

Università di Pisa  
Corso di laurea in Ingegneria Informatica

**Documentazione del progetto  
per il corso di Basi di Dati**

Anno Accademico 2016-2017

Luca barsellotti, stefano petrocchi

Sommario

[Analisi delle specifiche 4](#_Toc499046033)

[Livello di usura dei macchinari (2.2.1) 4](#_Toc499046034)

[Scala dei valori relativi al potenziamento muscolare (2.2.3) 4](#_Toc499046035)

[Metrica per la misurazione delle performance del cliente (2.3.6 e 2.5.2) 4](#_Toc499046036)

[Politica di attribuzione della credibilità di un utente (2.4.1) 5](#_Toc499046037)

[Suggerimento per la creazione di una cerchia (2.4.2) 5](#_Toc499046038)

[Valutazione dello sforzo psicofisico (2.4.3) 6](#_Toc499046039)

[Rotazione del magazzino e promozioni (2.5.3) 6](#_Toc499046040)

[Progettazione concettuale 7](#_Toc499046041)

[Strategia di progettazione 7](#_Toc499046042)

[Dizionario dei dati 7](#_Toc499046043)

[Entità 7](#_Toc499046044)

[Relazioni 11](#_Toc499046045)

[Business rules 14](#_Toc499046046)

[Regole di vincolo 14](#_Toc499046047)

[Regole di derivazione 14](#_Toc499046048)

[Ridondanze 14](#_Toc499046049)

[Commenti del diagramma E-R 15](#_Toc499046050)

[Schema Non Ristrutturato 15](#_Toc499046051)

[Area gestione 15](#_Toc499046052)

[Area servizi 15](#_Toc499046053)

[Area Social 16](#_Toc499046054)

[Schema ristrutturato 17](#_Toc499046055)

[Area gestione 17](#_Toc499046056)

[Area servizi 17](#_Toc499046057)

[Area Commercio 18](#_Toc499046058)

[Area Social 18](#_Toc499046059)

[Calendari 18](#_Toc499046060)

[Attività e Obbiettivi 18](#_Toc499046061)

[Individuazione di operazioni sui dati 19](#_Toc499046062)

[Operazione 1 - Svolgimento di un esercizio 19](#_Toc499046063)

[Operazione 2 - Scrittura di una scheda di allenamento (con relative sessioni) 20](#_Toc499046064)

[Operazione 3 - Accesso ad uno dei centri e assegnamento dell’armadietto 20](#_Toc499046065)

[Operazione 4 - Visualizzazione dei post 21](#_Toc499046066)

[Operazione 5 - Ricerca di informazioni riguardanti un corso 23](#_Toc499046067)

[Operazione 6 - Stipulazione di un contratto 24](#_Toc499046068)

[Operazione 7 - Modifica dell’usura di un macchinario 25](#_Toc499046069)

[Operazione 8 - Acquisto di integratori 26](#_Toc499046070)

[Analisi delle prestazioni delle operazioni 27](#_Toc499046071)

[Tavola dei volumi 27](#_Toc499046072)

[Entità 27](#_Toc499046073)

[Relazioni 29](#_Toc499046074)

[Stima della frequenza giornaliera 31](#_Toc499046075)

[Introduzione di ridondanze 32](#_Toc499046076)

[Tavola degli accessi 32](#_Toc499046077)

[Operazione 1: 32](#_Toc499046078)

[Operazione 2: 32](#_Toc499046079)

[Operazione 3: 32](#_Toc499046080)

[Operazione 4: 34](#_Toc499046081)

[Operazione 5: 36](#_Toc499046082)

[Operazione 6: 37](#_Toc499046083)

[Operazione 7: 37](#_Toc499046084)

[Operazione 8: 38](#_Toc499046085)

[Progettazione logica 39](#_Toc499046086)

[Vincoli di integrità referenziale 40](#_Toc499046087)

[Analisi delle dipendenze funzionali e normalizzazione 44](#_Toc499046088)

[dipendenze funzionali 44](#_Toc499046089)

[Normalizzazione 48](#_Toc499046090)

[Implementazione su DBMS Oracle MySQL 49](#_Toc499046091)

[Operazioni 49](#_Toc499046092)

[Operazione 1 49](#_Toc499046093)

[Operazione 2 49](#_Toc499046094)

[Operazione 3 50](#_Toc499046095)

[Operazione 4 51](#_Toc499046096)

[Operazione 5 56](#_Toc499046097)

[Operazione 6 56](#_Toc499046098)

[Operazione 7 60](#_Toc499046099)

[Operazione 8 62](#_Toc499046100)

[Vincoli generici 63](#_Toc499046101)

[Turnazione del personale 63](#_Toc499046102)

[Scheda di allenamento 64](#_Toc499046103)

[Corso 64](#_Toc499046104)

[Accesso 65](#_Toc499046105)

[Scheda alimentazione 66](#_Toc499046106)

[Rata 66](#_Toc499046107)

[Funzionalità 67](#_Toc499046108)

[Aggiornamento crediblità 67](#_Toc499046109)

[Rotazione del magazzino 68](#_Toc499046110)

[Reporting 70](#_Toc499046111)

[Performance sportiva 70](#_Toc499046112)

[Suggerimenti per le cerchie 71](#_Toc499046113)

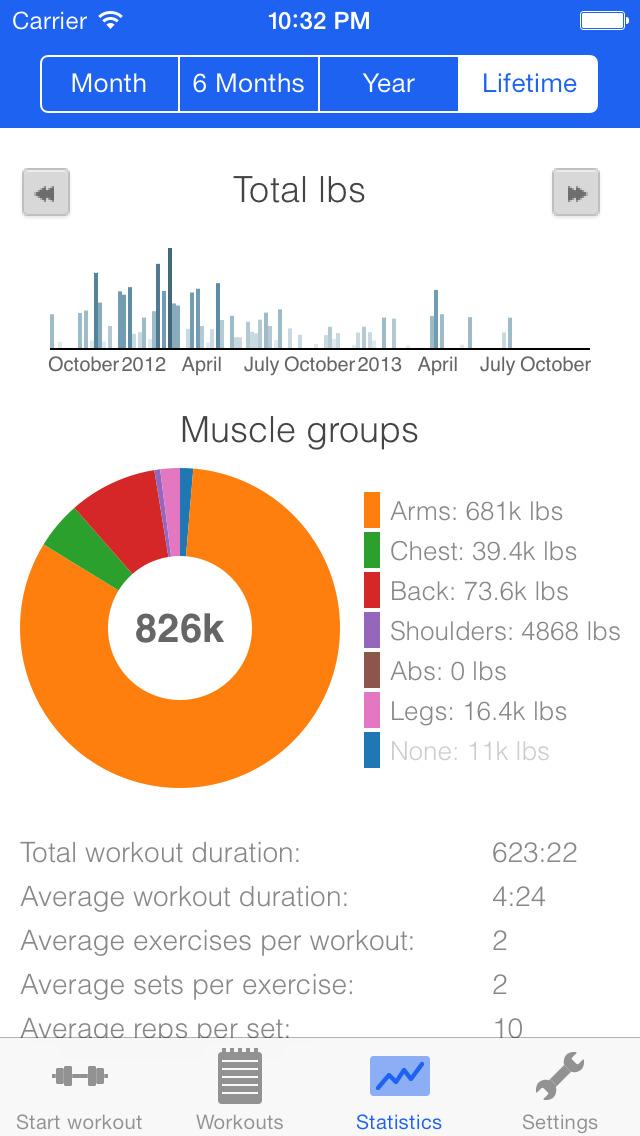
# Analisi delle specifiche

Di seguente sono fornite le descrizioni dettagliate dei punti del progetto lasciati a discrezione degli studenti.

Livello di usura dei macchinari (2.2.1)  
I macchinari dopo un certo periodo di utilizzo devono essere messi in manutenzione prima di poter essere riutilizzati. Un macchinario raggiunge il 100% del proprio livello di usura dopo 2000 ore di utilizzo (equivale a circa 6 mesi di utilizzo). Una volta raggiunto, il macchinario entrerà in uno stato “di manutenzione” e non sarà utilizzabile finché l’intervento non sarà effettuato, quest’ultimo diminuirà la percentuale di usura a livelli accettabili.

Scala dei valori relativi al potenziamento muscolare (2.2.3)  
Il cliente, al momento della sottoscrizione del contratto, nell’atto di indicare gli obiettivi da porre, potrà anche stabilire il livello di potenziamento muscolare per ciascuno dei muscoli target, basandosi su una scala di cinque valori: {minimo, lieve, moderato, elevato, massimo}. Questo permetterà di accogliere un numero di casistiche superiore. Per esempio un cliente, a causa di altri sport praticati, potrebbe avere l’esigenza di effettuare un allenamento molto intenso e specifico per un determinato gruppo muscolare (selezionando quindi “massimo”), un altro cliente invece potrebbe preferire un allenamento leggero, per un gruppo muscolare per cui necessita di effettuare riabilitazione (selezionando quindi “lieve”).

Metrica per la misurazione delle performance del cliente (2.3.6 e 2.5.2)  
Per la misurazione delle performance del cliente durante gli allenamenti saranno presenti diverse metriche, al fine di avere una panoramica completa sulla qualità dei servizi offerti dal centro.

1. ***Fedeltà della frequenza***: percentuale di allenamenti eseguiti sia in eccesso che in difetto rispetto a quelli previsti per ogni settimana.
2. ***Fedeltà della sessione***: percentuale di sessioni eseguite nel giorno indicato dalla scheda di allenamento (il rapporto tra giorni di riposo e di esercizio, scelto dagli istruttori, è fondamentale per un buon allenamento).
3. ***Fedeltà degli esercizi***: percentuale di esercizi svolti rispetto al totale indicato per le singole sessioni.
4. ***Fedeltà di esecuzione***: percentuale di esercizi svolti con un tempo effettivo di esecuzione entro un range del 10% rispetto alla durata indicata sulla scheda di allenamento.
5. ***Battito cardiaco***: grazie al dispositivo indossato dall’utente durante le sessioni di allenamento è possibile ottenere statistiche sul battito cardiaco. Confrontando questi valori con i dati memorizzati dell’utente (età, peso, altezza e sesso) è possibile valutare lo sforzo compiuto.
6. ***Calorie consumate***: basandosi sui dati specifici del cliente (come il peso, il sesso e l’età), sulla durata dell’esercizio svolto e sui dati raccolti dal dispositivo indossato, è possibile fornire una valutazione indicativa delle calorie bruciate.

Tutte le varie misurazioni saranno visualizzabili attraverso l’applicazione per smartphone della palestra.

Se i risultati ottenuti dovessero essere negativi, la palestra dovrà provvedere ad apportare delle modifiche alla scheda di allenamento o a proporre nuove soluzioni per il cliente:

1. ***Fedeltà alla palestra***: questa percentuale è basata sulla media pesata tra *la fedeltà della frequenza*, che influirà per e la *fedeltà della sessione*, che influirà per . Se il valore dovesse scendere **sotto al 60%**, al cliente *verrà proposta una nuova scheda di allenamento* con più o meno giorni rispetto alla precedente. Nel caso, può anche essere suggerito qualche corso inerente agli interessi del cliente, in modo tale da rendergli più divertente ed interessante l’esperienza presso la palestra e aumentare la frequenza di accesso ad essa. Una bassa frequenza è sintomo di probabile abbandono.
2. ***Fedeltà alla scheda***: questa percentuale è basata sulla media pesata tra la *fedeltà degli esercizi*, che influirà per , e la *fedeltà di esecuzione*, che influirà per . Se il valore dovesse scendere **sotto al 60%**, *sarà offerto al cliente un periodo di prova di due settimane con un personal trainer*, nel caso in cui non sia già stato fatto. Egli provvederà a parlare col cliente e, di conseguenza, andare a proporre una scheda di allenamento più adatta alle sue esigenze, oltre che seguirlo nell’esecuzione di quest’ultima per la durata del periodo di prova.
3. ***Sforzo***: questa percentuale si basa sul numero di esercizi eseguiti con un *battito cardiaco* superiore o inferiore a quello previsto e con un *consumo calorico* superiore o inferiore alla media per quel tipo di esercizio. Nel caso in cui dovesse **superare il 30%,** *verrà consigliata al cliente una sostituzione della scheda di allenamento con una più o meno impegnativa*.
4. ***Performance sportiva***: questa percentuale è ottenuta dalla media tra la *fedeltà alla palestra*, la *fedeltà alla scheda* e lo *sforzo*. **Le migliori cinque**, per palestra, valutate su allenamenti di almeno sei mesi ogni anno, *riceveranno in omaggio un abbonamento da sei mesi* (in modo da incoraggiare. Inoltre, **il primo** riceverà il *Bicipite di Platino*, ossia una piccola coppa a forma di bicipite, e potrà aggiungere accanto al suo nome utente sul forum questa icona: 👑.

Politica di attribuzione della credibilità di un utente (2.4.1)  
La politica di attribuzione della credibilità è stata implementata attraverso un attributo ***credibilità*** per ogni cliente, aggiornato periodicamente. Ogni utente potrà esprimere un giudizio da 0 a 5 “💪”, la cui media servirà per calcolare il valore della credibilità totale. Per coloro che avranno una credibilità tra i 4 e i 5 “💪” sarà disponibile l’icona “👍” da affiancare al proprio nome utente durante la pubblicazione dei post ad indicare la loro affidabilità.

Suggerimento per la creazione di una cerchia (2.4.2)  
Le cerchie possono rappresentare varie tipologie di gruppi di interesse. Il suggerimento scatterà, quindi, quando verrà riscontrato qualcosa in comune tra due o più amici:

* Se due o più amici *hanno selezionato una stessa attività* tra i propri interessi durante la personalizzazione iniziale del profilo, verrà suggerita una cerchia “*nome\_attività*”.
* Se due o più amici *hanno stipulato un contratto nella stessa palestra*, verrà suggerita una cerchia “Nella mia palestra”.
* Se due o più amici *frequentano gli stessi corsi*, verrà suggerita una cerchia “*nome\_corso*”.
* Se due o più amici hanno, nella stessa palestra, *le stesse sessioni di esercizi* (con giorni eventualmente diversi), verrà suggerita una cerchia “Gli schedati”.
* Se due o più amici *accedono generalmente nelle stesse fasce orarie* (stessi giorni a orari sovrapposti), verrà suggerita una cerchia “Fuori di fitness”.

Può essere al massimo suggerita una cerchia per tipo di suggerimento alla volta, se la cerchia è già presente viene suggerita l’aggiunta di uno o più nuovi partecipanti ad essa.

Valutazione dello sforzo psicofisico (2.4.3)  
Lo sforzo psicofisico compiuto durante una sfida sarà valutato sulla base dell’andamento della ***performance sportiva:***

1. Se resta *invariata* (variazione sotto il 5%), lo sforzo viene definito “irrisorio”.
2. Se *cala più del 5%* dall’inizio della sfida, si ha uno sforzo “moderato”.
3. Se *cala più del 15%* dall’inizio della sfida, si ha uno sforzo “elevato”.
4. Se *cala oltre il 30%* lo sforzo sarà definito “proibitivo” e il cliente sarà squalificato dalla sfida. Questo permette sia di preservare la salute dei partecipanti che di scoraggiare eventuali sfide troppo impegnative. Tutto ciò va anche incontro agli interessi dell’azienda, evitando infortuni o altre problematiche relative ad allenamenti troppo intensi.

Nel momento in cui uno dei partecipanti completa la sfida (ossia ne raggiunge l’obiettivo), essa termina. A quel punto, viene attribuito a tutti i partecipanti un **malus** sulla percentuale di avanzamento nella sfida *pari al livello dello sforzo psicofisico*. Colui che, *considerato anche il malus*, è arrivato più avanti nella percentuale di avanzamento, **avrà vinto la sfida**. Inoltre, riceverà una *medaglia di bronzo*, da poter affiancare al proprio nome utente durante la pubblicazione dei post. Dopo 10 sfide vinte, sarà disponibile una *medaglia d’argento*. Dopo 50 sfide vinte, sarà disponibile una *medaglia d’oro*. Colui che, durante l’anno, avrà vinto più sfide di tutti, per ogni palestra, sarà riconosciuto come il “*Re delle Sfide*”, e potrà affiancare l’icona accanto al proprio nome utente.

Rotazione del magazzino e promozioni (2.5.3)  
La presenza di offerte e suggerimenti sulla merce nel magazzino si basa su alcune regole:

* **I** **15 prodotti più venduti** appaiono sempre *tra i suggerimenti per il cliente* nell’applicazione per smartphone, e saranno *automaticamente riordinati* nei vari magazzini.
* **I prodotti che hanno la propria scadenza entro un mese** riceveranno uno *sconto del 10%.* Inoltre, se acquistati in coppia con uno dei 15 prodotti più venduti, che saranno suggeriti al momento dell’acquisto, riceveranno uno sconto del 30%.
* **I prodotti la cui scadenza è prevista prima di due settimane**, riceveranno uno *sconto casuale tra il 30% e il 50%,* onde evitare di far capire ai clienti che è prevista una politica sistematica riguardo i prodotti in scadenza. Se acquistati in coppia con uno dei 15 prodotti più venduti, che saranno suggeriti al momento dell’acquisto, riceveranno sempre uno sconto totale del 50%.
* **I 10 prodotti meno venduti** del magazzino *riceveranno uno sconto del 30%* e saranno proposti in simbiosi con altri prodotti più venduti. Questo permetterà anche di mascherare la politica adottata per i prodotti in scadenza, evitando che la clientela distingua la causa dello sconto.

# Progettazione concettuale

## Strategia di progettazione

Per la progettazione concettuale si è deciso di adottare una **strategia mista**, basata sui metodi ***top-down*** e ***bottom-up***. Per prima cosa è stato creato uno *schema scheletro*, individuando i concetti principali*,* che potessero offrire una visione unitaria, seppure astratta, delle specifiche. Per fare ciò ci si è attenuti alla strategia *top-down*. Questo schema, nonostante fosse minimale, conteneva il maggior numero di informazioni non necessitanti di relazioni con altri dati (ad esempio, i dati anagrafici di un cliente). Una volta costruita l’idea alla base dello schema sono stati individuati i settori principali. Essi sono stati poi raffinati uno ad uno, aggiungendo entità, relazioni, attributi, generalizzazioni, cardinalità e identificatori, come previsto dalla strategia *bottom-up*.  
In fine, quando ogni settore era pressoché completo, sono stati collegati, tramite relazioni, tra di loro (come, per esempio, tra l’entità ‘*sfida*’, ‘*scheda di nutrizione*’ e ‘*scheda di allenamento*’), integrando lo schema e ottenendo la progettazione concettuale definitiva.

## Dizionario dei dati

### Entità

|  |  |  |  |
| --- | --- | --- | --- |
| Entità | Attributi | Identificatore(esterno) | Descrizione |
| Accesso | Momento entrata  Momento uscita | Momento Codice fiscale (cliente) | *Informazioni relative all’accesso e all’uscita di un cliente da una palestra.* |
| Acquisto | Data | Codice (prodotto)  Codice fiscale (cliente) | *Acquisto di un integratore da parte di un cliente.* |
| Amicizia | Stato | Codice fiscale (cliente)  Codice fiscale (cliente) | *Amicizia virtuale tra due clienti, lo stato può essere “accettata”, “rifiutata” o “in sospeso”.* |
| Armadietto | Codice identificativo Centro (ridondanza)  Occupato (ridondanza) | Codice identificativo | *Armadietto presente in uno degli spogliatoi.* |
| Attività | Nome | Nome | *Attività di interesse dei clienti.* |
| Attrezzatura | Tipologia | Tipologia | *Tipologia di attrezzatura per l’esecuzione di un esercizio.* |
| Calendario centro | Orario chiusura Orario apertura Giorno settimana | Orario apertura Giorno settimana Codice (centro) | *Orario relativo ad un centro.* |
| Calendario corso | Orario inizio Giorno settimana Durata | Orario inizio Giorno settimana  Nome (sala)  Codice (centro) | *Orario relativo ad un corso.* |
| Centro | Indirizzo  Telefono  Dimensione  Capienza  Codice  Clienti all’interno (ridondanza) | Codice | *Centro fitness gestito dall’azienda.* |
| Cerchia | Nome | Codice fiscale (cliente)  Codice fiscale (cliente) | *Cerchia virtuale in cui sono aggiunti alcuni amici a scelta del cliente.* |
| Cliente | Username  Password  Altezza  Peso  Grassa  Magra  Acqua  Stato  Entità  Nome  Cognome  Documento  Codice  Prefettura  Data nascita  Codice fiscale  Indirizzo | Codice fiscale | *Cliente che ha stipulato un contratto presso uno o più centri.* |
| Contratto | Codice  Prezzo  Data sottoscrizione Accessi settimanali  Durata  Tipo | Codice | *Contratto stipulato tra l’azienda e il cliente.* |
| Corso | Codice  Nome  Livello  Centro (ridondanza)  Data fine  Data inizio | Codice | *Corso tenuto da un istruttore.* |
| Dieta personalizzata | Calorie minime  Calorie massime Codice  Numero pasti  Composizione pasti  Apporto energetico | Codice | *Dieta personalizzata relativa alla scheda di alimentazione di un cliente.* |
| Dipendente | Nome  Cognome  Documento  Codice  Prefettura  Data nascita  Codice fiscale  Indirizzo | Codice fiscale | *Impiegato che lavora presso uno o più centri.* |
| Esercizio | Dispendio energetico  Codice  Nome  Durata recupero  Numero ripetizioni  Numero per ripetizione  Durata ripetizione | Codice | *Esercizio assegnato dal tutor per il cliente.* |
| Esercizio svolto | Timestamp  Battito cardiaco medio  Tempo recupero medio  Tempo impiegato | Timestamp  Codice fiscale (cliente) | *Esercizio concretamente eseguito dal cliente.* |
| Fornitore | Indirizzo  Partita IVA  Nome commerciale  Forma societaria  Numero telefonico | Partita IVA | *Ente che rifornisce il centro di integratori.* |
| Gruppo muscolare | Nome gruppo | Nome gruppo | *Insieme di muscoli.* |
| Integratore | Nome commerciale Sostanza contenuta  Pezzi per confezione Quantità sostanza  Scadenza  Codice | Codice | *Integratore alimentare commercializzato dall’azienda fitness.* |
| Link esterno | Link | Link  Codice post (post) | *Link di contenuti esterni allegato ai post.* |
| Macchinario | % usura  Consumo energetico  Tipologia  Codice identificativo | Codice identificativo | *Attrezzatura o macchinario (di una certa rilevanza) presenti fisicamente nei centri fitness.* |
| Misurazione | Data  % grassa % magra  % acqua | Data  Codice fiscale (dipendente)  Codice fiscale (cliente) | *Misurazione periodica effettuata dal centro per i clienti che ne vogliono usufruire* |
| Obiettivo | Scopo | Scopo | *Obiettivo da raggiungere indicato dal cliente.* |
| Ordine | Stato  Codice interno  Codice esterno  Data consegna preferita Data effettuazione ordine  Data consegna | Codice interno | *Ordine effettuato da un centro fitness.* |
| Pagamento  rateizzato | Importo  Saldato  Numero rate  Istituto  Interesse | Codice (contratto) | *Informazioni sul pagamento a rate di un contratto.* |
| Post | Timestamp  Testo  Codice post | Codice post | *Post scritto da un cliente nel social network della società di fitness.* |
| Rate | Stato  Importo  Scadenza | Scadenza  Codice (contratto) | *Informazioni sul pagamento di una rata specifica.* |
| Sala | Nome | Nome  Codice (centro) | *Sala appartenente ad un centro specifico.* |
| Scheda allenamento | Data inizio  Data fine  Codice | Codice | *Programma degli esercizi assegnati dal tutor al cliente.* |
| Scheda di alimentazione | Intervallo visite  Data Fine  Data Inizio  Codice | Codice | *Scheda assegnata da un medico nutrizionista ad un cliente.* |
| Sessione | Variante  Tipologia | Variante  Tipologia | *Insieme di esercizi da svolgere durante lo stesso accesso al centro.* |
| Sfida | Data  Scopo  Codice sfida | Codice sfida | *Sfida proposta dal cliente proponente.* |
| Spogliatoio | Identificatore  Posizione  Capienza | Identificatore | *Spogliatoio al servizio di un determinato centro.* |
| Turno | Giorno settimana  Orario inizio  Orario fine | Codice fiscale (dipendente)  Orario inizio | *Turni del personale, ossia orari in cui lavora ciascun dipendente.* |

