



Università di Pisa
Corso di Laurea in Ingegneria Informatica

Progetto per il corso Basi di Dati:
Database Azienda di Fitness

Progettato da Leo Maltese

1 Analisi delle Specifiche

Il database che si desidera progettare ha lo scopo di memorizzare e gestire efficientemente i dati a supporto delle funzionalità del sistema informativo di una grande azienda che si occupa di fitness.

L'azienda è proprietaria di più centri fitness distribuiti sul territorio nazionale. Ciascun centro è dotato di sale fitness equipaggiate da attrezzi di vario genere. I centri sono anche dotati di spazi adibiti a corsi nonché di aree allestibili per far praticare ai clienti diversi sport, quali tennis, pallavolo, calcetto eccetera. I centri fitness mettono a disposizione dei loro clienti delle apparecchiature smart in grado di monitorare costantemente la loro attività sportiva, per individuare sia le lacune e i traguardi raggiunti. Inoltre, l'azienda mette a disposizione un servizio di scienza della nutrizione per assistere i clienti anche dal punto di vista dell'alimentazione. I clienti dei centri fitness possono infine usufruire di una piattaforma social per creare amicizie, condividere i traguardi e lanciare sfide.

2 Progettazione Concettuale

2.0 Strategia di Progetto

Nella realizzazione del progetto è stata seguita una Strategia Mista, cercando di combinare i vantaggi della Strategia Top-Down e di quella Bottom-Up. Nella Strategia Top-Down infatti, si parte da uno Schema con pochi concetti molto Astratti.

Avendo a disposizione sin da subito tutte le Specifiche, è stato possibile fissare in mente i concetti Principali, trascurando i dettagli. Nella Strategia Bottom-Up, il vantaggio sta' nella Decomposizione del Problema, il quale preso tutto insieme può essere ingestibile, ma che suddiviso in tanti sottoproblemi è sicuramente più abbordabile. (Divide et Impera)

Un caso particolare della Bottom-Up è la Inside-Out, che è stata la metodologia prevalentemente usata nella realizzazione del progetto. In quest'ultima Strategia infatti, si individuano inizialmente solo alcuni *concetti importanti* e poi si procede a “**macchia d'olio**”.

Si rappresentano cioè prima i concetti iniziali e poi ci si muove verso concetti più lontani e contorti attraverso una “navigazione”.

Nella Realizzazione del progetto, si è cercato di mantenere un'ottima *qualità* dello Schema Concettuale, in termini di *Correttezza* (quale costruito usare per Rappresentare i concetti), *Completezza* (Rappresentare Tutti i dati), *Leggibilità* (Impostando le entità Principali al centro) e *Minimalità* (Non appesantendo troppo la base di dati) .

2.1 Dettagli di Progetto

1. Nelle entità “*SALA*” e “*PISCINA*” è stato inserito l'attributo “**Tipologia Contratto**”, quest'ultimo sta ad indicare il requisito minimo di contratto che bisogna avere per poter entrare in quella sala o piscina.
L'ordine di importanza di contratto è elencato, in ordine crescente: Silver, Gold, Platinum, Personalizzato.
Ad esempio: se un cliente ha un contratto di tipo silver, quest'ultimo può accedere solamente alle sale di tipo silver e quindi non potrà accedere ad una sala di tipologia gold. Se invece un altro cliente ha un contratto di tipo platinum quest'ultimo potrà accedere alle sale, o piscine, di tipologia silver, gold e platinum.
2. Nell'entità “*CLIENTE OCCASIONALE*” è stato inserito un attributo “**Tipologia Tariffa**” utilizzato per indicare la tipologia di contratto che l'utente utilizzerà per quell' accesso al centro fitness. La tipologia di contratto utilizzata dal cliente dipende dalla tariffa d'accesso da lui pagata in quella giornata.
3. Si assume che ogni Centro Fitness sia chiuso nei giorni festivi nel corso dell'anno, compresi festività religiose e nazionali. Quest'ultimi dipendono dalla nazione in cui viene attivato questo database.
4. Si assume che nel caso in cui il pagamento di un contratto avvenga in maniera rateizzata, ogni rata venga pagata mensilmente.
5. Si assume che un cliente non utilizza sempre un attrezzo per svolgere un esercizio

6. Si assume che contratti non superiori ad una quota di 30 euro mensile non possano accedere alle piscine del centro in cui si sono iscritti come specificato nello schema seguente:

Tipologia Contratto	Costo Mensile	Accesso Piscina	Numero Accessi Piscina Mensili	Numero Accessi settimanali
Silver	20	NO	NO	1
Silver	30	SI	2	1
Gold	50	SI	6	2
Gold	55	SI	8	3
Platinum	65	SI	10	3
Platinum	70	SI	12	4
Personalizzato	100	SI	20	Illimitato

2.2 Individuazione e Descrizione delle Entità

In questo paragrafo vengono elencate e descritte le entità che sono state individuate dalla lettura delle specifiche.

- CLIENTE: Entità utilizzata per descrivere un cliente che è iscritto ad uno dei centri fitness dell'azienda.
È caratterizzata da: Codice Fiscale Cliente, Nome, Cognome, Data di Nascita, Indirizzo (a sua volta è composto da una Via, una Città, un CAP, un Numero Civico), Sesso, Il Codice di un documento di riconoscimento e la Prefettura che lo ha rilasciato.
- CENTRO FITNESS: Entità utilizzata per descrivere uno dei centri fitness dell'azienda.
È caratterizzata da: Un codice identificativo diverso per ogni centro fitness, un Indirizzo (a sua volta composto da una Via, una Città, un CAP, un Numero Civico), Capienza Massima, che descrive il numero massimo che di clienti che può contenere contemporaneamente, un Numero di Telefono e la Dimensione del centro in metri quadrati.
- SALA: Entità utilizzata per descrivere una particolare sala di un centro fitness.
È caratterizzata da: ID Sala il quale identifica in modo univoco una sala e non ci sono sale di centri diversi con lo stesso ID sala, un Nome e una Tipologia Contratto che specifica il contratto minimo richiesto per entrare in quella determinata sala (per dettagli leggere paragrafo 2.1).
- CLIENTE OCCASIONALE: Entità utilizzata per descrivere un cliente con non ha un contratto con nessun centro fitness.
È caratterizzato da: Tariffa d'accesso, che è la quota pagata dal cliente, e Tipologia Tariffa (per dettagli leggere paragrafo 2.1).
Un "Cliente Occasionale", in quanto Specializzazione di un Cliente, contiene per la proprietà dell'ereditarietà anche tutte le caratteristiche di un Cliente come concetto generale.

- ATTREZZO: Entità utilizzata per descrivere un attrezzo del centro fitness. È caratterizzato da: ID attrezzo, diverso per ogni centro fitness, Nome Attrezzo, Tipologia, Consumo Energetico e un livello di usura. Quest'ultimo aumenta man mano che viene utilizzato.
Quando il livello di usura raggiunge il 100%, che corrisponde circa a 1.000 utilizzi dell'attrezzo, il quale necessita di manutenzione e non può essere utilizzato fintanto che l'intervento manutenzione non viene fatto.
- PISCINA: Entità utilizzata per descrivere le piscine presenti in ogni Centro fitness.
È caratterizzata da: ID Piscina, diverso per ogni centro fitness, Dimensione in metri quadri, Tipologia Contratto (per dettagli leggere paragrafo 2.1) e Locazione che indica se una piscina è all'esterno o all'interno della struttura del Centro Fitness.
- CALENDARIO
CENTROFITNESS: Entità utilizzata per descrivere il calendario settimanale di ogni Centro Fitness. Il Centro Fitness rimane chiuso nei giorni festivi (per dettagli vedere paragrafo 2.1).
È caratterizzato da: Nome Giorno che indica il giorno della settimana, Orario di apertura e un Orario di chiusura. Si assuma che ogni centro faccia un orario continuato dalla mattina fino alla sera.
- DIPENDENTE: Entità utilizzata per descrivere un dipendente di ogni Centro Fitness.
È caratterizzato da: Codice Fiscale, Nome, Cognome, Data di Nascita, Indirizzo (a sua volta è composto da una Via, una Città, un CAP, un Numero Civico), Sesso, Il Codice di un documento di riconoscimento e la Prefettura che lo ha rilasciato.

- CONTRATTO: Entità utilizzata per descrivere ogni contratto che un Centro Fitness sottoscrive con un Cliente.
È caratterizzato da: ID Contratto che è diverso per ogni Cliente e diverso per ogni Centro Fitness, Data di Iscrizione, Cliente, Codice Consulente col quale il Cliente ha sottoscritto il contratto, Durata e Tipologia Contratto.
Quest'ultimo attributo verrà analizzato meglio nel paragrafo delle generalizzazioni (paragrafo 2.2).
- FASCI MUSCOLARI: Entità utilizzata per descrivere i muscoli che devono essere allenati in caso in cui lo scopo di un contratto fosse "Potenziamento".
È caratterizzato da: Nome del muscolo e un Livello di Potenziamento che può assumere tre valori (lieve, moderato e elevato) dipendenti dal potenziamento che il cliente desidera raggiungere per ciascuno dei muscoli.
- ACCESSO: Entità utilizzata per descrivere ogni Accesso che il cliente effettua in un Centro Fitness.
È caratterizzata da: ID Accesso che identifica l'accesso di un Cliente in un Centro Fitness, Cliente, Centro Fitness, Data, Orario Entrata e un Orario Uscita.
- ARMADIETTO: Entità utilizzata per descrivere un Armadietto che sta All'interno di uno spogliatoio di un Centro Fitness il quale viene assegnato ad un cliente ogni qual volta quest'ultimo entra in un Centro Fitness.
È caratterizzata da: ID Armadietto che identifica univocamente un armadietto e una Combinazione Numerica che viene visualizzata sullo smartwatch del cliente.
- SPOGLIATOIO: Entità utilizzata per descrivere uno spogliatoio presente in un centro fitness.

È caratterizzata da: ID Spogliatoio che lo identifica univocamente, una capienza, posti disponibili, posizione indicato con un punto cardinale.

- MISURAZIONE

IMPEDENZIOMETRICHE :

Entità che descrive le misurazioni

Impedenziometriche fatte dai medici nutrizionisti ai clienti.

È caratterizzata da: ID Misurazione che lo identifica in modo univoco, la Data della misurazione, Acqua Totale, Massa Grassa, Peso, Altezza, Massa Magra, Stato del Cliente che può assumere tre valori (Sottopeso, Normopeso, Sovrappeso) e infine l'Entità di tale stato.

Questi due ultimi attributi vengono calcolati in base alle misurazioni effettuate e si basano sulla seguente formula:

$$\frac{\text{Peso} * (\text{MassaGrassa} + \text{MassaMagra})}{(\text{Altezza})^2 * \text{AcquaTotale}}$$

In base al valore di tale formula, si stabilisce lo stato del cliente e l'entità di tale stato. La classificazione specifica avviene attraverso la seguente tabella:

Stato	Entità	Range
<i>Sottopeso</i> III grado	Grave	< 16.5
<i>Sottopeso</i> II grado	Medio	Da 16.6 a 17.5
<i>Sottopeso</i> I grado	Lieve	Da 17.6 a 18.4
<i>Normopeso</i> III grado	Grave	Da 18.5 a 20.6
<i>Normopeso</i> II grado	Medio	Da 20.7 a 22.8
<i>Normopeso</i> I grado	Lieve	Da 22.9 a 24.9
<i>Sovrappeso</i> I grado	Lieve	Da 25 a 32
<i>Sovrappeso</i> II grado	Medio	Da 32.1 a 40
<i>Sovrappeso</i> III grado	Grave	> 40

- ORARIO DI

LAVORO:

Entità utilizzata per descrivere la turnazione di un dipendente.

È caratterizzata da: Dipendente, ID Centro, Giornata della settimana, orario inizio e un orario di fine. Tutti gli attributi sono necessari che identificare univocamente un orario di lavoro di un dipendente.

- CORSO: Entità che descrive un corso che si svolge in un centro fitness (Ovviamente si tratterà di un corso di natura sportiva).
È caratterizzata da: ID Corso che lo identifica univocamente, Codice della Sala in cui si svolge il corso, Il Numero Massimo di Partecipanti, Nome della Disciplina, l'istruttore responsabile, Data d'inizio, Data di fine e un livello che indica la difficoltà del corso (*principiante, intermedio, avanzato*).
- CALENDARIO
CORSO: Entità che descrive il calendario e quindi in quali giorni si svolge e gli orari.
È caratterizzata da: ID Calendario che lo identifica univocamente, Centro Fitness, Giornata della Settimana, Orario d'inizio e orario di fine corso.
- UTENTE: Entità utilizzata per descrivere un cliente nel suo profilo online che un centro fitness mette a disposizione.
È caratterizzato da: Username che lo identifica univocamente, una Password e un indice di Credibilità. Quest'ultimo è tanto più elevato quanti più post sono stati scritti dall'utente e quanto più le risposte fornite dall'utente vengono valutate positivamente. L'indice di Credibilità è così definito:

$$Credibilità = \frac{NumeroPost + NumeroRisposte}{MediaValutazioneRisposte}$$

- POST: Entità utilizzata per descrivere un post che un utente ha pubblicato per chiedere informazioni sul centro, sui corsi svolti o anche per

condividere le proprie esperienze sportive.

È caratterizzato da: Codice Post che lo identifica univocamente, Username dell'utente che lo ha pubblicato, un Timestamp che l'ora in cui è stato pubblicato, un Testo e la presenza o meno di Link associati.

- THREAD: Entità utilizzata per descrivere i thread dove vengono pubblicati i post.
È caratterizzata da: un Codice Thread e un Titolo.
- VOTO: Entità utilizzata per descrivere il voto che un utente dà a un post post risposta, il quale può contribuire o meno ad aumentare la credibilità di un utente.
È caratterizzata da: un Codice Voto che lo identifica univocamente, l'Username dell'utente che ha dato il voto e da Stelle che indica la valutazione (va da 1 a 5 stelle).
- RICHIESTA AMICIZIA: Entità utilizzata per descrivere una richiesta di amicizia che un utente chiede ad un altro utente.
È caratterizzato da: un Codice Richiesta che lo identifica univocamente, l'username dell'utente che riceve la richiesta e lo stato della richiesta che ci dice se appunto quest'ultima è stata accettata o meno.
- PIANO PAGAMENTO: Entità utilizzata per descrivere il piano di pagamento che il Cliente sceglie al momento della sottoscrizione del contratto.
È caratterizzato da: un ID pagamento che lo identifica univocamente e un Importo Totale che indica la cifra da Pagare per il contratto sottoscritto.

- **RATA:** Entità utilizzata per descrivere una rata nel caso in cui un cliente Dovesse scegliere il pagamento di tipo rateizzato.
È caratterizzato da: un ID Rata che lo identifica univocamente, una Data di scadenza, l'importo della rata, l'Istituto Finanziario, il Tasso d'interesse e infine uno Stato che indica, appunto, lo stato del pagamento che assume uno dei seguenti valori:
eseguito, non ancora dovuto, scaduto.
- **SFIDA:** Entità che descrive una particolare sfida che un utente può lanciare ai Suoi amici.
È caratterizzata da: un Proponente che è colui che lancia la sfida, Data di inizio sfida, Data di scadenza della sfida e infine uno Scopo della sfida.
- **AMICO:** Entità utilizzata per descrivere un amico di un cliente.
È caratterizzato da: Username amico e utente che lo identifica univocamente
- **CERCHIA:** Entità utilizzata per descrivere una cerchia di un utente.
È caratterizzata da: Nome Cerchia che lo identifica univocamente e dall'argomento che interessa la cerchia.
- **SCHEDA**
ALLENAMENTO: Entità utilizzata per descrivere la/e scheda/e di allenamento che viene/vengono associata/e al cliente di un centro fitness.
È caratterizzata da: Codice Scheda che lo identifica univocamente, una Data di inizio e una Data di fine e il tutor che ha assegnato la scheda al cliente.
- **ESERCIZIO:** Entità utilizzata per descrivere gli esercizi che compongono una scheda di allenamento.
È caratterizzata da: un ID Esercizio che lo identifica univocamente, il nome dell'esercizio e da un Dispendio Energetico Orario Medio.
All'entità sono associate delle generalizzazioni che verranno

esaminate nel dettaglio nel paragrafo 2.2.

- SCHEDA

- ALIMENTAZIONE:

- Entità utilizzata per descrivere la/e scheda/e di alimentazione che viene/vengono associata/e al cliente di un centro fitness.

- È caratterizzata da: un ID Scheda Alimentazione che lo identifica univocamente, una Data d'inizio, una Data di fine, un Obiettivo che il cliente desidera raggiungere con tale alimentazione.

- DIETA:

- Entità utilizzata per descrivere la dieta che viene associata alla scheda di alimentazione del cliente.

- È caratterizzata da: un ID Dieta che lo identifica univocamente, un Numero di pasti giornalieri e un Apporto Calorico Giornaliero che indica quante calorie in media deve assumere giornalmente il cliente.

- PIETANZA:

- Entità utilizzata per descrivere una pietanza che può comporre una dieta.

- È caratterizzata da: Nome della Pietanza che lo identifica univocamente e la Composizione della pietanza sotto forma di campo testuale.

- MONITORAGGIO

- SMART:

- Entità utilizzata per descrivere come un utente svolge un determinato esercizio.

- È caratterizzato da: un ID Monitoraggio che lo identifica univocamente, Data del monitoraggio.

.

- DATI RIPETIZIONE:

- Entità utilizzata per descrivere il monitoraggio della Ripetizione di un esercizio svolta dal cliente.

- È caratterizzata da: un ID Ripetizione che identifica univocamente, Nome della ripetizione, Numero di esecuzioni realmente eseguiti dal cliente, e il tempo di recupero effettivamente fatto dal cliente.

- **CONFIGURAZIONE**

- UTILIZZO:**

- Entità utilizzata per rappresentare le configurazioni di un attrezzo per lo svolgimento di un esercizio.

- È caratterizzata da: ID Configurazione che lo identifica univocamente, e da Istruzione Esatta che serve a distinguere una configurazione corretta da una configurazione eseguita da un cliente.

2.2 Individuazione delle Generalizzazioni

Inseguito si dichiarano le generalizzazioni che sono state introdotte nel Diagramma ER non ristrutturato.

Si ha una Generalizzazione quando ciascuna occorrenza di un'entità Figlia è sicuramente occorrenza anche dell'entità Padre e ogni Proprietà dell'entità Padre è anche proprietà dell'entità Figlia.

Nel nostro caso sono presenti 3 generalizzazioni:

1. GENERALIZZAZIONE

ESERCIZIO:

Nell'entità esercizio è stata inserita una generalizzazione totale ed esclusiva caratterizzata dalle due entità figlie: **Aerobico** ed **Anaerobico**.

È stata introdotta in quanto un esercizio può rientrare solamente in una delle due entità figlie sopra descritte. L'entità figlia **Aerobico** è composta dall'attributo *Durata* che sta ad indicare per quanto tempo l'esercizio deve essere svolto.

L'entità figlia **Anaerobico** è composta dagli attributi: *Numero Esecuzioni*, che sta a indicare quante volte deve essere quell'esercizio, un *Tempo di Recupero* che è il tempo che deve intercorrere tra una ripetizione e l'altra e un *Nome*.

2. GENERALIZZAZIONE

CONTRATTO:

Nell'entità Contratto è stata inserita una generalizzazione totale ed esclusiva con le entità *Standard* e *Personalizzato* che stanno ad indicare rispettivamente un contratto di tipo prestabilito dal centro fitness e proposto al cliente e un contratto che è stato costruito sui desideri del cliente.

L'entità figlia *Standard* è composta da un solo attributo, *Tipologia Contratto*, che sta ad indicare quale tipo di tipologia standard il cliente ha scelto.

L'entità figlia *Personalizzato* invece non ha attributi associati.

3. CLIENTE

OCCASIONALE:

Nell'entità Cliente è stata inserita una generalizzazione parziale con l'entità "Cliente Occasionale" che sta ad indicare quel particolare cliente che non ha un contratto con il centro fitness e accede comunque ai servizi di tale centro pagando una quota d'ingresso.

In particolare, l'entità figlia è caratterizzata da: una *Tariffa d'accesso* che il cliente paga per usufruire dei servizi del centro e una *tipologia tariffa* che identifica, in base al prezzo della tariffa pagata dal cliente, in quale

sale o piscine può accedere. Di conseguenza più la tariffa pagata dal cliente è alta, più servizi avrà a disposizione quest'ultimo. La Tipologia Tariffa fa riferimento alla tipologia di contratto standard che mette a disposizione il centro (Per dettagli vedere paragrafo 2.1).

2.3 Individuazione degli attributi e degli identificatori

Nella tabella che segue verranno elencate tutte le Entità con i rispettivi Attributi. Tra parentesi verranno messi le entità padre di entità figlie, esempio “EntitàFiglia(EntitàPadre)”.

Entità	Attributi	Attributi Composti
--------	-----------	--------------------

Cliente	<u>CodFiscale</u> Cliente, Nome, Cognome, DataNascita, Sesso, IDDocumentoRiconoscimento, Prefettura	Indirizzo: Via, Città, NumCivico, C.A.P
ClienteOccasionale (Contratto)	TipologiaTariffa, TariffaAccesso	
CentroFitness	<u>IDCentroFitness</u> , CapienzaMassima, Dimensione, Telefono	Indirizzo: Via, Città, NumCivico, C.A.P
Calendario Centro Fitness	<u>IDCentroFitness</u> , NomeGiorno, OrarioApertura, OrarioChiusura	
Contratto	<u>IDContratto</u> , DataIscrizione, Durata, Scopo, AccessiSettimanali	
Standard (Contratto)	TipologiaContratto, CostoMensile	
Dipendente	<u>CodiceFiscale</u> , Nome, Cognome, DataNascita, Sesso, IDDocumentoRiconoscimento, Prefettura	Indirizzo: Via, Città, NumCivico, C.A.P
Accesso	<u>IDAccesso</u> , Data, OrarioEntrata, EntrataUscita	
Armadietto	<u>IDArmadietto</u> , CombinazioneNumerica	
Spogliatoio	<u>IDSpogliatoio</u> , Capienza, PostiDisponibili, Posizione, Sesso	
PianoPagamento	<u>IDPagamento</u> , ImportoTotale	
Rata	<u>IDRata</u> , DataScadenza, Stato, ImportoRata, IstitutoFinanziario, TassoInteresse	
FasciMuscolari	<u>NomeFasciMuscolari</u> , <u>LivelloPotenziamento</u>	
Misurazioni Impedimetriche	<u>IDMisurazione</u> , Data, AcquaTotale, MassaGrassa, MassaMagra, Altezza, Peso, StatoCliente, Entità	
SchedaAlimentazione	<u>IDSchedaAlimentazione</u> , DataInizio, Cliente, DataFine, Obiettivo, Medico	
Dieta	<u>IDDieta</u> , NumeroPasti, ApportoCaloricoGiornaliero	

Pietanza	<u>NomePietanza</u> , ComposizionePietanza	
SchedaAllenamento	<u>IDSchedaAllenamento</u> , DataInzio, DataFine, Tutor	
Esercizio	<u>IDEsercizio</u> , Nome, DispendioEnergeticoOrarioMedio	
Aerobico (Esercizio)	Durata	
Anaerobico (Esercizio)	Nome, NumeroEsecuzioni, TempoDiRecupero	
ConfigurazioneUtilizzo	<u>IDConfigurazione</u> , IstruzioneEsatta	
Attrezzo	<u>IDAttrezzo</u> , NomeAttrezzo, Tipologia, ConsumoEnergetico, LivelloUsura	
DatiRipetizione	<u>IDRipetizione</u> , NomeRipetizione, NumeroEsecuzioni, TempoDiRecupero	
MonitoraggioSmart	<u>IDMonitoraggio</u> , DataMonitoraggio	
OrarioDiLavoro	<u>Dipendente</u> , <u>IDCentro</u> , <u>InizioOra</u> , <u>FineOra</u> , <u>GiornoSettimana</u> , <u>Attività</u>	
Sala	<u>IDSala</u> , Nome, TipologiaContratto	
Piscina	<u>IDPiscina</u> , Dimensioni, Locazione, TipologiaContratto	
Corso	<u>IDCorso</u> , <u>Livello</u> , NomeDisciplina, NumeroMaxPartecipanti, DataInizio, DataFine	
CalendarioCorso	<u>IDCalendarioCorso</u> , GiornoSettimana, OrarioInizio, OrarioFine	
Utente	<u>Username</u> , Password, Credibilità	
Post	<u>IDPost</u> , Timestamp, Testo, Link	
Thread	<u>IDThread</u> , Titolo	
Voto	<u>IDVoto</u> , Stelle	
RichiestaAmicizia	<u>IDRichiesta</u> , UsernameDestinatario, StatoRichiesta	

Sfida	<u>Proponente, DataInizioSfida, DataScadenza, Scopo</u>	
Amico	<u>UsernameAmico, Utente</u>	
Cerchia	<u>NomeCerchia, Argomento</u>	

2.4 Individuazione e Descrizione delle Associazioni

Inseguito verranno descritte le associazioni utilizzate per collegare le varie entità.

Verranno mostrate, per motivi di chiarezza, due tabelle: nella prima verrà descritta ogni singola associazione con le cardinalità e i rispettivi attributi, nella seconda, verranno spiegati le motivazioni delle cardinalità inserite per ogni associazione.

Inseguito verrà mostrata la **tabella** che **descrive** l'associazione.

Associazione	Descrizione	Entità Coinvolte	Attributi
Archivio	Associa ad ogni Centro Fitness i Contratti che appartengono a quel Centro Fitness	Centro Fitness (1, N) Contratto (1, 1)	
Composizione Centro	Associa le sale che appartengono a quel Centro Fitness	Centro Fitness (1, N) Sala (1, 1)	
Orario Servizio	Associa ad ogni Centro Fitness un Calendario che mostra i giorni e le ore di apertura	Centro Fitness (1, 1) Calendario Centro Fitness (1, 1)	
Frequenza	Associa ad ogni Centro Fitness gli accessi che sono avvenuti	Centro Fitness (1, N) Accesso (1, 1)	

Composizione Piscine	Associa ad ogni Centro Fitness le Piscine presenti in quel centro	Centro Fitness (1, N) Piscina (1, 1)	
Direzione	Associa ad ogni Centro Fitness un Dipendente che il direttore del centro	Centro Fitness (1, 1) Dipendente (0, 1)	
Supervisione	Associazione ricorsiva che associa un dipendente Responsabile a un Impiegato	Dipendente Responsabile (1, N) Dipendente Impiegato (1, 1)	
Multisede	Associa a un Centro Fitness un Contratto con opzione Multisede	Centro Fitness (1, N) Contratto (0, 1)	Fascia Oraria, Ingressi Max
Programmazione Corso	Associa ad ogni Centro Fitness un Calendario di un Corso	Centro Fitness (0, N) Calendario Corso (1, 1)	
Intestatario	Associa ad ogni cliente il contratto che ha sottoscritto in un centro fitness	Cliente (1, 1) Contratto (1, 1)	
Effettuato	Associa ad un cliente i dati di accesso in un centro fitness	Cliente (0, N) Accesso (1, 1)	
Visita	Associa ad ogni cliente i dati di una visita impedenziometrica	Cliente (1, N) Misurazione Impedenziometrica(1,1)	
Alimentazione Passata	Associa a un cliente le schede di alimentazione passate	Cliente (0, N) Scheda Alimentazione (0, N)	

Alimentazione Attuale	Associa ad un cliente la scheda di alimentazione che sta utilizzando	Cliente (1, 1) Scheda Alimentazione (1, 1)	
Preferenze Alimentari	Associa ad una scheda di alimentazione una dieta	Scheda Alimentazione (1, 1) Dieta (0, N)	
Composizione Dieta	Associa ad ogni dieta serie di pietanze da prendere in un determinato pasto della giornata	Dieta (1, N) Pietanza (0, N)	Pasto
Tutor	Associa ad un cliente un dipendente tutor	Cliente (1, 1) Dipendente (0, N)	
Medico Nutrizionista	Associa ad un cliente un dipendente che è un medico nutrizionista	Cliente (1, 1) Dipendente (0, N)	
Corso Frequentato	Associa ad ogni cliente il/i corso/i che frequenta nel centro fitness	Cliente (0, N) Corso (1, N)	
Allenamento Passato	Associa ad ogni cliente le schede di allenamento passate	Cliente (0, N) Scheda Allenamento (0, N)	
Allenamento Attuale	Associa ad ogni cliente la scheda di allenamento attualmente utilizzata	Cliente (1, 1) Scheda Allenamento (1, 1)	
Assegnazione	Associa ad ogni accesso di un cliente un	Accesso (1, 1) Armadietto (1, 1)	

	armadietto del centro fitness		
Collocazione	Associa ad ogni armadietto lo spogliatoio dove è collocato	Armadietto (1, 1) Spogliatoio (1, N)	
Dotazione	Associa ad ogni sala del centro gli attrezzi presenti in essa	Sala (1, N) Attrezzo (1, 1)	
Responsabile Sala	Associa ad ogni sala un dipendente che ne è responsabile	Sala (1, 1) Dipendente (0, N)	
Turnazione Dipendente	Associa ad un dipendente il proprio orario di lavoro	Dipendente (1, N) Orario Lavoro (1, 1)	
Turnazione Centro	Associa ad un centro fitness gli orari di lavoro che si svolgono in quel centro	Centro Fitness (1, N) Orario di Lavoro (1, 1)	
Istruttore Corso	Associa un dipendente istruttore un corso da lui diretto	Dipendente (0, N) Corso (1, 1)	

Tutor Scheda	Associa ad una scheda di allenamento un dipendente tutor che l'ha assegnata ad un cliente	Scheda Allenamento (1, 1) Dipendente (0, N)	
--------------	---	--	--

Medico Scheda	Associa ad una scheda di alimentazione un dipendente medico nutrizionista che l'ha assegnata d un cliente	Scheda Alimentazione (1, 1) Dipendente (0, N)	
Consulente Contratto	Associa ad un contratto consulente dipendente del centro	Contratto (1, 1) Dipendente(1, N)	
Composizione Scheda	Associa ad una scheda di allenamento una serie di Esercizi	Scheda Allenamento (1, N) Esercizio (0, N)	
Attrezzo Utilizzato	Associa ad un monitoraggio l'attrezzo che viene utilizzato per svolgere l'esercizio	Monitoraggio Smart (0, N) Attrezzo (1, N)	
Regolazione	Associa l'attrezzo alla configurazione utilizzato che deve essere adottata per quell'attrezzo	Attrezzo (0, N) Configurazione Utilizzo (1, 1)	Nome Regolazione
Istruzione Intensità	Associa ad un esercizio una configurazione di utilizzo tramite l'intensità che deve essere inserita in	Esercizio (0, N) Configurazione Utilizzo (1, 1)	Intensità

	una particolare regolazione		
Configurazione Ripetizione	Associa un esercizio di tipo Anaerobico ad una particolare configurazione di utilizzo	Anaerobico (0, 1) Configurazione Utilizzo (0, 1)	
Esecuzione	Associa ad un esercizio il modo con cui un cliente lo ha eseguito	Esercizio (1, N) Monitoraggio Smart (1, 1)	Istante Inizio Istante Fine
Configurazione Cliente	Associa ad un monitoraggio smart la configurazione di un attrezzo che un cliente ha effettivamente utilizzato	Monitoraggio Smart (0, 1) Configurazione Utilizzo (0, 1)	
Monitoraggio Ripetizione	Associa ad un monitoraggio smart dei dati su ogni ripetizione dell'esercizio nel caso in cui quest'ultima comprenda delle ripetizioni	Monitoraggio Smart (0, 1) Dati Ripetizione (0, N)	
Pagamento	Associa ad un contratto il tipo di pagamento che il cliente ha deciso di scegliere per pagare il contratto	Contratto (1, 1) Piano Pagamento (1, 1)	
Rateizzato	Associa al Piano Pagamento una possibile rateizzazione di quest'ultimo	Piano Pagamento (0, 1) Rata (1, 1)	

Potenziamento	Associa ad un contratto una serie di muscoli nel caso in cui il cliente scelga uno scopo di tipo “potenziamento”	Contratto (0, 1) Fasci Muscolari (1, N)	
Corso Disponibile	Associa ad un contratto una serie di corsi che, in base al contratto che ha sottoscritto il cliente, il centro mette a disposizione per il cliente	Contratto (0, N) Corso (0, N)	
Accesso Piscina	Associa ad un contratto la possibilità o meno di accedere ad una o più piscine del centro fitness	Contratto (0, N) Piscina (0, N)	Numero Accessi
Corso In Sala	Associa ad un corso la sala in cui viene svolto	Corso (0, 1) Sala (0, N)	
Corso In Piscina	Associa ad un corso la piscina in cui viene svolto	Corso (0, 1) Piscina (0, N)	
Programmazione Centro	Associa ad un calendario di un corso il centro in cui viene seguito	Centro Fitness (1, N) Calendario Corso (1, 1)	
Profilo Online	Associa ad un contratto un profilo online messo a disposizione per il cliente	Contratto (1, 1) Utente (1, 1)	

Forum	Associa ad un utente il post di in un determinato forum	Utente (0, N) Post (1, 1)	Area Forum
Post Risposta	Associa ad ogni un Post Principale un post risposta	Post Principale (0, N) Risposta (1, 1)	
Giudizio	Associa ad un post un determinato voto	Post (0, N) Voto (1, 1)	
Thread	Associa ad un post in quale thread è avvenuta la pubblicazione	Post (1, 1) Thread (1, N)	
Attività Social	Associa ad un utente le richieste di amicizia inviate da lui stesso ad un altro cliente	Utente (0, N) Richiesta Amicizia (1, 1)	
Amicizia	Associa ad un utente un amico	Utente (0, N) Amico (1, 1)	
Partecipante	Associa ad un amico la partecipazione o meno ad una sfida	Amico (0, 1) Sfida (0, N)	
Appartenenza	Associa un amico di un utente ad una o più cerchie	Amico (0, N) Cerchia (1, N)	
Suggerimento Inserimento	Associa ad una cerchia l'inserimento di un amico che non ne ha parte	Cerchia (0, N) Amico (0, N)	
Thread Sfida	Associa ad una sfida un particolare thread	Sfida (1, 1) Thread (0, 1)	
Lancio Sfida	Associa ad un utente una sfida che lui stesso ha lanciato	Utente (0, N) Sfida (1, 1)	

Scheda Alimentazione Sfida	Associa ad una sfida una scheda di alimentazione	Sfida (1, 1) Scheda Alimentazione (1, N)	
Scheda Allenamento Sfida	Associa ad una sfida una scheda di allenamento	Sfida (1, 1) Scheda Allenamento (1, N)	
Vincitore	Associa ad una sfida un utente che ha vinto la sfida	Sfida (1, 1) Utente (0, 1)	

La seguente tabella mostra le **motivazioni** delle varie **cardinalità**.

