****

**UNIVERSITA’ DI PISA**

**CORSO DI INGEGNERIA INFORMATICA**

**BASI DI DATI**

**PROGETTO A CURA DI:**

**Chiumiento Vito**

**Speranza Ludwig**

**Introduzione**

Il progetto in questione ha come scopo la creazione di un database per la gestione dei dati di un’impresa, la cui funzionalità riguarda il settore dell’attività sportiva: l’impresa è suddivisa in una serie di centri fitness dislocati sul territorio nazionale. Il Database permette di memorizzare le specifiche di ciascun centro fitness e le informazioni relative a ciascun cliente, che può accedervi, sia tramite contratto, sia sprovvisto del suddetto citato pocanzi. E’ peculiarità del database creato, quella di contenere informazioni anche dei servizi che la palestra mette a disposizione del cliente, di tutti i progressi che quest’ultimi raggiungono attraverso un monitoraggio Smart e degli accessi effettuati che vengono poi rilevati tramite delle apparecchiature meglio note con il nome di smartwatch, utili non solo per questioni tecnico-burocratiche, ma di fondamentale importanza per una migliore gestione dei servizi concessi ai clienti dai centri, come ad esempio la giusta ripartizione degli armadietti per ciascun cliente, in modo che non vi siano conflitti riguardanti l’orario. Ogni dipendente possiede un proprio turno di lavoro, a norma di legge, e ogni sua prestazione svolta all’interno dell’impresa viene memorizzata al fine di garantire un servizio in continuo miglioramento per i clienti. Tra i servizi precedentemente citati vi è inoltre la commercializzazione di integratori e tutte le transazioni con fornitori e clienti, nonché l’attività di controllo dello spazio a disposizione del magazzino, ove vengono sistemati gli integratori, sia quelli non ancora venduti, sia quelli prossimi alla vendita. Per garantire i miglioramenti prefissati della qualità dei vari centri è fondamentale anche il consenso dei clienti e le loro opinioni espresse attraverso l’area social. Ogni cliente diventa nel contesto social un utente fornito di username e password e può accedere ad un forum all’interno del quale possono pubblicare post, condividere interessi, creare amicizie e lanciare sfide monitorando i loro progressi e confrontandoli con quelli degli altri, al fine di migliorare le proprie prestazioni sportive e garantire un ambiente più positivo. In fine il database contiene un’area Analytics che si occupa di funzionalità che permettono di analizzare i dati per ricavare informazioni utili per la buona gestione dei servizi offerti, sulle performance dei clienti e per risolvere eventuali problemi legati alla gestione degli integratori e del magazzino.

**Indice**

Introduzione pag. 1

Cardinalità Associazioni pag. 3

Traduzione delle generalizzazioni pag. 15

Modello Logico Relazionale pag. 18

Vincoli di Integrità Referenziale pag. 21

Vincoli di integrità generici pag. 26

Gestione Indici pag. 27

Normalizzazione pag. 29

Tavola dei Volumi pag. 43

Elenco delle ridondanze pag. 47

Elenco delle operazioni pag. 48

Tavola degli accessi pag. 50

Analisi costo delle ridondanze pag. 69

Area Analytics pag. 70

Script SQL

* Creazione pag. 72
* Popolamento pag. 105
* Trigger ed Event pag. 139
* Operazioni pag. 148
* Area Analytics pag. 155

**Cardinalità associazioni**

**Associazione Accede**

* Cardinalità (0,N) con Contratto, poiché un contratto può dare accesso a 0 o più piscine.
* Cardinalità (0,N) con Piscina, poiché una piscina è riferita da 0 o più contratti.

**Associazione Accede\_1**

* Cardinalità (1,N) con Contratto, poiché un contratto può dare accesso a 1 o più sale.
* Cardinalità (0,N) con Sala, poiché una sala è riferita da 0 o più contratti.

**Associazione Accede\_2**

* Cardinalità (1,N) con Contratto, poiché un contratto può dare accesso a 1 o più corsi.
* Cardinalità (1,N) con Corso, poiché un corso è riferito da 1 o più contratti.

**Associazione Acquista**

* Cardinalità (0,N) con Cliente, poiché un cliente può acquistare 0 o più integratori.
* Cardinalità (1,N) con Integratore, poiché un integratore è acquistato da uno o più integratori.

**Associazione Aderisce**

* Cardinalità (0,N) con Utente, poiché un utente può aderire a 0 o più cerchie.
* Cardinalità (1,N) con Cerchia, poiché un cerchia è composta da almeno uno o più utenti.

**Associazione Adopera**

* Cardinalità (0,N) con Utente, poiché un utente può adoperare 0 o più post.
* Cardinalità (1,1) con Post, poiché un dato post appartiene ad uno ed un solo cliente.

**Associazione Amicizia**

* Cardinalità (0,N) con Utente, poiché un utente può avere 0 o più amicizie.
* Cardinalità (0,N) con Utente, poiché un utente può avere 0 o più amicizie.

**Associazione Appartiene**

* Cardinalità (0,N) con Cliente, poiché un cliente può avere 0 o più schede di alimentazione.
* Cardinalità (1,1) con SchedaAlimentazione, poiché una scheda appartiene ad uno ed un solo cliente.

**Associazione Associa**

* Cardinalità (1,N) con Sfida, poiché una sfida può associare 1 o più schede di allenamento.
* Cardinalità (0,1) con SchedaAllenamento, poiché una scheda è associata ad una sola sfida oppure a nessuna.

**Associazione Associa\_1**

* Cardinalità (1,N) con Sfida, poiché una sfida può associare 1 o più schede di alimentazione.
* Cardinalità (0,1) con SchedaAlimentazione, poiché una scheda è associata ad una sola sfida oppure a nessuna.

**Associazione Autorizza**

* Cardinalità (1,N) con CentroFitness, poiché in 1 centro ci possono essere 1 o più accessi.
* Cardinalità (1,1) con Accesso, poiché 1 accesso avviene in 1 ed un solo centro.

**Associazione Configura**

* Cardinalità (1,N) con Medico, poiché in 1 medico configura 1 o più diete.
* Cardinalità (1,1) con Dieta, poiché tale dieta è prescritta da 1 ed un solo medico .

**Associazione Contenuta**

* Cardinalità (1,N) con Dieta, poiché una stessa dieta può essere presente in una o più schede di alimentazione.
* Cardinalità (1,N) con SchedaAlimentazione, poiché una scheda può prevedere una o più diete.

**Associazione Corrisponde**

* Cardinalità (1,N) con Post\_Risposta, poiché un post di risposta ha uno o più giudizi.
* Cardinalità (1,1) con Giudizio, poiché tale giudizio è proprio di uno ed un solo post di risposta.

**Associazione Dirige**

* Cardinalità (0,N) con Dipendente, poiché un dipendente dirige 0 o più corsi.
* Cardinalità (1,1) con Corso, poiché ogni corso è gestito da uno ed un solo dipendente.

**Associazione Effettua**

* Cardinalità (1,N) con Cliente, poiché un cliente effettua uno o più accessi al centro.
* Cardinalità (1,1) con Accesso, poiché l’accesso effettuato è proprio del cliente e non può essere uguale a quello di un altro.

**Associazione Effettua\_1**

* Cardinalità (1,N) con CentroFitness, poiché un centro effettua uno o più ordini al fornitore.
* Cardinalità (1,1) con Ordine, poiché l’ordine effettuato è proprio del centro.

**Associazione Esprime**

* Cardinalità (0,N) con Utente, poiché un utente può esprimere 0 o più valutazioni.
* Cardinalità (1,1) con Valutazioni, poiché la valutazione effettuata è propria dell’utente.

**Associazione E’\_Assegnato**

* Cardinalità (1,1) con Cliente, poiché ad un cliente è associato uno ed un solo utente.
* Cardinalità (1,1) con Utente, poiché un utente è riferito solo ed esclusivamente ad un cliente.

**Associazione E’\_Assegnato1**

* Cardinalità (1,N) con Medico, poiché un medico segue uno o più clienti.
* Cardinalità (1,1) con Cliente, poiché ad ogni cliente è assegnato un ed un solo medico .

**Associazione E’\_Composta**

* Cardinalità (1,N) con SchedaAllenamento, poiché una scheda è composta da uno o più esercizi.
* Cardinalità (1,1) con Esercizio, poiché ogni esercizio è proprio di una ed una sola scheda .

**Associazione E’\_Consegnata**

* Cardinalità (0,N) con Dipendente, poiché un dipendente può consegnare 0 o più schede di allenamento.
* Cardinalità (1,1) con SchedaAllenamento, poiché ogni scheda è propria di uno ed un solo dipendente.

**Associazione E’\_Dotato**

* Cardinalità (0,N) con CentroFitness, poiché un centro può avere a disposizione 0 o più piscine.
* Cardinalità (1,1) con Piscina, poiché ogni piscina è propria di ogni centro fitness.

**Associazione E’\_Dotato1**

* Cardinalità (1,N) con CentroFitness, poiché un centro può avere a disposizione una o più apparecchiature.
* Cardinalità (1,1) con Apparecchiatura, poiché ogni apparecchiatura è in uno ed un solo centro fitness.

**Associazione E’\_In**

* Cardinalità (0,N) con Sala, poiché una sala può avere a disposizione 0 o più apparecchiature.
* Cardinalità (1,1) con Apparecchiatura, poiché ogni apparecchiatura è in una ed un sola sala.

**Associazione E’\_Sottoscritto**

* Cardinalità (0,1) con Cliente, poiché una cliente sottoscrive un massimo di un contratto o può anche non sottoscrivere alcun contratto.
* Cardinalità (1,1) con Contratto, poiché ogni contratto è proprio di uno ed un solo cliente.

**Associazione Fa\_Uso**

* Cardinalità (0,N) con Sfida, poiché una sfida fa uso di 0 o più post.
* Cardinalità (1,N) con Post, poiché un post è in una o più sfide.

**Associazione Fascia**

* Cardinalità (1,N) con CentroFitness, poiché un centro ha una o più fasce orarie.
* Cardinalità (1,1) con Oraria, poiché ogni fascia è propria di ogni centro fitness.

**Associazione Frequenta**

* Cardinalità (1,N) con CentroFitness, poiché ogni centro è frequentato da uno o più clienti.
* Cardinalità (1,N) con Cliente, poiché ogni cliente frequenta uno o più centri.

**Associazione Giace**

* Cardinalità (1,N) con Magazzino, poiché in ogni magazzino giacciono uno o più integratori.
* Cardinalità (1,1) con Integratore, poiché ogni singolo integratore giace in uno ed un solo magazzino.

**Associazione Gestisce**

* Cardinalità (1,N) con Dipendente, poiché un dipendente gestisce una o più sale.
* Cardinalità (1,N) con Sala, poiché ogni sala è gestita da uno o più dipendenti.

**Associazione Gestisce\_1**

* Cardinalità (1,N) con Utente, poiché un utente gestisce una o più cerchie.
* Cardinalità (1,1) con Cerchia, poiché ogni cerchia è propria di un utente.

**Associazione Ha\_1**

* Cardinalità (1,N) con Cliente, poiché un cliente gestisce uno o più scopi.
* Cardinalità (1,1) con Scopo, poiché uno scopo è proprio di uno ed un solo cliente.

**Associazione Ha\_2**

* Cardinalità (1,N) con Apparecchiatura, poiché un’apparecchiatura ha una o più regolazioni.
* Cardinalità (1,1) con Regolazione, poiché una regolazione è propria di una ed un sola apparecchiatura.

**Associazione Lavora\_In**

* Cardinalità (1,N) con Dipendente, poiché un dipendente lavora in uno o più centri.
* Cardinalità (1,N) con CentroFitness, poiché in un centro lavorano uno o più dipendente.

**Associazione Link**

* Cardinalità (0,N) con Post, poiché un post può avere 0 o più link esterni.
* Cardinalità (1,1) con Esterno, poiché un link esterno è proprio di un post.

**Associazione Nutre**

* Cardinalità (1,N) con Cerchia, poiché in una cerchia nutre uno o più interessi.
* Cardinalità (0,N) con Interesse, poiché un interesse è nutrito da 0 o più cerchie.

**Associazione Partecipa**

* Cardinalità (1,N) con Cliente, poiché un cliente partecipa ad uno o più corsi.
* Cardinalità (1,N) con Corso, poiché in un corso partecipano uno o più clienti.

**Associazione Partecipa\_1**

* Cardinalità (0,N) con Utente, poiché un utente partecipa ad uno o più sfide.
* Cardinalità (1,N) con Sfida, poiché ad ogni sfida partecipano uno o più utenti .

**Associazione Possiede**

* Cardinalità (0,N) con Cliente, poiché un cliente possiede 0 o più schede di allenamento.
* Cardinalità (1,1) con SchedaAllenamento, poiché ogni scheda è propria di uno ed un solo cliente.

**Associazione Prende**

* Cardinalità (1,N) con Medico, poiché un medico prende una o più misurazioni.
* Cardinalità (1,1) con Misurazione, poiché ogni misurazione è propria di uno ed un solo medico.

**Associazione Prevede**

* Cardinalità (1,N) con Sfida, poiché una sfida prende una o più valutazioni.
* Cardinalità (1,1) con Valutazione, poiché ogni valutazione è propria di una ed una sola sfida.

**Associazione Prevede\_1**

* Cardinalità (1,1) con Contratto, poiché ogni contratto ha uno ed un solo pagamento.
* Cardinalità (1,1) con Pagamento, poiché ogni pagamento è proprio di uno ed un solo contratto.

**Associazione Ricevuto**

* Cardinalità (1,N) con Fornitore, poiché un fornitore riceve uno o più ordine.
* Cardinalità (1,1) con Ordine, poiché ogni ordine è proprio di uno ed un solo fornitore.

**Associazione Richiede**

* Cardinalità (0,N) con Esercizio, poiché ogni esercizio richiede 0 o più apparecchiature.
* Cardinalità (1,N) con Apparecchiatura, poiché ogni apparecchiatura serve per uno o più esercizi.

**Associazione Rilevata**

* Cardinalità (1,N) con Cliente, poiché ad ogni cliente viene rilevata una o più misurazioni.
* Cardinalità (1,1) con Misurazione, poiché ogni misurazione è propria di uno ed un solo cliente.

**Associazione Ripetizione**

* Cardinalità (1,N) con Esercizio, poiché ogni esercizio richiede 1 o più sessioni.
* Cardinalità (1,1) con Sessione, poiché ogni sessione è propria di uno ed un solo esercizio.

**Associazione Riserva**

* Cardinalità (1,N) con Armadietto, poiché ad ogni armadietto è riservato 1 o più accessi.
* Cardinalità (1,1) con Accesso, poiché ogni accesso riserva uno ed un solo armadietto.

**Associazione Risponde**

* Cardinalità (0,N) con Post, poiché ogni post risponde ad uno o più post di risposta.
* Cardinalità (1,1) con Post\_Risposta, poiché ogni post di risposta è riservato ad uno ed un solo post.

**Associazione Scrive**

* Cardinalità (0,N) con Utente, poiché ogni utente scrive 0 o più post di risposta.
* Cardinalità (1,1) con Post\_Risposta, poiché ogni post di risposta è proprio di uno ed un solo utente.

**Associazione Segue**

* Cardinalità (1,N) con Corso, poiché ogni corso segue uno o più calendari.
* Cardinalità (1,1) con Calendari, poiché ogni calendario è proprio di uno ed un solo utente.

**Associazione Si\_Associa**

* Cardinalità (1,N) con Dieta, poiché una dieta può associarsi ad uno o più integratore.
* Cardinalità (1,N) con Integratore, poiché un integratore può associarsi ad uno o più integratori.

**Associazione Si\_Colloca**

* Cardinalità (1,N) con CentroFitness, poiché in ogni centro si collocano uno o più clienti.
* Cardinalità (1,1) con Sala, poiché ogni sala è propria di uno ed un solo centro.

**Associazione Si\_Divide**

* Cardinalità (0,N) con Pagamento, poiché ogni pagamento si divide in 0 o più rate.
* Cardinalità (1,1) con Rata, poiché ogni rata è propria di uno ed un solo pagamento.

**Associazione Si\_Tiene**

* Cardinalità (1,N) con Sala, poiché in ogni sala ci sono uno o più corsi.
* Cardinalità (0,1) con Corso, poiché ogni corso è in 0 o massimo una sala.

**Associazione Situato**

* Cardinalità (1,N) con Spogliatoio, poiché in ogni spogliatoio ci sono uno o più armadietto.
* Cardinalità (1,1) con Armadietto, poiché ogni armadietto è proprio di uno ed un solo spogliatoio.

**Associazione Sostiene**

* Cardinalità (1,N) con Piscina, poiché ogni piscina è usata in uno o più corsi.
* Cardinalità (0,1) con Corso, poiché ogni corso prevede l’uso di 0 o 1 piscina.

**Associazione Stipula**

* Cardinalità (0,N) con CentroFitness, poiché un centro stipula 0 o più contratti.
* Cardinalità (1,3) con Contratto, poiché ogni contratto consente l’accesso ad un minimo di uno ed un massimo di tre centri.

**Associazione Stila**

* Cardinalità (1,N) con Medico, poiché ogni medico stila una o più schede di alimentazione.
* Cardinalità (1,1) con SchedaAlimentazione, poiché ogni scheda è propria di uno ed un solo medico.

**Associazione Turnazione**

* Cardinalità (1,N) con Dipendente, poiché un dipendente ha una turnazione di uno o più personale.
* Cardinalità (1,1) con Personale, poiché ogni turnazione è propria di 1 ed 1 solo dipendente.

**Associazione Vende**

* Cardinalità (1,N) con Fornitore, poiché un fornitore vende 1 o più integratori.
* Cardinalità (1,1) con Integratore, poiché un integratore è proprio di uno ed un solo fornitore.

Traduzione delle generalizzazioni

* E’ presente una generalizzazione parziale con una sola entità figlia.

E’ stata tradotta senza effettuare alcun accorpamento: all’interno dell’E-R ristrutturato saranno presenti sia l’entità padre **Scopo** che l’entità figlia **PotenziamentoMuscolare.**





* E’ presente una generalizzazione totale con tre entità figlie.

E’ stata tradotta effettuando l’accorpamento dei figli sul padre: all’interno dell’E-R ristrutturato sarà presente la sola entità **Contratto**.





* E’ presente una generalizzazione parziale con tre entità figlie.

E’ stata tradotta effettuando l’accorpamento dei figli sul padre: all’interno dell’E-R ristrutturato sarà presente la sola entità **Dipendente**.





**Modello Logico Relazionale**

* **CentroFitness** (IdCentro, CAP, Dimensione, NumCivico, Città, N°Telefono, MaxClienti, Via);
* **Lavora\_In** (CentroFitness, Dipendente, Ruolo);
* **Dipendente** (CodFiscale, Nome, Cognome, CAP, DataNascita, Via, NumCivico, Città, N°Prefettura, CodDocumento);
* **Frequenta** (CentroFitness, Cliente);
* **Cliente** (CodFiscale, Nome, Cognome, Città, NumCivico, CAP, N°Prefettura, CodDocumento, DataNascita, Via, PreferenzeAlimentari, Medico);
* **Stipula** (CentroFitness, Contratto);
* **Contratto** (IdContratto, Durata, Costo, Variante, MaxIngressi, MaxIngressiPiscina, CodConsulente, DataSottoscrizione, Cliente);
* **Accede** (Contratto, Piscina);
* **Piscina** (IdPiscina, Dimensione, Tipo, CentroFitness);
* **Gestisce** (Dipendente, Sala);
* **Accede\_1** (Contratto, Sala);
* **Sala** (IdSala, NomeSala, CentroFitness);
* **Accede\_2** (Contratto¸ Corso);
* **Partecipa** (Cliente, Corso);
* **Corso** (IdCorso, MaxPartecipanti, NomeDisciplina, Livello, DataInizio, DataFine, Sala,Piscina,Dipendente);
* **Scopo** (IdScopo, Tipo, Cliente);
* **PotenziamentoMuscolare** (IdScopo, Muscolo, Livello);
* **Medico** (CodMedico, Nome, Cognome, Telefono, Studio);
* **Misurazione** (IdMisurazione, AcquaTotale, Data, MassaGrassa, MassaMagra, Peso, DataVisita, Cliente, Medico);
* **SchedaAlimentazione** (IdSchedaAlimentazione, Obiettivo, DataInizio, DataFine,Intervallo, Cliente, Medico, Sfida);
* **Contenuta** (SchedaAlimentazione, Dieta);
* **Dieta** (IdDieta, MaxKilocal, MinKilocal, N°Pasti, Pasto, Medico);
* **Magazzino** (IdMagazzino, Giacenza);
* **Acquista** (Cliente, Integratore, Data, Quantità);
* **Si\_Associa** (Integratore, Dieta);
* **Integratore** (IdIntegratore, DataScadenza, Forma, NomeCommerciale, NomeSostanza, QuantitàSostanza, N°Pezzi, Magazzino, Fornitore);
* **Fornitore** (IdFornitore, PartitaIva, N°Telefono, CAP, FormaSocietaria, Città, NumCivico, Via, Nome);
* **Ordine** (CodiceInterno, CodiceEsterno, DataConsegna, DataEvasione, Stato, Fornitore, CentroFitness);
* **Apparecchiatura** (IdApparecchiatura, Nome, Tipo, Consumo, Manutenzione, PercentualeUsura, Sala, CentroFitness);
* **Richiede** (Apparecchiatura, Esercizio);
* **Esercizio** (IdEsercizio, Nome, DispendioEnergetico, Peso, NRipetizioni, Tipo, IstanteInizio, Durata, SchedaAllenamento);
* **SchedaAllenamento** (IdSchedaAllenamento, Peso, Stato, DataInizio, EntitàStato, Altezza, MassaGrassa, AcquaTotale, DataFine, MassaMagra, Dipendente, Cliente, Sfida);
* **Sessione** (IdSessione, Recupero, IstanteFine, RipetizioniEffettive, IstanteInizio, Esercizio);
* **Personale** (Dipendente, Giorno, OrarioIngresso, OrarioUscita);
* **Accesso** (IdAccesso, Data, OrarioIngresso, OrarioUscita, Tariffa, Durata, CentroFitness, Armadietto, Cliente, Smartwatch);
* **Armadietto** (IdArmadietto, CodiceSblocco, Spogliatoio);
* **Spogliatoio** (IdSpogliatoio, Capienza, N°Posti, Posizione);
* **Oraria** (CentroFitness, Giorno, OrarioApertura, OrarioChiusura);
* **Rata** (IdRata, IstitutoFinanziario, Importo, DataScadenza, TassoInteresse, Pagamento);
* **Pagamento** (IdPagamento, ImportoTotale, Stato, ImportoAttuale, Contratto);
* **Calendario** (Corso, Giorno, OrarioInizio, OrarioFine);
* **Regolazione** (ConfigurazioneUtilizzo, Inclinazione, Carico, Velocità, Tempo, Intensità, Apparecchiatura);
* **Utente** (Username, Password, Cliente);
* **Amicizia** (UtenteRichiedente, UtenteRicevente, DataRichiesta, DataAccettazione);
* **Aderisce** (Utente, Cerchia);
* **Cerchia** (IdCerchia, NomeCerchia, MaxPartecipanti, NumPartecipanti, Utente);
* **Partecipa \_1**(Utente, Sfida, Ruolo);
* **Sfida** (IdSfida, Scopo, DataLancio, DataInizio, Scadenza);
* **Valutazione** (IdValutazione, KcalBruciate, BattitoCardiaco, ConsumoOssigeno, SforzoPsichico, DataFine, Utente, Sfida);
* **Nutre** (Cerchia, Interesse);
* **Interesse** (IdInteresse, Tipo);
* **Fa\_Uso** (Post, Sfida);
* **Post** (IdPost, Testo, AreaForum, Timestamp, Utente);
* **PostRisposta** (IdPostRisposta, AreaForum, Testo, Timestamp, Utente, Post);
* **Giudizio** (PostRisposta, CodGiudizio, Modalità, Voto);
* **Esterno** (Url, Dominio, Post);