### Relazioni

|  |  |  |  |
| --- | --- | --- | --- |
| Relazione | Componenti | Attributi | Descrizione |
| Acquirente | Acquisto Cliente |  | *Associa ad ogni cliente gli acquisti da lui effettuati.* |
| Allegato | Post Link esterni |  | *Associa ad ogni post i link esterni che vi sono allegati.* |
| Allenamento | Sessione Esercizio |  | *Lega ad una sessione i singoli esercizi che dovrebbero essere svolti.* |
| Area | Centro Sala |  | *Associa le sale al centro a cui appartengono.* |
| Assegnamento | Accesso Armadietto | Codice sblocco | *Associa ad ogni accesso un armadietto (se richiesto dal cliente) con un codice di sblocco monouso.* |
| Consulente | Dipendente Contratto |  | *Associa il dipendente ai contratti che ha stipulato con i clienti.* |
| Consulenza | Dipendente Scheda alimentazione |  | *Associa il dipendente alle schede d’alimentazione che egli ha scritto* |
| Dati raccolti | Esercizio  Svolto |  | *Associa il cliente agli esercizi da lui svolti.* |
| Divisione | Rateizzato Rate |  | *Lega le singole rate ad un pagamento rateizzato.* |
| Dotazione | Sala Attrezzatura |  | *Dislocazione dell’attrezzatura nelle varie sale.* |
| Fornimento | Integratore Fornitore |  | *Indica la disponibilità di integratori da parte dei fornitori.* |
| Giudizio | Cliente Post | Valore Timestamp | *Associa ad ogni utente i giudizi espressi sui vari post.* |
| Inserimento | Amicizia Cerchia | Direzione | *Associa alla cerchia le varie amicizie di ci è composta, direzione indica se la cerchia è del ricevente o del richiedente di un’amicizia.* |
| Interesse | Cliente Attività |  | *Indica le attività di interesse del cliente.* |
| Iscrizione | Cliente Corso |  | *Indica le iscrizioni fatte dai vari clienti per partecipare ai corsi.* |
| Istruttore | Corso Dipendente |  | *Associa ai corsi i dipendenti che svolgono il ruolo di istruttore.* |
| Magazzino | Centro Integratore |  | *Associa ad ogni integratore il centro che li conserva in magazzino.* |
| Medico | Dipendente Misurazione |  | *Associa il medico che esegue le visite alle relative misurazioni.* |
| Negozio | Acquisto Centro |  | *Associa li acquisti al centro in cui vengono eseguiti.* |
| Occupazione | Sala  Calendario corso |  | *Associa alle sale il calendario dei corsi che le occupano.* |
| Orario | Centro Calendario centro |  | *Associa ad ogni centro le rispettive fasce orarie di apertura.* |
| Ordinazione | Ordine Centro |  | *Associa ad ogni ordine il centro che lo ha eseguito.* |
| Partecipante | Cliente Sfida | Avanzamento | *Associa ad ogni utente le sfide lanciate quelle alle quali ha aderito e il suo livello di avanzamento.* |
| Pasto | Scheda alimentazione Dieta personalizzata |  | *Lega ad ogni scheda di alimentazione i dati relativi ai pasti indicati.* |
| Personale | Turno Centro |  | *Associa al centro i turni del suo personale.* |
| Posizione | Armadietto Spogliatoio |  | *Associa ad ogni armadietto il rispettivo spogliatoio.* |
| Possesso | Cliente  Scheda allenamento |  | *Associa ad ogni cliente le proprie schede di allenamento.* |
| Possibilità accesso | Contratto Sala | Accessi settimanali | *Associa ad ogni contratto le sale alle quali può accedere e quante volte.* |
| Potenziamento Muscolare | Potenziamento muscolare  Gruppo muscolare | Livello | *Lega un obiettivo di potenziamento muscolare al gruppo muscolare target e il livello per cui si vuole potenziare.* |
| Prodotto | Integratore Acquisto |  | *Associa ad ogni acquisto gli integratori acquistati.* |
| Prodotto ordinato | Ordine Integratore |  | *Associa ad ogni ordine gli integratori richiesti.* |
| Proponente | Cliente Sfida |  | *Associa colui che propone la sfida alla sfida stessa.* |
| Proprietà | Cliente Scheda alimentazione |  | *Associa il cliente alle sue relative schede li alimentazione.* |
| Pubblicazione | Cliente Post |  | *Associa ad ogni utente i post che ha creato.* |
| Responsabile | Centro Dipendente  Dipendente |  | *Associa ad ogni dipendente un altro dipendente che ne è responsabile all’interno dell’organigramma di un centro.* |
| Responsabile sala | Sala Dipendente |  | *Associa ad ogni sala il suo responsabile.* |
| Ricezione | Amicizia Cliente |  | *Associa al cliente le richieste di amicizia ricevute.* |
| Richiesta | Amicizia Cliente |  | *Associa al cliente le richieste d’amicizia chieste.* |
| Risposta | Post Post |  | *Associa ad ogni post altri post che sono in risposta ad esso.* |
| Saldo | Contratto Pagamento rateizzato |  | *Associa ad ogni contratto le informazioni sul pagamento rateizzato.* |
| Scelta | Contratto Obbiettivo |  | *Associa al contratto gli obbiettivi scelti dal cliente al momento della sottoscrizione.* |
| Scheda alimentazione associata | Sfida Scheda alimentazione |  | *Associa alla scheda di alimentazione le sfide a cui essa è associata.* |
| Scheda allenamento associata | Sfida Scheda allenamento |  | *Associa alla scheda d’allenamento le sfide a cui essa è associata.* |
| Scrittura | Dipendente Scheda allenamento |  | *Associa il dipendente alle schede d’allenamento da lui scritte.* |
| Sede | Contratto Centro |  | *Associa ad ogni contratto i centri ai quali ha accesso.* |
| Segnalazione | Cliente  Accesso |  | *Associa ad ogni cliente gli accessi eseguiti.* |
| Servizio | Spogliatoio Centro |  | *Associa ad ogni spogliatoio il centro nel quale è situato.* |
| Sottoscrizione | Cliente Contratto |  | *Associa il cliente ai contratti che egli ha sottoscritto.* |
| Suddivisione | Scheda allenamento Sessione | Giorno settimana | *Indica le sessioni nelle quali le schede di allenamento sono suddivise e il giorno in cui vanno eseguite.* |
| Supporto | Esercizio Attrezzatura | Configurazione | *Associa ad ogni esercizio l’attrezzatura necessaria acciocché possa essere svolto.* |
| Lezione | Sala Corso |  | *Associa ad ogni corso il centro nel quale viene svolto e le fasce orarie.* |
| Svolgimento | Esercizio svolto Esercizio |  | *Associa l’esercizio ai suoi svolgimenti da parte dei clienti.* |
| Thread | Sfida Post |  | *Associa ad ogni sfida i post legati ad essa.* |
| Turnazione | Dipendente Turno |  | *Associa al dipendente i turni che deve svolgere.* |
| Tutor | Cliente Dipendente  Scheda allenamento |  | *Associa ad ogni cliente un tutor personale.* |
| Ubicazione | Accesso Centro |  | *Associa ad ogni accesso il centro nel quale è avvenuto.* |
| Utilizzo | Esercizio svolto Attrezzo reale | Configurazione | *Associa all’esercizio svolto l’eventuale attrezzo utilizzato con la configurazione adottata.* |
| Visita | Misurazione Cliente |  | *Associa il cliente alle misurazioni effettuate su di egli.* |
|  |  |  |  |

## Business rules

### Regole di vincolo

1. Gli orari di un corso devono essere conformi con gli orari del dipendente che ne è responsabile.
2. Ciascun dipendente lavora per un massimo di 8 ore al giorno.
3. Non devono esistere fasce orarie di uno stesso giorno in cui al dipendente sono assegnati centri diversi.
4. Un dipendente non può tenere due corsi in due sale differenti nelle stesse fasce orarie.
5. Il numero di accessi contemporanei ad un centro non deve superare la capienza massima di quel centro.
6. Il numero di accessi del cliente non deve superare il numero di accessi previsti dal contratto (sia accessi totali ai centri, sia accessi specifici per le sale).
7. Un cliente non deve poter accedere ad una sala che non è prevista nella propria tipologia di contratto.
8. Non deve esistere un accesso da parte di un cliente il cui contratto è scaduto.
9. Ciascun centro deve essere dotato di almeno una piscina.
10. Solo un medico nutrizionista può assegnare una scheda di alimentazione ad un cliente.
11. Un integratore può essere ordinato soltanto se esiste un fornitore che ha disponibilità di quell’integratore.
12. Ogni sfida può avere un solo cliente in vece di *proponente*.
13. In una sfida un utente può associare solo una scheda di allenamento o di nutrizione che gli appartiene, non quella di un altro cliente (privacy).

### Regole di derivazione

1. La quantità di scorta di un magazzino rispetto ad un certo integratore deve essere uguale al numero di ordinazioni fatte da quel centro meno il numero di acquisti eseguiti dai clienti.
2. La somma degli importi delle rate deve essere uguale all’importo totale del pagamento (incrementato dell’interesse dell’istituto).

## Ridondanze

* Il concetto espresso dall’associazione ‘Possesso’ è incluso all’interno dell’associazione ‘Tutor’.
* Possiamo essere a conoscenza della scorta del magazzino anche attraverso il numero di ordini eseguiti e di acquisti fatti dagli utenti (regola di derivazione 1).

# Commenti del diagramma E-R

## Schema Non Ristrutturato

### Area gestione

#### Organizzazione e strutture

La sezione Organizzazione e Strutture è stata suddivisa nelle entità *Centro*, *Calendario centro* (sotto-entità di *Calendario*), *Sala* e *Attrezzatura*. Scelte significative in questo settore sono state quella di considerare le piscine come una sotto-entità delle sale, inoltre le istanze di *Attrezzatura* rappresentano una singola attrezzatura presente fisicamente in un centro e perciò è stato aggiunto un codice identificativo come chiave primaria di ognuna di esse. È stato inoltre aggiunto un codice come chiave primaria per l’entità *Centro*.

#### Anagrafica clienti e dipendenti

L’anagrafica dei clienti e dipendenti è stata rappresentata come una generalizzazione avente *Anagrafica* come entità padre e *Dipendente* e *Cliente* come entità figlie. La chiave primaria scelta è il *codice fiscale*, naturalmente univoco sia per i dipendenti che per i clienti. I responsabili sono stati rappresentati tramite una relazione a tre che unisce *Centro* e l’entità *Dipendente* a sé stessa: è stato pensato che un dipendente che lavora in centri diversi potrebbe avere un responsabile diverso in ogni centro.  
Per rappresentare le varie attività che i dipendenti svolgono nei centri si è introdotta una relazione tripla *Personale* con attributo *Attività* che unisce *Centro*, *Dipendente* e *Turnazione* (una sotto-entità di calendario).

#### Contratti

La gestione dei contratti è stata implementata attraverso una generalizzazione avente l’entità *Contratto* come padre e le entità *Standard*, *Personalizzato* e *Giornaliero* come sotto-entità, è stata introdotto come chiave un codice a *Centro*. La proprietà di un contratto di poter essere multi-sede è stata rappresentata con una cardinalità (1,3) con la relazione *Sede* che associa *Contratto* a *Centro*. Le *possibilità di accesso* alle sale sono state rappresentate con una relazione omonima che associa *Contratto* a *Sala* e presenta un attributo *Accessi Settimanali*.  
Il pagamento di un contratto è stato rappresentato con una generalizzazione avente *Pagamento* come entità padre e *Singolo* e *Rateizzato* come figlie. La sotto-entità *Rateizzato* è legata all’entità *Rata* tramite la relazione *Divisione*. Gli obiettivi che si intendono raggiungere sono rappresentati mediante una generalizzazione avente come padre *Obiettivo* e come entità figlia *Potenziamento* *Muscolare* legata attraverso la relazione *Potenziamento* (avente l’attributo *Livello*) a *Gruppo Muscolare*. L’attività di tutoring è stata rappresentata attraverso una relazione *Tutor* tra *Cliente* e *Dipendente*.

### Area servizi

#### Scheda di allenamento

La sezione *Scheda di Allenamento* è stata suddivisa nelle entità *Scheda Allenamento* e *Sessione* e nella generalizzazione avente come padre *Esercizio* e come sotto-entità *a Tempo* e *con Ripetizioni*.  
Ogni sessione rappresenta una serie di esercizi consigliati, da eseguire in un accesso alla palestra.  
Alla sotto-entità *con Ripetizione* è stato aggiunto l’attributo *per Ripetizione*, in modo da indicare il numero di trazioni, flessioni o simili, da eseguire per ogni ripetizione. La configurazione e l’attrezzatura consigliati per ogni esercizio sono rappresentati tramite la relazione *Supporto* avente come attributo *Configurazione*.

#### Scheda di alimentazione

La sezione Scheda di Alimentazione è stata rappresentata attraverso le entità *Scheda Alimentazione*, *Dieta Personalizzata* e *Misurazione*. Ad ogni dieta è stato assegnato un codice come chiave primaria e ogni istanza è stata pensata come una dieta appositamente studiata per un cliente e quindi diversa dalle altre. Le schede di alimentazione sono associate all’obiettivo da raggiungere con esse mediante la relazione *Fine*. L’entità *Misurazione* contiene le misurazioni periodiche effettuate ai clienti ed è associata ad essi e al dipendente che la esegue mediante le relazioni *Visita* e *Medico*.

#### Accessi

La sezione Accessi è stata suddivisa nelle entità *Accesso*, *Armadietto* e *Spogliatoio*. L’accesso è associato al centro e al cliente che l’ha eseguito tramite le relazioni *Segnalazione* e *Ubicazione*, ad ogni accesso viene assegnato al cliente un armadietto e fornito un codice momentaneo di sblocco valido fino all’uscita del cliente dalla palestra.

#### Corsi

La sezione Corsi è stata rappresentata mediante le entità *Corso* e la sotto-entità *Calendario* *Corso*. Le iscrizioni sono rappresentate attraverso la relazione *Iscrizione* che associa il corso al cliente, l’associazione a tre *Svolgimento* collega il corso alla sala e agli orari in cui è svolto.

#### Turnazione e Allenamenti e monitoraggio smart

Nello schema entità-relazione queste due sezioni sono rappresentate con la sotto-entità *Turno* e l’entità *Esercizio Svolto*. *Esercizio Svolto* raccoglie tutti i dati sullo svolgimento effettivo di un esercizio da parte di un cliente ed è associata mediante la relazione *Misurazione* al cliente e all’esercizio che sono stati monitorati. La sotto-entità *Turno* raccoglie le informazioni sui turni che svolgono i dipendenti dei vari centri.

#### Integratori

La sezione Integratori è suddivisa nelle entità *Fornitore*, *Ordine*, *Acquisto* e *Integratore*. L’entità *Integratore* rappresenta una singola confezione fisica di integratore. L’ordine effettuato dal centro è rappresentato con un’entità associata agli integratori acquistati, al fornitore di tali integratori e al centro. Gli integratori disponibili nel centro sono rappresentati mediante una relazione *Magazzino* tra *Integratore* e *Centro*. L’acquisto di un cliente è rappresentato tramite un’entità *Acquisto* associata all’integratore acquistato, al centro dove viene effettuato l’acquisto e al cliente.

### Area Social

#### Post

La sezione Post è stata suddivisa nelle entità *Post* e *Link Esterno*.  
Le risposte ai post sono state rappresentate attraverso una relazione *Risposta* che collega l’entità *Post* a sé stessa. I giudizi sono stati rappresentati attraverso una relazione *Giudizio* contenente un attributo valore che collega chi ha giudicato il post al post stesso.  
*Credibilità*, *Password* e *Username* sono stati assegnati come attributi a *Cliente*.

#### Amicizia e cerchie

L’amicizia è stata rappresentata come un’entità che è collegata mediante due relazioni (*Ricezione* e *Richiesta*) al cliente. Il sistema di cerchie è stato rappresentato tramite un’entità *Cerchia* associata a *Amicizia* tramite *Inserimento*.

#### Sfide

Le sfide sono state rappresentate attraverso un’entità *Sfida*. L’entità e collegata attraverso due relazioni *Scheda Alimentazione Associata* e *Scheda Allenamento Associata* rispettivamente alla scheda di alimentazione e a quella di allenamento da associare alla sfida, il thread della sfida è rappresentato come una relazione che collega i post alla sfida. I partecipanti sono associati tramite una relazione omonima.

## Schema ristrutturato

La ristrutturazione dello schema entità-relazione ha portato alla suddivisione dello schema in altre tre aree, oltre che alle altrettante aree originali. L*’Area Commercio* raccoglie le entità elle relazioni che rappresentano tutto ciò che riguarda la compravendita di integratori. La sezione *Attività e Obbiettivi* raccoglie le entità e relazioni che rappresentano obbiettivi e attività di interesse di un cliente e che sono al confine tra le tre aree originarie. La sezione *Calendari* contiene le entità e relazione che rappresentano i calendari di corsi e centri oltre che la turnazione del personale, che sono un nesso tra l’*Area* *Servizi* e l’*Area* *Gestione*.

### Area gestione

#### Organizzazione e strutture

Questa sezione è stata ristrutturata nelle entità *Centro*, *Macchinario*, *Sala* e *Calendario Centro*.  
La generalizzazione *Sala*🡨*Piscina* è stata accorpata in un’unica entità *Sala*. L’entità figlia *Calendario Centro* è stata divisa dalla generalizzazione a cui apparteneva e indica gli orari settimanali del centro. L’entità *Attrezzatura* è stata divisa in *Macchinario*: che rappresenta i macchinari fisicamente presenti nelle palestre, e in *Attrezzatura* (descritta nel paragrafo Scheda di Allenamento).

#### Anagrafica clienti e dipendenti

La generalizzazione che costituiva l’anagrafica di clienti e dipendenti è stata suddivisa in due entità distinte *Dipendente* e *Cliente*. La relazione *Responsabile* è rimasta invariata in quanto rappresenta una corretta associazione tripla. La relazione *Personale* è stata suddivisa in due relazioni distinte *Personale* e *Turnazione* che associano rispettivamente *Centro* e *Dipendente* all’entità *Turno*.

#### Contratti

La generalizzazione *Contratto*🡨*Standard-Personalizzato-Giornaliero* è stata unita in un'unica entità *Contratto* le cui istanze rappresentano i singoli contratti sottoscritti da ogni cliente, ognuno con un codice differente. La relazione quadrupla *Stipulazione* è stata suddivisa in tre relazioni distinte *Scelta*, *Sottoscrizione* e *Consulente* che collegano separatamente *Centro* a: *Obiettivo*, *Cliente* e *Dipendente*.  
La generalizzazione *Pagamento*🡨*Singolo-Rateizzato* è stata ridotta nella singola entità *Pagamento* *Rateizzato* e l’attributo *Saldato* è stato spostato nell’entità *Contratto*.

### Area servizi

#### Scheda di allenamento

La generalizzazione *Esercizio*🡨*a Tempo-con Ripetizioni* è stata unita in un'unica entità *Esercizio* contenente tutti gli attributi della generalizzazione. È stata introdotta l’entità *Attrezzatura* che va ad indicare solamente la tipologia di attrezzatura ed è associata a esercizio tramite la relazione *Supporto* che possiede un attributo *Configurazione* indicante la configurazione necessaria per l’attrezzatura in ogni esercizio.

#### Scheda di alimentazione

La relazione quadrupla *Assegnamento* è stata suddivisa nelle tre relazioni *Fine*, *Consulenza* e *Proprietà* che legano la scheda di alimentazione agli obiettivi, al nutrizionista che l’ha redatta e al cliente che la possiede.

#### Accessi

È stato introdotto l’attributo *Uscita* in accesso che indica se un cliente è uscito dal centro dopo un accesso.

#### Corsi

La relazione tripla *Svolgimento* è stata suddivisa in due relazioni separate *Svolgimento* e *Occupazione* che collegano il corso e la sala al calendario del corso.

#### Allenamenti e monitoraggio smart

Sono stati introdotti gli attributi *Calorie Consumate*, *Battito Cardiaco Medio* e *Tempo di Recupero Medio*, in modo da poter memorizzare dati utili per il monitoraggio delle prestazioni. La relazione tripla *Misurazione* è stata suddivisa in *Utilizzo*, *Raccolta Dati* e *Svolgimento* che collega l’esercizio svolto all’attrezzatura utilizzata, all’esercizio svolto e al cliente che ha svolto l’esercizio.

### Area Commercio

La relazione tripla *Fornimento* che legava *Fornitore* con *Ordine* e *Integratore* è stata eliminata in favore di una relazione *Fornimento*, che collega il fornitore al solo integratore, e *Prodotto Ordinato*, che lega l’ordine all’integratore ordinato. La chiave primaria di ordine è stata modificata nel singolo codice interno.

### Area Social

#### Post

È stato introdotto un attributo *Timestamp* per la relazione *Giudizio* in modo da poter permettere il calcolo della credibilità.

#### Amicizia e cerchie

È stato introdotto un attributo *Stato* indicante lo stato (accettata, non accettata, rifiutata) di un’amicizia. È stata introdotto l’attributo *Direzione* alla relazione *Inserimento* che indica se la cerchia sia del richiedente o del ricevente.

#### Sfide

È stata introdotta una relazione *Proponente* che collega il proponente di una sfida alla sfida.

### Calendari

La generalizzazione *Calendario*🡨*Calendario Centro-Calendario Corso-Turnazione* è stata suddivisa nelle tre entità *Calendario Centro*, *Calendario Corso* e *Turno*.

### Attività e Obbiettivi

La generalizzazione *Obiettivo*🡨*Potenziamento Muscolare* è stata unita nella sola entità *Obiettivo*, *Potenziamento Muscolare* si è trasformato in una relazione tra *Obiettivo* e *Gruppo Muscolare*.

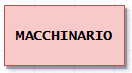
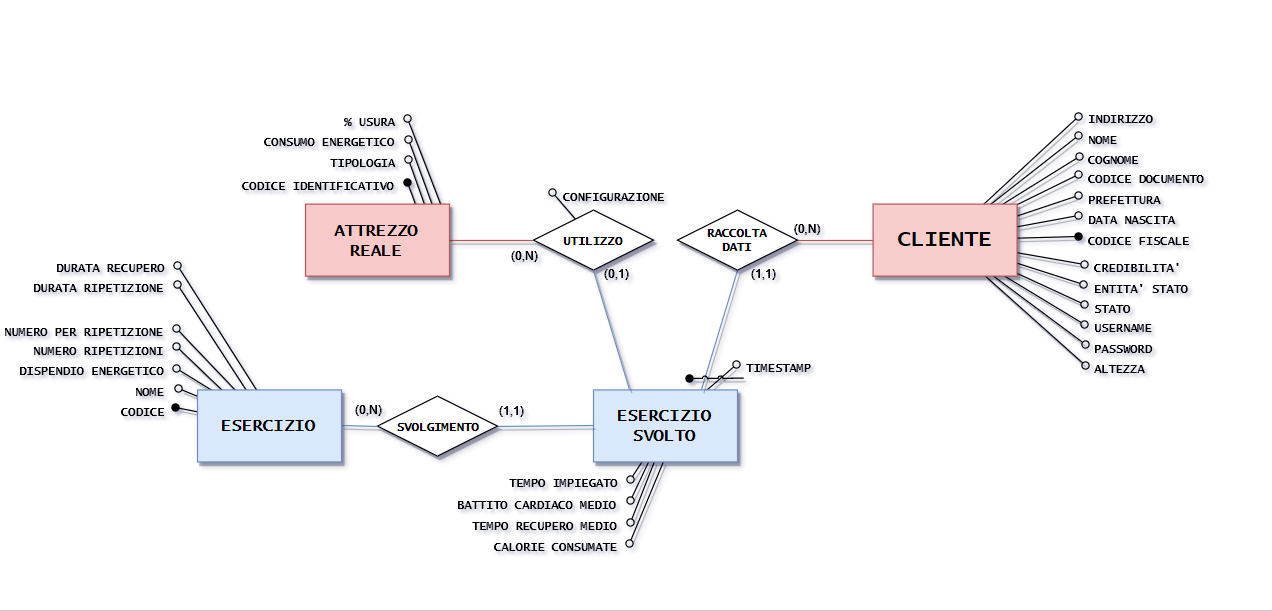
# Individuazione di operazioni sui dati

Le seguenti otto operazioni sono state individuate, mediante l’analisi delle *tavole dei volumi,* in modo tale da rappresentare la maggior parte del normale carico applicativo del database. Grazie alla loro ottimizzazione e all’opportuno uso di ridondanze si è cercato di ridurne al minimo l’impatto sulle prestazioni della base di dati, per renderla più efficace ed efficiente.

Operazione 1 - Svolgimento di un esercizioOgni volta che un esercizio viene svolto da un utente, esso deve essere memorizzato all’interno del database con i relativi dati, quali: numero di ripetizioni, numero di serie, durata, etc.

