

ARIANNA AGRESTA, MICHELE DI RENZO

**PROGETTAZIONE DI UN
DATABASE PER LA
GESTIONE DI UNA
PALESTRA**

2022

Progetto di Basi di Dati,
Corso di laurea di Ingegneria Informatica e dell'Automazione,
Anno Accademico 2021/2022,
Università Politecnica delle Marche.

Indice

1 ANALISI DEI REQUISITI	5
1.1 Raccolta informazioni con intervista al titolare	5
1.2 Raccolta informazioni con intervista alla segretaria	7
1.3 Raccolta informazioni con intervista al personal trainer	8
1.4 Raccolta informazioni(Modulistica)	9
1.5 Raccolta informazioni(Analisi processi interni)	11
1.6 Requisiti espressi in linguaggio naturale	12
1.7 Glossario dei termini	13
1.8 Ambiguità di terminologie	16
1.9 Specifica delle operazioni	16
2 PROGETTAZIONE CONCETTUALE	18
2.1 Spiegazione della modalità di procedimento	18
2.2 Identificazione delle entità e delle relazioni fondamentali	19
2.3 Scheletro dello schema iniziale	21
2.4 Sviluppo delle componenti	22
2.4.1 Cliente	22
2.4.2 Abbonamento	23
2.4.3 Prenotazione	24
2.4.4 Dipendente	25
2.4.5 Transazione	26
2.4.6 Scheda tecnica	27
2.4.7 Corso	27
2.5 Schema E-R	28
2.6 Analisi della qualità del modello E-R	30
2.7 Dizionario dei dati	30
2.7.1 Entità	30
2.7.2 Relazioni	34
2.8 Regole aziendali	35
2.8.1 Regole di vincolo	35
2.8.2 Regole di derivazione	37
3 PROGETTAZIONE LOGICA	38
3.1 Tavole dei volumi e delle operazioni	38
3.1.1 Tavola dei volumi	38
3.1.2 Tavola delle operazioni	40
3.2 Ristrutturazione dello schema concettuale	41
3.2.1 Analisi derivazioni e ridondanze	41
3.2.2 Eliminazione delle gerarchie	44
3.3 Partizionamento-accorpamento di concetti	48
3.3.1 Eliminazione degli attributi multivalore	48
3.3.2 Accorpamento relazioni	49

3.3.3	Schema E-R ristrutturato	50
3.4	Elenco degli identificatori principali	52
3.5	Normalizzazione	53
3.6	Traduzione verso il modello relazionale	54
4	Codifica SQL e testing	55
4.1	Implementazione delle tabelle	56
4.2	Codifica delle operazioni	60

1 ANALISI DEI REQUISITI

Questa prima parte della progettazione è funzionale alla ricerca di informazioni utili per la creazione di una base di dati per la gestione di una palestra, in modo tale che il database in questione possa soddisfare gli obiettivi principali per il funzionamento efficace dell'attività presa in considerazione. Perciò il primo passo da noi svolto è quello di intervistare il titolare e alcuni dipendenti della palestra presa come spunto per lo svolgimento del progetto, al fine di comprendere a pieno le problematiche che il nostro database deve essere in grado di risolvere. In totale, sono state svolte tre interviste a persone interne alla palestra, in modo tale da avere una visione globale ed avere la possibilità di prendere in considerazione diversi punti di vista. A tale scopo, le persone che si sono rese disponibili al colloquio sono il titolare e due dipendenti, ossia la segretaria e un personal trainer. Sottolineiamo che per una questione di privacy, il nome della palestra e gli intervistati preferiscono rimanere anonimi.

1.1 Raccolta informazioni con intervista al titolare

Team: "Salve, facciamo parte del team di sviluppo che avevate contattato nei giorni scorsi per una progettazione di una base di dati per la vostra attività."

Committente: "Salve, si accomodi pure."

Team: "La ringrazio... dunque, mi parli della sua attività e cosa svolgete, così da capire meglio di cosa ci andremo ad occupare."

Committente: "Dunque, siamo una palestra di medie dimensioni che lavora dal lunedì al venerdì dalle ore 8:00 alle ore 23:00 e il sabato dalle ore 8:00 alle ore 12:30; contiamo 11 dipendenti tra cui istruttori, allenatori e collaboratori. Nella nostra palestra c'è la possibilità per un cliente di svolgere sia attività fisica nella sala pesi con le opportune attrezzature, sia di svolgere diversi corsi."

Team: "La ringraziamo delle informazioni riguardanti la sua attività. Possiamo chiederle fino ad ora come ha tenuto traccia della gestione interna e come mai sente la necessità di dover digitalizzare l'amministrazione?"

Committente: "Fino ad ora non abbiamo avuto la necessità di gestire informaticamente la nostra attività, ma vista la diffusione del Covid-19, riteniamo che sia necessario avere un accesso più veloce ai dati dei nostri clienti, evitando inutili code all'ingresso della palestra per il controllo degli abbonamenti, dei certificati e degli accessi giornalieri alla nostra struttura. Perciò, abbiamo ritenuto opportuno contattarvi per una soluzione informatica a questo problema che fino ad ora abbiamo gestito a livello cartaceo."

Team: "Va bene, comprendiamo la necessità. Volevamo chiederle anche se a livello legislativo siete sottoposti a delle limitazioni dovute al Covid."

Committente: "Ah sì, vi ringrazio della domanda perché vista l'estensio-

ne in metri quadri del nostro stabile, la legge ci impone di poter accogliere al massimo 40 persone contemporaneamente all'interno della palestra, corsi compresi. Preciso che nel conteggio dei clienti contemporaneamente presenti all'interno dell'edificio, sono esclusi i dipendenti."

Team: "Riguardo ciò, come eravate organizzati?"

Committente: "Fino ad ora avevamo organizzato fasce orarie da 90 min, nelle quali ogni cliente aveva il tempo necessario di svolgere il proprio allenamento/corso e scaduti questi sarebbero così potuti entrare altri con le stesse modalità."

Team: "Risulta quindi che serva un prodotto limitato alla sola gestione della clientela? Se così non fosse, ci illustri quali altre situazioni si vogliano gestire. Grazie."

Committente: "A dire il vero, oltre alla gestione dei clienti, sarebbe utile che il prodotto da voi sviluppato possa anche tenere conto della gestione del personale e della cassa dell'attività, relativa esclusivamente alle entrate che si conteggiano grazie ai pagamenti degli abbonamenti o ingressi singoli."

Team: "Mi scusi se la interrompo, potrebbe mostrarmi quello che è ad oggi il registro dei clienti dell' attività? In maniera tale che possiamo renderci conto dei requisiti che il nostro database dovrà soddisfare. Grazie."

-Ci viene mostrato il registro e ci viene illustrato.-

Committente: "Ecco a voi. Vi spiego meglio di cosa si tratta. Noi registriamo i clienti sul registro tramite la loro anagrafica e ne indichiamo la tipologia di abbonamento o ingresso giornaliero e/o l'iscrizione a determinati corsi che offre la nostra palestra."

Team: "Perfetto, ci parli anche dei corsi, come vengono gestiti e come sono organizzati."

Committente: "Non le so dire con precisione quanti e quali corsi sono svolti nella palestra, per cui le lascio il recapito della mia segretaria che ne è sicuramente al corrente."

Team: "La ringraziamo per averci fornito il contatto, sicuramente ci sarà utile svolgere un colloquio più approfondito. Invece per quanto riguarda gli abbonamenti e/o ingressi singoli, come sono strutturate le tariffe e le relative scadenze?"

Committente: "Mi dispiace, ma anche su questo argomento è necessario che vi confrontiate con la segretaria. Non so darvi informazioni a riguardo."

Team: "Al momento queste informazioni possono essere necessarie per farci un'idea generale del lavoro che si dovrà svolgere, e magari torneremo prossimamente per parlare direttamente con chi si occupa dell'amministrazione vera e propria della clientela, del personale e della cassa così da svolgere ulteriori approfondimenti per realizzare un prodotto che soddisfi le vostre esigenze in maniera più efficiente e completa possibile."

Committente: "Vi ringrazio. Per ulteriori dubbi potete contattarmi telefonicamente se fosse necessario."

Team: "Grazie a lei per la disponibilità al colloquio. Ci terremo in contatto

con lei, arrivederci."

Committente: "Arrivederci."

1.2 Raccolta informazioni con intervista alla segretaria

Contattando telefonicamente la segretaria grazie al numero che ci è stato previamente fornito dal proprietario, abbiamo trovato un accordo per un colloquio di persona per discutere delle questioni amministrative relative all'attività, che nella prima intervista non siamo riusciti ad ottenere.

Team: "Salve buongiorno, è lei la segretaria del signor X?"

Segretaria: "Buongiorno, sì sono io, piacere di conoservi."

Team: "Il piacere è tutto nostro. Il signor X ci ha indicato lei come la persona con cui svolgere un'analisi più approfondita per la realizzazione della base di dati che si vorrà implementare per la digitalizzazione della gestione dell'attività."

Segretaria: "Benissimo, sono a vostra completa disposizione, chiedete pure."

Team: "Parlando della gestione dei clienti il signor X ci ha mostrato il registro cartaceo di cui siete in possesso e di come questo è organizzato. Volevamo chiederle quali e quante sono le tariffe per gli abbonamenti che i clienti possono sottoscrivere nella vostra palestra."

Segretaria: "Per quanto riguarda gli abbonamenti disponiamo di 3 tariffe per la sala pesi, quali una tariffa mensile al costo di 40€, una tariffa trimestrale al costo di 100€ ed una tariffa annuale al costo di 320€. Inoltre, è possibile anche il pagamento per l'ingresso singolo al costo di 7€. Per quanto riguarda invece i corsi di cui il signor X sicuramente vi ha parlato, disponiamo di un'unica tariffa mensile al costo di 35€ per corso, al termine della quale il cliente dovrà rinnovare per proseguire il corso. Inoltre, la nostra palestra offre la possibilità di interrompere un abbonamento di qualsiasi tipo per al massimo un mese, senza cancellare il profilo. Nel caso in cui l'abbonamento non venga rinnovato, il cliente viene eliminato dal registro cartaceo."

Team: "Si come ha detto lei il signor X ha citato l'esistenza dei corsi ma non ci ha saputo dire nel particolare quali e quanti sono. Ci sa dire qualcosa in più al riguardo? Grazie."

Segretaria: "Sì, nella nostra palestra svolgiamo corsi di zumba, pilates, crossfit, military training e pugilato. In particolare vi sono 2 aree, o meglio sale dedicate allo svolgimento di queste 5 attività che come ben capite non possono essere svolte contemporaneamente. Ad oggi siamo così organizzati:"

-Ci viene mostrato il calendario delle attività-

Team: "Perfetto. Per quanto riguarda la gestione dei dipendenti invece come siete organizzati?"

Segretaria: "Ovviamente ogni dipendente, me compresa, compie turni di 8 ore al massimo giornaliere. Il nostro staff è composto da 5 istruttori, uno per corso, da 5 personal trainer e dalla sottoscritta. Questo è quanto c'è da sapere riguardo il personale."

Team: "Come ultima cosa, ci eravamo appuntati che sul registro dei clienti, in particolare a coloro che erano iscritti alla sala pesi vi era associata una scheda tecnica, è lei che se ne occupa?"

Segretaria: "Purtroppo non so dirvi nulla al riguardo, poiché di questo se ne occupano i nostri personal trainer."

Team: "Perfetto, torneremo prossimamente per parlare con loro dopo esserci fatti ulteriori idee sul da farsi. È stato un piacere lavorare con lei, arrivederci e grazie. Buon lavoro e grazie del suo tempo."

Segretaria: "Grazie a voi, buon lavoro. Arrivederci."

1.3 Raccolta informazioni con intervista al personal trainer

Una volta terminata l'intervista con la segretaria, siamo riusciti ad accordarci con uno dei personal trainer per un piccolo colloquio, in modo da raccogliere informazioni sulle schede tecniche dei clienti ed avere tutta la documentazione necessaria per iniziare la nostra progettazione.

Personal trainer: "Buongiorno piacere di conoscervi."

Team: "Buongiorno a lei, piacere nostro."

Personal trainer: "Come posso esservi utile?"

Team: "La signorina Y, la segretaria, ci ha indicato di parlare con lei per quanto riguarda la gestione delle schede tecniche dei clienti, in quanto è di nostro interesse per far sì che il prodotto che vi verrà fornito sia completo in tutte le sue funzionalità."

Personal trainer: "Perfetto, vi spiego subito. Ad ogni cliente che si iscrive in sala pesi viene assegnata da noi PT una scheda tecnica contenente tutti i vari esercizi da compiere nell'allenamento corrente. Ovviamente questa scheda non sarà per sempre uguale, ma si dovrà avere la possibilità di svolgere delle modifiche su di essa, in quanto noi istruttori terremo conto dei progressi che il cliente compie durante il percorso. La modifica viene effettuata mensilmente, per tutti i possessori di un qualsiasi abbonamento alla sala pesi, poiché scientificamente un mese di allenamento è sufficiente per compiere dei progressi e quindi aggiornare la propria scheda."

Team: "Invece per quanto riguarda i clienti che sottoscrivono l'ingresso giornaliero è prevista qualche tipologia di scheda personalizzata?"

Personal trainer: "Purtroppo non siamo tenuti a sottoscrivere alcun tipo di scheda tecnica per clienti che non possiedono abbonamenti alla sala pesi, poiché non possiamo monitorare i loro progressi mensili."

Team: "Quindi sono previste schede di questo tipo anche per i clienti in

possesso dell'abbonamento speciale che combina sala pesi e corsi?"

Personal trainer: "Sì, assolutamente. Anche loro sono in possesso di schede personalizzate."

Team: "Perfetto, dovremmo avere tutte le informazioni necessarie. La ringraziamo per la disponibilità a questo colloquio. Arrivederci e buon lavoro."

Personal trainer: "Arrivederci e buon lavoro a voi."

1.4 Raccolta informazioni(Modulistica)

Durante l'intervista al titolare, alla segretaria e al personal trainer ci sono stati mostrati dei documenti che abbiamo ritenuto importanti da inserire all'interno della progettazione, al fine di comprendere meglio i processi interni. In particolare, ci sono stati forniti gli orari dei corsi, che ci permettono di analizzare più attentamente le fasce orarie che il database deve prendere in considerazione nella fase di prenotazione dei clienti, la tabella delle tariffe relative agli abbonamenti e agli ingressi giornalieri ed un documento fac-simile della scheda dei dati di ogni cliente.

ORARI E CORSI

	LUNEDÌ		MARTEDÌ		MERCOLEDÌ		GIOVEDÌ		VENERDÌ		SABATO	
	Sala 1	Sala 2	Sala 1	Sala 2	Sala 1	Sala 2	Sala 1	Sala 2	Sala 1	Sala 2	Sala 1	Sala 2
8:00-9:30	Zumba				Pilates	Zumba					Crossfit	
9:30-11:00		Pilates									Zumba	Pilates
11:00-12:30			Crossfit						Crossfit	Military T.		
12:30-14:00												
14:00-15:30	Military T.						Military T.					
15:30-17:00												
17:00-18:30				Pugilato								
18:30-20:00	Pilates				Zumba	Pilates		Pugilato			Pugilato	
20:00-21:30		Zumba	Crossfit				Crossfit			Military T.		
21:30-23:00	Military T.				Military T.							

Figura 1: Orari e corsi

ABBONAMENTI E TARIFFE

GIORNALIERO (SALA PESI)	7€
MENSILE (SALA PESI)	40€
TRIMESTRALE (SALA PESI)	100€
ANNUALE (SALA PESI)	320€
CORSO (MENSILE)	35€

Figura 2: Abbonamenti e tariffe

Scheda cliente (ingresso giornaliero)	
Codice cliente:	1
Nome:	Michele
Cognome:	Di Renzo
Cod. fiscale:	XXX
Certificato medico:	Valido
Abbonamento:	Giornaliero
Pagamento:	Effettuato
Corso frequentato:	Sala pesi

Scheda cliente (Abb. Mensile/trimestrale/Annuale)	
Codice cliente:	2
Nome:	Arianna
Cognome:	Agresta
Cod. fiscale:	YYY
Certificato medico:	Valido
Abbonamento:	Mensile
Pagamento:	Effettuato
Corso frequentato:	Sala pesi

Figura 3: Fac-simile scheda dati cliente

1.5 Raccolta informazioni(Analisi processi interni)

Dopo aver raccolto i dati relativi alle interviste e alla documentazione a noi fornita, abbiamo ritenuto necessario schematizzare graficamente il flusso interno dei processi, per renderci conto in quale modo ogni azione effettuata all'interno dell'attività ne influenzi un'altra.

A partire dall'ingresso del cliente nello stabile, è possibile notare in che modo le varie aree interagiscono l'una con l'altra. In particolare, si può osservare la divisione dello schema in quattro settori principali.