Associazione	Cardinalità 1	Cardinalità 2
--------------	---------------	---------------

Archivio	Centro Fitness (1, N) con Contratto, in quanto un centro fitness possiede più contratti	Contratto (1, 1) con Centro Fitness, in quanto un contratto può essere stato sottoscritto in un solo Centro Fitness
Composizione Centro	Centro Fitness (1, N) con Sala, in quanto un centro fitness possiede al suo interno più sale	Sala (1, 1) con Centro Fitness, in quanto una determinata sala, avendo un codice unico in tutti centri fitness, appartiene ad un solo centro fitness
Orario Servizio	Centro Fitness (1, 1) con Calendario Centro Fitness, in quanto un centro fitness ha un unico calendario	Calendario Centro Fitness (1, 1) con Centro Fitness, in quanto ogni centro, per motivi di turnazione, ha un calendario diverso
Frequenza	Centro Fitness (1, N) con Accesso, in quanto in un centro fitness ci sono molti accessi dei clienti	Accesso (1, 1) con Centro Fitness, in quanto ogni accesso, grazie al suo codice univoco, può essere associato ad un solo centro fitness
Composizione Piscine	Centro Fitness (1, N) con Piscina, in quanto un centro fitness possiede una o più piscine	Piscina (1, 1) con Centro Fitness, in quanto ogni piscina, grazie al suo codice univoco, può appartenere ad un solo centro fitness
Direzione	Centro Fitness (1, 1) con Dipendente, in quanto in un centro fitness c'è un solo dipendente che è il direttore del centro	Dipendente (0, 1) con Centro Fitness, in quanto tra i dipendenti ci può essere un solo direttore, quindi un dipendente può o non essere il direttore del centro fitness
Supervisione	Dipendente Responsabile (1, N) con Dipendente Impiegato, in quanto un responsabile può avere in	Dipende Impiegato (1, 1) con Dipende Responsabile, in quanto ogni dipendente del centro

	un centro fitness uno o più dipendente sotto la sua supervisione	è supervisionato da un e uno solo responsabile
Multisede	Centro Fitness (1, N) con Contratto, in quanto uno o più centri fitness (per precisione al massimo 3) possono essere associato ad un contratto con opzione multisede	Contratto (0, 1) con Centro Fitness, in quanto la modalità multisede è modalità opzionale del contratto che un cliente sottoscrive
Programmazione Corso	Centro Fitness (1, N) con Calendario Corso, in quanto in centro fitness sono presenti una serie di programmazioni dei vari corsi presenti in esso	Calendario Corso (1, 1) con Calendario Corso, in quanto ogni Calendario Corso, grazie al suo codice univoco, può essere associato ad un solo centro fitness specifico
Intestatario	Cliente (1, 1) con Contratto, in quanto ad un cliente del centro fitness può essere associato uno e un solo contratto	Contratto (1, 1) con Cliente, in quanto ogni contratto è associato ad uno ed un solo cliente del centro fitness
Effettuato	Cliente (0, N) con Accesso, in quanto un cliente è possibile che ancora non abbiamo effettuato alcun accesso in un centro fitness o ne può aver fatti molti accessi in un centro	Accesso (1, 1) con Cliente, in quanto ogni accesso, grazie al suo codice univoco, può essere associato ad uno ed un solo cliente
Visita	Cliente (1, N) con Misurazione Impedenziometrica, in quanto ad un cliente vengono fatte, nel corso	Misurazione Impedenziometrica (1, 1) con Cliente, in quanto ogni misurazione, grazie al suo

	del suo percorso sportivo, varie misurazioni impedenziometriche	codice univoco, può essere associata ad un solo cliente
Alimentazione Passata	Cliente (0, N) con Scheda di Alimentazione, in quanto un cliente può avere alcuna scheda di alimentazione passata ma ne può avere un numero illimitato	Scheda Alimentazione (0, N) con Cliente, in quanto schede di alimentazione associate ad un cliente ce ne possono essere zero o un numero illimitato
Alimentazione Attuale	Cliente (1, 1) con Scheda Alimentazione, in quanto un cliente può avere una ed una sola scheda di alimentazione che sta seguendo attualmente	Scheda Alimentazione (1, 1) con Cliente, in quanto ogni scheda di alimentazione attuale è associata ad uno ed un solo cliente
Preferenze Alimentari	Scheda Alimentazione (1, 1) con Dieta, in quanto ad ogni scheda di alimentazione è associata una ed una sola dieta	Dieta (0, N) con Scheda Alimentazione, in quanto una dieta può essere associata o meno a più schede di alimentazione
Composizione Dieta	Dieta (1, N) con Pietanza, in quanto una dieta può essere composta da una o più pietanze	Pietanza (0, N) con Dieta, in quanto una pietanza può far parte o meno in più diete presenti nel database
Tutor	Dipendente (0, N) con Cliente, in quanto un dipendente tutor può avere a suo carico zero o più clienti del centro fitness	Cliente (1, 1) con Dipendente, in quanto un cliente può avere uno ed un solo tutor durante la validità del contratto
Medico Nutrizionista	Dipendente (0, N) con Cliente, in quanto un dipendente medico nutrizionista può avere a suo carico zero o più clienti del centro fitness	Cliente (1, 1) con Dipendente, in quanto un cliente può avere no ed un solo dipendente medico nutrizionista durante la validità del contratto

Corso Frequentato	Cliente (0, N) con Corso, in quanto un cliente può frequentare nessun corso o ne può frequentare più di uno tra quelli messi a disposizione dal contratto	Corso (1, N) con Cliente, in quanto in un corso ci possono essere uno o più clienti che lo frequentano
Allenamento Passato	Cliente (0, N) con Allenamento Passato, in quanto un cliente può o non avere delle vecchie schede di allenamento che non segue più	Allenamento Passato (0, N) con Cliente, in quanto ci possono essere zero o più schede di allenamento passate associate ai clienti
Allenamento Attuale	Cliente (1, 1) con Scheda Allenamento, in quanto un cliente del centro può essere associata una ed una sola scheda di allenamento che sta utilizzando attualmente	Allenamento Attuale (1, 1) con Cliente, in ogni scheda di allenamento attualmente utilizzata è associata ad un e un solo cliente del centro fitness
Assegnazione	Accesso (1, 1) con Armadietto, in quanto ad ogni accesso nel centro fitness viene assegnato uno ed un solo armadietto	Armadietto (1, 1) con Accesso, in quanto ogni armadietto viene assegnato al momento di un ed un solo accesso
Collocazione	Armadietto (1, 1) con Spogliatoio, in quanto un armadietto è collocato in un solo spogliatoio del centro fitness	Spogliatoio (1, N) con Armadietto, in quanto in uno spogliatoio sono collocati uno o più armadietti
Dotazione	Sala (1, N) con Attrezzo, in quanto una sala sono presenti uno o più attrezzi	Attrezzo (1, 1) con Sala, in quanto un attrezzo può essere presente in una ed una sola sala
Responsabile Sala	Sala (1,1) con Dipendente, in quanto per ogni sala c'è un ed un solo dipendente che ne è il responsabile	Dipendente (0, N) con Sala, in quanto tra i dipendenti ci possono essere zero o più dipendenti i quali ogni è

		responsabile di una sala diversa
Turnazione Dipendente	Dipendente (1, N) con Orario di Lavoro, in quanto un dipendente può avere uno o più turni di lavoro	Orarti di Lavoro (1, 1) con Dipendente, in quanto ad un orario di lavoro è associato uno ed un solo dipendente
Turnazione Centro	Centro Fitness (1, N) con Orario di Lavoro, in quanto in un centro fitness sono associati uno o più di un orario di lavoro dei vari dipendenti	Orario di Lavoro (1, 1) con Centro Fitness, in quanto ogni orario di lavoro è riferito ad un solo centro fitness
Istruttore Corso	Dipendente (0, N) con Corso, in quanto tra i dipendenti ci possono essere o meno dipendenti che istruttori di corsi	Corso (1, 1) con Dipendente, in quanto un corso possiede uno ed un solo dipendente istruttore
Tutor Scheda	Scheda Allenamento (1, 1) con Dipendente, in quanto c'è un solo tutor associato ad un determinata scheda	Dipendente (0, N) con Scheda Allenamento, in quanto un tutor può assegnare zero o più schede di allenamento
Medico Scheda	Scheda Alimentazione (1, 1) con Dipendente, in quanto ad ogni scheda di alimentazione è associato un ed un solo dipendente medico nutrizionista	Dipendente (0, N) con Scheda Alimentazione, in quanto un dipendente medico nutrizionista può aver assegnato zero o più schede di alimentazione
Corso In Sala	Corso (0, 1) con Sala in quanto un corso si può sostenere o meno in una sala	Sala (0, N) con Corso poiché in una sala si possono svolgere nessuno o più corsi
Corso In Piscina	Corso (0, 1) con Piscina in quanto un corso si può sostenere o meno in una piscina	Piscina (0, N) con Corso poiché in una piscina si possono svolgere nessuno o più corsi

Programmazione Centro	Calendario Corso (1, 1) con Centro Fitness in quanto il calendario è associato ad un solo centro fitness	Centro Fitness (1, N) con Calendario Fitness poiché in un centro fitness si svolgono più corsi e di conseguenza ci sono più calendari per quanto riguarda i corsi
Composizione Scheda	Scheda Allenamento (1, N) con Esercizio, in quanto una scheda di allenamento può essere composta da uno o più esercizi	Esercizio (0, N) con Scheda Allenamento, in quanto un esercizio può far parte o meno tra gli esercizi di una scheda di allenamento
Attrezzo Utilizzato	Esercizio (0, N) con Attrezzo, in quanto per un esercizio può non essere utilizzato nessun attrezzo ma ne possono servire anche tanti	Attrezzo (1, N) con Esercizio, in quanto un attrezzo può servire per svolgere uno o più esercizi
Regolazione	Attrezzo (0, N) con Configurazione Utilizzo, in quanto un attrezzo può avere zero o più regolazioni che posso essere utili per la configurazione di utilizzo	Configurazione Utilizzo (1, 1) con Attrezzo, in quanto una determinata regolazione per la configurazione di utilizzo può essere associata ad un ed un solo attrezzo
Istruzione Intensità	Esercizio (0, N) con Configurazione Utilizzo, in quanto un esercizio può avere zero o più istruzioni intensità per completare la configurazione di utilizzo	Configurazione Utilizzo (1, 1) con Esercizio, in quanto ogni configurazione di utilizzo può essere relativa ad un ed un solo esercizio
Esecuzione	Esercizio (1, N) con Monitoraggio Smart, in quanto un esercizio può essere stato monitorato più volte	Monitoraggio Smart (1, 1) con Esercizio, in quanto un monitoraggio smart è relativo ad uno ed un solo esercizio
Configurazione Cliente	Monitoraggio Smart (0, 1) con Configurazione Utilizzo, in quanto non è detto che il monitoraggio	Configurazione Utilizzo (0, 1) con Monitoraggio Smart, in quanto una configurazione di utilizzo

	di un esercizio preveda una configurazione di utilizzo di un attrezzo	può essere o meno associata ad un monitoraggio smart
Monitoraggio Ripetizione	Monitoraggio Smart (0, 1) con Dati Ripetizione, in quanto il monitoraggio di un esercizio può prevedere delle ripetizioni o meno	Dati Ripetizione (0, N) con Monitoraggio Smart, in quanto a dei dati ripetizione è associato uno ed un solo monitoraggio smart
Pagamento	Contratto (1, 1) con Piano di Pagamento, in quanto ad un contratto è associato uno ed un solo piano di pagamento	Piano di Pagamento (1, 1) con Contratto, in quanto un piano di pagamento è associato ad uno ed un solo contratto
Rateizzato	Piano Pagamento (0, 1) con Rata, in quanto il pagamento di un contratto può venire o meno rateizzato tramite l'utilizzo di rate	Rata (1, 1) con Piano Pagamento, in quanto una rata può essere associata ad un solo piano di pagamento
Potenziamento	Contratto (0, 1) con Fasci Muscolari in quanto un contratto può avere l'opzione di potenziamento o meno	Fasci Muscolari (1, N) con Contratto, in quanto in caso di potenziamento uno o più fasci muscolari possono interessare quest'ultimo
Corso Disponibile	Contratto (0, N) con Corso, in quanto un contratto può mettere a disposizione del cliente zero o più corsi	Corso (0, N) con Contratto, in quanto un nessun corso o tanti possono essere inseriti in un contratto
Accesso Piscina	Contratto (0, N) con Piscina, in quanto ci possono essere contratti che non permettono l'accesso alle piscine e contratti che permettono l'accesso a più piscine	Piscina (0, N) con Contratto, in quanto nessuna piscina o più piscine possono essere messe in un contratto per permettere l'accesso al cliente
Consulente Contratto	Contratto (1, 1) con Dipendente in quanto ad un contratto è associato un	Dipendente (1, N) con Contratto in quanto ad un dipendente consulente

	solo dipendente consulente	possono essere associati uno o più contratti
Profilo Online	Contratto (1, 1) con Utente, in quanto ogni contratto associa un ed un solo utente	Utente (1, 1) con Contratto, in quanto ad un utente è associato uno ed un solo contratto
Forum	Utente (0, N) con Post, in quanto un utente può pubblicare nessuno o più post in un forum	Post (1, 1) con Utente, in quanto un post è associato ad un ed un solo utente
Post Risposta	Post Principale (0, N) con Risposta, in quanto un post può avere nessuna o più risposte	Risposta (1, 1) con Post Principale, in quanto ad una risposta è associato un ed un solo post
Giudizio	Post (0, N) con Voto, in quanto un post può avere nessun o più voti i quali poi influenzeranno la credibilità dell'utente	Voto (1, 1) con Post, in quanto ogni voto, grazie al proprio codice univoco, è relativo ad un solo Post
Thread	Post (1, 1) con Thread, in quanto ogni post è associato ad un solo thread	Thread (1, N) con Post, in quanto ci sono uno o più post in un thread
Attività Social	Utente (0, N) con Richiesta Amicizia, in quanto un utente può aver inviato o nessuna o tante richieste di amicizia	Richiesta Amicizia (1, 1) con Utente, in quanto a ogni richiesta di amicizia è associato un solo cliente
Amicizia	Utente (1, N) con Amico, in quanto un utente può avere nessun amico ma ne può avere anche molti	Amico (1, 1) con Utente, in quanto l'amicizia è relativa ad un utente solo
Partecipante	Amico (0, 1) con Sfida, in quanto un amico può partecipare o meno ad una sfida	Sfida (0, N) con Amico, in quanto ad una sfida possono partecipare nessuno o più amici dell'utente
Appartenenza	Amico (0, N) con Cerchia, in quanto un amico può appartenere a nessuna o più cerchie dell'utente	Cerchia (1, N) con Amico, in quanto ad un cerchia posso essere presenti uno o più amici

Suggerimento Inserimento	Cerchi (0, N) con Amico, in quanto in una cerchia ci possono essere suggerimenti di inserimento di nessun o molti amici dell'utente	Amico (0, N) con Cerchia, in quanto un amico può essere un suggerimento inserimento per nessuna o per più cerchie
Thread Sfida	Sfida (1, 1) con Thread, in quanto ad una sfida è associato uno ed un solo thread sfida	Thread (0, 1) con Sfida, in quanto un thread può essere o meno un thread relativo ad una sfida
Lancio Sfida	Utente (0, N) con Sfida, in quanto un utente può lanciare nessuna o più sfide	Sfida (1, 1) con Utente, in quanto una sfida può essere lanciata da un solo utente
Scheda Alimentazione Sfida	Sfida (1, 1) con Scheda Alimentazione, in quanto ad una sfida è associata una ed una scheda di alimentazione	Scheda Alimentazione (1, N) con Sfida, in quanto i partecipanti posso utilizzare schede di alimentazioni differenti da quella proposta dal proponente della sfida
Scheda Allenamento Sfida	Sfida (1, 1) con Scheda Allenamento, in quanto ad una sfida è associata una ed una scheda di allenamento	Scheda Allenamento (1, N) con Sfida, in quanto i partecipanti posso utilizzare schede di allenamento differenti da quella proposta dal proponente della sfida
Vincitore	Sfida (1, 1) con Utente poiché la sfida può avere un solo vincitore	Utente (0, 1) con Sfida in quanto un utente può essere vincitore o meno di una sfida

2.5 Individuazione delle ridondanze

Nel diagramma E-R sono presenti alcune ridondanze riportate qui di seguito.

Una ridondanza costituisce un'informazione derivabile in modi diversi dal diagramma E-R, la quale può comunque dimostrarsi utile in fase di ottimizzazione delle operazioni.

Ridondanza 1:

L'entità "MONITORAGGIO SMART", l'attributo "Tempo Impiegato", così come è stato definito, costituisce una ridondanza, infatti il tempo impiegato per svolgere un esercizio può essere ricavato sottraendo "Istante Fine" e "Istante Inizio" dell'associazione "ESECUZIONE".

Ridondanza 2:

Nell'associazione "MONITORAGGIO RIPETIZIONE", l'attributo "Numero Ripetizioni", così com'è formulata è una ridondanza, infatti il numero di ripetizioni può essere ricavato contando le occorrenze dell'entità "DATI RIPETIZIONE".

Ridondanza 3:

Nell'associazione "RATEIZZATO", l'attributo "Numero Rate", così com'è stato definito, costituisce una ridondanza, infatti il numero di rate di un pagamento si può ricavare le occorrenze dell'associazione "RATEIZZATO" in cui compare quel pagamento.

Ridondanza 4:

L'associazione "TUTOR" così come è stata definita rappresenta una ridondanza in quanto il tutor di un cliente si può ricavare vedendo quale tutor ha assegnato la scheda di allenamento al cliente.

Ridondanza 5:

L'associazione "MEDICO NUTRIZIONISTA", così come è stata definita rappresenta una ridondanza in quanto il medico nutrizionista del cliente può essere ricavato dalla scheda di alimentazione associata al cliente

Ridondanza 6:

Nell'entità "ATTREZZO", l'attributo "Livello Usura" è una ridondanza in quanto potrebbe essere calcolato contando le occorrenze dell'associazione "Attrezzo Utilizzato" calcolando ogni occorrenza come un valore pari a "0.1".

Ridondanza 7:

Nell'entità "SPOGLIATOIO", l'attributo "Posti Disponibili" è una ridondanza in quanto può essere ricavato dalle occorrenze dell'associazione "COLLOCAZIONE" e dall'attributo "Capienza" dell'entità "SPOGLIATOIO".

Ridondanza 8:

Nell'entità "DIETA", l'attributo "Numero Pasti" costituisce una ridondanza, infatti il numero di pasti di una dieta può essere calcolato contando le occorrenze dell'associazione "COMPOSIZIONE DIETA".

Ridondanza 9:

Nell'entità "PIANO PAGAMENTO", l'attributo "Importo totale" costituisce una ridondanza, infatti il costo di totale di un contratto può essere calcolato, se non rateizzato, dal costo mensile e dalla durata dell'entità "CONTRATTO", o dalla somma degli importi delle rate dell'entità "RATA".

3 Ristrutturazione del Diagramma E-R

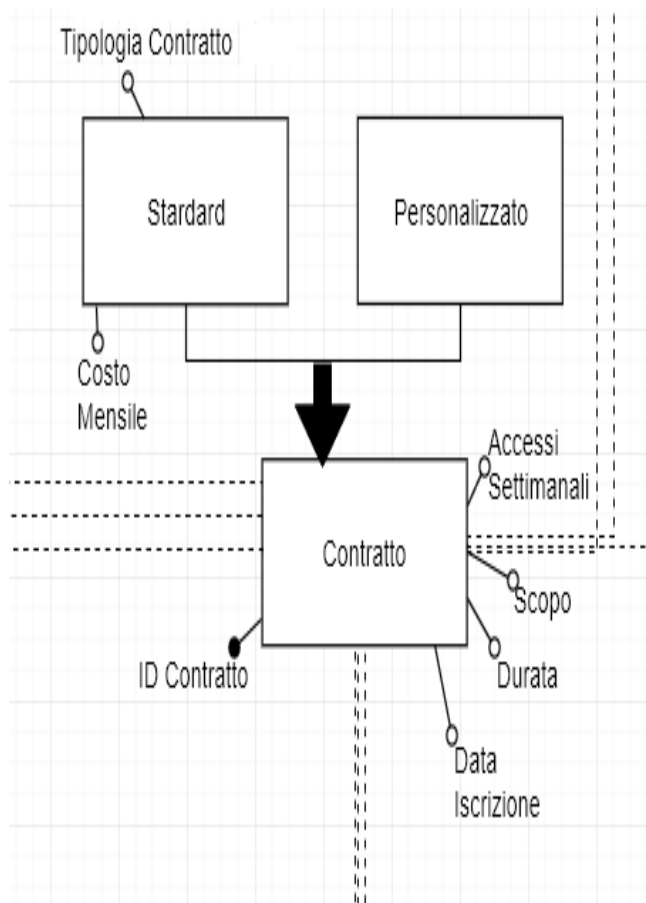
3.1 Risoluzione delle Generalizzazioni

Nel diagramma in questione sono stati inserite 3 generalizzazioni, le quali risoluzioni verranno illustrate in seguito:

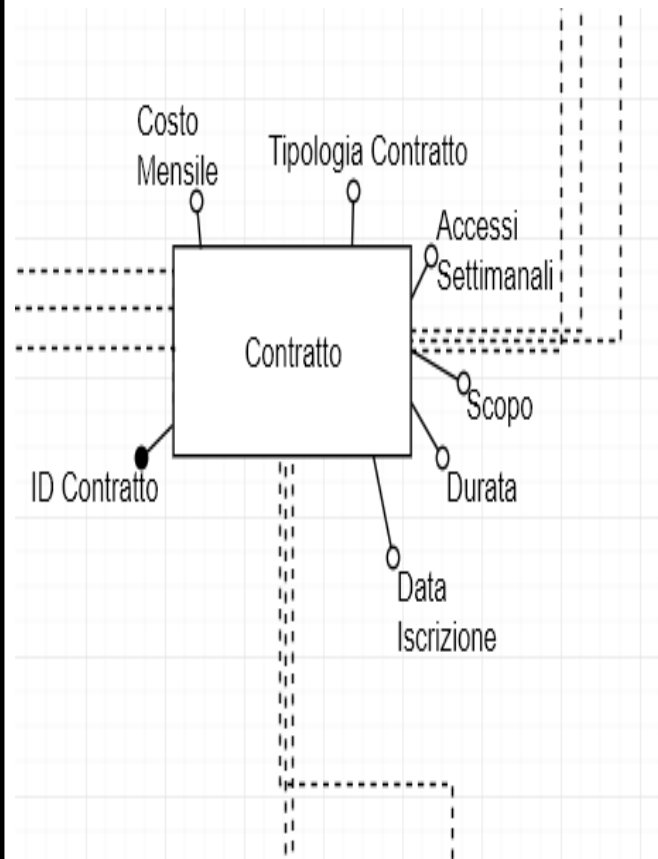
1. GENERALIZZAZIONE CONTRATTO:

Tale generalizzazione viene risolta accorpendo le entità figlie nell'entità padre e si trasferiscono le loro proprietà, appunto, sull'entità padre. È stata questa tipo di risoluzione poiché la base di dati tratta principalmente occorrenze del dell'entità padre.

Prima:



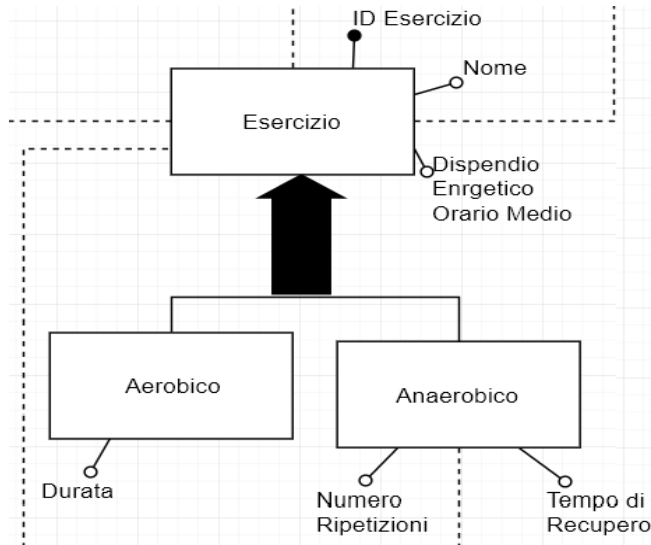
Dopo:



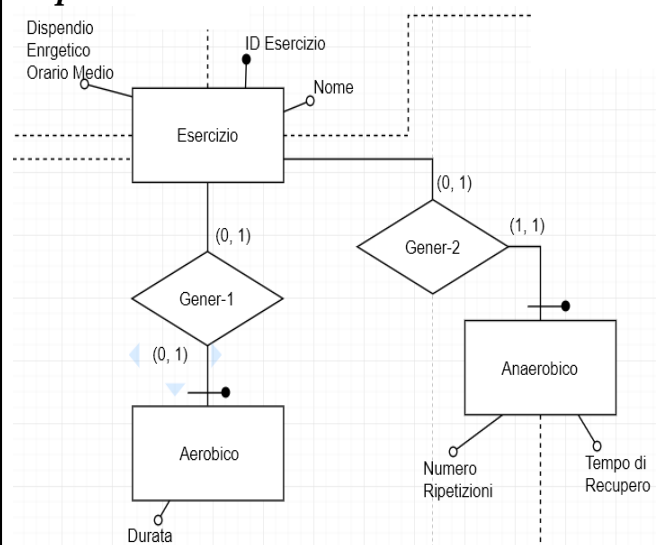
2. GENERALIZZAZIONE ESERCIZIO:

Tale generalizzazione viene risolta tramite nessun accorpamento aggiungendo due associazioni “GENERALIZZAZIONE 1” e “GENERALIZZAZIONE 2”, dove entrambe le generalizzazioni partecipano con l’entità padre con cardinalità (0, 1) e (1, 1) con identificatore esterno. È stato scelto questo tipo di risoluzione poiché la base di dati in questione tratta in modo diverso le occorrenze dell’entità padre e le occorrenze delle entità figlie.

Prima:



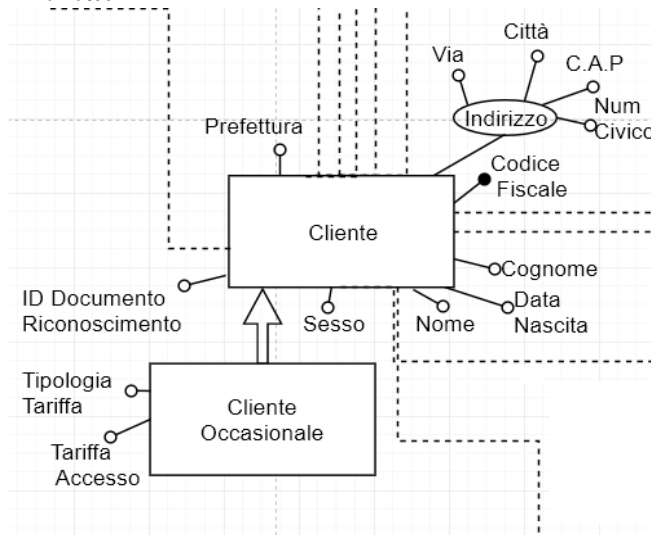
Dopo:



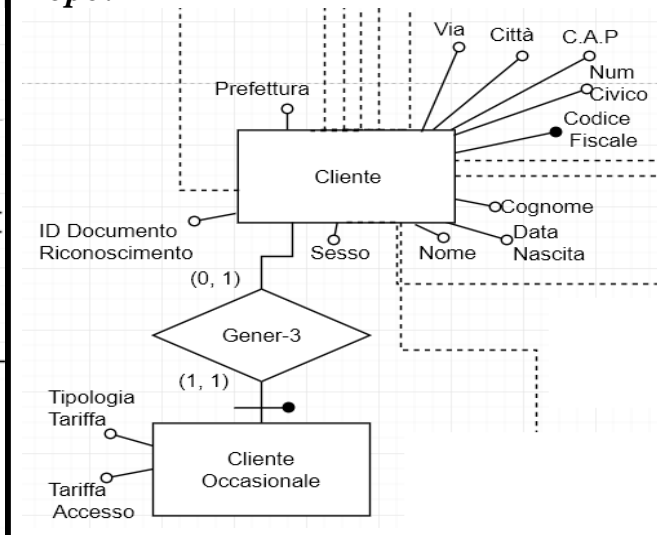
3. GENERALIZZAZIONE CLIENTE:

Questa generalizzazione è stata risolta tramite nessun accorpamento inserendo una associazione “GENERALIZZAZIONE 3” tra le entità “Cliente” e “Cliente Occasionale” con cardinalità rispettivamente (0, 1) e (1, 1) con identificatore esterno.

Prima:



Dopo:



3.2 Risoluzione degli attributi composti

Per maggiore leggibilità risolviamo con degli *Attributi Semplici* l'attributo *Composto* "Indirizzo"

Attributo Composto	Descrizione	Entità Coinvolte	Attributi Atomici
Indirizzo	Informazioni relative al C.A.P, Numero Civico, Città, Via	Centro Fitness Cliente Dipendente	C.A.P Numero Civico Città Via

3.3 Risoluzione degli attributi multivalore

Nel diagramma E-R non ristrutturato sono presenti 2 *Attributi Multivalore* che sono stati "sostituiti" da Entità appositamente create con le rispettive Associazioni.

Attributo Multivalore	Cardinalità	Descrizione	Entità Coinvolta	Ristrutturazione
Link	(0, N)	Link che eventualmente vengono inseriti in un post	POST	È stata creata una nuova entità LINK che possiede il solo attributo <u>URL</u> . LINK è associato a POST mediante "LINK POST" a cui le due entità partecipano rispettivamente con cardinalità (1, N) e (0, N)
Interesse	(1, N)	Interessi che un utente aggiunge per completare il suo profilo online	UTENTE	È stata creata una nuova entità INTERESSE che possiede il solo attributo <u>Nome Interesse</u> . INTERESSE è associato a UTENTE mediante "INTERESSE UTENTE" a cui le due entità partecipano rispettivamente con cardinalità (1, N) e (0, N)

Nota: La porzione di diagramma presente nelle immagini precedenti per spiegare le risoluzioni delle generalizzazioni potrebbe non essere uguale alla relativa parte del diagramma E-R finale.

4 Individuazione operazioni significative

4.1 Lista delle operazioni

***Operazione 1:**

Descrizione: Modificare lo stato di una rata non ancora pagata di un contratto rateizzato

Frequenza: 20 volte al giorno

***Operazione 2:**

Descrizione: Inserimento di una nuova scheda di allenamento

Frequenza: 1000 volte al giorno

***Operazione 3:**

Descrizione: Inserire contratto con scopo potenziamento

Frequenza: 50 volte al giorno

***Operazione 4:**

Descrizione: Inserimento di una Nuova visita di un cliente con controllo dello stato e della sua entità

Frequenza: 20 volte al giorno

***Operazione 5:**

Descrizione: Trovare l'attrezzo che ha bisogno di Manutenzione dopo inserimento del monitoraggio di un esercizio

Frequenza: 100 volte al giorno

***Operazione 6:**

Descrizione: Trovare i Post in un determinato Thread

Frequenza: 100 volte al giorno

***Operazione 7:**

Descrizione: Trovare il numero di ripetizioni effettuate da un cliente per l'esercizio con Dispendio energetico medio massimo in un determinato giorno

Frequenza: 300 volte al giorno

***Operazione 8:**

Descrizione: Trovare quanti posti sono disponibili in un determinato Spogliatoio

Frequenza: 1000 volte al giorno:

4.1 Tavola dei volumi

Per poter analizzare le operazioni significative sopra elencate sopra bisogna prima procedere alla determinazione dei volumi di ciascuna entità e associazione. In mancanza di numero precisi, si è cercato di effettuare delle stime più realistiche possibili.

NOTA: Viene preso in considerazione l'arco temporale di un anno in alcune stime.