**Vincoli di Integrità Referenziale**

* Esiste un vincolo di integrità referenziale tra l’attributo **CentroFitness** della tabella **Oraria** e l’attributo **IdCentro** della tabella **CentroFitness;**
* Esiste un vincolo di integrità referenziale tra l’attributo **CentroFitness** della tabella **Apparecchiatura** e l’attributo **IdCentro** della tabella **CentroFitness;**
* Esiste un vincolo di integrità referenziale tra l’attributo **CentroFitness** della tabella **Accesso** e l’attributo **IdCentro** della tabella **CentroFitness;**
* Esiste un vincolo di integrità referenziale tra l’attributo **CentroFitness** della tabella **Lavora­In** e l’attributo **IdCentro** della tabella **CentroFitness;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Dipendente** della tabella **LavoraIn** e l’attributo **CodFiscale** della tabella **Dipendente;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Dipendente** della tabella **Corso** e l’attributo **CodFiscale** della tabella **Dipendente;**
* Esiste un vincolo di integrità referenziale tra l’attributo **CentroFitness** della tabella **Ordine** e l’attributo **IdCentro** della tabella **CentroFitness;**
* Esiste un vincolo di integrità referenziale tra l’attributo **CentroFitness** della tabella **Frequenta** e l’attributo **IdCentro** della tabella **CentroFitness;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Cliente** della tabella **Frequenta** e l’attributo **CodFiscale** della tabella **Cliente;**
* Esiste un vincolo di integrità referenziale tra l’attributo **CentroFitness** della tabella **Stipula** e l’attributo **IdCentro** della tabella **CentroFitness;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Contratto** della tabella **Stipula** e l’attributo **IdContratto** della tabella **Contratto;**
* Esiste un vincolo di integrità referenziale tra l’attributo **CentroFitness** della tabella **Piscina** e l’attributo **IdCentro** della tabella **CentroFitness;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Sala** della tabella **CentroFitness** e l’attributo **IdSala** della tabella **Sala;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Piscina** della tabella **Accede** e l’attributo **IdPiscina** della tabella **Piscina;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Contratto** della tabella **Accede** e l’attributo **IdContratto** della tabella **Contratto;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Piscina** della tabella **Corso** e l’attributo **IdPiscina** della tabella **Piscina;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Sala** della tabella

**Apparecchiatura** e l’attributo **IdSala** della tabella **Sala;**

* Esiste un vincolo di integrità referenziale tra l’attributo **Sala** della tabella

**Gestisce** e l’attributo **IdSala** della tabella **Sala;**

* Esiste un vincolo di integrità referenziale tra l’attributo **Dipendente** della tabella **Gestisce** e l’attributo **CodFiscale** della tabella **Dipendente;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Sala** della tabella

**Corso** e l’attributo **IdSala** della tabella **Sala;**

* Esiste un vincolo di integrità referenziale tra l’attributo **Sala** della tabella

**Accede1** e l’attributo **IdSala** della tabella **Sala;**

* Esiste un vincolo di integrità referenziale tra l’attributo **Contratto** della tabella **Accede1** e l’attributo **IdContratto** della tabella **Contratto;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Corso** della tabella **Calendario** e l’attributo **IdCorso** della tabella **Corso;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Contratto** della tabella **Accede2** e l’attributo **IdContratto** della tabella **Contratto;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Corso** della tabella **Accede2** e l’attributo **IdCorso** della tabella **Corso;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Corso** della tabella **Partecipa** e l’attributo **IdCorso** della tabella **Corso;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Cliente** della tabella **Partecipa** e l’attributo **CodFiscale** della tabella **Cliente;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Cliente** della tabella **Contratto** e l’attributo **CodFiscale** della tabella **Cliente;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Contratto** della tabella **Pagamento** e l’attributo **IdContratto** della tabella **Contratto;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Pagamento** della tabella **Rata** e l’attributo **IdPagamento** della tabella **Pagamento;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Dipendente** della tabella **Personale** e l’attributo **CodFiscale** della tabella **Dipendente;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Dipendente** della tabella **SchedaAllenamento** e l’attributo **CodFiscale** della tabella **Dipendente;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Apparecchiatura** della tabella **Regolazione** e l’attributo **IdApparecchiatura** della tabella **Apparecchiatura;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Apparecchiatura** della tabella **Richiede** e l’attributo **IdApparecchiatura** della tabella **Apparecchiatura;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Esercizio** della tabella **Richiede** e l’attributo **IdEsercizio** della tabella **Esercizio;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Armadietto** della tabella **Accesso** e l’attributo **IdArmadietto** della tabella **Armadietto;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Cliente** della tabella **Accesso** e l’attributo **CodFiscale** della tabella **Cliente;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Spogliatoio** della tabella **Armadietto** e l’attributo **IdSpogliatoio** della tabella **Spogliatoio;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Esercizio** della tabella **Sessione** e l’attributo **IdEsercizio** della tabella **Esercizio;**
* Esiste un vincolo di integrità referenziale tra l’attributo **SchedaAllenamento** della tabella **Esercizio** e l’attributo **IdSchedaAllenamento** della tabella **SchedaAllenamento;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Cliente** della tabella **SchedaAllenamento** e l’attributo **CodFiscale** della tabella **Cliente;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Sfida** della tabella **SchedaAllenamento** e l’attributo **IdSfida** della tabella **Sfida;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Cliente** della tabella **Scopo** e l’attributo **CodFiscale** della tabella **Cliente;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Medico** della tabella **Cliente** e l’attributo **CodMedico** della tabella **Medico;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Cliente** della tabella **Misurazione** e l’attributo **CodFiscale** della tabella **Cliente;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Medico** della tabella **Misurazione** e l’attributo **CodMedico** della tabella **Medico;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Magazzino** della tabella **Integratore** e l’attributo **IdMagazzino** della tabella **Magazzino;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Cliente** della tabella **Acquista** e l’attributo **CodFiscale** della tabella **Cliente;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Integratore** della tabella **Acquista** e l’attributo **IdIntegratore** della tabella **Integratore;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Cliente** della tabella **SchedaAlimentazione** e l’attributo **CodFiscale** della tabella **Cliente;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Cliente** della tabella **Utente** e l’attributo **CodFiscale** della tabella **Cliente;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Medico** della tabella **SchedaAlimentazione** e l’attributo **CodMedico** della tabella **Medico;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Sfida** della tabella **SchedaAlimentazione** e l’attributo **IdSfida** della tabella **Sfida;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Medico** della tabella **Dieta** e l’attributo **CodMedico** della tabella **Medico;**
* Esiste un vincolo di integrità referenziale tra l’attributo **SchedaAlimentazione** della tabella **Contenuta** e l’attributo **IdSchedaAlimentazione** della tabella **SchedaAlimentazione;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Dieta** della tabella **Contenuta** e l’attributo **IdDieta** della tabella **Dieta;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Integratore** della tabella **SiAssocia** e l’attributo **IdIntegratore** della tabella **Integratore;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Dieta** della tabella **SiAssocia** e l’attributo **IdDieta** della tabella **Dieta;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Fornitore** della tabella **Integratore** e l’attributo **IdFornitore** della tabella **Fornitore;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Fornitore** della tabella **Ordine** e l’attributo **IdFornitore** della tabella **Fornitore;**
* Esiste un vincolo di integrità referenziale tra l’attributo **PostRisposta** della tabella **Giudizio** e l’attributo **IdPostRisposta** della tabella **PostRisposta;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Utente** della tabella **PostRisposta** e l’attributo **CodFiscale** della tabella **Utente;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Post** della tabella

**PostRisposta** e l’attributo **IdPost** della tabella **Post;**

* Esiste un vincolo di integrità referenziale tra l’attributo **Utente** della tabella **Post** e l’attributo **Username** della tabella **Utente;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Utente** della tabella **Aderisce** e l’attributo **Username** della tabella **Utente;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Cerchia** della tabella **Aderisce** e l’attributo **IdCerchia** della tabella **Cerchia;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Utente** della tabella **Cerchia** e l’attributo **Username** della tabella **Utente;**
* Esiste un vincolo di integrità referenziale tra l’attributo **UtenteRichiedente** della tabella **Amicizia** e l’attributo **Username** della tabella **Utente;**
* Esiste un vincolo di integrità referenziale tra l’attributo **UtenteRicevente** della tabella **Amicizia** e l’attributo **Username** della tabella **Utente;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Utente** della tabella **Valutazione** e l’attributo **Username** della tabella **Utente;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Utente** della tabella **Partecipa\_1** e l’attributo **Username** della tabella **Utente;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Sfida** della tabella **Partecipa\_1** e l’attributo **IdSfida** della tabella **Sfida;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Cerchia** della tabella **Nutre** e l’attributo **IdCerchia** della tabella **Cerchia;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Interesse** della tabella **Nutre** e l’attributo **IdInteresse** della tabella **Interesse;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Sfida** della tabella **Valutazione** e l’attributo **IdSfida** della tabella **Sfida;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Sfida** della tabella **FaUso** e l’attributo **IdSfida** della tabella **Sfida;**
* Esiste un vincolo di integrità referenziale tra l’attributo **Post** della tabella

**FaUso** e l’attributo **IdPost** della tabella **Post;**

* Esiste un vincolo di integrità referenziale tra l’attributo **Post** della tabella

**Esterno** e l’attributo **IdPost** della tabella **Post;**

**Vincoli di integrità generici**

* Gli attributi *Giorno* e *Ora*della tabella **Personale** devono rispettare i turni del personale:

un dipendente non puo’ lavorare più di 8 ore al giorno e non puo’ tenere più di un corso contemporaneamente;

* Gli attributo *Giorno* e *Ora* della tabella **Calendario** devono rispettare gli orari di apertura del centro presenti nella tabella **Oraria;**
* **SchedaAllenamento ->** *EntitàStato* {Sottopeso, Normopeso, Sovrappeso};
* **Contratto ->** *Variante* {Silver, Gold, Platinum};
* **Scopo ->** *Tipo* {Potenziamento Muscolare, Dimagrimento, Scopo Ricreativo};
* **PotenziamentoMuscolare ->** *Livello* {Lieve, Moderato, Elevato};
* **Pagamento ->** *Stato* {Eseguito, Non ancora dovuto, Scaduto};
* **Corso ->** *Livello* {Principiante, Intermedio, Avanzato};
* **Ordine ->** *Stato* {Incompleto, Evaso};
* **Giudizio ->** *Voto* {>=1 AND <=5};
* **Piscina ->** *Tipo* {Interna, Esterna};
* **SchedaAlimentazione ->** *Obiettivo* {Perdere Peso, Acquisire Massa Muscolare, Prendere Peso, Diminuire Massa Grassa};
* **Apparecchiatura ->** *PercentualeUsura* {>=0 AND <=100}

**-**> *Manutenzione* {SI, NO};

* **Regolazione ->** *Inclinazione* {>=0 AND <=25}

**->***Velocità* {>=0 AND <=20}

**->** *Intensità* {>=1 AND <=10}

**->** *Carico* {>=0 AND <=350};

* **Valutazione ->** *BattitoCardiaco* {>=85 AND <=220}

**->** *SforzoPsichico* {Basso, Medio, Alto};

**Gestione degli indici**

**Indice di massa corporea:**

Per calcolare l’indice di massa corporea occorre eseguire dei calcoli tramite l’altezza e il peso: in particolare il calcolo è espresso come il rapporto tra peso ed il quadrato dell’altezza di un individuo. Un paziente è normopeso se ha una percentuale di massa magra maggiore della percentuale di massa grassa. Ipotizzando che un individuo abbia altezza pari a 195 cm e peso espresso in 75 kg, allora il suo indice di massa corporea si calcolerà nel seguente modo:

1. Altezza al quadrato: 1,95 \* 1,95 = 3,802.
2. Peso/Altezza al quadrato: 75/3,802 = 19,726.

Gli indici di riferimento per descrivere l’entita dell’individuo sono i seguenti:

|  |  |  |
| --- | --- | --- |
| Stato | Min | Max |
| Sovrappeso | ≥ 25,0 |  |
| Normopeso | 18,50 | 24,99 |
| aottopeso |  | <18,50 |

**Indice di manutenzione delle apparecchiature:**

Un attrezzo possiede una percentuale di usura che aumenta in relazione al tempo durante il quale esso è usato. Il livello di usura aumenta di un 5% ogni quattro settimane.

Una volta raggiunto il 100% nella percentuale di usura dell’attrezzo si può procedere con la manutenzione dello stesso. Non appena l’attrezzo torna a disposizione, essendo stato in manutenzione per una settimana, può nuovamente essere inserito tra le apparecchiature utilizzabili nel centro.

**Indice di sforzo psichico:**

Il livello del battito cardiaco determina il livello dello sforzo psichico al quale è sottoposto il cliente durante l’attività fisica. I 3 livelli di sforzo si dividono in: “Basso”, “Medio”, “Alto”, a seconda che:

|  |  |  |
| --- | --- | --- |
| Livello | Min (Battito cardiaco) | Max (Battito cardiaco) |
| Alto | 200 | 220 |
| Medio | 121 | 199 |
| Basso | ≥80 | 120 |

**Normalizzazione**

**Tabella CentroFitness:**

**CentroFitness**(IdCentro, CAP, Dimensione, NumCivico, Città, N°Telefono,

MaxClienti, Via);

* IdCentro -> CAP, Dimensione, NumCivico, Città, N°Telefono, MaxClienti, Via

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

**Tabella Dipendente:**

**Dipendente** (CodFiscale, Nome, Cognome, CAP, DataNascita, Via, NumCivico, Città, N°Prefettura, CodDocumento);

* CodFiscale -> Nome, Cognome, CAP, DataNascita, Via, NumCivico,Città, N°Prefettura, CodDocumento

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

**Tabella Cliente:**

**Cliente** (CodFiscale, Nome, Cognome, Città, NumCivico, CAP, N°Prefettura,CodDocumento,DataNascita,Via, PreferenzeAlimentari, Medico);

* CodFiscale -> Nome, Cognome, Città, NumCivico, CAP, N°Prefettura,CodDocumento,DataNascita,Via, PreferenzeAlimentari, Medico

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

**Tabella Contratto:**

**Contratto** (IdContratto, Durata, Costo, Variante, MaxIngressi, MaxIngressiPiscina, CodConsulente, DataSottoscrizione, Cliente);

* IdContratto -> Durata, Costo, Variante, MaxIngressi, MaxIngressiPiscina, CodConsulente, DataSottoscrizione, Cliente

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

**Tabella Piscina:**

**Piscina** (IdPiscina, Dimensione, Tipo, CentroFitness);

* IdPiscina -> Dimensione, Tipo, CentroFitness

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

**Tabella Sala:**

**Sala** (IdSala, NomeSala, CentroFitness);

* IdSala -> NomeSala, CentroFitness

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

**Tabella Corso:**

**Corso** (IdCorso, MaxPartecipanti, NomeDisciplina, Livello, DataInizio,

DataFine, Sala, Piscina, Dipendente);

* IdCorso -> MaxPartecipanti, NomeDisciplina, Livello, DataInizio, DataFine, Sala, Piscina, Dipendente

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

**Tabella Scopo:**

**Scopo** (IdScopo, Tipo, Cliente);

* IdScopo -> Tipo, Cliente

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

**Tabella Medico:**

**Medico** (CodMedico, Nome, Cognome, Telefono, Studio);

* CodMedico -> Nome, Cognome, Telefono, Studio

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

**Tabella Misurazione:**

**Misurazione** (IdMisurazione, AcquaTotale, Data, MassaGrassa, Massa

Magra, Peso, DataVisita, Cliente, Medico);

* IdMisurazione -> AcquaTotale, Data, MassaGrassa, MassaMagra, Peso, DataVisita, Cliente, Medico

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

**Tabella Potenzia mentoMuscolare:**

**PotenziamentoMuscolare** (IdScopo, Muscolo, Livello);

* IdScopo -> Muscolo, Livello

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

**Tabella SchedaAlimentazione:**

**SchedaAlimentazione**(IdSchedaAlimentazione, Obiettivo, DataInizio,

DataFine, Intervallo, Cliente, Medico, Sfida);

* IdSchedaAlimentazione -> Obiettivo, DataInizio, DataFine, Intervallo, Cliente, Medico, Sfida

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

**Tabella Dieta:**

**Dieta** (IdDieta, MaxKilocal, MinKilocal, N°Pasti, Pasto, Medico);

* IdDieta -> MaxKilocal, MinKilocal, N°Pasti, Pasto, Medico

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

**Tabella Magazzino:**

**Magazzino (**IdMagazzino, Giacenza**);**

* IdMagazzino -> Giacenza

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

**Tabella Acquista:**

**Acquista** (Cliente, Integratore, Data, Quantità);

* Cliente, Integratore -> Data, Quantità

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

**Tabella Integratore:**

**Integratore (**IdIntegratore, DataScadenza, Forma, NomeCommerciale, NomeSostanza, QuantitàSostanza, N°Pezzi, Magazzino, Fornitore);

* IdIntegratore -> DataScadenza, Forma, NomeCommerciale, NomeSostanza, QuantitàSostanza, N°Pezzi, Magazzino, Fornitore

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

**Tabella Fornitore:**

**Fornitore** (IdFornitore, PartitaIva, N°Telefono, CAP, FormaSocietaria, Città, NumCivico, Via, Nome);

* IdFornitore -> PartitaIva, N°Telefono, CAP, FormaSocietaria,Città, NumCivico, Via, Nome

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

**Tabella Ordine:**

**Ordine** (CodiceInterno, CodiceEsterno, DataConsegna, DataEvasione, Stato, Fornitore, CentroFitness);

* CodiceInterno -> CodiceEsterno, DataConsegna, DataEvasione, Stato, Fornitore, CentroFitness

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

**Tabella Apparecchiatura:**

**Apparecchiatura** (IdApparecchiatura, Nome, Tipo, Consumo, Manutenzione, PercentualeUsura, Sala, CentroFitness);

* IdApparecchiatura -> Nome, Tipo, Consumo, Manutenzione, PercentualeUsura, Sala, CentroFitness

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

**Tabella Esercizio:**

**Esercizio** (IdEsercizio, Nome, DispendioEnergetico, Peso, NRipetizioni, Tipo, Durata, SchedaAllenamento);

* IdEsercizio -> Nome, DispendioEnergetico, Peso, NRipetizioni, Tipo, IstanteInizio, Durata, SchedaAllenamento

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

**Tabella SchedaAllenamento:**

**SchedaAllenamento** (IdSchedaAllenamento,Peso, Stato, DataInizio, EntitàStato, Altezza, MassaGrassa, AcquaTotale, DataFine, MassaMagra, Dipendente, Cliente, Sfida);

* IdSchedaAllenamento -> Peso, Stato, DataInizio, EntitàStato, Altezza, MassaGrassa, AcquaTotale, DataFine, MassaMagra, Dipendente, Cliente, Sfida

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

**Tabella Sessione:**

**Sessione** (IdSessione, Recupero, IstanteFine, RipetizioniEffettive, IstanteInizio, Esercizio);

* IdSessione -> Recupero, IstanteFine, RipetizioniEffettive, IstanteInizio, Esercizio

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

**Tabella Accesso:**

**Accesso** (IdAccesso, Data, OrarioIngresso, OrarioUscita, Tariffa, Durata, CentroFitness, Armadietto, Cliente, Smartwatch);

* IdAccesso -> Data, OrarioIngresso, OrarioUscita, Tariffa, Durata, CentroFitness, Armadietto, Cliente, Smartwatch

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

**Tabella Personale:**

Personale (Dipendente, Giorno, OrarioIngresso, OrarioUscita);

* Dipendente, Giorno -> OrarioIngresso, OrarioUscita

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

**Tabella Oraria :**

Personale (CentroFitness, Giorno, OrarioApertura, OrarioChiusura);

* CentroFitness, Giorno -> OrarioApertura, OrarioChiusura

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

**Tabella Calendario:**

**Calendario** (Corso, Giorno, OrarioInizio, OrarioFine);

* Corso, Giorno -> OrarioInizio, OrarioFine

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

**Tabella Amicizia:**

**Amicizia** (UtenteRichiedente, UtenteRicevente, DataRichiesta, DataAccettazione);

* UtenteRichiedente, UtenteRicevente -> DataRichiesta, DataAccettazione

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

**Tabella Partecipa\_1:**

**Partecipa\_1** (Utente, Sfida, Ruolo);

* Utente, Sfida -> Ruolo

Poiché la parte sinistra è superchiave, Partecipa\_1 è in BCNF.