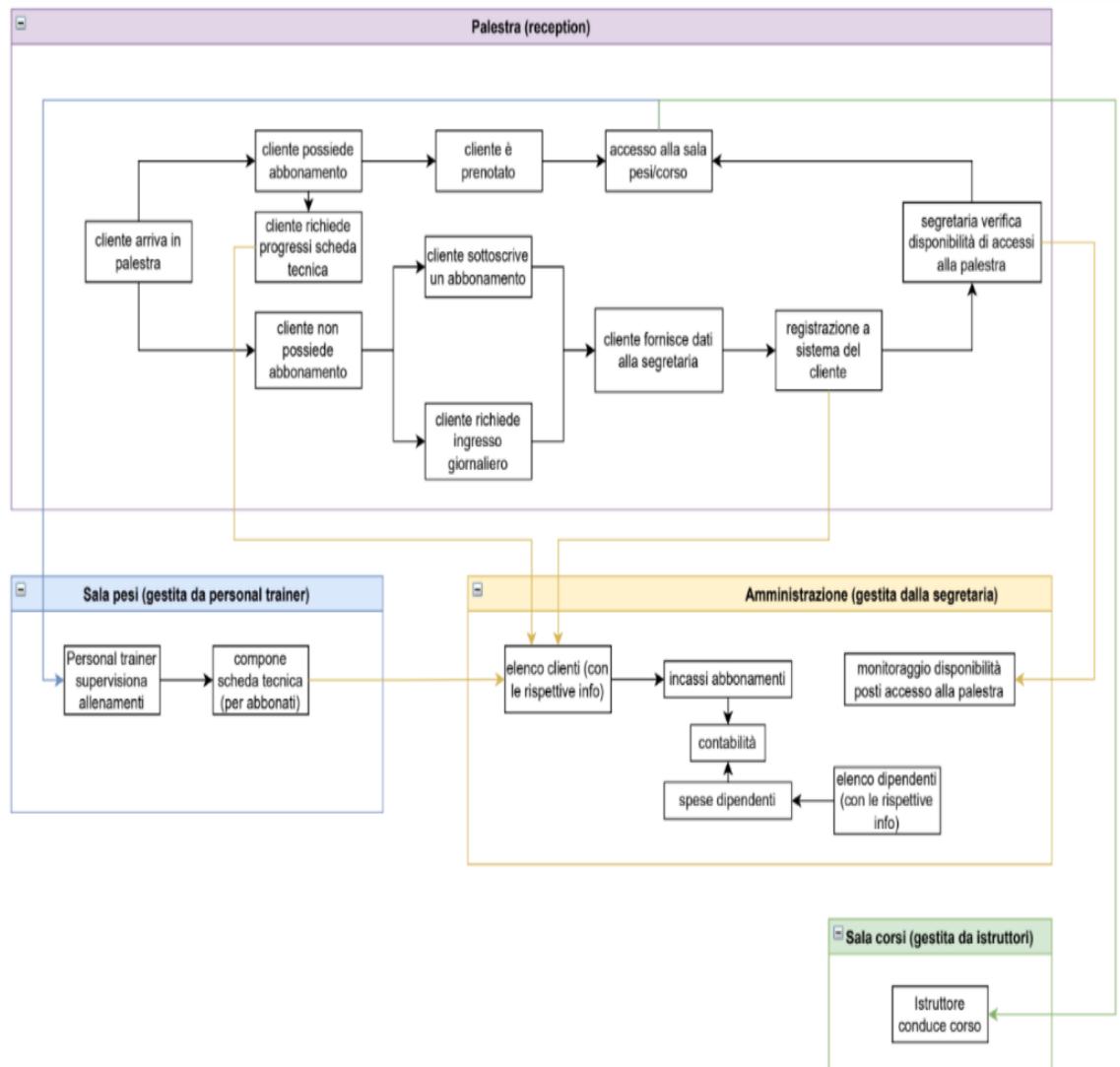


Figura 4: Analisi di azioni e processi interni

1.6 Requisiti espressi in linguaggio naturale

Dopo aver effettuato le dovute interviste e, di conseguenza, essere a conoscenza delle attività e dell'organizzazione interna, è stata possibile la stesura degli obiettivi che la base di dati sarà in grado di raggiungere correttamente e in maniera efficiente.

Il progetto consiste nella realizzazione di un sistema informativo per la gestione di una **palestra**, aperta dal lunedì al venerdì dalle ore 8:00 alle ore 23:00 e il sabato dalle ore 8:00 alle ore 12:30 escluse le festività. È costituita da una **sala pesi** completamente allestita e da 2 aree dedicate allo svolgimento dei **corsi** di zumba, pilates, military training, crossfit e pugilato.

I **dipendenti** del centro sportivo sono attualmente 11 (organizzati in una segretaria, 5 personal trainer e 5 istruttori [uno per corso]). I lavoratori verranno registrati nel sistema tramite la loro **anagrafica** e il ruolo ricoperto all'interno dell'organizzazione lavorativa. I personal trainer sono coloro che si occupano di seguire gli allenamenti dei **clienti** nella sala pesi, gli istruttori, invece, si occuperanno dello svolgimento dei corsi e la segretaria è colei che all'ingresso della palestra ha il compito di dare informazioni utili ai clienti e si occupa della loro gestione.

Per quanto riguarda i corsi suddetti, sono 5, e ogni corso viene svolto due volte nel giorno prestabilito (una volta al mattino e una volta al pomeriggio), eccetto il pugilato. Ogni corso è coordinato da un istruttore differente e può accogliere al più 20 iscritti per volta, per una più facile gestione del corso e in primis per le norme anti-covid vigenti. Ovviamente il cliente che partecipa al corso a cui è iscritto la mattina, non potrà presentarsi il giorno stesso nel pomeriggio per seguire quello stesso corso.

Per quanto riguarda i clienti, essi saranno memorizzati a sistema, sia che facciano un ingresso giornaliero, sia se sottoscrivono un **abbonamento** per la sala pesi o per un determinato corso. La loro memorizzazione consiste nel registrare nel database l'anagrafica, il certificato di idoneità sportiva (senza il quale non si potrà prendere parte a nessuna attività all'interno della palestra) e per l'appunto se hanno sottoscritto un abbonamento o un **ingresso singolo** e verrà gestito l'avvenuto pagamento di questi ultimi. Ad ogni abbonamento è assegnato un **codice univoco**, generato direttamente dal sistema stesso. Inoltre, verrà memorizzata anche la **scheda tecnica** per i soli possessori di un abbonamento alla sala pesi (e non ingresso singolo). Ogni cliente per partecipare ad una sessione d'allenamento, in un determinato giorno e ad una determinata **fascia oraria**, deve prima prenotarsi nel sistema tramite la segretaria. In questo modo il sistema sarà in grado di registrare gli **ingressi** e le **uscite** di ogni singolo cliente della palestra e mo-

nitorare quindi la capienza interna ad ogni fascia oraria.

Per quanto riguarda la gestione della **cassa**, di cui si occupa la segretaria, verranno gestite le transazioni in entrata relative alle sottoscrizioni degli abbonamenti e degli ingressi giornalieri da parte dei clienti, dunque i relativi **pagamenti** per ogni cliente, e le transazioni in uscita riguardanti la devoluzione degli **emolumenti** dei dipendenti. Ogni transazione sarà memorizzata nel sistema con un codice di fatturazione, la data in cui questa avviene, il tipo (entrata o uscita) e il segretario che ha registrato la transazione. La gestione della cassa è utile al bilancio dell'azienda poiché si potrà accedere al saldo per osservare la situazione finanziaria della palestra. Per l'abbonamento sono previste cinque diverse durate e tariffe. Si potranno effettuare abbonamenti di tipo giornaliero, mensile, trimestrale, annuale e/o corso. Scaduti i termini, questi possono essere rinnovati.

Ciascun cliente che ne sottoscriverà uno (almeno) mensile avrà il diritto di possedere una scheda tecnica gestita da un esperto dipendente della palestra (un personal trainer) nella quale verranno riportati tutti gli esercizi di cui si compone l'allenamento.

1.7 Glossario dei termini

Questa sezione è relativa alla descrizione più accurata di alcuni termini comparsi in 1.6, che necessitano un'analisi più accurata al fine di evitare ambiguità.

I termini riportati nella tabella del glossario sono evidenziati nella precedente sezione in grassetto (Esempio: **palestra**)

TERMINI	DESCRIZIONE	SINONIMI	COLLEGAMENTI
Palestra	Edificio con attrezzi adatti al fitness e con sale adibite a corsi	Attività, stabile, edificio, struttura	Clienti, Sala pesi, Corsi, Dipendenti
Sala pesi	Area adibita agli esercizi a corpo libero	Area pesi	Dipendenti, Clienti
Corsi	Tipologie di lezioni offerte dalla palestra	Pratica di zumba, pilates, crossfit, military training o pugilato	Dipendenti, Clienti
Dipendenti	Persona fisica che opera all'interno dell'attività e riceve una retribuzione monetaria. Nella palestra si dividono in istruttori, personal trainer e segretaria	Personale, stuff	Emolumenti, Sala pesi, Corsi, Anagrafica
Anagrafica	Documentazione relativa al cliente che comprende i dati personali significativi al riconoscimento		Clienti, Dipendenti
Clienti	Persona fisica che usufruisce del servizio offerto dalla palestra	Persone	Abbonamento, Corsi, Sala pesi, Pagamenti, Anagrafica
Abbonamento	Usufruire dei servizi della palestra senza dover pagare ad ogni ingresso per un periodo uguale a quello per cui si è versata la somma di denaro (mensile/trimestrale/annuale) e/o l'iscrizione relativa ai corsi della palestra		Clienti, Sala pesi, Corsi

Ingresso singolo	Usufruire dei servizi della palestra con pagamento ad ogni ingresso nella struttura	Ingresso giornaliero, accesso giornaliero, accesso singolo	Clienti, Abbonamento
Codice univoco	Codice numerico che identifica univocamente un abbonato	ID	Abbonamento
Scheda tecnica	Scheda personalizzata in base alle necessità di allenamento di un cliente della palestra	Scheda personalizzata, scheda d'allenamento	Clienti, Dipendenti
Fascia oraria	Quantità di tempo a disposizione di un cliente per svolgere l'allenamento all'interno della struttura (limite imposto dalle restrizioni per il Covid-19)	Tempo di allenamento	Clienti, Abbonamento
Ingressi/uscite	Flusso in entrata/uscita di clienti dall'edificio	Entrate/uscite	Palestra, Clienti, Sala pesi, Corso
Cassa	Flusso monetario in entrata basato sulla somma dei pagamenti di abbonamenti/ingressi giornalieri		Pagamenti, Emolumenti
Pagamenti	Processo di saldo del conto da parte di un cliente in base alla tipologia di abbonamento o ingresso giornaliero	Transazione monetaria, saldo	Cassa, Clienti
Emolumenti	Retribuzione mensile corrisposta agli impiegati come compenso per il lavoro svolto, in base alle ore lavorative accumulate	Stipendio	Cassa, Dipendenti

1.8 Ambiguità di terminologie

Attraverso il glossario dei termini nella precedente sezione sono state evidenziate tutte le precisazioni del caso. Conviene comunque specificare la differenza tra **ingressi singoli**(nella riga 8 della tabella) e **ingressi** (nella riga 12 della tabella). La prima espressione si riferisce ad una modalità di accesso alla struttura senza abbonamento di alcun tipo, con pagamento all'entrata prima dell'allenamento del cliente. Il secondo termine invece è riferito all'accesso del cliente in generale nella palestra, sia esso in possesso di abbonamento oppure sia con modalità di accesso giornaliero.

Ci è sembrato necessario sottolineare la differenza tra queste espressioni per evitare incomprensioni durante il resto della fase progettuale.

1.9 Specifica delle operazioni

Di seguito vengono elencate le operazioni eseguibili nella base di dati:

1. Inserimento nuovo cliente
2. Inserimento nuovo dipendente
3. Inserimento della scheda tecnica di un cliente
4. Inserimento transazione
5. Inserimento nuovo corso
6. Inserimento abbonamento di un cliente
7. Aggiornamento stato abbonamento di un cliente già esistente
8. Aggiornamento saldo della cassa
9. Aggiornamento contatore dei clienti prenotati ad una determinata fascia oraria di un determinato giorno
10. Modifica dati di un cliente
11. Modifica dati di un dipendente
12. Modifica dati transazione
13. Eliminazione di un cliente
14. Eliminazione di un dipendente
15. Eliminazione di un corso
16. Eliminazione di una prenotazione per Id

17. Visualizzazione profilo di un cliente
18. Visualizzazione profilo di un dipendente
19. Visualizzazione abbonamento in corso di validità di un cliente
20. Visualizzazione scheda tecnica di un cliente
21. Conto delle prenotazioni di una determinata fascia orario in un determinato giorno
22. Visualizzazione stato di un abbonamento
23. Visualizza dettagli di una transazione
24. Statistiche (miglioramento/peggioramento) massa grassa
25. Statistiche (miglioramento/peggioramento) massa muscolare
26. Filtraggio schede tecniche per altezza e peso
27. Numero di abbonamenti che un cliente possiede
28. Filtraggio schede tecniche per nome, cognome
29. Filtraggio clienti per nome, cognome
30. Filtraggio abbonamenti per nome, cognome
31. Filtraggio dipendenti per nome, cognome
32. Filtraggio dipendenti per ruolo
33. Conteggio clienti entrati in palestra in una giornata
34. Nome e cognome clienti abbonati validi
35. Conteggiare il numero di abbonamenti sottoscritti il mese corrente
36. Conteggiare il numero di abbonamenti sottoscritti in un anno (gennaio-dicembre)
37. Filtraggio abbonamenti per tipo di abbonamento
38. Filtraggio di nome, cognome dei clienti che non sono in possesso del certificato medico
39. Inserimento prenotazione
40. Inserimento numero di telefono
41. Elimina numero di telefono

42. Inserimento nuovo esercizio
43. Elimina esercizio
44. Trovare il numero di telefono associato a un cliente per nome, cognome
45. Fare una classifica degli esercizi più svolti
46. Aggiungere esercizio in una scheda tecnica
47. Inserimento segretaria
48. Inserimento personal trainer
49. Inserimento istruttore

Non è stato trattato il tema relativo all'orario giornaliero dei dipendenti in quanto saranno muniti di un badge e tramite la scansione di quest'ultimo verrà monitorata la loro attività lavorativa.

2 PROGETTAZIONE CONCETTUALE

2.1 Spiegazione della modalità di procedimento

Dopo aver effettuato un'analisi preliminare grazie all'intervista e all'analisi dei processi interni (come raffigurato nel punto 1.1.e della prima parte della progettazione), abbiamo avuto la possibilità di formulare delle idee riguardanti le problematiche da risolvere.

Per l'implementazione del sistema, abbiamo deciso di utilizzare due strategie di progettazione, ovvero gli approcci bottom-up e top-down: in questo modo abbiamo avuto la possibilità di sfruttare i vantaggi di entrambi.

Ricordiamo che la strategia bottom-up specifica in dettaglio parti individuali, connettendole poi tra loro in modo tale da formare componenti più grandi a loro volta interconnesse tra loro, così da creare un sistema completo; la strategia top-down formula inizialmente una visione generale del sistema, descrivendo la finalità principale senza scendere nel dettaglio delle sue parti, che poi saranno successivamente rifinite aggiungendo maggiori dettagli.

In sintesi, la nostra modalità di procedimento è stata:

- analisi primitiva delle informazioni raccolte, dei requisiti e delle operazioni in modo da comprendere e individuare le entità principali;
- stesura dello schema concettuale, in modo da individuare le relazioni principali;

- applicazione dell'approccio top-down, formulando una visione d'insieme in modo tale da scendere poi nel dettaglio delle parti;
- applicazione della strategia bottom-up, per interconnettere le parti individuali in modo tale da creare uno schema concettuale che rispecchi le caratteristiche principali che esso deve presentare.

Si ricorda che uno schema concettuale di qualità, necessario per il passaggio alla progettazione logica, deve rispettare:

- **correttezza**: deve utilizzare nel giusto modo i costrutti messi a disposizione del modello;
- **completezza**: deve rappresentare tutti i requisiti sui dati e tutte le transazioni devono essere eseguite a partire dallo schema;
- **minimalità**: non deve contenere concetti superflui o ridondanti;
- **leggibilità**: deve essere uno schema comprensibile.

2.2 Identificazione delle entità e delle relazioni fondamentali

Dopo aver analizzato con attenzione le informazioni raccolte nella parte preliminare della progettazione, abbiamo deciso di individuare delle categorie più grandi dalle quali partire per analizzare i dati a disposizione. In particolare, abbiamo selezionato quattro sezioni che coinvolgono la maggior quantità di dati possibile.

La sezione Cliente è quella che permette di raggruppare tutto ciò che riguarda i suoi dati anagrafici.

L'area Abbonamento è relativa alla presenza effettiva di una tipologia di abbonamento o meno e il relativo costo.

Il blocco Transazione è quello che ingloba ciò che riguarda i pagamenti in entrata da parte dei clienti e il pagamento in uscita degli stipendi per i dipendenti.

La parte Dipendente rappresenta tramite l'anagrafica i componenti dello staff della palestra, il ruolo che ricoprono, dell'emolumento.

Grazie a tale divisione in varie sezioni è stato possibile creare uno schema iniziale, nel quale abbiamo individuato altre entità da mettere in relazione con le macro-sezioni inizialmente designate.