Nome	E/R	Numero Istanze	Motivazione
Cento Fitness	E	100	L'azienda possiede 100 centri fitness diverse
Composizione Centro	R	1500	In media ci sono 15 sale in un centro fitness $1500 = 100 \times 15$
Sala	E	1500	Cardinalità (1, 1) con Composizione Centro
Orario Servizio	R	100	In media c'è un solo calendario per ogni centro fitness
Calendario Centro Fitness	E	100	Cardinalità (1, 1) con Orario Servizio
Composizione Piscine	R	400	In media in ogni centro fitness ci sono 4 piscine $400 = 100 \times 4$
Piscina	E	400	Cardinalità (1, 1) con Composizione Piscine
Direzione	R	100	In ogni centro fitness c'è un solo direttore
Responsabile Sala	R	1500	Ogni sala ha un solo responsabile
Cliente	E	80000	Ogni centro fitness ha in media 800 clienti $80000 = 100 \times 800$
Gener-3	R	8000	In media 1 cliente su 10 è cliente occasionale $8000 = 80000 / 10$
Cliente Occasionale	E	8000	Cardinalità (1, 1) con Gener-3
Intestatario	R	80000	Ogni contratto ha un solo intestatario

Contratto	E	80000	Cardinalità (1, 1) con Intestatario
Multisede	R	4000	In media 1 contratto su 20 è Multisede $4000 = 80000 / 20$
Archivio	R	80000	Ogni centro fitness ha in media 800 contratti
Dipendente	E	7000	In media in un centro fitness ci sono 70 dipendenti $7000 = 70 \times 100$
Supervisione	R	700	In media un dipendente è responsabile per 10 dipendenti $700 = 7000/10$
Effettuato	R	115200	In media un cliente effettua 3 accessi a settimana $115200 = 800 \times 48$
Accesso	E	115200	Cardinalità (1, 1) con Effettuato
Frequenza	E	115200	Cardinalità (1, 1) con Accesso
Spogliatoio	E	400	In media in un centro ci sono 4 spogliatoi $400 = 4 \times 100$
Assegnazione	R	115200	Cardinalità (1, 1) con Acceso
Collocazione	R	12000	In uno spogliatoio sono collocati 30 armadietti $12000 = 30 \times 400$
Armadietto	E	12000	Cardinalità (1, 1) con Collocazione
Visita	R	560000	In media ad ogni cliente vengo fatte 7 visite $560000 = 80000 \times 7$
Misurazioni Impedenziometriche	E	560000	Cardinalità (1, 1) con Visita
Alimentazione Passata	R	160000	Ogni cliente in media ha 2 schede di alimentazione che non utilizza più $160000 = 80000 \times 2$
Alimentazione Attuale	R	80000	Una sola Scheda Alimentazione attuale

Scheda Alimentazione	E	240000	$240000 =$ Alimentazione attuale (80000) + Alimentazione Passata (160000)
Preferenza Alimentari	R	240000	Cardinalità (1, 1) con Scheda Alimentazione
Dieta	E	2400	In media una dieta è presente in 100 Schede Alimentazione $2400 = 240000 / 100$
Composizione Dieta	R	7200	In media ogni dieta è composta tra 3 pietanze $7200 = 2400 \times 3$
Pietanza	E	480	Una pietanza in media appartiene a 5 diete $480 = 2400 / 5$
Potenziamento	R	40000	In media 1 su 4 contratti comprende lo scopo potenziamento e allena 2 fasci muscolari $20000 = (80000 / 4) \times 2$
Fasci Muscolari	E	5	Fasci Muscolari che si possono allenare sono 5
Pagamento	R	80000	Cardinalità (1, 1) con Contratto
Piano Pagamento	E	80000	Cardinalità (1, 1) con Pagamento
Rateizzato	R	40000	In media 1 su 2 è rateizzato $40000 = 80000 / 2$
Rata	E	240000	In media ogni pagamento rateizzato ha 6 rate $240000 = 6 \times 40000$
Allenamento Passato	R	160000	In media un cliente ha 2 schede di allenamento passate $160000 = 80000 \times 2$
Allenamento Attuale	R	80000	Cardinalità (1, 1) con Cliente

Scheda Allenamento	E	240000	$240000 = \text{Allenamento Passato (160000)} + \text{Allenamento Attuale (80000)}$
Composizione Scheda	R	2400000	In media ogni scheda contiene 10 Esercizi $2400000 = 240000 \times 10$
Esercizio	E	240	Un esercizio è presente circa in 1000 schede di allenamento $240 = 240000 / 1000$
Gener-1	R	120	1 esercizio su 2 è aerobico $120 = 240 / 2$
Aerobico	E	120	Cardinalità (1, 1) con Gener-1
Gener-2	R	120	1 esercizio su 2 è anaerobico $120 = 240 / 2$
Anaerobico	E	120	Cardinalità (1, 1) con Gener-2
Dotazione	R	45000	In media in una sala ci sono 30 attrezzi $45000 = 30 \times 1500$
Attrezzo	E	45000	Cardinalità (1, 1) con Dotazione
Istruzione Intensità	R	480	In media gli esercizi con intensità hanno due istruzioni $480 = 2 \times 240$
Configurazione Utilizzo	E	90000	Cardinalità (1, 1) con Regolazione
Regolazione	R	90000	In media un attrezzo ha 2 regolazioni $90000 = 45000 \times 2$
Configurazione Ripetizione	R	22500	1 su 4 configurazioni è per un esercizio con ripetizione $22500 = 90000 / 4$
Esecuzione	R	48000000	In media ogni cliente esegue 1 su 12 esercizi presenti e li esegue circa 30 volte $(240/12) \times 30 \times 80000$

Monitoraggio Smart	E	48000000	Cardinalità (1, 1) con Esecuzione
Attrezzo Utilizzato	R	48000000	In media per un esercizio serve un attrezzo
Configurazione Utente	R	45000	1 su 2 è una configurazione dell'utente $45000 = 90000 / 2$
Monitoraggio Ripetizione	R	12000000	1 su 4 è un monitoraggio per una ripetizione $1200000 = 48000000 / 4$
Dati Ripetizione	E	24000000	In media di un esercizio con ripetizioni l'utente svolge 2 ripetizioni
Corso Disponibile	R	320000	In media un contratto mette a disposizione 4 corsi $320000 = 80000 \times 4$
Corso Frequentato	R	40000	In media 1 su 4 utenti frequenta metà dei corsi disponibili $40000 = (80000 / 4) \times 2$
Accesso Piscina	R	40000	In media la metà dei contratti permette l'accesso in una piscina $40000 = 80000 / 2$
Corso In Piscina	R	250	1 su 4 Corsi si svolge in Piscina $250 = 1000 / 4$
Corso In Sala	R	750	3 su 4 dei Corsi si svolge nelle sale $750 = (1000 / 4) \times 3$
Istruttore Corso	R	1000	Ogni Corso ha un solo istruttore
Corso	E	1000	In un Centro ci sono in media 10 corsi $1000 = 10 \times 100$
Programma Corso	R	1000	Cardinalità (1, 1) con Corso
Calendario Corso	E	1000	Cardinalità (1, 1) con Programma Corso
Programmazione Centro	R	1000	Cardinalità (1, 1) con Calendario Corso

Tutor Scheda	R	3000	In media un tutor è associato a 800 schede di allenamento $3000 = 240000 / 800$
Medico Scheda	R	3000	In media un medico è associato a 400 schede di alimentazione $3000 = 240000 / 800$
Consulente Contratto	R	800	Un consulente sottoscrive in media 100 contratti
Profilo Online	R	80000	Cardinalità (1, 1) con Contratto
Utente	E	80000	Cardinalità (1, 1) con Profilo Online
Forum	R	160000	In media un utente pubblica 2 post $160000 = 80000 \times 2$
Post	E	160000	Cardinalità (1, 1) con Forum
Post Risposta	R	80000	1 su 2 riceve una risposta $80000 = 160000 / 2$
Giudizio	R	240000	Ogni post risposta riceve in media 3 giudizi $240000 = 80000 \times 3$
Voto	E	240000	Cardinalità (1, 1) con Giudizio
Link Post	R	160000	In media ogni post contiene un solo link
Link	E	160000	Cardinalità (1, 1) con Link Post
Pubblicazione	R	160000	In media un utente pubblica 2 post
Thread	E	20000	In ogni thread ci sono in media 8 post $20000 = 160000 / 8$
Interessi Utente	R	80000	In media un utente ha 1 interesse
Interesse	E	80	In media un interesse è comune a 1000 utenti $80 = 80000 / 1000$

Amicizia	R	8000000	In media un utente ha 100 amici $8000000 = 80000 \times 100$
Amico	E	8000000	Cardinalità (1, 1) con Amicizia
Appartenenza	R	6000000	In media 3 su 4 amici appartiene ad una cerchia $6000000 = (8000000 / 4) \times 3$
Cerchia	E	6000000	In media 3 su 4 amici appartiene ad una cerchia
Suggerimento Inserimento	R	1000000	Degli amici che non sono in una cerchia la metà viene suggerito in una cerchia $1000000 = (8000000 / 4) / 2$
Attività Social	R	8000000	In media ogni utente invia 100 richieste di amicizie $8000000 = 80000 \times 100$
Richiesta Amicizia	E	8000000	Cardinalità (1, 1) con utente
Lancio Sfida	R	4000	In media 1 su 20 degli utenti lancia una sfida $4000 = 80000 / 20$
Sfida	E	40000	Cardinalità (1, 1) con Lancio Sfida
Vincitore	R	40000	Una sfida ha sempre un vincitore
Partecipante	R	200000	In media ad una sfida partecipano 5 amici $200000 = 40000 \times 5$
Thread sfida	R	4000	1 su 5 Thread sono riferiti ad una sfida $4000 = 20000 / 5$
Scheda Alimentazione Sfida	R	40000	Una sfida ha in media una scheda di alimentazione
Scheda Allenamento Sfida	R	40000	Una sfida ha in media una scheda di allenamento

Turnazione Dipendente	R	14000	In media un dipendente ha 2 turni di lavoro $14000 = 7000 \times 2$
Turnazione Centro	R	7000	In ogni centro un dipendente ha un orario di lavoro $7000 = 14000 / 2$
Orario di Lavoro	E	14000	Cardinalità (1, 1) con Turnazione Dipendente

4.3 Tavole degli accessi

Passiamo all'analisi delle operazioni descritte.

Si ipotizza che un accesso abbia un peso di un kilobyte e che una scrittura valga 2 kilobyte

Operazione 1

Descrizione: Modificare lo stato di una rata non ancora pagata di un contratto rateizzato

Input: Supponiamo noto il Codice del contratto in questione

Output: L'operazione modifica lo stato di una rata del contratto

Porzione E-R interessata: entità Contratto, associazione Pagamento, entità Piano Pagamento, associazione Rateizzato. Entità Rata.

Porzione tavola dei volumi:

Nome	E/R	Numero Istanze	Motivazione
Contratto	E	80000	Cardinalità (1, 1) con Intestatario
Piano Pagamento	E	80000	Cardinalità (1, 1) con Pagamento
Rateizzato	R	40000	In media 1 su 2 è rateizzato $40000 = 80000 / 2$
Rata	E	240000	In media ogni pagamento rateizzato ha 6 rate $240000 = 6 \times 40000$

Tavola degli accessi:

Numero Operazioni Elementari	Tipo Operazioni	Tipo Costrutto	Nome Costrutto	Motivazione
1	L	E	Contratto	Effettuo una lettura per trovare il contratto in questione
1	L	R	Pagamento	Effettuo lettura su pagamento
1	L	E	Piano Pagamento	Effettuo lettura su Piano Pagamento
1	L	R	Rateizzato	Effettuo lettura in rateizzato
6	L	E	Rata	Effettuo 6 letture per trovare la rata in quanto ogni pagamento rateizzato ha 6 rate
$1 \times 2 = 2$	S	E	Rata	Scrivo il nuovo stato della rata trovata

Totale operazioni elementari per singola operazione: 12

Totale operazioni elementari in un giorno $12 \times 20 = 240$

Operazione 2

Descrizione: Inserire una nuova scheda di Allenamento di un cliente

Input: Supponiamo noto il CodFiscale del Cliente

Output: L'operazione Inserisce la nuova scheda di allenamento del cliente

Porzione E-R interessata: entità Cliente, associazione Allenamento Attuale, entità Scheda Allenamento, associazione Allenamento Passato.

Porzione tavola dei volumi:

Nome	E/R	Numero Istanze	Motivazione
Cliente	E	80000	Ogni centro fitness ha in media 800 clienti $80000 = 100 \times 800$
Allenamento Passato	R	160000	In media un cliente ha 2 schede di allenamento passate $160000 = 80000 \times 2$
Allenamento Attuale	R	80000	Cardinalità (1, 1) con Cliente
Scheda Allenamento	E	240000	$240000 =$ Allenamento Passato (160000) + Allenamento Attuale (80000)

Tavola degli accessi:

Numero Operazioni Elementari	Tipo Operazioni	Tipo Costrutto	Nome Costrutto	Motivazione
1	L	E	Cliente	Effettuo una lettura per trovare il cliente in questione
1	L	R	Allenamento Attuale	Effettuo lettura su allenamento attuale
1	L	E	Scheda Allenamento	Effettuo lettura su scheda allenamento
$1 \times 2 = 2$	S	R	Scheda Passato	Effettuo scrittura della ormai vecchia scheda di allenamento
1	L	E	Cliente	Effettuo lettura su cliente
1×2	S	R	Allenamento Attuale	Scrivo la nuova scheda da inserire

Totale operazioni elementari per singola operazione: 7

Totale operazioni elementari in un giorno $7 \times 1000 = 7000$

Operazione 3

Descrizione: Inserire Contratto con Scopo Potenziamento

Input: Supponiamo noto il codice le informazioni relative al contratto

Output: L'operazione inserisce un nuovo contratto con scopo potenziamento e le relative informazioni

Porzione E-R interessata: entità Contratto, associazione Potenziamento, entità Fasci Muscolari.

Porzione tavola dei volumi:

Nome	E/R	Numero Istanze	Motivazione
Contratto	E	80000	Cardinalità (1, 1) con Intestatorio
Potenziamento	R	40000	In media 1 su 4 contratti comprende lo scopo potenziamento e allena 2 fasci muscolari $20000 = (80000 / 4) \times 2$
Fasci Muscolari	E	5	Fasci Muscolari che si possono allenare sono 5

Tavola degli accessi:

Numero Operazioni Elementari	Tipo Operazioni	Tipo Costrutto	Nome Costrutto	Motivazione
$1 \times 2 = 2$	S	E	Contratto	Effettuo scrittura con relative informazioni
$1 \times 2 = 2$	S	R	Potenziamento	Effettuo i scrittura in potenziamento
2	L	E	Fasci Muscolari	Effettuo In media un cliente allena 2 fasci muscolari

Totale operazioni elementari per singola operazione: 6

Totale operazioni elementari in un giorno $6 \times 50 = 350$

Operazione 4

Descrizione: Inserimento di una nuova visita di un cliente

Input: Supponiamo noto il codice del cliente

Output: L'operazione inserisce la visita del relativo cliente

Porzione E-R interessata: entità Cliente, associazione Visita, entità Misurazioni Impedenziometriche.

Porzione tavola dei volumi:

Nome	E/R	Numero Istanze	Motivazione
Visita	R	560000	In media ad ogni cliente vengo fatte 7 visite $560000 = 80000 \times 7$
Misurazioni Impedenziometriche	E	560000	Cardinalità (1, 1) con Visita
Cliente	E	80000	Ogni centro fitness ha in media 800 clienti $80000 = 100 \times 800$

Tavola degli accessi:

Numero Operazioni Elementari	Tipo Operazioni	Tipo Costrutto	Nome Costrutto	Motivazione
1	L	E	Cliente	Effettuo una lettura per trovare il cliente in questione
$1 \times 2 = 2$	S	R	Visita	Effettuo scrittura in visita
$1 \times 2 = 2$	S	E	Misurazioni Impedenziometriche	Effettuo 1 scrittura per inserire i relativi dati

Totale operazioni elementari per singola operazione: 5

Totale operazioni elementari in un giorno $5 \times 20 = 100$

Operazione 5

Descrizione: Trovare un Attrezzo che ha bisogno di manutenzione dopo inserimento del monitoraggio di un esercizio

Input: Supponiamo noto il codice del Monitoraggio

Output: L'operazione restituisce l'usura dell'attrezzo

Porzione E-R interessata: entità Attrezzo, associazione Attrezzo Utilizzato, entità Monitoraggio Smart

Porzione tavola dei volumi:

Nome	E/R	Numero Istanze	Motivazione
Attrezzo	E	45000	In media in una sala ci sono 30 attrezzi $45000 = 30 \times 1500$
Attrezzo Utilizzato	R	48000000	In media per un esercizio serve un attrezzo
Monitoraggio Smart	E	48000000	Cardinalità (1, 1) con Esecuzione

Tavola degli accessi:

Senza Ridondanza 6

Numero Operazioni Elementari	Tipo Operazioni	Tipo Costrutto	Nome Costrutto	Motivazione
$1 \times 2 = 2$	S	E	Monitoraggio Smart	Effettuo una scrittura
1	S	R	Attrezzo utilizzato	Effettuo una scrittura in attrezzo utilizzato
1000	L	R	Attrezzo utilizzato	Effettuo 1000 letture in quanto un attrezzo viene utilizzato 1000 volte prima della manutenzione

Totale operazioni elementari per singola operazione: 1004

Totale operazioni elementari in un giorno $1004 \times 100 = 100400$

Con Ridondanza 6

Numero Operazioni Elementari	Tipo Operazioni	Tipo Costrutto	Nome Costrutto	Motivazione
$1 \times 2 = 2$	S	E	Monitoraggio Smart	Effettuo una scrittura
$1 \times 2 = 2$	S	R	Attrezzo utilizzato	Effettuo una scrittura in attrezzo utilizzato
1	L	E	Attrezzo	Effettuo lettura in attrezzo
$1 \times 2 = 2$	S	E	Attrezzo	Aggiorno il Livello di usura
1	L	E	Attrezzo	Leggo il livello di usura

Totale operazioni elementari per singola operazione: 8

Totale operazioni elementari in un giorno $7 \times 100 = 800$

Operazione 6

Descrizione: Trovare i Post in un determinato Thread

Input: Supponiamo noto il codice del thread cercato

Output: L'operazione restituisce i post del thread in questione

Porzione E-R interessata: entità Thread, associazione Pubblicazione, entità Post.

Porzione tavola dei volumi:

Nome	E/R	Numero Istanze	Motivazione
Post	E	160000	Cardinalità (1, 1) con Forum
Pubblicazione	R	160000	In media un utente pubblica 2 post
Thread	E	20000	In ogni thread ci sono in media 8 post $20000 = 160000 / 8$

Tavola degli accessi:

Numero Operazioni Elementari	Tipo Operazioni	Tipo Costrutto	Nome Costrutto	Motivazione
1	L	E	Thread	Effettuo una lettura per trovare il thread in questione
8	L	R	Pubblicazione	Effettuo 8 letture perché in un thread si trovano in media 8 post
8	L	E	Post	Effettuo 8 letture per restituire le informazioni sui post cercati

Totale operazioni elementari per singola operazione: 17

Totale operazioni elementari in un giorno $17 \times 100 = 1500$

Operazione 7

Descrizione: Trovare quante ripetizioni il cliente svolge per un determinato esercizio

Input: Supponiamo noto il CodFiscale del Cliente

Output: L'operazione restituisce quante ripetizione un cliente ha fatto per un esercizio

Porzione E-R interessata: entità Cliente, associazione Allenamento Attuale, entità Scheda Allenamento, associazione Composizione Scheda, entità Esercizio, associazione Esecuzione, entità Monitoraggio Smart, associazione Monitoraggio Ripetizione

Porzione tavola dei volumi:

Nome	E/R	Numero Istanze	Motivazione
Cliente	E	80000	Ogni centro fitness ha in media 800 clienti $80000 = 100 \times 800$
Allenamento Attuale	R	80000	Cardinalità (1, 1) con Cliente
Scheda Allenamento	E	240000	$240000 =$ Allenamento Passato (160000) + Allenamento Attuale (80000)
Composizione Scheda	R	2400000	In media ogni scheda contiene 10 Esercizi $2400000 = 240000 \times 10$
Esercizio	E	240	Un esercizio è presente circa in 1000 schede di allenamento $240 = 240000 / 1000$
Esecuzione	R	48000000	In media ogni cliente esegue 1 su 12 esercizi presenti e li esegue circa 30 volte $(240/12) \times 30 \times 80000$
Monitoraggio Smart	E	48000000	Cardinalità (1, 1) con Esecuzione
Monitoraggio Ripetizione	R	12000000	1 su 4 è un monitoraggio per una ripetizione $1200000 = 48000000 / 4$

Tavola degli accessi:

Numero Operazioni Elementari	Tipo Operazioni	Tipo Costrutto	Nome Costrutto	Motivazione
1	L	E	Cliente	Effettuo una lettura per trovare il cliente in questione
1	L	R	Allenamento Attuale	Effettuo lettura su allenamento attuale
1	L	E	Scheda Allenamento	Effettuo lettura su scheda allenamento
10	L	R	Composizione scheda	In un scheda ci sono in media 10 esercizi
1	L	E	Esercizio	Leggo esercizio che mi interessa
1	L	R	Esecuzione	Leggo l'esecuzione dell'esercizio
1	L	E	Monitoraggio Smart	Leggo il monitoraggio relativo all'esercizio
2	L	R	Monitoraggio Ripetizione	In media un cliente fa 2 ripetizioni di un esercizio

Totale operazioni elementari per singola operazione:18

Totale operazioni elementari in un giorno $18 \times 300 = 5400$

Operazione 8

Descrizione: Trovare Posti liberi in uno spogliatoio

Input: Inserimento di un accesso

Output: Trovo il numero di posti liberi in uno spogliatoio

Porzione E-R interessata: entità Accesso, associazione Assegnazione, entità Armadietto, associazione Collocazione, entità Spogliatoio

Porzione tavola dei volumi:

Nome	E/R	Numero Istanze	Motivazione
Accesso	E	115200	Cardinalità (1, 1) con Effettuato
Frequenza	E	115200	Cardinalità (1, 1) con Accesso
Spogliatoio	E	400	In media in un centro ci sono 4 spogliatoi $400 = 4 \times 100$
Assegnazione	R	115200	Cardinalità (1, 1) con Acceso
Collocazione	R	12000	In uno spogliatoio sono collocati 30 armadietti $12000 = 30 \times 400$
Armadietto	E	12000	Cardinalità (1, 1) con Collocazione

Tavola degli accessi:

Senza Ridondanza 7:

Numero Operazioni Elementari	Tipo Operazioni	Tipo Costrutto	Nome Costrutto	Motivazione
$1 \times 2 = 2$	S	E	Accesso	Effettuo una scrittura in accesso

1	L	R	Assegnazione	Leggo l'armadietto associato all'accesso
1	L	E	Armadietto	Leggo armadietto associato all'accesso
30	L		Collocazione	In media in uno spogliatoio ci 30 armadietti quindi devo leggerli tutti per capire quale è libero o no
1	L		Spogliatoio	Leggo in spogliatoio

Totale operazioni elementari per singola operazione: 35

Totale operazioni elementari in un giorno $35 \times 1000 = 35000$

Con Ridondanza 7:

Numero Operazioni Elementari	Tipo Operazioni	Tipo Costrutto	Nome Costrutto	Motivazione
$1 \times 2 = 2$	S	E	Accesso	Effettuo una scrittura in accesso
1	L	R	Assegnazione	Leggo l'armadietto associato all'accesso
1	L	E	Armadietto	Leggo armadietto associato all'accesso
1	L	R	Collocazione	In media in uno spogliatoio ci 30 armadietti quindi devo leggerli tutti per capire quale è libero o no
1	L	E	Spogliatoio	Leggo in spogliatoio
$1 \times 2 = 2$	S	E	Spogliatoio	Modifico il numero di posti disponibili

Totale operazioni elementari per singola operazione: 8

Totale operazioni elementari in un giorno $8 \times 1000 = 8000$

4.4 Riassunto degli accessi

Vengono di seguito riportati i benefici derivanti dall'uso delle due ridondanze.

Ridondanza 7

Senza ridondanza	Con ridondanza
Operazione 8: 35000 operazioni al giorno	Operazione 8: 8000 operazioni al giorno

Quindi la ridondanza appena considerata contribuisce a ridurre i tempi di esecuzione delle operazioni.

Ridondanza 8

Senza ridondanza	Con ridondanza
Operazione 5: 100400 operazioni al giorno	Operazione 5: 800 operazioni al giorno

Quindi anche questa ridondanza contribuisce a ridurre i tempi di esecuzione delle operazioni

Le altre ridondanze sono state Eliminate dal Diagramma ER in fase di Ristrutturazione per migliorare le Prestazioni del Database evitando l'aggiornamento di Informazioni Ridondanti.

5 Progettazione Logica

5.1 Schema Logico

- CENTRO FITNESS (IDCentro, Dimensione, Telefono, CapienzaMassima, Città, Via, NumCivico, C.A.P, Direttore)
- CALENADARIO
CENTRO FITNESS (IDCentro, NomeGiorno, OrarioApertura, OrarioChiusura)
- SALA (IDSala, IDCentro, Nome, TipologiaContratto, ResponsabileSala)
- ATTREZZO (IDAttrezzo, NomeAttrezzo IDSala, Tipologia, ConsumoEnergetico, LivelloUsura)
- DIPENDENTE (CodFiscaleDip, Nome, Cognome, DataNascita, Città, C.A.P, Via, NumCivico, Sesso, IDDocumentoRiconoscimento, Prefettura, Responsabile)
- ORARIO DI LAVORO (IDCentro, CodFiscaleDip, NomeGiornoLavoro, Attività OrarioInizio, OrarioFine)
- CONTRATTO (IDContratto, IDCentro, Cliente, Consulente, TipologiaContratto, AccessiSettimanali, CostoMensile, DataIscrizione, DurataMensile, Scopo)
- CLIENTE (CodFiscaleCliente, Nome, Cognome, DataNascita, Città, C.A.P, Via, NumCivico, Sesso, IDDocumentoRiconoscimento, Prefettura)
- POTENZIAMENTO (IDContratto, NomeFasciMuscolari, LivelloPotenziamento)
- FASCI MUSCOLARI (NomeFasciMuscolari, LivelloPotenziamento)
- PIANO PAGAMENTO (IDPagamento, IDContratto, TipologiaPagamento)
- RATA (IDRata, IDPagamento, ImportoRata, IstitutoFinanziario, TassoInteresse, DataScadenza, Stato)

- ACCESSO (**IDAccesso**, IDCentro, CodFiscaleCliente, IDArmadietto, DataAccesso, OrarioEntrata, OrarioUscita)
- ARMADIETTO (**IDArmadietto**, IDSpogliatoio, CombinazioneNumerica)
- SPOGLIATOIO (**IDSpogliatoio**, Capienza, PostiDisponibili, Posizione, Sesso)
- MISURAZIONI
IMPEDENZIOMETRICHE (**IDMisurazione**, CodFiscaleCliente, DataMisurazione, AcquaTotale, MassaMagra, Peso, Altezza, MassaGrassa, StatoCiente, Entità)
- SCHEDA ALIMENTAZIONE (**IDSchedaAlimentazione**, CodFiscaleCliente, IDDieta, DataInizio, DataFine, MedicoScheda, Obiettivo)
- ALIMENTAZIONE PASSATA (**CodFiscaleCliente**, **IDSchedaAlimentazione**)
- DIETA (**IDDieta**, ApportoCalorico, NumeroPasti)
- COMPOSIZIONE DIETA (**IDDieta**, **NomePietanza**, Pasto)
- PIETANZA (**NomePietanza**, ComposizionePietanza)
- SCHEDA ALLENAMENTO (**IDSchedaAlimentazione**, CodFiscaleCliente, TutorScheda, DataInizio, DataFine)
- ALLENAMENTO PASSATO (**IDSchedaAlimentazione**, **CodFiscaleCliente**)
- COMPOSIZIONE ALLENAMENTO (**IDSchedaAllenamento**, **IDEsercizio**)
- ESERCIZIO (**IDEsercizio**, Nome, DispendioEnergeticoMedio)
- AEROBICO (**IDEsercizio**, DurataMinuti)
- ANAEROBICO (**IDEsercizio**, **Nome**, NumEsecuzioni, TempoRecupero)

- CONFIGURAZIONE UTILIZZO (**IDConfigurazione**, IDAttrezzo, IDEsercizio, NomeRegolazione, Intensità, IstruzioneEsatta)
- MONITORAGGIO SMART (**IDMonitoraggio**, IDSchedaAllenamento, IDEsercizio, IstanteInizio, IstanteFine, DataMonitoraggio)
- MONITORAGGIO RIPETIZIONE (**IDMonitoraggio, IDRipetizione**)
- DATI RIPETIZIONE (**IDRipetizione**, NomeRipetizione, NumeroEsecuzioni, TempoRecupero)
- CONFIGURAZIONE CLIENTE (**IDMonitoraggio, IDConfigurazione**)
- CONFIGURAZIONE RIPETIZIONE (**EsercizioAnaerobico, IDConfigurazione, Nome**)
- ATTREZZO UTILIZZATO (**IDMonitoraggio, IDAttrezzo**)
- PISCINA (**IDPiscina**, IDCentro, TipologiaContratto, Dimensione, Locazione)
- ACCESSO PISCINA (**IDPiscina, IDContratto**, NumeroAccessi)
- MULTISEDE (**IDContratto, IDCentro**, FasciaOraria, IngressiMax)
- CORSO FREQUENTATO (**IDCorso, CodFiscaleCliente**)
- CORSO DISPONIBILE (**IDCorso, IDContratto**)
- CORSO IN PISCINA (**IDCorso, IDPiscina**)
- CORSO IN SALA (**IDCorso, IDSala**)
- CORSO (**IDCorso**, NomeDisciplina, IstruttoreCorso, DataInizio, DataFine, NumeroMaxPartecipanti, Livello)
- CALENDARIO CORSO (**IDCalendarioCorso**, IDCorso, IDCentro, GiornoSettimana, OrarioInizio, OrarioFine)
- UTENTE (**Username**, IDContratto, Password, Credibilità)

- POST (**IDPost**, Testo, Timestamp, Risposta, Usermane, AreaForum, ThreadPost)
- THREAD (**IDThread**, Titolo)
- LINK (**URL**)
- LINK POST (**URL, IDPost**)
- VOTO (**IDVoto**, Risposta, Stelle)
- INTERESSI UTENTE (**Username, NomeInteresse**)
- INTERESSE (**NomeInteresse**)
- RICHIESTA AMICIZIA (**IDRichiesta**, Username, UsernameDestinatario, StatoRichiesta)
- AMICO (**UsernameAmico, Username**)
- APPARTENENZA (**NomeCerchia, UsernameAmico, Username**)
- CERCHIA (**NomeCerchia**, Argomento)
- SUGGERIMENTO INSERIMENTO (**NomeCerchia, UsernameAmico, Username**)
- PARTECIPANTE (**UsernameAmico, Proponente, DataInizioSfida**)
- SFIDA (**Proponente, DataInizioSfida**, DataLancio, DataScadenza, Scopo, ThreadSfida, AllenamentoSfida, AlimentazioneSfida, Vincitore)
- CLIENTE OCCASIONALE (**CodFiscaleCliente**, TipologiaTariffa, TariffaAccesso)

5.2 Vincoli di integrità referenziale

- Esiste un vincolo di integrità referenziale tra l'attributo "Direttore" della tabella *Centro Fitness* e l'attributo "CodFiscaleDip" della tabella *Dipendente*.
- Esiste un vincolo di integrità referenziale tra l'attributo "ResponsabileSala" della tabella *Sala* e l'attributo "CodFsicaleDip" della tabella *Dipendente*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDSala" della tabella *Sala* e l'attributo "IDSala" della tabella *Attrezzo*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDCentro" della tabella *Centro Fitness* e l'attributo "IDCentro" della tabella *Sala*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDCentro" della tabella *Centro Fitness* e l'attributo "IDCentro" della tabella *Calendario Centro Fitness*.
- Esiste un vincolo di integrità referenziale tra l'attributo "CodFiscaleDip" della tabella *Dipendente* e l'attributo "Responsabile" della tabella *Dipendente*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDCentro" della tabella *Centro Fitness* e l'attributo "IDCentro" della tabella *Contratto*.
- Esiste un vincolo di integrità referenziale tra l'attributo "CodFiscaleDip" della tabella *Dipendente* e l'attributo "Consulente" della tabella *Contratto*.
- Esiste un vincolo di integrità referenziale tra l'attributo "Cliente" della tabella *Contratto* e l'attributo "CodFiscaleCliente" della tabella *Cliente*.
- Esiste un vincolo di integrità referenziale tra l'attributo "CodFiscaleDip" della tabella *Dipendente* e l'attributo "Tutor" della tabella *Cliente*.
- Esiste un vincolo di integrità referenziale tra l'attributo "CodFiscaleDip" della tabella *Dipendente* e l'attributo "MedicoNutrizionista" della tabella *Cliente*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDContratto" della tabella *Contratto* e l'attributo "IDContratto" della tabella *Potenziamento*.

- Esiste un vincolo di integrità referenziale tra l'attributo "NomeFasciMuscolari" della tabella *Potenziamento* e l'attributo "NomeFasciMuscolari" della tabella *Fasci Muscolari*.
- Esiste un vincolo di integrità referenziale tra l'attributo "LivelloPotenziamento" della tabella *Potenziamento* e l'attributo "LivelloPotenziamento" della tabella *Fasci Muscolari*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDContratto" della tabella *Contratto* e l'attributo "IDContratto" della tabella *Piano Pagamento*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDPagamento" della tabella *Piano Pagamento* e l'attributo "IDPagamento" della tabella *Rata*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDCentro" della tabella *Accesso* e l'attributo "IDCentro" della tabella *Centro Fitness*.
- Esiste un vincolo di integrità referenziale tra l'attributo "CodFiscaleCliente" della tabella *Cliente* e l'attributo "CodFiscaleCliente" della tabella *Accesso*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDArmadietto" della tabella *Accesso* e l'attributo "IDArmadietto" della tabella *Armadietto*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDSpogliatoio" della tabella *Armadietto* e l'attributo "IDSpogliatoio" della tabella *Spogliatoio*.
- Esiste un vincolo di integrità referenziale tra l'attributo "CodFiscaleCliente" della tabella *Misurazioni Impedenziometriche* e l'attributo "CodFiscaleCliente" della tabella *Cliente*.
- Esiste un vincolo di integrità referenziale tra l'attributo "CodFiscaleCliente" della tabella *Cliente* e l'attributo "CodFiscaleCliente" della tabella *Scheda Alimentazione*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDDieta" della tabella *Dieta* e l'attributo "IDDieta" della tabella *SchedaAlimentazione*.

- Esiste un vincolo di integrità referenziale tra l'attributo "CodFiscaleDip" della tabella *Dipendente* e l'attributo "MedicoScheda" della tabella *Scheda Alimentazione*.
- Esiste un vincolo di integrità referenziale tra l'attributo "CodFiscaleCliente" della tabella *Cliente* e l'attributo "CodFiscaleCliente" della tabella *Alimentazione Passata*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDSchedaAlimentazione" della tabella *Alimentazione Passata* e l'attributo "IDSchedaAlimentazione" della tabella *Scheda Alimentazione*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDDieta" della tabella *Dieta* e l'attributo "IDDieta" della tabella *Composizione Dieta*.
- Esiste un vincolo di integrità referenziale tra l'attributo "NomePietanza" della tabella *Pietanza* e l'attributo "NomePietanza" della tabella *Composizione Dieta*.
- Esiste un vincolo di integrità referenziale tra l'attributo "CodFiscaleCliente" della tabella *Cliente* e l'attributo "CodFiscaleCliente" della tabella *Scheda Allenamento*.
- Esiste un vincolo di integrità referenziale tra l'attributo "CodFiscaleDip" della tabella *Dipendente* e l'attributo "TutorScheda" della tabella *Scheda Allenamento*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDSchedaAlimentazione" della tabella *Scheda Allenamento* e l'attributo "IDSchedaAlimentazione" della tabella *Scheda Allenamento Passato*.
- Esiste un vincolo di integrità referenziale tra l'attributo "CodFiscaleCliente" della tabella *Cliente* e l'attributo "CodFiscaleCliente" della tabella *Scheda Allenamento Passata*.