**Tabella Giudizio:**

**Giudizio** (PostRisposta, CodGiudizio, Modalità, Voto);

* PostRisposta, CodGiudizio -> Modalità, Voto

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

**Tabella Armadietto:**

**Armadietto** (IdArmadietto, CodiceSblocco, Spogliatoio);

* IdArmadietto -> CodiceSblocco, Spogliatoio

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

**Tabella Spogliatoio:**

**Spogliatoio** (IdSpogliatoio, Capienza, N°Posti, Posizione);

* IdSpogliatoio -> Capienza, N°Posti, Posizione

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

**Tabella Rata:**

**Rata** (IdRata, IstitutoFinanziario, Importo, DataScadenza, TassoInteresse, Pagamento);

* IdRata -> IstitutoFinanziario, Importo, DataScadenza, TassoInteresse, Pagamento

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

**Tabella Pagamento:**

**Pagamento** (IdPagamento, ImportoTotale, Stato, ImportoAttuale, Contratto);

* IdPagamento -> ImportoTotale, Stato, ImportoAttuale, Contratto

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

**Tabella Regolazione:**

**Regolazione** (ConfigurazioneUtilizzo, Inclinazione, Carico, Velocità, Tempo, Intensità, Apparecchiatura);

* ConfigurazioneUtilizzo -> Inclinazione, Carico, Velocità, Tempo, Intensità, Apparecchiatura

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

**Tabella Utente:**

**Utente** (Username, Password, Cliente);

* Username -> Password, Cliente

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

**Tabella Cerchia:**

**Cerchia** (IdCerchia, NomeCerchia, MaxPartecipanti, NumPartecipanti, Utente);

* IdCerchia -> NomeCerchia, MaxPartecipanti, NumPartecipanti, Utente

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

**Tabella Sfida:**

**Sfida** (IdSfida, Scopo, DataLancio, DataInizio, Scadenza);

* IdSfida -> Scopo, DataLancio, DataInizio, Scadenza

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

**Tabella Valutazione:**

**Valutazione** (IdValutazione, KcalBruciate, BattitoCardiaco, ConsumoOssigeno, SforzoPsichico, DataFine, Utente, Sfida);

* IdValutazione -> KcalBruciate, BattitoCardiaco, ConsumoOssigeno, SforzoPsichico, DataFine, Utente, Sfida

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

**Tabella Interesse:**

**Interesse** (IdInteresse, Tipo);

* IdInteresse -> Tipo

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

**Tabella Post:**

**Post** (IdPost, Testo, AreaForum, Timestamp, Utente);

* IdPost -> Testo, AreaForum, Timestamp, Utente

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

**Tabella PostRisposta:**

**Post** (IdPostRisposta, AreaForum, Testo, Timestamp, Utente,Post);

* IdPostRisposta -> AreaForum, Testo, Timestamp, Utente, Post

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

**Tabella Esterno:**

**Esterno** (Url, Dominio, Post);

* Url -> Dominio, Post

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

**Tabella Lavora\_In:**

**Lavora\_In** (CentroFitness, Dipendente, Ruolo);

* CentroFitness -> Dipendente, Ruolo

Poiché la parte sinistra è superchiave, Lavora\_In è in BCNF.

**Tabella Frequenta:**

**Frequenta** (CentroFitness, Cliente);

Poiché non ci sono dipendenze funzionali, Lavora\_In è in BCNF.

**Tabella Stipula:**

**Stipula** (CentroFitness, Contratto);

Poiché non ci sono dipendenze funzionali, Stipula è in BCNF.

**Tabella Accede:**

**Accede** (Contratto, Piscina);

Poiché non ci sono dipendenze funzionali, Accede è in BCNF.

**Tabella Gestisce:**

**Gestisce** (Dipendente, Sala);

Poiché non ci sono dipendenze funzionali, Gestisce è in BCNF.

**Tabella Accede\_1:**

**Accede\_1** (Contratto, Sala);

Poiché non ci sono dipendenze funzionali, Accede\_1 è in BCNF.

**Tabella Accede\_2:**

**Accede\_1** (Contratto, Corso);

Poiché non ci sono dipendenze funzionali, Accede\_2 è in BCNF.

**Tabella Partecipa:**

**Partecipa** (Cliente, Corso);

Poiché non ci sono dipendenze funzionali, Partecipa è in BCNF.

**Tabella Contenuta:**

**Contenuta** (SchedaAlimentazione, Dieta);

Poiché non ci sono dipendenze funzionali, Contenuta è in BCNF.

**Tabella Si\_Associa:**

**Si\_Associa** (Integratore, Dieta);

Poiché non ci sono dipendenze funzionali, Si\_Associa è in BCNF.

**Tabella Richiede:**

**Richiede** (Apparecchiatura, Esercizio);

Poiché non ci sono dipendenze funzionali, Richiede è in BCNF.

**Tabella Aderisce:**

**Aderisce** (Utente, Cerchia);

Poiché non ci sono dipendenze funzionali, Aderisce è in BCNF.

**Tabella Nutre:**

**Nutre** (Cerchia, Interesse);

Poiché non ci sono dipendenze funzionali, Nutre è in BCNF.

**Tabella Fa\_Uso:**

**Fa\_Uso** (Post, Sfida);

Poiché non ci sono dipendenze funzionali, Fa\_Uso è in BCNF.

**Tavola Dei Volumi**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **E/R** | **Numero**  **Istanze** | **Motivazione** |
| **CentroFitness** | **E** | **7** | **Ipotesi iniziale.** |
| **Lavora\_In** | **R** | **210** | **In media un dipendente lavora in 2 centri fitness: 105 \* 2 = 210.** |
| **Dipendente** | **E** | **105** | **Ipotiziamo una media di 15 dipendenti per ogni centro.** |
| **Frequenta** | **R** | **3150** | **In media un cliente frequenta 1.5 centri fitness: 2100 \* 1.5 = 3150.** |
| **Cliente** | **E** | **2100** | **In media ogni centro fitness ospita 300 clienti: 300 \* 7 = 2100.** |
| **Stipula** | **R** | **2940** | **In media ogni contratto consente l’accesso a 2 centri fitness: 1470 \* 2 = 2940.** |
| **E’\_Sottoscritto** | **R** | **1470** | **Cardinalità (1,1) con Contratto.** |
| **Contratto** | **E** | **1470** | **In media 7 clienti su 10 sottoscrivono un contratto: 2100 \* 0.7 = 1470.** |
| **Accede** | **R** | **1470** | **In media ogni contratto consente l’accesso ad una piscina.** |
| **E’\_Dotato** | **R** | **14** | **Cardinalità (1,1) con Piscina.** |
| **Piscina** | **E** | **14** | **In media ogni centro fitness possiede 2 piscine: 7 \* 2 = 14.** |
| **Gestisce** | **R** | **70** | **In media ogni sala è gestita da 2 dipendenti : 35 \* 2 = 70.** |
| **Si\_Colloca** | **R** | **35** | **Cardinalità (1,1) con Sala.** |
| **Accede\_1** | **R** | **4410** | **In media ogni contratto consente l’accesso a 3 sale: 1470 \* 3 = 4410.** |
| **Sala** | **E** | **35** | **In media in ogni centro ci sono 5 sale : 7 \* 5 = 35.** |
| **Accede\_2** | **R** | **22050** | **In media ogni contratto consente l’accesso a 15 corsi: 1470 \* 15 = 22050.** |
| **Partecipa** | **R** | **4200** | **In media ogni cliente partecipa a 2 corsi:**  **2100 \* 2 = 4200.** |
| **Sostiene** | **R** | **35** | **In media ogni centro fitness sostiene 5 corsi in piscina: 5 \* 7 = 35.** |
| **Si\_Tiene** | **R** | **175** | **In media ogni centro fitness tiene 25 corsi in sala: 25 \* 7 = 175.** |
| **Dirige** | **R** | **210** | **Cardinalità (1,1) con Corso.** |
| **Ha\_1** | **R** | **2100** | **Cardinalità (1,1) con Scopo.** |
| **Scopo** | **E** | **2100** | **In media ogni cliente ha 1 scopo:**  **2100 \* 1 = 2100.** |
| **Corso** | **E** | **210** | **In media ogni centro fitness organizza 30 corsi: 7 \* 30 = 210.** |
| **Gener\_** | **R** | **700** | **In media 1 cliente su 3 ha come scopo potenziamento muscolare: 2100 / 3 = 700.** |
| **Potenziamento**  **Muscolare** | **E** | **700** | **Cardinalità (1,1) con Gener\_.** |
| **E’\_Assegnato1** | **R** | **2100** | **Cardinalità (1,1) con Cliente.** |
| **Medico** | **E** | **420** | **In media ad un medico sono assegnati 5 clienti: 2100 / 5 = 420.** |
| **Rilevata** | **R** | **25200** | **Cardinalità (1,1) con Misurazione.** |
| **Prende** | **R** | **25200** | **Cardinalità (1,1) con Misurazione.** |
| **Misurazione** | **E** | **25200** | **In media ad ogni cliente vengono rilevate 12 misurazioni: 12 \* 2100 = 25200.** |
| **Stila** | **R** | **2100** | **Cardinalità (1,1) con SchedaAlimentazione.** |
| **Appartiene** | **R** | **2100** | **Cardinalità (1,1) con SchedaAlimentazione.** |
| **Scheda**  **Alimentazione** | **E** | **2100** | **In media ad ogni cliente appartiene una scheda di alimentazione.** |
| **Contenuta** | **R** | **2100** | **In media in ogni scheda di alimentazione è contenuta una dieta.** |
| **Configura** | **R** | **2100** | **Cardinalità (1,1) con Dieta.** |
| **Dieta** | **E** | **2100** | **In media in ogni scheda di alimentazione è contenuta una dieta.** |
| **Magazzino** | **E** | **7** | **Per ipotesi gli integratori giacciono in 7 magazzini.** |
| **Acquista** | **R** | **6300** | **In media ogni cliente acquista un integratore.** |
| **Si\_Associa** | **R** | **6300** | **In media ad ogni dieta vengono associati 3 integratori: 2100 \* 3 = 6300.** |
| **Giace** | **R** | **500** | **Cardinalità (1,1) con Integratore.** |
| **Integratore** | **E** | **500** | **Per ipotesi consideriamo 500 integratori.** |
| **Vende** | **R** | **3000** | **Cardinalità (1,1) con Integratore.** |
| **Ricevuto** | **R** | **84** | **Cardinalità (1,1) con Ordine.** |
| **Fornitore** | **E** | **1000** | **In media ogni integratore è venduto da 2 fornitori: 500 \* 2 = 1000.** |
| **Effettua­\_1** | **R** | **84** | **Cardinalità (1,1) con Ordine.** |
| **Ordine** | **E** | **84** | **In media ogni centro fitness effettua 12 ordini: 7 \* 12 = 84.** |
| **E’­\_Dotato1** | **R** | **1400** | **Cardinalità (1,1) con Apparecchiatura.** |
| **E’\_In** | **R** | **1400** | **Cardinalità (1,1) con Apparecchiatura.** |
| **Apparecchiatura** | **E** | **1400** | **In media ogni centro fitness possiede 200 apparecchiature: 200 \* 7 = 1400.** |
| **Richiede** | **R** | **21000** | **In media ogni esercizio richiede una apparecchiatura.** |
| **Esercizio** | **E** | **21000** | **In media ogni scheda di allenamento richiede 10 esercizi: 2100 \* 10 = 21000.** |
| **Ripetizione** | **R** | **630000** | **Cardinalità (1,1) con Sessione.** |
| **Sessione** | **E** | **630000** | **In media ogni esercizio si compone di 3 sessioni: 21000 \* 3 = 630000.** |
| **Turnazione** | **R** | **1260** | **Cardinalità (1,1) con Personale.** |
| **Personale** | **E** | **1260** | **In media ogni dipendente effettua 12 turni: 105 \*12 = 1260.** |
| **Possiede** | **R** | **2100** | **Cardinalità (1,1) con SchedaAllenamento.** |
| **E’\_Composta** | **R** | **21000** | **Cardinalità (1,1) con Esercizio.** |
| **E’\_Consegnata** | **R** | **2100** | **Cardinalità (1,1) con SchedaAllenamento.** |
| **Scheda**  **Allenamento** | **E** | **2100** | **In media ogni cliente possiede una scheda di allenamento.** |
| **Autorizza** | **R** | **420000** | **Cardinalità (1,1) con Accesso.** |
| **Effettua** | **R** | **420000** | **Cardinalità (1,1) con Accesso.** |
| **Accesso** | **E** | **420000** | **In media ogni cliente effettua 200 accessi:**  **200 \* 2100 = 420000.** |
| **Riserva** | **R** | **420000** | **Cardinalità (1,1) con Accesso.** |
| **Armadietto** | **E** | **350** | **In media ogni spogliatoio possiede 10 armadietti: 10 \* 35 = 350.** |
| **Situato** | **R** | **350** | **Cardinalità (1,1) con Armadietto.** |
| **Spogliatoio** | **E** | **35** | **In media ogni centro fitness possiede 5 spogliatoi: 7 \* 5 = 35.** |
| **Fascia** | **R** | **42** | **Cardinalità (1,1) con Oraria.** |
| **Oraria** | **E** | **42** | **Ogni centro fitness possiede 6 fasce orarie:**  **6 \* 7 = 42.** |
| **Si\_Divide** | **R** | **11760** | **Cardinalità (1,1) con Rata.** |
| **Rata** | **E** | **11760** | **In media ogni pagamento si divide in 8 rate:**  **1470 \* 8 = 11760.** |
| **Prevede\_1** | **R** | **1470** | **Cardinalità (1,1) con Contratto.** |
| **Pagamento** | **E** | **1470** | **Cardinalità (1,1) con Prevede\_1.** |
| **Segue** | **R** | **1260** | **Cardinalità (1,1) con Calendario.** |
| **Calendario** | **E** | **1260** | **In media ogni corso segue 6 calendari:**  **210 \* 6 = 1260.** |
| **Ha\_2** | **R** | **18200** | **Cardinalità (1,1) con Regolazione.** |
| **E’\_Assegnato** | **R** | **2100** | **Cardinalità (1,1) con Cliente.** |
| **Regolazione** | **E** | **18200** | **In media ogni apparecchiatura ha 13 regolazioni: 1400 \* 13 = 18200.** |
| **Utente** | **E** | **2100** | **Cardinalità (1,1) con E’\_Assegnato.** |
| **Amicizia** | **R** | **420000** | **In media ogni utente invia/riceve 200 richieste: 200 \* 2100 = 420000.** |
| **Aderisce** | **R** | **1800** | **In media ad ogni cerchia aderiscono 6 utenti:**  **6 \* 300 = 1800.** |
| **Gestisce\_1** | **R** | **300** | **Cardinalità (1,1) con Cerchia.** |
| **Cerchia** | **E** | **300** | **Per ipotesi esistono 300 cerchie.** |
| **Partecipa\_1** | **R** | **2000** | **In media ad ogni sfida partecipano 20 utenti: 20 \* 100 = 2000.** |
| **Associa** | **R** | **200** | **In media ad ogni sfida sono associate 2 schede di allenamento: 100 \* 2 = 200.** |
| **Associa\_1** | **R** | **300** | **In media ad ogni sfida sono associate 3 schede di alimentazione: 100 \* 3 = 300.** |
| **Sfida** | **E** | **100** | **Per ipotesi abbiamo 100 sfide.** |
| **Esprime** | **R** | **20000** | **Cardinalità (1,1) con Valutazione.** |
| **Prevede** | **R** | **20000** | **Cardinalità (1,1) con Valutazione.** |
| **Valutazione** | **E** | **20000** | **In media ogni utente esprime 10 valutazioni a sfida: 10 \* 2000 = 20000.** |
| **Nutre** | **R** | **600** | **In media ogni cerchia condivide 2 interessi:**  **2 \* 300 = 600.** |
| **Interesse** | **E** | **21** | **Per ipotesi esistono 21 interessi diversi.** |
| **Fa\_Uso** | **R** | **10000** | **In media per ogni sfida vengono pubblicati 100 post: 100 \* 100 = 10000.** |
| **Adopera** | **R** | **10500** | **Cardinalità (1,1) con Post.** |
| **Post** | **E** | **10500** | **In media ogni utente pubblica 50 post:**  **2100 \* 50 = 10500.** |
| **Scrive** | **R** | **525000** | **Cardinalità (1,1) con PostRisposta.** |
| **Risponde** | **R** | **525000** | **Cardinalità (1,1) con PostRisposta.** |
| **PostRisposta** | **E** | **525000** | **In media ogni utente pubblica 250 post di risposta: 250 \* 2100 = 525000.** |
| **Corrisponde** | **R** | **5250000** | **Cardinalità (1,1) con Giudizio.** |
| **Giudizio** | **E** | **5250000** | **In media ad ogni post di risposta corrispondono 10 giudizi: 525000 \* 10: 5250000.** |
| **Link** | **R** | **21000** | **Cardinalità (1,1) con Esterno.** |
| **Esterno** | **E** | **21000** | **In media ad ogni post si associano 2 link esterni: 10500 \* 2 = 21000.** |

**Elenco delle ridondanze**

1. **Accesso ->** *Durata*

L’attributo *Durata* della tabella **Accesso** è una ridondanza, in quanto codesta informazione è ottenibile tramite la differenza tra gli attributi *OrarioUscita* e *OrarioIngresso* della tabella **Accesso**.

1. **Pagamento ->** *ImportoAttuale*

L’attributo *ImportoAttuale* della tabella **Pagamento** è una ridondanza, poichè codesta informazione è ottenibile tramite la somma dei singoli importi,registrati attraverso l’attributo *Importo*,della tabella **Rata**.

1. **Cerchia ->** *NumPartecipanti*

L’Attributo *NumPartecipanti* della tabella **Cerchia** è una ridondanza, in quanto codesta informazione si può ottenere contando il numero delle occorenze relative alle associazioni Aderisce e Gestisce.

**Operazioni**

**Operazione 1**

**Descrizione:** Indicare la durata media degli accessi ad un centro in una determinata data.

**Frequenza:** 5 volte al giorno.

**Operazione 2**

**Descrizione:** Per un dato cliente mostrare l’importo non ancora versato.

**Frequenza:** 6 volte al mese.

**Operazione 3**

**Descrizione:** Quanti partecipanti possono ancora aderire ad una cerchia.

**Frequenza:** 1 volta al giorno.

**Operazione 4**

**Descrizione:** Data una sostanza indicare l’integratore più acquistato, contenente la sostanza stessa.

**Frequenza:** 1 volta al mese.

**Operazione 5**

**Descrizione:** Un utente esce da un centro.

**Frequenza:** 1 volte al giorno.

**Operazione 6**

**Descrizione:** Un cliente paga una rata.

**Frequenza:** 40 volte al mese.

**Operazione 7**

**Descrizione:** Un utente aderisce ad una cerchia.

**Frequenza:** 5 volte al giorno.

**Operazione 8**

**Descrizione:** Un utente valuta una sfida.

**Frequenza:** 105 volte al giorno.

|  |  |
| --- | --- |
| **Nome** | **Numero**  **Istanze** |
| **CentroFitness** | **7** |
| **Autorizza** | **420000** |
| **Accesso** | **420000** |

**Tavola degli accessi**

**Operazione 1**

**Descrizione:** Indicare la durata media degli accessi ad un centro in una determinata data.

**Input:** *IdCentro, Data.*

**Output:** Durata media accessi*.*

**Frequenza:** 5 volte al giorno**.**

**Ridondanza: Accesso** ->*Durata*.



|  |  |  |  |
| --- | --- | --- | --- |
| **Numero Operazioni Elementari** | **Tipo** | **Costrutto** | **Motivazione** |
| 60000 | L | R | Si eseguono 60000 operazioni di lettura su Autorizzaper trovare l’*IdCentro* relativo all’accesso e la data interessata. |
| 1 | L | E | Si eseguono un’operazione di lettura su **Accesso** per trovare la differenza tra *OrarioUscita* ed *OrarioIngresso*. |
| 60001 | Operazioni elementari per singola operazione. | | |
| 60001 \* 5 = 300005 | Totale operazioni elementari al giorno. | | |

|  |  |
| --- | --- |
| **Nome** | **Numero**  **Istanze** |
| **CentroFitness** | **7** |
| **Autorizza** | **420000** |
| **Accesso** | **420000** |

La tavola degli accessi di questa

operazione cambia a seconda

che sia presente o meno la ridondanza

*Durata* presente nella tabella **Accesso.**



|  |  |  |  |
| --- | --- | --- | --- |
| **Numero Operazioni Elementari** | **Tipo** | **Costrutto** | **Motivazione** |
| 60000 | L | R | Si esegue un’operazione di lettura su Autorizzaper trovare l’*IdCentro* relativo all’accesso, la data interessata e la relativa durata. |
| 60000 | Operazioni elementari per singola operazione. | | |
| 60000 \* 5 = 300000 | Totale operazioni elementari al giorno. | | |

|  |  |
| --- | --- |
| **Nome** | **Numero**  **Istanze** |
| **Cliente** | **2100** |
| **E’\_Sottoscritto** | **1470** |
| **Contratto** | **1470** |
| **Prevede\_1** | **1470** |
| **Pagamento** | **1470** |
| **Si\_Divide** | **11760** |
| **Rata** | **11760** |

**Operazione 2**

**Descrizione:** Per un dato cliente mostrare l’importo non ancora versato.

**Input:** *CodFiscale.*

**Output:** Importo da versare*.*

**Frequenza:** 6 volte al mese**.**

**Ridondanza: Pagamento ->** *ImportoAttuale.*



|  |  |  |  |
| --- | --- | --- | --- |
| **Numero Operazioni Elementari** | **Tipo** | **Costrutto** | **Motivazione** |
| 1 | L | R | Si esegue un’operazione di lettura su E’Sottoscrittoper trovare il *CodFiscale* relativo al cliente. |
| 1 | L | R | Si esegue un’operazione di lettura su Prevede\_1 per trovare l’*IdContratto* relativo al pagamento |
| 1 | L | E | Si esegue un’operazione di lettura su **Pagamento** per trovare l’*ImportoTotale* del cliente interessato. |
| 4 | L | R | Si eseguono quattro operazioni di lettura su Si\_Divide per trovare le rate interessate. |
| 4 | L | E | Si eseguono 4 operazioni di lettura su **Rata** per trovare la somma già pagata dal cliente interessato. |
| 11 | Operazioni elementari per singola operazione. | | |
| 11 \* 6 = 66 | Totale operazioni elementari al mese. | | |

|  |  |
| --- | --- |
| **Nome** | **Numero**  **Istanze** |
| **Cliente** | **2100** |
| **E’\_Sottoscritto** | **1470** |
| **Contratto** | **1470** |
| **Prevede\_1** | **1470** |
| **Pagamento** | **1470** |