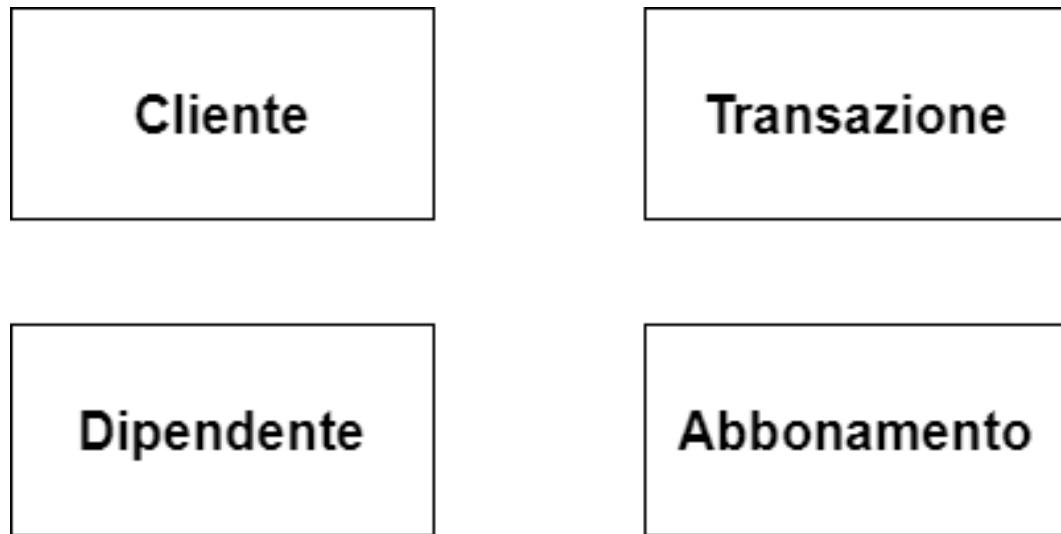


Figura 5: Prime entità

2.3 Scheletro dello schema iniziale

Lo schema concettuale iniziale da noi sviluppato, è un sommario delle entità principali individuate e le relazioni che intercorrono tra le tali, le quali in seguito verranno implementate ed ampliate tramite l'utilizzo della strategia top-down.

In particolare, il Cliente per partecipare ad una qualsiasi sessione di allenamento, deve essere in possesso di un Abbonamento (del tipo giornaliero, mensile, trimestrale, annuale o corso) e deve aver compiuto la Prenotazione alla fascia oraria nella quale si vuole allenare. Gli abbonamenti sono gestiti dalla segretaria, ovvero una dei Dipendenti, che gestisce anche le Transazioni (sia in entrata che in uscita), che vengono registrati nella Cassa, e, ovviamente, si occupa anche del controllo delle prenotazioni degli accessi in palestra. I dipendenti, inoltre, si occuperanno di comporre una Scheda tecnica (per i clienti abbonati alla sala pesi) che ogni possessore, richiedendolo in segreteria, potrà visualizzare. Infine, un'ultima entità è Corso, la quale risulta distaccata dalle altre e entra in relazione solo con una tipologia specifica di dipendenti che gestisce ciò che riguarda i corsi offerti ai clienti abbonati a tali attività.

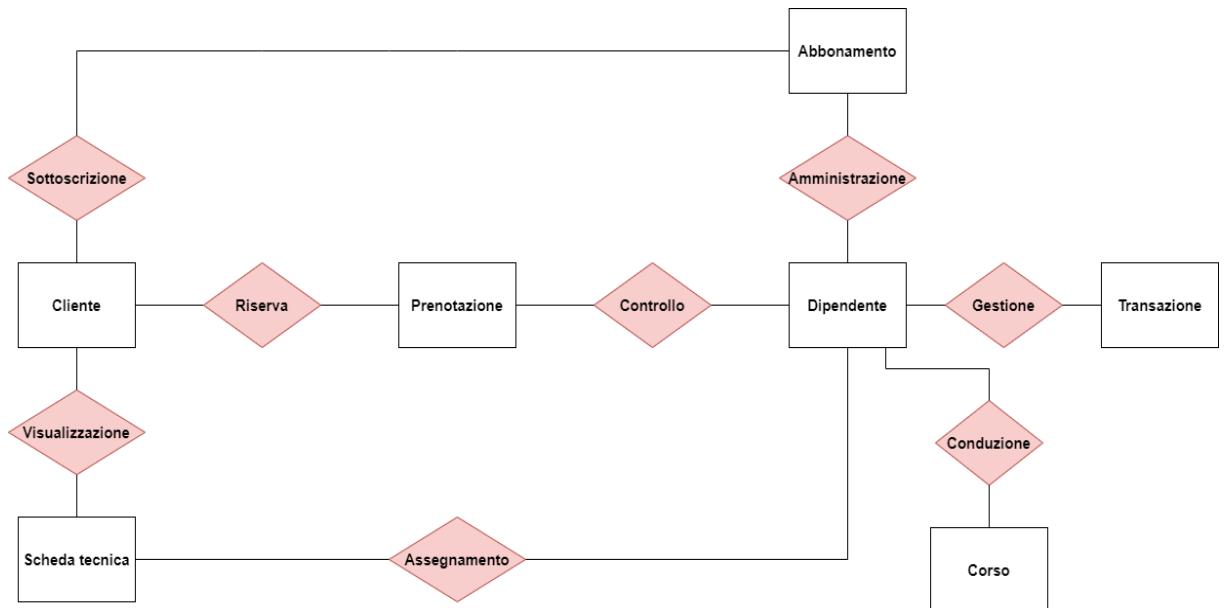


Figura 6: Scheletro dello schema

2.4 Sviluppo delle componenti

Dopo aver schematizzato in linea di massima le entità e le relazioni principali, abbiamo iniziato lo sviluppo dello schema concettuale attraverso la strategia top-down. In questo modo, abbiamo sviluppato le caratteristiche e le componenti di ogni singola entità principale individuata. Inoltre, ci teniamo ad appuntare che le generalizzazioni sono state indicate con una freccia a corpo bianco e punta nera, come per convenzione.

2.4.1 Cliente

La prima entità da prendere in considerazione è quella relativa al Cliente, ovvero la persona fisica che usufruisce del servizio offerto dalla palestra, come descritto nel **Glossario dei termini** (sezione 1.3). Il cliente si suddivide in tre tipologie, ovvero il cliente con abbonamento alla sala pesi, il cliente con abbonamento al corso e il cliente con l'ingresso giornaliero. È stata ritenuta da noi necessaria una generalizzazione dell'entità poiché per ogni tipologia di cliente sono necessarie le registrazioni nel database di dati anagrafici (nome, cognome, codice fiscale, numero di telefono), del possesso di un certificato medico e di un codice cliente specifico. Per ulteriori chiarimenti, si consiglia di consultare il **Dizionario dei dati** nella sezione 2.7, nel paragrafo relativo alle **Entità**.

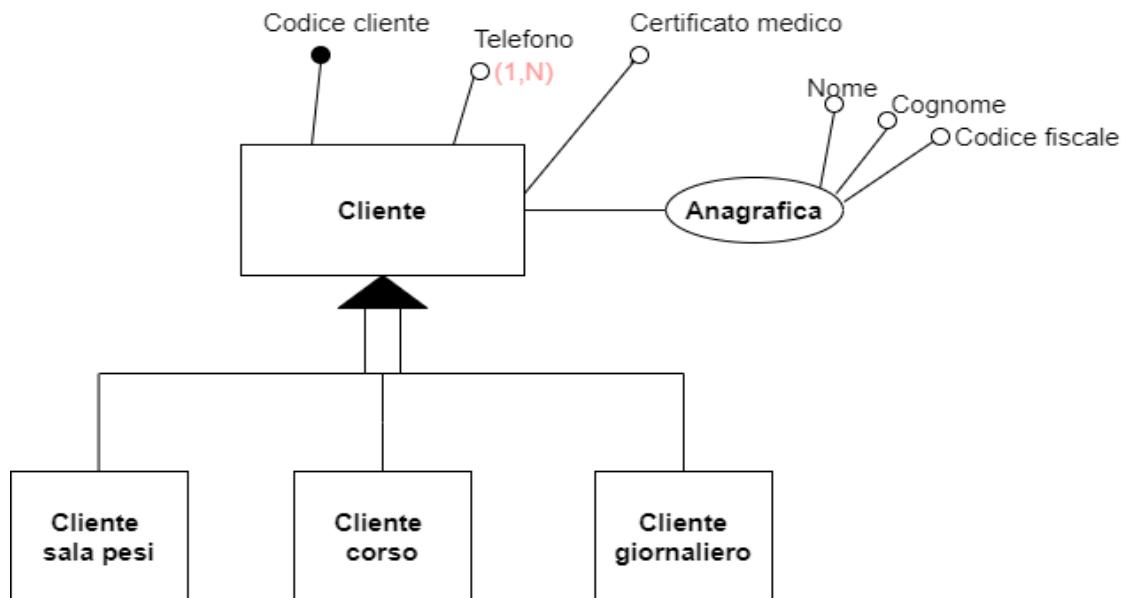


Figura 7: Entità Cliente

2.4.2 Abbonamento

L'entità Abbonamento è un raggruppamento riferito al modo in cui il cliente è in relazione con la palestra, cioè come usufruisce dei servizi offerti dall'attività per un periodo uguale a quello per cui è versata la somma di denaro. Ogni abbonamento è caratterizzato da dati relativi al codice identificativo, alla data di emissione iniziale dell'abbonamento, alla data di scadenza (quindi la relativa data di possibile rinnovo), al costo e allo stato di validità dell'abbonamento. In particolare, questa entità risulta essere una generalizzazione delle varie tipologie di abbonamenti che la palestra offre al cliente. Si consiglia di osservare le tali dallo schema in questa sezione e di consultare il **Dizionario dei dati** nella sezione 2.7, nel paragrafo relativo alle **Entità**.

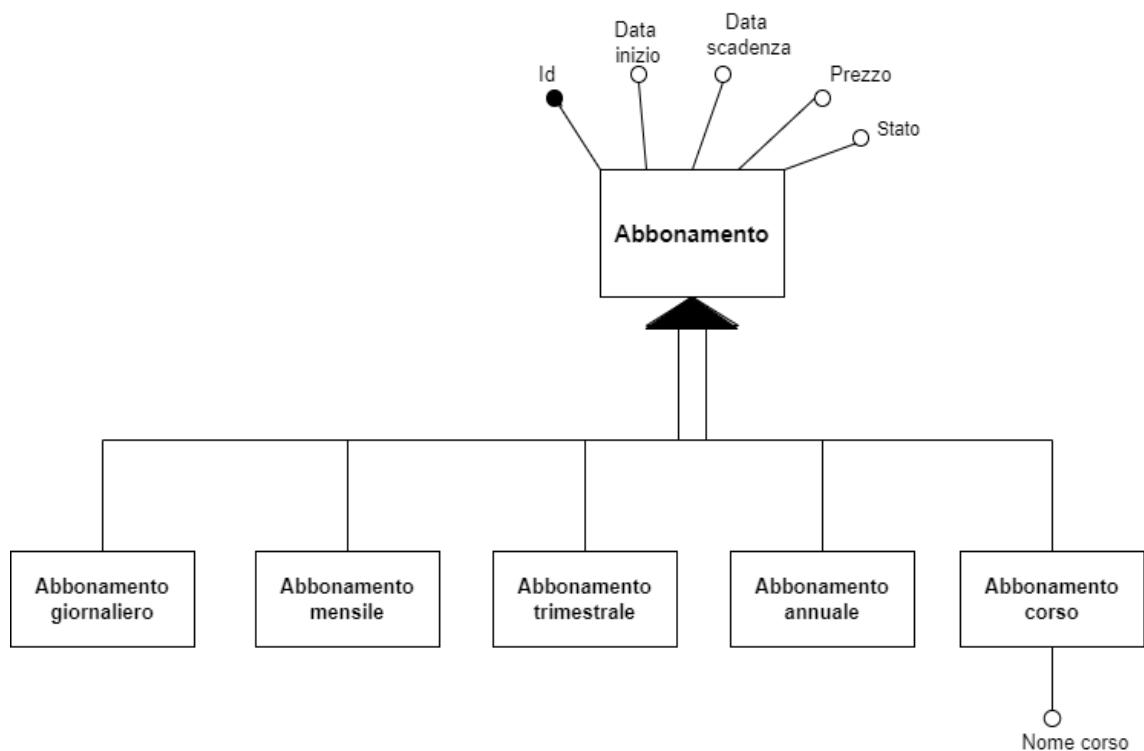


Figura 8: Entità Abbonamento

2.4.3 Prenotazione

Tale entità è riferita all'insieme dei dati relativi allo storico delle prenotazioni effettuate dai clienti all'interno della palestra per svolgere l'attività motoria. In particolare, si nota che Prenotazione si divide in due diverse tipologie, ossia la prenotazione relativa alla sala pesi e quella relativa al corso. Ogni tipologia, comunque, avrà come attributi relativi il codice Id della prenotazione, la data, l'orario di inizio e l'orario di fine prenotazione. Si ricorda che questi ultimi due attributi dovranno sempre avere una distanza esattamente uguale a 90 minuti, per le normative vigenti all'interno dell'attività. Per ulteriori informazioni sugli attributi, si consulti la sezione **2.7** relativa al **Dizionario dei dati** nella sezione **Entità**.

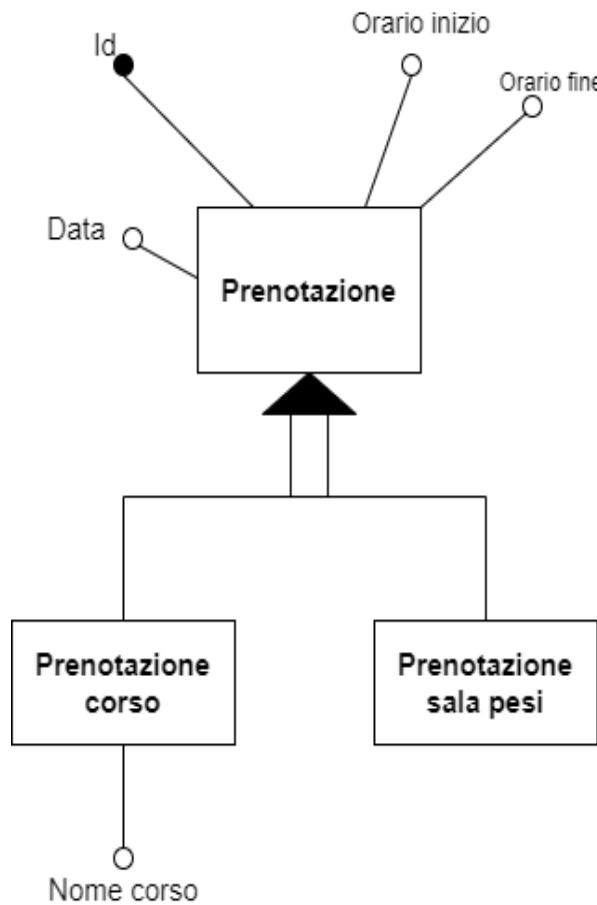


Figura 9: Entità Prenotazione

2.4.4 Dipendente

Il Dipendente è riferito alla persona fisica che opera all'interno dell'attività e riceve una retribuzione monetaria. Perciò possiamo considerare tale entità come una generalizzazione delle tre diverse tipologie di dipendenti che sono presenti all'interno della palestra. Per ogni tipologia di lavoratore si registrano su database il codice fiscale, il nome, il cognome, la data di nascita (in giorno, mese e anno) e lo stipendio. Si consiglia di visitare la sezione **2.7** relativa al **Dizionario dei dati** per ulteriori chiarimenti a riguardo, in particolare nella parte relativa alle **Entità**.

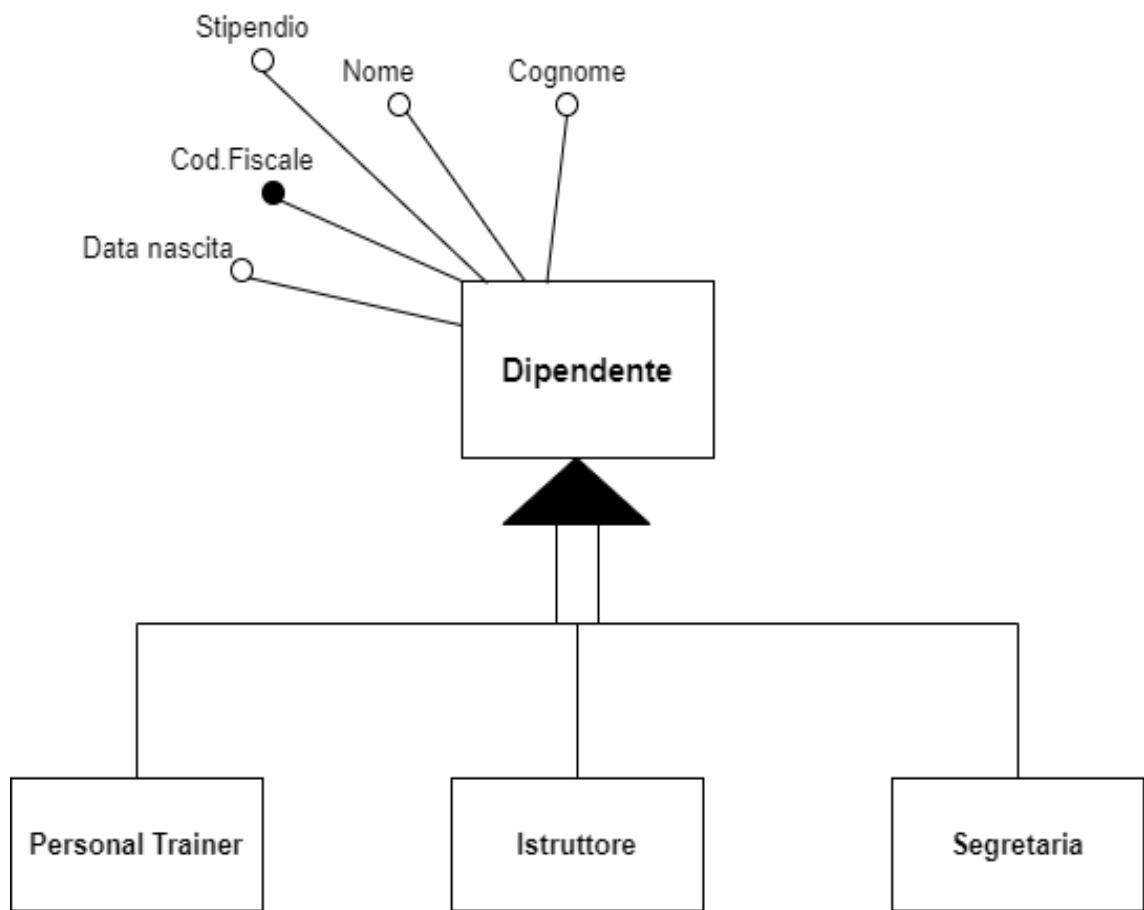


Figura 10: Entità Dipendente

2.4.5 Transazione

Con Transazione si identifica ogni processo di saldo del conto da parte di un cliente oppure ad una transazione in uscita per devolvere lo stipendio a un dipendente. Ogni tipologia di saldo sarà identificata da un codice identificativo, dalla causale e dalla data di emissione del pagamento. Proprio per distinguere le tipologie di transazioni, cioè in entrata e in uscita, abbiamo ritenuta necessaria la generalizzazione dell'entità Transazione. Per ulteriori informazioni, consultare il **Dizionario dei dati (2.7)**, nel paragrafo **Entità**.

