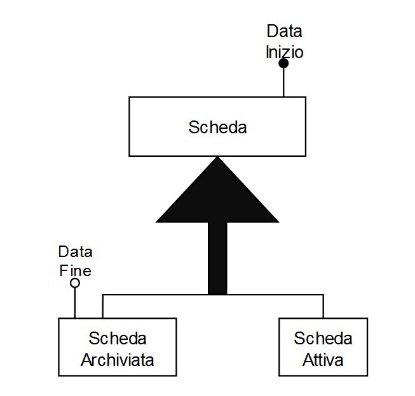
- Il primo concetto del mini-mondo di riferimento è racchiuso nell’entità Cliente. Un Cliente del sistema è identificato dal proprio codice fiscale. Di esso terremo traccia anche del suo nome e cognome. Raggruppando le specifiche che descrivono un cliente si ottiene il seguente schema:



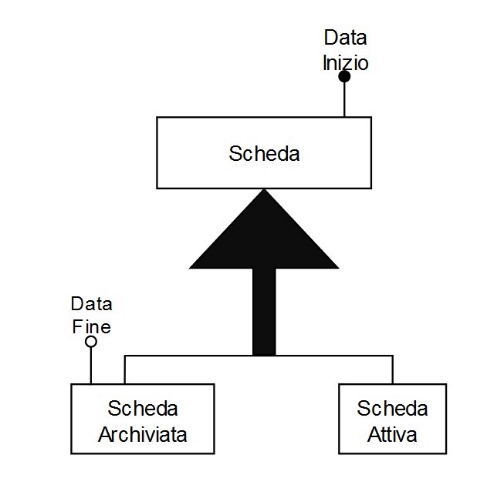
- Procediamo con l’introduzione dell’entità Personal Trainer, identificata all’interno della base di dati, come un Cliente, attraverso il suo Codice Fiscale, e di cui teniamo traccia di nome e cognome. Dalla specifica emerge che ogni Cliente è associato con un Personal Trainer, dunque risulta necessaria l’introduzione di una relazione fra le due entità precedentemente descritte. La relazione Assegnare è del tipo ‘uno a molti’, questo perché i Personal Trainer possono essere associati a più Clienti, ma, contemporaneamente ogni cliente fa riferimento ad un singolo Personal Trainer.



- Successivamente introduciamo il concetto di scheda. Ogni scheda ha una data di inizio identificante.

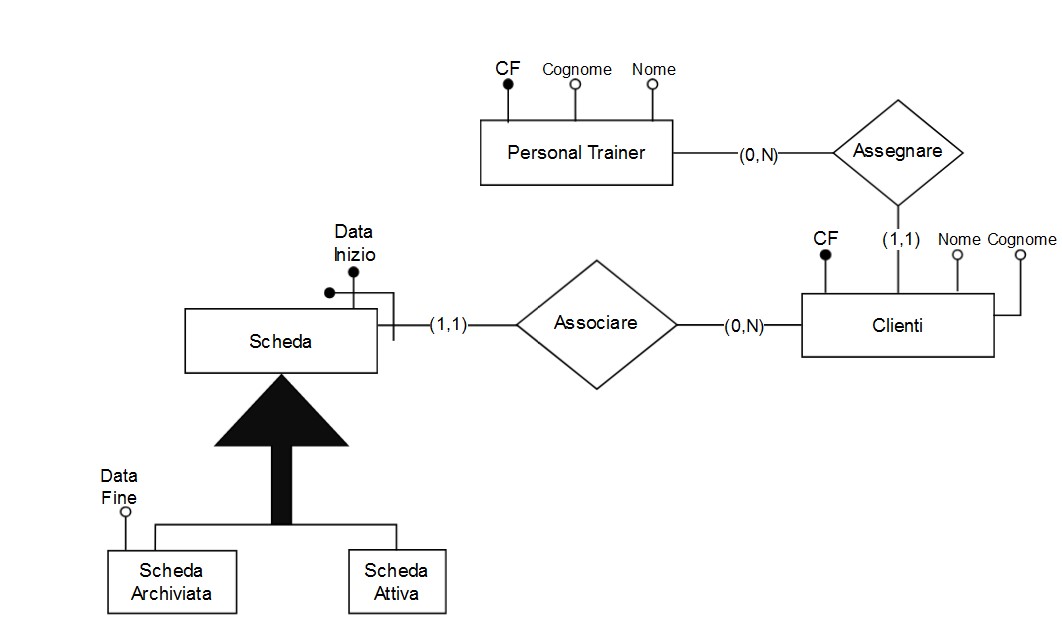


- La specifica ci illustra come le schede si possono suddividere in schede attive e schede archiviate, e per illustrare questo concetto, utilizziamo una generalizzazione. Le prime saranno completamente accessibili agli utenti, i quali potranno interagirci; mentre le seconde saranno disponibili solamente in lettura (per consultazione), e presenteranno inoltre la data di fine attivazione (giorno in cui tali schede sono passate da attive ad archiviate).