La tavola degli accessi di questa

operazione cambia a seconda

che sia presente o meno la ridondanza

*ImportoAttuale* presente nella tabella **Pagamento.**



|  |  |  |  |
| --- | --- | --- | --- |
| **Numero Operazioni Elementari** | **Tipo** | **Costrutto** | **Motivazione** |
| 1 | L | R | Si esegue un’operazione di lettura su E’Sottoscrittoper trovare il *CodFiscale* relativo al cliente. |
| 1 | L | R | Si esegue un’operazione di lettura su Prevede\_1 per trovare l’*IdContratto* relativo al pagamento |
| 1 | L | E | Si esegue un’operazione di lettura su **Pagamento** per trovare l’importo non ancora versato dal cliente interessato. |
| 3 | Operazioni elementari per singola operazione. | | |
| 3 \* 6 = 18 | Totale operazioni elementari al mese. | | |

**Operazione 3**

|  |  |
| --- | --- |
| **Nome** | **Numero**  **Istanze** |
| **Utente** | **2100** |
| **Aderisce** | **1800** |
| **Gestisce** | **300** |
| **Cerchia** | **300** |

**Descrizione:** Quanti partecipanti possono ancora aderire ad una cerchia.

**Input:** *IdCerchia.*

**Output:** Iscrizioni possibili.

**Frequenza:** 1 volta al giorno.

**Ridondanza: Cerchia ->** *NumPartecipanti.*



|  |  |  |  |
| --- | --- | --- | --- |
| **Numero Operazioni Elementari** | **Tipo** | **Costrutto** | **Motivazione** |
| 1 | L | E | Si esegue un’operazione di lettura in **Cerchia** per leggere il massimo numero di partecipanti ad una cerchia. |
| 300 | L | R | Si eseguono 300 operazioni di lettura su Aderisceper trovare il numero attuale di partecipanti alla cerchia. |
| 301 | Operazioni elementari per singola operazione. | | |
| 301 \* 1 = 301 | Totale operazioni elementari al giorno. | | |

|  |  |
| --- | --- |
| **Nome** | **Numero**  **Istanze** |
| **Cerchia** | **300** |

La tavola degli accessi di questa

operazione cambia a seconda

che sia presente o meno la ridondanza

*NumPartecipanti* presente nella tabella **Cerchia.**



|  |  |  |  |
| --- | --- | --- | --- |
| **Numero Operazioni Elementari** | **Tipo** | **Costrutto** | **Motivazione** |
| 1 | L | E | Si esegue un’operazione di lettura in **Cerchia** per leggere il massimo numero di partecipanti ad una cerchia ed il numero attuale. |
| 1 | Operazioni elementari per singola operazione. | | |
| 1 \* 1 = 1 | Totale operazioni elementari al giorno. | | |

**Operazione 4**

**Descrizione:** Data una sostanza indicare l’integratore più acquistato, contenente la sostanza stessa.

**Input:** *NomeSostanza.*

**Output:** Integratore più acquistato.

**Frequenza:** 1 volta al mese.

**Ridondanza:** L’operazione non coinvolge ridondanze.

|  |  |
| --- | --- |
| **Nome** | **Numero**  **Istanze** |
| **Acquista** | **6300** |
| **Integratore** | **500** |



|  |  |  |  |
| --- | --- | --- | --- |
| **Numero Operazioni Elementari** | **Tipo** | **Costrutto** | **Motivazione** |
| 500 | L | E | Si eseguono cinquecento operazioni di lettura in **Integratore** per leggere il *NomeSostanza.* |
| \*630 | L | R | Si eseguono 630 operazioni di lettura su Acquista per contare il numero di occorrenze e leggere *Quantità* in ogni acquisto. |
| 1130 | Operazioni elementari per singola operazione. | | |
| 1130 \* 1 = 1130 | Totale operazioni elementari al mese. | | |

\*Abbiamo ipotizzato per la lettura delle 630 operazioni elementari una quantità pari a 10 sostanze diverse che possono costituire gli integratori.

|  |  |
| --- | --- |
| **Nome** | **Numero**  **Istanze** |
| **Cliente** | **2100** |
| **Effettua** | **6300** |
| **Accesso** | **3000** |

**Operazione 5**

**Descrizione:** Un cliente esce da un centro.

**Input:** *IdAccesso, OrarioUscita.*

**Frequenza:** 1 volte al giorno.

**Ridondanza: Accesso ->** *Durata.*



|  |  |  |  |
| --- | --- | --- | --- |
| **Numero Operazioni Elementari** | **Tipo** | **Costrutto** | **Motivazione** |
| 1 | S | E | Si eseguono cinque operazioni di scrittura su **Accesso** per inserire i dati relativi ad esso. |
| 1 | Operazioni elementari per singola operazione. | | |
| 1 \* 1 = 1 | Totale operazioni elementari al giorno. | | |

|  |  |
| --- | --- |
| **Nome** | **Numero**  **Istanze** |
| **Cliente** | **2100** |
| **Effettua** | **6300** |
| **Accesso** | **3000** |

La tavola degli accessi di questa

operazione cambia a seconda

che sia presente o meno la ridondanza

*Durata* presente nella tabella

**Accesso.**



|  |  |  |  |
| --- | --- | --- | --- |
| **Numero Operazioni Elementari** | **Tipo** | **Costrutto** | **Motivazione** |
| 1 | S | R | Si esegue un’operazioni di lettura in Autorizzaper inserire *Durata*. |
| 1 | S | E | Si esegue un’operazione di scrittura su **Accesso** per inserire i dati relativi ad esso. |
| 2 | Operazioni elementari per singola operazione. | | |
| 2 \* 1 = 2 | Totale operazioni elementari al giorno. | | |

**Operazione 6**

|  |  |
| --- | --- |
| **Nome** | **Numero**  **Istanze** |
| **Cliente** | **2100** |
| **E’\_Sottoscritto** | **1470** |
| **Contratto** | **1470** |
| **Prevede\_1** | **1470** |
| **Pagamento** | **1470** |
| **Si\_Divide** | **11760** |
| **Rata** | **11760** |

**Descrizione:** Un cliente paga una rata.

**Input:** *CodFiscale*

**Frequenza:** 50 volte al mese**.**

**Ridondanza: Pagamento ->** *ImportoAttuale.*



|  |  |  |  |
| --- | --- | --- | --- |
| **Numero Operazioni Elementari** | **Tipo** | **Costrutto** | **Motivazione** |
| 1 | L | R | Si esegue un’operazione di lettura su E’Sottoscrittoper trovare il *CodFiscale* relativo al cliente. |
| 1 | L | R | Si esegue un’operazione di lettura su Prevede\_1 per trovare l’*IdContratto* relativo al pagamento |
| 1 | S | R | Si esegue un’operazione di scrittura per inserire l’associazione tra **Rata** e **Pagamento**. |
| 1 | S | E | Si esegue un’operazione di scrittura su **Rata** per inserirne i dati relativi. |
| 4 | Operazioni elementari per singola operazione. | | |
| 4 \* 40 = 160 | Totale operazioni elementari al mese. | | |

La tavola degli accessi di questa

|  |  |
| --- | --- |
| **Nome** | **Numero**  **Istanze** |
| **Cliente** | **2100** |
| **E’\_Sottoscritto** | **1470** |
| **Contratto** | **1470** |
| **Prevede\_1** | **1470** |
| **Pagamento** | **1470** |
| **Si\_Divide** | **11760** |
| **Rata** | **11760** |

operazione cambia a seconda

che sia presente o meno la ridondanza

*ImportoAttuale* presente nella tabella **Pagamento.**



|  |  |  |  |
| --- | --- | --- | --- |
| **Numero Operazioni Elementari** | **Tipo** | **Costrutto** | **Motivazione** |
| 1 | L | R | Si esegue un’operazione di lettura su E’Sottoscrittoper trovare il *CodFiscale* relativo al cliente. |
| 1 | S | R | Si esegue un’operazione di scrittura per inserire l’associazione tra **Rata** e **Pagamento**. |
| 1 | L | R | Si esegue un’operazione di lettura su Prevede\_1 per trovare l’*IdContratto* relativo al pagamento |
| 1 | S | E | Si esegue un’operazione di scrittura nella tabella **Pagamento** per aggiornare l’attributo *ImportoAttuale*. |
| 1 | S | E | Si esegue un’operazione di scrittura su **Rata** per inserirne i dati relativi. |
| 5 | Operazioni elementari per singola operazione. | | |
| 5 \* 40 = 200 | Totale operazioni elementari al mese. | | |

**Operazione 7**

|  |  |
| --- | --- |
| **Nome** | **Numero**  **Istanze** |
| **Utente** | **2100** |
| **Aderisce** | **1800** |
| **Cerchia** | **300** |

**Descrizione:** Un utente aderisce ad una cerchia.

**Input:** *Username.*

**Frequenza:** 5 volte al giorno.

**Ridondanza: Cerchia ->** *NumPartecipanti.*



|  |  |  |  |
| --- | --- | --- | --- |
| **Numero Operazioni Elementari** | **Tipo** | **Costrutto** | **Motivazione** |
| 1 | S | R | Si esegue un’operazione di scrittura per inserire l’associazione tra **Utente** e **Cerchia**. |
| 1 | Operazioni elementari per singola operazione. | | |
| 1 \* 5 = 5 | Totale operazioni elementari al giorno. | | |

|  |  |
| --- | --- |
| **Nome** | **Numero**  **Istanze** |
| **Utente** | **2100** |
| **Aderisce** | **1800** |
| **Cerchia** | **300** |

La tavola degli accessi di questa

operazione cambia a seconda

che sia presente o meno la ridondanza

*NumPartecipanti* presente

nella tabella **Cerchia.**



|  |  |  |  |
| --- | --- | --- | --- |
| **Numero Operazioni Elementari** | **Tipo** | **Costrutto** | **Motivazione** |
| 1 | S | R | Si esegue un’operazione di scrittura per inserire l’associazione tra **Utente** e **Cerchia**. |
| 1 | S | E | Si esegue un’operazione di scrittura nella tabella **Cerchia** per aggiornare l’attributo *NumPartecipanti****.*** |
| 2 | Operazioni elementari per singola operazione. | | |
| 2 \* 5 = 10 | Totale operazioni elementari al giorno. | | |

**Operazione 8**

|  |  |
| --- | --- |
| **Nome** | **Numero**  **Istanze** |
| **Utente** | **2100** |
| **Esprime** | **20000** |
| **Valutazione** | **20000** |

**Descrizione:** Un utente valuta una sfida.

**Input:** *Username.*

**Frequenza:** 105 volte al giorno.

**Ridondanza:** L’operazione non coinvolge ridondanze.



|  |  |  |  |
| --- | --- | --- | --- |
| **Numero Operazioni Elementari** | **Tipo** | **Costrutto** | **Motivazione** |
| 1 | S | R | Si esegue un’operazione di scrittura per inserire l’associazione tra **Utente** e **Valutazione**. |
| 1 | S | E | Si esegue un’operazione di scrittura su **Valutazione** per inserirne i dati relativi. |
| 2 | Operazioni elementari per singola operazione. | | |
| 2 \* 105 = 210 | Totale operazioni elementari al giorno. | | |

ANALISI COSTO DELLE RIDONDANZE

1. **Accesso ->** *Durata*

|  |  |  |
| --- | --- | --- |
| Operazione | Numero operazioni elementari al giorno | |
|  | Senza Ridondanza | Con Ridondanza |
| Operazione 1 | 300005 | 300000 |
| Operazione 5 | 1 | 2 |
| Totale | 300006 | 300002 |

A seguito dell’analisi del costo della ridondanza si evince una riduzione dei tempi di esecuzione delle operazioni. Pertanto si decide di inserire tale ridondanza.

1. **Pagamento ->** *ImportoAttuale*

|  |  |  |
| --- | --- | --- |
| Operazione | Numero operazioni elementari al giorno | |
|  | Senza Ridondanza | Con Ridondanza |
| Operazione 2 | 66 | 18 |
| Operazione 6 | 160 | 200 |
| Totale | 226 | 218 |

A seguito dell’analisi del costo della ridondanza si evince una riduzione dei tempi di esecuzione delle operazioni. Pertanto si decide di inserire tale ridondanza.

1. **Cerchia ->** *NumPartecipanti*

|  |  |  |
| --- | --- | --- |
| Operazione | Numero operazioni elementari al giorno | |
|  | Senza Ridondanza | Con Ridondanza |
| Operazione 3 | 301 | 1 |
| Operazione 7 | 5 | 10 |
| Totale | 306 | 11 |

A seguito dell’analisi del costo della ridondanza si evince una riduzione dei tempi di esecuzione delle operazioni. Pertanto si decide di inserire tale ridondanza.

**Area Analytics**

**Reporting**

La prima funzione di reporting consiste nello stabilire il numero di fasce orarie del centro,in base al maggiore o minore numero di clienti che accedono al centro.Il numero di accessi durante la giornata dei vari clienti stabilisce come il centro possa suddividersi in 3 fasce:

* Mattina: fascia collocata tra le ore 09:00 e le ore 12:59.
* Pranzo: fascia collocata tra le ore 13:00 e le ore 14:59.
* Pomeriggio: fascia collocata tra le ore 15:00 e le ore 20:00.

Tramite l’uso di un “trigger” si è in grado di stabilire il numero dei totali accessi effettuati dai clienti nei vari centri. Questo numero è indicativo per i gestori dei centri fitness, in quanto permette loro una suddivisione di essi in precise fasce orarie, ottenute in base all’affluenza dei clienti in determinati momenti della giornata rispetto ad altri. Questi dati permettono poi una migliore distribuzione dei corsi,in modo che vengano accontentate,il più possibile,le richieste e le esigenze dei vari clienti, come richiesto dalla seconda funzione di reporting.

In fine, l’ultima richiesta di reporting, riguardante il monitoraggio dell’utilizzo delle attrezzature per capire se i vari centri sono adeguatamente dimensionati, viene ovviata tramite l’implementazione di una procedura. Questa, tramite la consultazione di una tabella appositamente creata, restituisce le apparecchiature di cui un centro necessita: inserendo il codice del centro di riferimento la procedura esamina i tempi medi di attesa di ogni apparecchiatura all’interno di esso. Qualora questi dovessero superare i 10 minuti, l’attrezzo in questione viene stampato a video con conseguente necessità di acquistarne altri.

**Performance Sportiva**

La funzione riguardante la performance sportiva di un cliente permette di ricalibrare le schede di allenamento in base ai carichi di lavoro a cui vengono sottoposti i clienti. La soluzione proposta consiste nella creazione di una tabella in cui vengono riportati il tempo impiegato tra gli esercizi privi di ripetizioni e quelli che invece le prevedono. Si intende ovviare al problema tramite una procedura, che prevede il calcolo delle varie ripetizioni e dei tempi impiegati: tale calcolo permette di apportare modifiche utili a migliorare le attività dei clienti all’interno dei vari centri. Se la media delle differenze tra le ripetizioni degli esercizi assegnati e degli esercizi svolti è uguale o superiore a 3 o la media delle differenze tra la durata prevista e quella effettiva è uguale o superiore a 15 minuti, allora la scheda di allenamento risulta essere troppo pesante. Nel caso in cui la media delle differenze tra le ripetizioni degli esercizi assegnati e degli esercizi svolti è uguale a 0 o la media delle differenze tra la durata prevista e quella effettiva è uguale a 0, allora la scheda risulta essere troppo leggera. In tutti i restanti casi la scheda risulta essere corretta e il cliente può continuare con gli esercizi predisposti dal tutor nella sua scheda personale.

**Rotazione del magazzino**

La funzione di rotazione del magazzino permette di gestire gli integratori presenti in esso. Gli integratori sono gestiti in modo che sia possibile conoscere la tipologia,la quantità e il loro nome commerciale. Il database in questione si occupa di gestire il magazzino in modo che sia reperibile la quantità giacente e quella venduta al cliente, che ha dunque lasciato il centro fitness, in seguito ad accordi presi tra il centro stesso, i clienti e i fornitori presso i quali vengono acquistati i suddetti. Stabiliti dei criteri di acquisto e delle percentuali di sconto da usare in base alla data di scadenza, possono essere gestiti gli acquisti degli integratori, mediante l’implementazione di un event, che settimanalmente aggiorna la percentuale di sconto e il prezzo dei suddetti. Le specifiche di sconto vengono scelte arbitrariamente e cambiano in base all’avvicinarsi della data di scadenza degli integratori. Si può ottenere uno sconto pari al 10% per gli integratori la cui data di scadenza è prossima ad un mese, 15% per quelli la cui data di scadenza è prossima a 14 giorni e pari a 20% per quelli la cui data è prossima a 7 giorni.

**SCRIPT SQL**

**Creazione:**

DROP DATABASE IF EXISTS progetto;

CREATE DATABASE progetto;

USE progetto;

-- ------------------------------------

-- Tabella CentroFitness

-- ------------------------------------

DROP TABLE IF EXISTS CentroFitness;

CREATE TABLE CentroFitness (

IdCentro INT NOT NULL AUTO\_INCREMENT,

CAP INT(5) NOT NULL,

Dimensione INT(5) NOT NULL,

NumCivico VARCHAR(5) NOT NULL,

Citta VARCHAR(50) NOT NULL,

NTelefono VARCHAR(11) NOT NULL,

MaxClienti INT(4) NOT NULL,

Via VARCHAR(50) NOT NULL,

PRIMARY KEY(IdCentro)

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Dipendente

-- ------------------------------------

DROP TABLE IF EXISTS Dipendente;

CREATE TABLE Dipendente (

CodFiscale VARCHAR(16) NOT NULL,

Nome VARCHAR(50) NOT NULL,

Cognome VARCHAR(50) NOT NULL,

CAP INT(5) NOT NULL,

DataNascita DATE NOT NULL,

Via VARCHAR(50) NOT NULL,

NumCivico VARCHAR(5) NOT NULL,

Citta VARCHAR(50) NOT NULL,

NPrefettura VARCHAR(20) NOT NULL,

CodDocumento VARCHAR(10) NOT NULL,

PRIMARY KEY(CodFiscale)

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Medico

-- ------------------------------------

DROP TABLE IF EXISTS Medico;

CREATE TABLE Medico (

CodMedico VARCHAR(5) NOT NULL,

Nome VARCHAR(50) NOT NULL,

Cognome VARCHAR(50) NOT NULL,

Telefono VARCHAR(11) NOT NULL,

Studio VARCHAR(50) NOT NULL,

PRIMARY KEY(CodMedico)

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Cliente

-- ------------------------------------

DROP TABLE IF EXISTS Cliente;

CREATE TABLE Cliente (

CodFiscale VARCHAR(16) NOT NULL,

Nome VARCHAR(50) NOT NULL,

Cognome VARCHAR(50) NOT NULL,

Citta VARCHAR(50) NOT NULL,

NumCivico VARCHAR(5) NOT NULL,

CAP INT(5) NOT NULL,

NPrefettura VARCHAR(20) NOT NULL,

CodDocumento VARCHAR(10) NOT NULL,

DataNascita DATE NOT NULL,

Via VARCHAR(50) NOT NULL,

PreferenzeAlimentari VARCHAR(100) NOT NULL,

Medico VARCHAR(5) NOT NULL,

PRIMARY KEY(CodFiscale),

FOREIGN KEY(Medico) REFERENCES Medico(CodMedico) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Contratto

-- ------------------------------------

DROP TABLE IF EXISTS Contratto;

CREATE TABLE Contratto (

IdContratto VARCHAR(5) NOT NULL,

Durata VARCHAR(20) NOT NULL,

Costo INT(3) NOT NULL,

Variante VARCHAR(20) NOT NULL,

MaxIngressi INT(4) NOT NULL,

MaxIngressiPiscina INT(4) NOT NULL,

CodConsulente VARCHAR(20) NOT NULL,

DataSottoscrizione DATE,

Cliente VARCHAR(16) NOT NULL,

CONSTRAINT CHK\_Contratto CHECK(Variante = 'Gold' OR Variante = 'Silver'

OR Variante = 'Platinum'),

PRIMARY KEY(IdContratto),

FOREIGN KEY(Cliente) REFERENCES Cliente(CodFiscale) ON DELETE CASCADE

ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Piscina

-- ------------------------------------

DROP TABLE IF EXISTS Piscina;

CREATE TABLE Piscina (

IdPiscina VARCHAR(5) NOT NULL,

Dimensione VARCHAR(6) NOT NULL,

Tipo VARCHAR(20) NOT NULL,

CentroFitness INT,

CONSTRAINT CHK\_Piscina CHECK(Tipo = 'Interna' OR Tipo = 'Esterna'),

PRIMARY KEY(IdPiscina),

FOREIGN KEY(CentroFitness) REFERENCES CentroFitness(IdCentro) ON DELETE CASCADE

ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Sala

-- ------------------------------------

DROP TABLE IF EXISTS Sala;

CREATE TABLE Sala (

IdSala VARCHAR(5) NOT NULL,

NomeSala VARCHAR(50) NOT NULL,

CentroFitness INT,

PRIMARY KEY(IdSala),

FOREIGN KEY(CentroFitness) REFERENCES CentroFitness(IdCentro) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Corso

-- ------------------------------------

DROP TABLE IF EXISTS Corso;

CREATE TABLE Corso (

IdCorso VARCHAR(10) NOT NULL,

MaxPartecipanti INT(5) NOT NULL,

NomeDisciplina VARCHAR(50) NOT NULL,

Livello VARCHAR(20) NOT NULL,

DataInizio DATE,

DataFine DATE,

Sala VARCHAR(5),

Piscina VARCHAR(5),

Dipendente VARCHAR(16),

CONSTRAINT CHK\_Corso CHECK(Livello = 'Principiante' OR Livello = 'Intermedio' OR Livello = 'Avanzato'),

PRIMARY KEY(IdCorso),

FOREIGN KEY(Dipendente) REFERENCES Dipendente(CodFiscale) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(Sala) REFERENCES Sala(IdSala) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(Piscina) REFERENCES Piscina(IdPiscina) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Scopo

-- ------------------------------------

DROP TABLE IF EXISTS Scopo;