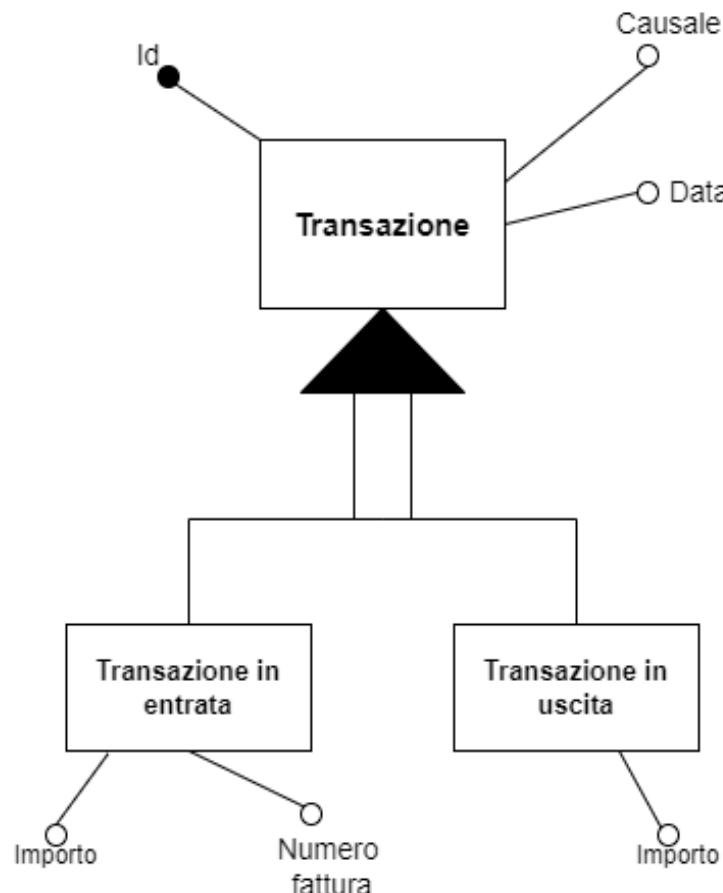


Figura 11: Entità Transazione

2.4.6 Scheda tecnica

La Scheda tecnica è quell'entità che racchiude le informazioni relative alle schede personalizzate in base alle necessità di allenamento di un cliente della palestra. Ogni scheda di questo tipo è caratterizzata da un codice ID, dalla data di creazione (per il monitoraggio a seconda dell'abbonamento in possesso del cliente e del rinnovo di questa) e dalle caratteristiche fisiche (peso, altezza, percentuale massa grassa e muscolare). In ogni caso, si consulti il **Dizionario dei dati (2.7)** nella sezione **Entità**.

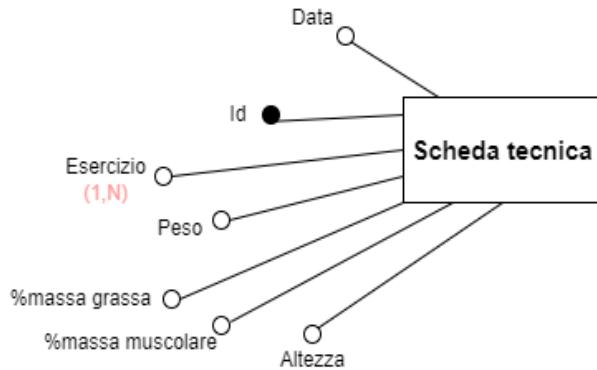


Figura 12: Entità Scheda tecnica

2.4.7 Corso

Questa è un'entità che entra in relazione solamente con “Istruttore” dell'entità “Dipendente”, poiché tale tipologia di dipendente è l'unica che dirige le sale adibite ai corsi offerti dalla palestra ai clienti abbonati ad essi. In particolare, dell'entità Corso si registrano nel database l'attributo relativo al nome dell'attività che si svolge. Per ulteriori informazioni, si consiglia la visione del **Dizionario dei dati** in **2.7** nella sezione **Entità**.

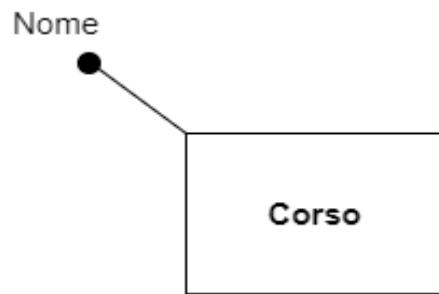


Figura 13: Entità Corso

2.5 Schema E-R

Lo schema alla fine di questa sottosezione è quello relativo al modello E-R complessivo. È stato ottenuto dalla composizione dei singoli diagrammi sviluppati in precedenza a questo opportunamente legati tra loro, e in seguito, tramite la strategia Bottom-up, abbiamo reso più complete ed esaustive le singole componenti.

Notiamo che il cliente è una delle figure centrali dell'attività presa in considerazione. Egli sarà in possesso di un abbonamento che gli permetterà di svolgere l'attività fisica desiderata previa prenotazione in fasce orarie prestabilite e una volta al giorno.

L'abbonamento, che può essere giornaliero, mensile, trimestrale, annuale e/o mensile al corso, è amministrato dalla segretaria, che ne gestisce il pagamento da parte del cliente.

I dipendenti, a seconda del loro ruolo, condurranno un corso se sono istruttori, o seguiranno, tramite una scheda tecnica, i clienti della palestra se sono personal trainer. La segretaria ha dei compiti di controllo e amministrazione.

La scheda tecnica assegnata ai clienti, mette in risalto, oltre che l'allenamento da svolgere, il peso, l'altezza, la massa grassa e la massa muscolare che verranno aggiornati ad ogni inizio di una nuova scheda per poter così tracciare i progressi compiuti.

È da osservare che la scheda tecnica è assegnata ai clienti che posseggono un abbonamento alla sala pesi dalla durata di almeno un mese (i clienti giornalieri non ne saranno in possesso).

Le transazioni, infine, sono gestite attraverso un flusso in ingresso di soldi con i saldi degli abbonamenti e con l'uscita dovuta al pagamento degli stipendi per i dipendenti.

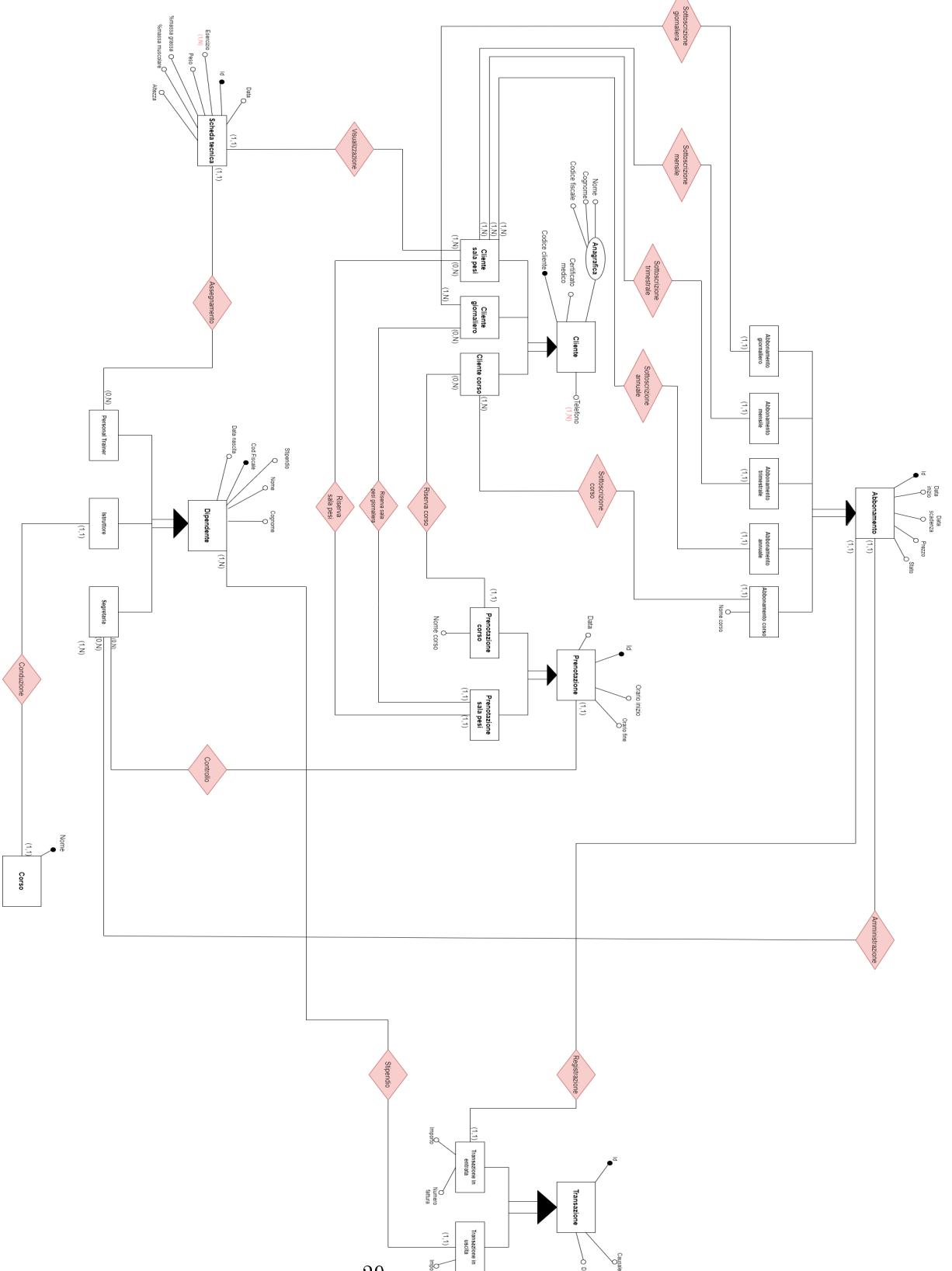


Figura 14: Schema E-R completo

2.6 Analisi della qualità del modello E-R

Come introdotto all'inizio della sezione relativa alla progettazione concettuale (nella sezione **2.1**), il nostro modello Entity-Relationship deve soddisfare dei requisiti di qualità affinché possa essere valido per proseguire l'implementazione della base di dati. Notiamo che il nostro modello soddisfa la Correttezza: ogni costrutto al suo interno viene utilizzato correttamente, sia al livello sintattico che semantico. Allo stesso modo soddisfa la Completezza, poiché ogni tema trattato nella prima parte della progettazione, ovvero quella relativa alla raccolta di informazioni e dei requisiti, viene rappresentato con efficienza all'interno dello schema ER. La Leggibilità anche è soddisfatta: le entità sono collegate in maniera piuttosto concettuale, seguendo l'idea da noi sviluppata al principio con lo **Scheletro dello schema** nella sezione **2.3**. Abbiamo cercato di evitare, ove possibile, l'intersezione di Entità e Relazioni, anche se spesso è risultato complicato a causa del livello di astrazione piuttosto alto del nostro schema E-R. Comunque, abbiamo fornito numerose descrizioni di ogni componente da noi aggiunta nello schema in numerose parti di questa trattazione, al fine di rendere chiaro il nostro modus operandi. Quindi, lo schema appare leggibile. Infine, lo schema risulta Minimale poiché non presenta ridondanze o cicli. Quindi possiamo affermare che lo schema E-R risulta corretto.

2.7 Dizionario dei dati

Dopo la modellazione dello schema E-R e il controllo della qualità di quest'ultimo, il passo da noi effettuato è relativo alla stesura di due tabelle distinte per Entità e Relazioni, contenenti le informazioni necessarie alla comprensione degli schemi visti in precedenza.

2.7.1 Entità

Prima di introdurre il dizionario dei dati relativo alle Entità, c'è da sottolineare che alcune di esse hanno bisogno di un appunto di chiarimento. A tale scopo, abbiamo inserito degli asterischi che rimandano l'appunto alla fine della tabella e inseriscono ulteriori spiegazioni.

NOME	DESCRIZIONE	ATTRIBUTI	IDENTIFICATORE
Cliente	Persona fisica che usufruisce dei servizi offerti dalla palestra, e di cui vengono registrati i dati personali nel database. Comprende le entità Cliente sala pesi, Cliente corso, Cliente special e Cliente giornaliero	Codice cliente (numerico), Certificato medico (booleano), Nome (stringa), Cognome(stringa), Codice fiscale (stringa), Telefono (numerico)	Codice cliente (numerico)
Cliente sala pesi	Tipologia di cliente che usufruisce attraverso un abbonamento solamente del servizio relativo alla sala pesi	Attributi di "Cliente"	Codice cliente (numerico)
Cliente corso	Tipologia di cliente che usufruisce attraverso un abbonamento solamente del servizio relativo ai corsi offerti	Attributi di "Cliente"	Codice cliente (numerico)
Cliente giornaliero	Tipologia di cliente che svolge l'attività di allenamento sottoscrivendo un abbonamento di singola entrata Nella palestra si dividono in istruttori, personal trainer e segretaria	Attributi di "Cliente"	Codice cliente (numerico)
Abbonamento	Modalità di accordo con la palestra per usufruire dei servizi da essa offerta. Generalizza le entità abbonamento giornaliero, mensile, trimestrale, annuale, corso	Id (numerico), Data inizio(data), Data scadenza (data), Descrizione tipo abbonamento (stringa), Prezzo (numerico), Stato (stringa)	Id (numerico)
Abbonamento giornaliero*	Tipo di abbonamento relativo al cliente che usufruisce della singola entrata	Attributi di "Abbonamento"	Id (numerico)

Abbonamento mensile	Tipo di abbonamento relativo al cliente che usufruisce del servizio di sala pesi con scadenza mensile	Attributi di "Abbonamento"	Id (numerico)
Abbonamento trimestrale	Tipo di abbonamento relativo al cliente che usufruisce del servizio di sala pesi con scadenza trimestrale	Attributi di "Abbonamento"	Id (numerico)
Abbonamento annuale	Tipo di abbonamento relativo al cliente che usufruisce del servizio di sala pesi con scadenza annuale	Attributi di "Abbonamento"	Id (numerico)
Abbonamento corso	Tipo di abbonamento relativo al cliente che usufruisce dei corsi offerti dalla palestra	Attributi di "Abbonamento"+ Nome corso(stringa)	Id (numerico)
Prenotazione	Atto compiuto dal cliente per usufruire in una determinata fascia oraria dei servizi della palestra. Generalizza le entità relative alle prenotazioni in sala pesi e quelle in sala corsi	Id (numerico), Descrizione (stringa), Orario inizio (orario), Orario fine (orario), Data (data)	Id (numerico)
Prenotazione corso	Tipo di prenotazione inherente ai clienti abbonati ai corsi	Attributi di "Prenotazione"+ Nome corso(stringa)	Id (numerico)
Prenotazione sala pesi	Tipo di prenotazione inherente ai clienti abbonati alla sala pesi	Attributi di "Prenotazione"	Id (numerico)
Transazione	Movimento in entrata o in uscita all'interno dell'attività, che comporta la registrazione di introito o di spesa. Generalizza le entità Transazione in entrata e Transazione in uscita	Id (numerico), Importo (numerico), Causale(stringa), Data(data)	Id (numerico)

Transazione in entrata	Movimento in entrata, corrispondente al saldo del conto di un cliente	Attributi di "Transazione"+ Numero fattura (numerico)+ Importo (numerico)	Id (numerico)
Transazione in uscita	Movimento in uscita, relativo al pagamento dello stipendio di un dipendente	Attributi di "Transazione"+ Importo (numerico)	Id (numerico)
Dipendente	Persona fisica che opera all'interno dell'attività e riceve una retribuzione monetaria. È una generalizzazione delle entità Personal trainer, Istruttore e Segretaria	Codice fiscale (stringa), Data nascita (data), Stipendio (numerico), Nome (stringa), Cognome (stringa), Ruolo (stringa)	Codice fiscale (stringa)
Personal trainer	Tipo di dipendente, addetto a seguire la sala pesi	Attributi di "Dipendente"	Codice fiscale (stringa)
Istruttore	Tipo di dipendente, addetto a seguire la sala corsi	Attributi di "Dipendente"	Codice fiscale (stringa)
Segretaria	Tipo di dipendente, addetto alla parte amministrativa e di coordinazione	Attributi di "Dipendente"	Codice fiscale (stringa)
Corso	Attività offerta all'interno della palestra per alcuni abbonati	Nome (stringa)	Nome (stringa)
Scheda tecnica	Scheda personalizzata in base alle necessità di allenamento di un cliente della palestra	Id (numerico), Data (data), Elenco esercizi (stringa), Peso(numerico), %Massa grassa (numerico), %Massa muscolare (numerico), Altezza (numerico)	Id (numerico)

ANNOTAZIONI:

- Per quanto riguarda l'entità “Abbonamento giornaliero”, l'attributo “Data scadenza” funziona nello stesso modo della sua generalizzazione, però la sua scadenza sarà indicata alla fine del giorno stesso.