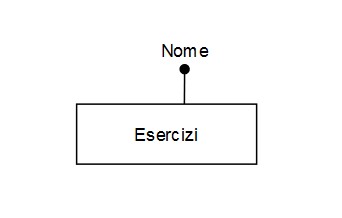


- A questo punto, però, sorge un problema: più di una scheda può avere la stessa data di inizio, perciò la chiave adottata inizialmente risulta non essere realmente identificante.

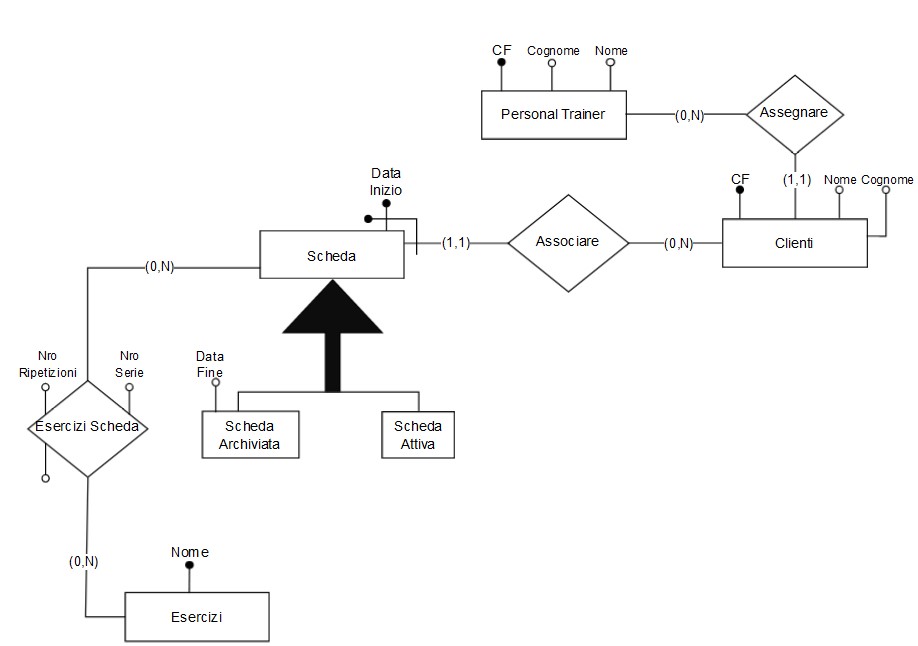
Per risolvere tale problema, utilizziamo anche una chiave esterna, identificando ogni scheda sia dalla data di inizio, che dal codice fiscale del cliente al quale tale scheda è assegnata, rendendo così Scheda entità debole verso Cliente. Tutto ciò avviene anche grazie alla relazione Associare, del tipo ‘uno a molti’, che ci permette inoltre di tenere traccia di tutte le schede che sono state associate ad un preciso cliente.



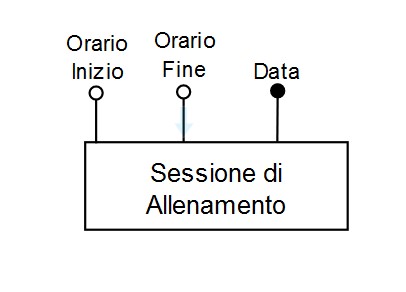
- Adesso introduciamo l’entità Esercizi. Essi sono identificati da un nome.



- La specifica ci spiega bene come questa entità è strettamente legata all’entità Scheda, precedentemente illustrata. Utilizziamo la relazione Esercizi Scheda, ‘molti a molti’, che ci permetterà di capire quali esercizi sono presenti all’interno di una determinata scheda, tenendo inoltre traccia anche di quante serie, quante ripetizioni, ogni cliente deve fare per ciascun esercizio presente nella scheda di allenamento; specificando inoltre in quale posizione della Scheda è ciascun Esercizio.