- Esiste un vincolo di integrità referenziale tra l'attributo "IDSchedaAllenamento" della tabella *Scheda Allenamento* e l'attributo "IDSchedaAllenamento" della tabella *Composizione Allenamento*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDEsercizio" della tabella *Esercizio* e l'attributo "IDEsercizio" della tabella *Composizione Allenamento*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDEsercizio" della tabella *Esercizio* e l'attributo "IDEsercizio" della tabella *Aerobico*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDEsercizio" della tabella *Esercizio* e l'attributo "IDEsercizio" della tabella *Anaerobico*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDEsercizio" della tabella *Esercizio* e l'attributo "IDEsercizio" della tabella *Configurazione Utilizzo*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDAttrezzo" della tabella *Attrezzo* e l'attributo "IDAttrezzo" della tabella *Configurazione Utilizzo*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDEsercizio" della tabella *Esercizio* e l'attributo "IDEsercizio" della tabella *Monitoraggio Smart*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDSchedaAllenamento" della tabella *Scheda Allenamento* e l'attributo "IDSchedaAllenamento" della tabella *Monitoraggio Smart*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDMonitoraggio" della tabella *Monitoraggio Smart* e l'attributo "IDMonitoraggio" della tabella *Monitoraggio Ripetizione*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDRipetizione" della tabella *Dati Ripetizione* e l'attributo "IDRipetizione" della tabella *Monitoraggio Ripetizione*.

- Esiste un vincolo di integrità referenziale tra l'attributo "IDMonitoraggio" della tabella *Monitoraggio Smart* e l'attributo "IDMonitoraggio" della tabella *Configurazione Cliente*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDConfigurazione" della tabella *Configurazione Utilizzo* e l'attributo "IDConfigurazione" della tabella *Configurazione Cliente*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDEsercizio" della tabella *Anaerobico* e l'attributo "IDEsercizio" della tabella *Configurazione Ripetizione*.
- Esiste un vincolo di integrità referenziale tra l'attributo "Nome" della tabella *Anaerobico* e l'attributo "Nome" della tabella *Configurazione Ripetizione*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDConfigurazione" della tabella *Configurazione Utilizzo* e l'attributo "IDConfigurazione" della tabella *Configurazione Ripetizione*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDCentro" della tabella *Piscina* e l'attributo "IDCentro" della tabella *Centro Fitness*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDPiscina" della tabella *Piscina* e l'attributo "IDPiscina" della tabella *Accesso Piscina*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDContratto" della tabella *Contratto* e l'attributo "IDContratto" della tabella *Accesso Piscina*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDContratto" della tabella *Contratto* e l'attributo "IDContratto" della tabella *Multisede*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDCentro" della tabella *Centro Fitness* e l'attributo "IDCentro" della tabella *Multisede*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDCorso" della tabella *Corso* e l'attributo "IDCorso" della tabella *Corso Frequentato*.

- Esiste un vincolo di integrità referenziale tra l'attributo "CodFiscaleCliente" della tabella *Cliente* e l'attributo "CodFiscaleCliente" della tabella *Corso Frequentato*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDCorso" della tabella *Corso* e l'attributo "IDCorso" della tabella *Corso Disponibile*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDContratto" della tabella *Corso* e l'attributo "IDContratto" della tabella *Contratto*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDCorso" della tabella *Corso* e l'attributo "IDCorso" della tabella *Corso In Piscina*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDCorso" della tabella *Corso* e l'attributo "IDCorso" della tabella *Corso In Sala*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDPiscina" della tabella *Corso In Piscina* e l'attributo "IDPiscina" della tabella *Piscina*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDSala" della tabella *Corso In Sala* e l'attributo "IDSala" della tabella *Sala*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IstruttoreCorso" della tabella *Corso* e l'attributo "CodFiscaleDip" della tabella *Dipendente*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDCorso" della tabella *Corso* e l'attributo "IDCorso" della tabella *Calendario Corso*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDCentro" della tabella *Centro Fitness* e l'attributo "IDCentro" della tabella *Calendario Corso*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDContratto" della tabella *Contratto* e l'attributo "IDContratto" della tabella *Utente*.
- Esiste un vincolo di integrità referenziale tra l'attributo "Username" della tabella *Post* e l'attributo "Username" della tabella *Utente*.

- Esiste un vincolo di integrità referenziale tra l'attributo "Risposta" della tabella *Post* e l'attributo "IDPost" della tabella *Post*.
- Esiste un vincolo di integrità referenziale tra l'attributo "ThreadPost" della tabella *Post* e l'attributo "IDThread" della tabella *Thread*.
- Esiste un vincolo di integrità referenziale tra l'attributo "URL" della tabella *Link* e l'attributo "URL" della tabella *Link Post*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDPost" della tabella *Post* e l'attributo "IDPost" della tabella *Link Post*.
- Esiste un vincolo di integrità referenziale tra l'attributo "Risposta" della tabella *Post* e l'attributo "Risposta" della tabella *Voto*.
- Esiste un vincolo di integrità referenziale tra l'attributo "NomeInteresse" della tabella *Interesse* e l'attributo "NomeInteresse" della tabella *Interesse Utente*.
- Esiste un vincolo di integrità referenziale tra l'attributo "Username" della tabella *Utente* e l'attributo "Username" della tabella *Interesse Utente*.
- Esiste un vincolo di integrità referenziale tra l'attributo "Username" della tabella *Utente* e l'attributo "Username" della tabella *Richiesta Amicizia*.
- Esiste un vincolo di integrità referenziale tra l'attributo "Username" della tabella *Amico* e l'attributo "Username" della tabella *Utente*.
- Esiste un vincolo di integrità referenziale tra l'attributo "NomeCerchia" della tabella *Cerchia* e l'attributo "NomeCerchia" della tabella *Appartenenza*.
- Esiste un vincolo di integrità referenziale tra l'attributo "UsernameAmico" della tabella *Amico* e l'attributo "UsernameAmico" della tabella *Appartenenza*.
- Esiste un vincolo di integrità referenziale tra l'attributo "NomeCerchia" della tabella *Cerchia* e l'attributo "NomeCerchia" della tabella *Suggerimento Inserimento*.

- Esiste un vincolo di integrità referenziale tra l'attributo "UsernameAmico" della tabella *Amico* e l'attributo "UsernameAmico" della tabella *Suggerimento Inserimento*.
- Esiste un vincolo di integrità referenziale tra l'attributo "UsernameAmico" della tabella *Amico* e l'attributo "UsernameAmico" della tabella *Partecipante*.
- Esiste un vincolo di integrità referenziale tra l'attributo "Proponente" della tabella *Sfida* e l'attributo "Proponente" della tabella *Partecipante*.
- Esiste un vincolo di integrità referenziale tra l'attributo "Username" della tabella *Utente* e l'attributo "Proponente" della tabella *Sfida*.
- Esiste un vincolo di integrità referenziale tra l'attributo "DataInizioSfida" della tabella *Sfida* e l'attributo "DataInizioSfida" della tabella *Partecipante*.
- Esiste un vincolo di integrità referenziale tra l'attributo "ThreadSfida" della tabella *Sfida* e l'attributo "IDThread" della tabella *Thread*.
- Esiste un vincolo di integrità referenziale tra l'attributo "AllenamentoSfida" della tabella *Sfida* e l'attributo "IDSchedaAllenamento" della tabella *Scheda Allenamento*.
- Esiste un vincolo di integrità referenziale tra l'attributo "AlimentazioSfida" della tabella *Sfida* e l'attributo "IDSchedaAlimentazione" della tabella *Scheda Alimentazione*.
- Esiste un vincolo di integrità referenziale tra l'attributo "CodFiscaleCliente" della tabella *Cliente Occasionale* e l'attributo "CodFiscaleCliente" della tabella *Cliente*.
- Esiste un vincolo di integrità referenziale tra l'attributo "Username" della tabella *Utente* e l'attributo "Vincitore" della tabella *Sfida*.
- Esiste un vincolo di integrità referenziale tra l'attributo "IDMonitoraggio" della tabella *Monitoraggio Smart* e l'attributo "IDMonitoraggio" della tabella *Attrezzo Utilizzato*.

- Esiste un vincolo di integrità referenziale tra l'attributo "IDAttrezzo" della tabella *Attrezzo* e l'attributo "IDAttrezzo" della tabella *Attrezzo Utilizzato*.

6 Dipendenze Funzionali e Normalizzazione

6.1 Analisi delle Dipendenze Funzionali

Tabella CENTRO FITNESS:

- IDCentro → Dimensione, Telefono, CapienzaMassima, Città, Via, NumCivico, C.A.P, Direttore

Poiché la parte sinistra è superchiave, CENTRO FITNESS è in BCNF.

Tabella CALENADARIO CENTRO FITNESS:

- IDCentro, NomeGiorno → OrarioApertura, OrarioChiusura

Poiché la parte sinistra è superchiave, CALENDARIO CENTRO FITNESS è in BCNF.

Tabella SALA:

- IDSala → IDCentro, Nome, TipologiaContratto, ResponsabileSala

Poiché la parte sinistra è superchiave, SALA è in BCNF.

Tabella ATTREZZO:

- IDAttrezzo → IDSala, NomeAttrezzo, Tipologia, ConsumoEnergetico, LivelloUsura

Poiché la parte sinistra è superchiave, ATTREZZO è in BCNF.

Tabella DIPENDENTE:

- CodFiscaleDip → Nome, Cognome, DataNascita, Città, C.A.P, Via, NumCivico, Sesso, IDDocumentoRiconoscimento, Prefettura, Responsabile

Poiché la parte sinistra è superchiave, DIPENDENTE è in BCNF.

Tabella ATTREZZO:

- IDAttrezzo → IDSala, Tipologia, ConsumoEnergetico, LivelloUsura

Poiché la parte sinistra è superchiave, ATTREZZO è in BCNF.

Tabella ORARIO DI LAVORO:

- Centro, CodFiscaleDip, NomeGiornoLavoro, Attività ,OrarioInizio, → OrarioFine

Poiché la parte sinistra è superchiave, ORARIO DI LAVORO è in BCNF.

Tabella CONTRATTO:

- IDContratto → IDCentro, Cliente, Consulente, TipologiaContratto, AccessiSettimanali, CostoMensile, DataIscrizione, Durata, Scopo

Poiché la parte sinistra è superchiave, CONTRATTO è in BCNF.

Tabella CLIENTE:

- CodFiscaleCliente → Nome, Cognome, DataNascita, Città, C.A.P, Via, NumCivico, Sesso, IDDocumentoRiconoscimento, Prefettura, Tutor, MedicoNutrizionista

Poiché la parte sinistra è superchiave, CLIENTE è in BCNF.

Tabella POTENZIAMENTO:

Poiché non esistono dipendenze funzionali, POTENZIAMENTO è in BCNF

Tabella FASCI MUSCOLARI:

Poiché non esistono dipendenze funzionali, FASCI MUSCOLARI è in BCNF

Tabella PIANO PAGAMENTO:

- IDPagamento → IDContratto, TipologiaPagamento

Poiché la parte sinistra è superchiave, PIANO PAGAMENTO è in BCNF.

Tabella RATA:

- IDRata → IDPagamento, ImportoRata, IstitutoFinanziario, TassoInteresse, DataScadenza, Stato

Poiché la parte sinistra è superchiave, RATA è in BCNF.

Tabella ACCESSO:

- IDAccesso → IDCentro, CodFiscaleCliente, IDArmadietto, Data, OrarioEntrata, OrarioUscita

Poiché la parte sinistra è superchiave, ACCESSO è in BCNF.

Tabella ARMADIETTO:

- IDArmadietto → IDSpogliatoio, CombinazioneNumerica

Poiché la parte sinistra è superchiave, ARMADIETTO è in BCNF.

Tabella SPOGLIATOIO:

- IDSpogliatoio → Capienza, PostiDisponibili, Posizione, Sesso

Poiché la parte sinistra è superchiave, SPOGLIATOIO è in BCNF.

Tabella MISURAZIONI IMPEDENZIOMETRICHE:

- IDMisurazione → CodFiscaleCliente, Data, AcquaTotale, MassaMagra, Peso, Altezza, MassaGrassa, StatoCiente, Entità

Poiché la parte sinistra è superchiave, MISURAZIONI IMPEDENZIOMETRICHE è in BCNF.

Tabella SCHEDA ALIMENTAZIONE:

- IDSchedaAlimentazione → CodFiscaleCliente, IDDieta, DataInizio, DataFine, MedicoScheda, Obiettivo

Poiché la parte sinistra è superchiave, SCHEDA ALIMENTAZIONE è in BCNF.

Tabella ALIMENTAZIONE PASSATA:

Poiché non esistono dipendenze funzionali, ALIMENTAZIONE PASSATA è in BCNF.

Tabella DIETA:

- IDDieta → ApportoCalorico, NumeroPasti

Poiché la parte sinistra è superchiave, DIETA è in BCNF.

Tabella COMPOSIZIONE DIETA:

- IDDieta, NomePietanza → Pasto

Poiché la parte sinistra è superchiave, COMPOSIZIONE DIETA è in BCNF.

Tabella PIETANZA:

- NomePietanza → ComposizionePietanza

Poiché la parte sinistra è superchiave, PIETANZA è in BCNF.

Tabella SCHEDA ALLENAMENTO:

- IDSchedaAllenamento → CodFiscaleCliente, TutorScheda, DataInizio, DataFine

Poiché la parte sinistra è superchiave, SCHEDA ALLENAMENTO è in BCNF.

Tabella ALLENAMENTO PASSATO:

Poiché non esistono dipendenze funzionali, ALLENAMENTO PASSATO è in BCNF.

Tabella COMPOSIZIONE ALLENAMENTO:

Poiché non esistono dipendenze funzionali, COMPOSIZIONE ALLENAMENTO è in BCNF.

Tabella ESERCIZIO:

- IDEsercizio → Nome, DispendioEnergeticoMedio

Poiché la parte sinistra è superchiave, ESERCIZIO è in BCNF.

Tabella AEROBICO:

- IDEsercizio → Durata

Poiché la parte sinistra è superchiave, AEROBICO è in BCNF.

Tabella ANAEROBICO:

- IDEsercizio, Nome → NumEsecuzioni, TempoRecupero

Poiché la parte sinistra è superchiave, ANAEROBICO è in BCNF.

Tabella CONFIGURAZIONE UTILIZZO:

- IDConfigurazione → IDAttrezzo, IDEsercizio, NomeRegolazione, Intensità, IstruzioneEsatta

Poiché la parte sinistra è superchiave, CONFIGURAZIONE UTILIZZO è in BCNF.

Tabella MONITORAGGIO SMART:

- IDMonitoraggio → IDSchedaAlimentazione, IDEsercizio, IstanteInizio, IstanteFine, DataMonitoraggio

Poiché la parte sinistra è superchiave, MONITORAGGIO SMART è in BCNF.

Tabella MONITORAGGIO RIPETIZIONE:

Poiché non esistono dipendenze funzionali, MONITORAGGIO RIPETIZIONE è in BCNF.

Tabella DATI RIPETIZIONE:

- IDRipetizione → NomeRipetizione, NumeroEsecuzioni, TempoRecupero

Poiché la parte sinistra è superchiave, DATI RIPETIZIONE è in BCNF.

Tabella CONFIGURAZIONE CLIENTE:

Poiché non esistono dipendenze funzionali, CONFIGURAZIONE CLIENTE è in BCNF.

Tabella CONFIGURAZIONE RIPETIZIONE:

- IDEsercizio, IDConfigurazione → Nome, NumeroEsecuzioni

Poiché la parte sinistra è superchiave, CONFIGURAZIONE RIPETIZIONE è in BCNF.

Tabella ATTREZZO UTILIZZATO:

Poiché non esistono dipendenze funzionali, ATTREZZO UTILIZZATO è in BCNF

Tabella PISCINA:

- IDPiscina → IDCentro, TipologiaContratto, Dimensione, Locazione

Poiché la parte sinistra è superchiave, PISCINA è in BCNF.

Tabella ACCESSO PISCINA:

- IDPiscina, IDContratto → NumeroAccessi

Poiché la parte sinistra è superchiave, ACCESSO PISCINA è in BCNF.

Tabella MULTISEDE:

- IDContratto, IDEsercizio → FasciaOraria, IngressiMax

Poiché la parte sinistra è superchiave, MULTISEDE è in BCNF.

Tabella CORSO FREQUENTATO:

Poiché non esistono dipendenze funzionali, CORSO FREQUENTATO è in BCNF.

Tabella CORSO DISPONIBILE:

Poiché non esistono dipendenze funzionali, CORSO DISPONIBILE è in BCNF.

Tabella CORSO IN PISCINA:

Poiché non esistono dipendenze funzionali, CORSO IN PISCINA è in BCNF.

Tabella CORSO IN SALA:

Poiché non esistono dipendenze funzionali, CORSO IN SALA è in BCNF.

Tabella CORSO:

- IDCorso → NomeDisciplina, IstruttoreCorso, DataInizio, DataFine,
NumeroMaxPartecipanti, Livello

Poiché la parte sinistra è superchiave, CORSO è in BCNF.

Tabella CALENDARIO CORSO:

- IDCalendarioCorso → IDCorso, IDCentro, GiornoSettimana, OrarioInizio, OrarioFine

Poiché la parte sinistra è superchiave, CALENDARIO CORSO è in BCNF.

Tabella UTENTE:

- Username → IDContratto, Password, Credibilità

Poiché la parte sinistra è superchiave, UTENTE è in BCNF.

Tabella POST:

- IDPost → Testo, Timestamp, Risposta, Usermane, AreaForum, ThreadPost

Poiché la parte sinistra è superchiave, POST è in BCNF.

Tabella THREAD:

- IDThread → Titolo

Poiché la parte sinistra è superchiave, THREAD è in BCNF.

Tabella LINK:

Poiché non esistono dipendenze funzionali, LINK è in BCNF

.

Tabella AEROBICO:

Poiché non esistono dipendenze funzionali, LINK POST è in BCNF.

Tabella INTERESSI UTENTE:

Poiché non esistono dipendenze funzionali, INTERESSI UTENTE è in BCNF.

Tabella INTERESSE:

Poiché non esistono dipendenze funzionali, INTERESSE è in BCNF.

Tabella VOTO:

- IDVoto → Risposta, Stelle

Poiché la parte sinistra è superchiave, VOTO è in BCNF.

Tabella RICHIESTA AMICIZIA:

- IDRichiesta → Username, UsernameDestinatario, StatoRichiesta

Poiché la parte sinistra è superchiave, RICHIESTA AMICIZIA è in BCNF.

Tabella AMICO:

- UsernameAmico → Username

Poiché la parte sinistra è superchiave, AMICO è in BCNF.

Tabella APPARTENENZA:

Poiché non esistono dipendenze funzionali, APPARTENENZA è in BCNF.

Tabella SUGGERIMENTO INSERIMENTO:

Poiché non esistono dipendenze funzionali, SUGGERIMENTO INSERIMENTO è in BCNF.

Tabella CERCHIA:

- NomeCerchia → Argomento

Poiché la parte sinistra è superchiave, CERHIA è in BCNF.

Tabella PARTECIPANTE:

Poiché non esistono dipendenze funzionali, PARTECIPANTE è in BCNF.

Tabella SFIDA:

- Proponente, DataInizioSfida → DataLancio, DataScadenza, Scopo, ThreadSfida, AllenamentoSfida, AlimentazioneSfida, Vincitore

Poiché la parte sinistra è superchiave, SFIDA è in BCNF.

Tabella CLIENTE OCCASIONALE:

- CodFiscaleCliente → TipologiaTariffa, TariffaAccesso

Poiché la parte sinistra è superchiave, CLIENTE OCCASIONALE è in BCNF.

Tutte le tabelle ottenute sono nella *forma normale di Boyce-Codd*, che prevede che la chiusura dell'insieme degli attributi che compongono la chiave contenga tutti gli attributi della relazione considerata.

7 Descrizione Analytic

7.1 Performance sportiva

L'analisi della performance di un cliente del centro fitness ci permette di valutare il suo percorso sportivo, verificando se ha ottenuto dei progressi o meno e in tal caso di proporre una soluzione per vanificare al meglio gli sforzi del medesimo.

L'analisi consiste nel contare quante volte il cliente ha eseguito ogni esercizio della scheda in maniera corretta, in un tempo medio di 2 settimane.

Nelle volte in cui il cliente non è riuscito a completare l'esercizio con successo vengono presi in considerazione la differenza tra il reale sforzo del cliente e quello che in realtà andava svolto misurando lo scostamento tra i valori.

Per poter avere un giudizio positivo della performance del cliente, quest'ultimo deve aver concluso con successo **almeno il 70 % degli esercizi** da lui svolti.

In caso contrario si cercherà di creare una scheda di allenamento con esercizi che hanno un **Dispendio Energetico Medio** minore rispetto alla precedente scheda.

Gli esercizi da cambiare sono quelli in cui il cliente è riuscito meno volte a concludere in maniera completa, sostituendoli con esercizi di tipologia simile ma con un dispendio energetico medio minore.

Nello stesso tempo si deve valutare di non sostituire l'esercizio con un altro troppo semplice o comunque poco produttivo per l'obiettivo del cliente.

In seguito, verranno esposti i criteri con cui verranno scelti gli esercizi da sostituire con gli esercizi "complessi".

Supponendo che gli esercizi vanno da un Dispendio Energetico Medio da 5 a 100 sulla scala di alcuni valori si stabilizzerà con quale esercizio converrà sostituire l'esercizio incriminato.

D.E.M dell'Esercizio assegnato	Svolto correttamente	D.E.M dell'Esercizio da assegnare
100 - 90	70% - 50%	80 - 71
100 - 90	50% - 20%	70 - 51
100 - 90	20% - 0 %	50
90 - 70	70% - 50%	60 - 51
90 - 70	50% - 20%	50 - 41
90 - 70	20% - 0 %	40
70 - 50	70% - 50%	45 - 36
70 - 50	50% - 20%	35 - 26
70 - 50	20% - 0 %	25
50 - 30	70% - 50%	25 - 16
50 - 30	50% - 20%	15 - 11
50 - 30	20% - 0 %	10
30 - 10	50% - 0%	5

Nel caso in cui un esercizio dovrebbe essere con ripetizione la valutazione dell'intero esercizio viene valutata positiva se il cliente esegue in tutte le ripetizioni il numero di esecuzioni previste per ogni ripetizione. È ammessa una tollerabilità riguardo lo svolgimento dell'ultima ripetizione: il cliente può svolgere il 70% del numero di esecuzioni dell'ultima ripetizione per poter considerare la valutazione dell'esercizio come positiva.

Un ultimo aspetto da valutare è sulla configurazione di utilizzo utilizzata dal cliente. Nel momento in cui la configurazione dell'attrezzo utilizzata dal cliente è diversa dalla configurazione esatta allora la valutazione dell'esecuzione dell'esercizio viene considerata come negativa.

8 Implementazione

8.1 Definizione Database e Vincoli di Integrità Referenziale

```
SET NAMES latin1;
SET FOREIGN_KEY_CHECKS = 0;

-- -----
-- DEFINIZIONE DATABASE E TABELLE
-- -----

-- -----
-- Database AziendaCentriFitness
-- -----

BEGIN;
CREATE DATABASE IF NOT EXISTS `AziendaCentroFitness`;
COMMIT;

USE `AziendaCentroFitness`;

-- -----
-- Tabella Dipendente
-- -----

DROP TABLE IF EXISTS Dipendente;
CREATE TABLE Dipendente (
  CodFiscaleDip VARCHAR(45) NOT NULL,
  Nome VARCHAR (45) NOT NULL,
  Cognome VARCHAR (45) NOT NULL,
  DataNascita DATE NOT NULL,
  Citta VARCHAR(45) NOT NULL,
  Via VARCHAR (45) NOT NULL,
  NumCivico INT (5) NOT NULL,
  CAP INT(5) NOT NULL,
  Sesso CHAR(1) NOT NULL,
  IDDocumentoRiconoscimento VARCHAR (45) NOT NULL,
  Prefettura VARCHAR (45) NOT NULL,
  Responsabile VARCHAR (45) DEFAULT NULL,
  PRIMARY KEY(CodFiscaleDip),
  CONSTRAINT FK_Responsabile_Dipendente
    FOREIGN KEY(Responsabile)
      REFERENCES Dipendente(CodFiscaleDip)
      ON UPDATE CASCADE
      ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```

-- -----
-- Tabella CentroFitness
-- -----
DROP TABLE IF EXISTS CentroFitness;
CREATE TABLE CentroFitness (
    IDCentro VARCHAR(45) NOT NULL,
    Dimensione INT(45) NOT NULL,
    Telefono INT(10) NOT NULL,
    CapienzaMassima INT (45) NOT NULL,
    Citta VARCHAR(45) NOT NULL,
    Via VARCHAR (45) NOT NULL,
    NumCivico INT (5) NOT NULL,
    CAP INT(5) NOT NULL,
    Direttore VARCHAR(45) NOT NULL,
    PRIMARY KEY(IDCentro),
    CONSTRAINT FK_Direttore_Dipendente
        FOREIGN KEY(Direttore)
            REFERENCES Dipendente(CodFiscaleDip)
            ON UPDATE CASCADE
            ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----
-- Tabella CalendarioCentroFitness
-- -----
DROP TABLE IF EXISTS CalendarioCentroFitness;
CREATE TABLE CalendarioCentroFitness (
    IDCentro VARCHAR(45) NOT NULL,
    NomeGiorno VARCHAR (45) NOT NULL,
    OrarioApertura TIME NOT NULL,
    OrarioChiusura TIME NOT NULL,
    PRIMARY KEY(IDCentro, NomeGiorno),
    CONSTRAINT FK_Calendario_Centro
        FOREIGN KEY(IDCentro)
            REFERENCES CentroFitness(IDCentro)
            ON UPDATE CASCADE
            ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----
-- Tabella Sala
-- -----
DROP TABLE IF EXISTS Sala;
CREATE TABLE Sala (
    IDSala VARCHAR (45) NOT NULL,
    IDCentro VARCHAR(45) NOT NULL,
    Nome VARCHAR(45) NOT NULL,
    TipologiaContratto VARCHAR (45) NOT NULL,
    ResponsabileSala VARCHAR(45) NOT NULL,
    PRIMARY KEY(IDSala),
    CONSTRAINT FK_Sala_Centro

```

```

FOREIGN KEY(IDCentro)
REFERENCES CentroFitness(IDCentro)
ON UPDATE NO ACTION
ON DELETE NO ACTION,
CONSTRAINT FK_Responsabile_Sala
FOREIGN KEY(ResponsabileSala)
REFERENCES Dipendente(CodFiscaleDip)
ON UPDATE CASCADE
ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella Attrezzo
-- -----

```

```

DROP TABLE IF EXISTS Attrezzo;
CREATE TABLE Attrezzo (
  IDAttrezzo VARCHAR(45) NOT NULL,
  NomeAttrezzo VARCHAR(45) NOT NULL,
  IDSala VARCHAR (45) NOT NULL,
  ConsumoEnergetico INT (45) NOT NULL,
  Tipologia VARCHAR(45) NOT NULL,
  LivelloUsura INT(5) NOT NULL DEFAULT 0,
  PRIMARY KEY(IDAttrezzo),
  CONSTRAINT FK_Attrezzo_Sala
  FOREIGN KEY(IDSala)
  REFERENCES Sala(IDSala)
  ON UPDATE CASCADE
  ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella OrarioDiLavoro
-- -----

```

```

DROP TABLE IF EXISTS OrarioDiLavoro;
CREATE TABLE OrarioDiLavoro (
  IDCentro VARCHAR(45) NOT NULL,
  CodFiscaleDip VARCHAR (45) NOT NULL,
  NomeGiornoLavoro VARCHAR(45) NOT NULL,
  attivita VARCHAR (45) NOT NULL,
  OrarioInizio TIME NOT NULL,
  OrarioFine TIME NOT NULL,
  PRIMARY KEY(IDCentro, CodFiscaleDip, NomeGiornoLavoro, Attivita,
OrarioInizio),
  CONSTRAINT FK_Centro_OrarioLavoro
  FOREIGN KEY(IDCentro)
  REFERENCES CentroFitness(IDCentro)
  ON UPDATE CASCADE
  ON DELETE CASCADE,
  CONSTRAINT FK_Dipendente_OrarioLavoro
  FOREIGN KEY(CodFiscaleDip)
  REFERENCES Dipendente(CodFiscaleDip)

```

```

        ON UPDATE CASCADE
        ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella Cliente
-- -----

```

```

DROP TABLE IF EXISTS Cliente;
CREATE TABLE Cliente (
    CodFiscaleCliente VARCHAR(45) NOT NULL,
    Nome VARCHAR (45) NOT NULL,
    Cognome VARCHAR (45) NOT NULL,
    DataNascita DATE NOT NULL,
    Citta VARCHAR(45) NOT NULL,
    Via VARCHAR (45) NOT NULL,
    NumCivico INT (5) NOT NULL,
    CAP INT(5) NOT NULL,
    Sesso CHAR(1) NOT NULL,
    IDDocumentoRiconoscimento VARCHAR (45) NOT NULL,
    Prefettura VARCHAR (45) NOT NULL,
    PRIMARY KEY(CodFiscaleCliente))
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella Contratto
-- -----

```

```

DROP TABLE IF EXISTS Contratto;
CREATE TABLE Contratto (
    IDContratto VARCHAR(45) NOT NULL,
    IDCentro VARCHAR (45) NOT NULL,
    Cliente VARCHAR (45) NOT NULL,
    Consulente VARCHAR (45) NOT NULL,
    TipologiaContratto VARCHAR(45) NOT NULL,
    AccessiSettimanali INT (5),
    ImportoMensile INT(5) NOT NULL,
    DurataMensile INT (5) NOT NULL,
    DataIscrizione DATE NOT NULL,
    Scopo VARCHAR (45) NOT NULL,
    PRIMARY KEY(IDContratto),
    CONSTRAINT FK_Centro_Contratto
        FOREIGN KEY(IDCentro)
            REFERENCES CentroFitness(IDCentro)
            ON UPDATE CASCADE
            ON DELETE CASCADE,
    CONSTRAINT FK_Consulente_Dipendente
        FOREIGN KEY(Consulente)
            REFERENCES Dipendente(CodFiscaleDip)
            ON UPDATE CASCADE
            ON DELETE CASCADE,
    CONSTRAINT FK_Cliente_Contratto
        FOREIGN KEY(Cliente)
            REFERENCES Cliente (CodFiscaleCliente)

```

```

        ON UPDATE CASCADE
        ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella FasciMuscolari
-- -----

```

```

DROP TABLE IF EXISTS FasciMuscolari;
CREATE TABLE FasciMuscolari (
    NomeFasciMuscolari VARCHAR(45) NOT NULL,
    Livello VARCHAR (45) NOT NULL,
    PRIMARY KEY(NomeFasciMuscolari, Livello))
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella Potenziamento
-- -----

```

```

DROP TABLE IF EXISTS Potenziamento;
CREATE TABLE Potenziamento (
    IDContratto VARCHAR(45) NOT NULL,
    NomeFasciMuscolari VARCHAR (45) NOT NULL,
    Livello VARCHAR (45) NOT NULL,
    PRIMARY KEY(IDContratto, NomeFasciMuscolari),
    CONSTRAINT FK_Potenziamento_Contratto
        FOREIGN KEY(IDContratto)
            REFERENCES Contratto(IDContratto)
            ON UPDATE CASCADE
            ON DELETE CASCADE,
    CONSTRAINT FK_Potenziamento_Muscoli
        FOREIGN KEY(NomeFasciMuscolari, Livello)
            REFERENCES FasciMuscolari(NomeFasciMuscolari, Livello)
            ON UPDATE CASCADE
            ON DELETE CASCADE
    )
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella PianoPagamento
-- -----

```

```

DROP TABLE IF EXISTS PianoPagamento;
CREATE TABLE PianoPagamento (
    IDPagamento VARCHAR (45) NOT NULL,
    IDContratto VARCHAR(45) NOT NULL,
    TipologiaPagamento VARCHAR (45) NOT NULL,
    PRIMARY KEY(IDPagamento),
    CONSTRAINT FK_Pagamento_Contratto
        FOREIGN KEY(IDContratto)
            REFERENCES Contratto(IDContratto)
            ON UPDATE CASCADE
            ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella Rata
-- -----
DROP TABLE IF EXISTS Rata;
CREATE TABLE Rata (
    IDRata VARCHAR(45) NOT NULL,
    IDPagamento VARCHAR (45) NOT NULL,
    ImportoRata INT(5) NOT NULL,
    IstitutoFinanziario VARCHAR (45) NOT NULL,
    TassoInteresse INT (5) NOT NULL,
    DataScadenza DATE NOT NULL,
    Stato VARCHAR(45) NOT NULL,
    PRIMARY KEY(IDRata),
    CONSTRAINT FK_Pagamento_Rata
        FOREIGN KEY(IDPagamento)
            REFERENCES PianoPagamento(IDPagamento)
            ON UPDATE CASCADE
            ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella Spogliatoio
-- -----
DROP TABLE IF EXISTS Spogliatoio;
CREATE TABLE Spogliatoio (
    IDSpogliatoio VARCHAR(45) NOT NULL,
    Capienza INT(5) NOT NULL,
    PostiDisponibili INT (5) NOT NULL,
    Posizione VARCHAR (45) NOT NULL,
    Sesso CHAR (1) NOT NULL,
    PRIMARY KEY(IDSpogliatoio))
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella Armadietto
-- -----
DROP TABLE IF EXISTS Armadietto;
CREATE TABLE Armadietto (
    IDArmadietto VARCHAR(45) NOT NULL,
    IDSpogliatoio VARCHAR (45) NOT NULL,
    CombinazioneNumerica VARCHAR (45) NOT NULL,
    PRIMARY KEY(IDArmadietto),
    CONSTRAINT FK_Armadietto_Spogliatoio
        FOREIGN KEY(IDSpogliatoio)
            REFERENCES Spogliatoio(IDSpogliatoio)
            ON UPDATE CASCADE
            ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```