2.7.2 Relazioni

NOME	DESCRIZIONE	ENTITÀ COINVOLTE	ATTRIBUTI
Sottoscrizione giornaliera	Associa un Abbonamento giornaliero a un Cliente giornaliero	Abbonamento giornaliero (1,1) Cliente giornaliero (1,N)	*
Sottoscrizione mensile	Associa un Abbonamento mensile ad un Cliente sala pesi	Abbonamento mensile (1,1) Cliente sala pesi (1,N)	*
Sottoscrizione trimestrale	Associa un Abbonamento trimestrale ad un Cliente sala pesi	Abbonamento trimestrale (1,1) Cliente sala pesi (1,N)	*
Sottoscrizione annuale	Associa un Abbonamento annuale ad un Cliente sala pesi	Abbonamento annuale (1,1) Cliente sala pesi (1,N)	*
Sottoscrizione corso	Associa un Abbonamento corso ad un Cliente corso	Abbonamento corso (1,1) Cliente corso (1,N)	*
Riserva sala pesi giornaliera	Associa una Prenotazione sala pesi ad un Cliente giornaliero	Prenotazione sala pesi (1,1) Cliente giornaliero (0,N)	*
Riserva sala pesi	Associa una Prenotazione sala pesi ad un Cliente sala pesi	Prenotazione sala pesi (1,1) Cliente sala pesi (0,N)	*
Riserva corso	Associa una Prenotazione corso ad un Cliente corso	Prenotazione corso (1,1) Cliente corso (0,N)	*
Visualizzazione	Associa una Scheda tecnica ad un Cliente sala pesi	Scheda tecnica (1,1) Cliente sala pesi (1,N)	*
Registrazione	Associa un Abbonamento ad una Transazione in entrata	Abbonamento (0,1) Transazione in entrata (1,1)	*

Controllo	Associa una Prenotazione alla Segreteria per controllare le prenotazioni	Prenotazione (1,1) Segreteria (0,N)	*
Amministrazione	Associa un Abbonamento alla Segreteria per amministrare gli abbonamenti	Abbonamento (1,1) Segreteria (0,N)	*
Stipendio	Associa una Transazione in uscita alla Segreteria per gestire gli stipendi dei dipendenti	Transazione in uscita (1,1) Segreteria (1,N)	*
Assegnamento	Associa una Scheda tecnica al Personal trainer che l'ha composta	Scheda tecnica (1,1) Personal trainer (0,N)	*
Conduzione	Associa un Corso ad un Istruttore	Corso (1,1) Istruttore (1,1)	*

2.8 Regole aziendali

2.8.1 Regole di vincolo

RV1 “Prezzo” relativo all’entità “Abbonamento” deve essere pari a 7 euro se “Descrizione tipo abbonamento” relativo all’entità “Abbonamento” è “Giornaliero”, pari a 40 euro se “Descrizione tipo abbonamento” relativo all’entità “Abbonamento” è “Mensile”, pari a 10 euro se “Descrizione tipo abbonamento” relativo all’entità “Abbonamento” è “Trimestrale”, pari a 320 euro se “Descrizione tipo abbonamento” relativo all’entità “Abbonamento” è “Annuale”, pari a 35 euro se “Descrizione tipo abbonamento” relativo all’entità “Abbonamento” è “Corso”.

RV2 “Descrizione tipo abbonamento” relativo all’entità “Abbonamento” può assumere esclusivamente i seguenti valori: “Giornaliero”, “Mensile”, “Trimestrale”, “Annuale” e “Corso”.

RV3 “Peso” dell’entità “Scheda tecnica” deve essere maggiore di 0.

RV4 “Percentuale Massa grassa” dell’entità “Scheda tecnica” deve essere maggiore o uguale a 0.

RV5 “Percentuale Massa muscolare” dell’entità “Scheda tecnica” deve essere maggiore di 0.

RV6 “Altezza” dell’entità “Scheda tecnica” deve essere maggiore di 0.

RV7 “Ora di inizio” dell’entità “Prenotazione” può essere uguale esclusivamente a: “8:00”, “9:30”, “11:00”, “12:30”, “14:00”, “15:30”, “17:00”, “18:30”, “20:00”, “21:30”.

RV8 “Ora di fine” dell’entità “Prenotazione” deve essere uguale a “Ora di inizio” più 90 minuti.

RV9 “Data scadenza” relativa all’entità “Abbonamento” se è “Giornaliero” ha scadenza uguale al giorno di sottoscrizione, se è “Mensile” o è “Corso” ha scadenza 30 giorni dopo la sottoscrizione, se è “Trimestrale” ha scadenza 90 giorni dopo, se è “Annuale” ha scadenza 365 giorni dopo la sottoscrizione.

RV10 Tutti i “Clienti” possono prenotarsi massimo ad una fascia oraria o corso al giorno.

RV11 “Tipo transazione” dell’entità “Transazione” può assumere solo il valore “In entrata” o “In uscita”.

RV12 “Abbonamento corso” deve avere una durata pari ad un mese.

RV13 Ogni 5 del mese deve essere pagato lo stipendio dei dipendenti.

RV14 Non possono esistere corsi con lo stesso nome.

RV15 Non possono esistere esercizi con lo stesso nome.

RV16 I clienti che possiedono un certificato medico scaduto non possono praticare attività all’interno della palestra.

RV17 L’attributo "Stato" in "Abbonamento" può assumere solamente i valori "Valido" o "Scaduto".

RV18 Una Transazione non può essere eliminata.

RV19 Una Scheda tecnica viene eliminata solo se viene eliminato il Cliente a cui è assegnata.

RV20 Un’ Abbonamento viene eliminato solo se viene eliminato il Clien-

te a cui è assegnata.

2.8.2 Regole di derivazione

Non abbiamo stilato nessuna regola di derivazione, dato che in seguito all'analisi della qualità dello schema E-R siamo giunti alla conclusione che lo schema risulta essere minimale. Quindi, nessun attributo è derivabile e non sono presenti ridondanze. Comunque, nella prossima parte della progettazione, durante l'analisi delle ridondanze, viene valutata la possibilità di aggiungere ridondanze o di mantenere lo schema in questo modo.

3 PROGETTAZIONE LOGICA

3.1 Tavole dei volumi e delle operazioni

3.1.1 Tavola dei volumi

CONCETTO	TIPO	VOLUME
Cliente	E	300
Cliente sala pesi	F	170
Cliente corso	E	100
Cliente giornaliero	E	30
Abbonamento	E	300
Abbonamento giornaliero	E	30
Abbonamento mensile	E	35
Abbonamento trimestrale	E	88
Abbonamento annuale	E	47
Abbonamento corso	E	100
Prenotazione	E	400(200+200) (al giorno)
Prenotazione corso	E	200
Prenotazione sala pesi	E	200
Transazione	E	311
Transazione in entrata	E	300
Transazione in uscita	E	11
Dipendente	E	11
Personal trainer	E	5
Istruttore	E	5
Segretaria	E	1
Corso	E	5
Scheda tecnica	E	170
Sottoscrizione giornaliera	R	30
Sottoscrizione mensile	R	35
Sottoscrizione trimestrale	R	88
Sottoscrizione annuale	R	47
Sottoscrizione corso	R	100

Riserva sala pesi giornaliera	R	30
Riserva sala pesi	R	170
Riserva corso	R	200
Visualizzazione	R	170(1*170)
Controllo	R	400
Amministrazione	R	300
Stipendi	R	11
Assegnamento	R	170
Conduzione	R	5
Registrazione	R	300

3.1.2 Tavola delle operazioni

OPERAZIONE	FREQUENZA
1	15 volte al giorno
2	1 volta all'anno
3	10 volte al giorno
4	15 volte all'anno
5	1 volta all'anno
6	25 volte al giorno
7	5 volte al giorno
8	15 volte al giorno
9	300 volte al giorno
10	2 volte al mese
11	1 volta all'anno
12	2 volte al giorno
13	10 volte all'anno
14	1 volta ogni 2 anni
15	1 volta ogni 2 anni
16	10 volte al giorno
17	80 volte al giorno
18	10 volte al giorno
19	20 volte al giorno
20	20 volte al giorno
21	300 volte al giorno
22	20 volte al giorno
23	20 volte al giorno
24	20 volte al giorno
25	20 volte al giorno
26	10 volte al giorno
27	2 volte al giorno
28	20 volte al giorno
29	50 volte al giorno
30	40 volte al giorno
31	10 volte al giorno
32	5 volte al giorno
33	1 volta al giorno
34	1 volta al giorno
35	1 volta al mese
36	1 volta all'anno
37	10 volte al giorno
38	1 volta al giorno
39	150 volte al giorno

40	10 volte al giorno
41	2 volte a settimana
42	1 volta al mese
43	1 volta al mese
44	20 volte al giorno
45	1 volta al mese
46	20 volte al giorno
47	1 volta all'anno
48	1 volta all'anno
49	1 volta all'anno

3.2 Ristrutturazione dello schema concettuale

3.2.1 Analisi derivazioni e ridondanze

Fatta un'attenta analisi del nostro progetto, abbiamo notato che non sono presenti ridondanze di alcun tipo; questo è un buon risultato, ma nella realtà, in progetti complessi, è raro che questo si verifichi. A scopo didattico, abbiamo pensato di aggiungere una ridondanza.

Aggiungiamo all'entità Cliente, l'attributo relativo al numero totale di abbonamenti sottoscritti fino al momento preso in considerazione. Nel particolare, analizzeremo il caso dell'operazione 27, cioè quella che permette di visualizzare il numero di abbonamenti fatto da un cliente, e dell'operazione 6 relativa all'inserimento di un abbonamento.

- **Assenza di ridondanza**

L'inserimento di un abbonamento per un dato cliente è la prima operazione coinvolta nell'analisi della ridondanza. Vengono coinvolte le entità "Cliente", "Abbonamento" e "Transazione" e le relazioni "Registrazione" e "Sottoscrizione". L'asterisco all'interno della tabella sta a specificare che quest'ultima relazione indica una qualsiasi delle varie tipologie che collegano "Cliente" con "Abbonamento", cioè "Giornaliera", "Mensile", "Trimestrale", "Annuale" o "Corso".

Tale operazione viene svolta 25 volte al giorno.

OPERAZIONE 6			
CONCETTO	COSTRUTTO	ACCESSI	TIPO
Cliente	E	1	L
Abbonamento	E	1	S
Transazione	E	1	S
Sottoscrizione*	R	1	S
Registrazione	R	1	S

Il numero di abbonamenti sottoscritti da un cliente può essere ricavato dalle relazioni “Sottoscrizione giornaliera”, “Sottoscrizione mensile”, “Sottoscrizione trimestrale”, “Sottoscrizione annuale” e “Sottoscrizione corso”. Analizziamo dunque la possibilità di memorizzare l’attributo Numero abbonamenti in "Cliente". Si specifica che tale operazione viene effettuata con una frequenza di 2 volte al giorno.

OPERAZIONE 27			
CONCETTO	COSTRUTTO	ACCESSI	TIPO
Sottoscrizione giornaliera	R	2	L
Sottoscrizione mensile	R	3	L
Sottoscrizione trimestrale	R	0.3	L
Sottoscrizione annuale	R	0.7	L
Sottoscrizione corso	R	1.8	L
Cliente	E	1	L

Quindi, andiamo a calcolare il costo singolo di ciascuna delle operazioni e il costo totale, in base al calcolo sugli accessi per tipo moltiplicato per la frequenza.

COSTO TOTALE			
OPERAZIONE	COSTO	FREQUENZA	TOTALE
27	8.8	2	$8.8 \cdot 2 = 17.6$
6	9	25	$9 \cdot 25 = 225$
			=242.6

- Presenza di ridondanza

Per aggiungere un abbonamento di un cliente nel caso di ridondanza, vengono coinvolte le entità "Cliente" (sia in lettura che in scrittura), "Abbonamento", "Transazione", e le relazioni "Registrazione" e "Sottoscrizione". L'asterisco all'interno della tabella sta a specificare che quest'ultima relazione indica una qualsiasi delle varie tipologie che collegano "Cliente" con "Abbonamento", cioè "Giornaliera", "Mensile", "Trimestrale", "Annuale" o "Corso".

Tale operazione viene svolta 25 volte al giorno.

OPERAZIONE 6			
CONCETTO	COSTRUTTO	ACCESSI	TIPO
Cliente	E	1	L
Abbonamento	E	1	S
Transazione	E	1	S
Sottoscrizione*	R	1	S
Cliente	E	1	S
Registrazione	R	1	S

Il numero di abbonamenti di un cliente in caso di presenza di ridondanza è accessibile semplicemente in lettura dall'entità "Cliente".

Questa operazione viene svolta con una frequenza di 2 volte al giorno.

OPERAZIONE 27			
CONCETTO	COSTRUTTO	ACCESSI	TIPO
Cliente	E	1	L

Quindi, andiamo a calcolare il costo singolo di ciascuna delle operazioni e il costo totale, in base al calcolo sugli accessi per tipo moltiplicato per la frequenza.

COSTO TOTALE			
OPERAZIONE	COSTO	FREQUENZA	TOTALE
27	1	2	$1*2=2$
6	11	25	$11*25=275$
			=277

- **Confronto dei costi**

Analizzando attentamente i costi delle operazioni sia in assenza che in presenza della ridondanza da noi presa in considerazione, possiamo giungere a delle conclusioni interessanti.

Confrontando il costo singolo totale dell'operazione 27, possiamo notare che è più conveniente quella corrispondente alla presenza di ridondanza: infatti essa ha un costo pari a 2, mentre quella in assenza di ridondanza ha un costo equivalente a 17.6.

Allo stesso modo, confrontando i costi singoli totali dell'operazione 6, possiamo notare che quello in assenza di ridondanza, pari a 225, è più conveniente rispetto a quello in presenza di ridondanza, che ammonta a 275.

Quindi, sommando i costi delle operazioni in assenza notiamo che il totale è uguale a 242.6, mentre quello in presenza è pari a 277.

La differenza possiamo notare che è minima, ma abbiamo ritenuto più utile in termini di implementazione l'esclusione della ridondanza sull'entità "Cliente".

3.2.2 Eliminazione delle gerarchie

Per quanto riguarda gli abbonamenti, abbiamo scelto di accorpate le figlie nel padre, quindi "Abbonamento giornaliero", "Abbonamento mensile", "Abbonamento trimestrale", "Abbonamento annuale" e "Abbonamento corso" in "Abbonamento" aggiungendo un attributo "Descrizione abbonamento" che appunto specifica il tipo di abbonamento. Lo abbiamo visto utile, in quanto le entità figlie hanno massimo un attributo in più rispetto al padre e, così facendo, pur ottenendo valori "NULL", ma non eccessivi, poiché solo "Abbonamento Corso" possiede un attributo in più, ci assicuriamo un numero minore di accessi rispetto alle altre due tecniche di eliminazione delle generalizzazioni.

Non abbiamo scelto di accorpore il padre nelle figlie perché andremmo a creare solo delle inutili ed eccessive ridondanze, poiché gli attributi del padre sarebbero ripetuti più volte in ogni entità senza una particolare esigenza. Al contempo non abbiamo reputato utile la sostituzione della generalizzazione con associazioni poiché essendo totale la generalizzazione, non è conveniente usare questo metodo.

Per quanto riguarda i dipendenti, abbiamo scelto la sostituzione della generalizzazione con associazioni, quindi tra “Dipendente” e le “attuali” figlie inseriremo delle relazioni introdotte dal termine “Ruolo”. Questa metodologia di sostituzione ci permette di tenere sia le entità figlie con le loro particolari mansioni, sia l’entità padre con gli attributi comuni a tutte le entità figlie.

Non abbiamo scelto di accorpore le figlie nel padre poiché la diversa “specializzazione” delle figlie ci complica la loro distinzione utile alla gestione di diverse mansioni che esse ricoprono.

Al contempo non abbiamo reputato utile l’accorpamento del genitore nelle figlie poiché con questa operazione avremmo ottenuto troppa ridondanza dei dati dovuta al fatto che tutti gli attributi del padre vengono ripetuti nelle figlie. Dobbiamo quindi aggiungere una regola di vincolo vincolo dovuta all’eliminazione di questa generalizzazione (tra dipendenti e le figlie):

RV21 Ogni occorrenza dell’entità “Dipendente” deve essere occorrenza di “Personal Trainer” o di “Istruttore” o di “Segretaria”, non potrà dunque essere occorrenza contemporaneamente di due o più di queste entità.