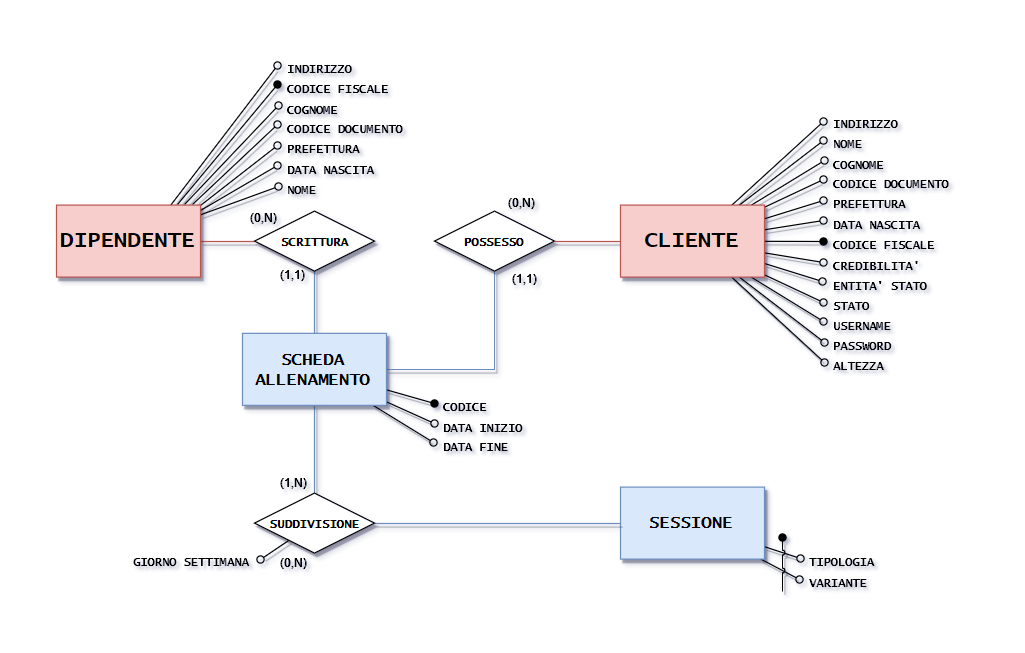
* Questa operazione riceve in **input** i dati relativi allo svolgimento di un esercizio: *il codice dell’esercizio svolto*, *il cliente* che lo sta eseguendo, *le calorie consumate*, *il battito cardiaco medio* durante lo svolgimento, *il tempo impiegato*, *il recupero medio* tra le serie eseguite, *la configurazione* con cui era stata settata la macchina e *il codice dell’attre*zzatura (le statistiche sono fornite dal *braccialetto smart* che ha memorizzati l’altezza, il peso e l’età del cliente in modo da poter eseguire i calcoli sulle calorie consumate in base al tipo di esercizio svolto e al suo svolgimento).
* L’operazione fornisce in **output** una nuova istanza per l’entità *Esercizio Svolto* e le relazioni *Svolgimento*, *Raccolta Dati* e *Utilizzo*, quest’ultima solo se viene utilizzato un macchinario.

Entità coinvolte nell’operazione 1



|  |  |  |  |
| --- | --- | --- | --- |
| Nome | E/R | Numero istanze | Motivazione |
| Cliente | E | 80.000 | *Ogni palestra ha un’utenza media di circa 2.000 clienti, di cui non tutti necessariamente attivi* |
| Raccolta dati | R | 276.480.000 | *(1,1) con esercizio svolto.* |
| Esercizio Svolto | E | 276.480.000 | *Di ogni esercizio viene salvato lo svolgimento. Un cliente medio esegue 2 sessioni a settimana in palestra per 6 mesi (contando eventuali allenamenti saltati) ed esegue 8 esercizi a sessione.* |
| Svolgimento | R | 276.480.000 | *(1,1) con esercizio svolto.* |
| Utilizzo | R | 138.240.000 | *In media un esercizio svolto su due ha richiesto l’ausilio di un macchinario o attrezzo particolare.* |

Operazione 2 - Scrittura di una scheda di allenamento (con relative sessioni)Il caricamento di una scheda di allenamento specifica per un determinato cliente da parte di un istruttore, con le relative sessioni di allenamento da seguire, è un’operazione piuttosto frequente e che comporta diverse scritture.

* Questa operazione riceve in **input** *la data di inizio* della scheda, *il dipendente* che la scrive, **il cliente** che la possiede, *le sessioni* che la compongono e *il giorno in cui eseguirle*.
* L’operazione fornisce in **output** *una nuova* *istanza* per l’entità *Scheda Allenamento* e per le relazioni *Scrittura*, *Possesso* e *Suddivisione*.

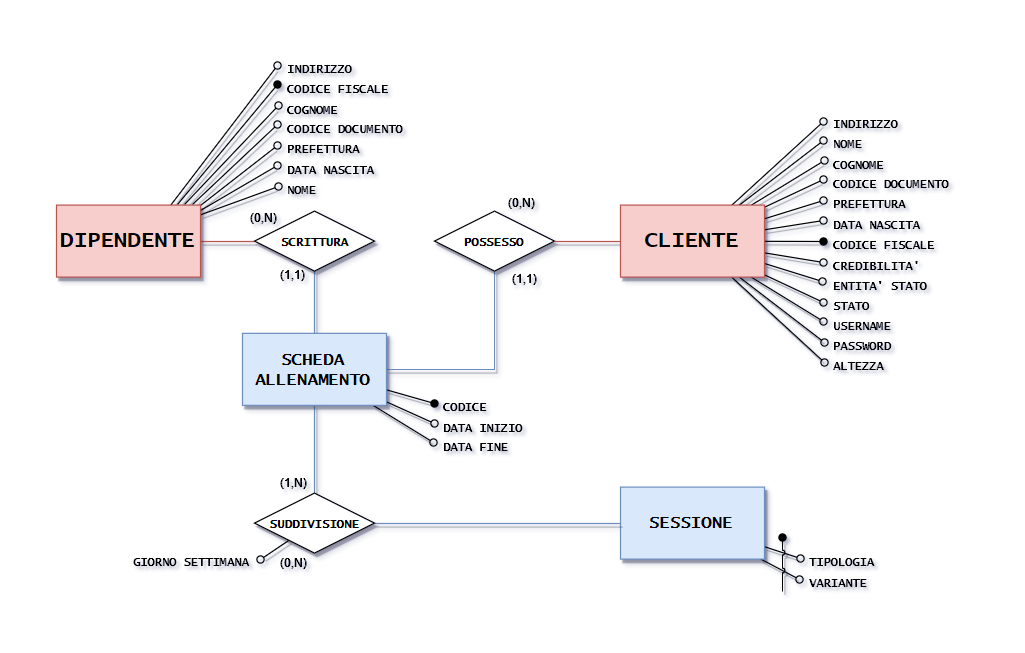
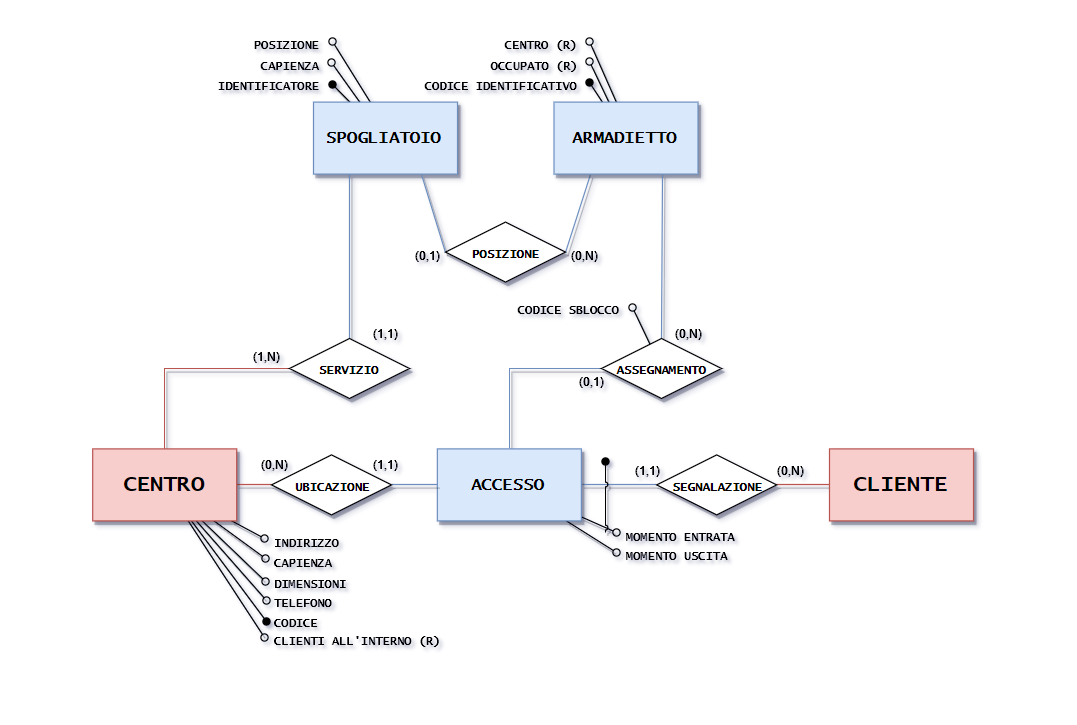
Entità coinvolte nell’operazione 2

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | E/R | Numero istanze | Motivazione |
| Scheda allenamento | E | 240.000 | *Ogni cliente possiede in media 3 schede di allenamento, in quanto un’iscrizione di media durata è di circa 6 mesi e una scheda si cambia ogni 2.* |
| Scrittura | R | 240.000 | *(1,1) con scheda allenamento.* |
| Possesso | R | 240.000 | *(1,1) con scheda allenamento.* |
| Suddivisione | R | 720.000 | *Si ipotizzano 3 sessioni per scheda.* |

Operazione 3 - Accesso ad uno dei centri e assegnamento dell’armadiettoL’accesso ad un centro comporterà automaticamente, per chi avesse richiesto tale servizio, l’assegnamento di un armadietto in uno degli spogliatoi del centro, con relativo codice monouso che permette lo sblocco solo al proprietario del braccialetto con cui è stato fatto l’accesso. L’accesso ad un centro perciò richiederà come operazione base la ricerca di un armadietto libero nel centro d’accesso, tale operazione può eventualmente essere raffinata assegnando un armadietto in uno spogliatoio vicino ad una delle sale più frequentate dal cliente che accede, o assegnandolo in modo da distribuire omogeneamente i clienti nei vari spogliatoi.

* Questa operazione riceve in **input** il *timestamp* *del momento di accesso* al centro, *il cliente* che effettua l’accesso e *il centro* in cui è stato effettuato.
* L’operazione fornisce in **output** *la prima occorrenza di un armadietto libero* all’interno del centro d’accesso con *il relativo codice di sblocco* generato casualmente.

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | E/R | Numero istanze | Motivazione |
| Accesso | E | 3.840.000 | *Un cliente medio accede 2 volte a settimana per 6 mesi.* |
| Segnalazione | R | 3.840.000 | *(1,1) con accesso.* |
| Ubicazione | R | 3.840.000 | *(1,1) con accesso.* |
| Assegnamento | R | 1.280.000 | *A un accesso su 3 viene assegnato un armadietto* |
| Armadietto | E | 3.600 | *Ogni spogliatoio in media possiede 20 armadietti* |
| Servizio | R | 180 | *(1,1) con spogliatoio.* |
| Posizione | R | 3.600 | *(1,1) con armadietto.* |



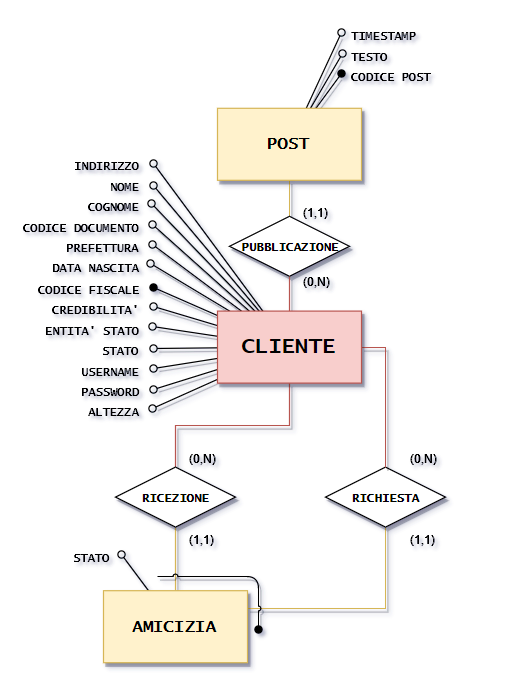
Entità coinvolte nell’operazione 3

Operazione 4 - Visualizzazione dei postAttraverso l’applicazione per smartphone, l’utente potrà accedere all’area del forum e visualizzare i post pubblicati con le relative risposte, rispondere ad altri post, pubblicare nuovi post, visualizzare la credibilità degli utenti ed esprimere dei giudizi riguardanti i post letti.

La più rilevante tra le operazioni inerenti al forum è la ricerca dei post scritti nelle ultime 24 ore dagli amici di un utente, che spesso può essere effettuata per la strutturazione della home page all’accesso al forum.

* Questa operazione riceve in **input** *il cliente* che vuole ricercare i post degli amici.
* L’operazione fornisce in **output** *la lista dei post pubblicati dagli amici*, *nelle ultime 24 ore*, in ordine di tempo con *le relative informazioni*: codice del post, utente che lo ha pubblicato, timestamp di pubblicazione, testo, eventuali allegati e risposte.

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | E/R | Numero istanze | Motivazione |
| Cliente | E | 80.000 | *Ogni palestra ha un’utenza media di circa 2.000 clienti, di cui non tutti necessariamente attivi.* |
| Richiesta | R | 800.000 | *(1,1) con amicizia.* |
| Ricezione | R | 800.000 | *(1,1) con amicizia.* |
| Amicizia | E | 800.000 | *In media ogni cliente ha 20 amicizie, ogni amicizia comprende due clienti.* |
| Pubblicazione | R | 1.600.000 | *(1,1) con post.* |
| Post | E | 1.600.000 | *In media ogni utente scrive 20 post comprese le risposte.* |
| Cliente | E | 80.000 | *Ogni palestra ha un’utenza media di circa 2.000 clienti, di cui non tutti necessariamente attivi.* |



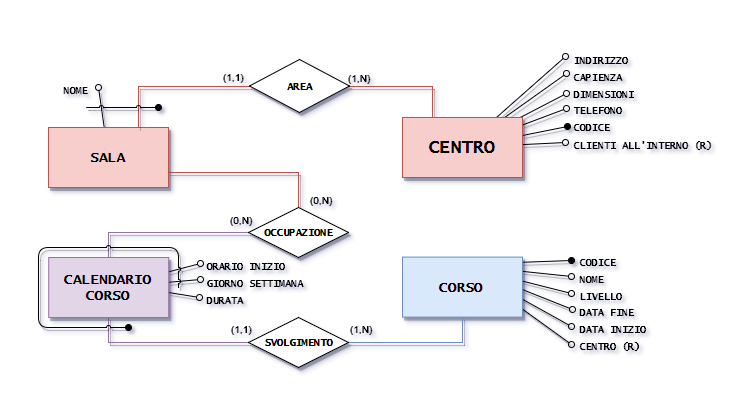
Entità coinvolte nell’operazione 4

Esempio di come potrebbe essere strutturato il forum

Operazione 5 - Ricerca di informazioni riguardanti un corsoOgni cliente, attraverso l’applicazione, può accedere all’elenco dei corsi presenti nei centri per poterne vedere le relative informazioni, come l’attività del corso, le fasce orarie e la sala in cui si svolge.

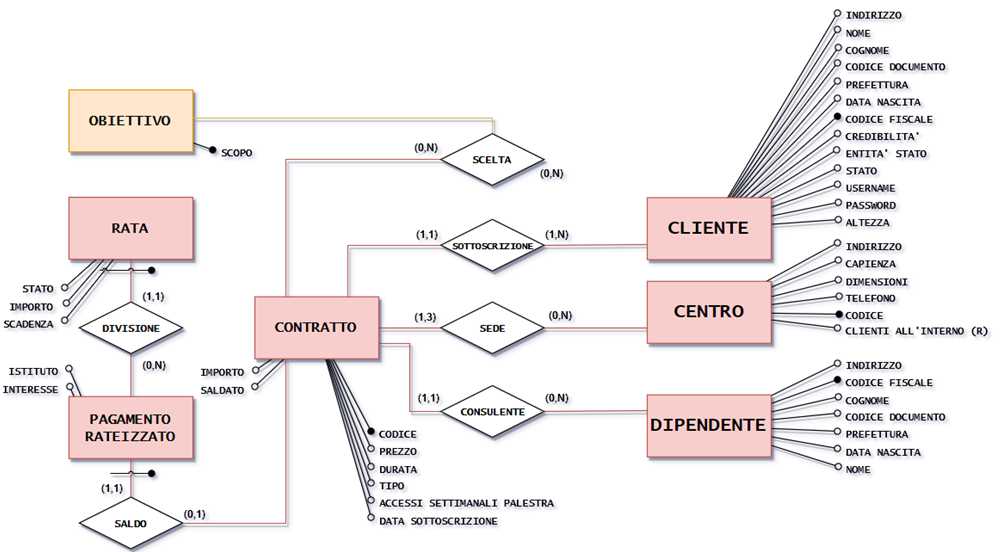
* Questa operazione riceve in **input** il *centro* di cui si vogliono ricercare i corsi.
* L’operazione fornisce in **output** tutte *le informazioni relative ad ogni corso* che viene svolto nel centro comprese *le sale che utilizza* e *il calendario* del corso.

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | E/R | Numero istanze | Motivazione |
| Centro | E | 40 | *Ipotesi sul numero di centri gestiti da una grossa azienda che opera sul territorio. In Italia sono attive circa 4.000 palestre e si suppone che una grossa azienda ne possieda circa l’1%.* |
| Area | R | 320 | *(1,1) con sala.* |
| Occupazione | R | 2.160 | *Una sessione di un corso su 5 occupa in media due sale, mentre le restanti in media una.* |
| Calendario corso | E | 1.800 | *Un corso viene eseguito in media 3 volte a settimana.* |
| Lezione | R | 1.800 | *(1,1) con calendario corso.* |
| Corso | E | 1.000 | *Ogni palestra offre in media 25 corsi differenti, come spinning, arti marziali, zumba, yoga, pilates, allenamenti di vario tipo e corsi sportivi.* |



Entità coinvolte nell’operazione 5

Operazione 6 - Stipulazione di un contrattonell’atto di stipulare un contratto, il cliente sceglie la tipologia di contratto, i centri in cui poter accedere, i relativi accessi alle sale, il numero di accessi settimanali, il metodo di pagamento, gli obiettivi posti (e, nel caso di potenziamento muscolare, il livello e i gruppi muscolari interessati), username e password per il profilo riguardante la parte social. Essendo la stipulazione un’operazione abbastanza frequente (necessaria anche per i rinnovi) e, con tutto ciò che comporta, molto pesante, è stata ritenuta interessante.

* Questa operazione riceve in **input** tutte *le informazioni necessarie alla compilazione del contratto,* compresa *l’eventuale scelta di pagamento rateizzato* e *gli obiettivi da raggiungere.*
* L’operazione fornisce in **output** *una nuova istanza* per l’entità *Contratto* e per le relazioni *Sede* e *Firmatario*, eventualmente viene creata una nuova istanza per *Pagamento* *Rateizzato*, nuove istanze per *Rata*, nuove istanze per *Divisione*, nuove istanze per *Scelta*, *Obiettivo* e *Possibilità di Accesso*.

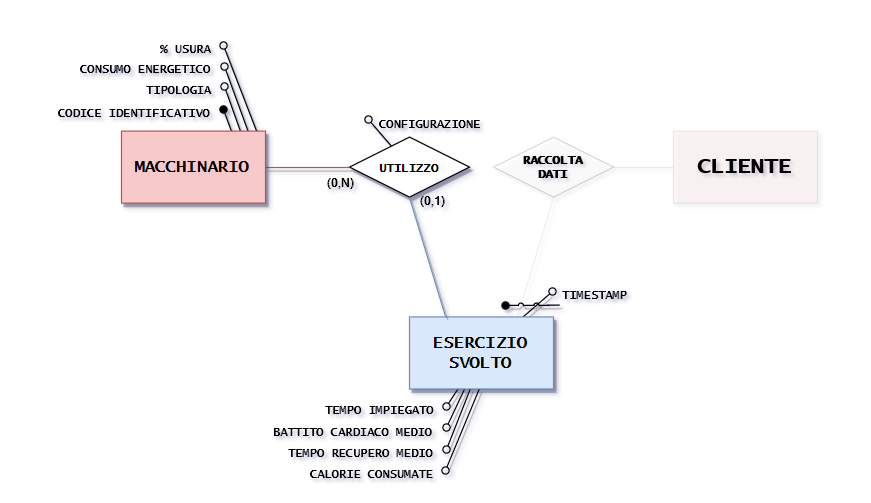
Entità coinvolte nell’operazione 6

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | E/R | Numero istanze | Motivazione |
| Contratto | E | 160.000 | *In media vengono stipulati 2 contratti a cliente.* |
| Consulente | R | 160.000 | *(1,1) con contratto.* |
| Sottoscrizione | R | 160.000 | *(1,1) con contratto.* |
| Scelta | R | 160.000 | *(1,1) con contratto.* |
| Sede | R | 192.000 | *In media un contratto su cinque è multisede e in tal caso ne menziona due.* |
| Possibilità accesso | R | 640.000 | *In media ogni contratto garantisce l’accesso a 4 sale.* |
| Pagamento rateizzato | E | 40.000 | *Ogni contratto necessita di un pagamento* |
| Saldo | R | 40.000 | *(1,1) con Pagamento Rateizzato.* |
| Rata | E | 120.000 | *Un contratto su quattro è suddiviso in 3 rate.* |
| Divisione | R | 120.000 | *(1,1) con rata.* |

Operazione 7 - Modifica dell’usura di un macchinarioOgni volta che un macchinario viene usato, la percentuale di usura deve essere aggiornata come descritto nella sezione “*Analisi delle specifiche*”. Questa operazione comporta un conteggio del tempo passato dai clienti durante l’utilizzo dei macchinari, e l’adattamento dello stato di manutenzione una volta superate le 2000 ore di utilizzo totali. Tale aggiornamento viene eseguito una volta a settimana per ogni macchinario e il cambiamento della percentuale di usura viene eseguito sommando alla percentuale attuale quella ottenuta dalla percentuale tra 2.000 ore e le ore per cui è stato usato il macchinario nella settimana passata.

* Questa operazione riceve in **input** *il macchinario di cui si vuole modificare l’usura*.
* L’operazione fornisce in **output** *l’aggiornamento del campo usura dell’entità Macchinario*.

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | E/R | Numero istanze | Motivazione |
| Macchinario | E | 2.800 | *Ogni palestra possiede in media 70 attrezzature differenti, tra macchinari di vario genere, attrezzi particolari per esercizi (senza contare pesi, palle ergonomiche e altre attrezzature di minor rilevanza) e attrezzature sportive.* |
| Utilizzo | R | 138.240.000 | *In media un esercizio svolto su due ha richiesto l’ausilio di un macchinario o attrezzo particolare.* |
| Esercizio Svolto | E | 276.480.000 | *Di ogni esercizio viene salvato lo svolgimento. Un cliente medio esegue 2 sessioni a settimana in palestra per 6 mesi (contando eventuali allenamenti saltati) ed esegue 8 esercizi a sessione.* |

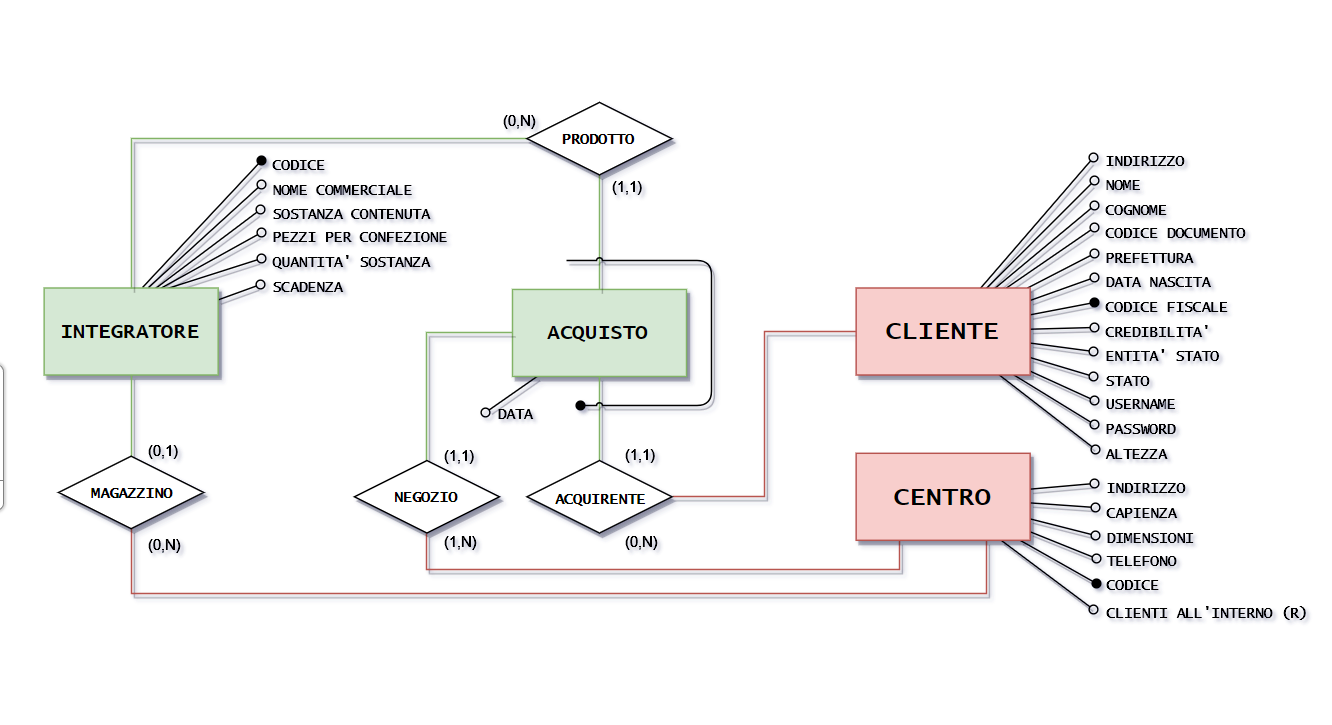


Entità coinvolte nell’operazione 7

Operazione 8 - Acquisto di integratoriAttraverso l’applicazione per smartphone, sarà possibile accedere alla merce presente nel magazzino dei vari centri, potendo così visualizzare eventuali offerte e promozioni. Inoltre, sarà anche possibile acquistarla sempre attraverso l’applicazione, sia pagando direttamente attraverso una carta di credito o Paypal, sia pagando al momento del ritiro presso la reception della palestra.