```

-- -----
-- Tabella Accesso
-- -----
DROP TABLE IF EXISTS Accesso;
CREATE TABLE Accesso (
  IDAccesso VARCHAR(45) NOT NULL,
  IDCentro VARCHAR (45) NOT NULL,
  CodFiscaleCliente VARCHAR (45) NOT NULL,
  IDArmadietto VARCHAR (45) NOT NULL,
  DataAccesso DATE NOT NULL,
  OrarioEntrata TIME NOT NULL,
  OrarioUscita TIME DEFAULT NULL,
  PRIMARY KEY(IDAccesso),
  CONSTRAINT FK_Centro_Accesso
    FOREIGN KEY(IDCentro)
      REFERENCES CentroFitness(IDCentro)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
  CONSTRAINT FK_Cliente_Accesso
    FOREIGN KEY(CodFiscaleCliente)
      REFERENCES Cliente(CodFiscaleCliente)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
  CONSTRAINT FK_Armadietto_Accesso
    FOREIGN KEY(IDArmadietto)
      REFERENCES Armadietto (IDArmadietto)
    ON UPDATE CASCADE
    ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----
-- Tabella MisurazioniImpedenziometriche
-- -----
DROP TABLE IF EXISTS MisurazioniImpedenziometriche;
CREATE TABLE MisurazioniImpedenziometriche (
  IDMisurazione VARCHAR(45) NOT NULL,
  CodFiscaleCliente VARCHAR (45) NOT NULL,
  DataMisurazione DATE NOT NULL,
  AcquaTotale DOUBLE NOT NULL,
  MassaMagra DOUBLE NOT NULL,
  MassaGrassa DOUBLE NOT NULL,
  Peso INT(5) NOT NULL,
  Altezza DOUBLE NOT NULL,
  StatoCliente VARCHAR (45) NOT NULL,
  Entita VARCHAR (45) NOT NULL,
  PRIMARY KEY(IDMisurazione),
  CONSTRAINT FK_Misurazione_Cliente
    FOREIGN KEY(CodFiscaleCliente)
      REFERENCES Cliente(CodFiscaleCliente)
    ON UPDATE CASCADE
    ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella Dieta
-- -----
DROP TABLE IF EXISTS Dieta;
CREATE TABLE Dieta (
    IDDieta VARCHAR(45) NOT NULL,
    ApportoCalorico INT (5) NOT NULL,
    NumeroPasti INT(5) NOT NULL,
    PRIMARY KEY(IDDieta)
)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----
-- Tabella SchedaAlimentazione
-- -----
DROP TABLE IF EXISTS SchedaAlimentazione;
CREATE TABLE SchedaAlimentazione (
    IDSchedaAlimentazione VARCHAR(45) NOT NULL,
    CodFiscaleCliente VARCHAR (45) NOT NULL,
    IDDieta VARCHAR (45) NOT NULL,
    DataInizio DATE NOT NULL,
    DataFine DATE NOT NULL,
    MedicoScheda VARCHAR (45) NOT NULL,
    Obiettivo VARCHAR (45) NOT NULL,
    PRIMARY KEY(IDSchedaAlimentazione),
    CONSTRAINT FK_Alimentaione_Cliente
        FOREIGN KEY(CodFiscaleCliente)
            REFERENCES Cliente(codFiscaleCliente)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    CONSTRAINT FK_Alimentazione_Medico
        FOREIGN KEY(MedicoScheda)
            REFERENCES Dipendente(CodFiscaleDip)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    CONSTRAINT FK_Dieta_Alimentazione
        FOREIGN KEY(IDDieta)
            REFERENCES Dieta (IDDieta)
        ON UPDATE CASCADE
        ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----
-- Tabella Pietanza
-- -----
DROP TABLE IF EXISTS Pietanza;
CREATE TABLE Pietanza (
    NomePietanza VARCHAR(45) NOT NULL,
    ComposizionePietanza VARCHAR (2000) NOT NULL,
    PRIMARY KEY(NomePietanza))
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella ComposizioneDieta
-- -----
DROP TABLE IF EXISTS ComposizioneDieta;
CREATE TABLE ComposizioneDieta (
    IDDieta VARCHAR(45) NOT NULL,
    NomePietanza VARCHAR (45) NOT NULL,
    Pasto VARCHAR (45) NOT NULL,
    PRIMARY KEY(IDDieta, NomePietanza),
    CONSTRAINT FK_ComposizioneDieta_NomePietanza
        FOREIGN KEY(NomePietanza)
            REFERENCES Pietanza(NomePietanza)
            ON UPDATE CASCADE
            ON DELETE CASCADE,
    CONSTRAINT FK_ComposizioneDieta_Dieta
        FOREIGN KEY(IDDieta)
            REFERENCES Dieta(IDDieta)
            ON UPDATE CASCADE
            ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----
-- Tabella AlimentazionePassata
-- -----
DROP TABLE IF EXISTS AlimentazionePassata;
CREATE TABLE AlimentazionePassata (
    IDSchedaAlimentazione VARCHAR(45) NOT NULL,
    CodFiscaleCliente VARCHAR (45) NOT NULL,
    PRIMARY KEY(IDSchedaAlimentazione, CodFiscaleCliente),
    CONSTRAINT FK_SchedaAlimentazione_AlimentazionePassata
        FOREIGN KEY(IDSchedaAlimentazione)
            REFERENCES SchedaAlimentazione(IDSchedaAlimentazione)
            ON UPDATE CASCADE
            ON DELETE CASCADE,
    CONSTRAINT FK_Cliente_AlimentazionePassata
        FOREIGN KEY(CodFiscaleCliente)
            REFERENCES Cliente (CodFiscaleCliente)
            ON UPDATE CASCADE
            ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella SchedaAllenamento
-- -----
DROP TABLE IF EXISTS SchedaAllenamento;
CREATE TABLE SchedaAllenamento (
    IDSchedaAllenamento VARCHAR(45) NOT NULL,
    CodFiscaleCliente VARCHAR (45) NOT NULL,
    TutorScheda VARCHAR (45) NOT NULL,
    DataInizio DATE NOT NULL,
    DataFine DATE DEFAULT NULL,
    PRIMARY KEY(IDSchedaAllenamento),
    CONSTRAINT FK_TutorScheda_Dipendente
    FOREIGN KEY(Tutorscheda)
        REFERENCES Dipendente(CodFiscaleDip)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
    CONSTRAINT FK_Cliente_SchedaAllenamento
    FOREIGN KEY(CodFiscaleCliente)
        REFERENCES Cliente (CodFiscaleCliente)
    ON UPDATE CASCADE
    ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----
-- Tabella AllenamentoPassato
-- -----
DROP TABLE IF EXISTS AllenamentoPassato;
CREATE TABLE AllenamentoPassato (
    IDSchedaAllenamento VARCHAR(45) NOT NULL,
    CodFiscaleCliente VARCHAR (45) NOT NULL,
    PRIMARY KEY(IDSchedaAllenamento, CodFiscaleCliente),
    CONSTRAINT FK_AllenamentoPassato_SchedaAllenamento
    FOREIGN KEY(IDSchedaAllenamento)
        REFERENCES SchedaAllenamento(IDSchedaAllenamento)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
    CONSTRAINT FK_Cliente_AllenamentoPassato
    FOREIGN KEY(CodFiscaleCliente)
        REFERENCES Cliente (CodFiscaleCliente)
    ON UPDATE CASCADE
    ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----
-- Tabella Esercizio
-- -----
DROP TABLE IF EXISTS Esercizio;
CREATE TABLE Esercizio (
    IDEsercizio VARCHAR(45) NOT NULL,
    Nome VARCHAR (45) NOT NULL,
    DispendioEnergeticoMedio INT (45) NOT NULL,
    PRIMARY KEY(IDEsercizio))
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella ComposizioneAllenamento
-- -----
DROP TABLE IF EXISTS ComposizioneAllenamento;
CREATE TABLE ComposizioneAllenamento (
    IDEsercizio VARCHAR(45) NOT NULL,
    IDSchedaAllenamento VARCHAR (45) NOT NULL,
    PRIMARY KEY(IDEsercizio, IdSchedaAllenamento),
    CONSTRAINT FK_EsercizioScheda_Esercizio
        FOREIGN KEY(IDEsercizio)
            REFERENCES Esercizio(IDEsercizio)
            ON UPDATE CASCADE
            ON DELETE CASCADE,
    CONSTRAINT FK_SchedaAllenamento_Esercizio
        FOREIGN KEY(IDSchedaAllenamento)
            REFERENCES SchedaAllenamento(IDSchedaAllenamento)
            ON UPDATE CASCADE
            ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----
-- Tabella Aerobico
-- -----
DROP TABLE IF EXISTS Aerobico;
CREATE TABLE Aerobico (
    IDEsercizioAerobico VARCHAR(45) NOT NULL,
    DurataMinuti INT(5) NOT NULL,
    PRIMARY KEY(IDEsercizioAerobico),
    CONSTRAINT FK_Aerobico_Esercizio
        FOREIGN KEY(IDEsercizioAerobico)
            REFERENCES Esercizio(IDEsercizio)
            ON UPDATE CASCADE
            ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----
-- Tabella Anaerobico
-- -----
DROP TABLE IF EXISTS Anaerobico;

CREATE TABLE Anaerobico (
    IDEsercizioAnaerobico VARCHAR(45) NOT NULL,
    NomeAnaerobico VARCHAR (45) NOT NULL,
    NumEsecuzioni INT (45) NOT NULL,
    TempoDiRecupero INT (45) NOT NULL,
    PRIMARY KEY(IDEsercizioAnaerobico, NomeAnaerobico),
    CONSTRAINT FK_Anaerobico_Esercizio
        FOREIGN KEY(IDEsercizioAnaerobico)
            REFERENCES Esercizio(IDEsercizio)
            ON UPDATE CASCADE
            ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella ConfigurazioneUtilizzo
-- -----
DROP TABLE IF EXISTS ConfigurazioneUtilizzo;
CREATE TABLE ConfigurazioneUtilizzo (
    IDConfigurazione VARCHAR(45) NOT NULL,
    IDEsercizio VARCHAR (45) NOT NULL,
    IDAttrezzo VARCHAR (45) NOT NULL,
    NomeRegolazione VARCHAR (45) NOT NULL,
    Intensita INT (5) NOT NULL,
    IstruzioneEsatta BOOLEAN NOT NULL,
    PRIMARY KEY(IDConfigurazione),
    CONSTRAINT FK_Esercizio_Configiurazione
        FOREIGN KEY(IDEsercizio)
            REFERENCES Esercizio(IDEsercizio)
            ON UPDATE CASCADE
            ON DELETE CASCADE,
    CONSTRAINT FK_Configurazione_Attrezzo
        FOREIGN KEY(IDAttrezzo)
            REFERENCES Attrezzo(IDAttrezzo)
            ON UPDATE CASCADE
            ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----
-- Tabella MonitoraggioSmart
-- -----
DROP TABLE IF EXISTS MonitoraggioSmart;
CREATE TABLE MonitoraggioSmart (
    IDMonitoraggio VARCHAR(45) NOT NULL,
    IDEsercizio VARCHAR (45) NOT NULL,
    IDSchedaAllenamento VARCHAR (45) NOT NULL,
    IstanteInizio TIME NOT NULL,
    IstanteFine TIME NOT NULL,
    DataMonitoraggio DATE NOT NULL,
    PRIMARY KEY(IDMonitoraggio),
    CONSTRAINT FK_Monitoraggio_Allenamento
        FOREIGN KEY(IDSchedaAllenamento)
            REFERENCES SchedaAllenamento(IDSchedaAllenamento)
            ON UPDATE CASCADE
            ON DELETE CASCADE,
    CONSTRAINT FK_Monitoraggio_Esercizio
        FOREIGN KEY(IDEsercizio)
            REFERENCES Esercizio(IDEsercizio)
            ON UPDATE CASCADE
            ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella DatiRipetizione
-- -----
DROP TABLE IF EXISTS DatiRipetizione;
CREATE TABLE DatiRipetizione (
    IDRipetizione VARCHAR(45) NOT NULL,
    NomeRipetizione VARCHAR (45) NOT NULL,
    NumeroEsecuzioni VARCHAR (45) NOT NULL,
    TempoRecupero INT(5) NOT NULL,
    PRIMARY KEY(IDRipetizione))
ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----
-- Tabella MonitoraggioRipetizione
-- -----
DROP TABLE IF EXISTS MonitoraggioRipetizione;
CREATE TABLE MonitoraggioRipetizione (
    IDMonitoraggio VARCHAR(45) NOT NULL,
    IDRipetizione VARCHAR (45) NOT NULL,
    PRIMARY KEY(IDMonitoraggio, IDRipetizione),
    CONSTRAINT FK_Monitoraggio_Ripetizione
        FOREIGN KEY(IDMonitoraggio)
            REFERENCES MonitoraggioSmart(IDMonitoraggio)
            ON UPDATE CASCADE
            ON DELETE CASCADE,
    CONSTRAINT FK_DatiRipetizione_Ripetizione
        FOREIGN KEY(IDRipetizione)
            REFERENCES DatiRipetizione(IDRipetizione)
            ON UPDATE CASCADE
            ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----
-- Tabella ConfigurazioneCliente
-- -----
DROP TABLE IF EXISTS ConfigurazioneCliente;
CREATE TABLE ConfigurazioneCliente (
    IDMonitoraggio VARCHAR(45) NOT NULL,
    IDConfigurazione VARCHAR (45) NOT NULL,
    PRIMARY KEY(IDMonitoraggio, IDConfigurazione),
    CONSTRAINT FK_Monitoraggio_Configurazione
        FOREIGN KEY(IDMonitoraggio)
            REFERENCES MonitoraggioSmart(IDMonitoraggio)
            ON UPDATE CASCADE
            ON DELETE CASCADE,
    CONSTRAINT FK_ConfigurazioneCli_MonitoraggioCli
        FOREIGN KEY(IDConfigurazione)
            REFERENCES ConfigurazioneUtilizzo(IDConfigurazione)
            ON UPDATE CASCADE
            ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella ConfigurazioneRipetizione
-- -----
DROP TABLE IF EXISTS ConfigurazioneRipetizione;
CREATE TABLE ConfigurazioneRipetizione (
    EsercizioAnaerobico VARCHAR(45) NOT NULL,
    IDConfigurazione VARCHAR (45) NOT NULL,
    Nome VARCHAR (45) NOT NULL,
    PRIMARY KEY(EsercizioAnaerobico, IDConfigurazione, Nome),
    CONSTRAINT FK_Anaerobico_Configurazione
        FOREIGN KEY(EsercizioAnaerobico, Nome)
            REFERENCES Anaerobico(IDEsercizioAnaerobico, NomeAnaerobico)
            ON UPDATE CASCADE
            ON DELETE CASCADE,
    CONSTRAINT FK_ConfigurazioneAnaerobico_Configurazione
        FOREIGN KEY(IDConfigurazione)
            REFERENCES ConfigurazioneUtilizzo(IDConfigurazione)
            ON UPDATE CASCADE
            ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----
-- Tabella AttrezzoUtilizzato
-- -----
DROP TABLE IF EXISTS AttrezzoUtilizzato;
CREATE TABLE AttrezzoUtilizzato (
    IDAttrezzo VARCHAR(45) NOT NULL,
    IDMonitoraggio VARCHAR (45) NOT NULL,
    PRIMARY KEY(IDAttrezzo, IDMonitoraggio),
    CONSTRAINT FK_Utilizzato_Attrezzo
        FOREIGN KEY(IDAttrezzo)
            REFERENCES Attrezzo(IDAttrezzo)
            ON UPDATE CASCADE
            ON DELETE CASCADE,
    CONSTRAINT FK_Utilizzato_Monitoraggio
        FOREIGN KEY(IDMonitoraggio)
            REFERENCES MonitoraggioSmart(IDMonitoraggio)
            ON UPDATE CASCADE
            ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----
-- Tabella Piscina
-- -----
DROP TABLE IF EXISTS Piscina;
CREATE TABLE Piscina (
    IDPiscina VARCHAR(45) NOT NULL,
    IDCentro VARCHAR (45) NOT NULL,
    Locazione VARCHAR (45) NOT NULL,
    TipologiaContratto VARCHAR(45) NOT NULL,

```



```

Dimensione INT (5) NOT NULL,
PRIMARY KEY(IDPiscina),
CONSTRAINT FK_Centro_Piscina
    FOREIGN KEY(IDCentro)
        REFERENCES CentroFitness(IDCentro)
    ON UPDATE CASCADE
    ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella AccessoPiscina
-- -----
DROP TABLE IF EXISTS AccessoPiscina;
CREATE TABLE AccessoPiscina (
    IDPiscina VARCHAR (45) NOT NULL,
    IDContratto VARCHAR(45) NOT NULL,
    NumeroAccessi INT (5) NOT NULL,
    PRIMARY KEY(IDPiscina, IDContratto),
    CONSTRAINT FK_AccessoPiscina_Contratto
        FOREIGN KEY(IDContratto)
            REFERENCES Contratto (IDContratto)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    CONSTRAINT FK_AccessoPiscina_Piscina
        FOREIGN KEY(IDPiscina)
            REFERENCES Piscina(IDPiscina)
        ON UPDATE CASCADE
        ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella MultiSede
-- -----
DROP TABLE IF EXISTS MultiSede;
CREATE TABLE MultiSede (
    IDContratto VARCHAR(45) NOT NULL,
    IDCentro VARCHAR (45) NOT NULL,
    FasciaOraria VARCHAR (45) NOT NULL,
    IngressiMax INT (5) NOT NULL,
    PRIMARY KEY(IDContratto, IDCentro),
    CONSTRAINT FK_Centro_Multisede
        FOREIGN KEY(IDCentro)
            REFERENCES CentroFitness(IDCentro)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    CONSTRAINT FK_MultiSede_Contratto
        FOREIGN KEY(IDContratto)
            REFERENCES Contratto(IDContratto)
        ON UPDATE CASCADE
        ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella Corso
-- -----
DROP TABLE IF EXISTS Corso;
CREATE TABLE Corso (
    IDCorso VARCHAR(45) NOT NULL,
    NomeDisciplina VARCHAR (45) NOT NULL,
    IstruttoreCorso VARCHAR (45) NOT NULL,
    DataInizio DATE NOT NULL,
    DataFine DATE NOT NULL,
    NumeroMaxPartecipanti INT (5) NOT NULL,
    Livello VARCHAR (45) NOT NULL,
    PRIMARY KEY(IDCorso),
    CONSTRAINT FK_Istruttore_Corso
        FOREIGN KEY(IstruttoreCorso)
            REFERENCES Dipendente(CodFiscaleDip)
            ON UPDATE CASCADE
            ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----
-- Tabella CorsoFrequentato
-- -----
DROP TABLE IF EXISTS CorsoFrequentato;
CREATE TABLE CorsoFrequentato (
    IDCorso VARCHAR(45) NOT NULL,
    CodFiscaleCliente VARCHAR (45) NOT NULL,
    PRIMARY KEY(IDCorso, CodFiscaleCliente),
    CONSTRAINT FK_CorsoFrequentato_Corso
        FOREIGN KEY(IDCorso)
            REFERENCES Corso(IDCorso)
            ON UPDATE CASCADE
            ON DELETE CASCADE,
    CONSTRAINT FK_CorsoFrequentato_Cliente
        FOREIGN KEY(CodFiscaleCliente)
            REFERENCES Cliente(CodFiscaleCliente)
            ON UPDATE CASCADE
            ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----
-- Tabella CorsoDisponibile
-- -----
DROP TABLE IF EXISTS CorsoDisponibile;
CREATE TABLE CorsoDisponibile (
    IDCorso VARCHAR (45) NOT NULL,
    IDContratto VARCHAR(45) NOT NULL,

```

```

PRIMARY KEY(IDCorso, IDContratto),
CONSTRAINT FK_CorsoDisponibile_Contratto
  FOREIGN KEY(IDContratto)
    REFERENCES Contratto(IDContratto)
  ON UPDATE CASCADE
  ON DELETE CASCADE,
CONSTRAINT FK_CorsoDisponibile_Corso
  FOREIGN KEY(IDCorso)
    REFERENCES Corso(IDCorso)
  ON UPDATE CASCADE
  ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella CorsoInPiscina
-- -----
DROP TABLE IF EXISTS CorsoInPiscina;
CREATE TABLE CorsoInPiscina (
  IDCorso VARCHAR(45) NOT NULL,
  IDPiscina VARCHAR (45) NOT NULL,
  PRIMARY KEY(IDCorso, IDPiscina),
  CONSTRAINT FK_CorsoPiscina_Corso
    FOREIGN KEY(IDCorso)
      REFERENCES Corso(IDCorso)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
  CONSTRAINT FK_CorsoPiscina_Piscina
    FOREIGN KEY(IDPiscina)
      REFERENCES Piscina (IDPiscina)
    ON UPDATE CASCADE
    ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella CorsoInSala
-- -----
DROP TABLE IF EXISTS CorsoInSala;
CREATE TABLE CorsoInSala (
  IDCorso VARCHAR(45) NOT NULL,
  IDSala VARCHAR (45) NOT NULL,
  PRIMARY KEY(IDCorso, IDSala),
  CONSTRAINT FK_CorsoSala_Corso
    FOREIGN KEY(IDCorso)
      REFERENCES Corso(IDCorso)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
  CONSTRAINT FK_CorsoSala_Sala
    FOREIGN KEY(IDSala)
      REFERENCES Sala(IDSala)
    ON UPDATE CASCADE
    ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella CalendarioCorso
-- -----
DROP TABLE IF EXISTS CalendarioCorso;
CREATE TABLE CalendarioCorso (
    IDCalendarioCorso VARCHAR(45) NOT NULL,
    IDCorso VARCHAR (45) NOT NULL,
    IDCentro VARCHAR (45) NOT NULL,
    GiornoSettimana VARCHAR (45) NOT NULL,
    OrarioInizio TIME NOT NULL,
    OrarioFine TIME NOT NULL,
    PRIMARY KEY(IDCalendarioCorso),
    CONSTRAINT FK_CalendararioCorso_Centro
        FOREIGN KEY(IDCentro)
            REFERENCES CentroFitness(IDCentro)
            ON UPDATE CASCADE
            ON DELETE CASCADE,
    CONSTRAINT FK_CalendararioCorso_Corso
        FOREIGN KEY(IDCorso)
            REFERENCES Corso(IDCorso)
            ON UPDATE CASCADE
            ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella Utente
-- -----
DROP TABLE IF EXISTS Utente;
CREATE TABLE Utente (
    Username VARCHAR(45) NOT NULL,
    IDContratto VARCHAR (45) NOT NULL,
    Password VARCHAR (100) NOT NULL,
    Credibilita DOUBLE DEFAULT NULL,
    PRIMARY KEY(Username),
    CONSTRAINT FK_Utente_Contratto
        FOREIGN KEY(IDContratto)
            REFERENCES Contratto(IDContratto)
            ON UPDATE CASCADE
            ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella Thread
-- -----
DROP TABLE IF EXISTS Thread;
CREATE TABLE Thread (
    IDThread VARCHAR(45) NOT NULL,
    Titolo VARCHAR (45) NOT NULL,
    PRIMARY KEY(IDThread))
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella Post
-- -----
DROP TABLE IF EXISTS Post;
CREATE TABLE Post (
  IDPost VARCHAR(45) NOT NULL,
  Testo VARCHAR (5000) NOT NULL,
  Username VARCHAR (45) NOT NULL,
  Timestamp TIMESTAMP NOT NULL,
  Risposta VARCHAR(45) NOT NULL,
  AreaForum VARCHAR (50) NOT NULL,
  ThreadPost VARCHAR(50) NOT NULL,
  PRIMARY KEY(IDPost),
  CONSTRAINT FK_Post_Risposta
    FOREIGN KEY(Risposta)
      REFERENCES Post(IDPost)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
  CONSTRAINT FK_Post_Username
    FOREIGN KEY(Username)
      REFERENCES Utente(Username)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
  CONSTRAINT FK_Post_Thread
    FOREIGN KEY(ThreadPost)
      REFERENCES Thread (IDThread)
    ON UPDATE CASCADE
    ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella Link
-- -----
DROP TABLE IF EXISTS Link;
CREATE TABLE Link (
  URL VARCHAR(300) NOT NULL,
  PRIMARY KEY(URL))
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella LinkPost
-- -----
DROP TABLE IF EXISTS LinkPost;
CREATE TABLE LinkPost (
  URL VARCHAR(300) NOT NULL,
  IDPost VARCHAR (45) NOT NULL,
  PRIMARY KEY(URL, IDPost),
  CONSTRAINT FK_URL_LinkPost
    FOREIGN KEY(URL)
      REFERENCES Link(URL)

```

```

        ON UPDATE CASCADE
        ON DELETE CASCADE,
        CONSTRAINT FK_Post_LinkPost
        FOREIGN KEY(IDPost)
            REFERENCES Post(IDPost)
        ON UPDATE CASCADE
        ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----
-- Tabella Voto
-- -----
DROP TABLE IF EXISTS Voto;
CREATE TABLE Voto (
    IDVoto VARCHAR(45) NOT NULL,
    Risposta VARCHAR (45) NOT NULL,
    Stelle INT(5) NOT NULL,
    PRIMARY KEY(IDVoto),
    CONSTRAINT FK_Voto_Risposta
        FOREIGN KEY(Risposta)
            REFERENCES Post(Risposta)
        ON UPDATE CASCADE
        ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----
-- Tabella Interesse
-- -----
DROP TABLE IF EXISTS Interesse;
CREATE TABLE Interesse (
    NomeInteresse VARCHAR(45) NOT NULL,
    PRIMARY KEY(NomeInteresse))
ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----
-- Tabella InteressiUtente
-- -----
DROP TABLE IF EXISTS InteressiUtente;
CREATE TABLE InteressiUtente (
    Username VARCHAR(45) NOT NULL,
    NomeInteresse VARCHAR (45) NOT NULL,
    PRIMARY KEY(Username, NomeInteresse),
    CONSTRAINT FK_Username_InteresseUtente
        FOREIGN KEY(Username)
            REFERENCES Utente(Username)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    CONSTRAINT FK_Interesse_InteresseUtente
        FOREIGN KEY(NomeInteresse)
            REFERENCES Interesse(NomeInteresse)
        ON UPDATE CASCADE
        ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella RichiestaAmicizia
-- -----
DROP TABLE IF EXISTS RichiestaAmicizia;
CREATE TABLE RichiestaAmicizia (
    IDRichiesta VARCHAR(45) NOT NULL,
    Username VARCHAR (45) NOT NULL,
    UsernameDestinatario VARCHAR (45) NOT NULL,
    StatoRichiesta VARCHAR (45) NOT NULL,
    PRIMARY KEY(IDRichiesta),
    CONSTRAINT FK_Richiesta_Username
        FOREIGN KEY(Username)
            REFERENCES Utente(Username)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    CONSTRAINT FK_UsernameDest_Utente
        FOREIGN KEY(UsernameDestinatario)
            REFERENCES Utente(Username)
        ON UPDATE CASCADE
        ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella Amico
-- -----
DROP TABLE IF EXISTS Amico;
CREATE TABLE Amico (
    UsernameAmico VARCHAR(45) NOT NULL,
    Username VARCHAR (45) NOT NULL,
    PRIMARY KEY(UsernameAmico, Username),
    CONSTRAINT FK_Amico_Utente
        FOREIGN KEY(UsernameAmico)
            REFERENCES Utente (Username)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    CONSTRAINT FK_Amico_UtenteAmico
        FOREIGN KEY(Username)
            REFERENCES Utente(Username)
        ON UPDATE CASCADE
        ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella Cerchia
-- -----
DROP TABLE IF EXISTS Cerchia;
CREATE TABLE Cerchia (
    NomeCerchia VARCHAR(45) NOT NULL,
    Argomento VARCHAR (45) NOT NULL,
    PRIMARY KEY(NomeCerchia))
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella Appartenenza
-- -----
DROP TABLE IF EXISTS Appartenenza;
CREATE TABLE Appartenenza (
  NomeCerchia VARCHAR(45) NOT NULL,
  UsernameAmico VARCHAR (45) NOT NULL,
  Username VARCHAR (45) NOT NULL,
  PRIMARY KEY(NomeCerchia, UsernameAmico, Username),
  CONSTRAINT FK_Cerchia_Appartenenza
    FOREIGN KEY(NomeCerchia)
      REFERENCES Cerchia (NomeCerchia)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
  CONSTRAINT FK_UsernameAmico_Appartenenza
    FOREIGN KEY(UsernameAmico, Username)
      REFERENCES Amico(UsernameAmico, Username)
    ON UPDATE CASCADE
    ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella SuggerimentoInserimento
-- -----
DROP TABLE IF EXISTS SuggerimentoInserimento;
CREATE TABLE SuggerimentoInserimento (
  NomeCerchia VARCHAR(45) NOT NULL,
  UsernameAmico VARCHAR (45) NOT NULL,
  Username VARCHAR (45) NOT NULL,
  PRIMARY KEY(NomeCerchia, UsernameAmico, Username),
  CONSTRAINT FK_Cerchia_Suggerimento
    FOREIGN KEY(NomeCerchia)
      REFERENCES Cerchia (NomeCerchia)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
  CONSTRAINT FK_UsernameAmico_Suggerimento
    FOREIGN KEY(UsernameAmico)
      REFERENCES Amico(UsernameAmico)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
  CONSTRAINT FK_Username_Suggerimento
    FOREIGN KEY(Username)
      REFERENCES Amico (Username)
    ON UPDATE CASCADE
    ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella Sfida
-- -----
DROP TABLE IF EXISTS Sfida;

```



```

CREATE TABLE Sfida (
  Proponente VARCHAR(45) NOT NULL,
  DataInizioSfida DATE NOT NULL,
  DataLancio DATE NOT NULL,
  DataScadenza DATE DEFAULT NULL,
  Scopo VARCHAR (50) NOT NULL,
  ThreadSfida VARCHAR (50) NOT NULL,
  AllenamentoSfida VARCHAR (50) NOT NULL,
  AlimentazioneSfida VARCHAR (50) NOT NULL,
  Vincitore VARCHAR (50) DEFAULT NULL,
  PRIMARY KEY(Proponente, DataInizioSfida),
  CONSTRAINT FK_Proponente_Utente
    FOREIGN KEY(Proponente)
      REFERENCES Utente (Username)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
  CONSTRAINT FK_Thread_Sfida
    FOREIGN KEY(ThreadSfida)
      REFERENCES Thread(IDThread)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
  CONSTRAINT FK_Allenamento_Sfida
    FOREIGN KEY(AllenamentoSfida)
      REFERENCES SchedaAllenamento (IDSchedaAllenamento)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
  CONSTRAINT FK_Alimentazione_Sfida
    FOREIGN KEY (AlimentazioneSfida)
      REFERENCES SchedaAlimentazione (IDSchedaAlimentazione)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
  CONSTRAINT FK_Vincitore_Sfida
    FOREIGN KEY (Vincitore)
      REFERENCES Utente (Username)
    ON UPDATE CASCADE
    ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella Partecipante
-- -----

```

```

DROP TABLE IF EXISTS Partecipante;
CREATE TABLE Partecipante (
  UsernameAmico VARCHAR(45) NOT NULL,
  Proponente VARCHAR (45) NOT NULL,
  DataInizioSfida DATE NOT NULL,
  PRIMARY KEY(Proponente, UsernameAmico, DataInizioSfida),
  CONSTRAINT FK_UsernameAmico_Partecipante
    FOREIGN KEY(UsernameAmico)
      REFERENCES Amico (UsernameAmico)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
  CONSTRAINT FK_Proponente_Partecipante
    FOREIGN KEY(Proponente, DataInizioSfida)

```

```

        REFERENCES Sfida(Proponente, DataInizioSfida)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
    CONSTRAINT FK_Username_Partecipante
    FOREIGN KEY(Proponente)
        REFERENCES Amico (Username)
    ON UPDATE CASCADE
    ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
-- Tabella ClienteOccasionale
-- -----
DROP TABLE IF EXISTS ClienteOccasionale;
CREATE TABLE ClienteOccasionale (
    CodFiscaleCliente VARCHAR(45) NOT NULL,
    TipologiaTariffa VARCHAR (45) NOT NULL,
    TariffaAccesso INT (50) NOT NULL,
    PRIMARY KEY(CodFiscaleCliente),
    CONSTRAINT FK_Cliente_ClienteOccasionale
    FOREIGN KEY(CodFiscaleCliente)
        REFERENCES Cliente (CodFiscaleCliente)
    ON UPDATE CASCADE
    ON DELETE CASCADE)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

8.2 Popolazione

```

-- -----
-- POPOLAZIONE DATABASE
-- -----

-- Record CentroFitness
-- -----
INSERT INTO `aziendacentrofitness`.`centrofitness` (`IDCentro`, `Dimensione`,
`Telefono`, `CapienzaMassima`, `Citta`, `Via`, `NumCivico`, `CAP`, `Direttore`)
VALUES ('cf0001', '700', '353698', '100', 'Marsala', 'Garibaldi', '25', '91025',
'd0001');

INSERT INTO `aziendacentrofitness`.`centrofitness` (`IDCentro`, `Dimensione`,
`Telefono`, `CapienzaMassima`, `Citta`, `Via`, `NumCivico`, `CAP`, `Direttore`)
VALUES ('cf0002', '800', '353221', '120', 'Gela', 'Giovanni Bosco', '98',
'93012', 'd0002');

INSERT INTO `aziendacentrofitness`.`centrofitness` (`IDCentro`, `Dimensione`,
`Telefono`, `CapienzaMassima`, `Citta`, `Via`, `NumCivico`, `CAP`, `Direttore`)
VALUES ('cf0003', '900', '845689', '150', 'Pisa', 'Vittorio Veneto', '26',