CREATE TABLE Scopo (

IdScopo VARCHAR(10) NOT NULL,

Tipo VARCHAR(50) NOT NULL,

Cliente VARCHAR(16) NOT NULL,

CONSTRAINT CHK\_Scopo CHECK(Tipo = 'Potenziamento Muscolare' OR Tipo = 'Dimagrimento' OR Tipo = 'Scopo Ricreativo'),

PRIMARY KEY(IdScopo),

FOREIGN KEY(Cliente) REFERENCES Cliente(CodFiscale) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella PotenziamentoMuscolare

-- ------------------------------------

DROP TABLE IF EXISTS PotenziamentoMuscolare;

CREATE TABLE PotenziamentoMuscolare (

IdScopo VARCHAR(10) NOT NULL,

Muscolo VARCHAR(50) NOT NULL,

Livello VARCHAR(10) NOT NULL,

CONSTRAINT CHK\_PotenziamentoMuscolare CHECK(Livello = 'Lieve' OR Livello = 'Moderato' OR Livello = 'Elevato'),

PRIMARY KEY(IdScopo),

FOREIGN KEY(IdScopo) REFERENCES Scopo(IdScopo) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Misurazione

-- ------------------------------------

DROP TABLE IF EXISTS Misurazione;

CREATE TABLE Misurazione (

IdMisurazione VARCHAR(5) NOT NULL,

AcquaTotale INT(5) NOT NULL,

Data DATE,

MassaGrassa INT(5) NOT NULL,

MassaMagra INT(5) NOT NULL,

Peso INT(3) NOT NULL,

DataVisita DATE,

Cliente VARCHAR(16) NOT NULL,

Medico VARCHAR(5) NOT NULL,

PRIMARY KEY(IdMisurazione),

FOREIGN KEY(Cliente) REFERENCES Cliente(CodFiscale) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(Medico) REFERENCES Medico(CodMedico) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Sfida

-- ------------------------------------

DROP TABLE IF EXISTS Sfida;

CREATE TABLE Sfida (

IdSfida VARCHAR(5) NOT NULL,

Scopo VARCHAR(50) NOT NULL,

DataLancio DATE,

DataInizio DATE,

Scadenza DATE,

PRIMARY KEY(IdSfida)

)ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella SchedaAlimentazione

-- ------------------------------------

DROP TABLE IF EXISTS SchedaAlimentazione;

CREATE TABLE SchedaAlimentazione (

IdSchedaAlimentazione VARCHAR(5) NOT NULL,

Obiettivo VARCHAR(50) NOT NULL,

DataInizio DATE,

DataFine DATE,

Intervallo VARCHAR(15) NOT NULL,

Cliente VARCHAR(16),

Medico VARCHAR(5),

Sfida VARCHAR(5),

CONSTRAINT CHK\_SchedaAlimentazione CHECK(Obiettivo = 'Perdere Peso' OR Obiettivo = 'Acquisire Massa Muscolare' OR Obiettivo = 'Prendere Peso' OR Obiettivo = 'Diminuire Massa Grassa'),

PRIMARY KEY(IdSchedaAlimentazione),

FOREIGN KEY(Cliente) REFERENCES Cliente(CodFiscale) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(Medico) REFERENCES Medico(CodMedico) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(Sfida) REFERENCES Sfida(IdSfida) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Dieta

-- ------------------------------------

DROP TABLE IF EXISTS Dieta;

CREATE TABLE Dieta (

IdDieta VARCHAR(5) NOT NULL,

MaxKilocal INT(5) NOT NULL,

MinKilocal INT(5) NOT NULL,

NPasti INT(1) NOT NULL,

Pasto VARCHAR(100) NOT NULL,

Medico VARCHAR(5) NOT NULL,

PRIMARY KEY(IdDieta),

FOREIGN KEY(Medico) REFERENCES Medico(CodMedico) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Magazzino

-- ------------------------------------

DROP TABLE IF EXISTS Magazzino;

CREATE TABLE Magazzino (

IdMagazzino VARCHAR(5) NOT NULL,

Giacenza INT(5) NOT NULL,

PRIMARY KEY(IdMagazzino)

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Fornitore

-- ------------------------------------

DROP TABLE IF EXISTS Fornitore;

CREATE TABLE Fornitore (

IdFornitore VARCHAR(5) NOT NULL,

PartitaIva VARCHAR(15) NOT NULL,

NTelefono INT(10) NOT NULL,

CAP INT(5) NOT NULL,

FormaSocietaria VARCHAR(10) NOT NULL,

Citta VARCHAR(50) NOT NULL,

NumCivico VARCHAR(5) NOT NULL,

Via VARCHAR(50) NOT NULL,

Nome VARCHAR(50) NOT NULL,

PRIMARY KEY(IdFornitore)

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Integratore

-- ------------------------------------

DROP TABLE IF EXISTS Integratore;

CREATE TABLE Integratore (

IdIntegratore VARCHAR(6) NOT NULL,

DataScadenza DATE,

Forma VARCHAR(20) NOT NULL,

NomeCommerciale VARCHAR(50) NOT NULL,

NomeSostanza VARCHAR(50) NOT NULL,

QuantitaSostanza INT(5) NOT NULL,

NPezzi INT(5) NOT NULL,

Magazzino VARCHAR(5) NOT NULL,

Fornitore VARCHAR(5) NOT NULL,

PRIMARY KEY(IdIntegratore),

FOREIGN KEY(Magazzino) REFERENCES Magazzino(IdMagazzino) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(Fornitore) REFERENCES Fornitore(IdFornitore) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Ordine

-- ------------------------------------

DROP TABLE IF EXISTS Ordine;

CREATE TABLE Ordine (

CodiceInterno VARCHAR(7) NOT NULL,

CodiceEsterno VARCHAR(6) NOT NULL,

DataConsegna DATE,

DataEvasione DATE,

Stato VARCHAR(20) NOT NULL,

Fornitore VARCHAR(5) NOT NULL,

CentroFitness INT,

CONSTRAINT CHK\_Ordine CHECK(Stato = 'Incompleto' OR Stato = 'Evaso'),

PRIMARY KEY(CodiceInterno),

FOREIGN KEY(Fornitore) REFERENCES Fornitore(IdFornitore) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(CentroFitness) REFERENCES CentroFitness(IdCentro) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Apparecchiatura

-- ------------------------------------

DROP TABLE IF EXISTS Apparecchiatura;

CREATE TABLE Apparecchiatura (

IdApparecchiatura VARCHAR(5) NOT NULL,

Nome VARCHAR(50) NOT NULL,

Tipo VARCHAR(50) NOT NULL,

Consumo INT(5) NOT NULL,

Manutenzione VARCHAR(2) NOT NULL,

PercentualeUsura INT(3) NOT NULL,

Sala VARCHAR(5) NOT NULL,

CentroFitness INT NOT NULL,

CONSTRAINT CHK\_Apparecchiatura CHECK((PercentualeUsura >= 0 AND PercentualeUsura <=100) AND (Manutenzione = 'SI' OR Manutenzione = 'NO')),

PRIMARY KEY(IdApparecchiatura),

FOREIGN KEY(Sala) REFERENCES Sala(IdSala) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(CentroFitness) REFERENCES CentroFitness(IdCentro) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella SchedaAllenamento

-- ------------------------------------

DROP TABLE IF EXISTS SchedaAllenamento;

CREATE TABLE SchedaAllenamento (

IdSchedaAllenamento VARCHAR(5) NOT NULL,

Peso INT(3) NOT NULL,

Stato VARCHAR(20) NOT NULL,

DataInizio DATE,

EntitaStato VARCHAR(50),

Altezza INT(5) NOT NULL,

MassaGrassa INT(5) NOT NULL,

AcquaTotale INT(5) NOT NULL,

DataFine DATE,

MassaMagra INT(5) NOT NULL,

Dipendente VARCHAR(16),

Cliente VARCHAR(16),

Sfida VARCHAR(5),

CONSTRAINT CHK\_SchedaAllenamento CHECK(Stato = 'Sovrappeso' OR Stato = 'Normopeso' OR Stato = 'Sottopeso'),

PRIMARY KEY(IdSchedaAllenamento),

FOREIGN KEY(Dipendente) REFERENCES Dipendente(CodFiscale) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(Cliente) REFERENCES Cliente(CodFiscale) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(Sfida) REFERENCES Sfida(IdSfida) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Esercizio

-- ------------------------------------

DROP TABLE IF EXISTS Esercizio;

CREATE TABLE Esercizio (

IdEsercizio VARCHAR(5) NOT NULL,

Nome VARCHAR(50) NOT NULL,

DispendioEnergetico INT(3) NOT NULL,

Peso INT(3),

NRipetizioni INT(3),

Tipo VARCHAR(50) NOT NULL,

IstanteInizio TIME,

Durata INT(3),

SchedaAllenamento VARCHAR(5) NOT NULL,

PRIMARY KEY(IdEsercizio),

FOREIGN KEY(SchedaAllenamento) REFERENCES SchedaAllenamento(IdSchedaAllenamento) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Sessione

-- ------------------------------------

DROP TABLE IF EXISTS Sessione;

CREATE TABLE Sessione (

IdSessione VARCHAR(5) NOT NULL,

Recupero INT(5) NOT NULL,

IstanteInizio TIME,

IstanteFine TIME,

RipetizioneEffettive INT(3),

Esercizio VARCHAR(5) NOT NULL,

PRIMARY KEY(IdSessione),

FOREIGN KEY(Esercizio) REFERENCES Esercizio(IdEsercizio) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Spogliatoio

-- ------------------------------------

DROP TABLE IF EXISTS Spogliatoio;

CREATE TABLE Spogliatoio (

IdSpogliatoio VARCHAR(5) NOT NULL,

Capienza INT(3) NOT NULL,

NPosti INT(5) NOT NULL,

Posizione VARCHAR(20),

PRIMARY KEY(IdSpogliatoio)

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Armadietto

-- ------------------------------------

DROP TABLE IF EXISTS Armadietto;

CREATE TABLE Armadietto (

IdArmadietto VARCHAR(5) NOT NULL,

CodiceSblocco INT(4) NOT NULL,

Spogliatoio VARCHAR(5) NOT NULL,

PRIMARY KEY(IdArmadietto),

FOREIGN KEY(Spogliatoio) REFERENCES Spogliatoio(IdSpogliatoio) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Accesso

-- ------------------------------------

DROP TABLE IF EXISTS Accesso;

CREATE TABLE Accesso (

IdAccesso VARCHAR(5) NOT NULL,

Data DATE,

OrarioIngresso TIME,

OrarioUscita TIME,

Tariffa INT(5),

Smartwatch VARCHAR(5) NOT NULL,

Durata INT,

CentroFitness INT,

Armadietto VARCHAR(5) NOT NULL,

Cliente VARCHAR(16) NOT NULL,

PRIMARY KEY(IdAccesso),

FOREIGN KEY(CentroFitness) REFERENCES CentroFitness(IdCentro) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(Armadietto) REFERENCES Armadietto(IdArmadietto) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(Cliente) REFERENCES Cliente(CodFiscale) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Personale

-- ------------------------------------

DROP TABLE IF EXISTS Personale;

CREATE TABLE Personale (

Dipendente VARCHAR(16) NOT NULL,

Giorno VARCHAR(10) NOT NULL,

OrarioIngresso TIME,

OrarioUscita TIME,

PRIMARY KEY(Dipendente, Giorno),

FOREIGN KEY(Dipendente) REFERENCES Dipendente(CodFiscale) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Oraria

-- ------------------------------------

DROP TABLE IF EXISTS Oraria;

CREATE TABLE Oraria (

CentroFitness INT NOT NULL AUTO\_INCREMENT,

Giorno VARCHAR(10) NOT NULL,

OrarioApertura TIME,

OrarioChiusura TIME,

PRIMARY KEY(CentroFitness, Giorno),

FOREIGN KEY(CentroFitness) REFERENCES CentroFitness(IdCentro) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Pagamento

-- ------------------------------------

DROP TABLE IF EXISTS Pagamento;

CREATE TABLE Pagamento (

IdPagamento VARCHAR(5) NOT NULL,

ImportoTotale INT(5) NOT NULL,

Stato VARCHAR(50) NOT NULL,

ImportoAttuale INT,

Contratto VARCHAR(5) NOT NULL,

CONSTRAINT CHK\_Pagamento CHECK(Stato = 'Eseguito' OR Stato = 'Non Ancora Dovuto' OR Stato = 'Scaduto'),

PRIMARY KEY(IdPagamento),

FOREIGN KEY(Contratto) REFERENCES Contratto(IdContratto) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Rata

-- ------------------------------------

DROP TABLE IF EXISTS Rata;

CREATE TABLE Rata (

IdRata VARCHAR(5) NOT NULL,

IstitutoFinanziario VARCHAR(50) NOT NULL,

Importo INT(5) NOT NULL,

DataScadenza DATE,

TassoInteresse INT(3) NOT NULL,

Pagamento VARCHAR(5) NOT NULL,

PRIMARY KEY(IdRata),

FOREIGN KEY(Pagamento) REFERENCES Pagamento(IdPagamento) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Calendario

-- ------------------------------------

DROP TABLE IF EXISTS Calendario;

CREATE TABLE Calendario (

Corso VARCHAR(7) NOT NULL,

Giorno VARCHAR(10) NOT NULL,

OrarioInizio TIME,

OrarioFine TIME,

PRIMARY KEY(Corso, Giorno),

FOREIGN KEY(Corso) REFERENCES Corso(IdCorso) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Regolazione

-- ------------------------------------

DROP TABLE IF EXISTS Regolazione;

CREATE TABLE Regolazione (

ConfigurazioneUtilizzo VARCHAR(10) NOT NULL,

Inclinazione INT,

Carico INT,

Velocita INT,

Tempo INT,

Intensita INT,

Apparecchiatura VARCHAR(5) NOT NULL,

CONSTRAINT CHK\_Regolazione CHECK((Inclinazione >= 0 AND Inclinazione <= 25) AND (Velocita >= 0 AND Velocita <= 20) AND (Intensita >= 1 AND Intensita <= 10) AND (Carico >= 0 AND Carico <= 350)),

PRIMARY KEY(ConfigurazioneUtilizzo),

FOREIGN KEY(Apparecchiatura) REFERENCES Apparecchiatura(IdApparecchiatura) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Utente

-- ------------------------------------

DROP TABLE IF EXISTS Utente;

CREATE TABLE Utente (

Username VARCHAR(20) NOT NULL,

Password VARCHAR(20) NOT NULL,

Cliente VARCHAR(16) NOT NULL,

PRIMARY KEY(Username),

FOREIGN KEY(Cliente) REFERENCES Cliente(CodFiscale) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Amicizia

-- ------------------------------------

DROP TABLE IF EXISTS Amicizia;

CREATE TABLE Amicizia (

UtenteRichiedente VARCHAR(20) NOT NULL,

UtenteRicevente VARCHAR(20) NOT NULL,

DataRichiesta DATE NOT NULL,

DataAccettazione DATE,

PRIMARY KEY(UtenteRichiedente, UtenteRicevente),

FOREIGN KEY(UtenteRichiedente) REFERENCES Utente(Username) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(UtenteRicevente) REFERENCES Utente(Username) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Cerchia

-- ------------------------------------

DROP TABLE IF EXISTS Cerchia;

CREATE TABLE Cerchia (

IdCerchia VARCHAR(5) NOT NULL,

NomeCerchia VARCHAR(50) NOT NULL,

MaxPartecipanti INT NOT NULL,

NumPartecipanti INT NOT NULL,

Utente VARCHAR(20) NOT NULL,

PRIMARY KEY(IdCerchia),

FOREIGN KEY(Utente) REFERENCES Utente(Username) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Valutazione

-- ------------------------------------

DROP TABLE IF EXISTS Valutazione;

CREATE TABLE Valutazione (

IdValutazione VARCHAR(6) NOT NULL,

KcalBruciate INT,

BattitoCardiaco INT,

ConsumoOssigeno INT,

SforzoPsichico VARCHAR(20),

DataFine DATE,

Utente VARCHAR(20) NOT NULL,

Sfida VARCHAR(5) NOT NULL,

CONSTRAINT CHK\_Valutazione CHECK((BattitoCardiaco >= 85 AND BattitoCardiaco <= 220) AND (SforzoPsichico = 'Basso' OR SforzoPsichico = 'Medio' OR SforzoPsichico = 'Alto')),

PRIMARY KEY(IdValutazione),

FOREIGN KEY(Utente) REFERENCES Utente(Username) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(Sfida) REFERENCES Sfida(IdSfida) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Interesse

-- ------------------------------------

DROP TABLE IF EXISTS Interesse;

CREATE TABLE Interesse (

IdInteresse VARCHAR(5) NOT NULL,

Tipo VARCHAR(20) NOT NULL,

PRIMARY KEY(IdInteresse)

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Post

-- ------------------------------------

DROP TABLE IF EXISTS Post;

CREATE TABLE Post (

IdPost VARCHAR(6) NOT NULL,

Testo VARCHAR(100) NOT NULL,

AreaForum VARCHAR(50) NOT NULL,

Timestamp VARCHAR(20) NOT NULL,

Utente VARCHAR(20) NOT NULL,

PRIMARY KEY(IdPost),

FOREIGN KEY(Utente) REFERENCES Utente(Username) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella PostRisposta

-- ------------------------------------

DROP TABLE IF EXISTS PostRisposta;

CREATE TABLE PostRisposta (

IdPostRisposta VARCHAR(15) NOT NULL,

AreaForum VARCHAR(50) NOT NULL,

Timestamp VARCHAR(20) NOT NULL,

Testo VARCHAR(100) NOT NULL,

Utente VARCHAR(20) NOT NULL,

Post VARCHAR(6) NOT NULL,

PRIMARY KEY(IdPostRisposta),

FOREIGN KEY(Utente) REFERENCES Utente(Username) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(Post) REFERENCES Post(IdPost) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Giudizio

-- ------------------------------------

DROP TABLE IF EXISTS Giudizio;

CREATE TABLE Giudizio (

CodGiudizio VARCHAR(5) NOT NULL,

PostRisposta VARCHAR(50) NOT NULL,

Modalita VARCHAR(50) NOT NULL,

Voto INT NOT NULL,

CONSTRAINT CHK\_Giudizio CHECK(Voto >= 1 AND Voto <= 5),

PRIMARY KEY(CodGiudizio, PostRisposta),

FOREIGN KEY(PostRisposta) REFERENCES PostRisposta(IdPostRisposta) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Esterno

-- ------------------------------------

DROP TABLE IF EXISTS Esterno;

CREATE TABLE Esterno (

Url VARCHAR(50) NOT NULL,

Dominio VARCHAR(50) NOT NULL,

Post VARCHAR(10) NOT NULL,

PRIMARY KEY(Url),

FOREIGN KEY(Post) REFERENCES Post(IdPost) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Lavora\_In

-- ------------------------------------

DROP TABLE IF EXISTS Lavora\_In;

CREATE TABLE Lavora\_In (

CentroFitness INT NOT NULL AUTO\_INCREMENT,

Dipendente VARCHAR(16) NOT NULL,

Ruolo VARCHAR(50) NOT NULL,

PRIMARY KEY(CentroFitness, Dipendente),

FOREIGN KEY(CentroFitness) REFERENCES CentroFitness(IdCentro) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(Dipendente) REFERENCES Dipendente(CodFiscale) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Frequenta

-- ------------------------------------

DROP TABLE IF EXISTS Frequenta;

CREATE TABLE Frequenta (

CentroFitness INT NOT NULL AUTO\_INCREMENT,

Cliente VARCHAR(16) NOT NULL,

PRIMARY KEY(CentroFitness, Cliente),

FOREIGN KEY(CentroFitness) REFERENCES CentroFitness(IdCentro) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(Cliente) REFERENCES Cliente(CodFiscale) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Stipula

-- ------------------------------------

DROP TABLE IF EXISTS Stipula;

CREATE TABLE Stipula (

CentroFitness INT NOT NULL AUTO\_INCREMENT,

Contratto VARCHAR(5) NOT NULL,

PRIMARY KEY(CentroFitness, Contratto),

FOREIGN KEY(CentroFitness) REFERENCES CentroFitness(IdCentro) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(Contratto) REFERENCES Contratto(IdContratto) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Accede

-- ------------------------------------

DROP TABLE IF EXISTS Accede;

CREATE TABLE Accede (

Contratto VARCHAR(5) NOT NULL,

Piscina VARCHAR(5) NOT NULL,

PRIMARY KEY(Contratto, Piscina),

FOREIGN KEY(Contratto) REFERENCES Contratto(IdContratto) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(Piscina) REFERENCES Piscina(IdPiscina) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Gestisce

-- ------------------------------------

DROP TABLE IF EXISTS Gestisce;

CREATE TABLE Gestisce (

Dipendente VARCHAR(16) NOT NULL,

Sala VARCHAR(5) NOT NULL,

PRIMARY KEY(Dipendente, Sala),

FOREIGN KEY(Dipendente) REFERENCES Dipendente(CodFiscale) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(Sala) REFERENCES Sala(IdSala) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Accede\_1

-- ------------------------------------

DROP TABLE IF EXISTS Accede\_1;

CREATE TABLE Accede\_1 (

Contratto VARCHAR(5) NOT NULL,

Sala VARCHAR(5) NOT NULL,

PRIMARY KEY(Contratto, Sala),

FOREIGN KEY(Contratto) REFERENCES Contratto(IdContratto) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(Sala) REFERENCES Sala(IdSala) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Accede\_2

-- ------------------------------------

DROP TABLE IF EXISTS Accede\_2;

CREATE TABLE Accede\_2 (

Contratto VARCHAR(5) NOT NULL,

Corso VARCHAR(6) NOT NULL,

PRIMARY KEY(Contratto, Corso),

FOREIGN KEY(Contratto) REFERENCES Contratto(IdContratto) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(Corso) REFERENCES Corso(IdCorso) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Partecipa

-- ------------------------------------

DROP TABLE IF EXISTS Partecipa;

CREATE TABLE Partecipa (

Cliente VARCHAR(16) NOT NULL,

Corso VARCHAR(6) NOT NULL,

PRIMARY KEY(Cliente, Corso),

FOREIGN KEY(Cliente) REFERENCES Cliente(CodFiscale) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(Corso) REFERENCES Corso(IdCorso) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Contenuta

-- ------------------------------------

DROP TABLE IF EXISTS Contenuta;