* Questa operazione riceve in **input** *il prodotto acquistato*, *il cliente* e *il centro* dove è avvenuto l’acquisto.
* L’operazione fornisce in **output** *una nuova istanza per Acquisto*, *Prodotto*, *Negozio* e *Acquirente*, oltre che *la cancellazione dell’istanza* *che collegava l’integratore al centro* dove era in magazzino.

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | E/R | Numero istanze | Motivazione |
| Acquisto | E | 40.000 | *Una persona su quattro in media ha acquistato almeno due integratori differenti.* |
| Acquirente | R | 40.000 | *(1,1) con acquisto.* |
| Prodotto | R | 40.000 | *In media viene acquistato un integratore alla volta.* |
| Negozio | R | 40.000 | *(1,1) con acquisto.* |
| Magazzino | R | 34.000 | *Gli integratori non acquistati sono conservati nel magazzino.* |



Entità coinvolte nell’operazione 8

# Analisi delle prestazioni delle operazioni

## Tavola dei volumi

### Entità

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | E/R | Numero istanze | Motivazione |
| Accesso | E | 3.840.000 | *Un cliente medio accede 2 volte a settimana per 6 mesi.* |
| Acquisto | E | 40.000 | *Una persona su quattro in media ha acquistato almeno due integratori differenti.* |
| Amicizia | E | 800.000 | *In media ogni cliente ha 20 amicizie, ogni amicizia comprende due clienti.* |
| Armadietto | E | 3.600 | *Ogni spogliatoio in media possiede 20 armadietti* |
| Attività | E | 100 | *Ipotesi sul numero di attività di possibile interesse.* |
| Attrezzatura | E | 250 | *In media un esercizio su due richiede l’ausilio di un macchinario o attrezzo particolare.* |
| Calendario centro | E | 240 | *Un centro è aperto in media 6 giorni a settimana.* |
| Calendario corso | E | 1.800 | *Un corso viene eseguito in media 3 volte a settimana.* |
| Centro | E | 40 | *Ipotesi sul numero di centri gestiti da una grossa azienda che opera sul territorio. In Italia sono attive circa 4.000 palestre e si suppone che una grossa azienda ne possieda circa l’1%.* |
| Cerchia | E | 800.000 | *In media ogni amico è inserito in una cerchia.* |
| Cliente | E | 80.000 | *Ogni palestra ha un’utenza media di circa 2.000 clienti, di cui non tutti necessariamente attivi.* |
| Contratto | E | 160.000 | *In media vengono stipulati 2 contratti a cliente.* |
| Corso | E | 1.000 | *Ogni palestra offre in media 25 corsi differenti, come spinning, arti marziali, zumba, yoga, pilates, allenamenti di vario tipo e corsi sportivi.* |
| Dieta personalizzata | E | 60.000 | *In media un cliente che possiede una scheda di alimentazione su due cambia almeno una volta la propria dieta.* |
| Dipendente | E | 800 | *Una palestra media ha 20 dipendenti, tra medici, personal trainer, istruttori, addetti alla reception e magazzinieri.* |
| Esercizio | E | 500 | *In media su ogni gruppo muscolare possono essere eseguiti 5 esercizi differenti.* |
| Esercizio Svolto | E | 276.480.000 | *Di ogni esercizio viene salvato lo svolgimento. Un cliente medio esegue 2 sessioni a settimana in palestra per 6 mesi (contando eventuali allenamenti saltati) ed esegue 8 esercizi a sessione.* |
| Fornitore | E | 30 | *Ipotesi sul numero di fornitori di integratori.* |
| Gruppo Muscolare | E | 10 | *I gruppi muscolari dove poter eseguire un potenziamento sono 10 in totale.* |
| Integratore | E | 74.000 | *Una persona su quattro in media ha acquistato almeno due integratori differenti e se ne ipotizzano 1.000 disponibili per fornitore, più altri 4.000 di riserva nei magazzini dei centri.* |
| Link esterno | E | 64.000 | *In media un post su 25 contiene un link esterno.* |
| Macchinario | E | 2.800 | *Ogni palestra possiede in media 70 attrezzature differenti, tra macchinari di vario genere, attrezzi particolari per esercizi (senza contare pesi, palle ergonomiche e altre attrezzature di minor rilevanza) e attrezzature sportive.* |
| Misurazione | E | 80.000 | *In media un cliente su tre si fa misurare almeno una volta ogni due mesi (un contratto medio dura 6 mesi).* |
| Obiettivo | E | 50 | *Ipotesi sul numero di obiettivi diversi esistenti.* |
| Ordine | E | 3.700 | *Un ordine medio da parte di un centro coinvolge 20 integratori.* |
| Pagamento rateizzato | E | 40.000 | *Ogni contratto necessita di un pagamento* |
| Post | E | 1.600.000 | *In media ogni utente scrive 20 post comprese le risposte.* |
| Rata | E | 120.000 | *Un contratto su quattro è suddiviso in 3 rate.* |
| Sala | E | 320 | *Ogni palestra ha una media di 8 sale, tra sale pesi, piscine, sale dove si eseguono corsi e campi per vari sport.* |
| Scheda alimentazione | E | 40.000 | *In media un cliente su quattro necessita di una scheda di alimentazione* |
| Scheda allenamento | E | 240.000 | *Ogni cliente possiede in media 3 schede di allenamento, in quanto un’iscrizione di media durata è di circa 6 mesi e una scheda si cambia ogni 2.* |
| Sessione | E | 480 | *Si ipotizzano esistenti 20 varianti di combinazioni di esercizi per le 4 zone del corpo solitamente allenate in una sessione (pettorali e tricipiti, dorsali e bicipiti, bambe e spalle, lombari e addominali) da combinare con 6 giorni settimanali.* |
| Sfida | E | 16.000 | *In media ogni utente ha partecipato ad almeno una sfida e ogni sfida in media ha 5 partecipanti.* |
| Spogliatoio | E | 180 | *Ogni centro in media possiede 3 spogliatoi.* |
| Turno | E | 4.000 | *In media un dipendente lavora 5 giorni a settimana.* |
|  |  |  |  |

### Relazioni

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | E/R | Numero istanze | Motivazione |
| Acquirente | R | 40.000 | *(1,1) con acquisto.* |
| Allegato | R | 64.000 | *In media ogni allegato comprende un solo link esterno.* |
| Area | R | 320 | *(1,1) con sala.* |
| Assegnamento | R | 1.280.000 | *A un accesso su 3 viene assegnato un armadietto* |
| Consulente | R | 160.000 | *(1,1) con contratto.* |
| Consulenza | R | 80.000 | *(1,1) con misurazione.* |
| Disponibilità | R | 30.000 | *In media ogni fornitore dispone di 1.000 integratori in magazzino.* |
| Divisione | R | 120.000 | *(1,1) con rata.* |
| Dotazione | R | 2.800 | *(1,1) con macchinario.* |
| Fine | R | 40.000 | *(1,1) con scheda di alimentazione* |
| Fornimento | R | 74.000 | *(1,1) con integratore.* |
| Giudizio | R | 107.000 | *In media un post su tre ha almeno due giudizi* |
| Inserimento | R | 800.000 | *In media esiste una cerchia per amicizia.* |
| Interesse | R | 160.000 | *In media ogni cliente è interessato a due attività.* |
| Iscrizione | R | 20.000 | *In media si iscrivono 20 clienti a corso.* |
| Istruttore | R | 800 | *In media un corso su tre ha più di un istruttore.* |
| Lezione | R | 1.800 | *(1,1) con calendario corso.* |
| Locazione | R | 3.840.000 | *(1,1) con accesso.* |
| Magazzino | R | 34.000 | *Gli integratori non acquistati sono conservati nel magazzino.* |
| Medico | R | 80.000 | *(1,1) con misurazione.* |
| Negozio | R | 40.000 | *(1,1) con acquisto.* |
| Occupazione | R | 2.160 | *Una sessione di un corso su 5 occupa in media due sale, mentre le restanti in media una.* |
| Orario | R | 240 | *(1,1) con calendario centro.* |
| Ordinazione | R | 3.700 | *(1,1) con ordine.* |
| Partecipante | R | 80.000 | *In media ogni cliente ha partecipato ad almeno una sfida.* |
| Pasto | R | 60.000 | *(1,1) con dieta personalizzata.* |
| Personale | R | 4.000 | *(1,1) con turno.* |
| Posizione | R | 3.600 | *(1,1) con armadietto.* |
| Possesso | R | 240.000 | *(1,1) con scheda allenamento.* |
| Possibilità accesso | R | 640.000 | *In media ogni contratto garantisce l’accesso a 4 sale.* |
| Potenziamento muscolare | R | 300 | *In media sono scelti tre livelli diversi per gruppo muscolare.* |
| Prodotto | R | 40.000 | *In media viene acquistato un integratore alla volta.* |
| Prodotto ordinato | R | 40.000 | *Una persona su quattro in media ha acquistato almeno due integratori differenti* |
| Proponente | R | 16.000 | *(1,1) con sfida.* |
| Proposta | R | 5.760.000 | *Ogni sessione in media è composta da 8 esercizi diversi.* |
| Proprietà | R | 40.000 | *(1,1) con scheda di alimentazione.* |
| Pubblicazione | R | 1.600.000 | *(1,1) con post.* |
| Raccolta dati | R | 276.480.000 | *(1,1) con esercizio svolto.* |
| Responsabile | R | 760 | *Ogni dipendente, tranne i direttori, ha un responsabile.* |
| Responsabile sala | R | 320 | *(1,1) con sala.* |
| Ricezione | R | 800.000 | *(1,1) con amicizia.* |
| Richiesta | R | 800.000 | *(1,1) con amicizia.* |
| Risposta | R | 1.280.000 | *In media 8 post su 10 sono di risposta.* |
| Saldo | R | 40.000 | *(1,1) con pagamento rateizzato.* |
| Scelta | R | 160.000 | *(1,1) con contratto.* |
| Scheda alimentazione associata | R | 3.200 | *In media una sfida su 5 ha una dieta associata* |
| Scheda allenamento associata | R | 16.000 | *(1,1) con sfida.* |
| Scrittura | R | 240.000 | *(1,1) con scheda allenamento.* |
| Sede | R | 192.000 | *In media un contratto su cinque è multisede e in tal caso ne menziona due.* |
| Segnalazione | R | 3.840.000 | *(1,1) con accesso.* |
| Servizio | R | 180 | *(1,1) con spogliatoio.* |
| Sottoscrizione | R | 160.000 | *(1,1) con contratto.* |
| Suddivisione | R | 720.000 | *Si ipotizzano 3 sessioni per scheda.* |
| Supporto | R | 250 | *In media un esercizio su due necessita di un’attrezzatura* |
| Svolgimento | R | 276.480.000 | *(1,1) con esercizio svolto.* |
| Thread | R | 640.000 | *In media 4 post su 10 fanno parte del thread di una sfida* |
| Turnazione | R | 800 | *(1,1) con dipendente.* |
| Tutor | R | 80.000 | *In media ogni cliente ha un tutor.* |
| Ubicazione | R | 3.840.000 | *(1,1) con accesso.* |
| Utilizzo | R | 138.240.000 | *In media un esercizio svolto su due ha richiesto l’ausilio di un macchinario o attrezzo particolare.* |
| Visita | R | 80.000 | *(1,1) con misurazione.* |
|  |  |  |  |

## Stima della frequenza giornaliera

|  |  |  |  |
| --- | --- | --- | --- |
| Operazione | Tipo | Frequenza | Motivazione |
| Op. 1 | I | 112.000 al giorno | Dalla tavola dei volumi si ipotizzano 350 persone attive al giorno per 40 palestre per 8 esercizi svolti in media. |
| Op. 2 | B | 670 al giorno | In media una scheda viene cambiata una volta ogni due mesi, inoltre si ipotizzano 40.000 clienti attivi nelle palestre. |
| Op. 3 | I | 12.000 al giorno | Si ipotizzano 350 accessi al giorno per palestra per 6 giorni su 7. |
| Op. 4 | I | 36.000 al giorno | Si ipotizzano 3 caricamenti dell'app della palestra al giorno per ognuno dei 12.000 che hanno fatto l'accesso in palestra. |
| Op. 5 | I | 2.850 al giorno | Si ipotizza che per ogni utente attivo venga eseguita una ricerca ogni due settimane. |
| Op. 6 | B | 220 al giorno | Si ipotizza la stipulazione di un contratto ogni 6 mesi per i 40.000 clienti attivi. |
| Op. 7 | B | 2.800 a settimana | Ogni settimana l'usura dei macchinari (2.800) viene aggiornata. |
| Op. 8 | B | 330 al giorno | Un quarto dei clienti attivi compra un integratore al mese. |

# Introduzione di ridondanze

## Tavola degli accessi

### Operazione 1:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Concetto | Costrutto | Accessi | Tipo | Motivazione |
| Cliente | Entità | 1 | L | Si parte dal cliente che possiede il bracciale elettronico. |
| Raccolta dati | Relazione | 1 | L | Si passa dalla relazione raccolta dati in modo da arrivare agli esercizi svolti dal cliente. |
| Esercizio svolto | Entità | 1 | S | Si scrivono i dati ricavati dal braccialetto elettronico. |
| Svolgimento | Relazione | 1 | S | Si scrive l'istanza che collega esercizio a esercizio svolto. |
| Utilizzo | Relazione | 0,5 | S | Si scrive l'istanza che lega l'esercizio svolto col macchinario reale in cui è stato svolto e la configurazione, ciò avviene in media una volta su due. |

### Operazione 2:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Concetto | Costrutto | Accessi | Tipo | Motivazione |
| Scheda allenamento | Entità | 1 | S | Si scrive la nuova scheda di allenamento. |
| Scrittura | Relazione | 1 | S | Si scrive l'istanza che collega scheda allenamento con dipendente. |
| Possesso | Relazione | 1 | S | Si scrive l'istanza che collega scheda allenamento con cliente. |
| Suddivisione | Relazione | 3 | S | Si scrivono le istanze che collegano le sessioni alla nuova scheda, mediamente sono 3 sessioni a scheda. |

### Operazione 3:

#### Senza ridondanza

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Concetto | Costrutto | Accessi | Tipo | Motivazione |
| Accesso | Entità | 1 | S | Si scrivono i dati dell'accesso presi dal braccialetto elettronico. |
| Segnalazione | Relazione | 1 | S | Si aggiunge un'istanza che collega cliente con accesso. |
| Ubicazione | Relazione | 1 | S | Si crea un'istanza che collega l'accesso al centro. |
| Assegnamento | Relazione | 26 | L | Due clienti su tre hanno settato l'opzione dell'assegnamento automatico di un armadietto al loro braccialetto elettronico, un cliente su tre non ne ha bisogno. Si accede in lettura ad assegnamento, armadietto, servizio e posizione, per vedere quali armadietti sono liberi nella palestra di accesso. Mediamente richiede l'accesso in lettura a 40 istanze delle entità e relazioni sopracitate. Ci sono 40 palestre in totale e mediamente 1 armadietto su 20 appartiene alla palestra giusta, 1 su 2 è mediamente libero. |
| Armadietto | Entità | 26 | L |  |
| Servizio | Relazione | 26 | L |  |
| Posizione | Relazione | 26 | L |  |
| Assegnamento | Entità | 0,66 | S | Mediamente si trova sempre un armadietto libero e si scrive l'istanza che collega l'armadietto all'accesso oltre al codice di sblocco. |

#### Con ridondanza

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Concetto | Costrutto | Accessi | Tipo | Motivazione | Ridondanza |
| Accesso | Entità | 1 | S | Si scrivono i dati dell'accesso presi dal braccialetto elettronico. |  |
| Segnalazione | Relazione | 1 | S | Si aggiunge un'istanza che collega cliente con accesso. |  |
| Ubicazione | Relazione | 1 | S | Si crea un'istanza che collega l'accesso al centro. |  |
| Armadietto | Entità | 13 | L | Due clienti su tre hanno settato l'opzione dell'assegnamento automatico di un armadietto al loro braccialetto elettronico, un cliente su tre non ne ha bisogno. Si accede in lettura ad armadietto per vedere quali armadietti sono liberi. | Occupato. |
| Servizio | Relazione | 13 | L | Si entra in lettura a servizio e posizione per verificare che l’armadietto sia nella giusta palestra. |  |
| Posizione | Relazione | 13 | L |  |  |
| Assegnamento | Entità | 0,66 | S | Mediamente si trova sempre un armadietto libero e si scrive l'istanza che collega l'armadietto all'accesso oltre al codice di sblocco. |  |
| Accesso | Entità | 1 | S | Si scrivono i dati dell'accesso presi dal braccialetto elettronico. |  |
| Segnalazione | Relazione | 1 | S | Si aggiunge un'istanza che collega cliente con accesso. |  |
| Ubicazione | Relazione | 1 | S | Si crea un'istanza che collega l'accesso al centro. |  |
| Armadietto | Entità | 0,66 | L | Due clienti su tre hanno settato l'opzione dell'assegnamento automatico di un armadietto al loro braccialetto elettronico, un cliente su tre non ne ha bisogno. Si accede in lettura ad armadietto per vedere quali armadietti sono liberi (tramite Occupato) nella palestra di accesso (tramite Centro). | Centro |
| Assegnamento | Entità | 0,66 | S | Mediamente si trova sempre un armadietto libero e si scrive l'istanza che collega l'armadietto all'accesso oltre al codice di sblocco. |  |

* L'operazione senza ridondanze richiede (26x4 + (3 + 0,66)x2)x12.000  1.300.000 accessi base giornalieri.
* La ridondanza Occupato occupa 1x3.600 3.600 byte di memoria.
* La ridondanza *Occupato* richiede ±8.000 accessi in scrittura al giorno: infatti la scrittura per l’occupazione viene eseguita nella stessa operazione 3, la scrittura per la liberazione dell’armadietto è in più e richiede circa 2/3 degli accessi giornalieri in operazioni di scrittura, infatti solo 2/3 degli utenti si avvalgono del servizio di assegnamento armadietto.
* L’operazione con la ridondanza *Occupato* richiede (13x3 + (3 + 0,66)x2)x12.000 + 8.000x2 570.000 accessi base al giorno (inoltre non coinvolge più l’associazione assegnamento in lettura per conoscere gli armadietti liberi).
* ***Conclusione:*** *con la ridondanza Occupato si risparmiano circa 730.000 accessi base giornalieri con un dispendio di memoria minimo, si ritiene dunque necessario implementarla.*
* La ridondanza Centro occupa 8x3.600 28.800 byte di memoria (28,8 kilobyte).
* La ridondanza *Centro* non richiede praticamente aggiornamenti.
* L'operazione senza tale ridondanza richiede 570.000 accessi base al giorno (calcolo sopra).
* L’operazione con la ridondanza *Centro* richiede (1x6 + 0,66x3)x12.000 96.000 accessi base al giorno.
* ***Conclusione:*** *con la ridondanza Centro si risparmiano circa 470.000 accessi base al giorno con un dispendio minimo di memoria, si ritiene dunque necessario implementarla.*

### Operazione 4:

#### Senza ridondanze

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Concetto | Costrutto | Accessi | Tipo | Motivazione |
| Cliente | Entità | 1 | L | Si parte dal cliente cha accede all'app social della palestra. |
| Chiesta | Relazione | 20 | L | Si leggono quali sono le amicizie chieste dall'utente (in media 10). Partendo dalle richieste ricevute dall'utente si arriva al richiedente (in media 10). |
| Ricevuta | Relazione | 20 | L | Si leggono quali sono le amicizie ricevute dall’utente (in media 10). Partendo dalle richieste chieste dall'utente si arriva a chi erano state inviate (in media 10). |
| Amicizia | Entità | 20 | L | Si leggono quali sono le richieste con stato "accettata". Un utente ha in media 20 amicizie. |
| Pubblicazione | Relazione | 400 | L | Si entra in lettura in pubblicazione per vedere i post pubblicati dagli amici (in media 20 per amico). |
| Post | Entità | 400 | L | Si entra in lettura in post per vedere quali sono stati pubblicati nelle ultime 24 ore. |

#### Con ridondanze

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Concetto | | Costrutto | Accessi | Tipo | | | Motivazione | Ridondanza |
| Cliente | Entità | | 1 | | L | Si parte dal cliente cha accede all'app social della palestra. | |  |
| Chiesta | Relazione | | 20 | L | | Si leggono quali sono le amicizie chieste dall'utente (in media 10). Partendo dalle richieste ricevute dall'utente si arriva al richiedente (in media 10). | |  |
| Ricevuta | Relazione | | 20 | L | | Si leggono quali sono le amicizie ricevute dall’utente (in media 10). Partendo dalle richieste chieste dall'utente si arriva a chi erano state inviate (in media 10). | |  |
| Amicizia | Entità | | 20 | L | | Si leggono quali sono le richieste con stato "accettata". Un utente ha in media 20 amicizie. | |  |
| Log\_post | Entità | | 2,22 | L | | Si entra in lettura in post recenti per vedere quali sono stati pubblicati nelle ultime 24 ore. In media un utente su 9 scrive un post in tale intervallo di tempo. | | Creazione "log\_post". |

* Senza la ridondanza si ipotizzano (20x3 + 400x2 + 1)x36.000 31.000.000 accessi al giorno.