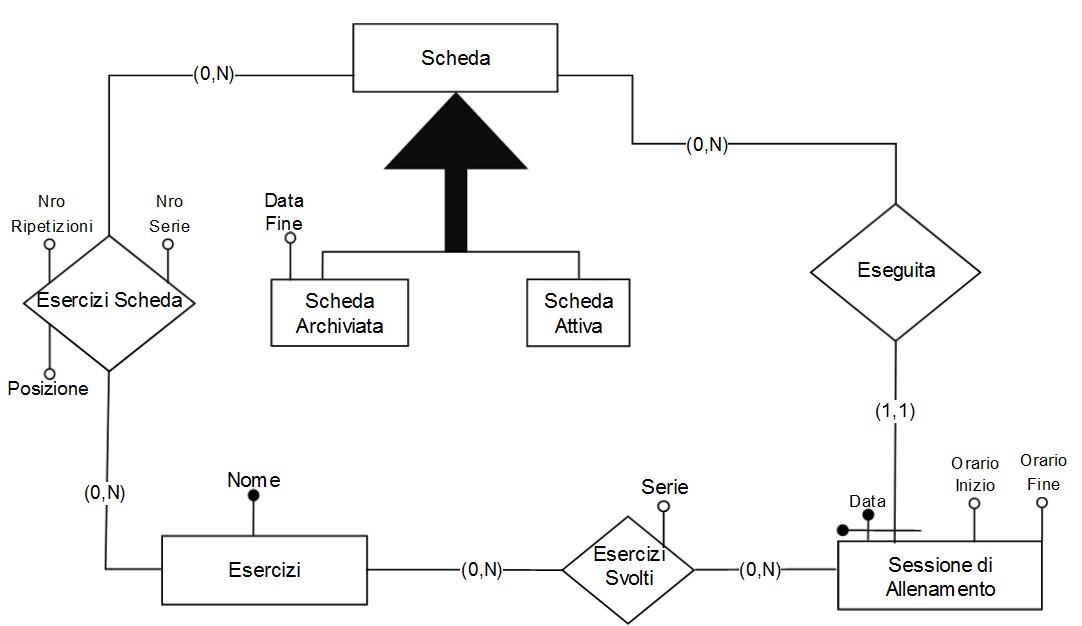


- Infine un ultimo concetto fondamentale è la Sessione di Allenamento. Essa è identificata dalla data in cui viene svolta, e raccoglie inoltre i dati che raffigurano l’orario di inizio e fine allenamento. Questo concetto è strettamente legato sia ai Clienti che svolgono le sessioni sia alle Schede di allenamento, che mostrano ciò che deve essere svolto all’interno di una Sessione di Allenamento.



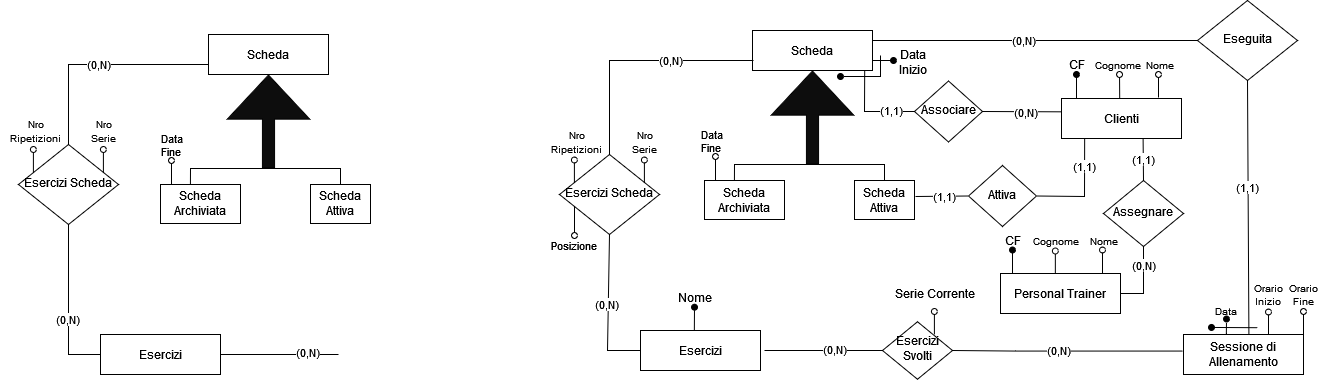
- Dunque andiamo a rappresentare la Sessione di Allenamento come un’istanza della Scheda. La Scheda andrà dunque a raccogliere tutti i dati degli esercizi che dovrebbero essere svolti all’interno di una sessione di allenamento (raffigurando così il concetto astratto di allenamento); e la Sessione di Allenamento vera e propria (concetto reale) che invece cattura i dati sugli esercizi che vengono realmente svolti, da parte del cliente.

Con l’introduzione fatta pocanzi, possiamo dunque rendere la Sessione di Allenamento entità debole verso Scheda. Concludiamo legando la sessione di allenamento agli esercizi con la relazione Esercizi Svolti.



### Integrazione finale

ATTENZIONEEEE METTI CLIENTI (0,1) CON SCEHDA ATTIVA SENNO SO CAZZI TUA



## Regole aziendali

1) Un Esercizio, eseguito o saltato, in una Sessione di Allenamento, deve far parte della Scheda Attiva.

2) L’orario di inizio allenamento deve precedere l’orario di fine allenamento.