Per quanto riguarda le transazioni, abbiamo scelto di accorpore le figlie nel padre, quindi “Transazione in entrata” e “Transazione in uscita” in “Transazione”, aggiungendo un attributo “Tipo transazione” che appunto specifica il tipo di transazione, e gli attributi relativi agli importi in entrata e in uscita che precedentemente erano delle entità figlie. Anche in questo caso abbiamo visto utile questa tecnica (come nel caso di Abbonamento) perché le entità figlie hanno massimo un attributo in più rispetto al padre e, così facendo, pur ottenendo valori “NULL”, ma non eccessivi, poiché solo “Transazione in entrata” possiede un attributo in più, ci assicuriamo un numero minore di accessi rispetto alle altre due tecniche di eliminazione delle generalizzazioni. Non abbiamo scelto di accorpore il padre nelle figlie perché andremmo a creare solo delle inutili ed eccessive ridondanze, poiché gli attributi del padre sarebbero ripetuti più volte in ogni entità senza una particolare esigenza.

Al contempo non abbiamo reputato utile la sostituzione della generalizzazione con associazioni poiché essendo totale la generalizzazione, non è conveniente usare questo metodo.

Per quanto riguarda le prenotazioni, abbiamo scelto di accorpore le figlie nel padre, quindi “Prenotazione corso” e “Prenotazione sala pesi” in “Preno-

tazione” aggiungendo un attributo “Descrizione prenotazione” che specifica il tipo di prenotazione. Lo abbiamo visto utile, in quanto le entità figlie hanno massimo un attributo in più rispetto al padre e, così facendo, pur ottenendo valori “NULL”, ma non eccessivi, poiché solo “Prenotazione corso” possiede un attributo in più, ci assicuriamo un numero minore di accessi rispetto alle altre due tecniche di eliminazione delle generalizzazioni. Non abbiamo scelto di accorpore il padre nelle figlie perché andremmo a creare solo delle inutili ed eccessive ridondanze, poiché gli attributi del padre sarebbero ripetuti più volte in ogni entità senza una particolare esigenza.

Al contempo non abbiamo reputato utile la sostituzione della generalizzazione con associazioni poiché essendo totale la generalizzazione, non è conveniente usare questo metodo.

Per quanto riguarda i clienti, abbiamo scelto di accorpore le figlie nel padre, quindi “Cliente giornaliero”, “Cliente corso” e “Cliente sala pesi” in “Cliente” aggiungendo un attributo “Descrizione cliente” che ne specifica il tipo di cliente. Lo abbiamo visto utile, in quanto le entità figlie non hanno attributi in più rispetto al padre e, così facendo, non otteniamo valori “NULL”, e ci assicuriamo un numero minore di accessi rispetto alle altre due tecniche di eliminazione delle generalizzazioni. Non abbiamo scelto di accorpore il padre nelle figlie perché andremmo a creare solo delle inutili ed eccessive ridondanze, poiché gli attributi del padre sarebbero ripetuti più volte in ogni entità senza una particolare esigenza.

Al contempo non abbiamo reputato utile la sostituzione della generalizzazione con associazioni poiché essendo totale la generalizzazione, non è conveniente usare questo metodo.

Di seguito, riportiamo il nuovo schema da noi modellato in seguito all’eliminazione delle generalizzazioni:

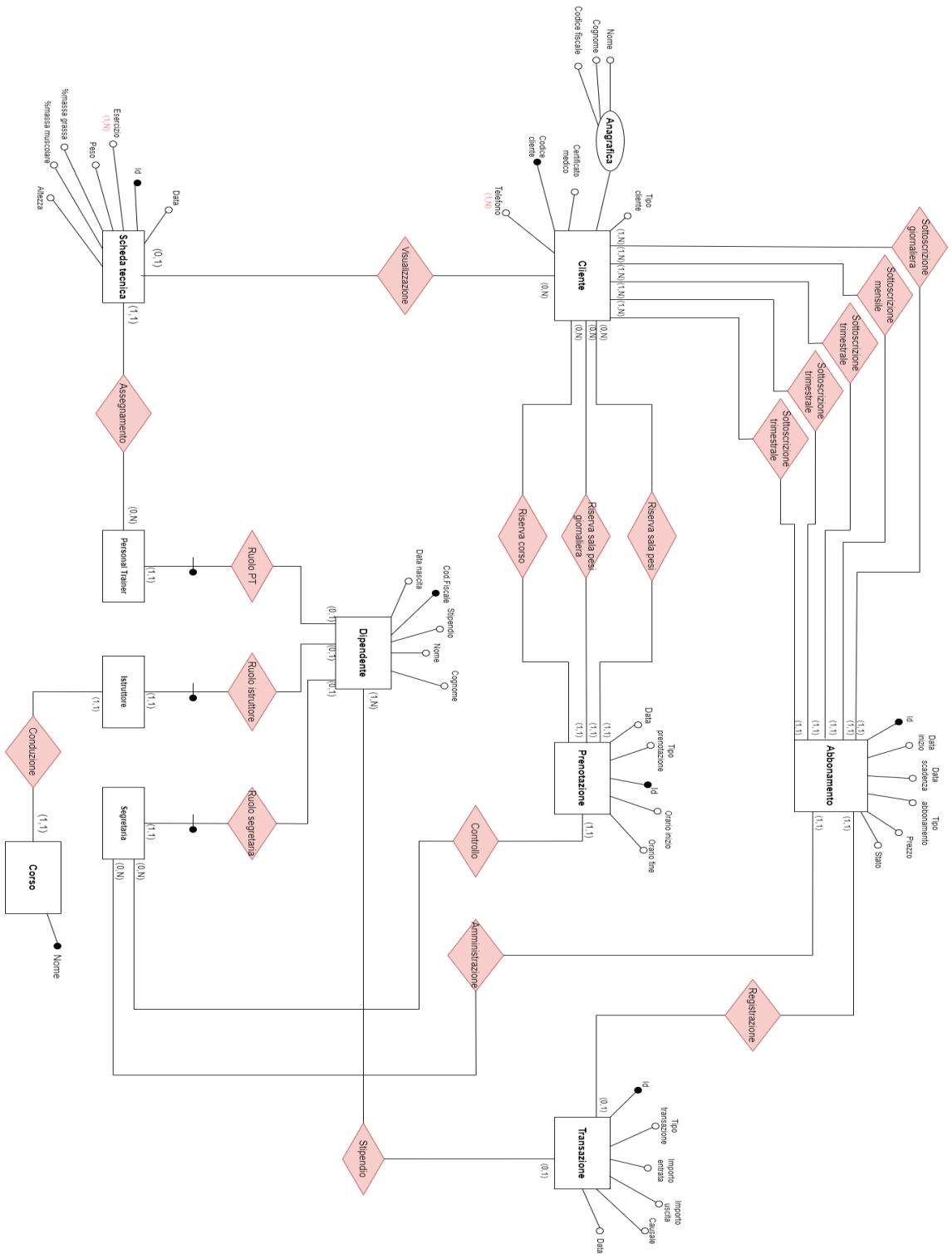


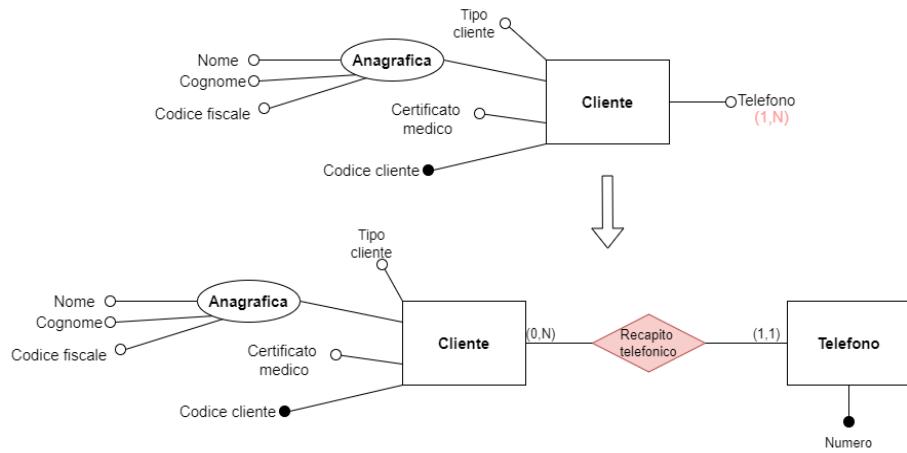
Figura 15: Schema E-R dopo l'eliminazione delle generalizzazioni

3.3 Partizionamento-accorpamento di concetti

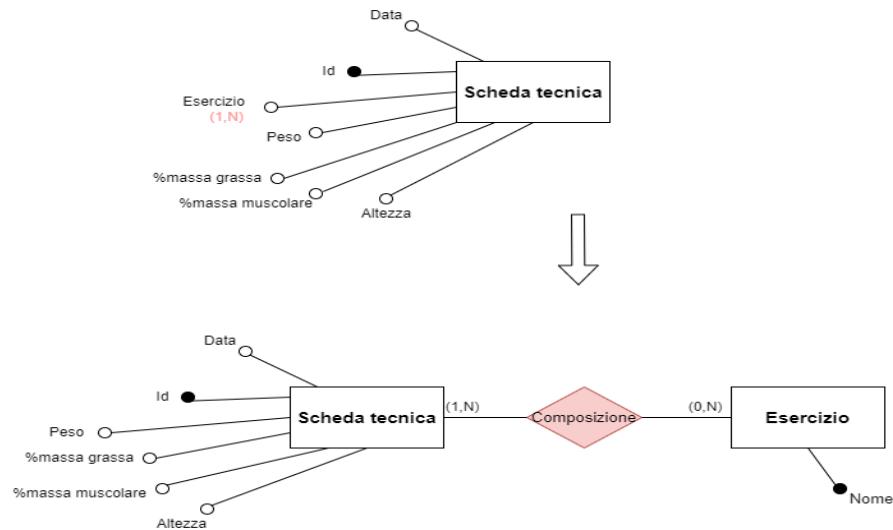
3.3.1 Eliminazione degli attributi multivalore

Come si può osservare anche dall'ultimo schema, gli attributi multivalore sono due, cioè “Telefono” relativo all'entità “Cliente” e “Elenco esercizi” relativo all'entità “Scheda tecnica”.

Di seguito è mostrata la ristrutturazione che abbiamo progettato per l'attributo “Telefono”:



Nella seguente figura, invece, è riportata la ristrutturazione che abbiamo proposto dell'attributo “Elenco esercizi” che individua gli esercizi di cui una “Scheda Tecnica” si compone:

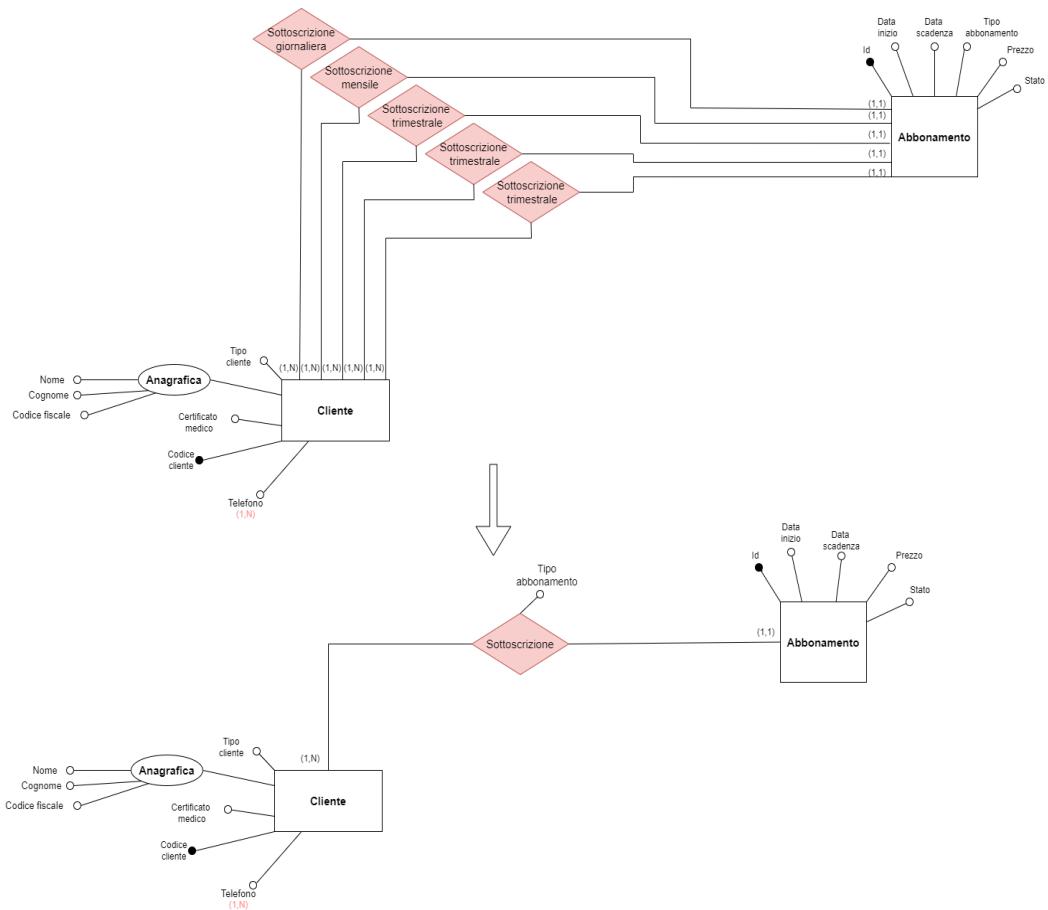


3.3.2 Accorpamento relazioni

Abbiamo, inoltre, individuato due accorpamenti di relazioni utili al miglioramento sia della visibilità sugli schemi, sia di efficienza in termini di accessi. Questi due accorpamenti individuati sono relativi ai blocchi di relazioni:

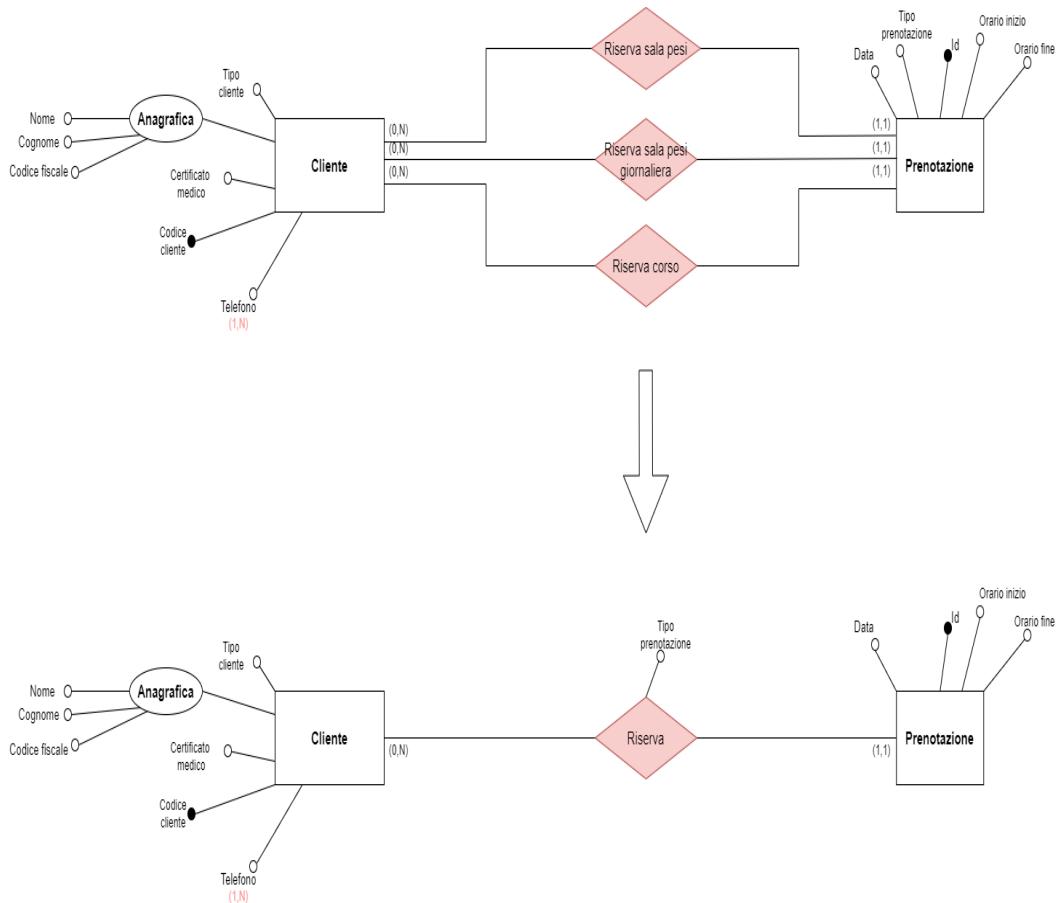
1. “Sottoscrizione giornaliera”, “Sottoscrizione mensile”, “Sottoscrizione trimestrale”, “Sottoscrizione annuale” e “Sottoscrizione corso”;
2. “Riserva sala pesi” e “Riserva sala pesi giornaliera”.

Per quanto riguarda il blocco 1, lo abbiamo ristrutturato come segue:



È stata presa la seguente decisione poiché, per osservare le relazioni che vigono tra clienti e abbonamenti, a prescindere, sarebbero stati chiamati in causa tutti e 5 i blocchi di associazione relativi alla sottoscrizione contemporaneamente.