* La log table log\_post contiene i post scritti nelle ultime 24 ore cioè i dati del post più l'autore.
* *Log\_post* richiede la memorizzazione di 4.400 istanze ipotizzate al giorno e occupa (255+4+8+8)x4.400 1.200.000 byte (1,2 megabyte) circa di memoria.
* La log table necessita di 8.800 accessi in scrittura per essere mantenuta aggiornata (scrittura e cancellazione dei post scadute le 24 ore dalla creazione) più altri 4.400 accessi base all’entità Post per prenderne i dati e altri 4.400 alla relazione pubblicazione per prenderne l’autore.
* Con l’introduzione della log table sono necessari (20x3 + 1 + 2,22)x36.000 + 8.800x2 + 4.400x2 2.300.000 accessi base al giorno.
* ***Conclusione:*** *con la log table si risparmiano circa 28.700.000 accessi base al giorno con un dispendio minimo di memoria, si ritiene dunque necessario implementarla.*

### Operazione 5:

#### Senza ridondanze

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Concetto | Costrutto | Accessi | Tipo | Motivazione |
| Centro | Entità | 1 | L | Si parte dal centro di cui si vogliono conoscere i dati dei corsi. |
| Area | Relazione | 8 | L | Si leggono quali sono le relazioni tra centro e sala (in media 8) attraverso area. |
| Occupazione | Relazione | 75 | L | Si leggono quali sono le relazioni tra sala e calendario corso attraverso occupazione. In media si svolgono 25 corsi a palestra tre volte a settimana. |
| Calendario corso | Entità | 75 | L | Si leggono i dati degli orari delle sale. |
| Lezione | Relazione | 25 | L | Si leggono quali sono le relazioni tra calendario corso e corso attraverso svolgimento. |
| Corso | Entità | 25 | L | Si leggono le informazioni dei corsi. |

#### Con ridondanze

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Concetto | Costrutto | Accessi | Tipo | Motivazione | Ridondanza |
| Corso | Entità | 25 | L | Si leggono le informazioni dei corsi tra quelli che sono del giusto centro (ridondanza). | Centro. |
| Lezione | Relazione | 25 | L | Si leggono quali sono le relazioni tra calendario corso e corso attraverso svolgimento. |  |
| Calendario corso | Entità | 75 | L | Si leggono i dati degli orari delle sale. In media si svolgono 25 corsi a palestra tre volte a settimana. |  |
| Occupazione | Relazione | 75 | L | Si leggono quali sono le relazioni tra sala e calendario corso attraverso occupazione. |  |

* Senza ridondanza sono necessari (25x2 + 75x2 + 8 + 1)x2.850 600.000 accessi giornalieri.
* Si aggiunge la ridondanza Centro, per ogni corso, memorizzante il centro di appartenenza.
* Tale ridondanza necessita di 4x1.000 = 4.000 byte di memoria e non richiede aggiornamenti.
* Con la ridondanza si hanno (25x2 + 75x2)x2850 570.000 accessi giornalieri che coinvolgono però 4 entità e tabelle su 6.
* ***Conclusione:*** *la ridondanza non apporta cambiamenti significativi al numero di accessi ma diminuisce di 1/3 il numero di entità e relazioni coinvolte, perciò si decide comunque di implementarla.*

### Operazione 6:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Concetto | Costrutto | Accessi | Tipo | Motivazione |
| Contratto | Entità | 1 | S | Si scrive il contratto. |
| Consulente | Relazione | 1 | S | Si scrive l'istanza che collega contratto a dipendente. |
| Sottoscrizione | Relazione | 1 | S | Si scrive l'istanza che collega contratto a cliente. |
| Scelta | Relazione | 1 | S | Si scrive l'istanza che collega contratto con obbiettivo (in media uno). |
| Sede | Relazione | 1,33 | S | Si scrive l'istanza che collega contratto a centro, in media un contratto su tre è multi-sede per due centri. |
| Possibilità accesso | Relazione | 4 | S | Si scrive l'istanza che collega contratto a sala, in media ogni contratto prevede 4 sale. |
| Pagamento rateizzato | Entità | 0,25 | S | In media un contratto su 4 è rateizzato. |
| Saldo | Relazione | 0,25 | S | Si scrive l'istanza che collega contratto a pagamento. |
| Rata | Entità | 0,75 | S | Si scrive la rata, in media un contratto su 4 è diviso in 3 rate. |
| Divisione | Relazione | 0,75 | S | Si scrive l'istanza che collega contratto a rata. |

### Operazione 7:

#### Senza ridondanza

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Concetto | Costrutto | Accessi | Tipo | Motivazione |
| Macchinario | Entità | 1 | L | Si parte dall'attrezzo a cui si vuole cambiare la percentuale di usura. |
| Utilizzo | Relazione | 49.000 | L | Si leggono gli utilizzi del macchinario, gli utilizzi salvati sono uno ogni due esercizi svolti divisi per ogni attrezzo reale. |
| Esercizio svolto | Entità | 120 | L | Per ogni esercizio tra quelli che utilizzano l'attrezzo e che hanno inizio nell'ultima settimana, se ne legge la durata (in media 120). |
| Macchinario | Entità | 1 | S | Si cambia la percentuale di usura sommando a quella precedente la percentuale della somma delle durate degli esercizi eseguiti con quell'attrezzo su 2.000 ore di lavoro. |
|  |  |  |  |  |

#### Con ridondanza

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Concetto | | Costrutto | | Accessi | Tipo | Motivazione | Ridondanza |
| Macchinario | Entità | | 1 | | L | Si parte dall'attrezzo a cui si vuole cambiare la percentuale di usura. |  |
| Log\_utilizzo | | Relazione | 120 | | L | Si leggono gli utilizzi del macchinario in media 120 a settimana (calcolo nella tabella precedente). | Si crea una log table "log\_utilizzo" che contiene i dati relativi all'utilizzo dell'ultima settimana per ogni attrezzo. |
| Macchinario | Entità | | 1 | | S | Si cambia la percentuale di usura sommando a quella precedente la percentuale della somma delle durate degli esercizi eseguiti con quell'attrezzo su 2.000 ore di lavoro. |  |

* Senza la log table sono necessari 2.800x(49.000 + 120) = 137.000.000 accessi base a settimana.
* La log table log\_utilizzo contiene la durata degli esercizi svolti con un macchinario e il codice di tale macchinario.
* La log table richiede (4 + 4)x336.000 2.700.000 byte (2,7 megabyte) di memoria.
* *Log\_utilizzo* necessita di 2.800x120 = 336.000 accessi settimanali in scrittura per scrivere i dati più altrettanti per eliminarli, inoltre richiede lo stesso numero di accessi in lettura all’entità *Esercizio Svolto* per ricavare i dati e altrettanti ancora alla relazione utilizzo per accoppiare gli esercizi svolti all’attrezzo reale.
* Con la log table sono necessari 336.000x6 + (120 + 2)x2.800 = 2.360.000 accessi base settimanali.
* ***Conclusione:*** *con la log table si risparmiano circa 135.000.000 accessi base a settimana con un dispendio minimo di memoria, si ritiene dunque necessario implementarla.*

### Operazione 8:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Concetto | Costrutto | Accessi | Tipo | Motivazione |
| Acquisto | Entità | 1 | S | Si scrive una nuova istanza per acquisto. |
| Acquirente | Relazione | 1 | S | Si scrive un'istanza che collega l'acquisto al cliente. |
| Prodotto | Relazione | 1 | S | Si scrive un'istanza che collega l'acquisto al prodotto (in media uno). |
| Negozio | Relazione | 1 | S | Si scrive un'istanza che collega l'acquisto al centro. |
| Magazzino | Relazione | 1 | S | Si cancella l'integratore acquistato dalle istanze nel magazzino. |

# Progettazione logica

**Accesso**(Cliente, Centro, Momento entrata, Uscita)  
**Acquisto**(Cliente, Integratore, Data, Centro)  
**Allegato**(Link esterno, Post)  
**Allenamento**(Esercizio, Sessione)  
**Amicizia**(Richiedente, Ricevente, Stato)  
**Armadietto**(Codice identificativo, Spogliatoio, Occupato, Centro)  
**Assegnamento**(Armadietto, Momento, CodiceSblocco, Cliente)  
**Attività**(Nome, Interessato)  
**Attrezzatura**(Esercizio, Configurazione, Tipologia macchinario)  
**Calendario Centro**(Giorno settimana, Orario apertura, Centro, Orario chiusura)  
**Calendario Corso**(Giorno settimana, Orario inizio, Corso, Durata, Sala, Centro)  
**Centro**(Codice, Numero telefonico, Capienza massima clienti, Dimensione, Indirizzo, Clienti dentro)  
**Cerchia** (Nome cerchia, Richiedente, Ricevente, Direzione inserimento)  
**Cliente**(Codice Fiscale, Codice documento, Prefettura, Cognome, Nome, Username, Password, Indirizzo, Altezza, Data nascita, Tutor attuale, Credibilità, Stato, Entità stato)  
**Contratto**(Codice, Cliente, Consulente, Prezzo, Data sottoscrizione, Accessi settimanali, Durata, Tipo, Saldato)  
**Corso**(Codice, Nome, Livello, Data fine, Data inizio, Centro)  
**Dieta**(Codice, Calorie massime, Calorie minime, Numero pasti, Composizione pasti, Scheda alimentazione)  
**Dipendente**(Codice fiscale, Codice documento, Data nascita, Prefettura, Cognome, Nome, Indirizzo)  
**Esercizio**(Codice, Dispendio energetico, Nome, Numero ripetizioni, Numero per ripetizione, Durata recupero, Durata esercizio)  
**Esercizio Svolto**(Inizio, Esercizio, Cliente, Calorie consumate, Battito cardiaco medio, Tempo impiegato, Configurazione, Attrezzatura, Tempo recupero medio)  
**Fine**(Obiettivo, Scheda alimentazione)  
**Fornitore**(Partita IVA, Indirizzo, Nome commerciale, Firma societaria, Numero telefonico)  
**Giudizio**(Post, Cliente, Valore, Timestamp)  
**Integratore**(Codice, Nome commerciale, Sostanza contenuta, Pezzi per confezione, Quantità prodotto, Scadenza, Fornitore)  
**Iscrizione**(Corso, Cliente)  
**Magazzino**(Integratore, Centro)   
**Macchinario**(Codice, Tipologia, Consumo energetico, Percentuale usura, Sala, Centro)  
**Misurazione**(Data, Medico, Cliente, Peso, Percentuale grassa, Percentuale magra, Percentuale acqua)  
**Obiettivo**(Scopo)  
**Ordine**(Codice interno, Codice esterno, Data effettuazione, Data consegna preferita, Data consegna, Centro)  
**Pagamento Rateizzato**(Contratto, Istituto, Interesse)  
**Partecipante**(Partecipante, Sfida, Avanzamento)  
**Possibilità accesso**(Contratto, Sala, Centro, Accessi settimanali)  
**Post**(Codice, Pubblicatore, Timestamp, Testo)  
**Potenziamento Muscolare**(Obiettivo, Gruppo muscolare, Livello)  
**Prodotto Ordinato**(Integratore, Codice ordine)  
**Rata**(Scadenza, Pagamento, Stato, Importo)  
**Responsabile**(Responsabile, Dipendente, Centro)  
**Risposta**(Post, Risposta)  
**Sala**(Nome, Centro, Responsabile)  
**Scelta**(Obiettivo, Contratto)  
**Scheda Alimentazione**(Codice, Cliente, Nutrizionista, Data inizio, Data fine, Intervallo visite programmato)  
**Scheda Allenamento**(Codice, Cliente, Data inizio, Data fine, Tutor)  
**Sede**(Contratto, Centro)  
**Sessione**(Tipo sessione)  
**Sfida**(Codice, Nome, Data lancio, Proponente, Scheda allenamento associata, Scopo, Scheda alimentazione associata, Data inizio, Scadenza)  
**Spogliatoio**(Identificatore, Posizione, Capienza, Centro)  
**Suddivisione**(Sessione, Scheda allenamento, Giorno settimana)  
**Thread**(Sfida, Post)  
**Turno**(Giorno settimana, Orario inizio, Dipendente, Centro, Orario fine, Attività)

## Vincoli di integrità referenziale

* Esiste un vincolo di integrità referenziale tra l’attributo “Tutor attuale” della tabella “Cliente” e l’attributo “Codice fiscale” della tabella “Dipendente”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Responsabile” della tabella “Responsabile” e l’attributo “Codice fiscale” della tabella “Dipendente”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Dipendente” della tabella “Responsabile” e l’attributo “Codice fiscale” della tabella “Dipendente”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Centro” della tabella “Responsabile” e l’attributo “Codice” della tabella “Centro”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Cliente” della tabella “Scheda Allenamento” e l’attributo “Codice fiscale” della tabella “Cliente”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Tutor” della tabella “Scheda Allenamento” e l’attributo “Codice fiscale” della tabella “Dipendente”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Esercizio” della tabella “Allenamento” e l’attributo “Codice” della tabella “Esercizio”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Sessione” della tabella “Allenamento” e l’attributo “Tipo sessione” della tabella “Sessione”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Centro” della tabella “Sala” e l’attributo “Codice” della tabella “Centro”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Responsabile” della tabella “Sala” e l’attributo “Codice fiscale” della tabella “Dipendente”.
* Esiste un vincolo di integrità referenziale tra gli attributi “Sala” e “Centro” della tabella “Macchinario” e gli attributi “Nome” e “Centro” della tabella “Sala”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Esercizio” della tabella “Attrezzatura” e l’attributo “Codice” della tabella “Esercizio”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Dipendente” della tabella “Turno” e l’attributo “Codice fiscale” della tabella “Dipendente”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Centro” della tabella “Turno” e l’attributo “Codice” della tabella “Centro”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Centro” della tabella “Corso” e l’attributo “Codice” della tabella “Centro”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Corso” della tabella “Iscrizione” e l’attributo “Codice” della tabella “Centro”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Cliente” della tabella “Iscrizione” e l’attributo “Codice fiscale” della tabella “Cliente”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Corso” della tabella “Calendario Corso” e l’attributo “Codice” della tabella “Corso”.
* Esiste un vincolo di integrità referenziale tra gli attributi “Sala” e “Centro” della tabella “Calendario Corso” e gli attributi “Nome” e “Centro” della tabella “Sala”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Centro” della tabella “Calendario Centro” e l’attributo “Codice” della tabella “Centro”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Medico” della tabella “Misurazione” e l’attributo “Codice fiscale” della tabella “Dipendente”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Cliente” della tabella “Misurazione” e l’attributo “Codice fiscale” della tabella “Cliente”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Cliente” della tabella “Scheda Alimentazione” e l’attributo “Codice fiscale” della tabella “Cliente”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Nutrizionista” della tabella “Scheda Alimentazione” e l’attributo “Codice fiscale” della tabella “Dipendente”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Responsabile” della tabella “Responsabile” e l’attributo “Codice fiscale” della tabella “Dipendente”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Scheda alimentazione” della tabella “Dieta” e l’attributo “Codice” della tabella “Scheda Alimentazione”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Obiettivo” della tabella “Fine” e l’attributo “Scopo” della tabella “Obiettivo”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Scheda alimentazione” della tabella “Fine” e l’attributo “Codice” della tabella “Scheda Alimentazione”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Obiettivo” della tabella “Potenziamento muscolare” e l’attributo “Scopo” della tabella “Obiettivo”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Cliente” della tabella “Contratto” e l’attributo “Codice fiscale” della tabella “Cliente”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Consulente” della tabella “Contratto” e l’attributo “Codice fiscale” della tabella “Dipendente”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Contratto” della tabella “Sede” e l’attributo “Codice” della tabella “Contratto”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Centro” della tabella “Sede” e l’attributo “Codice” della tabella “Centro”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Contratto” della tabella “Possibilità Accesso” e l’attributo “Codice” della tabella “Contratto”.
* Esiste un vincolo di integrità referenziale tra gli attributi “Sala” e “Centro” della tabella “Possibilità Accesso” e gli attributi “Nome” e “Centro” della tabella “Sala”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Contratto” della tabella “Pagamento Rateizzato” e l’attributo “Codice” della tabella “Contratto”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Pagamento” della tabella “Rata” e l’attributo “Codice” della tabella “Contratto”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Obiettivo” della tabella “Scelta” e l’attributo “Scopo” della tabella “Obiettivo”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Contratto” della tabella “Scelta” e l’attributo “Codice” della tabella “Contratto”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Cliente” della tabella “Accesso” e l’attributo “Codice fiscale” della tabella “Cliente”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Centro” della tabella “Accesso” e l’attributo “Codice” della tabella “Centro”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Centro” della tabella “Spogliatoio” e l’attributo “Codice” della tabella “Centro”.
* Esiste un vincolo di integrità referenziale tra gli attributi “Spogliatoio” e “Centro” della tabella “Armadietto” e gli attributi “Identificatore” e “Centro” della tabella “Spogliatoio”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Armadietto” della tabella “Assegnamento” e l’attributo “Codice identificativo” della tabella “Armadietto”.
* Esiste un vincolo di integrità referenziale tra gli attributi “Cliente” e “Momento” della tabella “Assegnamento” e gli attributi “Cliente” e “Momento entrata” della tabella “Assegnamento”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Centro” della tabella “Ordine” e l’attributo “Codice” della tabella “Centro”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Fornitore” della tabella “Integratore” e l’attributo “Partita IVA” della tabella “Fornitore”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Integratore” della tabella “Magazzino” e l’attributo “Codice” della tabella “Integratore”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Centro” della tabella “Magazzino” e l’attributo “Codice” della tabella “Centro”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Cliente” della tabella “Acquisto” e l’attributo “Codice fiscale” della tabella “Cliente”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Integratore” della tabella “Acquisto” e l’attributo “Codice” della tabella “Integratore”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Centro” della tabella “Acquisto” e l’attributo “Codice” della tabella “Centro”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Integratore” della tabella “Prodotto ordinato” e l’attributo “Codice” della tabella “Integratore”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Codice ordine” della tabella “Prodotto ordinato” e l’attributo “Codice interno” della tabella “Ordine”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Richiedente” della tabella “Amicizia” e l’attributo “Codice fiscale” della tabella “Cliente”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Ricevente” della tabella “Amicizia” e l’attributo “Codice fiscale” della tabella “Cliente”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Proponente” della tabella “Sfida” e l’attributo “Codice fiscale” della tabella “Cliente”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Scheda allenamento associata” della tabella “Sfida” e l’attributo “Codice” della tabella “Scheda Allenamento”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Scheda alimentazione associata” della tabella “Sfida” e l’attributo “Codice” della tabella “Scheda Alimentazione”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Partecipante” della tabella “Partecipante” e l’attributo “Codice fiscale” della tabella “Cliente”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Sfida” della tabella “Partecipante” e l’attributo “Codice” della tabella “Sfida”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Pubblicatore” della tabella “Post” e l’attributo “Codice fiscale” della tabella “Cliente”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Sfida” della tabella “Thead” e l’attributo “Codice” della tabella “Sfida”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Post” della tabella “Thread” e l’attributo “Codice” della tabella “Post”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Post” della tabella “Risposta” e l’attributo “Codice” della tabella “Post”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Risposta” della tabella “Risposta” e l’attributo “Codice” della tabella “Post”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Post” della tabella “Giudizio” e l’attributo “Codice” della tabella “Post”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Cliente” della tabella “Giudizio” e l’attributo “Codice fiscale” della tabella “Cliente”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Interessato” della tabella “Attività” e l’attributo “Codice fiscale” della tabella “Cliente”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Post” della tabella “Allegato” e l’attributo “Codice” della tabella “Post”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Esercizio” della tabella “Esercizio Svolto” e l’attributo “Codice” della tabella “Esercizio”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Cliente” della tabella “Esercizio Svolto” e l’attributo “Codice fiscale” della tabella “Cliente”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Attrezzatura” della tabella “Esercizio Svolto” e l’attributo “Codice” della tabella “Macchinario”.
* Esiste un vincolo di integrità referenziale tra gli attributi “Richiedente” e “Ricevente” della tabella “Cerchia” e gli attributi “Richiedente” e “Ricevente” della tabella “Amicizia”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Sessione” della tabella “Suddivisione” e l’attributo “Tipo sessione” della tabella “Sessione”.
* Esiste un vincolo di integrità referenziale tra l’attributo “Scheda allenamento” della tabella “Suddivisione” e l’attributo “Codice” della tabella “Scheda Allenamento”.

# Analisi delle dipendenze funzionali e normalizzazione

## dipendenze funzionali

**Tabella Accesso**:  
Accesso(Cliente, Centro, Momento entrata, Uscita)  
Poiché non esistono dipendenze funzionali, Accesso è in BCNF.

**Tabella Acquisto**:  
Acquisto(Cliente, Integratore, Data, Centro)  
- Cliente, Integratore -> Data, Centro  
Poiché la parte sinistra è superchiave, Acquisto è in BCNF.

**Tabella Allegato**:  
Allegato(Link esterno, Post)  
Poiché non esistono dipendenze funzionali, Allegato è in BCNF.

**Tabella Allenamento**:  
Allenamento(Esercizio, Sessione)  
Poiché non esistono dipendenze funzionali, Allenamento è in BCNF.

**Tabella Amicizia**:  
Amicizia(Richiedente, Ricevente, Stato)  
- Richiedente, Ricevente -> Stato  
Poiché la parte sinistra è superchiave, Amicizia è in BCNF.

**Tabella Armadietto**:  
Armadietto(Codice identificativo, Spogliatoio, Occupato, Centro)  
- Codice Identificativo -> Spogliatoio, Occupato, Centro  
Poiché la parte sinistra è superchiave, Armadietto è in BCNF.

**Tabella Assegnamento**:  
Assegnamento(Armadietto, Momento, CodiceSblocco, Cliente)  
- Armadietto, Momento -> CodiceSblocco, Cliente  
Poiché la parte sinistra è superchiave, Assegnamento è in BCNF.

**Tabella Attività**:  
Attività(Nome, Interessato)  
Poiché non esistono dipendenze funzionali, Attività è in BCNF.

**Tabella Attrezzatura**:  
Attrezzatura(Esercizio, Configurazione, Tipologia macchinario)  
- Esercizio, Tipologia macchinario -> Configurazione  
Poiché la parte sinistra è superchiave, Attrezzatura è in BCNF.

**Tabella Calendario Centro**:  
Calendario Centro(Giorno settimana, Orario apertura, Centro, Orario chiusura)  
- Giorno settimana, Orario apertura, Centro -> Orario chiusura  
Un centro può aprire più volte nel corso della giornata, ma ogni apertura avrà sempre, necessariamente, una chiusura.  
Poiché la parte sinistra è superchiave, Calendario Centro è in BCNF.

**Tabella Calendario Corso**:  
Calendario Corso(Giorno settimana, Orario inizio, Corso, Durata, Sala, Centro)  
- Giorno settimana, Orario inizio, Corso, Sala, Centro -> Durata  
Poiché la parte sinistra è superchiave, Calendario Corso è in BCNF.

**Tabella Centro**:  
Centro(Codice, Numero telefonico, Capienza massima clienti, Dimensione, Indirizzo, Clienti dentro)  
- Codice -> Numero telefonico, Capienza massima clienti, Dimensione, Indirizzo, Clienti dentro  
Poiché la parte sinistra è superchiave, Centro è in BCNF.

**Tabella Cerchia**:  
Cerchia (Nome cerchia, Richiedente, Ricevente, Direzione inserimento)  
Poiché non ci sono dipendenze funzionali, Cerchia è in BCNF.

**Tabella Cliente**:  
Cliente(Codice Fiscale, Codice documento, Prefettura, Cognome, Nome, Username, Password, Indirizzo, Altezza, Data nascita, Tutor attuale, Credibilità, Stato, Entità stato)  
- Codice fiscale -> Codice documento, Prefettura, Cognome, Nome, Username, Password, Indirizzo, Altezza, Data nascita, Tutor attuale, Credibilità, Stato, Entità stato  
Poiché la parte sinistra è superchiave, Cliente è in BCNF.

**Tabella Contratto**:  
Contratto(Codice, Cliente, Consulente, Prezzo, Data sottoscrizione, Accessi settimanali, Durata, Tipo, Saldato)  
- Codice -> Cliente, Consulente, Prezzo, Data sottoscrizione, Accessi settimanali, Durata, Tipo  
Poiché la parte sinistra è superchiave, Contratto è in BCNF.

**Tabella Corso**:  
Corso(Codice, Nome, Livello, Data fine, Data inizio, Centro)  
- Codice -> Nome, Livello, Data fine, Data inizio, Centro  
La ridondanza Centro è dipendente da Codice, il quale è la chiave di Corso. Quindi non influisce sulla normalizzazione (ma è un vincolo di integrità referenziale da aggiungere).  
Poiché la parte sinistra è superchiave, Corso è in BCNF.

**Tabella Dieta**:  
Dieta(Codice, Calorie massime, Calorie minime, Numero pasti, Composizione pasti, Scheda alimentazione)  
- Codice -> Calorie massime, Calorie minime, Numero pasti, Composizione pasti, Tabella Scheda di alimentazione  
Poiché la parte sinistra è superchiave, Dieta è in BCNF.

**Tabella Dipendente**:  
Dipendente(Codice fiscale, Codice documento, Data nascita, Prefettura, Cognome, Nome, Indirizzo)  
- Codice fiscale -> Codice documento, Data nascita, Prefettura, Cognome, Nome, Indirizzo  
Poiché la parte sinistra è superchiave, Dipendente è in BCNF.

**Tabella Esercizio**:  
Esercizio(Codice, Dispendio energetico, Nome, Numero ripetizioni, Numero per ripetizione, Durata recupero, Durata esercizio)  
- Codice -> Dispendio energetico, Nome, Numero ripetizioni, Numero per ripetizione, Durata recupero, Durata esercizio  
Poiché la parte sinistra è superchiave, Esercizio è in BCNF.

**Tabella Esercizio Svolto**:  
Esercizio Svolto(Inizio, Esercizio, Cliente, Calorie consumate, Battito cardiaco medio, Tempo impiegato, Configurazione, Attrezzatura, Tempo recupero medio)  
- Inizio, Esercizio, Cliente -> Calorie consumate, Battito cardiaco medio, Tempo impiegato, Configurazione, Attrezzatura, Tempo recupero medio  
Poiché la parte sinistra è superchiave, Esercizio Svolto è in BCNF.

**Tabella Fine**:  
Fine(Obiettivo, Scheda alimentazione)  
Poiché non ci sono dipendenze funzionali, Fine è in BCNF.

**Tabella Fornitore**:  
Fornitore(Partita IVA, Indirizzo, Nome commerciale, Firma societaria, Numero telefonico)  
- Partita IVA -> Indirizzo, Nome commerciale, Firma societaria, Numero telefonico  
Poiché la parte sinistra è superchiave, Fornitore è in BCNF.

**Tabella Giudizio**:  
Giudizio(Post, Cliente, Valore, Timestamp)  
- Post, Cliente -> Valore, Timestamp  
Poiché la parte sinistra è superchiave, Giudizio è in BCNF.

**Tabella Integratore**:  
Integratore(Codice, Nome commerciale, Sostanza contenuta, Pezzi per confezione, Quantità prodotto, Scadenza, Fornitore)  
- Codice -> Nome commerciale, Sostanza contenuta, Pezzi per confezione, Quantità prodotto, Scadenza, Fornitore  
Poiché la parte sinistra è superchiave, Integratore è in BCNF.

**Tabella Iscrizione**:  
Iscrizione(Corso, Cliente)  
Poiché non ci sono dipendenze funzionali, Iscrizione è in BCNF.

**Tabella Magazzino**:  
Magazzino(Integratore, Centro)  
Poiché non ci sono dipendenze funzionali, Magazzino è in BCNF.

**Tabella Macchinario**:  
Macchinario(Codice, Tipologia, Consumo energetico, Percentuale usura, Sala, Centro)  
- Codice -> Tipologia, Consumo energetico, Percentuale usura, Sala, Centro  
Poiché la parte sinistra è superchiave, Macchinario è in BCNF.

**Tabella Misurazione**:  
Misurazione(Data, Medico, Cliente, Peso, Percentuale grassa, Percentuale magra, Percentuale acqua)  
- Data, Medico, Cliente -> Peso, Percentuale grassa, Percentuale magra, Percentuale acqua  
Poiché la parte sinistra è superchiave, Misurazione è in BCNF.

**Tabella Obiettivo**:  
Obiettivo(Scopo)  
Poiché non ci sono dipendenze funzionali, Obiettivo è in BCNF.

**Tabella Ordine**:  
Ordine(Codice interno, Codice esterno, Data effettuazione, Data consegna preferita, Data consegna, Centro)  
- Codice interno -> Codice esterno, Data effettuazione, Data consegna preferita, Data consegna, Centro  
Poiché la parte sinistra è superchiave, Ordine è in BCNF.