## Dizionario dei dati

|  |  |  |  |
| --- | --- | --- | --- |
| **Entità** | **Descrizione** | **Attributi** | **Identificatori** |
| Cliente | Colui che si allena. | Nome, Cognome | Codice Fiscale |
| Personal Trainer | Colui che si occupa dell’allenamento dei clienti a lui associati. | Nome, Cognome | Codice Fiscale |
| Esercizi | Concetto astratto di esercizi. |  | Nome |
| Scheda | Elenco di esercizi che un cliente può svolgere in una sessione di allenamento. |  | Data Inizio, Cliente |
| Scheda Attiva | La Scheda può essere usata dal cliente per allenarsi. |  | Data Inizio,  Cliente |
| Scheda Archiviata | La Scheda può essere solo letta | Data Fine | Data Inizio,  Cliente |
| Sessione di Allenamento | Intervallo temporale in cui un cliente può svolgere gli esercizi sulla sua scheda attiva | Orario Inizio, Orario Fine | Data,  Cliente, Data Inizio Scheda |

# Progettazione logica

## Volume dei dati

RITOCCA DESCRIZIONE: CONSIDERA CANCELLAZIONE DOPO 5 ANNI

Per questa fase di progettazione si suppone che le informazioni mantenute nella base di dati siano relative agli ultimi 5 anni. La specifica richiede di mantenere i dati per sempre, quindi svolgeremo un’analisi sui volumi, le frequenze e i costi della BD; tenendo conto, comunque, del fatto che tolta una crescita inizialmente esponenziale, negli anni a venire dopo il quinto, i costi delle operazioni si stabilizzeranno, dal momento in cui ogni anno il numero di nuovi clienti registrati sarà pressoché analogo al numero di clienti registrati che però non utilizzeranno più l’applicazione. La frequenza delle operazioni rimarrà dunque intatta nel tempo, mentre il numero totale di Clienti e PT\* aumenterà.

1. Si suppone che i dati relativi ai Clienti e ai Personal Trainer che usano l’applicazione siano 1000.

2. Sovrastimiamo, inoltre, che ogni anno in media si registrano 190 clienti.

3. Supponiamo che ogni anno si iscrivano in media 10 nuovi Personal Trainer; dunque dopo 5 anni la base di dati conterrà circa 50 PT. Tali numeri sono stati scelti in modo tale da cercare di rendere costante il lavoro che ogni Personal Trainer deve svolgere: assegnando così dopo 5 anni, circa 19 clienti ad ogni PT.

4. Supponiamo che la media di Esercizi in una Scheda oscilli fra 10 e 6, e quindi sia pari a 8. Di questi ultimi stimiamo che la circa il 75% vengono svolti.

5. Supponiamo inoltre che: i clienti che interagiscono con la scheda attiva sono maggiori dei clienti che leggono le schede archiviate e dunque stimiamo che, ogni giorno, circa il 55% clienti svolgono sessioni di allenamento, interagendo con la scheda, mentre il 10% legge una scheda archiviata. Di conseguenza effettueranno il Login circa il 65% dei Clienti nella BD, ogni giorno.

6. Supponiamo anche che i Personal Trainer creino in media nuove schede ogni 2 mesi per ogni cliente. Dunque il numero di schede totali dopo 5 anni ammonterà a 17 100. Fra queste il numero di Schede Attive sarà pari al numero di Clienti presenti nella base di dati (950); le restanti saranno Schede Archiviate. Il numero di Schede Attive rimarrà costante nel tempo, mentre le Schede Archiviate si accumuleranno.

7. Il numero di possibili esercizi da inserire in una Scheda rimane stabile nel tempo e immutabile: 35.

8. I clienti hanno la possibilità di allenarsi tutti i giorni, una volta al giorno, dunque stimiamo che ci saranno fra 1 e 7 sessioni di allenamento settimanali (media 4).

9. Stimiamo che ogni esercizio abbia in media 3 serie, e che in media vengano eseguite tutte e 3.

|  |  |  |
| --- | --- | --- |
| **Concetto nello schema** | **Tipo[[1]](#footnote-1)** | **Volume atteso** |
| Clienti | E | 950 |
| Personal Trainer | E | 50 |
| Esercizi | E | 35 |
| Sessioni di Allenamento | E | 547 200 |
| Schede | E | 17 100 |
| Schede Attive | E | 950 |
| Schede Archiviate | E | 16 150 |
| Esercizi Scheda | R | 85 500 |
| Esercizi Svolti | R | 3 283 200 |
| Associare | R | 17 100 |
| Eseguita | R | 547 200 |
| Assegnare | R | 950 |

## Tavola delle operazioni

|  |  |  |
| --- | --- | --- |
| **Cod.** | **Descrizione** | **Frequenza attesa** |
| U1 | Inizia Sessione | 314/giorno |
| U2 | Termina Sessione | 314/giorno |
| U3 | Esegui Esercizio | 807/giorno |
| U4 | Visualizza Scheda Attiva | 314/giorno |
| U5 | Visualizza Scheda Archiviata | 57/giorno |
| U6 | Aggiungi Serie Esercizio | 5862/giorno |
| U7 | Visualizza Serie Esercizio | 1955/giorno |
| U8 | Crea Scheda | 3420/anno ≃ 10/giorno |
| U9 | Archivia Scheda | 3420/anno ≃ 10/giorno |
| U10 | Genera Report | 3420/anno ≃ 10/giorno |
| AM1 | Associa Personal Trainer e Cliente | 4/settimana ≃ 0.6/giorno |
| L1 | Login | 371/giorno |
| L2 | Registrazione alla piattaforma | 4/settimana ≃ 0.6/giorno |

## Costo delle operazioni

Si suppone, in questa fase di progettazione, che il costo di scrittura di un dato sia il doppio del costo

in lettura.