```

```

'56127', 'd0003');
INSERT INTO `aziendacentrofitness`.`centrofitness` (`IDCentro`, `Dimensione`,
`Telefono`, `CapienzaMassima`, `Citta`, `Via`, `NumCivico`, `CAP`, `Direttore`)
VALUES ('cf0004', '750', '356987', '110', 'Milano', 'Italia', '12', '20155',
'd0004');
INSERT INTO `aziendacentrofitness`.`centrofitness` (`IDCentro`, `Dimensione`,
`Telefono`, `CapienzaMassima`, `Citta`, `Via`, `NumCivico`, `CAP`, `Direttore`)
VALUES ('cf0005', '700', '856932', '100', 'Roma', 'Altino', '140', '00148',
'd0005');
INSERT INTO `aziendacentrofitness`.`centrofitness` (`IDCentro`, `Dimensione`,
`Telefono`, `CapienzaMassima`, `Citta`, `Via`, `NumCivico`, `CAP`, `Direttore`)
VALUES ('cf0006', '725', '125693', '105', 'Palermo', 'Contessa Matilda', '39',
'90125', 'd0006');
-----

-- Record Dipendente
-----
INSERT INTO `aziendacentrofitness`.`dipendente` (`CodFiscaleDip`, `Nome`,
`Cognome`, `DataNascita`, `Citta`, `Via`, `NumCivico`, `CAP`, `Sesso`,
`IDDocumentoRiconoscimento`, `Prefettura`)
VALUES ('d0001', 'Luigi', 'Caradanna', '1985-02-25', 'Marsala', 'Limoni', '45',
'91025', 'M', 'dr0001', 'Marsala');

INSERT INTO `aziendacentrofitness`.`dipendente` (`CodFiscaleDip`, `Nome`,
`Cognome`, `DataNascita`, `Citta`, `Via`, `NumCivico`, `CAP`, `Sesso`,
`IDDocumentoRiconoscimento`, `Prefettura`)
VALUES ('d0002', 'Ernesto', 'Fasulo', '1996-07-04', 'Gela', 'Da Vinci', '85',
'36584', 'M', 'dr0002', 'Gela');

INSERT INTO `aziendacentrofitness`.`dipendente` (`CodFiscaleDip`, `Nome`,
`Cognome`, `DataNascita`, `Citta`, `Via`, `NumCivico`, `CAP`, `Sesso`,
`IDDocumentoRiconoscimento`, `Prefettura`)
VALUES ('d0003', 'Leo', 'Maltese', '1995-02-12', 'Pisa', 'Geronimo', '78',
'65896', 'M', 'dr0003', 'Pisa');

INSERT INTO `aziendacentrofitness`.`dipendente` (`CodFiscaleDip`, `Nome`,
`Cognome`, `DataNascita`, `Citta`, `Via`, `NumCivico`, `CAP`, `Sesso`,
`IDDocumentoRiconoscimento`, `Prefettura`)
VALUES ('d0004', 'Marianna', 'Pulizzi', '1990-11-15', 'Milano', 'Viale', '25',
'63589', 'F', 'dr0004', 'Petrosino');

INSERT INTO `aziendacentrofitness`.`dipendente` (`CodFiscaleDip`, `Nome`,
`Cognome`, `DataNascita`, `Citta`, `Via`, `NumCivico`, `CAP`, `Sesso`,
`IDDocumentoRiconoscimento`, `Prefettura`)
VALUES ('d0005', 'Antonino', 'Maltese', '1962-01-04', 'Roma', 'Quasimodo', '52',
'85697', 'M', 'dr0007', 'Napoli');

INSERT INTO `aziendacentrofitness`.`dipendente` (`CodFiscaleDip`, `Nome`,
`Cognome`, `DataNascita`, `Citta`, `Via`, `NumCivico`, `CAP`, `Sesso`,
`IDDocumentoRiconoscimento`, `Prefettura`)
VALUES ('d0006', 'Giacomo', 'Marino', '1990-04-19', 'Palermo', 'Austria', '11',
'12365', 'M', 'dr0008', 'Palermo');

```

```

INSERT INTO `aziendacentrofitness`.`dipendente` (`CodFiscaleDip`, `Nome`,
`Cognome`, `DataNascita`, `Citta`, `Via`, `NumCivico`, `CAP`, `Sesso`,
`IDDocumentoRiconoscimento`, `Prefettura`, `Responsabile`)
VALUES ('d0007', 'Giovanni', 'Martelli', '1980-03-15', 'Bologna', 'Colli',
'12', '25846', 'M', 'dr0008', 'Bologna', 'd0001');

INSERT INTO `aziendacentrofitness`.`dipendente` (`CodFiscaleDip`, `Nome`,
`Cognome`, `DataNascita`, `Citta`, `Via`, `NumCivico`, `CAP`, `Sesso`,
`IDDocumentoRiconoscimento`, `Prefettura`, `Responsabile`)
VALUES ('d0009', 'Piero', 'Gambina', '1995-07-30', 'Siena', 'Europa', '52',
'45425', 'M', 'dr0009', 'Siena', 'd0001');

INSERT INTO `aziendacentrofitness`.`dipendente` (`CodFiscaleDip`, `Nome`,
`Cognome`, `DataNascita`, `Citta`, `Via`, `NumCivico`, `CAP`, `Sesso`,
`IDDocumentoRiconoscimento`, `Prefettura`, `Responsabile`)
VALUES ('d0010', 'Alessia', 'Paladino', '1990-06-30', 'Latina', 'Latino', '55',
'75488', 'F', 'dr0010', 'Latina', 'd0001');

INSERT INTO `aziendacentrofitness`.`dipendente` (`CodFiscaleDip`, `Nome`,
`Cognome`, `DataNascita`, `Citta`, `Via`, `NumCivico`, `CAP`, `Sesso`,
`IDDocumentoRiconoscimento`, `Prefettura`, `Responsabile`)
VALUES ('d0011', 'Vito', 'De Vita', '1980-12-31', 'Ferrara', 'Ferro', '14',
'47558', 'M', 'dr0011', 'Ferrara', 'd0001');

INSERT INTO `aziendacentrofitness`.`dipendente` (`CodFiscaleDip`, `Nome`,
`Cognome`, `DataNascita`, `Citta`, `Via`, `NumCivico`, `CAP`, `Sesso`,
`IDDocumentoRiconoscimento`, `Prefettura`, `Responsabile`)
VALUES ('d0012', 'Francesco', 'Asaro', '1981-01-01', 'Venezia', 'Fiume', '57',
'42533', 'M', 'dr0012', 'Venezia', 'd0001');

INSERT INTO `aziendacentrofitness`.`dipendente` (`CodFiscaleDip`, `Nome`,
`Cognome`, `DataNascita`, `Citta`, `Via`, `NumCivico`, `CAP`, `Sesso`,
`IDDocumentoRiconoscimento`, `Prefettura`, `Responsabile`)
VALUES ('dipl', 'Piero', 'Giallo', '1991-02-03', 'Ferrara', 'Messina', '12',
'23538', 'M', 'dr0013', 'Ferrara', 'd0011');

INSERT INTO `aziendacentrofitness`.`dipendente` (`CodFiscaleDip`, `Nome`,
`Cognome`, `DataNascita`, `Citta`, `Via`, `NumCivico`, `CAP`, `Sesso`,
`IDDocumentoRiconoscimento`, `Prefettura`, `Responsabile`)
VALUES ('dip2', 'Alberto', 'Nuovo', '1989-03-01', 'Bergamo', 'Palermo', '256',
'12365', 'M', 'dr0014', 'Bergamo', 'd0011');

INSERT INTO `aziendacentrofitness`.`dipendente` (`CodFiscaleDip`, `Nome`,
`Cognome`, `DataNascita`, `Citta`, `Via`, `NumCivico`, `CAP`, `Sesso`,
`IDDocumentoRiconoscimento`, `Prefettura`, `Responsabile`)
VALUES ('dip3', 'Alessia', 'Vecchi', '1986-03-06', 'Venezia', 'Cavour', '12',
'14569', 'F', 'dr0015', 'Venezia', 'd0011');

INSERT INTO `aziendacentrofitness`.`dipendente` (`CodFiscaleDip`, `Nome`,
`Cognome`, `DataNascita`, `Citta`, `Via`, `NumCivico`, `CAP`, `Sesso`,
`IDDocumentoRiconoscimento`, `Prefettura`, `Responsabile`)
VALUES ('dip4', 'Arianna', 'Buoni', '1995-12-31', 'Verona', 'Fiume', '15',
'12356', 'F', 'dr0016', 'Verona', 'd0011');

```

```

-- Record Accesso
-- -----
INSERT INTO `aziendacentrofitness`.`accesso` (`IDAccesso`, `IDCentro`,
`CodFiscaleCliente`, `IDArmadietto`, `DataAccesso`, `OrarioEntrata`,
`OrarioUscita`)
VALUES ('acc0001', 'cf0001', 'cfc0001', 'arm0001', '2005-02-22', '14:30',
'15:30');

INSERT INTO `aziendacentrofitness`.`accesso` (`IDAccesso`, `IDCentro`,
`CodFiscaleCliente`, `IDArmadietto`, `DataAccesso`, `OrarioEntrata`,
`OrarioUscita`)
VALUES ('acc0002', 'cf0001', 'cfc0002', 'arm0002', '2010-12-01', '11:00',
'12:00');

INSERT INTO `aziendacentrofitness`.`accesso` (`IDAccesso`, `IDCentro`,
`CodFiscaleCliente`, `IDArmadietto`, `DataAccesso`, `OrarioEntrata`,
`OrarioUscita`)
VALUES ('acc0003', 'cf0001', 'cfc0003', 'arm0003', '2017-01-03', '19:00',
'20:00');

INSERT INTO `aziendacentrofitness`.`accesso` (`IDAccesso`, `IDCentro`,
`CodFiscaleCliente`, `IDArmadietto`, `DataAccesso`, `OrarioEntrata`,
`OrarioUscita`)
VALUES ('acc0004', 'cf0001', 'cfc0004', 'arm0004', '2016-01-22', '16:00',
'17:30');

INSERT INTO `aziendacentrofitness`.`accesso` (`IDAccesso`, `IDCentro`,
`CodFiscaleCliente`, `IDArmadietto`, `DataAccesso`, `OrarioEntrata`,
`OrarioUscita`)
VALUES ('acc0005', 'cf0001', 'cfc0005', 'arm0006', '2018-01-22', '21:00',
'22:00');

INSERT INTO `aziendacentrofitness`.`accesso` (`IDAccesso`, `IDCentro`,
`CodFiscaleCliente`, `IDArmadietto`, `DataAccesso`, `OrarioEntrata`,
`OrarioUscita`)
VALUES ('acc0006', 'cf0001', 'cfc0001', 'arm0007', '2005-02-25', '14:30:00',
'15:30:00');

INSERT INTO `aziendacentrofitness`.`accesso` (`IDAccesso`, `IDCentro`,
`CodFiscaleCliente`, `IDArmadietto`, `DataAccesso`, `OrarioEntrata`,
`OrarioUscita`)
VALUES ('acc0008', 'cf0001', 'cfc0002', 'arm0008', '2010-12-02', '11:00:00',
'12:00:00');

INSERT INTO `aziendacentrofitness`.`accesso` (`IDAccesso`, `IDCentro`,
`CodFiscaleCliente`, `IDArmadietto`, `DataAccesso`, `OrarioEntrata`,
`OrarioUscita`)
VALUES ('acc0009', 'cf0001', 'cfc0003', 'arm0009', '2017-01-04', '19:00:00',
'20:00:00');

```

```

INSERT INTO `aziendacentrofitness`.`accesso` (`IDAccesso`, `IDCentro`,
`CodFiscaleCliente`, `IDArmadietto`, `DataAccesso`, `OrarioEntrata`,
`OrarioUscita`)
VALUES ('acc0010', 'cf0001', 'cfc0004', 'arm00010', '2016-01-26', '16:00:00',
'17:30:00');

INSERT INTO `aziendacentrofitness`.`accesso` (`IDAccesso`, `IDCentro`,
`CodFiscaleCliente`, `IDArmadietto`, `DataAccesso`, `OrarioEntrata`,
`OrarioUscita`)
VALUES ('acc0011', 'cf0001', 'cfc0005', 'arm00012', '2018-01-29', '21:00:00',
'22:00:00');

-- -----

-- Record Cliente
-- -----

INSERT INTO `aziendacentrofitness`.`cliente` (`CodFiscaleCliente`, `Nome`,
`Cognome`, `DataNascita`, `Citta`, `Via`, `NumCivico`, `CAP`, `Sesso`,
`IDDocumentoRiconoscimento`, `Prefettura`)
VALUES ('cfc0001', 'Leo', 'Maltese', '2001-02-28', 'Marsala', 'Mazzina', '23',
'91025', 'M', 'dr1', 'Marsala');

INSERT INTO `aziendacentrofitness`.`cliente` (`CodFiscaleCliente`, `Nome`,
`Cognome`, `DataNascita`, `Citta`, `Via`, `NumCivico`, `CAP`, `Sesso`,
`IDDocumentoRiconoscimento`, `Prefettura`)
VALUES ('cfc0002', 'Paolo', 'Marchese', '1980-03-01', 'Napoli', 'Asia', '01',
'36548', 'M', 'dr2', 'Napoli');

INSERT INTO `aziendacentrofitness`.`cliente` (`CodFiscaleCliente`, `Nome`,
`Cognome`, `DataNascita`, `Citta`, `Via`, `NumCivico`, `CAP`, `Sesso`,
`IDDocumentoRiconoscimento`, `Prefettura`)
VALUES ('cfc0003', 'Ettore', 'Grece', '1971-01-08', 'Chianti', 'Santa Teresa',
'141', '53017', 'M', 'dr3', 'Chianti');

INSERT INTO `aziendacentrofitness`.`cliente` (`CodFiscaleCliente`, `Nome`,
`Cognome`, `DataNascita`, `Citta`, `Via`, `NumCivico`, `CAP`, `Sesso`,
`IDDocumentoRiconoscimento`, `Prefettura`)
VALUES ('cfc0004', 'Maria', 'Ferrari', '1981-11-23', 'Bari', 'Santa Melania',
'114', '39020', 'F', 'dr4', 'Bari');

INSERT INTO `aziendacentrofitness`.`cliente` (`CodFiscaleCliente`, `Nome`,
`Cognome`, `DataNascita`, `Citta`, `Via`, `NumCivico`, `CAP`, `Sesso`,
`IDDocumentoRiconoscimento`, `Prefettura`)
VALUES ('cfc0005', 'Benito', 'Colombo', '1990-03-12', 'Roma', 'Belviglieri',
'148', '00146', 'M', 'dr5', 'Roma');

INSERT INTO `aziendacentrofitness`.`cliente` (`CodFiscaleCliente`, `Nome`,
`Cognome`, `DataNascita`, `Citta`, `Via`, `NumCivico`, `CAP`, `Sesso`,
`IDDocumentoRiconoscimento`, `Prefettura`)
VALUES ('cfc0006', 'Lara', 'Bellucci', '1993-04-26', 'Ribolla', 'Vittorio
Emanuele', '54', '58027', 'F', 'dr6', 'Ribolla');

-- -----

```

```

-- Record Piscina
-- -----
INSERT INTO `aziendacentrofitness`.`piscina` (`IDPiscina`, `IDCentro`,
`Locazione`, `TipologiaContratto`, `Dimensione`)
VALUES ('p0001', 'cf0001', 'Interna', 'Personalizzato', '100');

INSERT INTO `aziendacentrofitness`.`piscina` (`IDPiscina`, `IDCentro`,
`Locazione`, `TipologiaContratto`, `Dimensione`)
VALUES ('p0002', 'cf0001', 'Interna', 'Platinum', '80');

INSERT INTO `aziendacentrofitness`.`piscina` (`IDPiscina`, `IDCentro`,
`Locazione`, `TipologiaContratto`, `Dimensione`)
VALUES ('p0003', 'cf0001', 'Esterna', 'Gold', '60');

INSERT INTO `aziendacentrofitness`.`piscina` (`IDPiscina`, `IDCentro`,
`Locazione`, `TipologiaContratto`, `Dimensione`)
VALUES ('p0004', 'cf0001', 'Esterna', 'Silver', '50');
-- -----

-- Record Contratto
-- -----
INSERT INTO `aziendacentrofitness`.`contratto` (`IDContratto`, `IDCentro`,
`Cliente`, `Consulente`, `TipologiaContratto`, `AccessiSettimanali`,
`ImportoMensile`, `DurataMensile`, `DataIscrizione`, `Scopo`)
VALUES ('con001', 'cf0001', 'cfc0001', 'd0007', 'Gold', '2', '50', '6', '2002-
01-25', 'Dimagrimento');

INSERT INTO `aziendacentrofitness`.`contratto` (`IDContratto`, `IDCentro`,
`Cliente`, `Consulente`, `TipologiaContratto`, `AccessiSettimanali`,
`ImportoMensile`, `DurataMensile`, `DataIscrizione`, `Scopo`)
VALUES ('con002', 'cf0001', 'cfc0002', 'd0007', 'Silver', '1', '20', '9',
'2003-06-23', 'Potenziamento Muscolare');

INSERT INTO `aziendacentrofitness`.`contratto` (`IDContratto`, `IDCentro`,
`Cliente`, `Consulente`, `TipologiaContratto`, `AccessiSettimanali`,
`ImportoMensile`, `DurataMensile`, `DataIscrizione`, `Scopo`)
VALUES ('con003', 'cf0001', 'cfc0003', 'd0007', 'Platinum', '3', '65', '3',
'2010-08-31', 'Ricreativo');

INSERT INTO `aziendacentrofitness`.`contratto` (`IDContratto`, `IDCentro`,
`Cliente`, `Consulente`, `TipologiaContratto`, `ImportoMensile`,
`DurataMensile`, `DataIscrizione`, `Scopo`)
VALUES ('con004', 'cf0001', 'cfc0004', 'd0007', 'Personalizzato', '100', '3',
'2011-09-20', 'Potenziamento Muscolare');

INSERT INTO `aziendacentrofitness`.`contratto` (`IDContratto`, `IDCentro`,
`Cliente`, `Consulente`, `TipologiaContratto`, `AccessiSettimanali`,
`ImportoMensile`, `DurataMensile`, `DataIscrizione`, `Scopo`)
VALUES ('con005', 'cf0001', 'cfc0005', 'd0007', 'Gold', '2', '50', '3', '2012-
12-21', 'Ricreativo');
-- -----

```

```

-- Record Esercizio
-- -----
INSERT INTO `aziendacentrofitness`.`esercizio` (`IDEsercizio`, `Nome`,
`DispendioEnergeticoMedio`)
VALUES ('e0001', 'Addominali', '20');

INSERT INTO `aziendacentrofitness`.`esercizio` (`IDEsercizio`, `Nome`,
`DispendioEnergeticoMedio`)
VALUES ('e0002', 'Trazioni', '50');

INSERT INTO `aziendacentrofitness`.`esercizio` (`IDEsercizio`, `Nome`,
`DispendioEnergeticoMedio`)
VALUES ('e0003', 'Corsa', '10');

INSERT INTO `aziendacentrofitness`.`esercizio` (`IDEsercizio`, `Nome`,
`DispendioEnergeticoMedio`)
VALUES ('e0004', 'Cyclette', '10');

INSERT INTO `aziendacentrofitness`.`esercizio` (`IDEsercizio`, `Nome`,
`DispendioEnergeticoMedio`)
VALUES ('e0005', 'Lateral Pulley', '30');

INSERT INTO `aziendacentrofitness`.`esercizio` (`IDEsercizio`, `Nome`,
`DispendioEnergeticoMedio`)
VALUES ('e0006', 'Alzate Laterali', '20');

INSERT INTO `aziendacentrofitness`.`esercizio` (`IDEsercizio`, `Nome`,
`DispendioEnergeticoMedio`)
VALUES ('e0007', 'Squat', '15');

INSERT INTO `aziendacentrofitness`.`esercizio` (`IDEsercizio`, `Nome`,
`DispendioEnergeticoMedio`)
VALUES ('e0008', 'Affondi Laterali', '30');

INSERT INTO `aziendacentrofitness`.`esercizio` (`IDEsercizio`, `Nome`,
`DispendioEnergeticoMedio`)
VALUES ('e0009', 'Curl con Manubrio', '40');

INSERT INTO `aziendacentrofitness`.`esercizio` (`IDEsercizio`, `Nome`,
`DispendioEnergeticoMedio`)
VALUES ('e0010', 'Flessioni', '35');

INSERT INTO `aziendacentrofitness`.`esercizio` (`IDEsercizio`, `Nome`,
`DispendioEnergeticoMedio`)
VALUES ('e0011', 'Push Down ai Cavi', '20');

INSERT INTO `aziendacentrofitness`.`esercizio` (`IDEsercizio`, `Nome`,
`DispendioEnergeticoMedio`)
VALUES ('e0012', 'Distensioni a Presa Stretta', '25');

INSERT INTO `aziendacentrofitness`.`esercizio` (`IDEsercizio`, `Nome`,
`DispendioEnergeticoMedio`)
VALUES ('e0013', 'Dorsali', '35');

```



```

-- -----
-- Record Aerobico
-- -----
INSERT INTO `aziendacentrofitness`.`aerobico` (`IDESercizioAerobico`,
`DurataMinuti`)
VALUES ('e0003', '15');

INSERT INTO `aziendacentrofitness`.`aerobico` (`IDESercizioAerobico`,
`DurataMinuti`)
VALUES ('e0004', '20');
-- -----

-- Record Armadietto
-- -----
INSERT INTO `aziendacentrofitness`.`armadietto` (`IDArmadietto`,
`IDSpogliatoio`, `CombinazioneNumerica`)
VALUES ('arm0001', 'spo0001', '00000');

INSERT INTO `aziendacentrofitness`.`armadietto` (`IDArmadietto`,
`IDSpogliatoio`, `CombinazioneNumerica`)
VALUES ('arm0002', 'spo0001', '00001');

INSERT INTO `aziendacentrofitness`.`armadietto` (`IDArmadietto`,
`IDSpogliatoio`, `CombinazioneNumerica`)
VALUES ('arm0003', 'spo0001', '00002');

INSERT INTO `aziendacentrofitness`.`armadietto` (`IDArmadietto`,
`IDSpogliatoio`, `CombinazioneNumerica`)
VALUES ('arm0004', 'spo0002', '00003');

INSERT INTO `aziendacentrofitness`.`armadietto` (`IDArmadietto`,
`IDSpogliatoio`, `CombinazioneNumerica`)
VALUES ('arm0005', 'spo0002', '00004');

INSERT INTO `aziendacentrofitness`.`armadietto` (`IDArmadietto`,
`IDSpogliatoio`, `CombinazioneNumerica`)
VALUES ('arm0006', 'spo0002', '00005');

INSERT INTO `aziendacentrofitness`.`armadietto` (`IDArmadietto`,
`IDSpogliatoio`, `CombinazioneNumerica`)
VALUES ('arm0007', 'spo0003', '00006');

INSERT INTO `aziendacentrofitness`.`armadietto` (`IDArmadietto`,
`IDSpogliatoio`, `CombinazioneNumerica`)
VALUES ('arm0008', 'spo0003', '00007');

INSERT INTO `aziendacentrofitness`.`armadietto` (`IDArmadietto`,
`IDSpogliatoio`, `CombinazioneNumerica`)
VALUES ('arm0009', 'spo0003', '00008');

```

```

INSERT INTO `aziendacentrofitness`.`armadietto` (`IDArmadietto`,
`IDSpogliatoio`, `CombinazioneNumerica`)
VALUES ('arm00010', 'spo0004', '00009');

INSERT INTO `aziendacentrofitness`.`armadietto` (`IDArmadietto`,
`IDSpogliatoio`, `CombinazioneNumerica`)
VALUES ('arm00011', 'spo0004', '00010');

INSERT INTO `aziendacentrofitness`.`armadietto` (`IDArmadietto`,
`IDSpogliatoio`, `CombinazioneNumerica`)
VALUES ('arm00012', 'spo0004', '00011');
-- -----

-- Record Spogliatoio
-- -----
INSERT INTO `aziendacentrofitness`.`spogliatoio` (`IDSpogliatoio`, `Capienza`,
`PostiDisponibili`, `Posizione`, `Sesso`)
VALUES ('spo0001', '50', '50', 'N', 'M');

INSERT INTO `aziendacentrofitness`.`spogliatoio` (`IDSpogliatoio`, `Capienza`,
`PostiDisponibili`, `Posizione`, `Sesso`)
VALUES ('spo0002', '50', '50', 'S', 'F');

INSERT INTO `aziendacentrofitness`.`spogliatoio` (`IDSpogliatoio`, `Capienza`,
`PostiDisponibili`, `Posizione`, `Sesso`)
VALUES ('spo0003', '50', '50', 'E', 'M');

INSERT INTO `aziendacentrofitness`.`spogliatoio` (`IDSpogliatoio`, `Capienza`,
`PostiDisponibili`, `Posizione`, `Sesso`)
VALUES ('spo0004', '50', '50', 'O', 'F');
-- -----
---

-- Record Sala
-- -----
--
INSERT INTO `aziendacentrofitness`.`sala` (`IDSala`, `IDCentro`, `Nome`,
`TipologiaContratto`, `ResponsabileSala`)
VALUES ('sal0001', 'cf0001', 'Cardio Fitness', 'Silver', 'd0009');

INSERT INTO `aziendacentrofitness`.`sala` (`IDSala`, `IDCentro`, `Nome`,
`TipologiaContratto`, `ResponsabileSala`)
VALUES ('sal0002', 'cf0001', 'Bodybuilding', 'Gold', 'd0010');

INSERT INTO `aziendacentrofitness`.`sala` (`IDSala`, `IDCentro`, `Nome`,
`TipologiaContratto`, `ResponsabileSala`)
VALUES ('sal0003', 'cf0001', 'Aerobica', 'Platinum', 'd0011');

INSERT INTO `aziendacentrofitness`.`sala` (`IDSala`, `IDCentro`, `Nome`,
`TipologiaContratto`, `ResponsabileSala`)
VALUES ('sal0004', 'cf0001', 'Corpo Libero', 'Personalizzato', 'd0012');

```

```

INSERT INTO `aziendacentrofitness`.`sala` (`IDSala`, `IDCentro`, `Nome`,
`TipologiaContratto`, `ResponsabileSala`)
VALUES ('sal0005', 'cf0001', 'Cardio Fitness', 'Gold', 'd0010');

INSERT INTO `aziendacentrofitness`.`sala` (`IDSala`, `IDCentro`, `Nome`,
`TipologiaContratto`, `ResponsabileSala`)
VALUES ('sal0006', 'cf0001', 'Bodybuilding', 'Silver', 'd0009');

INSERT INTO `aziendacentrofitness`.`sala` (`IDSala`, `IDCentro`, `Nome`,
`TipologiaContratto`, `ResponsabileSala`)
VALUES ('sal0007', 'cf0001', 'Cardio Fitness', 'Platinum', 'd0011');

INSERT INTO `aziendacentrofitness`.`sala` (`IDSala`, `IDCentro`, `Nome`,
`TipologiaContratto`, `ResponsabileSala`)
VALUES ('sal0008', 'cf0001', 'Cardio Fitness', 'Personalizzato', 'd0012');

INSERT INTO `aziendacentrofitness`.`sala` (`IDSala`, `IDCentro`, `Nome`,
`TipologiaContratto`, `ResponsabileSala`)
VALUES ('sal0009', 'cf0001', 'Bodybuilding', 'Platinum', 'd0011');

INSERT INTO `aziendacentrofitness`.`sala` (`IDSala`, `IDCentro`, `Nome`,
`TipologiaContratto`, `ResponsabileSala`)
VALUES ('sal0010', 'cf0001', 'Bodybuilding', 'Personalizzato', 'd0012');

INSERT INTO `aziendacentrofitness`.`sala` (`IDSala`, `IDCentro`, `Nome`,
`TipologiaContratto`, `ResponsabileSala`)
VALUES ('sal0011', 'cf0001', 'Aerobica', 'Silver', 'd0009');

INSERT INTO `aziendacentrofitness`.`sala` (`IDSala`, `IDCentro`, `Nome`,
`TipologiaContratto`, `ResponsabileSala`)
VALUES ('sal0012', 'cf0001', 'Aerobica', 'Gold', 'd0010');

INSERT INTO `aziendacentrofitness`.`sala` (`IDSala`, `IDCentro`, `Nome`,
`TipologiaContratto`, `ResponsabileSala`)
VALUES ('sal0013', 'cf0001', 'Aerobica', 'Personalizzato', 'd0012');

INSERT INTO `aziendacentrofitness`.`sala` (`IDSala`, `IDCentro`, `Nome`,
`TipologiaContratto`, `ResponsabileSala`)
VALUES ('sal0014', 'cf0001', 'Corpo Libero', 'Silver', 'd0009');

INSERT INTO `aziendacentrofitness`.`sala` (`IDSala`, `IDCentro`, `Nome`,
`TipologiaContratto`, `ResponsabileSala`)
VALUES ('sal0015', 'cf0001', 'Corpo Libero', 'Gold', 'd0010');

INSERT INTO `aziendacentrofitness`.`sala` (`IDSala`, `IDCentro`, `Nome`,
`TipologiaContratto`, `ResponsabileSala`)
VALUES ('sal0016', 'cf0001', 'Corpo Libero', 'Platinum', 'd0011');

```

```

-- -----
-

```

```

-- Record MisurazioniImpedenziometriche
-- -----
---
INSERT INTO `aziendacentrofitness`.`misurazioniimpedenziometriche`
(`IDMisurazione`, `CodFiscaleCliente`, `DataMisurazione`, `AcquaTotale`,
`MassaMagra`, `MassaGrassa`, `Peso`, `Altezza`)
VALUES ('mis0001', 'cfc0001', '2010-02-05', '80', '50', '80', '65', '1.70');

INSERT INTO `aziendacentrofitness`.`misurazioniimpedenziometriche`
(`IDMisurazione`, `CodFiscaleCliente`, `DataMisurazione`, `AcquaTotale`,
`MassaMagra`, `MassaGrassa`, `Peso`, `Altezza`)
VALUES ('mis0002', 'cfc0001', '2010-02-15', '80', '50', '20', '80', '1.70');
-- -----
----

-- Record Interesse
-- -----
-----
INSERT INTO `aziendacentrofitness`.`interesse` (`NomeInteresse`)
VALUES ('Aerobica');

INSERT INTO `aziendacentrofitness`.`interesse` (`NomeInteresse`)
VALUES ('Yoga');

INSERT INTO `aziendacentrofitness`.`interesse` (`NomeInteresse`)
VALUES
('Bodybuilding');

INSERT INTO `aziendacentrofitness`.`interesse` (`NomeInteresse`)
VALUES ('Corsa');

INSERT INTO `aziendacentrofitness`.`interesse` (`NomeInteresse`)
VALUES ('Calcio');

INSERT INTO `aziendacentrofitness`.`interesse` (`NomeInteresse`)
VALUES ('Kickbox');

INSERT INTO `aziendacentrofitness`.`interesse` (`NomeInteresse`)
VALUES ('Karate');

INSERT INTO `aziendacentrofitness`.`interesse` (`NomeInteresse`)
VALUES ('Pallavolo');

INSERT INTO `aziendacentrofitness`.`interesse` (`NomeInteresse`)
VALUES ('Basket');

INSERT INTO `aziendacentrofitness`.`interesse` (`NomeInteresse`)
VALUES ('Baseball');

INSERT INTO `aziendacentrofitness`.`interesse` (`NomeInteresse`)
VALUES ('Rugby');

INSERT INTO `aziendacentrofitness`.`interesse` (`NomeInteresse`)
VALUES ('Nuoto');

```

```
INSERT INTO `aziendacentrofitness`.`interesse` (`NomeInteresse`)
VALUES ('Ciclismo');
```

```
INSERT INTO `aziendacentrofitness`.`interesse` (`NomeInteresse`)
VALUES ('Arrampicata Sportiva');
```

```
INSERT INTO `aziendacentrofitness`.`interesse` (`NomeInteresse`)
VALUES ('Ginnastica Artistica');
```

```
INSERT INTO `aziendacentrofitness`.`interesse` (`NomeInteresse`)
VALUES ('Tennis');
```

```
INSERT INTO `aziendacentrofitness`.`interesse` (`NomeInteresse`)
VALUES ('Tennis da Tavolo');
```

```
INSERT INTO `aziendacentrofitness`.`interesse` (`NomeInteresse`)
VALUES ('Tuffi');
```

```
INSERT INTO `aziendacentrofitness`.`interesse` (`NomeInteresse`)
VALUES ('Atletica Leggera');
```

```
INSERT INTO `aziendacentrofitness`.`interesse` (`NomeInteresse`)
VALUES ('Kitesurf');
```

```
INSERT INTO `aziendacentrofitness`.`interesse` (`NomeInteresse`)
VALUES ('Zumba');
```

```
-- Record Utente
```

```
INSERT INTO `aziendacentrofitness`.`utente` (`Username`, `IDContratto`,
`Password`)
VALUES ('Leo95', 'con001', 'ciao1');
```

```
INSERT INTO `aziendacentrofitness`.`utente` (`Username`, `IDContratto`,
`Password`)
VALUES ('erne96', 'con002', 'ciao2');
```

```
INSERT INTO `aziendacentrofitness`.`utente` (`Username`, `IDContratto`,
`Password`)
VALUES ('mari90', 'con003', 'ciao3');
```

```
INSERT INTO `aziendacentrofitness`.`utente` (`Username`, `IDContratto`,
`Password`)
VALUES ('ele89', 'con004', 'ciao4');
```

```
INSERT INTO `aziendacentrofitness`.`utente` (`Username`, `IDContratto`,
`Password`)
VALUES ('pi95', 'con005', 'ciao4');
```

-- InteressiUtente

```
-----  
-----  
INSERT INTO `aziendacentrofitness`.`interessiutente` (`Username`,  
`NomeInteresse`)  
VALUES ('Leo95', 'Calcio');
```

```
INSERT INTO `aziendacentrofitness`.`interessiutente` (`Username`,  
`NomeInteresse`)  
VALUES ('Leo95', 'Tennis');
```

```
INSERT INTO `aziendacentrofitness`.`interessiutente` (`Username`,  
`NomeInteresse`)  
VALUES ('ele89', 'Yoga');
```

```
INSERT INTO `aziendacentrofitness`.`interessiutente` (`Username`,  
`NomeInteresse`)  
VALUES ('ele89', 'Ginnastica Artistica');
```

```
INSERT INTO `aziendacentrofitness`.`interessiutente` (`Username`,  
`NomeInteresse`)  
VALUES ('mari90', 'Zumba');
```

```
INSERT INTO `aziendacentrofitness`.`interessiutente` (`Username`,  
`NomeInteresse`)  
VALUES ('erne90', 'Calcio');
```

```
INSERT INTO `aziendacentrofitness`.`interessiutente` (`Username`,  
`NomeInteresse`)  
VALUES ('erne90', 'Bodybuilding');
```