**Tabella Pagamento Rateizzato**:  
Pagamento Rate(Contratto, Istituto, Interesse)  
- Contratto -> Istituto, Interesse  
Poiché la parte sinistra è superchiave, Pagamento Rate è in BCNF.

**Tabella Partecipante**:  
Partecipante(Partecipante, Sfida, Avanzamento)  
- Partecipante, Sfida -> Avanzamento  
Poiché la parte sinistra è superchiave, Partecipante è in BCNF.

**Tabella Possibilità Accesso**:  
Possibilità accesso(Contratto, Sala, Centro, Accessi settimanali)  
- Contratto, Sala, Centro -> Accessi settimanali  
Poiché la parte sinistra è superchiave, Possibilità Accesso è in BCNF.

**Tabella Post**:  
Post(Codice, Pubblicatore, Timestamp, Testo)  
- Codice -> Pubblicatore, Timestamp, Testo  
Poiché la parte sinistra è superchiave, Post è in BCNF.

**Tabella Potenziamento Muscolare**:  
Potenziamento Muscolare(Obiettivo, Gruppo muscolare, Livello)  
Poiché non ci sono dipendenze funzionali, Potenziamento Muscolare è in BCNF.

**Tabella Prodotto Ordinato**:  
Prodotto Ordinato(Integratore, Codice ordine)  
Poiché non ci sono dipendenze funzionali, Prodotto Ordinato è in BCNF.

**Tabella Rata**:  
Rata(Scadenza, Pagamento, Stato, Importo)  
- Scadenza, Pagamento -> Stato, Importo  
Poiché la parte sinistra è superchiave, Rata è in BCNF.

**Tabella Responsabile**:  
Responsabile(Responsabile, Dipendente, Centro)  
Poiché non ci sono dipendenze funzionali, Responsabile è in BCNF.

**Tabella Risposta**:  
Risposta(Post, Risposta)  
Poiché non ci sono dipendenze funzionali, Risposta è in BCNF.

**Tabella Sala**:  
Sala(Nome, Centro, Responsabile)  
- Nome, Centro -> Responsabile  
Poiché la parte sinistra è superchiave, Sala è in BCNF.

**Tabella Scelta**:  
Scelta(Obiettivo, Contratto)  
Poiché non ci sono dipendenze funzionali, Scelta è in BCNF.

**Tabella Scheda Alimentazione**:  
Scheda Alimentazione(Codice, Cliente, Nutrizionista, Data inizio, Data fine, Intervallo visite programmato)  
- Codice -> Cliente, Nutrizionista, Data inizio, Data fine, Intervallo visite programmato  
Poiché la parte sinistra è superchiave, Scheda Alimentazione è in BCNF.

**Tabella Scheda Allenamento**:  
Scheda Allenamento(Codice, Cliente, Data inizio, Data fine, Tutor)  
- Codice -> Cliente, Data inizio, Data fine, Tutor  
Poiché la parte sinistra è superchiave, Scheda allenamento è in BCNF.

**Tabella Sede**:  
Sede(Contratto, Centro)  
Poiché non ci sono dipendenze funzionali, Sede è in BCNF.

**Tabella Sessione**:  
Sessione(Tipo sessione)  
Poiché non ci sono dipendenze funzionali, Sessione è in BCNF.

**Tabella Sfida**:  
Sfida(Codice, Nome, Data lancio, Proponente, Scheda allenamento associata, Scopo, Scheda alimentazione associata, Data inizio, Scadenza)  
- Codice -> Nome, Data lancio, Proponente, Scheda allenamento associata, Scopo, Scheda alimentazione associata, Data inizio, Scadenza  
Poiché la parte sinistra è superchiave, Sfida è in BCNF.

**Tabella Spogliatoio**:  
Spogliatoio(Identificatore, Posizione, Capienza, Centro)  
- Identificatore -> Posizione, Capienza, Centro  
Poiché la parte sinistra è superchiave, Spogliatoio è in BCNF.

**Tabella Suddivisione**:  
Suddivisione(Sessione, Scheda allenamento, Giorno settimana)  
Poiché non ci sono dipendenze funzionali, Suddivisione è in BCNF.

**Tabella Thread**:  
Thread(Sfida, Post)  
Poiché non ci sono dipendenze funzionali, Thread è in BCNF.

**Tabella Turno**:  
Turno(Giorno settimana, Orario inizio, Dipendente, Centro, Orario fine, Attività)  
- Giorno settimana, Orario inzio, Dipendente, Centro -> Orario fine, Attività  
Poiché la parte sinistra è superchiave, Turno è in BCNF.

## Normalizzazione

Possiamo notare come tutte le tabelle siano già nella Forma Normale di Boyce-Codd (BCNF). Questo implica che il database è già normalizzato e non necessita di altri passaggi.

# Implementazione su DBMS Oracle MySQL

## Operazioni

### Operazione 1

1. -- -----------------------------------------------------
2. -- Inserimento esercizio svolto
3. -- -----------------------------------------------------
4. **DROP** **PROCEDURE** IF EXISTS `Progetto`.`inserimento\_esercizio\_svolto`;
5. DELIMITER $$
6. USE `Progetto` $$
7. **CREATE** **PROCEDURE** `inserimento\_esercizio\_svolto` (IN inizio **TIMESTAMP**,
8. IN esercizio **INT**,
9. IN cliente **VARCHAR**(16),
10. IN calorie **INT**,
11. IN battito **INT**,
12. IN tempo **TIME**(6),
13. IN recupero **TIME**(6),
14. IN configurazione **VARCHAR**(45),
15. IN attrezzatura **INT**)
16. **BEGIN**
18. **DECLARE** EXIT HANDLER **FOR** SQLEXCEPTION
19. **BEGIN**
20. **ROLLBACK**;
21. **SELECT** "Si e' verificato un errore!";
22. **END**;
24. **INSERT** **INTO** `Esercizio\_Svolto`
25. **VALUES** (inizio, esercizio, cliente, calorie, battito, tempo, recupero, configurazione, attrezzatura);
26. **END** $$
27. DELIMITER ;

La **stored procedure** sopra riportata *permette l’inserimento dei dati relativi allo svolgimento di un esercizio*, sarà chiamata dal dispositivo indossato dall’utente durante le sessioni di allenamento dopo aver raccolto i dati necessari.  
Il codice semplicemente prende **in input** *i dati forniti dal dispositivo* e **inserisce** nel database *una row che li contiene*. Saranno conseguentemente attivati gli eventuali vincoli di integrità per controllare che i valori inseriti siano validi.

### Operazione 2

1. -- -----------------------------------------------------
2. -- Inserimento scheda allenamento
3. -- -----------------------------------------------------
4. **DROP** **PROCEDURE** IF EXISTS `Progetto`.`inserimento\_scheda\_allenamento`;
5. DELIMITER $$
6. USE `Progetto` $$
7. **CREATE** **PROCEDURE** `inserimento\_scheda\_allenamento` (IN codice **INT**,
8. IN cliente **VARCHAR**(45),
9. IN data\_fine **DATE**,
10. IN tutor **VARCHAR**(45))
11. **BEGIN**
13. **DECLARE** EXIT HANDLER **FOR** SQLEXCEPTION
14. **BEGIN**
15. **ROLLBACK**;
16. **SELECT** "Si e' verificato un errore!";
17. **END**;
19. **INSERT** **INTO** Scheda\_allenamento
20. **VALUES** (codice, cliente, **CURRENT\_DATE**, data\_fine, tutor);
21. **END** $$
22. DELIMITER ;

Questa **stored procedure** implementa *l’inserimento di una nuova scheda di allenamento*. **I parametri passat**i alla procedura sono gli stessi *attributi della tabella* *Scheda\_allenamento*, tranne che per la data di inizio, dove viene inserita quella del giorno corrente. Il controllo sulle date (la data di inizio deve essere minore di quella di fine) saranno implementate attraverso un trigger.

1. -- -----------------------------------------------------
2. -- Inserimento suddivisione
3. -- -----------------------------------------------------
4. **DROP** **PROCEDURE** IF EXISTS `Progetto`.`inserimento\_suddivisione`;
5. DELIMITER $$
6. USE `Progetto` $$
7. **CREATE** **PROCEDURE** `inserimento\_suddivisione` (IN sessione **VARCHAR**(45),
8. IN scheda\_allenamento **INT**,
9. IN giorno\_settimana **VARCHAR**(45))
10. **BEGIN**
12. **DECLARE** EXIT HANDLER **FOR** SQLEXCEPTION
13. **BEGIN**
14. **ROLLBACK**;
15. **SELECT** "Si e' verificato un errore!";
16. **END**;
18. **INSERT** **INTO** Suddivisione
19. **VALUES** (sessione, scheda\_allenamento, giorno\_settimana);
20. **END** $$
21. DELIMITER ;

Questa **stored procedure** implementa l’*inserimento di una nuova istanza per* *Suddivisione,* contenente la suddivisionedi una determinata scheda di allenamento. I parametri **in ingresso** sono *la sessione* da attribuire alla scheda di allenamento*, il codice della scheda di allenamento* e *il giorno della settimana* in cui essa deve essere svolta.

### Operazione 3

1. -- -----------------------------------------------------
2. -- Inserimento accesso
3. -- -----------------------------------------------------
4. **DROP** **PROCEDURE** IF EXISTS `Progetto`.`inserimento\_accesso`;
5. DELIMITER $$
6. USE `Progetto` $$
7. **CREATE** **PROCEDURE** `inserimento\_accesso` (IN cliente\_ **VARCHAR**(16),
8. IN centro\_ **INT**,
9. **OUT** armadietto\_output **INT**,
10. **OUT** sblocco\_output **INT**)
11. **BEGIN**
12. **DECLARE** istante\_ **TIMESTAMP** **DEFAULT** CURRENT\_TIMESTAMP;
13. **DECLARE** armadietto\_ **INT** **DEFAULT** 0;
14. **DECLARE** sblocco\_ **INT** **DEFAULT** 0;
16. **DECLARE** EXIT HANDLER **FOR** SQLEXCEPTION
17. **BEGIN**
18. **ROLLBACK**;
19. **SELECT** "Si e' verificato un errore!";
20. **END**;
22. **INSERT** **INTO** Accesso
23. **VALUES** (istante\_, cliente\_, centro\_, 0);
25. **SELECT** D.CodiceIdentificativo **INTO** armadietto\_
26. **FROM** (**SELECT** @row\_number:=@row\_number+1 **AS** RowNumber, A.CodiceIdentificativo
27. **FROM** (**SELECT** @row\_number:=0) **AS** R, Armadietto A
28. **WHERE** A.Occupato=0 AND A.Centro=centro\_) **AS** D
29. **WHERE** D.RowNumber=1;
31. IF @row\_number <> 0 **THEN**
32. **SET** sblocco\_ = floor(rand()\*1000000);
33. **INSERT** **INTO** Assegnamento
34. **VALUES** (armadietto\_, istante\_, sblocco\_, cliente\_);
36. **UPDATE** Armadietto
37. **SET** Occupato=1
38. **WHERE** CodiceIdentificativo=armadietto\_;
40. **UPDATE** Centro
41. **SET** ClientiDentro = ClientiDentro + 1
42. **WHERE** Codice=centro\_;
44. **ELSE**
45. **SELECT** 'Non ci sono armadietti disponibili';
46. **END** IF;
48. **SET** armadietto\_output := armadietto\_;
49. **SET** sblocco\_output := sblocco\_;
51. **END** $$
52. DELIMITER ;

Questa **stored procedure** permette, nel momento in cui l’utente accede ad uno dei centri, di *attribuirgli uno degli armadietti disponibili*.  
Vengono presi **in ingresso** *il codice fiscale* dell’utente e *il centro* nel quale vuole fare l’accesso, mentre vengono presi come **variabili d’uscita** due interi, che rappresentano *l’armadietto assegnato* e il relativo *codice di sblocco*.  
Prima di tutto viene inserita una row contente i dati di accesso, avente valore 0 nell’attributo riguardante l’uscita (che sarà aggiornata ad 1 nel momento in cui l’utente uscirà dal centro).  
Con la porzione di query successiva si ottengono gli armadietti disponibili relativi a quel centro. Le row number sono state usate per poter selezionare solo uno degli armadietti disponibili e per poter mettere il suo codice identificativo all’interno della variabile ***armadietto*** (la condizione “WHERE RowNumber=1” permette di scegliere il primo dei risultati, che, nel caso ci siano armadietti disponibili, esiste sempre).  
A questo punto, la variabile user-defined ***row\_number*** contiene il numero di armadietti disponibili. Pertanto, se è diversa da 0, vuol dire che è presente almeno un armadietto da poter assegnare all’utente che sta facendo l’accesso. Attraverso il contenuto dell’*IF* viene generato un numero casuale che rappresenterà il codice di sblocco. La funzione ***rand()*** genera un double compreso tra 0 e 1, moltiplicandolo per 1000000 e poi usando la funzione ***floor()***, che lo trasforma in un intero eliminando la parte decimale, si ottiene un numero a sei cifre e si inserisce all’interno della tabella *Assegnamento*. L’attributo ridondante ***ClientiDentro*** della tabella *Centro* viene incrementato per poter eseguire un controllo su un successivo accesso in quel centro, che non deve essere permesso nel caso in cui sia superata la capienza massima (il codice SQL dei vincoli generici si trova nella rispettiva sezione).  
Infine, sempre nel caso in cui ci sia almeno un armadietto disponibile, andiamo ad aggiornare l’armadietto scelto settando la colonna ***Occupato*** a 1. Nel caso in cui, invece, non ci sia nemmeno un armadietto disponibile, verrà selezionata la stringa ‘Non ci sono armadietti disponibili’.  
Le variabili di output prendono i valori dell’armadietto e del codice di sblocco, nel caso in cui non ci siano armadietti disponili saranno settate a 0 (il loro valore di default). Nel momento in cui l’utente esce dal centro, dovrà essere aggiornata la colonna ***Occupato*** dell’armadietto riportandola a 0.

### Operazione 4

1. -- -----------------------------------------------------
2. -- Log table post
3. -- -----------------------------------------------------
4. **CREATE** **TABLE** IF NOT EXISTS `Progetto`.`log\_post` (
5. `Codice` **INT** NOT NULL,
6. `Pubblicatore` **VARCHAR**(16) NOT NULL,
7. `**Timestamp**` **TIMESTAMP** NULL,
8. `Testo` **VARCHAR**(500) NULL,
9. **PRIMARY** **KEY** (`Codice`))
10. ENGINE = InnoDB;
11. -- -----------------------------------------------------
12. -- Log table allegato
13. -- -----------------------------------------------------
14. **CREATE** **TABLE** IF NOT EXISTS `Progetto`.`log\_allegato` (
15. `LinkEsterno` **VARCHAR**(500) NOT NULL,
16. `Post` **INT** NOT NULL,
17. `**Timestamp**` **TIMESTAMP** NULL,
18. **PRIMARY** **KEY** (`LinkEsterno`, `Post`))
19. ENGINE = InnoDB;
20. -- -----------------------------------------------------
21. -- Log table risposta
22. -- -----------------------------------------------------
23. **CREATE** **TABLE** IF NOT EXISTS `Progetto`.`log\_risposta` (
24. `Post` **INT** NOT NULL,
25. `Risposta` **INT** NOT NULL,
26. `**Timestamp**` **TIMESTAMP** NULL,
27. **PRIMARY** **KEY** (`Post`, `Risposta`))
28. ENGINE = InnoDB;

L’implementazione della stored procedure, chiamata dall’applicazione per la *visualizzazione dei post degli amici* di un determinato utente, avviene dopo aver analizzato le tavole dei volumi e degli accessi attraverso delle **log table**: una per memorizzare i post (***log\_post***), una per memorizzare gli allegati (***log\_allegato***) e una per memorizzare le risposte (***log\_risposta***).

1. -- -----------------------------------------------------
2. -- Trigger inserimento post delle ultime 24 ore
3. -- -----------------------------------------------------
4. **DROP** **TRIGGER** IF EXISTS `Progetto`.`trigger\_post\_24\_ore`;
5. DELIMITER $$
6. USE `Progetto`$$
7. **CREATE** **TRIGGER** `trigger\_post\_24\_ore`
8. **AFTER** **INSERT** **ON** Post
9. **FOR** EACH ROW
10. **BEGIN**
12. **DECLARE** EXIT HANDLER **FOR** SQLEXCEPTION
13. **BEGIN**
14. **INSERT** **INTO** Errore
15. **VALUES** (CURRENT\_TIMESTAMP, "Errore durante il trigger trigger\_post\_24\_ore");
16. **END**;
18. **INSERT** **INTO** log\_post
19. **VALUES** (NEW.Codice, NEW.Pubblicatore, NEW.**Timestamp**, NEW.Testo);
20. **END** $$
21. DELIMITER ;
22. -- -----------------------------------------------------
23. -- Trigger inserimento allegati delle ultime 24 ore
24. -- -----------------------------------------------------
25. **DROP** **TRIGGER** IF EXISTS `Progetto`.`trigger\_allegato\_24\_ore`;
26. DELIMITER $$
27. USE `Progetto`$$
28. **CREATE** **TRIGGER** `trigger\_allegato\_24\_ore`
29. **AFTER** **INSERT** **ON** Allegato
30. **FOR** EACH ROW
31. **BEGIN**
33. **DECLARE** EXIT HANDLER **FOR** SQLEXCEPTION
34. **BEGIN**
35. **INSERT** **INTO** Errore
36. **VALUES** (CURRENT\_TIMESTAMP, "Errore durante il trigger  trigger\_allegato\_24\_ore");
37. **END**;
39. **INSERT** **INTO** log\_allegato
40. **VALUES** (NEW.LinkEsterno, NEW.Post, CURRENT\_TIMESTAMP);
41. **END** $$
42. DELIMITER ;
43. -- -----------------------------------------------------
44. -- Trigger inserimento risposte delle ultime 24 ore
45. -- -----------------------------------------------------
46. **DROP** **TRIGGER** IF EXISTS `Progetto`.`trigger\_risposta\_24\_ore`;
47. DELIMITER $$
48. USE `Progetto`$$
49. **CREATE** **TRIGGER** `trigger\_risposta\_24\_ore`
50. **AFTER** **INSERT** **ON** Risposta
51. **FOR** EACH ROW
52. **BEGIN**
54. **DECLARE** EXIT HANDLER **FOR** SQLEXCEPTION
55. **BEGIN**
56. **INSERT** **INTO** Errore
57. **VALUES** (CURRENT\_TIMESTAMP, "Errore durante il trigger  trigger\_risposta\_24\_ore");
58. **END**;
60. **INSERT** **INTO** log\_risposta
61. **VALUES** (NEW.Post, NEW.Risposta, CURRENT\_TIMESTAMP);
62. **END** $$
63. DELIMITER ;
64. -- -----------------------------------------------------
65. -- Aggiornamento log tables post, allegato e risposta
66. -- -----------------------------------------------------
67. **DROP** EVENT IF EXISTS `Progetto`.`aggiornamento\_log\_post`;
68. DELIMITER $$
69. USE `Progetto`$$
70. **CREATE** EVENT `aggiornamento\_log\_post`
71. **ON** SCHEDULE EVERY 1 DAY
72. STARTS '2017-09-19 23:59:00'
73. DO
74. **BEGIN**
76. **DECLARE** EXIT HANDLER **FOR** SQLEXCEPTION
77. **BEGIN**
78. **ROLLBACK**;
79. **INSERT** **INTO** Errore
80. **VALUES** (CURRENT\_TIMESTAMP, "Errore durante l'evento aggiornamento\_usura");
81. **END**;
83. **DELETE** **FROM** log\_post
84. **WHERE** DAY(**Timestamp**)<>DAY(**CURRENT\_DATE**);
86. **DELETE** **FROM** log\_allegato
87. **WHERE** DAY(**Timestamp**)<>DAY(**CURRENT\_DATE**);
89. **DELETE** **FROM** log\_risposta
90. **WHERE** DAY(**Timestamp**)<>DAY(**CURRENT\_DATE**);
91. **END** $$
92. DELIMITER ;

Per la gestione delle log table, sono stati implementati:

1. Un **trigger** ***trigger\_post\_24\_or*e** che, ogni qualvolta viene pubblicato un post (ossia inserita una row nella tabella *Post*), *inserisce i suoi dati anche nella tabella* ***log\_post***;
2. Un **trigger** ***trigger\_allegato\_24\_ore*** che, ogni qualvolta viene aggiunto un allegato ad un post (ossia inserita una row nella tabella *Allegato*), *inserisce i suoi dati anche nella tabella* ***log\_allegato***;
3. Un **trigger** ***trigger\_risposta\_24\_ore*** che, ogni qualvolta un post viene pubblicato come risposta (ossia inserita una row nella tabella *Risposta*), *inserisce i suoi dati anche nella tabella* ***log\_risposta***;
4. Un **event *aggiornamento\_log\_post*** che, ogni sera alle 23:59:00, *elimina dalla log table i post, gli allegati e le risposte pubblicati nei giorni precedenti*.
5. -- -----------------------------------------------------
6. -- Ricerca post amici ultime 24 ore
7. -- -----------------------------------------------------
8. **DROP** **PROCEDURE** IF EXISTS `Progetto`.`post\_amici`;
9. DELIMITER $$
10. USE `Progetto`$$
11. **CREATE** **PROCEDURE** `post\_amici` (IN cliente **VARCHAR**(16))
12. **BEGIN**
14. **DECLARE** EXIT HANDLER **FOR** SQLEXCEPTION
15. **BEGIN**
16. **ROLLBACK**;
17. **SELECT** "Si e' verificato un errore!";
18. **END**;
20. **SELECT** LP.\*
21. **FROM** log\_post LP
22. **INNER** JOIN
23. Amicizia A
24. **ON** (LP.Pubblicatore=A.Richiedente AND A.Ricevente=cliente)
25. OR (LP.Pubblicatore=A.Ricevente AND A.Richiedente=cliente)
26. **WHERE** LP.Pubblicatore<>cliente AND
27. LP.**Timestamp** > (CURRENT\_TIMESTAMP - INTERVAL 24 **HOUR**)
28. AND A.Stato='Accettata'
29. **ORDER** **BY** LP.**Timestamp** **DESC**;
30. **END** $$
31. DELIMITER ;
32. -- -----------------------------------------------------
33. -- Ricerca allegati di un post delle ultime 24 ore
34. -- -----------------------------------------------------
35. **DROP** **PROCEDURE** IF EXISTS `Progetto`.`allegati\_post\_24\_ore`;
36. DELIMITER $$
37. USE `Progetto`$$
38. **CREATE** **PROCEDURE** `allegati\_post\_24\_ore` (IN post **INT**)
39. **BEGIN**
41. **DECLARE** EXIT HANDLER **FOR** SQLEXCEPTION
42. **BEGIN**
43. **ROLLBACK**;
44. **SELECT** "Si e' verificato un errore!";
45. **END**;
47. **SELECT** LA.LinkEsterno
48. **FROM** log\_allegato LA
49. **WHERE** LA.Post=post;
50. **END** $$
51. DELIMITER ;
52. -- -----------------------------------------------------
53. -- Ricerca risposte di un post delle ultime 24 ore
54. -- -----------------------------------------------------
55. **DROP** **PROCEDURE** IF EXISTS `Progetto`.`ottieni\_risposte\_24\_ore`;
56. DELIMITER $$
57. USE `Progetto`$$
58. **CREATE** **PROCEDURE** `ottieni\_risposte\_24\_ore` (IN post **INT**)
59. **BEGIN**
61. **DECLARE** EXIT HANDLER **FOR** SQLEXCEPTION
62. **BEGIN**
63. **ROLLBACK**;
64. **SELECT** "Si e' verificato un errore!";
65. **END**;
67. **SELECT** LP.\*
68. **FROM** log\_risposta LR
69. **INNER** JOIN
70. log\_post LP
71. **ON** LR.Risposta=LP.Codice
72. **WHERE** LR.Post=post;
73. **END** $$
74. DELIMITER ;

Sono state, poi, create tre **stored procedure** per la gestione dei dati raccolti nelle log table:

* In ***post\_amici*** vengono *selezionati dalla log table solo i post pubblicati dagli amici*: per farlo, si impone che il post sia pubblicato o dal richiedente o dal ricevente dell’amicizia (non possiamo sapere a priori se il cliente che sta cercando i post sia nelle amicizie di colui che ha fatto la richiesta o di colui che l’ha ricevuta). Successivamente si controlla che il pubblicatore non sia l’utente stesso. Il controllo sulla pubblicazione nelle ultime 24 ore (‘LP.Timestamp > (CURRENT\_TIMESTAMP – INTERVAL 24 HOUR)’) è sicuramente molto più veloce perché i post possono, al massimo, essere quelli pubblicati dopo le 23:59:00 di due giorni prima. Senza l’utilizzo delle log table, invece, il controllo avrebbe dovuto includere tutti i post pubblicati da quando esiste il database. Infine è stato scelto un ordinamento decrescente per visualizzare prima i post più recenti.
* In ***allegati\_post\_24\_ore***, procedura che sarà chiamata dall’applicazione nel momento in cui l’utente seleziona un determinato post pubblicato dagli amici nelle ultime 24 ore, vengono *visualizzati gli allegati relativi ad un determinato post*, prendendoli dalla log table log\_allegato.
* In ***ottieni\_risposte\_24\_ore***, anch’essa procedura che sarà chiamata dall’applicazione nel momento in cui l’utente seleziona un determinato post pubblicato dagli amici nelle ultime 24 ore, vengono *visualizzate le risposte relative a quel post*. Sicuramente le risposte sono state pubblicate nelle ultime 24 ore e quindi la loro ricerca all’interno della log table *log\_risposta* è molto più veloce rispetto alla ricerca in tutte le risposte della tabella Risposta.