U1 – Inizia Sessione

* Clienti: 950 (L)
* Attiva: 950 (L)
* Scheda Attiva: 950 (L)
* Eseguita: 547 200 (W)
* Sessione di Allenamento: 547 200 (W)

Costo Totale: 2 191 650

Accessi/giorno: (2 191 650) \* 314/giorno= 688 178 100

U2 – Termina Sessione

* Clienti: 950 (L)
* Attiva: 950 (L)
* Scheda Attiva: 950 (L)
* Eseguita: 547 200 (L)
* Eseguita: 547 200 (W)
* Sessione di Allenamento: 547 200 (W)

Costo Totale: 2 738 850

Accessi/giorno: (2 738 850) \* 314/giorno = 859.998.900

U3 – Esegui Esercizio

* Clienti: 950 (L)
* Attiva: 950 (L)
* Scheda Attiva: 950 (L)
* Eseguita: 547 200 (L)
* Sessione di Allenamento: 547 200 (L)
* Esercizi Svolti 85 500 (W)

Costo Totale: 1 268 250

Accessi/giorno: (1 268 250) \* 807/giorno = 1 023 477 750

U4 – Visualizza Scheda Archiviata

* Clienti: 950 (L)
* Associare: 17 100 (L)
* Scheda Archiviata: 16 150 (L)

Costo Totale: 34 200

Accessi/giorno: (34 200) \* 57/giorno = 1 949 400

U5 – Visualizza Scheda Attiva

* Clienti: 950 (L)
* Attiva: 950 (L)
* Scheda Attiva: 950 (L)

Costo Totale: 2850

Accessi/giorno: (2850) \* 314/giorno = 894 900

U6 – Aggiorna Serie Esercizio

* Clienti: 950 (L)
* Attiva: 950 (L)
* Scheda Attiva: 950 (L)
* Eseguita: 547 200 (L)
* Sessione di Allenamento: 547 200 (L)
* Esercizi Svolti 85 500 (L)
* Esercizi Svolti 85 500 (W)

Costo Totale: 1 353 750

Accessi/giorno: (1 353 750) \* 5862/giorno = 7 935 682 500

U7 – Visualizza Serie Esercizio

* Clienti: 950 (L)
* Associare: 950 (L)
* Scheda: 950 (L)
* Eseguita: 547 200 (L)
* Sessione di Allenamento: 547 200 (L)
* Esercizi Svolti 85 500 (L)

Costo Totale: 1 182 750

Accessi/giorno: (1 182 750) \* 1955/giorno = 2 312 276 250

U8 – Crea Scheda

* Personal Trainer 50 (L)
* Assegnare 950 (L)
* Clienti: 950 (L)
* Attiva: 950 (W)
* Scheda Attiva: 950 (W)
* Associare 950 (W)

Costo Totale: 7 650

Accessi/giorno: (7 650) \* 10/giorno = 76 500

U9 – Archivia Scheda

* Clienti: 950 (L)
* Attiva: 950 (L)
* Scheda Attiva: 950 (L)
* Scheda Archiviata: 16 150 (W)
* Associare: 950 (W)
* Attiva: 950 (W)

Costo Totale: 22 800

Accessi/giorno: (22 800) \* 10/giorno = 228 000

U10 – Genera Report

* Personal Trainer: 50 (L)
* Assegnare: 950 (L)
* Clienti: 950 (L)
* Associare: 17 100 (L)
* Scheda: 17 100 (L)
* Eseguita: 547 200 (L)
* Sessione di Allenamento: 547 200 (L)
* Esercizi Svolti 85 500 (L)

Costo Totale: 1 216 050

Accessi/giorno: (1 216 050) \* 10/giorno = 12 160 500

AM1 – Associa Personal Trainer e Cliente

* Personal Trainer: 50 (L)
* Assegnare: 950 (L)
* Clienti: 950 (L)
* Assegnare: 950 (W)

Costo Totale: 3850

Accessi/giorno: (3850) \* 0.6/giorno = 2 310

AM2 – Registrazione

* Utente: 950 (W)
* PT: 50 (W)