-- Record FasciMuscolari

```
-----  
-----  
INSERT INTO `aziendacentrofitness`.`fascimuscolari` (`NomeFasciMuscolari`,  
`Livello`)  
VALUES ('Arti Superiori', 'lieve');
```

```
INSERT INTO `aziendacentrofitness`.`fascimuscolari` (`NomeFasciMuscolari`,  
`Livello`)  
VALUES ('Arti Superiori', 'moderato');
```

```
INSERT INTO `aziendacentrofitness`.`fascimuscolari` (`NomeFasciMuscolari`,  
`Livello`)  
VALUES ('Arti Superiori', 'elevato');
```

```
INSERT INTO `aziendacentrofitness`.`fascimuscolari` (`NomeFasciMuscolari`,  
`Livello`)  
VALUES ('Arti Inferiori', 'lieve');
```

```
INSERT INTO `aziendacentrofitness`.`fascimuscolari` (`NomeFasciMuscolari`,
`Livello`)
VALUES ('Arti Inferiori', 'moderato');
```

```
INSERT INTO `aziendacentrofitness`.`fascimuscolari` (`NomeFasciMuscolari`,
`Livello`)
VALUES ('Arti Inferiori', 'elevato');
```

```
INSERT INTO `aziendacentrofitness`.`fascimuscolari` (`NomeFasciMuscolari`,
`Livello`)
VALUES ('Schiena', 'lieve');
```

```
INSERT INTO `aziendacentrofitness`.`fascimuscolari` (`NomeFasciMuscolari`,
`Livello`)
VALUES ('Schiena', 'moderato');
```

```
INSERT INTO `aziendacentrofitness`.`fascimuscolari` (`NomeFasciMuscolari`,
`Livello`)
VALUES ('Schiena', 'elevato');
```

```
INSERT INTO `aziendacentrofitness`.`fascimuscolari` (`NomeFasciMuscolari`,
`Livello`)
VALUES ('Petto', 'lieve');
```

```
INSERT INTO `aziendacentrofitness`.`fascimuscolari` (`NomeFasciMuscolari`,
`Livello`)
VALUES ('Petto', 'moderato');
```

```
INSERT INTO `aziendacentrofitness`.`fascimuscolari` (`NomeFasciMuscolari`,
`Livello`)
VALUES ('Petto', 'elevato');
```

```
INSERT INTO `aziendacentrofitness`.`fascimuscolari` (`NomeFasciMuscolari`,
`Livello`)
VALUES ('Addome', 'elevato');
```

```
INSERT INTO `aziendacentrofitness`.`fascimuscolari` (`NomeFasciMuscolari`,
`Livello`)
VALUES ('Addome', 'lieve');
```

```
INSERT INTO `aziendacentrofitness`.`fascimuscolari` (`NomeFasciMuscolari`,
`Livello`)
VALUES ('Addome', 'moderato');
```

```
-----
-----
```

```
-- Record Potenziamiento
```

```
-----
-----
```

```
INSERT INTO `aziendacentrofitness`.`potenziamento` (`IDContratto`,
`NomeFasciMuscolari`, `Livello`)
VALUES ('con002', 'Arti Superiori', 'elevato');
```

```

INSERT INTO `aziendacentrofitness`.`potenziamento` (`IDContratto`,
`NomeFasciMuscolari`, `Livello`)
VALUES ('con002', 'Addome', 'moderato');

INSERT INTO `aziendacentrofitness`.`potenziamento` (`IDContratto`,
`NomeFasciMuscolari`, `Livello`)
VALUES ('con004', 'Arti Inferiori', 'Elevato');

INSERT INTO `aziendacentrofitness`.`potenziamento` (`IDContratto`,
`NomeFasciMuscolari`, `Livello`)
VALUES ('con004', 'Schiena', 'lieve');

INSERT INTO `aziendacentrofitness`.`potenziamento` (`IDContratto`,
`NomeFasciMuscolari`, `Livello`)
VALUES ('con004', 'Arti Superioi', 'moderato');
-----

-- Record Dieta
-----

INSERT INTO `aziendacentrofitness`.`dieta` (`IDDieta`, `ApportoCalorico`,
`NumeroPasti`)
VALUES ('dieta1', '2000', '5');

INSERT INTO `aziendacentrofitness`.`dieta` (`IDDieta`, `ApportoCalorico`,
`NumeroPasti`)
VALUES ('dieta2', '1500', '3');

INSERT INTO `aziendacentrofitness`.`dieta` (`IDDieta`, `ApportoCalorico`,
`NumeroPasti`)
VALUES ('dieta3', '1900', '4');

INSERT INTO `aziendacentrofitness`.`dieta` (`IDDieta`, `ApportoCalorico`,
`NumeroPasti`)
VALUES ('dieta4', '2500', '5');

INSERT INTO `aziendacentrofitness`.`dieta` (`IDDieta`, `ApportoCalorico`,
`NumeroPasti`)
VALUES ('dieta5', '1000', '2');
-----

-- Record SchedaAlimentazione
-----

INSERT INTO `aziendacentrofitness`.`schedaalimentazione`
(`IDSchedaAlimentazione`, `CodFiscaleCliente`, `IDDieta`, `DataInizio`,
`DataFine`, `MedicoScheda`, `Obiettivo`)
VALUES ('alil1', 'cfc0001', 'dieta2', '2010-02-02', '2010-04-02', 'med1',
'Dimagrimento');

```



```
INSERT INTO `aziendacentrofitness`.`schedaalimentazione`
(`IDSchedaAlimentazione`, `CodFiscaleCliente`, `IDDieta`, `DataInizio`,
`DataFine`, `MedicoScheda`, `Obiettivo`)
VALUES ('ali2', 'cfc0002', 'dieta1', '2011-03-11', '2011-05-29', 'med2',
'Acquisire massa magra');
```

```
INSERT INTO `aziendacentrofitness`.`schedaalimentazione`
(`IDSchedaAlimentazione`, `CodFiscaleCliente`, `IDDieta`, `DataInizio`,
`DataFine`, `MedicoScheda`, `Obiettivo`)
VALUES ('ali3', 'cfc0003', 'dieta3', '2012-12-30', '2013-02-28', 'med3',
'Diminuire massa grassa');
```

```
INSERT INTO `aziendacentrofitness`.`schedaalimentazione`
(`IDSchedaAlimentazione`, `CodFiscaleCliente`, `IDDieta`, `DataInizio`,
`DataFine`, `MedicoScheda`, `Obiettivo`)
VALUES ('ali4', 'cfc0004', 'dieta4',
'2017-04-01', '2017-05-02', 'med4', 'Acquisire massa magra');
```

```
INSERT INTO `aziendacentrofitness`.`schedaalimentazione`
(`IDSchedaAlimentazione`, `CodFiscaleCliente`, `IDDieta`, `DataInizio`,
`DataFine`, `MedicoScheda`, `Obiettivo`)
VALUES ('ali5', 'cfc0005', 'dieta5', '2016-03-11', '2016-11-30', 'med5',
'Perdere Peso');
```

```
INSERT INTO `aziendacentrofitness`.`schedaalimentazione`
(`IDSchedaAlimentazione`, `CodFiscaleCliente`, `IDDieta`, `DataInizio`,
`DataFine`, `MedicoScheda`, `Obiettivo`)
VALUES ('ali6', 'cfc0006', 'dieta3', '2017-09-22', '2017-12-22', 'med6',
'Perdere Peso');
```

```
-- Record Pietanza
```

```
INSERT INTO `aziendacentrofitness`.`pietanza` (`NomePietanza`,
`ComposizionePietanza`)
VALUES ('Pasta al Pomodoro', 'Pasta, Passato Pomodoro, Basico');
```

```
INSERT INTO `aziendacentrofitness`.`pietanza` (`NomePietanza`,
`ComposizionePietanza`)
VALUES ('Latte e Biscotti', 'Latte, Biscotti Integrali');
```

```
INSERT INTO `aziendacentrofitness`.`pietanza` (`NomePietanza`,
`ComposizionePietanza`)
VALUES ('Pollo Arrostito', 'Pollo, Olio, Insalata');
```

```
INSERT INTO `aziendacentrofitness`.`pietanza` (`NomePietanza`,
`ComposizionePietanza`)
VALUES ('Riso in Bianco', 'Riso, Olio, Parmigiano');
```

```

INSERT INTO `aziendacentrofitness`.`pietanza` (`NomePietanza`,
`ComposizionePietanza`)
VALUES ('Riso al curry', 'Riso, Olio, Curry, Spezie a piacere');

INSERT INTO `aziendacentrofitness`.`pietanza` (`NomePietanza`,
`ComposizionePietanza`)
VALUES ('Cotoletta alla Palermitana', 'Carne Pollo, Pangrattato,');

INSERT INTO `aziendacentrofitness`.`pietanza` (`NomePietanza`,
`ComposizionePietanza`)
VALUES ('Pasta al Pesto', 'Pasta Penne, Pesto di basilico');

INSERT INTO `aziendacentrofitness`.`pietanza` (`NomePietanza`,
`ComposizionePietanza`)
VALUES ('Carne e contorno', 'Fettina di vitello, Funghi');

INSERT INTO `aziendacentrofitness`.`pietanza` (`NomePietanza`,
`ComposizionePietanza`)
VALUES ('Spuntino 1', 'Pancarrè, Marmellata');

INSERT INTO `aziendacentrofitness`.`pietanza` (`NomePietanza`,
`ComposizionePietanza`)
VALUES ('Spuntino 2', 'Pancarrè integrale, Fesa di tacchino');

INSERT INTO `aziendacentrofitness`.`pietanza` (`NomePietanza`,
`ComposizionePietanza`)
VALUES ('Parmiggiana al forno', 'Melanzana, Olio, Salsa al pomodoro, Basilico');

INSERT INTO `aziendacentrofitness`.`pietanza` (`NomePietanza`,
`ComposizionePietanza`)
VALUES ('Bresaola e contorno', 'Bresaola, Morrarella, Olio');

INSERT INTO `aziendacentrofitness`.`pietanza` (`NomePietanza`,
`ComposizionePietanza`)
VALUES ('Riso e lenticchie', 'Riso Basmati, Lenticchie ');

INSERT INTO `aziendacentrofitness`.`pietanza` (`NomePietanza`,
`ComposizionePietanza`)
VALUES ('Latte e cereali', 'Latte, Cereali integrali');

INSERT INTO `aziendacentrofitness`.`pietanza` (`NomePietanza`,
`ComposizionePietanza`)
VALUES ('The e Fette Biscottate', 'Acqua calda, Bustina the verde, Fette
Biscottate Integrali');

-----

-- Record ComposizioneDieta
-----

INSERT INTO `aziendacentrofitness`.`composizionedieta` (`IDDieta`,
`NomePietanza`, `Pasto`)
VALUES ('dietal', 'Spuntino1', 'Spuntino Mattino');

```

```

INSERT INTO `aziendacentrofitness`.`composizionedieta` (`IDDieta`,
`NomePietanza`, `Pasto`)
VALUES ('dieta1', 'Spuntino 2', 'Spuntino Pomeridiano');

INSERT INTO `aziendacentrofitness`.`composizionedieta` (`IDDieta`,
`NomePietanza`, `Pasto`)
VALUES ('dieta1', 'Pasta al Pomodoro', 'Pranzo');

INSERT INTO `aziendacentrofitness`.`composizionedieta` (`IDDieta`,
`NomePietanza`, `Pasto`)
VALUES ('dieta1', 'Carne e contorno', 'Cena');

INSERT INTO `aziendacentrofitness`.`composizionedieta` (`IDDieta`,
`NomePietanza`, `Pasto`)
VALUES ('dieta1', 'Latte e Biscotti', 'Colazione');

INSERT INTO `aziendacentrofitness`.`composizionedieta` (`IDDieta`,
`NomePietanza`, `Pasto`)
VALUES ('dieta2', 'Latte e Biscotti', 'Colazione');

INSERT INTO `aziendacentrofitness`.`composizionedieta` (`IDDieta`,
`NomePietanza`, `Pasto`)
VALUES ('dieta2', 'Riso e lenticchie', 'Pranzo');

INSERT INTO `aziendacentrofitness`.`composizionedieta` (`IDDieta`,
`NomePietanza`, `Pasto`)
VALUES ('dieta2', 'Cotoletta alla palermitana', 'Cena');

INSERT INTO `aziendacentrofitness`.`composizionedieta` (`IDDieta`,
`NomePietanza`, `Pasto`)
VALUES ('dieta3', 'Latte e cereali', 'Colazione');

INSERT INTO `aziendacentrofitness`.`composizionedieta` (`IDDieta`,
`NomePietanza`, `Pasto`)
VALUES ('dieta3', 'Riso al Curry', 'Pranzo');

INSERT INTO `aziendacentrofitness`.`composizionedieta` (`IDDieta`,
`NomePietanza`, `Pasto`)
VALUES ('dieta3', 'Spuntino 2', 'Spuntino Pomeridiano');

INSERT INTO `aziendacentrofitness`.`composizionedieta` (`IDDieta`,
`NomePietanza`, `Pasto`)
VALUES ('dieta3', 'Vitello e contorno', 'Cena');

INSERT INTO `aziendacentrofitness`.`composizionedieta` (`IDDieta`,
`NomePietanza`, `Pasto`)
VALUES ('dieta4', 'Spuntino 1', 'Spuntino Mattino');

INSERT INTO `aziendacentrofitness`.`composizionedieta` (`IDDieta`,
`NomePietanza`, `Pasto`)
VALUES ('dieta4', 'Spuntino 2', 'Spuntino Pomeridiano');

```

```

INSERT INTO `aziendacentrofitness`.`composizionedieta` (`IDDieta`,
`NomePietanza`, `Pasto`)
VALUES ('dieta4', 'The e Fette Biscottate', 'Colazione');

INSERT INTO `aziendacentrofitness`.`composizionedieta` (`IDDieta`,
`NomePietanza`, `Pasto`)
VALUES ('dieta4', 'Riso in Bianco', 'Pranzo');

INSERT INTO `aziendacentrofitness`.`composizionedieta` (`IDDieta`,
`NomePietanza`, `Pasto`)
VALUES ('dieta4', 'Bresaola e contorno', 'Cena');

INSERT INTO `aziendacentrofitness`.`composizionedieta` (`IDDieta`,
`NomePietanza`, `Pasto`)
VALUES ('dieta5', 'Riso e lenticchie', 'Pranzo');

INSERT INTO `aziendacentrofitness`.`composizionedieta` (`IDDieta`,
`NomePietanza`, `Pasto`)
VALUES ('dieta5', 'Parmiggiana al forno', 'Cena');
-----

-- Record Amico
-----
INSERT INTO `aziendacentrofitness`.`amico` (`UsernameAmico`, `Username`)
VALUES ('mari90', 'Leo95');

INSERT INTO `aziendacentrofitness`.`amico` (`UsernameAmico`, `Username`)
VALUES ('ele89', 'Leo95');

INSERT INTO `aziendacentrofitness`.`amico` (`UsernameAmico`, `Username`)
VALUES ('erne96', 'Leo95');
-----

-- Record Cerchia
-----
INSERT INTO `aziendacentrofitness`.`cerchia` (`NomeCerchia`, `Argomento`)
VALUES ('I Love Calcio', 'Calcio');

INSERT INTO `aziendacentrofitness`.`cerchia` (`NomeCerchia`, `Argomento`)
VALUES ('I Love Ciclismo', 'Ciclismo');
-----

```

-- Record Appartenenza

```
-----  
-----  
INSERT INTO `aziendacentrofitness`.`appartenenza` (`NomeCerchia`,  
`UsernameAmico`, `Username`)  
VALUES ('I Love Calcio', 'erne90', 'Leo95');
```

```
INSERT INTO `aziendacentrofitness`.`appartenenza` (`NomeCerchia`,  
`UsernameAmico`, `Username`)  
VALUES ('I Love Ciclismo', 'mari90', 'Leo95');
```

```
INSERT INTO `aziendacentrofitness`.`appartenenza` (`NomeCerchia`,  
`UsernameAmico`, `Username`)  
VALUES ('I Love Ciclismo', 'ele89', 'Leo95');
```

-- Record SuggerimentoInserimento

```
-----  
-----  
INSERT INTO `aziendacentrofitness`.`suggerimentoinserimento` (`NomeCerchia`,  
`UsernameAmico`, `Username`)  
VALUES ('I Love Ciclismo', 'erne90', 'Leo95');
```

-- Record Anaerobico

```
-----  
-----  
INSERT INTO `aziendacentrofitness`.`anaerobico` (`IDESercizioAnaerobico`,  
`NomeAnaerobico`, `NumEsecuzioni`, `TempoDiRecupero`)  
VALUES ('e0001', 'Addominali1', '15', '20');
```

```
INSERT INTO `aziendacentrofitness`.`anaerobico` (`IDESercizioAnaerobico`,  
`NomeAnaerobico`, `NumEsecuzioni`, `TempoDiRecupero`)  
VALUES ('e0001', 'Addominali2', '20', '40');  
INSERT INTO `aziendacentrofitness`.`anaerobico` (`IDESercizioAnaerobico`,  
`NomeAnaerobico`, `NumEsecuzioni`, `TempoDiRecupero`)  
VALUES ('e0001', 'Addominali3', '30', '60');
```

```
INSERT INTO `aziendacentrofitness`.`anaerobico` (`IDESercizioAnaerobico`,  
`NomeAnaerobico`, `NumEsecuzioni`, `TempoDiRecupero`)  
VALUES ('e0006', 'Alzate Lateralil', '20', '40');  
INSERT INTO `aziendacentrofitness`.`anaerobico` (`IDESercizioAnaerobico`,  
`NomeAnaerobico`, `NumEsecuzioni`, `TempoDiRecupero`)  
VALUES ('e0006', 'Alzate Lateralil2', '25', '60');
```

```
INSERT INTO `aziendacentrofitness`.`anaerobico` (`IDESercizioAnaerobico`,  
`NomeAnaerobico`, `NumEsecuzioni`, `TempoDiRecupero`)  
VALUES ('e0006', 'Alzate Lateralil3', '30', '90');
```

```
INSERT INTO `aziendacentrofitness`.`anaerobico` (`IDEsercizioAnaerobico`,
`NomeAnaerobico`, `NumEsecuzioni`, `TempoDiRecupero`)
VALUES ('e0009', 'Curl1', '10', '30');
```

```
INSERT INTO `aziendacentrofitness`.`anaerobico` (`IDEsercizioAnaerobico`,
`NomeAnaerobico`, `NumEsecuzioni`, `TempoDiRecupero`)
VALUES ('e0009', 'Curl2', '12', '30');
```

```
INSERT INTO `aziendacentrofitness`.`anaerobico` (`IDEsercizioAnaerobico`,
`NomeAnaerobico`, `NumEsecuzioni`, `TempoDiRecupero`)
VALUES ('e0009', 'Curl3', '14', '30');
```

```
-- -----
-----
```

```
-- Record Attrezzo
```

```
-- -----
-----
```

```
INSERT INTO `aziendacentrofitness`.`attrezzo` (`IDAttrezzo`, `NomeAttrezzo`,
`IDSala`, `ConsumoEnergetico`, `Tipologia`, `LivelloUsura`)
VALUES ('attr1', 'Tapis Roulant', 'sal0005', '20', 'Cardio Fitness', '0.1');
```

```
INSERT INTO `aziendacentrofitness`.`attrezzo` (`IDAttrezzo`, `NomeAttrezzo`,
`IDSala`, `ConsumoEnergetico`, `Tipologia`, `LivelloUsura`)
VALUES ('attr2', 'Tapis Roulant', 'sal0005', '20', 'Cardio Fitness', '0');
```

```
INSERT INTO `aziendacentrofitness`.`attrezzo` (`IDAttrezzo`, `NomeAttrezzo`,
`IDSala`, `ConsumoEnergetico`, `Tipologia`, `LivelloUsura`)
VALUES ('attr3', 'Tapis Roulant', 'sal0005', '25', 'Cardio Fitness', '0');
```

```
INSERT INTO `aziendacentrofitness`.`attrezzo` (`IDAttrezzo`, `NomeAttrezzo`,
`IDSala`, `ConsumoEnergetico`, `Tipologia`, `LivelloUsura`)
VALUES ('attr4', 'Tapis Roulant', 'sal0005', '30', 'Cardio Fitness', '0');
```

```
INSERT INTO `aziendacentrofitness`.`attrezzo` (`IDAttrezzo`, `NomeAttrezzo`,
`IDSala`, `ConsumoEnergetico`, `Tipologia`, `LivelloUsura`)
VALUES ('attr5', 'Pectoral Machine', 'sal0002', '30', 'Pesistica', '0');
```

```
INSERT INTO `aziendacentrofitness`.`attrezzo` (`IDAttrezzo`, `NomeAttrezzo`,
`IDSala`, `ConsumoEnergetico`, `Tipologia`, `LivelloUsura`)
VALUES ('attr6', 'Leg Machne', 'sal002', '20', 'Pesistica', '0');
```

```
INSERT INTO `aziendacentrofitness`.`attrezzo` (`IDAttrezzo`, `NomeAttrezzo`,
`IDSala`, `ConsumoEnergetico`, `Tipologia`, `LivelloUsura`)
VALUES ('attr7', 'Manubrio Regolabile', 'sal002', '10', 'Pesistica', '0');
```

```
-- -----
-----
```

-- Record CalendarioCentroFitness

```
-----  
-----  
INSERT INTO `aziendacentrofitness`.`calendariocentrofitness` (`IDCentro`,  
`NomeGiorno`, `OrarioApertura`, `OrarioChiusura`)  
VALUES ('cf0001', 'Lunedì', '15:00', '22:00');
```

```
INSERT INTO `aziendacentrofitness`.`calendariocentrofitness` (`IDCentro`,  
`NomeGiorno`, `OrarioApertura`, `OrarioChiusura`)  
VALUES ('cf0001', 'Martedì', '9:00', '22:00');
```

```
INSERT INTO `aziendacentrofitness`.`calendariocentrofitness` (`IDCentro`,  
`NomeGiorno`, `OrarioApertura`, `OrarioChiusura`)  
VALUES ('cf0001', 'Mercoledì', '9:00', '21:00');
```

```
INSERT INTO `aziendacentrofitness`.`calendariocentrofitness` (`IDCentro`,  
`NomeGiorno`, `OrarioApertura`, `OrarioChiusura`)  
VALUES ('cf0001', 'Giovedì', '10:00', '21:00');
```

```
INSERT INTO `aziendacentrofitness`.`calendariocentrofitness` (`IDCentro`,  
`NomeGiorno`, `OrarioApertura`, `OrarioChiusura`)  
VALUES ('cf0001', 'Venerdì', '9:00', '22:00');
```

```
INSERT INTO `aziendacentrofitness`.`calendariocentrofitness` (`IDCentro`,  
`NomeGiorno`, `OrarioApertura`, `OrarioChiusura`)  
VALUES ('cf0001', 'Sabato', '9:00', '13:00');
```

```
INSERT INTO `aziendacentrofitness`.`attrezzo` (`IDAttrezzo`, `NomeAttrezzo`,  
`IDSala`, `ConsumoEnergetico`, `Tipologia`, `LivelloUsura`)  
VALUES ('attr7', 'Manubrio', 'sal002', '10', 'Pesistica', '0');
```

-- Record Corso

```
-----  
-----  
INSERT INTO `aziendacentrofitness`.`corso` (`IDCorso`, `NomeDisciplina`,  
`IstruttoreCorso`, `DataInizio`, `DataFine`, `NumeroMaxPartecipanti`, `Livello`)  
VALUES ('corso1', 'Yoga', 'dip1', '2017-02-02', '2017-05-02', '50',  
'Principiante');
```

```
INSERT INTO `aziendacentrofitness`.`corso` (`IDCorso`, `NomeDisciplina`,  
`IstruttoreCorso`, `DataInizio`, `DataFine`, `NumeroMaxPartecipanti`, `Livello`)  
VALUES ('corso2', 'Zumba', 'dip2', '2017-03-05', '2017-09-06', '60',  
'Principiante');
```

```
INSERT INTO `aziendacentrofitness`.`corso` (`IDCorso`, `NomeDisciplina`,  
`IstruttoreCorso`, `DataInizio`, `DataFine`, `NumeroMaxPartecipanti`, `Livello`)  
VALUES ('corso3', 'Nuoto', 'dip3', '2017-05-10', '2017-10-10', '50',  
'Intermedio');
```

```

INSERT INTO `aziendacentrofitness`.`corso` (`IDCorso`, `NomeDisciplina`,
`IstruttoreCorso`, `DataInizio`, `DataFine`, `NumeroMaxPartecipanti`, `Livello`)
VALUES ('corso4', 'Crossfir', 'dip4', '2017-06-02', '2017-11-02', '30',
'Avanzato');

```

```

-- Record CalendarioCorso

```

```

INSERT INTO `aziendacentrofitness`.`calendariocorso` (`IDCalendarioCorso`,
`IDCorso`, `IDCentro`, `GiornoSettimana`, `OrarioInizio`, `OrarioFine`)
VALUES ('calcorso1', 'corso1', 'cf0001', 'Martedi', '15:00', '17:00');

```

```

INSERT INTO `aziendacentrofitness`.`calendariocorso` (`IDCalendarioCorso`,
`IDCorso`, `IDCentro`, `GiornoSettimana`, `OrarioInizio`, `OrarioFine`)
VALUES ('calcorso2', 'corso1', 'cf0001', 'Giovedi', '18:00', '20:00');

```

```

-- Record Link

```

```

INSERT INTO `aziendacentrofitness`.`link` (`URL`) VALUES ('URL1');

```

```

INSERT INTO `aziendacentrofitness`.`link` (`URL`) VALUES ('URL2');

```

```

INSERT INTO `aziendacentrofitness`.`link` (`URL`) VALUES ('URL3');

```

```

INSERT INTO `aziendacentrofitness`.`link` (`URL`) VALUES ('URL4');

```

```

INSERT INTO `aziendacentrofitness`.`link` (`URL`) VALUES ('URL5');

```

```

-- Record Thread

```

```

INSERT INTO `aziendacentrofitness`.`thread` (`IDThread`, `Titolo`)
VALUES ('Thread1', 'Benvenuto');

```

```

INSERT INTO `aziendacentrofitness`.`thread` (`IDThread`, `Titolo`)
VALUES ('Thread2', 'Potenza Muscolare');

```

```

INSERT INTO `aziendacentrofitness`.`thread` (`IDThread`, `Titolo`)
VALUES ('Thread3', 'Esecuzione Esercizi');

```

```

INSERT INTO `aziendacentrofitness`.`thread` (`IDThread`, `Titolo`)
VALUES ('Thread4', 'Sfidal');

```



```
-----  
  
-- Record Post  
-----  
-----
```

```
INSERT INTO `aziendacentrofitness`.`post` (`IDPost`, `Testo`, `Username`,  
`AreaForum`, `ThreadPost`)  
VALUES ('post1', 'Ciao sono felice', 'Leo95',  
'Palestra', 'Thread1');
```

```
INSERT INTO `aziendacentrofitness`.`post` (`IDPost`, `Testo`, `Username`,  
`AreaForum`, `ThreadPost`)  
VALUES ('post2', 'Finalmente ho raggiunto il mio obiettivo', 'Leo95',  
'Felicità', 'Thread2');
```

```
INSERT INTO `aziendacentrofitness`.`post` (`IDPost`, `Testo`, `Username`,  
`Risposta`, `AreaForum`, `ThreadPost`)  
VALUES ('post3', 'Come si utilizza l\'attrezzo Pectoral Machine?', 'Leo95',  
'post6', 'Esercizio', 'Thread3');
```

```
INSERT INTO `aziendacentrofitness`.`post` (`IDPost`, `Testo`, `Username`,  
`AreaForum`, `ThreadPost`)  
VALUES ('post4', 'Bello essere qui', 'Leo95', 'Frase', 'Thread1');
```

```
INSERT INTO `aziendacentrofitness`.`post` (`IDPost`, `Testo`, `Username`,  
`AreaForum`, `ThreadPost`)  
VALUES ('post6', 'Basta leggere le istruzioni', 'ele89', 'Esercizio',  
'Thread3');
```

```
-----  
-----  
  
-- Record Voto  
-----  
-----
```

```
INSERT INTO `aziendacentrofitness`.`voto` (`IDVoto`, `Risposta`, `Stelle`)  
VALUES ('voto1', 'post6', '5');
```

```
-----  
-----  
  
-- Record LinkPost  
-----  
-----
```

```
INSERT INTO `aziendacentrofitness`.`linkpost` (`URL`, `IDPost`)  
VALUES ('URL1', 'post6');
```

```
INSERT INTO `aziendacentrofitness`.`linkpost` (`URL`, `IDPost`)  
VALUES ('URL2', 'post2');
```

-- Record PianoPagamento

```
-----  
-----  
INSERT INTO `aziendacentrofitness`.`pianopagamento` (`IDPagamento`,  
`IDContratto`, `TipologiaPagamento`)  
VALUES ('pag001', 'con001', 'Intero');
```

```
INSERT INTO `aziendacentrofitness`.`pianopagamento` (`IDPagamento`,  
`IDContratto`, `TipologiaPagamento`)  
VALUES ('pag002', 'con002', 'Intero');
```

```
INSERT INTO `aziendacentrofitness`.`pianopagamento` (`IDPagamento`,  
`IDContratto`, `TipologiaPagamento`)  
VALUES ('pag003', 'con003', 'Intero');
```

```
INSERT INTO `aziendacentrofitness`.`pianopagamento` (`IDPagamento`,  
`IDContratto`, `TipologiaPagamento`)  
VALUES ('pag004', 'con004', 'Rateizzato');
```

-- Record Rata

```
-----  
-----  
INSERT INTO `aziendacentrofitness`.`rata` (`IDRata`, `IDPagamento`,  
`ImportoRata`, `IstitutoFinanziario`, `TassoInteresse`, `DataScadenza`, `Stato`)  
VALUES ('rata1', 'pag004', '60', 'Credem', '20', '2018-03-05', 'Non dovuto');
```

```
INSERT INTO `aziendacentrofitness`.`rata` (`IDRata`, `IDPagamento`,  
`ImportoRata`, `IstitutoFinanziario`, `TassoInteresse`, `DataScadenza`, `Stato`)  
VALUES ('rata2', 'pag004', '60', 'Credem', '20', '2008-04-03', 'Non dovuto');
```

-- Record SchedaAllenamento

```
-----  
-----  
INSERT INTO `aziendacentrofitness`.`schedaallenamento` (`IDSchedaAllenamento`,  
`CodFiscaleCliente`, `TutorScheda`, `DataInizio`, `DataFine`)  
VALUES ('sall1', 'cfc0001', 'dip1', '2018-01-06', '2018-02-20');
```

```
INSERT INTO `aziendacentrofitness`.`schedaallenamento` (`IDSchedaAllenamento`,  
`CodFiscaleCliente`, `TutorScheda`, `DataInizio`, `DataFine`)  
VALUES ('sall2', 'cfc0002', 'dip2', '2017-05-06', '2017-09-30');
```

-- Record ComposizioneScheda

```
-----  
-----  
INSERT INTO `aziendacentrofitness`.`composizionesallenamento` (`IDesercizio`,  
`IDSchedaAllenamento`)  
VALUES ('e0001', 'sall1');
```

```
INSERT INTO `aziendacentrofitness`.`composizionesallenamento` (`IDesercizio`,  
`IDSchedaAllenamento`)  
VALUES ('e0002', 'sall1');
```

```
INSERT INTO `aziendacentrofitness`.`composizionesallenamento` (`IDesercizio`,  
`IDSchedaAllenamento`)  
VALUES ('e0003', 'sall1');
```

```
INSERT INTO `aziendacentrofitness`.`composizionesallenamento` (`IDesercizio`,  
`IDSchedaAllenamento`)  
VALUES ('e0004', 'sall2');
```

```
INSERT INTO `aziendacentrofitness`.`composizionesallenamento` (`IDesercizio`,  
`IDSchedaAllenamento`)  
VALUES ('e0009', 'sall2');
```

```
INSERT INTO `aziendacentrofitness`.`composizionesallenamento` (`IDesercizio`,  
`IDSchedaAllenamento`)  
VALUES ('e0011', 'sall2');
```

-- Record CongifurazioneUtilizzo

```
-----  
-----  
INSERT INTO `aziendacentrofitness`.`configurazioneutilizzo` (`IDConfigurazione`,  
`IDesercizio`, `IDAttrezzo`, `NomeRegolazione`, `Intensita`, `IstruzioneEsatta`)  
VALUES ('conf1', 'e0009', 'attr7', 'Peso', '4', '1');
```

```
INSERT INTO `aziendacentrofitness`.`configurazioneutilizzo` (`IDConfigurazione`,  
`IDesercizio`, `IDAttrezzo`, `NomeRegolazione`, `Intensita`, `IstruzioneEsatta`)  
VALUES ('conf2', 'e0009', 'attr7', 'Peso', '6', '1');
```

```
INSERT INTO `aziendacentrofitness`.`configurazioneutilizzo` (`IDConfigurazione`,  
`IDesercizio`, `IDAttrezzo`, `NomeRegolazione`, `Intensita`, `IstruzioneEsatta`)  
VALUES ('conf3', 'e0009', 'attr7', 'Peso', '8', '1');
```

```
INSERT INTO `aziendacentrofitness`.`configurazioneutilizzo` (`IDConfigurazione`,  
`IDesercizio`, `IDAttrezzo`, `NomeRegolazione`, `Intensita`, `IstruzioneEsatta`)  
VALUES ('conf4', 'e0009', 'attr7', 'Peso', '3', '0');
```

```
INSERT INTO `aziendacentrofitness`.`configurazioneutilizzo` (`IDConfigurazione`,  
`IDesercizio`, `IDAttrezzo`, `NomeRegolazione`, `Intensita`, `IstruzioneEsatta`)  
VALUES ('conf5', 'e0009', 'attr7', 'Peso', '6', '0');
```

```

INSERT INTO `aziendacentrofitness`.`configurazioneutilizzo` (`IDConfigurazione`,
`IDEsercizio`, `IDAttrezzo`, `NomeRegolazione`, `Intensita`, `IstruzioneEsatta`)
VALUES ('conf6', 'e0009', 'attr7', 'Peso', '7', '0');
-----

-- Record MonitoraggioSmart
-----

INSERT INTO `aziendacentrofitness`.`monitoraggiosmart` (`IDMonitoraggio`,
`IDEsercizio`, `IDSchedaAllenamento`, `IstanteInizio`, `IstanteFine`,
`DataMonitoraggio`)
VALUES ('mon001', 'e0003', 'sall1', '15:00', '15:10', '2017-02-03');

INSERT INTO `aziendacentrofitness`.`monitoraggiosmart` (`IDMonitoraggio`,
`IDEsercizio`, `IDSchedaAllenamento`, `IstanteInizio`, `IstanteFine`,
`DataMonitoraggio`)
VALUES ('mon002', 'e0009', 'sal2', '16:00', '16:20', '2018-01-11');
-----

-- Record DatiRipetizione
-----

INSERT INTO `aziendacentrofitness`.`datiripetizione` (`IDRipetizione`,
`NomeRipetizione`, `NumeroEsecuzioni`, `TempoRecupero`)
VALUES ('ripcl', 'curl1', '10', '42');

INSERT INTO `aziendacentrofitness`.`datiripetizione` (`IDRipetizione`,
`NomeRipetizione`, `NumeroEsecuzioni`, `TempoRecupero`)
VALUES ('ripc2', 'curl2', '12', '50');

INSERT INTO `aziendacentrofitness`.`datiripetizione` (`IDRipetizione`,
`NomeRipetizione`, `NumeroEsecuzioni`, `TempoRecupero`)
VALUES ('ripc3', 'curl3', '12', '100');
-----

-- Record MonitoraggioRipetizione
-----

INSERT INTO `aziendacentrofitness`.`monitoraggioripetizione` (`IDMonitoraggio`,
`IDRipetizione`)
VALUES ('man002', 'ripcl');

INSERT INTO `aziendacentrofitness`.`monitoraggioripetizione` (`IDMonitoraggio`,
`IDRipetizione`)
VALUES ('man002', 'ripc2');