Per quanto riguarda il blocco 2, lo abbiamo ristrutturato come segue:



Abbiamo preso questa decisione per lo stesso motivo di quella precedente, poiché esprimono lo stesso concetto e verrebbero anch'essi acceduti contemporaneamente.

3.3.3 Schema E-R ristrutturato

Dunque, in seguito alle nostre analisi finali, riportiamo di seguito lo schema E-R completamente ristrutturato, che abbiamo preso in considerazione per l'implementazione finale della base di dati:

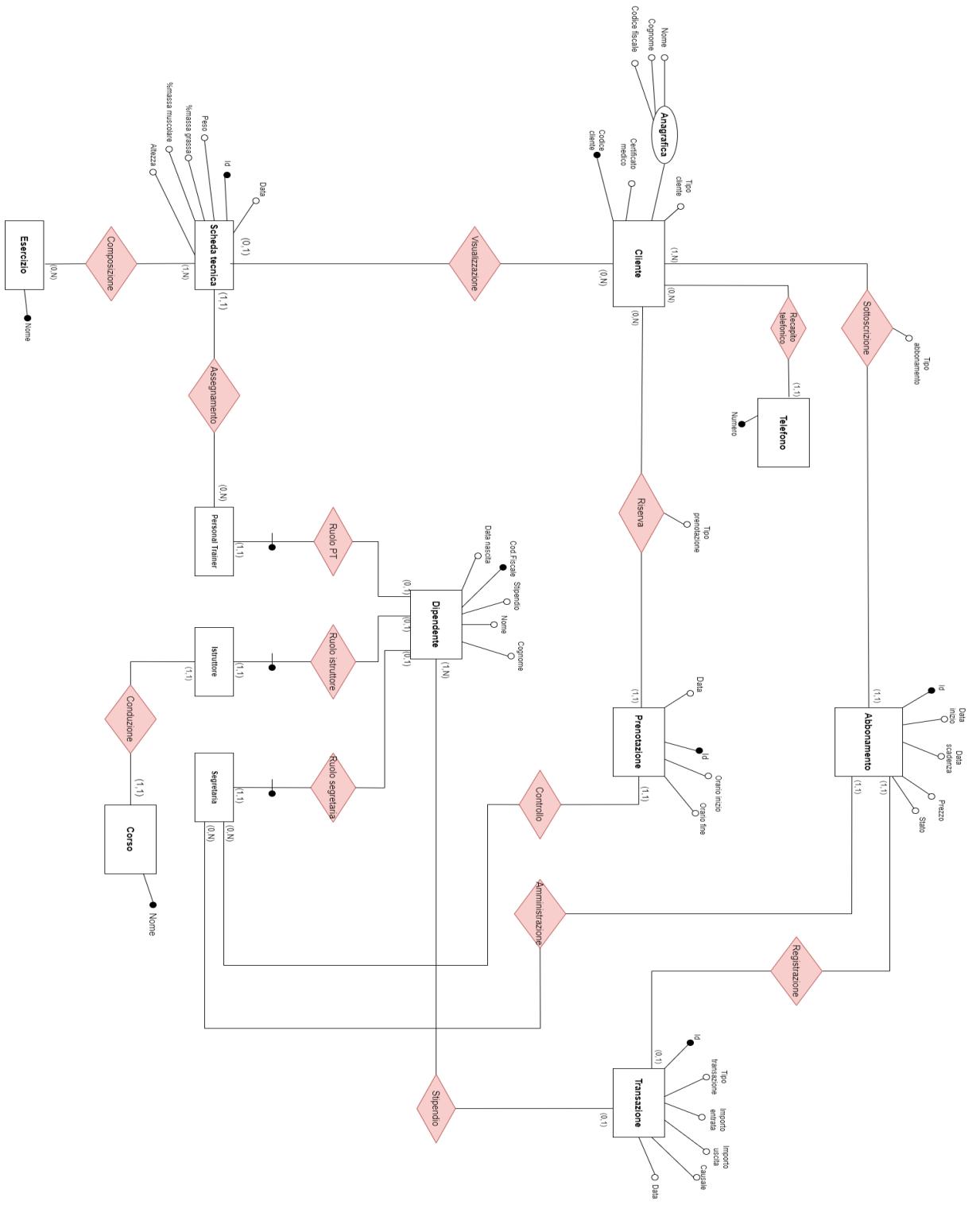


Figura 16: Schema E-R finale ristrutturato

3.4 Elenco degli identificatori principali

Nella tabella sottostante illustreremo gli identificatori delle nostre entità. Per quanto riguarda le entità “Personal Trainer”, “Istruttore” e “Segretaria” abbiamo utilizzato un identificatore esterno, più precisamente “Codice fiscale” di “Dipendente”.

Per l’entità “Cliente” abbiamo introdotto un “Codice cliente” come identificatore, nonostante la presenza di “Codice fiscale”, per semplificare la ricerca e l’identificazione di un cliente.

Nell’entità “Corso” abbiamo preso come identificatore il “Nome” del corso poiché non possono esistere due corsi con nome uguale. Stessa identica decisione è stata presa nell’entità “Esercizio” con l’identificatore “Nome”.

ENTITÀ	IDENTIFICATORE
Cliente	Codice cliente
Scheda tecnica	Id
Personal trainer	Codice fiscale
Dipendente	Codice fiscale
Istruttore	Codice fiscale
Segretaria	Codice fiscale
Corso	Nome
Transazione	Id
Abbonamento	Id
Telefono	Numero
Esercizio	Nome
Prenotazione	Id

3.5 Normalizzazione

ASSOCIAZIONI:

Analizzando lo schema concettuale, si osserva che sono presenti esclusivamente associazioni binarie, per cui tutte le associazioni sono della forma normale di Boyce e Codd.

ENTITÀ:

ENTITÀ	IDENTIFICATORE
Cliente	Non esistono dipendenze non banali tra gli attributi
Scheda tecnica	Non esistono dipendenze non banali tra gli attributi
Personal Trainer	Non esistono dipendenze non banali tra gli attributi
Dipendente	Non esistono dipendenze non banali tra gli attributi
Istruttore	Non esistono dipendenze non banali tra gli attributi
Segretaria	Non esistono dipendenze non banali tra gli attributi
Corso	Non esistono dipendenze non banali tra gli attributi
Transazione	Non esistono dipendenze non banali tra gli attributi
Abbonamento	Non esistono dipendenze non banali tra gli attributi
Telefono	Non esistono dipendenze non banali tra gli attributi
Esercizio	Non esistono dipendenze non banali tra gli attributi
Prenotazione	Non esistono dipendenze non banali tra gli attributi

3.6 Traduzione verso il modello relazionale

ENTITÀ/RELAZIONE	TRADUZIONE
Cliente	Cliente (<u>CodCli</u> , Nome, Cognome, CFcli, TipoCli, CertMedico)
Scheda tecnica	SchedaTecnica (<u>IdScheda</u> , Dip, Cliente, Data, Peso, Altezza, Grasso, Muscolo)
Personal Trainer	Pt(<u>CFDip</u>)
Dipendente	Dipendente (<u>CFDip</u> , Nome, Cognome, DataN, Stipendio)
Istruttore	Istruttore (<u>CFDip</u> , NomeCo)
Segretaria	Segretaria (<u>CFDip</u>)
Corso	Corso (<u>NomeCo</u>)
Transazione	Transazione (<u>IdTrans</u> , Tipo, Causale, ImportoEnt, ImportoUsc, Data, Dipendente)
Abbonamento	Abbonamento (<u>IdAbb</u> , Transaz, Cliente, Dip, DataIni, DataSca, Prezzo, Tipo, Stato)
Telefono	Telefono (<u>Num</u> , Cliente)
Esercizio	Esercizio (<u>NomeEs</u>)
Prenotazione	Prenotazione (<u>IdPre</u> , Cliente, Dip, Data, Oraln, OraFin, Tipo)
Composizione	Composizione(<u>IdScheda</u> , <u>NomeEs</u>)

VINCOLI DI RIFERIMENTO:

TRADUZIONI	VINCOLI DI RIFERIMENTO
Cliente (<u>CodCli</u> , Nome, Cognome, CFCli, TipoCli, CertMedico)	*
SchedaTecnica (<u>IdScheda</u> , Dip, Cliente, Data, Peso, Altezza, Grasso, Muscolo)	Dip -> PersonalTrainer.CFDip Cliente -> Cliente.CodCli
Pt (<u>CFDip</u>)	CFDip -> Dipendente.CFDip
Dipendente (<u>CFDip</u> , Nome, Cognome, DataN, Stipendio)	*
Istruttore (<u>CFDip</u> , Corso)	CFDip -> Dipendente.CFDip Corso -> Corso.NomeCo
Segretaria (<u>CFDip</u>)	CFDip -> Dipendente.CFDip
Corso (<u>NomeCo</u>)	*
Transazione (<u>IdTrans</u> , Tipo, Causale, Importo, Data, Dipendente)	Dipendente -> Dipendente.CFDip
Abbonamento (<u>IdAbb</u> , Transaz, Cliente, Dip, DataIn, DataSca, Prezzo, Tipo, Stato)	Transaz -> Transazione.IdTrans Cliente -> Cliente.CodCli Dip -> Segreteria.CFDip
Telefono (<u>Num</u> , Cliente)	Cliente -> Cliente.CodCli
Esercizio (<u>NomeEs</u>)	*
Prenotazione (<u>IdPre</u> , Cliente, Dip, Data, Orain, OraFin, Tipo)	Cliente -> Cliente.CodCli Dip -> Segreteria.CFDip
Composizione(<u>IdScheda</u> , <u>NomeEs</u>)	IdScheda -> SchedaTecnica.IdScheda NomeEs -> Esercizio.NomeEs

4 Codifica SQL e testing

L'ultima fase della nostra progettazione è quella che riguarda l'implementazione del Database nel linguaggio SQL. Di seguito abbiamo riportato il codice d'implementazione delle tabelle nel terminale, direttamente come sono nel dump.

Di seguito riportiamo uno screen dell'indice, ottenuto attraverso il comando "show tables" all'interno del prompt dei comandi, che mostra tutte le tabelle contenute all'interno della base di dati implementata:

Tables_in_progetto_ufficiale
abbonamenti
clienti
composizione
corso
dipendenti
esercizio
istruttore
personaltrainer
prenotazioni
schedatecnica
segretaria
telefono
transazioni

13 rows in set (0.001 sec)

4.1 Implementazione delle tabelle

ABBONAMENTI:

```
MariaDB [progetto_ufficiale]> create table abbonamenti (
->     IdAbb int auto_increment primary key,
->     Cliente smallint,
->     foreign key(Cliente) references clienti(CodCli)
->     on delete cascade
->     on update cascade,
->     Tipo varchar(25) not null check(Tipo IN('giornaliero','mensile', 'trimestrale', 'annuale', 'corso')),
->     DataIni date not null,
->     DataSca date not null,
->     Dip varchar(16),
->     foreign key(Dip) references segretaria(CFDip)
->     on delete set null
->     on update cascade,
->     Transaz int not null,
->     foreign key(Transaz) references transazioni(IdTrans)
->     on delete no action
->     on update cascade,
->     Prezzo smallint not null,
->     Stato char(7) not null default 'valido' check(Stato IN('valido','scaduto')),
->     unique(Transaz)
-> );
```

CLIENTI:

```
MariaDB [progetto_ufficiale]> create table clienti (
->      CodCli smallint auto_increment primary key,
->      Cognome varchar(25) not null,
->      Nome varchar(25) not null,
->      CFCli char(16) not null,
->      CertMedico boolean default null,
->      TipoCli varchar(11) check(TipoCli IN('sala pesi','corso')),
->      unique(Cognome,Nome,CFCli)
-> );
```

COMPOSIZIONE:

```
MariaDB [progetto_ufficiale]> create table composizione (
->      NomeEs varchar(25),
->      foreign key(NomeEs) references esercizio(NomeEs)
->      on delete cascade
->      on update cascade,
->      IdScheda smallint,
->      foreign key(IdScheda) references schedatecnica(IdScheda)
->      on delete cascade
->      on update cascade,
->      primary key(NomeEs, IdScheda)
-> );
```

CORSO:

```
MariaDB [progetto_ufficiale]> create table corso (
->      NomeCo varchar(25) primary key
-> );
```

DIPENDENTI:

```
MariaDB [progetto_ufficiale]> create table dipendenti (
->      CFDip varchar(16) primary key,
->      Cognome varchar(25) not null,
->      Nome varchar(25) not null,
->      Stipendio float(6,2) default 0.00,
->      DataN date not null
-> );
```

ESERCIZIO:

```
MariaDB [progetto_ufficiale]> create table esercizio (
    ->      NomeEs varchar(25) primary key
    ->      );
```

ISTRUTTORE:

```
MariaDB [progetto_ufficiale]> create table istruttore (
    ->      CFDip varchar(16),
    ->      foreign key (CFDip) references dipendenti(CFDip)
    ->      on delete cascade
    ->      on update cascade,
    ->      NomeCo varchar(25),
    ->      foreign key(NomeCo) references corso(NomeCo)
    ->      on delete set null
    ->      on update set null,
    ->      primary key(CFDip)
    ->      );
```

PERSONAL TRAINER:

```
MariaDB [progetto_ufficiale]> create table personaltrainer (
    ->      CFDip varchar(16),
    ->      foreign key (CFDip) references dipendenti(CFDip)
    ->      on delete cascade
    ->      on update cascade,
    ->      primary key(CFDip)
    ->      );
```

PRENOTAZIONI:

```
MariaDB [progetto_ufficiale]> create table prenotazioni (
    ->      IdPre int auto_increment primary key,
    ->      Cliente smallint not null,
    ->      Dip varchar(16),
    ->      Data date not null,
    ->      OraIn time not null check(OraIn IN('08:00:00', '09:30:00', '11:00:00', '12:30:00', '14:00:00', '15:30:00', '17:00:00', '18:30:00', '20:00:00', '21:30:00')),
    ->      OraFin time not null check(OraFin IN(OraIn + INTERVAL 90 MINUTE)),
    ->      Tipo varchar(9) not null check(Tipo IN('Sala pesi', 'Corso')),
    ->      foreign key(Cliente) references clienti(CodCli)
    ->      on delete cascade
    ->      on update cascade,
    ->      foreign key(Dip) references segretaria(CFDip)
    ->      on delete set null
    ->      on update cascade
    ->      );
```

SCHEMA TECNICA:

```
MariaDB [progetto_ufficiale]> create table schedatecnica (
->      IdScheda smallint auto_increment primary key,
->      Data date not null,
->      Peso float(4,1) not null check(Peso > 0),
->      Altezza smallint not null check(Altezza > 0),
->      Grasso smallint not null check(Grasso > 0),
->      Muscolo smallint not null check(Muscolo > 0),
->      Dip varchar(16),
->      foreign key(Dip) references personaltrainer(CFDip)
->      on delete set null
->      on update cascade,
->      Cliente smallint not null,
->      foreign key(Cliente) references clienti(CodCli)
->      on delete cascade
->      on update cascade
->      );
```

SEGRETARIA:

```
MariaDB [progetto_ufficiale]> create table segretaria (
->      CFDip varchar(16),
->      foreign key (CFDip) references dipendenti(CFDip)
->      on delete cascade
->      on update cascade,
->      primary key (CFDip)
->      );
```

TELEFONO:

```
MariaDB [progetto_ufficiale]> create table telefono (
->      Num varchar(10) primary key,
->      Cliente smallint not null,
->      foreign key(Cliente) references clienti(CodCli)
->      on delete cascade
->      on update cascade
->      );
```

TRASAZIONI:

```
MariaDB [progetto_ufficiale]> create table transazioni (
->      IdTrans int auto_increment primary key,
->      Tipo char(7) not null check(Tipo IN('entrata','uscita')),
->      Causale varchar(100),
->      ImportoUsc float(6,2) default 0.00 references dipendenti(Stipendio),
->      ImportoEnt smallint default 0 references abbonamenti(Prezzo),
->      Data date not null,
->      Dipendente varchar(16),
->      foreign key(Dipendente) references segretaria(CFDip)
->      on delete set null
->      on update cascade
->      );
```

4.2 Codifica delle operazioni

In questa sezione, abbiamo inserito la codifica di tutte le operazioni effettuate all'interno del Database, con esempi pratici dei risultati ottenuti. Ci teniamo a precisare che per alcune operazioni estremamente banali non abbiamo ritenuto necessario inserire screenshot.