Costo Totale: 2 000

Accessi/giorno: (2 000) \* 0.6/giorno ≃ 1200

VALUTA SE TOGLIERE PERCHE’ LO FA LA SEGRETERIA MANUALMENTE

L1 – Login

* Utente: 950 (L)
* PT: 50 (L)

Costo Totale: 1000

Accessi/giorno: (1 000) \* 10/giorno = 10 000

## Ristrutturazione dello schema E-R

Descrivere (laddove necessario fornendo anche degli schemi) quali passi vengono adottati per ristrutturare lo schema E-R, ad esempio in termini di:

Analisi delle ridondanze

Eliminazione delle generalizzazioni

Scelta degli identificatori primari

Si noti che in questa fase è possibile fare riferimento al costo delle operazioni precedentemente realizzato per guidare le scelte. Ad esempio, un leggero spreco di memoria legato alla non rimozione di ridondanze può essere facilmente giustificato da un guadagno in termini di prestazioni.

**Eliminazione delle generalizzazioni**

Consideriamo adesso l’entità Scheda analizziamo, dunque, la scelta migliore per eliminare la generalizzazione:

* La scelta più naturale risulterebbe portare il padre nei figli, questo principalmente perché l’entità Scheda ha due figli che supportano operazioni diverse e che quindi tendono a sottolineare la differenza fra le due entità.

Il grande problema di questa opzione però riguarda l’eccessivo costo che andrei a pagare in memoria, dato che si duplicherebbero quasi tutte le associazioni, che, essendo legate al padre, passerebbero ad entrambi i figli.

* L’utilizzo di associazioni fra il padre e le figlie comporta, anch’esso, un grosso spreco di memoria dato che dovendo tenere traccia di queste due relazioni, gli accessi per le operazioni U8, U9 risulterebbero notevolmente maggiori. Esempio, per l’operazione U9 (archiviazione di una scheda):

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accesso** | **Tipo** |
| Clienti | E | 950 | L |
| Attiva | R | 950 | L |
| Scheda Attiva | E | 950 | L |
| Scheda-SchedaAttiva | R | 950 | L |
| Scheda-SchedaArchiviata | R | 16 150 | W |
| Scheda Archiviata | E | 16 150 | W |
| Scheda-SchedaAttiva | R | 16 150 | W |
| Attiva | R | 950 | W |
| Associare | R | 16 150 | L |
| Associare | R | 16 150 | W |

Costo Totale: 151 050

Accessi/giorno: (151 050) \* 10/giorno = 1 510 500

* Se invece utilizzassimo l’accorpamento del genitore della generalizzazione nelle figlie si verifica un piccolo spreco di memoria, dato che le operazioni che richiedono la Scheda Attiva dovranno prima calcolarla, ma ciò non costituisce un grande problema. Inoltre così facendo, diminuirebbe il numero di accessi, così come la duplicazione dei dati, migliorando l’efficienza dello spazio di archiviazione e semplificando la gestione e la manutenzione dei dati.

Di conseguenza adottiamo la seguente strategia:

1. Aggiungere l’attributo Data fine a Scheda con cardinalità (0, 1) per indicare se una Scheda è Attiva o Archiviata.
2. Dovrebbe essere aggiunta una relazione Attiva tra Scheda e Clienti per identificare la Scheda Attiva, ma ciò comporterebbe la presenza di un’associazione ciclica e, tra l’altro, ridondante. Infatti, è possibile ricavare la Scheda Attiva attraverso la data di inizio delle Schede. La Scheda con data di inizio maggiore è la Scheda Attiva. Bisogna aggiungere due vincoli:

- La Scheda Attiva non deve avere una data di fine.

- Ogni Cliente deve avere al massimo una Scheda Attiva.

1. Aggiungere un vincolo per cui ogni Scheda Archiviata deve avere una data di fine.

L’operazione AM2 diverrebbe:

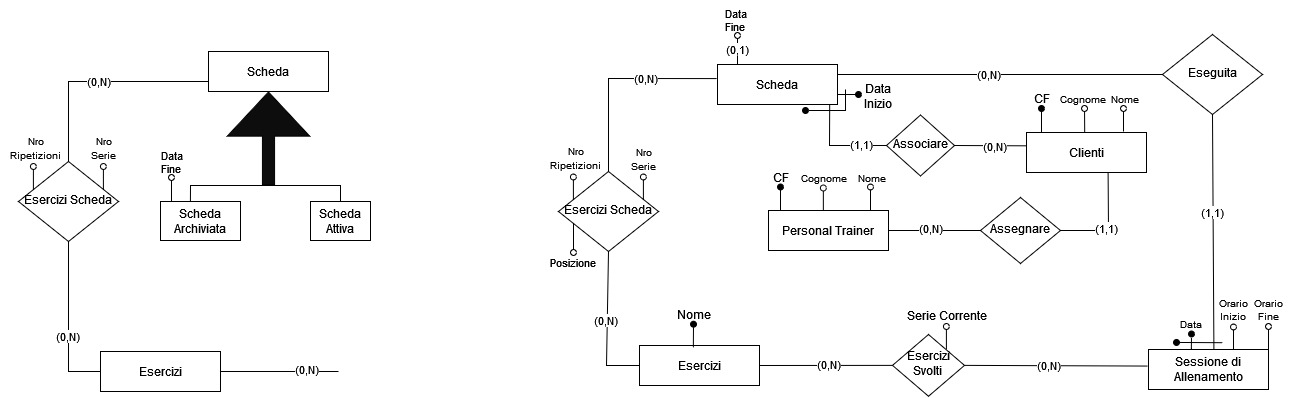
|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accesso** | **Tipo** |
| Clienti | E | 950 | L |
| Associare | R | 17 100 | L |
| Associare | R | 16 150 | W |
| Scheda | E | 17 100 | L |
| Scheda | E | 16 150 | W |