```

```

INSERT INTO `aziendacentrofitness`.`monitoraggioripetizione` (`IDMonitoraggio`,
`IDRipetizione`)
VALUES ('man002', 'ripc3');

```

```

-- Record ConfigurazioneRipetizione

```

```

INSERT INTO `aziendacentrofitness`.`configurazioneripetizione`
(`EsercizioAnaerobico`, `IDConfigurazione`, `Nome`)
VALUES ('e0009', 'conf1', 'Curl1');

```

```

INSERT INTO `aziendacentrofitness`.`configurazioneripetizione`
(`EsercizioAnaerobico`, `IDConfigurazione`, `Nome`)
VALUES ('e0009', 'conf2', 'Curl2');

```

```

INSERT INTO `aziendacentrofitness`.`configurazioneripetizione`
(`EsercizioAnaerobico`, `IDConfigurazione`, `Nome`)
VALUES ('e0009', 'conf3', 'Curl3');

```

```

-- Record ConfigurazioneCliente

```

```

INSERT INTO `aziendacentrofitness`.`configurazionecliente` (`IDMonitoraggio`,
`IDConfigurazione`)
VALUES ('mon002', ' conf4');

```

```

INSERT INTO `aziendacentrofitness`.`configurazionecliente` (`IDMonitoraggio`,
`IDConfigurazione`)
VALUES ('mon002', 'conf5');

```

```

INSERT INTO `aziendacentrofitness`.`configurazionecliente` (`IDMonitoraggio`,
`IDConfigurazione`)
VALUES ('mon002', 'conf6');

```

```

-- Record AttrezzoUtilizzato

```

```

INSERT INTO `aziendacentrofitness`.`attrezzoutilizzato` (`IDAttrezzo`,
`IDMonitoraggio`)
VALUES ('mon002', 'attr7');

```

```

-- Record CorsoDisponibile
-- -----
-----
INSERT INTO `aziendacentrofitness`.`corsodisponibile` (`IDCorso`, `IDContratto`)
VALUES ('corso2', 'con001');

INSERT INTO `aziendacentrofitness`.`corsodisponibile` (`IDCorso`, `IDContratto`)
VALUES ('corso1', 'con001');

INSERT INTO `aziendacentrofitness`.`corsodisponibile` (`IDCorso`, `IDContratto`)
VALUES ('corso3', 'con004');

INSERT INTO `aziendacentrofitness`.`corsodisponibile` (`IDCorso`, `IDContratto`)
VALUES ('corso4', 'con004');
-- -----
-----

-- Record CorsoFrequentato
-- -----
-----
INSERT INTO `aziendacentrofitness`.`corsofrequentato` (`IDCorso`,
`CodFiscaleCliente`)
VALUES ('corso2', 'cfc0001');

INSERT INTO `aziendacentrofitness`.`corsofrequentato` (`IDCorso`,
`CodFiscaleCliente`)
VALUES ('corso4', 'cfc0004');
-- -----
-----

-- Record CorsoInPiscina
-- -----
-----
INSERT INTO `aziendacentrofitness`.`corsoinpiscina` (`IDCorso`, `IDPiscina`)
VALUES ('Corso3', 'p0004');
-- -----
-----

-- Record CorsoInSala
-- -----
-----
INSERT INTO `aziendacentrofitness`.`corsoinsala` (`IDCorso`, `IDSala`)
VALUES ('corso1', 'sal0012');

INSERT INTO `aziendacentrofitness`.`corsoinsala` (`IDCorso`, `IDSala`)
VALUES ('corso2', 'sal0011');
-- -----
-----

```

-- Record OrarioDiLavoro

```
-----  
-----  
INSERT INTO `aziendacentrofitness`.`orariodilavoro` (`IDCentro`,  
`CodFiscaleDip`, `NomeGiornoLavoro`, `attivita`, `OrarioInizio`, `OrarioFine`)  
VALUES ('cf0001', 'dip1', 'Martedi', 'Istruttore', '15:00', '17:00');
```

```
INSERT INTO `aziendacentrofitness`.`orariodilavoro` (`IDCentro`,  
`CodFiscaleDip`, `NomeGiornoLavoro`, `attivita`, `OrarioInizio`, `OrarioFine`)  
VALUES ('cf0001', 'dip1', 'Giovedi', 'Istruttore', '18:00', '20:00');
```

```
INSERT INTO `aziendacentrofitness`.`orariodilavoro` (`IDCentro`,  
`CodFiscaleDip`, `NomeGiornoLavoro`, `attivita`, `OrarioInizio`, `OrarioFine`)  
VALUES ('cf0001', 'dip2', 'Lunedì', 'Tutor', '9:00', '13:00');
```

```
INSERT INTO `aziendacentrofitness`.`orariodilavoro` (`IDCentro`,  
`CodFiscaleDip`, `NomeGiornoLavoro`, `attivita`, `OrarioInizio`, `OrarioFine`)  
VALUES ('cf0001', 'dip3', 'Martedi', 'Tutor', '15:00', '20:00');
```

-- Record RichiestaAmicizia

```
-----  
-----  
INSERT INTO `aziendacentrofitness`.`richiestaamicizia` (`IDRichiesta`,  
`Username`, `UsernameDestinatario`, `StatoRichiesta`)  
VALUES ('rich1', 'Leo95', 'pi95', 'In sospeso');
```

-- Record Sfida

```
-----  
-----  
INSERT INTO `aziendacentrofitness`.`sfida` (`Proponente`, `DataInizioSfida`,  
`DataLancio`, `DataScadenza`, `Scopo`, `ThreadSfida`, `AllenamentoSfida`,  
`AlimentazioneSfida`, `Vincitore`)  
VALUES ('Leo95', '2017-03-03', '2017-02-03', '2017-03-18', '10 km in 20 minuti',  
'Thread5', 'sall1', 'ali5', 'erne90');
```

-- Record Partecipante

```
-----  
-----  
INSERT INTO `aziendacentrofitness`.`partecipante` (`UsernameAmico`,  
`Proponente`, `DataInizioSfida`)  
VALUES ('ele89', 'Leo95', '2017-03-03');
```

```
INSERT INTO `aziendacentrofitness`.`partecipante` (`UsernameAmico`,  
`Proponente`, `DataInizioSfida`)  
VALUES ('erne90', 'Leo95', '2017-03-03');
```

8.3 Vincoli Integrità Generici, Event, Analytic, Trigger

```
DROP TRIGGER IF EXISTS Entita_Stato_Visita_Cliente;

DELIMITER $$
CREATE TRIGGER Entita_Stato_Visita_Cliente
BEFORE INSERT ON misurazioniimpedenziometriche
FOR EACH ROW

BEGIN

    DECLARE rangecliente DOUBLE DEFAULT 0;

    SET rangecliente = ((NEW.Peso * (NEW.MassaGrassa +NEW.MassaMagra))
                        /(NEW.Altezza *
                           NEW.altezza * NEW.AcquaTotale)
                        );

    CASE
        WHEN (rangecliente <= 16.5) THEN
            SET NEW.entita = 'Grave';
            SET NEW.statocliente = 'Sottopeso III grado';
        WHEN (rangecliente <= 16.5 AND rangecliente >= 16.6) THEN
            SET NEW.entita = 'Medio';
            SET NEW.statocliente = 'Sottopeso II grado';
        WHEN (rangecliente <= 18.4 AND rangecliente >= 17.6) THEN
            SET NEW.entita = 'Lieve';
            SET NEW.statocliente = 'Sottopeso I grado';
        WHEN (rangecliente <= 20.6 AND rangecliente >= 18.5) THEN
            SET NEW.entita = 'Grave';
            SET NEW.statocliente = 'NormoPeso III grado';
        WHEN (rangecliente <= 22.8 AND rangecliente >= 20.7) THEN
            SET NEW.entita = 'Medio';
            SET NEW.statocliente = 'NormoPeso II grado';
        WHEN (rangecliente <= 24.9 AND rangecliente >= 22.9) THEN
            SET NEW.entita = 'Lieve';
            SET NEW.statocliente = 'NormoPeso I grado';
        WHEN (rangecliente <= 32 AND rangecliente >= 25) THEN
            SET NEW.entita = 'Lieve';
            SET NEW.statocliente = 'Sovrappeso I grado';
        WHEN (rangecliente <= 40 AND rangecliente >= 32.1) THEN
            SET NEW.entita = 'Medio';
            SET NEW.statocliente = 'Sovrappeso II grado';
        WHEN (rangecliente > 40) THEN
            SET NEW.entita = 'Grave';
            SET NEW.statocliente = 'Sovrappeso III grado';
    END CASE;
END $$
DELIMITER;
```



```

/*****
*****/

DROP TRIGGER IF EXISTS Accesso_Piscina;

DELIMITER $$

CREATE TRIGGER Accesso_Piscina
AFTER INSERT ON contratto
FOR EACH ROW

BEGIN

    IF (NEW.ImportoMensile = 30 AND NEW.TipologiaContratto = 'Silver') then
        INSERT INTO accessopiscina
            SELECT P.IDPiscina AS IDPiscina,
                   NEW.IDContratto AS IDContratto,  2 AS NumeroAccessi
            FROM piscina P
            WHERE P.IDCentro = NEW.IDCentro
                  AND P.TipologiaContratto = 'Silver';

    ELSEIF (NEW.ImportoMensile = 50 AND NEW.TipologiaContratto = 'Gold') THEN
        INSERT INTO accessopiscina
            SELECT P.IDPiscina AS IDPiscina,
                   NEW.IDContratto AS IDContratto,  6 AS NumeroAccessi
            FROM piscina P
            WHERE P.IDCentro = NEW.IDCentro
                  AND (P.TipologiaContratto = 'Gold'
                      OR
                      P.TipologiaContratto = 'Silver');

    ELSEIF (NEW.ImportoMensile = 55 AND NEW.TipologiaContratto = 'Gold') THEN
        INSERT INTO accessopiscina
            SELECT P.IDPiscina AS IDPiscina,
                   NEW.IDContratto AS IDContratto,  8 AS NumeroAccessi
            FROM piscina P
            WHERE P.IDCentro = NEW.IDCentro
                  AND (P.TipologiaContratto = 'Gold'
                      OR
                      P.TipologiaContratto = 'Silver');

    ELSEIF (NEW.ImportoMensile = 65 AND NEW.TipologiaContratto = 'Platinum')
    THEN
        INSERT INTO accessopiscina
            SELECT P.IDPiscina AS IDPiscina,
                   NEW.IDContratto AS IDContratto, 10 AS NumeroAccessi
            FROM piscina P
            WHERE P.IDCentro = NEW.IDCentro
                  AND (P.TipologiaContratto = 'Platinum'
                      OR
                      P.TipologiaContratto = 'Gold'
                      OR
                      P.TipologiaContratto = 'Silver');

```

```

ELSEIF (NEW.ImportoMensile = 70 AND NEW.TipologiaContratto = 'Platinum')
THEN
    INSERT INTO accessopiscina
        SELECT P.IDPiscina AS IDPiscina,
               NEW.IDContratto AS IDContratto, 12 AS NumeroAccessi
    FROM piscina P
    WHERE P.IDCentro = NEW.IDCentro
        AND (P.TipologiaContratto = 'Platinum'
            OR
            P.TipologiaContratto = 'Gold'
            OR
            P.TipologiaContratto = 'Silver');

ELSEIF (NEW.ImportoMensile = 100 and new.TipologiaContratto =
        'Personalizzato') THEN

    INSERT INTO accessopiscina
        SELECT P.IDPiscina AS IDPiscina,
               NEW.IDContratto AS IDContratto, 20 AS NumeroAccessi
    FROM piscina P
    WHERE P.IDCentro = NEW.IDCentro;

END IF;
END $$

DELIMITER;

/*****
*****/

DROP TRIGGER IF EXISTS Controllo_Stelle;

DELIMITER $$

CREATE TRIGGER Controllo_Stelle
BEFORE INSERT ON voto
FOR EACH ROW
BEGIN
    IF (NEW.Stelle <= 0) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Non si accettano valutazioni minori uguali a zero.';
    ELSEIF (NEW.Stelle > 5) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Non si accettano valutazioni maggiori di 5.';
    END IF;
END $$
DELIMITER;

/*****
*****/

```

*****/

DROP EVENT IF EXISTS Aggiornamento_Posti_Disponibili;

DELIMITER \$\$

CREATE EVENT Aggiornamento_Posti_Disponibili

ON SCHEDULE EVERY 5 MINUTE

DO

BEGIN

UPDATE spogliatoio S

SET S.PostiDisponibili = S.PostiDisponibili +

(

SELECT COUNT(*)

FROM accesso A

INNER JOIN

armadietto T ON A.IDArmadietto = T.IDArmadietto

WHERE T.IDSpogliatoio = S.IDSpogliatoio

AND

S.Capienza > S.PostiDisponibili

AND

A.OrarioUscita IS NOT NULL

);

END \$\$

DELIMITER;

/*****/

DROP TRIGGER IF EXISTS Inserimento_Accesso;

DELIMITER \$\$

CREATE TRIGGER Inserimento_Accesso

BEFORE INSERT ON accesso

FOR EACH ROW

BEGIN

DECLARE valore INTEGER DEFAULT 0;

DECLARE sessocliente VARCHAR (1) DEFAULT 0;

DECLARE sessospogliatoio VARCHAR (1) DEFAULT 0;

SET sessocliente = (SELECT DISTINCT C.Sesso

FROM accesso A

INNER JOIN

cliente C ON A.CodFiscaleCliente =

C.CodFiscaleCliente

WHERE A.CodFiscaleCliente = NEW.CodFiscaleCliente);

```

SET sessospogliatoio = ( SELECT DISTINCT S.Sesso
                        FROM accesso A
                        INNER JOIN
                        armadietto T ON A.IDArmadietto =
                        T.IDArmadietto
                        INNER JOIN
                        spogliatoio S ON S.IDSpogliatoio =
                        T.IDSpogliatoio
                        WHERE A.IDArmadietto = NEW.IDArmadietto
                        );

SET valore = ( SELECT COUNT(*)
                FROM accesso A
                INNER JOIN
                armadietto T ON A.IDArmadietto = T.IDArmadietto
                WHERE A.DataAccesso = NEW.DataAccesso
                AND
                A.IDArmadietto = NEW.IDArmadietto
                AND
                A.OrarioUscita IS NULL
                );

IF (valore > 0) THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Armadietto non disponibile';
ELSEIF (sessocliente <> sessospogliatoio) THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Spogliatoio Sbagliato';
ELSE
    UPDATE Spogliatoio L
    SET L.PostiDisponibili = L.PostiDisponibili -1
    WHERE L.IDSpogliatoio = ( SELECT DISTINCT T.IDSpogliatoio
                                FROM accesso A
                                INNER JOIN
                                armadietto T ON A.IDArmadietto =
                                T.IDArmadietto
                                WHERE A.IDArmadietto = NEW.IDArmadietto
                                );
END IF;

END $$
DELIMITER;

```

```

/*****
*****/

```

```

DROP TRIGGER IF EXISTS Controllo_Livello_Usura_Attrezzatura;

DELIMITER $$

CREATE TRIGGER Controllo_Livello_Usura_Attrezzatura
BEFORE INSERT ON attrezzoutilizzato
FOR EACH ROW
BEGIN

    DECLARE usura DOUBLE DEFAULT 0;

    SET usura = ( SELECT DISTINCT A.LivelloUsura
                  FROM attrezzatura A
                  INNER JOIN
                      attrezzoutilizzato T ON A.IDAttrezzatura = T.IDAttrezzatura
                  WHERE A.IDAttrezzatura = NEW.IDAttrezzatura
                );

    IF (usura = 100) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Attrezzatura non utilizzabile, bisogno di manutenzione';
    ELSE
        UPDATE attrezzatura P
        SET P.LivelloUsura = P.LivelloUsura + 0.1
        WHERE P.IDAttrezzatura = NEW.IDAttrezzatura;
    END IF;

END $$
DELIMITER;

/*****
*****/

DROP TRIGGER IF EXISTS Controllo_Multisede;

DELIMITER $$

CREATE TRIGGER Controllo_Multisede
BEFORE INSERT ON multisede
FOR EACH ROW
BEGIN

    DECLARE conta INTEGER DEFAULT 0;

    SET conta = ( SELECT COUNT(*)
                  FROM multisede M
                  WHERE M.IDContratto = NEW.IDContratto
                );

    IF (conta = 3) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Impossibile aggiungere altre sedi, il cliente è
iscritto già a 3 sedi diverse';
    END IF;

END $$
DELIMITER;

```

```

/*****
*****/

DROP TRIGGER IF EXISTS Gestione_Turnazione;

DELIMITER $$

CREATE TRIGGER Gestione_Turnazione
BEFORE INSERT ON orariodilavoro
FOR EACH ROW
BEGIN

    DECLARE finito INTEGER DEFAULT 0;
    DECLARE nuoveore TIME DEFAULT 0;
    DECLARE calcoloore TIME DEFAULT 0;
    DECLARE centropermesso VARCHAR (45) DEFAULT NULL;
    DECLARE orariol TIME DEFAULT 0;
    DECLARE orario2 TIME DEFAULT 0;

    DECLARE oraril CURSOR FOR
        SELECT P.OrarioInizio, P.OrarioFine
        FROM orariodilavoro P
        WHERE P.IDCentro = NEW.IDCentro
            AND
            P.NomeGiornoLavoro = NEW.NomeGiornoLavoro
            AND
            P.CodFiscaleDip = NEW.CodFiscaleDip;

    DECLARE CONTINUE HANDLER FOR NOT FOUND SET finito = 1;

    OPEN oraril;

    scan: LOOP
        FETCH oraril INTO orariol, orario2;
        IF finito = 1 THEN
            LEAVE scan;
        ELSEIF (orariol <= NEW.OrarioInizio AND orario2 >=
            NEW.OrarioInizio) THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Impossibile inserire
                turnazione, il cliente è
                impegnato in quella fascia
                oraria';
            LEAVE scan;
        END IF;
    END LOOP scan;
    CLOSE oraril;

    SET centropermesso = (SELECT DISTINCT O.IDCentro
        FROM orariodilavoro O
        WHERE O.NomeGiornoLavoro = NEW.NomeGiornoLavoro
            AND
            O.CodFiscaleDip = NEW.CodFiscaleDip
    );

```

```

SET nuoveore = (TIME_FORMAT(TIMEDIFF(NEW.OrarioFine, NEW.OrarioInizio),
                                '%H:%i'));

SET calcoloore = (
    SELECT TIME_FORMAT(SUM( TIMEDIFF(O.OrarioFine,
                                    O.OrarioInizio)), '%H:%i')
    FROM orariodilavoro O
    WHERE O.NomeGiornoLavoro = NEW.NomeGiornoLavoro
    AND
    O.CodFiscaleDip = NEW.CodFiscaleDip
    AND
    O.IDCentro = NEW.IDCentro
);

IF (ADDTIME(calcoloore, nuoveore) > '8:00') THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Errore, Impossibile inserire Turnazione poichè
                        il dipendente lavora più di 8 ore';
ELSEIF (NEW.IDCentro <> centropermesso) THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Errore, impossibile inserire fasce orario di uno
                        stesso giorno in cui un dipendente è assegnato a
                        centri diversi';
END IF;
END $$
DELIMITER;

/*****
*****/

DROP TRIGGER IF EXISTS InserimentoSpogliatoio;

DELIMITER $$

CREATE TRIGGER InserimentoSpogliatoio
BEFORE INSERT ON spogliatoio
FOR EACH ROW
BEGIN
    IF (NEW.Capienza <> NEW.PostiDisponibili) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Errore, Capienza deve essere uguale a posti
disponibili';
    END IF;
END $$
DELIMITER;

/*****
*****/

```

```

DROP TABLE IF EXISTS performancesportiva;
CREATE TABLE PerformanceSportiva (
    Cliente VARCHAR(45) NOT NULL,
    SchedaAllenamento VARCHAR (45) NOT NULL,
    Esercizio VARCHAR (45) NOT NULL,
    NumeroEsecuzioni INT NOT NULL,
    EsecuzioniCorrette INT NOT NULL,
    EsercizioDaSostituire BOOL,
    PRIMARY KEY (Cliente, SchedaAllenamento, Esercizio)
)ENGINE = InnoDB DEFAULT CHAR SET = latin1;

CREATE OR REPLACE VIEW EsecuzioneCorrettaEsercizi AS
SELECT *
FROM esercizio E
    LEFT OUTER JOIN
aerobico A ON E.IDEsercizio = A.IDEsercizioAerobico
    LEFT OUTER JOIN
anaerobico AN ON E.IDEsercizio = AN.IDEsercizioAnaerobico;

DROP PROCEDURE IF EXISTS refresh_Performance_Sportiva_Aerobica;
DELIMITER $$
CREATE PROCEDURE refresh_Performance_Sportiva_Aerobica (OUT esito INTEGER)
BEGIN

    DECLARE esito INTEGER DEFAULT 0;

    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        SET esito = 1;
        SELECT 'Si è verificato un errore, materialized view non aggiornata.';
    END;

    TRUNCATE PerformanceSportiva;

    INSERT INTO PerformanceSportiva (Cliente, SchedaAllenamento, Esercizio,
                                     NumeroEsecuzioni, EsecuzioniCorrette)
    SELECT S.CodFiscaleCliente,
           S.IDSchedaAllenamento,
           C.IDEsercizio, COUNT(*),
           SUM(IF(M.IstanteFine - M.IstanteInizio = EC.DurataMinuti, 1, 0))
    FROM schedaallenamento S
        INNER JOIN composizioneallenamento C ON S.IDSchedaAllenamento =
                                                    C.IDSchedaAllenamento
        INNER JOIN EsecuzioneCorrettaEsercizi EC ON C.IDEsercizio =
                                                    EC.IDEsercizio
        INNER JOIN monitoraggiosmart M ON (M.IDEsercizio = EC.IDEsercizio
                                           AND
                                           M.IDSchedaAllenamento =
                                           C.IDSchedaAllenamento)

```



```

WHERE EC.NomeAnaerobico IS NULL
GROUP BY S.CodFiscaleCliente, S.IDSchedaAllenamento, C.IDEsercizio;

END $$

DELIMITER;

DROP EVENT IF EXISTS Event_Performance_Sportiva_Aerobica;

DELIMITER $$

CREATE EVENT Event_Performance_Sportiva_Aerobica
ON SCHEDULE EVERY 2 WEEK
DO
BEGIN
    SET @esito = 0;

    CALL refresh_Performance_Sportiva_Aerobica (@esito);

    IF @esito = 1 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Errore refresh.';
    END IF;

    UPDATE performancesportiva P
    SET P.EsercizioDaSostituire = 1
    WHERE P.EsecuzioniCorrette < (P.NumeroEsecuzioni * 70)/100;

END $$

DELIMITER ;

```

```

/*****
*/

DROP TRIGGER IF EXISTS Controllo_Amicizia;

DELIMITER $$

CREATE TRIGGER Controllo_Amicizia
BEFORE INSERT ON amico
FOR EACH ROW
BEGIN

    DECLARE conta INTEGER DEFAULT 0;

    SET conta = ( SELECT COUNT(*)
                  FROM amico A
                  WHERE A.Username = new.UsernameAmico
                  AND
                  A.UsernameAmico = new.Username
                );

    IF (conta > 0) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Errore inserimento, Amicizia già presente.';
    END IF;
END $$

DELIMITER ;

/*****
*****/
DROP TRIGGER IF EXISTS Controllo_Partecipanti_Corso;

DELIMITER $$

CREATE TRIGGER Controllo_Partecipanti_Corso
BEFORE INSERT ON corsofrequentato
FOR EACH ROW
BEGIN

    DECLARE numeromax INTEGER DEFAULT 0;
    DECLARE numerol INTEGER DEFAULT 0;

    SET numeromax = (
        SELECT C.NumeroMaxPartecipanti
        FROM corso C
        WHERE C.IDCorso = NEW.IDCorso
    );

    SET numerol = (
        SELECT count(*)
        FROM corsofrequentato CF
        WHERE CF.IDCorso = NEW.IDCorso);

```

```

        IF (numero1 = numeromax) THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Errore, Corso ha raggiunto numero massimo di
                                partecipanti.';
        END IF;
    END $$

DELIMITER ;

/*****
*****/

DROP TRIGGER IF EXISTS Controllo_Richiesta_Amicizia;

DELIMITER $$

CREATE TRIGGER Controllo_Richiesta_Amicizia
AFTER UPDATE ON richiestaamicizia
FOR EACH ROW
BEGIN

    IF(NEW.StatoRichiesta = 'Accettata') THEN
        INSERT INTO amico
        VALUES(NEW.UsernameDestinatario, NEW.Username);
    END IF;
END $$

DELIMITER ;

/*****/

```

8.4 Operazioni Significative

-- Operazione 1

DROP PROCEDURE IF EXISTS ModificaRata;

DELIMITER \$\$

CREATE PROCEDURE ModificaRata (IN codice_contratto VARCHAR (45))

BEGIN

 DECLARE ratatarget VARCHAR (45) DEFAULT NULL;

 SET ratatarget =(

 SELECT R.IDRata

 FROM contratto C

 INNER JOIN

 pianopagamento P ON C.IDContratto = P.IDContratto

 INNER JOIN rata R ON P.IDPagamento = R.IDPagamento

 WHERE C.IDContratto = codice_contratto

 AND R.DataScadenza < current_date()

 AND R.Stato = 'non ancora dovuto'

);

 IF (ratatarget IS NOT NULL) THEN

 UPDATE rata L

 SET L.stato = 'scaduto'

 WHERE L.IDRata = ratatarget;

 ELSE

 SIGNAL SQLSTATE '45000'

 SET MESSAGE_TEXT = 'Non ci sono rate scadute o contratto non prevede
rateizzazione';

 END IF;

END \$\$

DELIMITER ;

-- Operazione 2

DROP PROCEDURE IF EXISTS InserimentoSchedaAllenamento;

DELIMITER \$\$

CREATE PROCEDURE InserimentoSchedaAllenamento (IN codice_cliente VARCHAR (45),
 IN codice_scheda VARCHAR (45), IN
 tutor_scheda VARCHAR (45), IN
 data_inizio DATE, IN data_fine
 DATE)

BEGIN

 DECLARE conta INTEGER DEFAULT 0;

```

SET conta = (
    SELECT COUNT(*)
    FROM schedaaallenamento A
    WHERE A.CodFiscaleCliente = codice_cliente
);

IF (conta > 0) THEN
    INSERT INTO AllenamentoPassato
        SELECT S.IDSchedaAllenamento AS IDSchedaAllenamento,
            S.CodFiscaleCliente AS CodFiscaleCliente
        FROM SchedaAllenamento S
        WHERE S.CodfiscaleCliente = codice_cliente
            AND
            S.DataFine = ( SELECT MAX(S1.DataFine)
                            FROM schedaaallenamento S1
                            WHERE S1.CodFiscaleCliente =
                                codice_cliente
                        );
END IF;

INSERT INTO SchedaAllenamento(IDSchedaAllenamento, CodFiscaleCliente,
                                TutorScheda, DataInizio, DataFine)
VALUES(codice_scheda, codice_cliente, tutor_scheda, data_inizio, data_fine);

END $$

```

DELIMITER ;

-- **Operazione 3**

DROP PROCEDURE IF EXISTS InserimentoPotenziamento;

DELIMITER \$\$

```

CREATE PROCEDURE InserimentoPotenziamento (IN codice_contratto VARCHAR(45), IN
                                             codice_cliente VARCHAR (45), IN
                                             codice_consulente VARCHAR (45), IN
                                             codice_centro VARCHAR (45), IN
                                             tipo_contratto VARCHAR (45), IN
                                             accessi_settimana INT, IN
                                             importo_mese INT, IN durata_mensile
                                             INT, IN data_iscrizione DATE, IN
                                             scopo_potenziamento VARCHAR (45), IN
                                             nome_muscolo VARCHAR (45),
                                             IN livello_pot VARCHAR (45))

BEGIN

    DECLARE liv VARCHAR (45) DEFAULT NULL;
    DECLARE mus VARCHAR (45) DEFAULT NULL;

```

```

INSERT INTO Contratto(IDContratto, IDCentro, Cliente, Consulente,
                      TipologiaContratto, AccessiSettimanali,
                      ImportoMensile, DurataMensile, DataIscrizione,
                      Scopo)
VALUES(codice_contratto, codice_centro, codice_cliente, codice_consulente,
      tipo_contratto, accessi_settimana, importo_mese, durata_mensile,
      data_iscirizione, scopo_potenziamento);

IF (scopo_potenziamento = 'Potenziamento Muscolare') THEN
    SET liv = ( SELECT DISTINCT Livello
                FROM FasciMuscolari
                WHERE Livello = livello_pot
                );
    SET mus = ( SELECT DISTINCT NomeFasciMuscolari
                FROM FasciMuscolari
                WHERE NomeFasciMuscolari = nome_muscolo
                );

    IF (liv IS NULL OR mus IS null) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Errore FascioMuscolare o Livello non
                           corrispondono';
    ELSE
        INSERT INTO Potenziamento(IDContratto, NomeFasciMuscolari,
                                   Livello)
        VALUES (codice_contratto, mus, liv);
    END IF;
END IF;
END $$

DELIMITER ;

-- -----
-- Operazione 4
-- -----

DROP PROCEDURE IF EXISTS InserimentoVisita;

DELIMITER $$
CREATE PROCEDURE InserimentoVisita (IN codice_cliente VARCHAR(45), IN
                                   codice_misurazione VARCHAR (45), IN data_mis
                                   DATE, IN acqua DOUBLE, IN massa_grassa
                                   DOUBLE, IN massa_magra DOUBLE, IN peso INT,
                                   IN altezza DOUBLE)
BEGIN
    INSERT INTO MisurazioniImpendenzimentriche
    VALUES(codice_misurazione, codice_cliente, data_mis, acqua, massa_magra,
           massa_grassa, peso, altezza);
END $$
DELIMITER;

-- -----

```

```

-- Operazione 5
-----
DROP PROCEDURE IF EXISTS AttrezzoUsurato;
DELIMITER $$
CREATE PROCEDURE AttrezzoUsurato (IN _attrezzo VARCHAR (45), OUT risposta_
                                VARCHAR (45))
BEGIN
    SELECT IF(A.LivelloUsura = 100, 'SI', 'NO') INTO risposta_
    FROM attrezzo A
    WHERE A.IDAttrezzo = _attrezzo;
END $$
DELIMITER ;

-----
-- Operazione 6
-----
DROP PROCEDURE IF EXISTS PostDiUnThread;
DELIMITER $$

CREATE PROCEDURE PostDiUnThread(IN thread_codice VARCHAR(45))
BEGIN
    SELECT *
    FROM post
    WHERE ThreadPost = thread_codice;
END $$

DELIMITER ;

-----
-- Operazione 7
-----
DROP PROCEDURE IF EXISTS ControlloRipetizioni;
DELIMITER $$

CREATE PROCEDURE ControlloRipetizioni (IN codice_cliente VARCHAR(45), IN
                                data_rip DATE, OUT numero_ripetizioni
                                INT)
BEGIN
    DECLARE scheda VARCHAR (100) DEFAULT 0;
    DECLARE es VARCHAR (100) DEFAULT 0;
    SET scheda =(
        SELECT S.IDSchedaAllenamento
        FROM cliente C
        INNER JOIN
            schedaaallenamento S ON C.CodFiscaleCliente =
                                S.CodFiscaleCliente
        WHERE C.CodFiscaleCliente =
            codice_cliente
            AND
            S.DataFine = (
                SELECT MAX(S1.DataFine)

```

```

FROM schedaaallenamento S1
WHERE S1.CodFiscaleCliente =
        codice_cliente
    )
);

SET es = (
    SELECT A.IDEsercizio
    FROM schedaaallenamento S
        INNER JOIN
        cliente C ON C.CodFiscaleCliente = S.CodFiscaleCliente
        INNER JOIN
        composizioneallenamento P ON P.IDSchedaAllenamento =
            S.IDSchedaAllenamento

        INNER JOIN
        esercizio E ON E.IDEsercizio = P.IDEsercizio
        INNER JOIN
        anaerobico A ON A.IDEsercizio = E.IDEsercizio
    WHERE C.CodFiscaleCliente = codice_cliente
        AND
        S.IDSchedaAllenamento = scheda
        AND
        E.DispendioEnergeticoMedio = ( SELECT
                                            MAX(E.DispendioEnergeticoMedio)
                                            FROM schedaaallenamento S1
                                                INNER JOIN
                                                composizioneallenamento P1
                                                ON P1.IDSchedaAllenamento
                                                    = S1.IDSchedaAllenamento
                                                INNER JOIN
                                                esercizio E1 ON
                                                E1.IDEsercizio =
                                                P1.IDEsercizio
                                                INNER JOIN
                                                anaerobico A1 ON
                                                A1.IDEsercizio =
                                                E1.IDEsercizio
                                            WHERE S1.IDSchedaAllenamento =
                                                scheda
                                        )
        LIMIT 1);

SELECT COUNT(R.IDRipetizione) INTO numero_ripetizioni
FROM monitoraggiosmart S
    INNER JOIN
    monitoraggioripetizione R ON S.IDMonitoraggio = R.IDMonitoraggio
WHERE S.IDEsercizio = es
    AND
    S.IDSchedaAllenamento = scheda
    AND
    D.DataMonitoraggio = data_rip;

END $$
DELIMITER ;
-- -----

```


-- **Operazione 8**

-- -----

DROP PROCEDURE IF EXISTS TrovarePosti;

DELIMITER \$\$

CREATE PROCEDURE TrovarePosti (IN _accesso VARCHAR(45), OUT posti_ INT)

BEGIN

SELECT IF(S.PostiDisponibili = 0, 0, S.PostiDisponibili - 1) INTO posti_

FROM accesso A

INNER JOIN

armadietto M ON A.IDArmadietto = M.IDArmadietto

INNER JOIN

spogliatoio S ON S.IDSpogliatoio = M.IDSpogliatoio

WHERE A.IDAccesso = _accesso;

END \$\$

DELIMITER ;