CREATE TABLE Contenuta (

SchedaAlimentazione VARCHAR(5) NOT NULL,

Dieta VARCHAR(5) NOT NULL,

PRIMARY KEY(SchedaAlimentazione, Dieta),

FOREIGN KEY(SchedaAlimentazione) REFERENCES SchedaAlimentazione(IdSchedaAlimentazione) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(Dieta) REFERENCES Dieta(IdDieta) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Acquista

-- ------------------------------------

DROP TABLE IF EXISTS Acquista;

CREATE TABLE Acquista (

Cliente VARCHAR(16) NOT NULL,

Integratore VARCHAR(6) NOT NULL,

Data DATE,

Quantita INT(5) NOT NULL,

PRIMARY KEY(Cliente, Integratore),

FOREIGN KEY(Cliente) REFERENCES Cliente(CodFiscale) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(Integratore) REFERENCES Integratore(IdIntegratore) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnODB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Si\_Associa

-- ------------------------------------

DROP TABLE IF EXISTS Si\_Associa;

CREATE TABLE Si\_Associa (

Integratore VARCHAR(5) NOT NULL,

Dieta VARCHAR(5) NOT NULL,

PRIMARY KEY(Integratore, Dieta),

FOREIGN KEY(Integratore) REFERENCES Integratore(IdIntegratore) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(Dieta) REFERENCES Dieta(IdDieta) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Richiede

-- ------------------------------------

DROP TABLE IF EXISTS Richiede;

CREATE TABLE Richiede (

Apparecchiatura VARCHAR(5) NOT NULL,

Esercizio VARCHAR(5) NOT NULL,

PRIMARY KEY(Apparecchiatura, Esercizio),

FOREIGN KEY(Apparecchiatura) REFERENCES Apparecchiatura(IdApparecchiatura) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(Esercizio) REFERENCES Esercizio(IdEsercizio) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Aderisce

-- ------------------------------------

DROP TABLE IF EXISTS Aderisce;

CREATE TABLE Aderisce (

Utente VARCHAR(20) NOT NULL,

Cerchia VARCHAR(5) NOT NULL,

PRIMARY KEY(Utente, Cerchia),

FOREIGN KEY(Utente) REFERENCES Utente(Username) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(Cerchia) REFERENCES Cerchia(IdCerchia) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Partecipa\_1

-- ------------------------------------

DROP TABLE IF EXISTS Partecipa\_1;

CREATE TABLE Partecipa\_1 (

Utente VARCHAR(20) NOT NULL,

Sfida VARCHAR(5) NOT NULL,

Ruolo VARCHAR(50) NOT NULL,

PRIMARY KEY(Utente, Sfida),

FOREIGN KEY(Utente) REFERENCES Utente(Username) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(Sfida) REFERENCES Sfida(IdSfida) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Nutre

-- ------------------------------------

DROP TABLE IF EXISTS Nutre;

CREATE TABLE Nutre (

Cerchia VARCHAR(5) NOT NULL,

Interesse VARCHAR(5) NOT NULL,

PRIMARY KEY(Cerchia, Interesse),

FOREIGN KEY(Cerchia) REFERENCES Cerchia(IdCerchia) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(Interesse) REFERENCES Interesse(IdInteresse) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

-- ------------------------------------

-- Tabella Fa\_Uso

-- ------------------------------------

DROP TABLE IF EXISTS Fa\_Uso;

CREATE TABLE Fa\_Uso (

Post VARCHAR(6) NOT NULL,

Sfida VARCHAR(5) NOT NULL,

PRIMARY KEY(Post, Sfida),

FOREIGN KEY(Post) REFERENCES Post(IdPost) ON DELETE CASCADE ON UPDATE NO ACTION,

FOREIGN KEY(Sfida) REFERENCES Sfida(IdSfida) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE = InnoDB DEFAULT CHARSET = Latin1;

**Popolamento:**

-- ------------------------------------

-- Popolamento CentroFitness

-- ------------------------------------

INSERT INTO CentroFitness

VALUES ('1','84127','500','17','Salerno','089/2435625','300','Arturo Di Napoli'),

('2','00118','300','17','Roma','06/25634987','200','I Laziali da Roma'),

('3','00119','400','71','Roma','06/23164235','250','Senad Lulic'),

('4','80011','150','10','Acerra','081/1326487','50','Diego Armando Maradona'),

('5','97010','75','8','Roccazzo','0932/234961','30','Col vento'),

('6','63065','120','13','Ripatransone','0735/243564','40','Ndante'),

('7','47025','175','44','Ciola','0547/243532','99','Obbligo del ghetto');

-- ------------------------------------

-- Popolamento Dipendente

-- ------------------------------------

INSERT INTO Dipendente

VALUES

('PDC01','Paolo','Di Canio','00118','1968-07-09','I Laziali da Roma','5','Roma','NP01','DOC1'),

('ANF02','Anna','Franco','00121','1990-04-25','Della Vittoria','15','Roma','NP02','DOC2'),

('ENO03','Enrico','Ottaro','00119','1980-08-04','Senad Lulic','25','Roma','NP03','DOC3'),

('GEM04','Germano','Mosconi','00120','1984-12-25','Delle portechiuse','35','Roma','NP04','DOC4'),

('ENP05','Enza','Puccio','84127','1977-12-31','Alza le mani','45','Salerno','NP05','DOC5'),

('MUP06','Mufasa','Pippot','00123','1979-06-17','Da qui','55','Roma','NP06','DOC6'),

('JAR07','Jack','Riper','63065','1983-02-15','Del dolore','65','Ripatransone','NP07','DOC7'),

('OLN08','Olga','Nigramma','84127','1988-11-16','Marco Fassone','75','Salerno','NP08','DOC8'),

('OLR09','Olindo','Romano','00124','1997-02-12','Regina Cieli','85','Roma','NP09','DOC9'),

('ROB10','Rosa','Bazzi','00124','1999-09-12','Regina Cieli','95','Roma','NP10','DOC10'),

('FAO11','Franco','Ordine','63065','1974-02-06','La Bianca','25','Ripatransone','NP11','DOC11'),

('BIC12','Bill','Cosby','84127','1972-07-12','Cloroformio','35','Salerno','NP12','DOC12'),

('TOR13','Tonino','Rudiger','47025','1993-03-03','Dei calzini','45','Ciola','NP13','DOC13'),

('PAC14','Pamela','Candela','84127','1978-08-13','Della mattonella','55','Salerno','NP14','DOC14'),

('MAB15','Marco','Branca','84127','1965-01-06','Dalla Nazione','65','Salerno','NP15','DOC15'),

('ALB16','Al','Bano','80011','1992-05-20','Romina Forza','75','Acerra','NP16','DOC16'),

('ROP17','Rocco','Pastore','80011','1997-08-01','Po','85','Acerra','NP17','DOC17'),

('OSR18','Oscar','Rafone','80011','1993-11-01','Tocco','15','Acerra','NP18','DOC18'),

('MIL19','Michele','Lozio','97010','1995-08-26','Asus','25','Roccazzo','NP19','DOC19'),

('GIC20','Gianni','Celeste','97010','1964-06-24','U Latitante','35','Roccazzo','NP20','DOC20'),

('PAR21','Pape','Reina','47025','1982-08-31','Dal Milan','45','Ciola','NP21','DOC21');

-- ------------------------------------

-- Popolamento Medico

-- ------------------------------------

INSERT INTO Medico

VALUES ('M1','Paolo','Veronesi','089/2435623','Veronesi'),

('M2','Giovanni','Battista','06/25634955','Battista'),

('M3','Marco','Evangelista','06/23164243','Evangelista'),

('M4','Matteo','Ianni','081/1326442','Ianni'),

('M5','Pietro','Simon','0932/234932','Simon'),

('M6','Isaia','Hjsaj','0735/243512','Hjsaj'),

('M7','Lucrezio','Panditoni','0547/243511','Panditoni');

-- ------------------------------------

-- Popolamento Cliente

-- ------------------------------------

INSERT INTO Cliente

VALUES

('GB1','Gianluca','Bazuca','Roma','13','00118','NP1','Cod2','1965-06-12','I laziali da Roma','Pesce','M1'),

('VV2','Virgil','Van Djik','Roma','16','00121','NP2','Cod5','1995-09-15','Fillo','Salatini','M7'),

('ED3','Edin','Dzeko','Roma','19','00118','NP1','Cod8','1967-012-18','Rastrello','Verdura','M1'),

('FO4','Francesco','Ordine','Roma','20','00119','NP1','Cod9','1969-01-19','Pipistrello','Carne','M1'),

('GI5','Giovanni','Impastato','Salerno','22','84127','NP3','Cod11','1984-03-21','Casadei','Frutta','M2'),

('AB6','Alisson','Becker','Salerno','25','84127','NP3','Cod14','1992-06-24','Toto Cutugno','Pane','M2'),

('AL7','Alessia','Lete','Salerno','26','84127','NP3','Cod15','1993-07-25','Calleri','Pane','M2'),

('LE8','Lapo','Elkann','Ripatransone','30','63065','NP5','Cod19','1977-11-29','Perticone','Pasta','M4'),

('GC9','Giosuè','Castello','Acerra','32','80011','NP4','Cod21','1992-12-01','Prosecco','Pasta','M3'),

('PD10','Paolo','Diaz','Acerra','33','80011','NP4','Cod22','1981-11-02','Pupazzo','Pasta','M3'),

('PC11','Peter','Comeback','Ripatransone','36','63065','NP5','Cod25','1963-08-05',

'Ndante','Carne','M4'),

('EB12','Eleonora','Boi','Roccazzo','40','97010','NP7','Cod29','1956-04-09','Col vento','Pesce','M6'),

('DL13','Nicola','Di Bari','Ciola','44','47025','NP6','Cod33','2001-05-17','Obbligo del ghetto','Verdura','M5'),

('CL14','Claudio','Lotirchio','Roma','46','00118','NP1','Cod35','1996-02-25','I laziali da Roma','Pasta','M1'),

('IZ15','Iva','Zanicchi','Ciola','48','47025','NP6','Cod37','1992-09-30','Obbligo del ghetto','Pesce','M5');

-- ------------------------------------

-- Popolamento Contratto

-- ------------------------------------

INSERT INTO Contratto

VALUES ('C01','2 mesi','50','Silver','3','2','Consulente1','2018-02-02','VV2'),

('C02','12 mesi','300','Platinum','5','8','Consulente2','2017-09-01','ED3'),

('C03','6 mesi','150','Gold','4','6','Consulente3','2018-03-04','EB12'),

('C04','3 mesi','70','Gold','4','3','Consulente4','2018-04-06','DL13'),

('C05','12 mesi','300','Platinum','5','8','Consulente5','2018-01-05','LE8'),

('C06','12 mesi','300','Platinum','5','8','Consulente6','2017-05-03','CL14'),

('C07','1 mese','30','Silver','3','1','Consulente7','2018-04-10','IZ15');

-- ------------------------------------

-- Popolamento Piscina

-- ------------------------------------

INSERT INTO Piscina

VALUES ('P01','50X10','Interna','1'),

('P02','50X10','Interna','2'),

('P03','50X10','Interna','3'),

('P04','10X5','Esterna','4'),

('P05','50X10','Interna','4'),

('P06','50X10','Interna','6'),

('P07','50X10','Interna','7');

-- ------------------------------------

-- Popolamento Sala

-- ------------------------------------

INSERT INTO Sala

VALUES ('S01','Sala Fitness','1'),

('S02','Sala Attrezzi','2'),

('S03','Sala Fitness','3'),

('S04','Sala Attrezzi','4'),

('S05','Sala Corpo Libero','5'),

('S06','Sala Boxe','6'),

('S07','Sala Momo','7');

-- ------------------------------------

-- Popolamento Corso

-- ------------------------------------

INSERT INTO Corso

VALUES ('CORSO1','70','Boxe','Principiante','2018-05-13','2018-07-13','S01',NULL,'MAB15'),

('CORSO2','80','Zumba','Intermedio','2018-06-12','2018-07-16','S03',NULL,'ROB10'),

('CORSO3','50','Nuoto sincronizzato','Avanzato','2018-07-13','2019-05-13',

NULL,'P01','MAB15'),

('CORSO4','20','Tuffi','Intermedio','2018-04-14','2018-06-29',NULL,'P04','OSR18'),

('CORSO5','15','Bodybuilding','Intermedio','2018-08-06','2018-09-17','S03',NULL,'ROB10'),

('CORSO6','18','Nuoto avanzato','Avanzato','2018-09-30','2018-11-13',

NULL,'P07','TOR13'),

('CORSO7','15','Ginnastica artistica','Principiante','2018-10-27','2018-12-31',

'S06',NULL,'FAO11');

-- ------------------------------------

-- Popolamento Scopo

-- ------------------------------------

INSERT INTO Scopo

VALUES ('SCOPO1','Dimagrimento','VV2'),

('SCOPO2','Potenziamento Muscolare','ED3'),

('SCOPO3','Dimagrimento','EB12'),

('SCOPO4','Scopo Ricreativo','DL13'),

('SCOPO5','Potenziamento Muscolare','LE8'),

('SCOPO6','Dimagrimento','CL14'),

('SCOPO7','Scopo Ricreativo','IZ15');

-- ------------------------------------

-- Popolamento PotenziamentoMuscolare

-- ------------------------------------

INSERT INTO PotenziamentoMuscolare

VALUES ('SCOPO2','Tricipite','Lieve'),

('SCOPO5','Deltoide Posteriore','Moderato');

-- ------------------------------------

-- Popolamento Misurazione

-- ------------------------------------

INSERT INTO Misurazione

VALUES ('MS01','10','2018-04-01','70','30','95','2018-04-03','FO4','M1'),

('MS02','15','2018-02-12','50','50','85','2018-02-14','AL7','M2'),

('MS03','8','2018-03-15','25','75','70','2018-03-17','PD10','M3'),

('MS04','15','2018-04-11','40','60','78','2018-04-13','PC11','M4'),

('MS05','13','2018-01-07','20','80','73','2018-01-09','GI5','M5'),

('MS06','18','2018-03-18','95','5','130','2018-03-20','CL14','M6'),

('MS07','14','2018-04-03','50','50','82','2018-04-05','DL13','M7');

-- ------------------------------------

-- Popolamento Sfida

-- ------------------------------------

INSERT INTO Sfida

VALUES ('SF1','Finire 10 schede di allenamento','2018-04-12','2018-06-18','2018-06-28'),

('SF2','Perdere peso','2018-04-13','2018-06-19','2018-06-29'),

('SF3','Aumentare la massa muscolare','2018-04-14','2018-06-20','2018-06-30'),

('SF4','Perdere peso','2018-04-15','2018-06-21','2018-07-12'),

('SF5','Ridurre i tempi di recupero','2018-04-16','2018-06-22','2018-07-15');

-- ------------------------------------

-- Popolamento SchedaAlimentazione

-- ------------------------------------

INSERT INTO SchedaAlimentazione

VALUES ('SA1','Perdere Peso','2018-03-21','2018-08-21','5 Mesi','CL14','M6',NULL),

('SA2','Acquisire Massa Muscolare','2018-04-15','2018-07-15','3 Mesi','AL7','M2',NULL),

('SA3','Prendere Peso','2018-02-08','2018-03-08','1 Mese','PD10','M3',NULL),

('SA4','Diminuire Massa Grassa','2018-01-13','2018-02-13','1 Mese','PC11','M4',NULL),

('SA5','Perdere Peso','2018-04-10','2018-06-10','2 Mesi','GI5','M5',NULL),

('SA6','Acquisire Massa Muscolare','2018-03-31','2018-07-31','4 Mesi','FO4','M1',NULL),

('SA7','Prendere Peso','2018-02-27','2018-05-27','3 Mesi','DL13','M7',NULL),

('SA8','Perdere Peso','2018-04-13','2018-06-19','2 Mesi',NULL,NULL,'SF2'),

('SA9','Perdere Peso','2018-04-12','2018-06-18','2 Mesi',NULL,NULL,'SF1'),

('SA10','Diminuire Massa Grassa','2018-04-15','2018-06-21','2 Mesi',NULL,NULL,'SF4');

-- ------------------------------------

-- Popolamento Dieta

-- ------------------------------------

INSERT INTO Dieta

VALUES ('D1','2800','2400','5','Pasto1','M1'),

('D2','3000','2600','6','Pasto2','M2'),

('D3','2200','1800','4','Pasto3','M3'),

('D4','2000','1700','3','Pasto4','M6'),

('D5','2400','2000','5','Pasto5','M7');

-- ------------------------------------

-- Popolamento Magazzino

-- ------------------------------------

INSERT INTO Magazzino

VALUES ('M1','100'),

('M2','120'),

('M3','90'),

('M4','50'),

('M5','70'),

('M6','60'),

('M7','80');

-- ------------------------------------

-- Popolamento Fornitore

-- ------------------------------------

INSERT INTO Fornitore

VALUES ('FOR1','Iva1','1264758668','84127','Sas','Salerno','14','Dei delitti','Gruppo Sas1'),

('FOR2','Iva2','1264758644','00118','Spa','Roma','17','Delle pene','Gruppo Spa1'),

('FOR3','Iva3','1264755558','00119','Srl','Roma','19','Geforce','Gruppo Srl1'),

('FOR4','Iva4','1264758669','80011','Sapa','Acerra','24','Mouse','Gruppo Sapa1'),

('FOR5','Iva5','1264758662','97010','Spa','Roccazzo','29','Tastiera','Gruppo Spa2'),

('FOR6','Iva6','1264758661','63065','Sapa','Ripatransone','23','Display','Gruppo Sapa2'),

('FOR7','Iva7','1264758663','47025','Srl','Ciola','21','Franco Califano','Gruppo Srl2');

-- ------------------------------------

-- Popolamento Integratore

-- ------------------------------------

INSERT INTO Integratore

VALUES ('INTE1','2018-05-12','Liquida','Acetil L.Carnitina','Acetil Carnitina','250','400','M1','FOR1'),

('INTE2','2018-05-13','Solida','L.Timosina','Timosina','500','500','M2','FOR2'),

('INTE3','2018-06-14','Liquida','L.Gluttamina','Gluttamina','250','300','M3','FOR3'),

('INTE4','2018-06-15','Solida','Bcaa','Leucina','1000','400','M4','FOR4'),

('INTE5','2018-05-16','Liquida','S.Taurina','Taurina','250','500','M5','FOR5'),

('INTE6','2018-06-17','Solida','Creatina Monoidrato','Creatina','250','400','M6','FOR6'),

('INTE7','2018-06-18','Solida','Maltodestrina','Glutine','1000','350','M7','FOR7'),

('INTE8','2018-04-25','Liquida','S1.Taurina','Taurina','250','500','M5','FOR5'),

('INTE9','2018-04-30','Liquida','S2.Taurina','Taurina','250','500','M5','FOR5'),

('INTE10','2018-05-20','Solida','Bcaa (1)','Leucina','1000','400','M4','FOR4'),

('INTE11','2018-06-25','Solida','Bcaa (2)','Leucina','1000','400','M4','FOR4');

-- ------------------------------------

-- Popolamento Ordine

-- ------------------------------------

INSERT INTO Ordine

VALUES ('FRANCO1','FRANK1','2018-02-13','2018-02-03','Evaso','FOR1','1'),

('FRANCO2','FRANK2',NULL,NULL,'Incompleto','FOR2','2'),

('FRANCO3','FRANK3','2018-03-15','2018-03-05','Evaso','FOR3','3'),

('FRANCO4','FRANK4','2018-04-16','2018-04-06','Evaso','FOR4','4'),

('FRANCO5','FRANK5','2018-03-17','2018-03-07','Evaso','FOR5','5'),

('FRANCO6','FRANK6',NULL,NULL,'Incompleto','FOR6','6'),

('FRANCO7','FRANK7','2018-01-19','2018-01-09','Evaso','FOR7','7');

-- ------------------------------------

-- Popolamento Apparecchiatura

-- ------------------------------------

INSERT INTO Apparecchiatura

VALUES ('APP1','Elittica','Riscaldamento gambe','50','SI','100','S01','1'),

('APP2','Lat Machine','Potenziamento spalle','30','NO','45','S01','1'),

('APP3','Lat Machine','Potenziamento spalle','29','NO','22','S02','2'),

('APP4','Parallele','Distensione muscolare','13','SI','100','S02','3'),

('APP5','Panca','Potenziamento','76','SI','100','S03','2'),

('APP6','Cyclette','Dimagrimento','67','SI','100','S03','3'),

('APP7','Tapis Roulant','Dimagrimento','89','NO','46','S04','5'),

('APP8','Leg Press','Potenziamento delle gambe','23','SI','100','S04','4'),

('APP9','Pesi','Potenziamento','12','SI','100','S05','6'),

('APP10','Parallele','Distensione muscolare','45','NO','87','S05','4'),

('APP11','Cyclette','Dimagrimento','36','SI','100','S06','5'),

('APP12','Pesi','Potenziamento','98','SI','100','S06','7'),

('APP13','Tapis Roulant','Dimagrimento','99','SI','100','S07','7'),

('APP14','Vogatore','Potenziamento','26','NO','92','S07','6');

-- ------------------------------------

-- Popolamento SchedaAllenamento

-- ------------------------------------

INSERT INTO SchedaAllenamento

VALUES

('SAL1','95','Incompleta','2018-04-18',NULL,'173','70','10','2018-05-15','30','PDC01','FO4',NULL),

('SAL2','85','Incompleta','2018-03-21',NULL,'183','50','15','2018-06-18','50','OLN08','AL7',NULL),

('SAL3','70','Incompleta','2018-02-05',NULL,'175','25','8','2018-04-30','75','ALB16','PD10',NULL),

('SAL4','78','Completa','2018-01-07',NULL,'165','40','15','2018-04-13','60','FAO11','PC11',NULL),

('SAL5','73','Incompleta','2018-03-04',NULL,'185','20','13','2018-05-01','80','OLN08','GI5',NULL),

('SAL6','130','Incompleta','2018-02-05',NULL,'180','95','18','2020-08-08','5','PDC01','CL14',NULL),

('SAL7','82','Incompleta','2018-03-21',NULL,'176','50','14','2018-05-06','50','PAR21','DL13',NULL),

('SAL8','100','Incompleta','2018-02-25',NULL,'156','25','25','2018-04-28','75','PAR21','IZ15','SF1'),

('SAL9','93','Completa','2018-01-15',NULL,'195','5','20','2018-03-25','95','ANF02','ED3','SF5'),

('SAL10','90','Incompleta','2018-02-15',NULL,'201','20','20','2018-05-05','80','PDC01','VV2','SF3');