Costo Totale: 67 450

Accessi/giorno: (67 450) \* 10/giorno = 674 500

ATTENZIONEEEE METTI CLIENTI (0,1) CON PERSONAL TRAINER SENNO SO CAZZI TUA

**Schema Ristrutturato**



## Trasformazione di attributi e identificatori

Qualora siano presenti, in questa fase della progettazione, attributi ripetuti o identificatori esterni, descrivere quali trasformazioni vengono realizzate sul modello per facilitare la traduzione nello schema relazionale.

## Traduzione di entità e associazioni

Riportare in questa sezione la traduzione di entità ed associazioni nello schema relazionale.

Fornire una rappresentazione grafica del modello relazionale completo.

Partendo dallo schema ristrutturato di cui sopra, costruiamo le seguenti relazioni per il modello relazionale

● *PersonalTrainer*(CF, Nome, Cognome)

● *Cliente*(CF, Nome, Cognome, PersonalTrainer\*)

○ *Cliente*(PersonalTrainer) ⊆ *PersonalTrainer*(CF)

● *Scheda*(DataInizio, Cliente, DataFine\*)

○ *Scheda*(Cliente) ⊆ *Cliente*(CF)

● *SessioneDiAllenamento*(Data, Cliente, DataInizioScheda, OrarioInizio, OrarioFine)

○ *SessioneDiAllenamento*(Cliente) ⊆ *Scheda*(Cliente)

○ *SessioneDiAllenamento*(DataInizioScheda) ⊆ *Scheda*(DataInizio)

● *EserciziSvolti*(DataSessione, Cliente, DataInizioScheda, Esercizio, SerieCorrente)

○ *EserciziSvolti*(Cliente, DataInizioScheda) ⊆ *SessioneDiAllenamento*(Cliente, DataInizioScheda)

○ *EserciziSvolti*() ⊆ *SessioneDiAllenamento*()

○ *EserciziSvolti*(DataSessione) ⊆ *SessioneDiAllenamento*(Data)

○ *EserciziSvolti*(Esercizio) ⊆ *Esercizi*(Nome)

● *Esercizi*(Nome)

● *EserciziScheda*(Cliente, DataInizioScheda, Esercizio, NumeroRipetizioni, NumeroSerie, Posizione)

○ *EserciziScheda*(Cliente, DataInizioScheda) ⊆ *SessioneDiAllenamento*(Cliente, DataInizioScheda)

○ *EserciziScheda*(Esercizio) ⊆ *Esercizi*(Nome)

\* Tutti i valori contrassegnati dall’asterisco possono accettare valori nulli

## Normalizzazione del modello relazionale

Mostrare se il modello relazionale precedentemente descritto è in forma normale. Se non lo è, descrivere quali trasformazioni si effettuano per normalizzarlo. Se si sceglie di non normalizzarlo, giustificare le motivazioni da un punto di vista prestazionale.

# Progettazione fisica

## Utenti e privilegi

Descrivere, all’interno dell’applicazione, quali utenti sono stati previsti con quali privilegi di accesso su quali tabelle, giustificando le scelte progettuali.

Sono previsti 4 ruoli:

● Login (Sono tutti I client non registrati E registrati ma non loggati) (qua dentro avrò anche la segreteria quando non è loggata, quando lo sarà avrà il suo ruolo)

(E’ UN RUOLO NON UNA TABELLA)

- Grant in esecuzione su L1.

- Clienti

- Grant in esecuzione su operazioni .