1. -- -----------------------------------------------------
2. -- Ricerca post di cui conosciamo una risposta
3. -- -----------------------------------------------------
4. **DROP** **PROCEDURE** IF EXISTS `Progetto`.`ottieni\_post\_originario`;
5. DELIMITER $$
6. USE `Progetto`$$
7. **CREATE** **PROCEDURE** `ottieni\_post\_originario` (IN post **INT**)
8. **BEGIN**
10. **DECLARE** EXIT HANDLER **FOR** SQLEXCEPTION
11. **BEGIN**
12. **ROLLBACK**;
13. **SELECT** "Si e' verificato un errore!";
14. **END**;
16. **SELECT** P.\*
17. **FROM** log\_risposta LR
18. **INNER** JOIN
19. Post P
20. **ON** LR.Post=P.Codice
21. **WHERE** LR.Risposta=post;
22. **END** $$
23. DELIMITER ;

Inoltre è stata aggiunta una **stored procedure** ***ottieni\_post\_originario*** che, poichè i post degli amici pubblicati nelle ultime 24 ore e visualizzati dall’utente potrebbero essere delle risposte ad altri post, *permette di risalire a quello che era il post originario di un eventuale post di risposta* (dato come valore in ingresso). Questa ricerca dovrà essere fatta in tutti i post nel database non potendo essere a conoscenza della data di pubblicazione del post originario.

### Operazione 5

1. -- -----------------------------------------------------
2. -- Ricerca corsi
3. -- -----------------------------------------------------
4. **DROP** **PROCEDURE** IF EXISTS `Progetto`.`ricerca\_corsi`;
5. DELIMITER $$
6. **CREATE** **PROCEDURE** `Progetto`.`ricerca\_corsi` (IN centro **INT**)
7. **BEGIN**
9. **DECLARE** EXIT HANDLER **FOR** SQLEXCEPTION
10. **BEGIN**
11. **ROLLBACK**;
12. **SELECT** "Si e' verificato un errore!";
13. **END**;
15. **SELECT** C.Codice, C.Nome, C.Livello, C.DataInizio, C.DataFine,  CC.GiornoSettimana, CC.OrarioInizio, CC.Durata, CC.Sala
16. **FROM** Corso C
17. **INNER** JOIN
18. Calendario\_corso CC
19. **ON** C.Codice=CC.Corso
20. **WHERE** C.Centro=centro
21. **ORDER** **BY** C.Codice;
22. **END** $$
23. DELIMITER ;

Questa **stored procedure**, chiamata dall’applicazione quando un cliente vuole visualizzare i corsi disponibili in un determinato centro, prende **in input** *il codice di un centro* e **restituisce** *l’elenco dei corsi disponibili in quel centro* *con le relative informazioni*, quali nome del corso, livello, data di inizio, data di fine, giorni della settimana in cui viene svolto (saranno visualizzati su più row) con orario di inizio, durata e sala.

### Operazione 6

1. -- -----------------------------------------------------
2. -- Log table rata
3. -- -----------------------------------------------------
4. **CREATE** **TABLE** IF NOT EXISTS `Progetto`.`log\_rata` (
5. `Scadenza` **DATE** NOT NULL,
6. `Pagamento` **INT** NOT NULL,
7. `Importo` **VARCHAR**(45) NOT NULL,
8. **PRIMARY** **KEY** (`Scadenza`, `Pagamento`))
9. ENGINE = InnoDB;

La tabella ***log\_rata*** ha lo *scopo di memorizzare le rate inserite nelle ultime 24 ore*. Questa funzione non serve ai fini dell’Operazione 6, ma è stata implementata per la business rules che controlla che la somma degli importi delle rate relative ad un contratto sia pari all’importo totale del contratto stesso. L’evento che avrà lo scopo di svuotare questa log table sarà implementato nella sezione relativa a tale business rule.  
La creazione della log table è stata inserita in questa sezione perché verrà utilizzata nell’implementazione della stored procedure che inserisce le varie rate (in questo modo si evita di dover creare un trigger inutile).

1. -- -----------------------------------------------------
2. -- Inserimento contratto
3. -- -----------------------------------------------------
4. **DROP** **PROCEDURE** IF EXISTS `Progetto`.`inserimento\_contratto`;
5. DELIMITER $$
6. USE `Progetto`$$
7. **CREATE** **PROCEDURE** `inserimento\_contratto` (IN codice\_ **INT**,
8. IN cliente\_ **VARCHAR**(16),
9. IN consulente\_ **VARCHAR**(16),
10. IN prezzo\_ **INT**,
11. IN sottoscrizione\_ **DATE**,
12. IN accessi\_ **VARCHAR**(45),
13. IN durata\_ **VARCHAR**(45),
14. IN tipo\_ **VARCHAR**(45),
15. IN saldato\_ BOOL,
16. IN centro1\_ **INT**,
17. IN centro2\_ **INT**,
18. IN centro3\_ **INT**,
19. IN obiettivo\_ **VARCHAR**(45),
20. IN rate\_ BOOL,
21. IN istituto\_ **VARCHAR**(45),
22. IN interesse\_ **INT**,
23. IN username\_ **VARCHAR**(45),
24. IN password\_ **VARCHAR**(45))
25. **BEGIN**
26. **DECLARE** numero\_contratti **INT** **DEFAULT** 0;
27. **DECLARE** EXIT HANDLER **FOR** SQLEXCEPTION
28. **BEGIN**
29. **ROLLBACK**;
30. **SELECT** "Si e' verificato un errore!";
31. **END**;
33. **SELECT** COUNT(\*) **INTO** numero\_contratti
34. **FROM** Contratto Co
35. **WHERE** Co.Cliente=cliente\_;
37. IF numero\_contratti=0 **THEN**
38. **INSERT** **INTO** Contratto
39. **VALUES** (codice\_, cliente\_, consulente\_, prezzo\_,  sottoscrizione\_, accessi\_, durata\_, tipo\_, saldato\_);
41. **INSERT** **INTO** Sede
42. **VALUES** (codice\_, centro1\_);
44. IF centro2\_ **IS** NOT NULL **THEN**
45. **INSERT** **INTO** Sede
46. **VALUES** (codice\_, centro2\_);
47. IF centro3\_ **IS** NOT NULL **THEN**
48. **INSERT** **INTO** Sede
49. **VALUES** (codice\_, centro3\_);
50. **END** IF;
51. **END** IF;
53. **INSERT** **INTO** Obiettivo
54. **VALUES** (obiettivo\_);
56. **INSERT** **INTO** Scelta
57. **VALUES** (obiettivo\_, codice\_);
59. **UPDATE** Cliente
60. **SET** Username=username\_, **Password**=password\_
61. **WHERE** CodiceFiscale=cliente\_;
63. IF rate\_=1 AND saldato\_=0 **THEN**
64. **INSERT** **INTO** Pagamento\_rateizzato
65. **VALUES** (codice\_, istituto\_, interesse\_);
66. **END** IF;
67. **END** IF;
68. **END** $$
69. DELIMITER ;

Questa **stored procedure** implementa *l’inserimento di un contratto*, con i relativi cliente, username e password scelti al momento della sottoscrizione, centri ai quali sarà possibile accedere, consulente, prezzo, data della sottoscrizione, obiettivo, tipo di contratto, importo totale, se è stato pagato o se deve essere rateizzato e l’eventuale istituto e interesse nel caso si sia scelta la formula a rate.  
Viene controllato che il numero di contratti di quel cliente sia zero poiché non ci possono essere due contratti o più dello stesso cliente. Se questa condizione è verificata, allora si procede con **l’inserimento** nella tabella *Contratto*, *Sede* (siccome si possono inserire un massimo di tre centri, questo inserimento è stato implementato con degli IF annidati), *Obiettivo* e *Scelta*. Successivamente, vengono **aggiornati** gli attributi *Username* e *Password* del cliente. Infine, se il contratto deve essere rateizzato e non è stato saldato al momento, allora viene **inserita** *una nuova row nella tabella Pagamento\_rateizzato*.

1. -- -----------------------------------------------------
2. -- Inserimento potenziamento muscolare
3. -- -----------------------------------------------------
4. **DROP** **PROCEDURE** IF EXISTS `Progetto`.`inserimento\_potenziamento`;
5. DELIMITER $$
6. USE `Progetto`$$
7. **CREATE** **PROCEDURE** `inserimento\_potenziamento` (IN obiettivo **VARCHAR**(45),
8. IN gruppo **VARCHAR**(45),
9. IN livello **VARCHAR**(45))
10. **BEGIN**
12. **DECLARE** EXIT HANDLER **FOR** SQLEXCEPTION
13. **BEGIN**
14. **ROLLBACK**;
15. **SELECT** "Si e' verificato un errore!";
16. **END**;
18. IF SUBSTRING(obiettivo **FROM** 1 **FOR** 13)='Potenziamento' **THEN**
19. **INSERT** **INTO** Potenziamento\_muscolare
20. **VALUES** (obiettivo, gruppo, livello);
21. **END** IF;
23. **END** $$
24. DELIMITER ;

Questa **stored procedure** implementa l*’inserimento della scelta da parte di un utente di potenziare un determinato gruppo muscolare*, scegliendone inoltre il livello. Ogni utente, al momento della sottoscrizione del contratto, stipula un obiettivo; nel caso in cui questo obiettivo riguardi il potenziamento muscolare, sarà inserito in memoria come “PotenziamentoN”, dove N è un numero identificativo. Pertanto, affinché si possa procedere con l’inserimento, è necessario accertarsi che la stringa dell’obiettivo cominci con “Potenziamento”. Nel caso in cui questo sia vero, l’inserimento nella tabella *Potenziamento\_muscolare* sarà completato.

1. -- -----------------------------------------------------
2. -- Inserimento rata
3. -- -----------------------------------------------------
4. **DROP** **PROCEDURE** IF EXISTS `Progetto`.`inserimento\_rata`;
5. DELIMITER $$
6. USE `Progetto`$$
7. **CREATE** **PROCEDURE** `inserimento\_rata` (IN contratto **INT**,
8. IN scadenza **DATE**,
9. IN importo **VARCHAR** (45))
10. **BEGIN**
12. **DECLARE** saldato TINYINT **DEFAULT** 0;
13. **DECLARE** controllo\_pagamento\_rateizzato **INT** **DEFAULT** 0;
14. **DECLARE** costo **INT** **DEFAULT** 0;
15. **DECLARE** EXIT HANDLER **FOR** SQLEXCEPTION
16. **BEGIN**
17. **ROLLBACK**;
18. **SELECT** "Si e' verificato un errore!";
19. **END**;
21. **SELECT** COUNT(\*) **INTO** controllo\_pagamento\_rateizzato
22. **FROM** Pagamento\_rateizzato PT
23. **WHERE** PT.Contratto=contratto;
25. **SELECT** C.Saldato, C.Prezzo **INTO** saldato, costo
26. **FROM** Contratto C
27. **WHERE** C.Codice=contratto;
29. IF controllo\_pagamento\_rateizzato>0 AND saldato=0 AND importo<costo
30. AND scadenza>**CURRENT\_DATE** **THEN**
31. **INSERT** **INTO** Rata
32. **VALUES** (scadenza, contratto, 'non saldato', importo);
34. **INSERT** **INTO** log\_rata
35. **VALUES** (scadenza, contratto, importo);
36. **END** IF;
38. **END** $$
39. DELIMITER ;

Questa **stored procedure** implementa *l’inserimento di una rata relativa ad un determinato contratto*. Prima di procedere con l’inserimento nella tabella **Rata**, vengono eseguiti alcuni controlli: il contratto deve prevedere un pagamento rateizzato (variabile *controllo\_pagamento\_rateizzato*, che conterrà 0 se non è previsto oppure 1 se è previsto), il pagamento del contratto non deve essere già stato saldato, l’importo del contratto deve essere minore dell’importo della singola rata che stiamo inserendo e la scadenza deve essere maggiore della data in cui si inserisce il contratto. Manca un controllo secondo il quale la somma degli importi delle rate deve essere uguale all’importo totale del contratto, ma sarà implementato attraverso un evento nella sezione business rules: da questo deriva l’inserimento nella tabella **log\_rata** degli stessi dati della rata che stiamo inserendo.

1. -- -----------------------------------------------------
2. -- Inserimento accesso sala
3. -- -----------------------------------------------------
4. **DROP** **PROCEDURE** IF EXISTS `Progetto`.`inserimento\_accesso\_sala`;
5. DELIMITER $$
6. USE `Progetto`$$
7. **CREATE** **PROCEDURE** `inserimento\_accesso\_sala` (IN contratto **INT**,
8. IN centro **INT**,
9. IN sala **VARCHAR**(45),
10. IN accessi\_settimanali **INT**)
11. **BEGIN**
13. **DECLARE** controllo\_sede\_contratto **INT** **DEFAULT** 0;
14. **DECLARE** EXIT HANDLER **FOR** SQLEXCEPTION
15. **BEGIN**
16. **ROLLBACK**;
17. **SELECT** "Si e' verificato un errore!";
18. **END**;
20. **SELECT** COUNT(\*) **INTO** controllo\_sede\_contratto
21. **FROM** Sede S
22. **WHERE** S.Contratto=contratto AND S.Centro=centro;
24. IF controllo\_sede\_contratto>0 **THEN**
25. **INSERT** **INTO** Possibilita\_accesso
26. **VALUES** (contratto, sala, centro, accessi\_settimanali);
27. **END** IF;
29. **END** $$
30. DELIMITER ;

Questa **stored procedure** *implementa l’inserimento di un determinato numero di accessi settimanali* ad una specifica sala per un contratto. L’unico controllo necessario riguarda l’esistenza del centro, da passare come argomento alla stored procedure nella tabella *Sede,* in una row insieme a quel determinato contratto. Se il centro è presente, allora l’inserimento nella tabella *Possibilita\_accesso* viene eseguito.

### Operazione 7

1. -- -----------------------------------------------------
2. -- Tabella errore
3. -- -----------------------------------------------------
4. **CREATE** **TABLE** IF NOT EXISTS `Progetto`.`Errore` (
5. `TimestampErrore` **TIMESTAMP** NOT NULL,
6. `TestoErrore` **VARCHAR**(300) NOT NULL,
7. **PRIMARY** **KEY**(`TimestampErrore`, `TestoErrore`))
8. ENGINE = InnoDB;

La tabella ***Errore*** viene *creata con il fine di poter memorizzare eventuali errori* in event o trigger, i quali non possono mostrare qualcosa come risultato (soluzione che è stata adottata nelle procedure attraverso un handler di uscita che, qualora si presenti una sqlexception, stampa “Si è verificato un errore!”). Grazie al timestamp e al testo dell’errore, sarà possibile risalire alla causa di esso e correggerla.

1. -- -----------------------------------------------------
2. -- Log table utilizzo
3. -- -----------------------------------------------------
4. **CREATE** **TABLE** IF NOT EXISTS `Progetto`.`log\_utilizzo` (
5. `Inizio` **TIMESTAMP** NOT NULL,
6. `Esercizio` **INT** NOT NULL,
7. `Cliente` **VARCHAR**(16) NOT NULL,
8. `TempoImpiegato` **TIME**(6) NULL,
9. `Attrezzatura` **INT** NULL,
10. **PRIMARY** **KEY** (`Inizio`, `Esercizio`, `Cliente`))
11. ENGINE = InnoDB;

La log table ***log\_utilizzo*** è una ridondanza *utilizzata per velocizzare il processo di aggiornamento dell’usura* dei macchinari. Gli attributi di tale tabella sono gli stessi della tabella *Esercizio\_svolto*, la quale fornisce i dati necessari relativi al tempo di utilizzo dei macchinari.

1. -- -----------------------------------------------------
2. -- Inserimento utlizzo
3. -- -----------------------------------------------------
4. **DROP** **TRIGGER** IF EXISTS `Progetto`.`inserimento\_utilizzo`;
5. DELIMITER $$
6. USE `Progetto`$$
7. **CREATE** **TRIGGER** `inserimento\_utilizzo`
8. **AFTER** **INSERT** **ON** Esercizio\_svolto
9. **FOR** EACH ROW
10. **BEGIN**
12. **DECLARE** EXIT HANDLER **FOR** SQLEXCEPTION
13. **BEGIN**
14. **INSERT** **INTO** Errore
15. **VALUES** (CURRENT\_TIMESTAMP, "Errore durante il trigger inserimento\_utilizzo");
16. **END**;
18. **INSERT** **INTO** log\_utilizzo
19. **VALUES** (NEW.Inizio, NEW.Esercizio, NEW.Cliente, NEW.TempoImpiegato,  NEW.Atrezzatura);
21. **END** $$
22. DELIMITER ;

Questo **trigger** implementa *l’inserimento dei dati all’interno della log table* ***log\_utilizzo***. Esso scatta nel momento in cui viene eseguito un inserimento all’interno della tabella *Esercizio\_svolto* e inserisce esattamente gli stessi dati della nuova row.

1. -- -----------------------------------------------------
2. -- Aggiornamento usura
3. -- -----------------------------------------------------
4. **DROP** EVENT IF EXISTS `Progetto`.`aggiornamento\_usura`;
5. DELIMITER $$
6. USE `Progetto`$$
7. **CREATE** EVENT `aggiornamento\_usura`
8. **ON** SCHEDULE EVERY 7 DAY
9. STARTS '2017-09-19 23:59:00'
10. DO
11. **BEGIN**
13. **DECLARE** finito **INT** **DEFAULT** 0;
14. **DECLARE** macchinario **INT** **DEFAULT** 0;
15. **DECLARE** tempo **DOUBLE** **DEFAULT** 0;
16. **DECLARE** percentuale\_usura **DOUBLE** **DEFAULT** 0;
17. **DECLARE** coefficiente **DOUBLE** **DEFAULT** 0.000014; -- 100:(2000\*60\*60)
19. **DECLARE** cursore **CURSOR** **FOR**
20. **SELECT** LU.Attrezzatura, SUM(TIME\_TO\_SEC(LU.TempoImpiegato)) **AS**  TempoTotale, LU.PercentualeUsura
21. **FROM** log\_utilizzo LU
22. **GROUP** **BY** LU.Attrezzatura;
24. **DECLARE** **CONTINUE** HANDLER **FOR** NOT FOUND
25. **SET** finito=1;
27. **DECLARE** EXIT HANDLER **FOR** SQLEXCEPTION
28. **BEGIN**
29. **ROLLBACK**;
30. **INSERT** **INTO** Errore
31. **VALUES** (CURRENT\_TIMESTAMP, "Errore durante l'evento aggiornamento\_usura");
32. **END**;
34. **OPEN** cursore;
35. preleva: LOOP
36. **FETCH** cursore **INTO** macchinario, tempo, percentuale\_usura;
37. IF finito=1 **THEN**
38. LEAVE preleva;
39. **END** IF;
41. **UPDATE** Macchinario
42. **SET** PercentualeUsura=PercentualeUsura+tempo\*coefficiente
43. **WHERE** Codice=macchinario;
45. IF (percentuale\_usura+tempo\*coefficiente)>=100 **THEN**
46. **UPDATE** Macchinario
47. **SET** Stato = "in manutenzione"
48. **WHERE** Codice=macchinario;
49. **END** IF;
51. **END** LOOP preleva;
52. **CLOSE** cursore;
54. **TRUNCATE** **TABLE** log\_utilizzo;
56. **END** $$
57. DELIMITER ;

Questo **event** ha lo scopo di *gestire i dati raccolti all’interno della tabella* ***log\_utilizzo*** e di usarli per *aggiornare la percentuale di usura dei macchinari* nella tabella *Macchinario*. L’aggiornamento viene eseguito ogni settimana di notte, in modo tale da non appesantire il server.  
Il tempo di utilizzo dei macchinari viene convertito in secondi, attraverso la funzione TIME\_TO\_SEC(), in modo tale da poter eseguire delle operazioni numeriche che sul tipo TIME non sono possibili. Il *coefficiente* 0,000014 deriva dal seguente ragionamento: nella sezione “*Analisi delle specifiche*” è stato deciso che 2000 ore di utilizzo corrispondono al 100%; trasformate in secondi, 2000 ore sono 2000\*60\*60, ossia 7.200.000; la proporzione tempo\_utilizzo: 100 = percentuale: 7.200.000 ci porta a tale coefficiente. All’interno del lavoro compiuto dal cursor, l’attributo *PercentualeUsura* viene aggiornato con il valore PercentualeUsura + tempo\_utilizzo\*coefficiente. Alla fine dell’event, la tabella *log\_utilizzo* viene svuotata.

### Operazione 8

1. -- -----------------------------------------------------
2. -- Acquisto prodotto
3. -- -----------------------------------------------------
4. **DROP** **PROCEDURE** IF EXISTS `Progetto`.`acquisto\_prodotto`;
5. DELIMITER $$
6. USE `Progetto`$$
7. **CREATE** **PROCEDURE** acquisto\_prodotto (IN cliente\_ **VARCHAR**(16),
8. IN integratore\_ **INT**,
9. IN centro\_ **INT**)
10. **BEGIN**
11. **DECLARE** presenza **INT** **DEFAULT** 0;
13. **DECLARE** EXIT HANDLER **FOR** SQLEXCEPTION
14. **BEGIN**
15. **ROLLBACK**;
16. **SELECT** "Si e' verificato un errore!";
17. **END**;
19. **SELECT** COUNT(\*) **INTO** presenza
20. **FROM** Magazzino M
21. **WHERE** M.Integratore=integratore\_ AND M.Centro=centro\_;
23. IF presenza > 0 **THEN**
24. **INSERT** **INTO** Acquisto
25. **VALUES** (cliente\_, integratore\_, **CURRENT\_DATE**, centro\_);
27. **DELETE** **FROM** Magazzino
28. **WHERE** Integratore=integratore\_ AND Centro=centro\_;
29. **ELSE**
30. **SELECT** "L'integratore selezionato non e' presente nel magazzino";
31. **END** IF;
33. **END** $$
34. DELIMITER ;

Questa **stored procedure** implementa *l’acquisto di un singolo integratore presente in un determinato centro*. La variabile ***presenza*** serve per controllare se l’integratore passato come parametro in ingresso sia veramente presente nel centro (anch’esso passato come parametro in ingresso). Se il controllo fallisce, allora viene stampata la stringa “L’integratore selezionato non è presente nel magazzino”. Nel caso in cui sia presente, viene creata una nuova istanza per la tabella *Acquisto* avente come dati il codice fiscale del cliente, il codice dell’integratore, la data corrente e il codice del centro passati come argomenti alla procedura. Successivamente viene eliminato quell’integratore della tabella **Magazzino**. Gli *underscore* nelle variabili ***integratore*\_** e ***centro\_*** sono stati inseriti onde evitare ambiguità.

## Vincoli generici

### Turnazione del personale

1. -- -----------------------------------------------------
2. -- Vincolo turnazione del personale
3. -- -----------------------------------------------------
4. **DROP** **TRIGGER** IF EXISTS `Progetto`.`vincoli\_turnazione`;
5. DELIMITER $$
6. USE `Progetto`$$
7. **CREATE** **TRIGGER** `vincoli\_turnazione`
8. BEFORE **INSERT** **ON** Turno
9. **FOR** EACH ROW
10. **BEGIN**
11. **DECLARE** ore\_giornaliere **INT** **DEFAULT** 0;
12. **DECLARE** orario\_inizio **INT** **DEFAULT** 0;
13. **DECLARE** orario\_fine **INT** **DEFAULT** 0;
14. **DECLARE** centro\_ **INT** **DEFAULT** NEW.Centro;
15. **DECLARE** finito **INT** **DEFAULT** 0;
16. **DECLARE** cursore **CURSOR** **FOR**
17. **SELECT** TIME\_TO\_SEC(T0.OrarioInizio), TIME\_TO\_SEC(T0.OrarioFine), T0.Centro
18. **FROM** Turno T0
19. **WHERE** T0.Dipendente=NEW.Dipendente AND T0.GiornoSettimana=NEW.GiornoSettimana;
20. **DECLARE** **CONTINUE** HANDLER **FOR** NOT FOUND
21. **SET** finito = 1;
23. IF TIME\_TO\_SEC(NEW.OrarioInizio)>=TIME\_TO\_SEC(NEW.OrarioFine) **THEN**
24. SIGNAL SQLSTATE '45000'
25. **SET** MESSAGE\_TEXT = "L'orario di inizio e' maggiore o uguale a quello di fine";
26. **END** IF;
28. **SELECT** SUM(TIME\_TO\_SEC(T1.OrarioFine)-TIME\_TO\_SEC(T1.OrarioInizio))
29. **INTO** ore\_giornaliere
30. **FROM** Turno T1
31. **WHERE** T1.Dipendente=NEW.Dipendente AND T1.GiornoSettimana=NEW.GiornoSettimana;
33. IF (TIME\_TO\_SEC(NEW.OrarioFine)-TIME\_TO\_SEC(NEW.OrarioInizio))+ore\_giornaliere
34. >(8\*60\*60)
35. **THEN**
36. SIGNAL SQLSTATE '45000'
37. **SET** MESSAGE\_TEXT = "Un dipendente non puo' superare le 8 ore  di lavoro giornaliere";
38. **END** IF;
40. **OPEN** cursore;
42. preleva: LOOP
43. IF finito = 1 **THEN**
44. LEAVE preleva;
45. **END** IF;
47. **FETCH** cursore **INTO** orario\_inizio, orario\_fine, centro\_;
49. IF (TIME\_TO\_SEC(NEW.OrarioInizio) BETWEEN orario\_inizio AND orario\_fine)
50. OR (TIME\_TO\_SEC(NEW.OrarioFine) BETWEEN orario\_inizio AND orario\_fine)
51. **THEN**
52. SIGNAL SQLSTATE '45000'
53. **SET** MESSAGE\_TEXT = "Due fasce orarie dello stesso dipendente  non si possono sovrapporre";
54. **END** IF;
56. IF centro\_ **IS** NOT NULL AND centro\_<>NEW.Centro **THEN**
57. SIGNAL SQLSTATE '45000'
58. **SET** MESSAGE\_TEXT = "Un dipendente non puo' lavorare in  centri diversi nello stesso giorno";
59. **END** IF;
60. **END** LOOP preleva;
61. **CLOSE** cursore;
63. **END** $$
64. DELIMITER ;

Questo vincolo generico implementato attraverso un **trigger** esegue alcuni controlli riguardanti la turnazione del personale:

1. *L’orario di inizio del turno non deve essere maggiore dell’orario di fine del turno* (considerando che un turno non possa includere due giorni consecutivi, ma che si debba concludere almeno entro la mezzanotte).
2. *Il dipendente non deve superare le 8 ore lavorative* durante l’arco della giornata.
3. *Non ci devono essere fasce orarie sovrapposte*: l’orario di inizio e di fine turno di un dipendente non deve sovrapporsi ad un’altra fascia oraria in cui e di turno lo stesso.
4. *Un dipendente non può lavorare in centri diversi nello stesso giorno*, ossia non devono esistere fasce orarie con centro diverso da quello che stiamo inserendo per lo stesso giorno.