-- ------------------------------------

-- Popolamento Esercizio

-- ------------------------------------

INSERT INTO Esercizio

VALUES ('EX1','Rematore Su Macchina Verticale','23','60','12','Aerobico','10:50:00',NULL,'SAL1'),

('EX2','Lat Machine Anteriore','24','30','11','Anaerobico','11:15:00',NULL,'SAL2'),

('EX3','Lat Machine Posteriore','25','40','13','Aerobico','11:30:00',NULL,'SAL3'),

('EX4','Distensioni Su Panca','26','20','14','Anaerobico','13:00:00',NULL,'SAL9'),

('EX5','Corsa Su Tapis Roulant','27',NULL,NULL,'Aerobico','13:20:00','120','SAL6'),

('EX6','Leg Press','28','100','12','Aerobico','15:45:00',NULL,'SAL4'),

('EX7','Push Down','29','50','15','Anaerobico','16:15:00',NULL,'SAL5'),

('EX8','Squat','30','30','12','Aerobico','17:00:00',NULL,'SAL7'),

('EX9','Curl Su Macchina','31','30','14','Anaerobico','18:00:00',NULL,'SAL8'),

('EX10','Cyclette','32',NULL,NULL,'Aerobico','18:40:00','30','SAL10');

-- ------------------------------------

-- Popolamento Richiede

-- ------------------------------------

INSERT INTO Richiede

VALUES ('APP12','EX1'),

('APP2','EX2'),

('APP3','EX3'),

('APP5','EX4'),

('APP13','EX5'),

('APP8','EX6'),

('APP12','EX7'),

('APP12','EX8'),

('APP12','EX9'),

('APP6','EX10');

-- ------------------------------------

-- Popolamento Sessione

-- ------------------------------------

INSERT INTO Sessione

VALUES ('SS1','30','11:00:00','11:05:00','12','EX1'),

('SS2','32','11:15:00','11:20:00','6','EX2'),

('SS3','33','11:30:00','11:40:00','13','EX3'),

('SS4','35','13:00:00','13:05:00','14','EX4'),

('SS5','0','13:30:00','15:20:00',NULL,'EX5'),

('SS6','53','16:00:00','16:05:00','12','EX6'),

('SS7','52','16:20:00','16:30:00','10','EX7'),

('SS8','60','17:00:00','17:15:00','12','EX8'),

('SS9','32','19:10:00','19:20:00',NULL,'EX10'),

('SS10','32','19:00:00','19:10:00',NULL,'EX10');

-- ------------------------------------

-- Popolamento Spogliatoio

-- ------------------------------------

INSERT INTO Spogliatoio

VALUES ('SP1','15','50','Ala Est'),

('SP2','15','50','Ala Ovest'),

('SP3','15','50','Centrale'),

('SP4','12','35','Ala Sud'),

('SP5','12','32','Ala Nord'),

('SP6','12','30','Ala Ovest'),

('SP7','10','28','Ala Est');

-- ------------------------------------

-- Popolamento Armadietto

-- ------------------------------------

INSERT INTO Armadietto

VALUES ('ARMA1','3452','SP1'),

('ARMA2','1298','SP2'),

('ARMA3','5421','SP3'),

('ARMA4','9864','SP4'),

('ARMA5','6709','SP5'),

('ARMA6','1365','SP6'),

('ARMA7','9032','SP7');

-- ------------------------------------

-- Popolamento Accesso

-- ------------------------------------

INSERT INTO Accesso

VALUES ('AX1','2018-03-22','11:00:00','13:00:00',NULL,'SM1',NULL,'2','ARMA2','GB1'),

('AX2','2018-03-22','11:30:00','13:30:00',NULL,'SM2',NULL,'2','ARMA2','VV2'),

('AX3','2018-03-22','15:00:00','17:00:00',NULL,'SM3',NULL,'2','ARMA2','ED3'),

('AX4','2018-03-12','13:30:00','14:50:00',NULL,'SM4',NULL,'3','ARMA3','FO4'),

('AX5','2018-02-12','09:00:00','11:00:00',NULL,'SM5',NULL,'1','ARMA1','GI5'),

('AX6','2018-02-12','09:30:00','11:00:00',NULL,'SM6',NULL,'1','ARMA1','AB6'),

('AX7','2018-02-12','12:00:00','15:00:00',NULL,'SM7',NULL,'1','ARMA1','AL7'),

('AX8','2018-04-17','18:00:00','19:00:00',NULL,'SM8',NULL,'6','ARMA6','LE8'),

('AX9','2018-03-15','11:50:00','13:30:00',NULL,'SM9',NULL,'4','ARMA4','GC9'),

('AX10','2018-03-10','10:00:00','11:25:00',NULL,'SM10',NULL,'4','ARMA4','PD10'),

('AX11','2018-02-11','12:40:00','13:30:00',NULL,'SM11',NULL,'6','ARMA6','PC11'),

('AX12','2018-01-12','11:20:00','14:00:00',NULL,'SM12',NULL,'5','ARMA5','EB12'),

('AX13','2018-01-14','17:00:00','18:30:00',NULL,'SM13',NULL,'7','ARMA7','DL13'),

('AX14','2018-03-12','09:00:00','11:00:00',NULL,'SM14',NULL,'3','ARMA3','CL14'),

('AX15','2018-04-02','09:45:00','11:55:00',NULL,'SM15',NULL,'7','ARMA7','IZ15');

-- ------------------------------------

-- Popolamento Personale

-- ------------------------------------

INSERT INTO Personale

VALUES ('PDC01','Lunedì','09:00:00','13:00:00'),

('ANF02','Martedì','13:00:00','18:00:00'),

('ENP05','Mercoledì','15:00:00','20:00:00'),

('OLN08','Giovedì','09:00:00','11:00:00'),

('ROB10','Venerdì','10:00:00','15:00:00'),

('OLR09','Sabato','18:00:00','20:00:00'),

('ROP17','Lunedì','12:00:00','20:00:00'),

('TOR13','Martedì','09:00:00','12:00:00'),

('GIC20','Sabato','12:00:00','18:00:00'),

('PAR21','Giovedì','15:00:00','20:00:00');

-- ------------------------------------

-- Popolamento Oraria

-- ------------------------------------

INSERT INTO Oraria

VALUES ('1','Lunedì','09:00:00','20:00:00'),

('2','Martedì','13:00:00','20:00:00'),

('3','Mercoledì','09:00:00','20:00:00'),

('4','Giovedì','09:00:00','13:00:00'),

('5','Venerdì','09:00:00','20:00:00'),

('6','Sabato','09:00:00','20:00:00'),

('7','Lunedì','09:00:00','20:00:00');

-- ------------------------------------

-- Popolamento Pagamento

-- ------------------------------------

INSERT INTO Pagamento

VALUES ('PAG01','50','Eseguito','50','C01'),

('PAG02','300','Eseguito','300','C02'),

('PAG03','150','Eseguito','150','C03'),

('PAG04','70','Eseguito','70','C04'),

('PAG05','300','Non Ancora Dovuto','0','C05'),

('PAG06','300','Non Ancora Dovuto','0','C06'),

('PAG07','30','Eseguito','30','C07');

-- ------------------------------------

-- Popolamento Rata

-- ------------------------------------

INSERT INTO Rata

VALUES ('R1','IS1','50','2018-01-15','2','PAG06'),

('R2','IS1','50','2018-02-15','3','PAG06'),

('R3','IS1','50','2018-03-15','2','PAG06'),

('R4','IS2','100','2018-01-25','3','PAG05'),

('R5','IS2','100','2018-02-25','2','PAG05');

-- ------------------------------------

-- Popolamento Calendario

-- ------------------------------------

INSERT INTO Calendario

VALUES ('CORSO1','Lunedì','10:00:00','12:00:00'),

('CORSO2','Martedì','15:00:00','16:00:00'),

('CORSO3','Mercoledì','17:00:00','18:00:00'),

('CORSO4','Giovedì','09:00:00','11:00:00'),

('CORSO5','Venerdì','12:00:00','13:00:00'),

('CORSO6','Sabato','15:00:00','17:00:00'),

('CORSO7','Lunedì','17:00:00','18:00:00');

-- ------------------------------------

-- Popolamento Regolazione

-- ------------------------------------

INSERT INTO Regolazione

VALUES ('CONFIG01',NULL,'60',NULL,NULL,NULL,'APP14'),

('CONFIG02',NULL,'30',NULL,NULL,NULL,'APP2'),

('CONFIG03',NULL,'40',NULL,NULL,NULL,'APP3'),

('CONFIG04',NULL,'20',NULL,NULL,NULL,'APP5'),

('CONFIG05','15',NULL,'15','120',NULL,'APP7'),

('CONFIG06',NULL,'100',NULL,NULL,NULL,'APP8'),

('CONFIG07',NULL,'50',NULL,NULL,NULL,'APP12'),

('CONFIG08',NULL,'30',NULL,NULL,NULL,'APP12'),

('CONFIG09',NULL,'30',NULL,NULL,NULL,'APP12'),

('CONFIG10',NULL,NULL,NULL,'30','2','APP11');

-- ------------------------------------

-- Popolamento Utente

-- ------------------------------------

INSERT INTO Utente

VALUES ('Username1','Bazuka65','GB1'),

('Username2','Mr 87 milioni','VV2'),

('Username3','Ciao Barça','ED3'),

('Username4','Solo bianca','FO4'),

('Username5','Buonanima','GI5'),

('Username6','La paro io','AB6'),

('Username7','Acqua frizzante','AL7'),

('Username8','Altra bianca','LE8'),

('Username9','Forza Napoli','GC9'),

('Username10','Ovvio','PD10'),

('Username11','Indietro Torno','PC11'),

('Username12','Mediaset Premium','EB12'),

('Username13','Bendò','DL13'),

('Username14','Paga la luce','CL14'),

('Username15','Zia di Bendò','IZ15');

-- ------------------------------------

-- Popolamento Amicizia

-- ------------------------------------

INSERT INTO Amicizia

VALUES ('Username14','Username15','2018-02-02',NULL),

('Username2','Username3','2018-03-03','2018-03-04'),

('Username6','Username1','2018-01-01','2018-01-03'),

('Username14','Username9','2018-04-05',NULL),

('Username5','Username4','2018-03-05','2018-03-07'),

('Username13','Username7','2018-03-27','2018-04-01'),

('Username12','Username3','2018-04-04','2018-04-04');

-- ------------------------------------

-- Popolamento Cerchia

-- ------------------------------------

INSERT INTO Cerchia

VALUES ('CER1','Asso','40','1','Username14'),

('CER2','Re','50','1','Username7'),

('CER3','Jack','40','1','Username5'),

('CER4','Donna','50','1','Username10'),

('CER5','Joker','40','1','Username2');

-- ------------------------------------

-- Popolamento Valutazione

-- ------------------------------------

INSERT INTO Valutazione

VALUES ('VAL1','1000','88','14',NULL,NULL,'Username12','SF1'),

('VAL2','800','120','12',NULL,'2018-04-02','Username13','SF3'),

('VAL3','10','220','60',NULL,NULL,'Username14','SF2'),

('VAL4','700','160','32',NULL,'2018-04-06','Username3','SF4'),

('VAL5','650','195','21',NULL,'2018-04-08','Username2','SF5');

-- ------------------------------------

-- Popolamento Interesse

-- ------------------------------------

INSERT INTO Interesse

VALUES ('INTR1','Bastoni'),

('INTR2','Denari'),

('INTR3','Spade'),

('INTR4','Coppe'),

('INTR5','Picche');

-- ------------------------------------

-- Popolamento Post

-- ------------------------------------

INSERT INTO Post

VALUES ('POST1','Testo1','Af1','Ts1','Username1'),

('POST2','Testo2','Af2','Ts2','Username2'),

('POST3','Testo3','Af3','Ts3','Username3'),

('POST4','Testo4','Af4','Ts4','Username4'),

('POST5','Testo5','Af5','Ts5','Username5'),

('POST6','Testo6','Af6','Ts6','Username6'),

('POST7','Testo7','Af7','Ts7','Username7'),

('POST8','Testo8','Af8','Ts8','Username8'),

('POST9','Testo9','Af9','Ts9','Username9'),

('POST10','Testo10','Af10','Ts10','Username10'),

('POST11','Testo11','Af11','Ts11','Username11'),

('POST12','Testo12','Af12','Ts12','Username12'),

('POST13','Testo13','Af13','Ts13','Username13'),

('POST14','Testo14','Af14','Ts14','Username14'),

('POST15','Testo15','Af15','Ts15','Username15');

-- ------------------------------------

-- Popolamento PostRisposta

-- ------------------------------------

INSERT INTO PostRisposta

VALUES ('RISP1','Af1','Ts1','Testo1','Username15','POST1'),

('RISP2','Af2','Ts2','Testo2','Username14','POST2'),

('RISP3','Af3','Ts3','Testo3','Username13','POST3'),

('RISP4','Af4','Ts4','Testo4','Username12','POST4'),

('RISP5','Af5','Ts5','Testo5','Username11','POST5'),

('RISP6','Af6','Ts6','Testo6','Username10','POST6'),

('RISP7','Af7','Ts7','Testo7','Username9','POST7'),

('RISP8','Af8','Ts8','Testo8','Username8','POST8'),

('RISP9','Af9','Ts9','Testo9','Username7','POST9'),

('RISP10','Af10','Ts10','Testo10','Username6','POST10'),

('RISP11','Af11','Ts11','Testo11','Username5','POST11'),

('RISP12','Af12','Ts12','Testo12','Username4','POST12'),

('RISP13','Af13','Ts13','Testo13','Username3','POST13'),

('RISP14','Af14','Ts14','Testo14','Username2','POST15'),

('RISP15','Af15','Ts15','Testo15','Username1','POST15');

-- ------------------------------------

-- Popolamento Giudizio

-- ------------------------------------

INSERT INTO Giudizio

VALUES ('GIU1','RISP1','Da Una Stella A 5 Stelle','3'),

('GIU2','RISP1','Da Una Stella A 5 Stelle','4'),

('GIU3','RISP1','Da Una Stella A 5 Stelle','4'),

('GIU4','RISP2','Da Una Stella A 5 Stelle','1'),

('GIU5','RISP2','Da Una Stella A 5 Stelle','1'),

('GIU6','RISP2','Da Una Stella A 5 Stelle','1'),

('GIU7','RISP5','Da Una Stella A 5 Stelle','2'),

('GIU8','RISP5','Da Una Stella A 5 Stelle','3'),

('GIU9','RISP5','Da Una Stella A 5 Stelle','3'),

('GIU10','RISP7','Da Una Stella A 5 Stelle','5'),

('GIU11','RISP7','Da Una Stella A 5 Stelle','5'),

('GIU12','RISP7','Da Una Stella A 5 Stelle','4'),

('GIU13','RISP11','Da Una Stella A 5 Stelle','5'),

('GIU14','RISP11','Da Una Stella A 5 Stelle','5'),

('GIU15','RISP11','Da Una Stella A 5 Stelle','5');

-- ------------------------------------

-- Popolamento Esterno

-- ------------------------------------

INSERT INTO Esterno

VALUES ('www.DiMunch1.com','DiMunch1.com','POST1'),

('www.DiMunch2.com','DiMunch2.com','POST3'),

('www.DiMunch3.com','DiMunch3.com','POST4'),

('www.DiMunch4.com','DiMunch4.com','POST5'),

('wwww.DiMunch5.com','DiMunch5.com','POST12'),

('wwww.DiMunch6.com','DiMunch6.com','POST11'),

('wwww.DiMunch7.com','DiMunch7.com','POST10');

-- ------------------------------------

-- Popolamento LavoraIn

-- ------------------------------------

INSERT INTO Lavora\_In

VALUES ('2','PDC01','Direttore'),

('3','ANF02','Direttore'),

('2','ENO03','Tutor'),

('3','GEM04','Tutor'),

('1','ENP05','Direttore'),

('2','MUP06','Responsabile'),

('6','JAR07','Direttore'),

('1','OLN08','Pulizie'),

('2','OLR09','Pulizie'),

('3','ROB10','Responsabile'),

('6','FAO11','Responsabile'),

('1','BIC12','Tutor'),

('7','TOR13','Responsabile'),

('1','PAC14','Tutor'),

('1','MAB15','Responsabile'),

('4','ALB16','Direttore'),

('4','ROP17','Pulizie'),

('4','OSR18','Responsabile'),

('5','MIL19','Direttore'),

('5','GIC20','Responsabile'),

('7','PAR21','Tutor');

-- ------------------------------------

-- Popolamento Frequenta

-- ------------------------------------

INSERT INTO Frequenta

VALUES ('3','CL14'),

('2','GB1'),

('2','VV2'),

('3','ED3'),

('3','FO4'),

('1','GI5'),

('1','AB6'),

('1','AL7'),

('6','LE8'),

('4','GC9'),

('6','PD10'),

('5','PC11'),

('5','EB12'),

('7','DL13'),

('2','CL14'),

('7','IZ15');

-- ------------------------------------

-- Popolamento Stipula

-- ------------------------------------

INSERT INTO Stipula

VALUES ('2','C01'),

('1','C02'),

('5','C03'),

('7','C04'),

('6','C05'),

('2','C06'),

('3','C06'),

('7','C07');

-- ------------------------------------

-- Popolamento Accede

-- ------------------------------------

INSERT INTO Accede

VALUES ('C01','P02'),

('C02','P01'),

('C03','P03'),

('C05','P06'),

('C06','P06');

-- ------------------------------------

-- Popolamento Gestisce

-- ------------------------------------

INSERT INTO Gestisce

VALUES ('MUP06','S02'),

('ROB10','S03'),

('FAO11','S06'),

('TOR13','S07'),

('OSR18','S04'),

('GIC20','S05');

-- ------------------------------------

-- Popolamento Accede\_1

-- ------------------------------------

INSERT INTO Accede\_1

VALUES ('C01','S02'),

('C02','S01'),

('C03','S05'),

('C04','S07'),

('C05','S06'),

('C06','S02'),

('C06','S03'),

('C07','S07');

-- ------------------------------------

-- Popolamento Accede\_2

-- ------------------------------------

INSERT INTO Accede\_2

VALUES ('C02','CORSO1'),

('C06','CORSO2'),

('C06','CORSO5'),

('C05','CORSO7');

-- ------------------------------------

-- Popolamento Partecipa

-- ------------------------------------

INSERT INTO Partecipa

VALUES ('ED3','CORSO1'),

('CL14','CORSO2'),

('CL14','CORSO5'),

('LE8','CORSO7');

-- ------------------------------------

-- Popolamento Contenuta

-- ------------------------------------

INSERT INTO Contenuta

VALUES ('SA1','D4'),

('SA2','D2'),

('SA3','D1'),

('SA4','D3'),

('SA5','D5'),

('SA6','D4'),

('SA7','D3');

-- ------------------------------------

-- Popolamento Acquista

-- ------------------------------------

INSERT INTO Acquista

VALUES ('ED3','INTE1','2018-02-15','20'),

('CL14','INTE2','2018-02-03','20'),

('CL14','INTE3','2018-03-18','25'),

('EB12','INTE5','2018-03-25','35'),

('DL13','INTE7','2018-01-25','15'),

('EB12','INTE8','2018-03-18','40'),

('EB12','INTE9','2018-03-14','55'),

('VV2','INTE10','2018-03-2','20'),

('VV2','INTE11','2018-03-5','18'),

('PD10','INTE4','2018-03-12','40');

-- ------------------------------------

-- Popolamento SiAssocia

-- ------------------------------------

INSERT INTO Si\_Associa

VALUES ('INTE1','D5'),

('INTE2','D4'),

('INTE3','D3'),

('INTE4','D4'),

('INTE5','D5'),

('INTE6','D2'),

('INTE7','D1');

-- ------------------------------------

-- Popolamento Aderisce

-- ------------------------------------

INSERT INTO Aderisce

VALUES ('Username1','CER5'),

('Username3','CER2'),

('Username4','CER4'),

('Username12','CER3'),

('Username6','CER5'),

('Username13','CER2'),

('Username11','CER4'),

('Username9','CER3'),

('Username8','CER5');

-- ------------------------------------

-- Popolamento Partecipa\_1

-- ------------------------------------

INSERT INTO Partecipa\_1

VALUES ('Username2','SF5','Proponente'),

('Username14','SF2','Proponente'),

('Username4','SF3','Proponente'),

('Username5','SF2','Proponente'),

('Username7','SF1','Proponente'),

('Username8','SF5','Partecipante'),

('Username3','SF4','Partecipante'),

('Username11','SF4','Partecipante'),

('Username13','SF3','Partecipante'),

('Username12','SF1','Partecipante');

-- ------------------------------------

-- Popolamento Nutre

-- ------------------------------------

INSERT INTO Nutre

VALUES ('CER1','INTR5'),

('CER2','INTR4'),

('CER3','INTR3'),

('CER4','INTR2'),

('CER5','INTR1');

-- ------------------------------------

-- Popolamento Fa\_Uso

-- ------------------------------------

INSERT INTO Fa\_Uso

VALUES ('POST2','SF5'),

('POST14','SF2'),

('POST4','SF3'),

('POST5','SF2'),

('POST7','SF1'),

('POST8','SF5'),

('POST3','SF4'),

('POST11','SF2'),

('POST13','SF3'),

('POST12','SF1');

**Trigger ed Event:**

-- Trigger che gestisce l'indice di massa corporea;

DROP TRIGGER IF EXISTS gestisci\_massa;

DELIMITER $$

CREATE TRIGGER gestisci\_massa

BEFORE INSERT ON SchedaAllenamento

FOR EACH ROW

BEGIN

SET @altezza = (NEW.Altezza / 100);

SET @indice = (NEW.Peso / (@altezza \* @altezza));

IF (@indice < 18.50) THEN

SET NEW.EntitaStato = 'Sottopeso';

ELSE IF (@indice >= 18.50 AND @indice <= 24.99) THEN

SET NEW.EntitaStato = 'Normopeso';

ELSE

SET NEW.EntitaStato = 'Sovrappeso';

END IF;

END IF;

END $$

DELIMITER ;

-- Trigger che gestisce l'indice di sforzo psichico;

DROP TRIGGER IF EXISTS gestisci\_sforzo;

DELIMITER $$

CREATE TRIGGER gestisci\_sforzo

BEFORE INSERT ON Valutazione

FOR EACH ROW

BEGIN

IF (NEW.BattitoCardiaco <= 120) THEN

SET NEW.SforzoPsichico = 'Basso';