## Strutture di memorizzazione

Compilare la tabella seguente indicando quali tipi di dato vengono utilizzati per memorizzare le informazioni di interesse nelle tabelle, per ciascuna tabella.

|  |  |  |
| --- | --- | --- |
| **Tabella <Utenti>** | | |
| **Attributo** | **Tipo di dato** | **Attributi[[2]](#footnote-2)** |
| **Username** | VARCHAR (15) | PK, NN |
| **Password** | VARCHAR (15) | NN |
| **Ruolo** | ENUM('personaltrainer',  'segreteria', 'clienti') | NN |

|  |  |  |
| --- | --- | --- |
| **Tabella <Clienti>** | | |
| **Colonna** | **Tipo di dato** | **Attributi** |
| **CF** | VARCHAR (15) | PK |
| **Nome** | VARCHAR (15) | NN |
| **Cognome** | VARCHAR (25) | NN |
| **PersonalTrainer** | VARCHAR (15) |  |
| **Username** | VARCHAR (15) | UQ, NN |

|  |  |  |
| --- | --- | --- |
| **Tabella <PersonalTrainer>** | | |
| **Colonna** | **Tipo di dato** | **Attributi** |
| **CF** | VARCHAR (15) | PK, NN |
| **Nome** | VARCHAR (15) | NN |
| **Cognome** | VARCHAR (25) | NN |
| **Username** | VARCHAR (15) | UQ, NN |

|  |  |  |
| --- | --- | --- |
| **Tabella <Scheda>** | | |
| **Colonna** | **Tipo di dato** | **Attributi** |
| **DataInizio** | DATE | PK, NN |
| **Cliente** | VARCHAR (15) | PK, NN |
| **DataFine** | DATE |  |

|  |  |  |
| --- | --- | --- |
| **Tabella <Esercizi>** | | |
| **Colonna** | **Tipo di dato** | **Attributi** |
| **Nome** | VARCHAR (15) | PK, NN |

|  |  |  |
| --- | --- | --- |
| **Tabella <EserciziScheda>** | | |
| **Colonna** | **Tipo di dato** | **Attributi** |
| **Cliente** | VARCHAR (15) | PK, NN |
| **DataInizioScheda** | DATE | PK, NN |
| **Esercizio** | VARCHAR (15) | PK, NN |
| **NumeroRipetizioni** | INT | NN |
| **NumeroSerie** | INT | NN |
| **Posizione** | INT | NN |

|  |  |  |
| --- | --- | --- |
| **Tabella <SessioneDiAllenamento>** | | |
| **Colonna** | **Tipo di dato** | **Attributi** |
| **Data** | DATE | PK, NN |
| **Cliente** | VARCHAR (15) | PK, NN |
| **DataInizioScheda** | DATE | PK, NN |
| **OrarioInizio** | TIME | NN |
| **OrarioFine** | TIME | NN |

|  |  |  |
| --- | --- | --- |
| **Tabella <EserciziSvolti>** | | |
| **Colonna** | **Tipo di dato** | **Attributi** |
| **DataSessione** | DATE | PK, NN |
| **Cliente** | VARCHAR (15) | PK, NN |
| **DataInizioScheda** | DATE | PK, NN |
| **Esercizio** | VARCHAR (15) | PK, NN |
| **SerieCorrente** | INT | NN |

## Indici

Compilare la seguente tabella, per ciascuna tabella del database in cui sono presenti degli indici. Descrivere le motivazioni che hanno portato alla creazione di un indice, facendo riferimento al costo delle operazioni individuate nella sezione precedente.

Attenzione: non è necessario riportare gli indici autogenerati in fase di definizione dello schema (ad esempio, per la gestione della chiave primaria), ma *soltanto* quelli introdotti per motivi prestazionali.

|  |  |
| --- | --- |
| **Tabella <nome>** | |
| **Indice <nome>** | **Tipo[[3]](#footnote-3):** |
| Colonna 1 | <nome> |

## Trigger

Descrivere quali trigger sono stati implementati, mostrando il codice SQL per la loro istanziazione. Si faccia riferimento al fatto che il DBMS di riferimento richiede di utilizzare trigger anche per realizzare vincoli di check ed asserzioni.

## Eventi

Descrivere quali eventi sono stati implementati, mostrando il codice SQL per la loro istanziazione. Si descriva anche se gli eventi sono istanziati soltanto in fase di configurazione del sistema, o se alcuni eventi specifici vengono istanziati in maniera effimera durante l’esecuzione di alcune procedure.

## Viste

Mostrare e commentare il codice SQL necessario a creare tutte le viste necessarie per l’implementazione dell’applicazione.

## Stored Procedures e transazioni

Mostrare e commentare le stored procedure che sono state realizzate per implementare la logica applicativa delle operazioni sui dati, evidenziando quando (e perché) sono state realizzate operazioni transazionali complesse.

1. Indicare con E le entità, con R le relazioni [↑](#footnote-ref-1)
2. PK = primary key, NN = not null, UQ = unique, UN = unsigned, AI = auto increment. È ovviamente possibile specificare più di un attributo per ciascuna colonna. [↑](#footnote-ref-2)
3. IDX = index, UQ = unique, FT = full text, PR = primary. [↑](#footnote-ref-3)