È stato ritenuto particolarmente conveniente il passaggio da **TIME** a **INT** attraverso la funzione **TIME\_TO\_SEC()**, in modo tale da non avere problemi durante le operazioni algebriche e di confronto.

### Scheda di allenamento

1. -- -----------------------------------------------------
2. -- Vincolo scheda allenamento
3. -- -----------------------------------------------------
4. **DROP** **TRIGGER** IF EXISTS `Progetto`.`vincoli\_scheda\_allenamento`;
5. DELIMITER $$
6. USE `Progetto`$$
7. **CREATE** **TRIGGER** `vincoli\_scheda\_allenamento`
8. BEFORE **INSERT** **ON** Scheda\_allenamento
9. **FOR** EACH ROW
10. **BEGIN**
12. IF NEW.`Data inizio`>=NEW.`Data fine` **THEN**
13. SIGNAL SQLSTATE '45000'
14. **SET** MESSAGE\_TEXT = "La data della fine della scheda di allenamento  non puo' essere minore di quella di inizio";
15. **END** IF;
17. **END** $$
18. DELIMITER ;

Questo vincolo generico implementato mediante **trigger** impone che *la data di inizio di una scheda di allenamento sia minore della data di fine della stessa.*

### Corso

1. -- -----------------------------------------------------
2. -- Vincolo corso
3. -- -----------------------------------------------------
4. **DROP** **TRIGGER** IF EXISTS `Progetto`.`vincoli\_corso`;
5. DELIMITER $$
6. USE `Progetto`$$
7. **CREATE** **TRIGGER** `vincoli\_corso`
8. BEFORE **INSERT** **ON** Corso
9. **FOR** EACH ROW
10. **BEGIN**
12. IF NEW.DataInizio>=NEW.DataFine **THEN**
13. SIGNAL SQLSTATE '45000'
14. **SET** MESSAGE\_TEXT = "La data della fine di un corso non puo' essere  minore di quella di inizio";
15. **END** IF;
17. **END** $$
18. DELIMITER ;

Questo vincolo generico implementato mediante **trigger** impone che *la data di inizio di un corso sia minore della data di fine.*

### Accesso

1. -- -----------------------------------------------------
2. -- Vincoli inserimento accesso
3. -- -----------------------------------------------------
5. **DROP** **TRIGGER** IF EXISTS `Progetto`.`vincoli\_accesso`;
6. DELIMITER $$
7. USE `Progetto`$$
8. **CREATE** **TRIGGER** `vincoli\_accesso`
9. BEFORE **INSERT** **ON** Accesso
10. **FOR** EACH ROW
11. **BEGIN**
12. **DECLARE** acceduto **INT** **DEFAULT** 0;
13. **DECLARE** accessi\_attuali **INT** **DEFAULT** 0;
14. **DECLARE** max\_capienza **INT** **DEFAULT** 0;
15. **DECLARE** max\_accessi **INT** **DEFAULT** 0;
16. **DECLARE** numero\_accessi **INT** **DEFAULT** 0;
17. **DECLARE** codice\_contratto **INT** **DEFAULT** 0;
18. **DECLARE** presenza\_sede **INT** **DEFAULT** 0;
20. **SELECT** COUNT(\*) **INTO** acceduto
21. **FROM** Accesso A0
22. **WHERE** A0.Cliente=NEW.Cliente AND A0.Uscita=0;
24. IF acceduto<>0 **THEN**
25. SIGNAL SQLSTATE '45000'
26. **SET** MESSAGE\_TEXT = "Il cliente ha gia' eseguito l'accesso e non e'  ancora uscito";
27. **END** IF;
29. **SELECT** C0.CapienzaMassimaClienti **INTO** max\_capienza
30. **FROM** Centro C0
31. **WHERE** C0.Codice=NEW.Centro;
33. **SELECT** C1.ClientiDentro **INTO** accessi\_attuali
34. **FROM** Centro C1
35. **WHERE** C1.Codice=NEW.Centro;
37. IF (accessi\_attuali+1)>max\_capienza **THEN**
38. SIGNAL SQLSTATE '45000'
39. **SET** MESSAGE\_TEXT = "Il centro ha gia' raggiunto la capienza massima";
40. **END** IF;
42. **SELECT** C2.AccessiSettimanali, C2.Codice **INTO** max\_accessi, codice\_contratto
43. **FROM** Contratto C2
44. **WHERE** C2.Cliente=NEW.Cliente;
46. **SELECT** COUNT(\*) **INTO** numero\_accessi
47. **FROM** Accesso A2
48. **WHERE** A2.Cliente=NEW.Cliente AND  WEEKOFYEAR(A2.MomentoEntrata)=WEEKOFYEAR(NEW.MomentoEntrata);
50. IF (numero\_accessi+1)>max\_accessi **THEN**
51. SIGNAL SQLSTATE '45000'
52. **SET** MESSAGE\_TEXT = "Il cliente ha gia' eseguito il numero di  accessi settimanali previsti";
53. **END** IF;
55. **SELECT** COUNT(\*) **INTO** presenza\_sede
56. **FROM** Sede S
57. **WHERE** S.Contratto=codice\_contratto AND S.Centro=NEW.Centro;
59. IF presenza\_sede=0 **THEN**
60. SIGNAL SQLSTATE '45000'
61. **SET** MESSAGE\_TEXT = "Il centro a cui il cliente sta accedendo  non e' previsto nel suo contratto";
62. **END** IF;
64. **END** $$
65. DELIMITER ;

Questo vincolo generico implementato mediante **trigger** *controlla che l’inserimento di un accesso sia valido*, ossia quando sono rispettate le seguenti condizioni:

* *Il cliente che sta eseguendo l’accesso non deve avere altri accessi “attivi”,* ossia dove l’attributo ***Uscita*** della tabella *Accesso* è 0;
* *Il numero di clienti attualmente all’interno del centro al quale il cliente sta accedendo deve essere minore della sua capienza massima.* Questi valori sono dati dall’attributo ***ClientiDentro*** e dall’attributo ***CapienzaMassimaClienti*** della tabella *Centro*;
* *Il cliente che sta eseguendo l’accesso non deve aver già effettuato il numero massimo di accessi settimanali possibili*, indicato dall’attributo ***AccessiSettimanali*** della tabella *Contratto*;
* *Il centro nel quale il cliente sta accedendo deve essere previsto nel suo contratto*, indicato dalla tabella *Sede*.

### Scheda alimentazione

1. -- -----------------------------------------------------
2. -- Vincolo scheda alimentazione
3. -- -----------------------------------------------------
4. **DROP** **TRIGGER** IF EXISTS `Progetto`.`vincoli\_scheda\_alimentazione`;
5. DELIMITER $$
6. USE `Progetto`$$
7. **CREATE** **TRIGGER** `vincoli\_scheda\_alimentazione`
8. BEFORE **INSERT** **ON** Scheda\_alimentazione
9. **FOR** EACH ROW
10. **BEGIN**
12. IF NEW.DataInizio>=NEW.DataFine **THEN**
13. SIGNAL SQLSTATE '45000'
14. **SET** MESSAGE\_TEXT = "La data della fine di una scheda di alimentazione  non puo' essere minore di quella di inizio";
15. **END** IF;
17. **END** $$
18. DELIMITER ;

Questo vincolo generico implementato mediante **trigger** impone che *la data di inizio di una scheda di alimentazione sia minore della data di fine.*

### Rata

1. -- -----------------------------------------------------
2. -- Controllo rate
3. -- -----------------------------------------------------
4. **DROP** EVENT IF EXISTS `Progetto`.`controllo\_rate`;
5. DELIMITER $$
6. USE `Progetto` $$
7. **CREATE** EVENT `controllo\_rate`
8. **ON** SCHEDULE EVERY 1 DAY
9. STARTS '2017-11-19 23:59:59'
10. DO
11. **BEGIN**
12. **DECLARE** contratto\_ **INT** **DEFAULT** 0;
13. **DECLARE** somma\_tasse **DOUBLE** **DEFAULT** 0;
14. **DECLARE** importo\_totale **INT** **DEFAULT** 0;
15. **DECLARE** interesse **INT** **DEFAULT** 0;
16. **DECLARE** finito **INT** **DEFAULT** 0;
18. **DECLARE** cursore **CURSOR** **FOR**
19. **SELECT** D2.Codice, D1.SommaTasse, D2.Importo, D2.Interesse
20. **FROM** (**SELECT** LR.Pagamento, SUM(LR.Importo) **AS** SommaTasse
21. **FROM** log\_rata LR
22. **GROUP** **BY** LR.Pagamento) **AS** D1
23. **INNER** JOIN
24. (**SELECT** C.Codice, C.Importo, PR.Interesse
25. **FROM** Contratto C
26. **INNER** JOIN
27. Pagamento\_rateizzato PR **ON** PR.Contratto=C.Codice) **AS** D2
28. **ON** D1.Pagamento=D2.Codice;
30. **DECLARE** **CONTINUE** HANDLER **FOR** NOT FOUND
31. **SET** finito=1;
33. **OPEN** cursore;
34. preleva: LOOP
36. IF finito = 1 **THEN**
37. LEAVE preleva;
38. **END** IF;
40. **FETCH** cursore **INTO** contratto\_, somma\_tasse, importo\_totale, interesse;
42. IF (importo\_totale < ((100-interesse)\*somma\_tasse/100)-1 OR
43. importo\_totale > ((100-interesse)\*somma\_tasse/100)+1)
44. **THEN**
45. **INSERT** **INTO** Errore
46. **VALUES** (CONCAT("L'importo delle tasse per il contratto ",  contratto\_, " non e' valido"));
47. **END** IF;
49. **END** LOOP;
51. **TRUNCATE** log\_rata;
52. **END** $$
53. DELIMITER ;

Questa regola di derivazione implementata mediante **event** *ha lo scopo di controllare che la somma degli importi delle singole tasse*, relative al pagamento di un contratto, *sia uguale all’importo totale del contratto stesso,* tenendo conto del valore del tasso di interesse verso l’istituto finanziario che deve essere rimosso dall’operazione. Si ipotizza che le tasse debbano essere inserite nell’arco di una sola giornata, in modo tale da eseguire l’evento la notte (quando le palestre sono teoricamente chiuse). Viene sfruttata quindi la log table ***log\_rata*** creata durante l’operazione 6 relativamente all’inserimento del contratto, la quale sarà poi svuotata. Viene inoltre lasciato un piccolo margine di errore (il valore della somma delle tasse deve stare tra l’importo totale incrementato di 1 e decrementato di 1) che può generarsi a causa delle approssimazioni.

## Funzionalità

### Aggiornamento crediblità

1. -- -----------------------------------------------------
2. -- Aggiornamento credibilità
3. -- -----------------------------------------------------
4. **DROP** EVENT IF EXISTS `Progetto`.`aggiornamento\_credibilita`;
5. DELIMITER $$
6. USE `Progetto`$$
7. **CREATE** EVENT `aggiornamento\_credibilita`
8. **ON** SCHEDULE EVERY 7 DAY
9. STARTS '2017-09-19 23:59:00'
10. DO
11. **BEGIN**
12. **DECLARE** cliente **VARCHAR**(16) **DEFAULT** '';
13. **DECLARE** numero\_giudizi **INT** **DEFAULT** 0;
14. **DECLARE** somma\_giudizi **INT** **DEFAULT** 0;
15. **DECLARE** finito **INT** **DEFAULT** 0;
16. **DECLARE** cursore **CURSOR** **FOR**
17. **SELECT** P.Pubblicatore, COUNT(\*), SUM(G.Valore)
18. **FROM** Giudizio G
19. **INNER** JOIN
20. Post P **ON** G.Post=P.Codice
21. **GROUP** **BY** P.Pubblicatore;
23. **DECLARE** **CONTINUE** HANDLER **FOR** NOT FOUND
24. **SET** finito = 1;
26. **OPEN** cursore;
28. preleva: LOOP
29. IF finito = 1 **THEN**
30. LEAVE preleva;
31. **END** IF;
33. **FETCH** cursore **INTO** cliente, numero\_giudizi, somma\_giudizi;
34. **UPDATE** Cliente
35. **SET** Credibilita = somma\_giudizi/numero\_giudizi
36. **WHERE** CodiceFiscale=cliente;
38. **END** LOOP;
39. **END** $$
40. DELIMITER ;

Questo **event** implementa *l’aggiornamento della credibilità dell’utente* come descritto nella sezione *Analisi delle specifiche*. Per ogni utente che abbia ricevuto almeno un giudizio: viene preso il numero di giudizi ricevuti e la somma totale dei valori dei giudizi ricevuti, dopodichè viene aggiornato l’attributo ***Credibilità*** con la media ricavata dai valori precedenti.

### Rotazione del magazzino

1. -- -----------------------------------------------------
2. -- Prodotti piu' venduti
3. -- -----------------------------------------------------
4. **DROP** **PROCEDURE** IF EXISTS `Progetto`.`prodotti\_piu\_venduti`;
5. DELIMITER $$
6. USE `Progetto`$$
7. **CREATE** **PROCEDURE** prodotti\_piu\_venduti ()
8. **BEGIN**
10. **DECLARE** EXIT HANDLER **FOR** SQLEXCEPTION
11. **BEGIN**
12. **ROLLBACK**;
13. **SELECT** "Si e' verificato un errore!";
14. **END**;
16. **SELECT** \*
17. **FROM** (**SELECT** @row\_number := @row\_number+1 **AS** Rank, D.`Nome commerciale`, D.NumeroVendite
18. **FROM** (**SELECT** @row\_number := 0) **AS** N,
19. (**SELECT** I.`Nome commerciale`, COUNT(\*) **AS** NumeroVendite
20. **FROM** Integratore I
21. **INNER** JOIN
22. Acquisto A **ON** I.Codice=A.Integratore
23. **GROUP** **BY** I.`Nome commerciale`) **AS** D
24. **ORDER** **BY** D.NumeroVendite **DESC**) **AS** DD
25. **WHERE** DD.Rank<=15;
26. **END** $$
27. DELIMITER ;

Questa **stored procedure** *restituisce i 15 prodotti più venduti*, che saranno suggeriti agli utenti e ordinati automaticamente nei vari magazzini.

1. -- -----------------------------------------------------
2. -- Prodotti in scadenza
3. -- -----------------------------------------------------
4. **DROP** **PROCEDURE** IF EXISTS `Progetto`.`prodotti\_in\_scadenza`;
5. DELIMITER $$
6. USE `Progetto`$$
7. **CREATE** **PROCEDURE** prodotti\_in\_scadenza ()
8. **BEGIN**
10. **DECLARE** EXIT HANDLER **FOR** SQLEXCEPTION
11. **BEGIN**
12. **ROLLBACK**;
13. **SELECT** "Si e' verificato un errore!";
14. **END**;
16. **SELECT** \*, IF(I.Scadenza BETWEEN **CURRENT\_DATE** AND **CURRENT\_DATE** + INTERVAL 14 DAY, 30+floor(rand()\*21), 10) **AS** Sconto
17. **FROM** Integratore I
18. **WHERE** I.Scadenza BETWEEN **CURRENT\_DATE** AND **CURRENT\_DATE** + INTERVAL 1 MONTH;
19. **END** $$
20. DELIMITER ;

Questa **stored procedure** *restituisce l’elenco dei prodotti in scadenza*: quelli che scadono entro due settimane ricevono uno sconto casuale compreso tra il 30% e il 50%, mentre quelli che scadono entro un mese ricevono uno sconto del 10%.

1. -- -----------------------------------------------------
2. -- Prodotti meno venduti
3. -- -----------------------------------------------------
4. **DROP** **PROCEDURE** IF EXISTS `Progetto`.`prodotti\_meno\_venduti`;
5. DELIMITER $$
6. USE `Progetto`$$
7. **CREATE** **PROCEDURE** prodotti\_meno\_venduti ()
8. **BEGIN**
10. **DECLARE** EXIT HANDLER **FOR** SQLEXCEPTION
11. **BEGIN**
12. **ROLLBACK**;
13. **SELECT** "Si e' verificato un errore!";
14. **END**;
16. **SELECT** \*, 30 **AS** Sconto
17. **FROM** (**SELECT** @row\_number := @row\_number+1 **AS** Rank, D.`Nome commerciale`, D.NumeroVendite
18. **FROM** (**SELECT** @row\_number := 0) **AS** N,
19. (**SELECT** I.`Nome commerciale`, COUNT(\*) **AS** NumeroVendite
20. **FROM** Integratore I
21. **INNER** JOIN
22. Acquisto A **ON** I.Codice=A.Integratore
23. **GROUP** **BY** I.`Nome commerciale`) **AS** D
24. **ORDER** **BY** D.NumeroVendite) **AS** DD
25. **WHERE** DD.Rank<=10;
26. **END** $$
27. DELIMITER ;

Questa **stored procedure** *restituisce i 10 prodotti meno venduti*: essi riceveranno uno sconto del 30%.

### Reporting

1. -- -----------------------------------------------------
2. -- Report corsi meno frequentati
3. -- -----------------------------------------------------
4. **DROP** **PROCEDURE** IF EXISTS `Progetto`.`corsi\_meno\_frequentati`;
5. DELIMITER $$
6. USE `Progetto`$$
7. **CREATE** **PROCEDURE** corsi\_meno\_frequentati ()
8. **BEGIN**
10. **DECLARE** EXIT HANDLER **FOR** SQLEXCEPTION
11. **BEGIN**
12. **ROLLBACK**;
13. **SELECT** "Si e' verificato un errore!";
14. **END**;
16. **SELECT** @row\_number := @row\_number+1 **AS** Rank, D.\*
17. **FROM** (**SELECT** @row\_number := 0) **AS** N,
18. (**SELECT** \*, COUNT(\*) **AS** NumeroIscritti
19. **FROM** Corso C
20. **INNER** JOIN
21. Iscrizione I **ON** C.Codice=I.Corso
22. **GROUP** **BY** C.Codice) **AS** D
23. **ORDER** **BY** D.NumeroIscritti;
24. **END** $$
25. DELIMITER ;

Questa **stored procedure** *restituisce i corsi ordinati in maniera crescente* in base al numero di iscritti. Questo permette di vedere quali sono quelli meno frequentati.

### Performance sportiva

1. -- -----------------------------------------------------
2. -- Fedelta di esecuzione
3. -- -----------------------------------------------------
4. **DROP** **PROCEDURE** IF EXISTS `Progetto`.`fedelta\_di\_esecuzione`;
5. DELIMITER $$
6. USE `Progetto`$$
7. **CREATE** **PROCEDURE** fedelta\_di\_esecuzione (IN cliente\_ **VARCHAR**(16))
8. **BEGIN**
9. **DECLARE** esercizi\_totali **INT** **DEFAULT** 0;
10. **DECLARE** esercizi\_corretti **INT** **DEFAULT** 0;
12. **DECLARE** EXIT HANDLER **FOR** SQLEXCEPTION
13. **BEGIN**
14. **ROLLBACK**;
15. **SELECT** "Si e' verificato un errore!";
16. **END**;
18. **SELECT** COUNT(\*) **INTO** esercizi\_totali
19. **FROM** Esercizio E
20. **INNER** JOIN
21. Esercizio\_svolto ES **ON** E.Codice=ES.Esercizio
22. **WHERE** ES.Cliente=cliente\_;
24. **SELECT** COUNT(\*) **INTO** esercizi\_corretti
25. **FROM** Esercizio E
26. **INNER** JOIN
27. Esercizio\_svolto ES **ON** E.Codice=ES.Esercizio
28. **WHERE** ES.Cliente=cliente\_ AND
29. (ES.TempoImpiegato BETWEEN (E.DurataEsercizio-0.1\*E.DurataEsercizio) AND (E.DurataEsercizio+0.1\*E.DurataEsercizio));
31. **SELECT** esercizi\_corretti\*100/esercizi\_totali **AS** `Fedelta di esercizio`;
33. **END** $$
34. DELIMITER ;

Questa **stored procedure** è un *esempio di implementazione della valutazione della performance sportiva* descritta nella sezione *Analisi delle specifiche*. In particolare, essa implementa la valutazione della ***Fedeltà di esecuzione***, ossia la percentuale che indica quanti esercizi sono stati eseguiti dal cliente nel tempo consigliato per la loro esecuzione, con una tolleranza del 10% rispetto alla durata prevista. Vengono quindi calcolati gli esercizi totali eseguiti dal cliente e gli esercizi eseguiti con un tempo adeguato, dopodichè ne viene restituita la percentuale.

### Suggerimenti per le cerchie

1. -- -----------------------------------------------------
2. -- Suggerimento cerchia stessa palestra
3. -- -----------------------------------------------------
4. **DROP** **PROCEDURE** IF EXISTS `Progetto`.`suggerimento\_cerchia\_palestra`;
5. DELIMITER $$
6. USE `Progetto`$$
7. **CREATE** **PROCEDURE** suggerimento\_cerchia\_palestra (IN cliente\_ **VARCHAR**(16))
8. **BEGIN**
9. **DECLARE** sedi\_amici\_richiedenti **INT** **DEFAULT** 0;
10. **DECLARE** sedi\_amici\_riceventi **INT** **DEFAULT** 0;
12. **DECLARE** EXIT HANDLER **FOR** SQLEXCEPTION
13. **BEGIN**
14. **ROLLBACK**;
15. **SELECT** "Si e' verificato un errore!";
16. **END**;
18. **SELECT** COUNT(\*) **INTO** sedi\_amici\_richiedenti
19. **FROM** (**SELECT** S.Centro
20. **FROM** Sede S
21. **INNER** JOIN
22. Contratto C **ON** S.Contratto=C.Codice
23. **INNER** JOIN
24. Amicizia A **ON** C.Cliente=A.Richiedente
25. **WHERE** A.Ricevente=cliente\_) **AS** C0
26. **INNER** JOIN
27. Sede S0 **ON** C0.Centro=S0.Centro
28. **INNER** JOIN
29. Contratto C1 **ON** S0.Contratto=C1.Codice
30. **WHERE** C1.Cliente=cliente\_;
32. **SELECT** COUNT(\*) **INTO** sedi\_amici\_riceventi
33. **FROM** (**SELECT** S.Centro
34. **FROM** Sede S
35. **INNER** JOIN
36. Contratto C **ON** S.Contratto=C.Codice
37. **INNER** JOIN
38. Amicizia A **ON** C.Cliente=A.Ricevente
39. **WHERE** A.Richiedente=cliente\_) **AS** C0
40. **INNER** JOIN
41. Sede S0 **ON** C0.Centro=S0.Centro
42. **INNER** JOIN
43. Contratto C1 **ON** S0.Contratto=C1.Codice
44. **WHERE** C1.Cliente=cliente\_;
46. IF sedi\_amici\_richiedenti + sedi\_amici\_riceventi > 0 **THEN**
47. **SELECT** "Nella mia palestra" **AS** Suggerimento;
48. **END** IF;
50. **END** $$
51. DELIMITER ;

Questa **stored procedure** è un *esempio di implementazione di suggerimento delle cerchie* in base alle amicizie dell’utente e le loro attività, come descritto nella sezione *Analisi delle specifiche*. In particolare, implementa il suggerimento della creazione della cerchia “Nella mia palestra” nel caso in cui venga individuato almeno un amico (**Richiedente** o **Ricevente** nella tabella *Amicizia*) che ha almeno un centro a cui poter accedere in comune con il cliente sulla quale stiamo eseguendo la ricerca (dati contenuti nella tabella *Sede*).