ELSE IF (NEW.BattitoCardiaco >= 121 AND NEW.BattitoCardiaco <= 199) THEN

SET NEW.SforzoPsichico = 'Medio';

ELSE

SET NEW.SforzoPsichico = 'Alto';

END IF;

END IF;

END $$

DELIMITER ;

-- Trigger che aggiorna la ridondanza Durata presente nella tabella Accesso;

DROP TRIGGER IF EXISTS aggiorna\_durata;

DELIMITER $$

CREATE TRIGGER aggiorna\_durata

BEFORE INSERT ON Accesso

FOR EACH ROW

BEGIN

SET NEW.Durata = EXTRACT(HOUR FROM TIMEDIFF(NEW.OrarioUscita, NEW.OrarioIngresso)) \* 60

+ EXTRACT(MINUTE FROM TIMEDIFF(NEW.OrarioUscita, NEW.OrarioIngresso));

IF (NEW.Durata <= 60) THEN

SET NEW.Tariffa = 5;

ELSE IF (NEW. Durata > 60 AND NEW.Durata <= 120) THEN

SET NEW.Tariffa = 10;

ELSE

SET NEW.Tariffa = 12;

END IF;

END IF;

END $$

DELIMITER ;

-- Trigger che aggiorna la ridondanza ImportoAttuale presente nella tabella Pagamento;

DROP TRIGGER IF EXISTS aggiorna\_importo;

DELIMITER $$

CREATE TRIGGER aggiorna\_importo

BEFORE INSERT ON Rata

FOR EACH ROW

BEGIN

SET @attuale = (SELECT ImportoAttuale

FROM Pagamento

WHERE IdPagamento = NEW.Pagamento);

SET @totale = (SELECT ImportoTotale

FROM Pagamento

WHERE IdPagamento = NEW.Pagamento);

IF (@attuale + NEW.Importo <= @totale) THEN

UPDATE Pagamento

SET ImportoAttuale = @attuale + NEW.Importo

WHERE IdPagamento = NEW.Pagamento;

ELSE

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Limite superato!';

END IF;

END $$

DELIMITER ;

-- Trigger che aggiorna la ridondanza NumPartecipanti presente in Cerchia;

DROP TRIGGER IF EXISTS aggiorna\_partecipanti;

DELIMITER $$

CREATE TRIGGER aggiorna\_partecipanti

BEFORE INSERT ON Aderisce

FOR EACH ROW

BEGIN

SET @attuali = (SELECT COUNT(\*)

FROM Aderisce

WHERE Cerchia = NEW.Cerchia) + 2;

IF (@attuali > (SELECT MaxPartecipanti

FROM Cerchia

WHERE IdCerchia = NEW.Cerchia)) THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Limite superato!';

ELSE

UPDATE Cerchia

SET NumPartecipanti = @attuali

WHERE IdCerchia = NEW.Cerchia;

END IF;

END $$

DELIMITER ;

-- Event e Trigger che controllano lo stato di usura delle apparecchiature e ne curano la manutenzione;

DROP EVENT IF EXISTS usura\_apparecchiatura;

DROP TRIGGER IF EXISTS controllo\_manutenzione;

DROP EVENT IF EXISTS fine\_manutenzione;

DELIMITER $$

CREATE EVENT usura\_apparecchiatura

ON SCHEDULE EVERY 4 WEEK

DO

BEGIN

UPDATE Apparecchiatura

SET Manutenzione = Manutenzione + 5;

END $$

CREATE TRIGGER controllo\_manutenzione

BEFORE UPDATE ON Apparecchiatura

FOR EACH ROW

BEGIN

SET @contatore = (SELECT COUNT(\*)

FROM Apparecchiatura

WHERE NEW.Manutenzione = 100);

IF (@contatore <> 0) THEN

SET NEW.Manutenzione = 'SI';

END IF;

END $$

CREATE EVENT fine\_manutenzione

ON SCHEDULE EVERY 1 WEEK

DO

BEGIN

UPDATE Apparecchiatura

SET Manutenzione = 'NO' AND PercentualeUsura = 0

WHERE Manutenzione = 'SI';

END $$

DELIMITER ;

-- Trigger che gestisce il vincolo di integrità generico relativo agli orari dei corsi ed all'apertura del centro fitness;

DROP TRIGGER IF EXISTS corso\_centro;

DELIMITER $$

CREATE TRIGGER corso\_centro

BEFORE INSERT ON Calendario

FOR EACH ROW

BEGIN

SET @orari\_sbagliati = (SELECT COUNT(\*)

FROM Calendario CA INNER JOIN Corso C ON NEW.Corso = C.IdCorso

INNER JOIN Sala S ON C.Sala = S.IdSala

INNER JOIN CentroFitness CE ON S.CentroFitness = CE.IdCentro

INNER JOIN Oraria O ON O.CentroFitness = CE.IdCentro

WHERE NEW.OrarioInizio < O.OrarioApertura

OR NEW.OrarioFine > O.OrarioChiusura

OR NEW.OrarioInizio > O.OrarioChiusura

OR NEW.OrarioFine < O.OrarioApertura);

SET @orari\_sbagliati = (SELECT COUNT(\*)

FROM Calendario CA INNER JOIN Corso C ON NEW.Corso = C.IdCorso

INNER JOIN Piscina P ON C.Piscina = P.IdPiscina

INNER JOIN CentroFitness CE ON P.CentroFitness = CE.IdCentro

INNER JOIN Oraria O ON O.CentroFitness = CE.IdCentro

WHERE NEW.OrarioInizio < O.OrarioApertura

OR NEW.OrarioFine > O.OrarioChiusura

OR NEW.OrarioInizio > O.OrarioChiusura

OR NEW.OrarioFine < O.OrarioApertura);

IF ((@orari\_sbagliati <> 0) OR (NEW.OrarioFine < NEW.OrarioInizio)) THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Orario non conforme alle norme!';

END IF;

END $$

DELIMITER ;

-- Trigger che gestisce il vincolo di integrità generico sugli orari dei dipendenti;

DROP TRIGGER IF EXISTS orari\_personale;

DELIMITER $$

CREATE TRIGGER orari\_personale

BEFORE INSERT ON Personale

FOR EACH ROW

BEGIN

IF (EXTRACT(HOUR FROM TIMEDIFF(NEW.OrarioUscita, NEW.OrarioIngresso)) > 8) THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Superato il limite di 8 ore al giorno!';

END IF;

END $$

DELIMITER ;

DROP TRIGGER IF EXISTS orari\_corsi;

DELIMITER $$

CREATE TRIGGER orari\_corsi

BEFORE INSERT ON Calendario

FOR EACH ROW

BEGIN

SET @verifico = (SELECT COUNT(\*)

FROM Calendario CA INNER JOIN Corso C ON NEW.Corso = C.IdCorso

WHERE C.Dipendente IN (Select C1.Dipendente

FROM Calendario CA1

INNER JOIN Corso C1 ON CA1.Corso = C1.IdCorso

WHERE CA.Giorno = CA1.Giorno

AND ((NEW.OrarioInizio BETWEEN CA1.OrarioInizio

AND CA1.OrarioFine)

OR (NEW.OrarioFine BETWEEN CA1.OrarioInizio

AND CA1.OrarioFine)

OR (NEW.OrarioInizio BETWEEN CA1.OrarioInizio

AND CA1.OrarioFine

AND NEW.OrarioFine BETWEEN CA1.OrarioInizio

AND CA1.OrarioFine))));

IF (@verifico <> 0) THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Un dipendente non puo'''' tenere due corsi contemporaneamente!';

END IF;

END $$

DELIMITER ;

**Operazione 1:**

DROP PROCEDURE IF EXISTS operazione\_1;

DELIMITER $$

CREATE PROCEDURE operazione\_1 (IN id\_centro INT,

IN data DATE,

OUT durata\_media INT)

BEGIN

SELECT AVG(Durata) INTO durata\_media

FROM Accesso

WHERE CentroFitness = id\_centro

AND Data = data;

END $$

DELIMITER ;

**Operazione 2:**

DROP PROCEDURE IF EXISTS operazione\_2;

DELIMITER $$

CREATE PROCEDURE operazione\_2 (IN codfiscale VARCHAR(16),

OUT importo\_mancante INT)

BEGIN

SELECT (ImportoTotale - ImportoAttuale) INTO importo\_mancante

FROM Pagamento P INNER JOIN Contratto C ON P.Contratto = C.IdContratto

WHERE codfiscale = C.Cliente;

END $$

DELIMITER ;

**Operazione 3:**

DROP PROCEDURE IF EXISTS operazione\_3;

DELIMITER $$

CREATE PROCEDURE operazione\_3 (IN cerchia VARCHAR(16),

OUT utenti\_mancanti INT)

BEGIN

SELECT (MaxPartecipanti - NumPartecipanti) INTO utenti\_mancanti

FROM Cerchia

WHERE IdCerchia = cerchia;

END $$

DELIMITER ;

**Operazione 4:**

DROP PROCEDURE IF EXISTS operazione\_4;

DELIMITER $$

CREATE PROCEDURE operazione\_4 (IN sostanza VARCHAR(16),

OUT nome\_integratore VARCHAR(20))

BEGIN

SELECT I.NomeCommerciale INTO nome\_integratore

FROM Integratore I INNER JOIN Acquista A ON I.IdIntegratore = A.Integratore

WHERE I.NomeSostanza = sostanza

GROUP BY A.Integratore

HAVING SUM(A.Quantita) >= ALL (SELECT SUM(A1.Quantita)

FROM Integratore I1 INNER JOIN Acquista A1

ON I1.IdIntegratore = A1.Integratore

WHERE I1.NomeSostanza = sostanza

GROUP BY A1.Integratore);

END $$

DELIMITER ;

**Operazione 5:**

DROP PROCEDURE IF EXISTS operazione\_5;

DELIMITER $$

CREATE PROCEDURE operazione\_5(IN codfiscale VARCHAR(16),

IN codaccesso VARCHAR(6),

IN data DATE,

IN ingresso TIME,

IN uscita TIME,

IN smartwatch VARCHAR(5),

IN centro INT,

IN armadietto VARCHAR(6))

BEGIN

DECLARE EXIT HANDLER FOR 1062

SELECT 'Duplicate entry for key PRIMARY';

INSERT INTO Accesso

VALUES (codaccesso,data,ingresso,uscita,NULL,smartwatch,NULL,centro,armadietto,codfiscale);

END $$

DELIMITER ;

**Operazione 6:**

DROP PROCEDURE IF EXISTS operazione\_6;

DELIMITER $$

CREATE PROCEDURE operazione\_6(IN codfiscale VARCHAR(16),

IN idrata VARCHAR(6),

IN importo INT,

IN data DATE,

IN istituto VARCHAR(15),

IN tasso INT)

BEGIN

DECLARE EXIT HANDLER FOR 1062

SELECT 'Duplicate entry for key PRIMARY';

SET @pagamento = (SELECT DISTINCT P.IdPagamento

FROM Rata R INNER JOIN Pagamento P ON R.Pagamento = P.IdPagamento

INNER JOIN Contratto C ON C.IdContratto = P.Contratto

WHERE C.Cliente = codfiscale);

INSERT INTO Rata

VALUES (idrata,istituto,importo,data,tasso,@pagamento);

END $$

DELIMITER ;

**Operazione 7:**

DROP PROCEDURE IF EXISTS operazione\_7;

DELIMITER $$

CREATE PROCEDURE operazione\_7(IN utente VARCHAR(16),

IN cerchia VARCHAR(6))

BEGIN

DECLARE EXIT HANDLER FOR 1062

SELECT 'Duplicate entry for key PRIMARY';

INSERT INTO Aderisce

VALUES (utente,cerchia);

END $$

DELIMITER ;

**Operazione 8:**

DROP PROCEDURE IF EXISTS operazione\_8;

DELIMITER $$

CREATE PROCEDURE operazione\_8(IN utente VARCHAR(16),

IN sfida VARCHAR(6),

IN idval VARCHAR(6),

IN kcal INT,

IN battito INT,

IN ossigeno INT,

IN data DATE)

BEGIN

DECLARE EXIT HANDLER FOR 1062

SELECT 'Duplicate entry for key PRIMARY';

INSERT INTO Valutazione

VALUES (idval,kcal,battito,ossigeno,NULL,data,utente,sfida);

END $$

DELIMITER ;

**Area Analytics**

**Reporting:**

-- Creazione tabella Tempi\_Attesa;

DROP TABLE IF EXISTS Tempi\_Attesa;

CREATE TABLE Tempi\_Attesa(

IdAttesa INT AUTO\_INCREMENT,

CentroFitness INT NOT NULL,

Apparecchiatura VARCHAR(50) NOT NULL,

Attesa INT,

PRIMARY KEY(IdAttesa)

) ENGINE = INNODB DEFAULT CHARSET = Latin1;

-- Trigger che popola la tabella Tempi\_Attesa;

DROP TRIGGER IF EXISTS popola\_tempi\_attesa;

DELIMITER $$

CREATE TRIGGER popola\_tempi\_attesa

AFTER INSERT ON Sessione

FOR EACH ROW

BEGIN

SET @attesa = (SELECT (EXTRACT(HOUR FROM TIMEDIFF(NEW.IstanteInizio, E.IstanteInizio)) \* 60

+ EXTRACT(MINUTE FROM TIMEDIFF(NEW.IstanteInizio, E.IstanteInizio)))

FROM Esercizio E

WHERE IdEsercizio = NEW.Esercizio);

SET @centro = (SELECT A.CentroFitness

FROM Apparecchiatura A INNER JOIN RICHIEDE R

ON A.IdApparecchiatura = R.Apparecchiatura

INNER JOIN Esercizio E ON R.Esercizio = E.IdEsercizio

WHERE E.IdEsercizio = NEW.Esercizio);

SET @apparecchiatura = (SELECT A.Nome

FROM Apparecchiatura A INNER JOIN RICHIEDE R

ON A.IdApparecchiatura = R.Apparecchiatura

INNER JOIN Esercizio E ON R.Esercizio = E.IdEsercizio

WHERE E.IdEsercizio = NEW.Esercizio);

INSERT INTO Tempi\_Attesa (CentroFitness, Apparecchiatura, Attesa)

VALUES (@centro, @apparecchiatura, @attesa);

END $$

DELIMITER ;

-- Procedura che mostra le apparecchiature da comprare;

DROP PROCEDURE IF EXISTS necessita\_apparecchiature;

DELIMITER $$

CREATE PROCEDURE necessita\_apparecchiature(IN centro INT)

BEGIN

SELECT Apparecchiatura

FROM Tempi\_Attesa

WHERE CentroFitness = centro

GROUP BY Apparecchiatura, CentroFitness

HAVING AVG(Attesa) > 10;

END $$

DELIMITER ;

-- Creazione tabella Tot\_Accessi;

DROP TABLE IF EXISTS Tot\_Accessi;

CREATE TABLE Tot\_Accessi(

CentroFitness INT,

Mattina INT DEFAULT 0,

Pranzo INT DEFAULT 0,

Pomeriggio INT DEFAULT 0,

PRIMARY KEY(CentroFitness)

) ENGINE = INNODB DEFAULT CHARSET = Latin1;

-- Trigger che popola Tot\_Accessi;

DROP TRIGGER IF EXISTS popola\_tot\_accessi;

DELIMITER $$

CREATE TRIGGER popola\_tot\_accessi

BEFORE INSERT ON Accesso

FOR EACH ROW

BEGIN

SET @ver = (SELECT COUNT(\*)

FROM Tot\_Accessi

WHERE NEW.CentroFitness = CentroFitness);

IF (@ver = 0) THEN

INSERT INTO Tot\_Accessi (CentroFitness)

VALUES (NEW.CentroFitness);

END IF;

IF (NEW.OrarioIngresso BETWEEN '09:00:00' AND '12:59:59') THEN

UPDATE Tot\_Accessi

SET Mattina = Mattina + 1

WHERE CentroFitness = NEW.CentroFitness;

ELSE IF (NEW.OrarioIngresso BETWEEN '13:00:00' AND '14:59:59') THEN

UPDATE Tot\_Accessi

SET Pranzo = Pranzo + 1

WHERE CentroFitness = NEW.CentroFitness;

ELSE

UPDATE Tot\_Accessi

SET Pomeriggio = Pomeriggio + 1

WHERE CentroFitness = NEW.CentroFitness;

END IF;

END IF;

END $$

DELIMITER ;

**Performance Sportiva:**

-- Creazione tabella Performance\_Sportiva;

DROP TABLE IF EXISTS Performance\_Sportiva;

CREATE TABLE Performance\_Sportiva(

Sessione VARCHAR(10) NOT NULL,

Esercizio VARCHAR(10) NOT NULL,

Diff\_Tempo INT,

Diff\_Ripetizioni INT,

PRIMARY KEY(Sessione, Esercizio)

) ENGINE = INNODB DEFAULT CHARSET = Latin1;

-- Trigger che popola Performance\_Sportiva;

DROP TRIGGER IF EXISTS popola\_performance;

DELIMITER $$

CREATE TRIGGER popola\_performance

AFTER INSERT ON Sessione

FOR EACH ROW

BEGIN

IF (NEW.RipetizioneEffettive IS NOT NULL) THEN

SET @diff\_rip = (SELECT (E.NRipetizioni - NEW.RipetizioneEffettive)

FROM Esercizio E

WHERE E.IdEsercizio = NEW.Esercizio);

SET @diff\_durata = NULL;

ELSE

SET @diff\_rip = NULL;

SET @diff\_durata = (SELECT (E.Durata - (EXTRACT(HOUR FROM TIMEDIFF(NEW.IstanteFine,

NEW.IstanteInizio)) \* 60 + EXTRACT(MINUTE FROM

TIMEDIFF(NEW.IstanteFine,NEW.IstanteInizio))))

FROM Esercizio E

WHERE E.IdEsercizio = NEW.Esercizio);

END IF;

INSERT INTO Performance\_Sportiva

VALUES (NEW.IdSessione, NEW.Esercizio, @diff\_durata, @diff\_rip);

END $$

DELIMITER ;

-- Stored\_procedure che dato un cliente restituisce l'intensa della scheda di allenamento;

DROP PROCEDURE IF EXISTS verifica\_scheda;

DELIMITER $$

CREATE PROCEDURE verifica\_scheda(IN cliente VARCHAR(16))

BEGIN

SET @correttezza = (SELECT COUNT(\*)

FROM Performance\_Sportiva PS INNER JOIN Esercizio E

ON PS.Esercizio = E.IdEsercizio

INNER JOIN SchedaAllenamento SA

ON E.SchedaAllenamento = SA.IdSchedaAllenamento

WHERE SA.Cliente = cliente);

IF (@correttezza = 0) THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Questo cliente non è presente nel centro o non possiede una scheda di allenamento';

END IF;

SET @rip = (SELECT AVG(PS.Diff\_Ripetizioni)

FROM Performance\_Sportiva PS INNER JOIN Esercizio E ON PS.Esercizio = E.IdEsercizio

INNER JOIN SchedaAllenamento SA

ON E.SchedaAllenamento = SA.IdSchedaAllenamento

WHERE SA.Cliente = cliente

AND PS.Diff\_Ripetizioni IS NOT NULL);

SET @durata = (SELECT AVG(PS.Diff\_Tempo)

FROM Performance\_Sportiva PS INNER JOIN Esercizio E

ON PS.Esercizio = E.IdEsercizio

INNER JOIN SchedaAllenamento SA

ON E.SchedaAllenamento = SA.IdSchedaAllenamento

WHERE SA.Cliente = cliente

AND PS.Diff\_Tempo IS NOT NULL);

IF (@rip >= 3 OR @durata >= 15) THEN

SELECT 'La scheda di allenamento associata al cliente risulta troppo pesante';

ELSE IF (@rip = 0 OR @durata = 0) THEN

SELECT 'La scheda di allenamento associata al cliente risulta troppo leggera';

ELSE IF (@rip <> 0 OR @durata <> 0) THEN

SELECT 'La scheda di allenamento associata al cliente risulta corretta';

END IF;

END IF;

END IF;

END $$

DELIMITER ;

**Rotazione del Magazzino:**

-- Alter table che inserisce nella tabella integratore la percentuale di sconto ed il relativo prezzo;

ALTER TABLE Integratore

ADD Sconto INT DEFAULT 0,

ADD Prezzo INT DEFAULT 0;

-- Inserimento dei prezzi all'interno della tabella Integratore;

UPDATE Integratore

SET Prezzo = 15

WHERE IdIntegratore = 'INTE1';

UPDATE Integratore

SET Prezzo = 10

WHERE IdIntegratore = 'INTE2';

UPDATE Integratore

SET Prezzo = 20

WHERE IdIntegratore = 'INTE3';

UPDATE Integratore

SET Prezzo = 18

WHERE IdIntegratore = 'INTE4';

UPDATE Integratore

SET Prezzo = 20

WHERE IdIntegratore = 'INTE5';

UPDATE Integratore

SET Prezzo = 15

WHERE IdIntegratore = 'INTE6';

UPDATE Integratore

SET Prezzo = 10

WHERE IdIntegratore = 'INTE7';

UPDATE Integratore

SET Prezzo = 12

WHERE IdIntegratore = 'INTE8';

UPDATE Integratore

SET Prezzo = 13

WHERE IdIntegratore = 'INTE9';

UPDATE Integratore

SET Prezzo = 18

WHERE IdIntegratore = 'INTE10';

UPDATE Integratore

SET Prezzo = 35

WHERE IdIntegratore = 'INTE11';

--

Event che sconta settimanalmente gli integratori che si avvicinano alla scadenza;

DROP EVENT IF EXISTS aggiorna\_sconto;

DELIMITER $$

CREATE EVENT aggiorna\_sconto

ON SCHEDULE EVERY 1 WEEK

DO

BEGIN

UPDATE Integratore

SET Sconto = '10' AND Prezzo = (Prezzo \* 90) / 100

WHERE DATEDIFF(current\_date, DataScadenza) BETWEEN 31 AND 15;

UPDATE Integratore

SET Sconto = '15' AND Prezzo = (Prezzo \* 85) / 100

WHERE DATEDIFF(current\_date, DataScadenza) BETWEEN 14 AND 8;

UPDATE Integratore

SET Sconto = '20' AND Prezzo = (Prezzo \* 90) / 100

WHERE DATEDIFF(current\_date, DataScadenza) <= 8;

END $$

DELIMITER ;