1. Inserimento nuovo cliente

```
insert into clienti(Cognome,Nome,CFCli,CertMedico,TipoCli)
values(...);
```

```
MariaDB [progetto_ufficiale]> insert into clienti(Cognome,Nome,CFCli,CertMedico,TipoCli)
->      values('Di Renzo', 'Michele', 'ABCDEFGHIJKLMOP', 1, 'Sala pesi');
Query OK, 1 row affected (0.046 sec)

MariaDB [progetto_ufficiale]> insert into clienti(Cognome,Nome,CFCli,CertMedico,TipoCli)
->      values('Agresta', 'Arianna', 'ABCDEFGHIJKLMOP', 1, 'Corso');
Query OK, 1 row affected (0.022 sec)

MariaDB [progetto_ufficiale]> insert into clienti(Cognome,Nome,CFCli,CertMedico,TipoCli)
->      values('Rossi', 'Valentino', 'AQDERFGHIJKLMOP', 1, 'Sala pesi');
Query OK, 1 row affected (0.016 sec)

MariaDB [progetto_ufficiale]> select * from clienti;
+-----+-----+-----+-----+-----+
| CodCli | Cognome | Nome    | CFCli        | CertMedico | TipoCli |
+-----+-----+-----+-----+-----+
|     1  | Di Renzo | Michele | ABCDEFGHIJKLMOP |          1 | Sala pesi |
|     2  | Agresta  | Arianna | ABCDEFGHIJKLMOP |          1 | Corso   |
|     3  | Rossi    | Valentino | AQDERFGHIJKLMOP |          1 | Sala pesi |
+-----+-----+-----+-----+-----+
```

2. Inserimento nuovo dipendente

```
insert into dipendenti(CFDip,Cognome,Nome,Stipendio,DataN)
values(...);
```

```

MariaDB [progetto_ufficiale]> insert into dipendenti(CFDip,Cognome,Nome,Stipendio,DataN)
->      values('ASDFGHJKLQWERTYU','Rossi','Mario',1220.13,'1990-06-29');
Query OK, 1 row affected (0.032 sec)

MariaDB [progetto_ufficiale]> insert into dipendenti(CFDip,Cognome,Nome,Stipendio,DataN)
->      values('ASDFGHJKLOLHVBYU','Neri','Carlo',1225.44,'1976-04-29');
Query OK, 1 row affected (0.019 sec)

MariaDB [progetto_ufficiale]> insert into dipendenti(CFDip,Cognome,Nome,Stipendio,DataN)
->      values('ASDFGHJKLASDCFVG','Verdi','Giuseppe',1025.74,'1993-11-09');
Query OK, 1 row affected (0.016 sec)

MariaDB [progetto_ufficiale]> select * from dipendenti;
+-----+-----+-----+-----+
| CFDip | Cognome | Nome   | Stipendio | DataN   |
+-----+-----+-----+-----+
| ASDFGHJKLASDCFVG | Verdi   | Giuseppe | 1025.74  | 1993-11-09 |
| ASDFGHJKLOLHVBYU | Neri    | Carlo   | 1225.44  | 1976-04-29 |
| ASDFGHJKLQWERTYU | Rossi   | Mario   | 1220.13  | 1990-06-29 |
+-----+-----+-----+-----+

```

3. Inserimento della scheda tecnica di un cliente

```
insert into schedatecnica(Data, Peso, Altezza, Grasso, Muscolo, Dip, Cliente) values(...);
```

```

MariaDB [progetto_ufficiale]> insert into schedatecnica(Data, Peso, Altezza, Grasso, Muscolo, Dip, Cliente)
->      values(CURDATE(), 86.4,187,12,60, 'ASDFGHJKLOLHVBYU', 1);
Query OK, 1 row affected (0.013 sec)

MariaDB [progetto_ufficiale]> select * from schedatecnica;
+-----+-----+-----+-----+-----+-----+-----+
| IdScheda | Data     | Peso  | Altezza | Grasso | Muscolo | Dip       | Cliente |
+-----+-----+-----+-----+-----+-----+-----+
| 2        | 2022-01-13 | 86.4  | 187     | 12     | 60      | ASDFGHJKLOLHVBYU | 1         |
+-----+-----+-----+-----+-----+-----+-----+

```

4. Inserimento transazione

```
insert into transazioni(Tipo,Causale,ImportoEnt,Data, Dipendente) values (...);
```

```

MariaDB [progetto_ufficiale]> insert into transazioni(Tipo,Causale,ImportoEnt,Data, Dipendente)
->      values ('entrata', 'Pagamento abbonamento giornaliero', 7,CURDATE() - INTERVAL 1 DAY,'ASDFGHJKLASDCFVG');
Query OK, 1 row affected (0.016 sec)

MariaDB [progetto_ufficiale]> select * from transazioni;
+-----+-----+-----+-----+-----+-----+-----+
| IdTrans | Tipo   | Causale           | ImportoUsc | ImportoEnt | Data       | Dipendente |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | entrata | Pagamento abbonamento giornaliero | 0.00 | 7 | 2022-01-13 | ASDFGHJKLASDCFVG |
| 2 | entrata | Pagamento abbonamento corso | 0.00 | 40 | 2022-01-13 | ASDFGHJKLASDCFVG |
| 3 | entrata | Pagamento abbonamento mensile | 0.00 | 40 | 2022-01-13 | ASDFGHJKLASDCFVG |
| 4 | uscita  | Pagamento stipendio | 1025.74 | 0 | 2022-01-13 | ASDFGHJKLASDCFVG |
| 5 | uscita  | Pagamento stipendio | 1225.44 | 0 | 2022-01-13 | ASDFGHJKLASDCFVG |
| 6 | uscita  | Pagamento stipendio | 1220.13 | 0 | 2022-01-13 | ASDFGHJKLASDCFVG |
| 7 | entrata | Pagamento abbonamento giornaliero | 0.00 | 7 | 2022-01-12 | ASDFGHJKLASDCFVG |
+-----+-----+-----+-----+-----+-----+-----+

```

5. Inserimento nuovo corso

```

insert into corso(NomeCo)
values (...);

```

```

MariaDB [progetto_ufficiale]> insert into corso(NomeCo)
->      values ('Zumba');
Query OK, 1 row affected (0.039 sec)

MariaDB [progetto_ufficiale]> Select * from corso;
+-----+
| NomeCo |
+-----+
| Zumba  |
+-----+

```

6. Inserimento abbonamento di un cliente

```

insert into abbonamenti(Cliente,Tipo,DataIni,DataSca,Dip,Transaz,Prezzo,Stato)
values(...);

```

```

MariaDB [progetto_ufficiale]> insert into abbonamenti(Cliente,Tipo,DataIni,DataSca,Dip,Transaz,Prezzo,Stato)
->      values(1, 'giornaliero',CURDATE() - INTERVAL 1 DAY, CURDATE(), 'ASDFGHJKLASDCFVG', 7, 7,'valido')
Query OK, 1 row affected (0.021 sec)

MariaDB [progetto_ufficiale]> select * from abbonamenti;
+-----+-----+-----+-----+-----+-----+-----+-----+
| IdAbb | Cliente | Tipo     | DataIni | DataSca | Dip          | Transaz | Prezzo | Stato  |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | 1 | giornaliero | 2022-01-13 | 2022-01-14 | ASDFGHJKLASDCFVG | 1 | 7 | valido |
| 3 | 2 | corso | 2022-01-13 | 2022-02-13 | ASDFGHJKLASDCFVG | 2 | 40 | valido |
| 4 | 3 | mensile | 2022-01-13 | 2022-02-13 | ASDFGHJKLASDCFVG | 3 | 40 | valido |
| 5 | 1 | giornaliero | 2022-01-12 | 2022-01-13 | ASDFGHJKLASDCFVG | 7 | 7 | valido |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

7. Aggiornamento stato abbonamento di un cliente già esistente

```
update abbonamenti set Stato = 'scaduto' where DataSca <= CURDATE();
```

```
MariaDB [progetto_ufficiale]> update abbonamenti set Stato = 'scaduto' where DataSca <= CURDATE();
Query OK, 1 row affected (0.022 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
MariaDB [progetto_ufficiale]> select * from abbonamenti;
```

IdAbb	Cliente	Tipo	DataIni	DataSca	Dip	Transaz	Prezzo	Stato
1	1	giornaliero	2022-01-13	2022-01-14	ASDFGHJKLASDCFVG	1	7	valido
3	2	corso	2022-01-13	2022-02-13	ASDFGHJKLASDCFVG	2	40	valido
4	3	mensile	2022-01-13	2022-02-13	ASDFGHJKLASDCFVG	3	40	valido
5	1	giornaliero	2022-01-12	2022-01-13	ASDFGHJKLASDCFVG	7	7	scaduto

8. Aggiornamento saldo della cassa

```
select sum(ImportoEnt - ImportoUsc) as Saldocassa from transazioni;
```

```
MariaDB [progetto_ufficiale]> select sum(ImportoEnt - ImportoUsc) as Saldo_cassa from transazioni;
+-----+
| Saldo_cassa |
+-----+
|    -3377.31 |
+-----+
```

9. Aggiornamento contatore dei clienti prenotati ad una determinata fascia oraria di un determinato giorno

```
select OraIn, OraFin, count(Cliente) as NumClienti from prenotazioni
where (Data = '2022-01-18' and OraIn = '08:00:00');
```

Oppure per una fascia oraria più ampia:

```
select count(Cliente) as NumClienti from prenotazioni where (Data =
'aaaa-mm-gg' and OraIn >= 'hh:mm:ss' and OraFin <= 'hh:mm:ss');
```

```
MariaDB [progetto_ufficiale]> select OraIn, OraFin, count(Cliente) as NumClienti from prenotazioni where
(Data = '2022-01-18' and OraIn = '08:00:00');
+-----+-----+
| OraIn | OraFin | NumClienti |
+-----+-----+
| 08:00:00 | 09:30:00 |      2 |
+-----+-----+
```

10. Modifica dati di un cliente

update clienti set campodamodificare = valoreconcuisimodifica where valoredelcamposucuifareverifica = 'condizione dove si vuole cambiare';

11. Modifica dati di un dipendente

update dipendenti set campodamodificare = valoreconcuisimodifica where valoredelcamposucuifareverifica = 'condizione dove si vuole cambiare';

12. Modifica dati transazione

update transazione set campodamodificare = valoreconcuisimodifica where valoredelcamposucuifareverifica = 'condizione dove si vuole cambiare';

13. Eliminazione di un cliente

delete from clienti where condizione;

```
MariaDB [progetto_ufficiale]> delete from clienti where CodCli = 1;
Query OK, 1 row affected (0.039 sec)

MariaDB [progetto_ufficiale]> select * from clienti;
+-----+-----+-----+-----+-----+
| CodCli | Cognome | Nome   | CFcli  | CertMedico | TipoCli |
+-----+-----+-----+-----+-----+
|     2  | Agresta  | Arianna | ABCDEFGHIJKLMNOPQRSTUVWXYZ |      1 | Corso    |
|     3  | Rossi    | Valentino | ABCDEFGHIJKLMNOPQRSTUVWXYZ |      1 | Sala pesi |
+-----+-----+-----+-----+-----+
2 rows in set (0.000 sec)

MariaDB [progetto_ufficiale]> select * from abbonamenti;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| IdAbb | Cliente | Tipo   | DataIni | DataSca | Dip       | Transaz | Prezzo | Stato  |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|     3  |          2 | corso  | 2022-01-13 | 2022-02-13 | ASDFGHJKLMNOPQRSTUVWXYZ |      2 |    40 | valido |
|     4  |          3 | mensile | 2022-01-13 | 2022-02-13 | ASDFGHJKLMNOPQRSTUVWXYZ |      3 |    40 | valido |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.000 sec)

MariaDB [progetto_ufficiale]> select * from prenotazioni;
+-----+-----+-----+-----+-----+-----+
| IdPre | Cliente | Dip        | Data      | OraIn    | OraFin    | Tipo     |
+-----+-----+-----+-----+-----+-----+
|     1  |          2 | ASDFGHJKLMNOPQRSTUVWXYZ | 2022-01-18 | 08:00:00 | 09:30:00 | Corso    |
|     3  |          3 | ASDFGHJKLMNOPQRSTUVWXYZ | 2022-01-18 | 08:00:00 | 09:30:00 | Sala pesi |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.000 sec)

MariaDB [progetto_ufficiale]> select * from transazioni;
+-----+-----+-----+-----+-----+-----+-----+-----+
| IdTrans | Tipo   | Causale           | ImportoUsc | ImportoEnt | Data      | Dipendente |
+-----+-----+-----+-----+-----+-----+-----+
|     1  | entrata | Pagamento abbonamento giornaliero |     0.00 |     7 | 2022-01-13 | ASDFGHJKLMNOPQRSTUVWXYZ |
|     2  | entrata | Pagamento abbonamento corso   |     0.00 |    40 | 2022-01-13 | ASDFGHJKLMNOPQRSTUVWXYZ |
|     3  | entrata | Pagamento abbonamento mensile |     0.00 |    40 | 2022-01-13 | ASDFGHJKLMNOPQRSTUVWXYZ |
|     4  | uscita  | Pagamento stipendio      | 1025.74 |     0 | 2022-01-13 | ASDFGHJKLMNOPQRSTUVWXYZ |
|     5  | uscita  | Pagamento stipendio      | 1225.44 |     0 | 2022-01-13 | ASDFGHJKLMNOPQRSTUVWXYZ |
|     6  | uscita  | Pagamento stipendio      | 1220.13 |     0 | 2022-01-13 | ASDFGHJKLMNOPQRSTUVWXYZ |
|    10  | entrata | Pagamento abbonamento giornaliero |     0.00 |    7 | 2022-01-12 | ASDFGHJKLMNOPQRSTUVWXYZ |
+-----+-----+-----+-----+-----+-----+-----+
```

14. Eliminazione di un dipendente

delete from dipendenti where condizione;

15. Eliminazione di un corso

```
delete from corso where NomeCo = 'Zumba';
```

```
MariaDB [progetto_ufficiale]> delete from corso where NomeCo = 'Zumba';
Query OK, 1 row affected (0.014 sec)

MariaDB [progetto_ufficiale]> select * from corso;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual
MariaDB [progetto_ufficiale]> select * from corso;
Empty set (0.000 sec)

MariaDB [progetto_ufficiale]> select * from istruttore;
+-----+-----+
| CFDip | NomeCo |
+-----+-----+
| ASDFGHJKLASTNCPI | NULL |
+-----+-----+
```

16. Eliminazione di una prenotazione per Id

```
delete from prenotazioni where IdPre = 1;
```

```
MariaDB [progetto_ufficiale]> select * from prenotazioni;
+-----+-----+-----+-----+-----+-----+
| IdPre | Cliente | Dip | Data | OraIn | OraFin | Tipo |
+-----+-----+-----+-----+-----+-----+
| 1 | 5 | ASDFGHJKLASDCCBM | 2022-01-18 | 08:00:00 | 09:30:00 | Sala pesi |
| 2 | 5 | ASDFGHJKLASDCCBM | 2022-01-20 | 11:00:00 | 12:30:00 | Sala pesi |
| 3 | 4 | ASDFGHJKLASDCCBM | 2022-01-14 | 11:00:00 | 12:30:00 | Sala pesi |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.000 sec)

MariaDB [progetto_ufficiale]> delete from prenotazioni where IdPre = 1;
Query OK, 1 row affected (0.011 sec)

MariaDB [progetto_ufficiale]> select * from prenotazioni;
+-----+-----+-----+-----+-----+-----+
| IdPre | Cliente | Dip | Data | OraIn | OraFin | Tipo |
+-----+-----+-----+-----+-----+-----+
| 2 | 5 | ASDFGHJKLASDCCBM | 2022-01-20 | 11:00:00 | 12:30:00 | Sala pesi |
| 3 | 4 | ASDFGHJKLASDCCBM | 2022-01-14 | 11:00:00 | 12:30:00 | Sala pesi |
+-----+-----+-----+-----+-----+-----+
```

17. Visualizzazione profilo di un cliente

```
select * from clienti where CodCli = '7';
```

```
MariaDB [progetto_ufficiale]> select * from clienti where CodCli = '7';
+-----+-----+-----+-----+-----+
| CodCli | Cognome | Nome | CFCli | CertMedico | TipoCli |
+-----+-----+-----+-----+-----+
| 7 | Azzurri | Luca | APOIUYTAEGHMERFG | 1 | Corso |
+-----+-----+-----+-----+-----+
```

18. Visualizzazione profilo di un dipendente

```
select * from dipendenti where CFDip = 'ASDFGHJKLASDCFVG';
```

```
MariaDB [progetto_ufficiale]> select * from dipendenti where CFDip = 'ASDFGHJKLASDCFVG';
+-----+-----+-----+-----+
| CFDip | Cognome | Nome   | Stipendio | DataN   |
+-----+-----+-----+-----+
| ASDFGHJKLASDCFVG | Verdi    | Giuseppe | 1025.74  | 1993-11-09 |
+-----+-----+-----+-----+
```

19. Visualizzazione abbonamento in corso di validità di un cliente

Prima di fare questa operazione è necessario effettuare l'operazione 7

```
select * from abbonamenti where (Cliente = 5 and Stato = 'valido');
```

```
MariaDB [progetto_ufficiale]> update abbonamenti set Stato = 'scaduto' where DataSca <= CURDATE();
Query OK, 1 row affected (0.009 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [progetto_ufficiale]> select * from abbonamenti;
+-----+-----+-----+-----+-----+-----+-----+-----+
| IdAbb | Cliente | Tipo      | DataIni  | DataSca  | Dip       | Transaz | Prezzo | Stato   |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 3     | 4        | giornaliero | 2022-01-13 | 2022-01-14 | ASDFGHJKLASDCCBM | 1        | 7       | scaduto |
| 4     | 6        | mensile    | 2022-01-12  | 2022-02-12  | ASDFGHJKLASDCCBM | 2        | 40      | valido  |
| 5     | 7        | corso     | 2022-01-09  | 2022-02-09  | ASDFGHJKLASDCCBM | 3        | 40      | valido  |
| 6     | 8        | corso     | 2022-01-12  | 2022-02-12  | ASDFGHJKLASDCCBM | 4        | 40      | valido  |
| 8     | 5        | mensile    | 2022-01-14  | 2022-02-14  | ASDFGHJKLASDCCBM | 14       | 40      | valido  |
+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.001 sec)

MariaDB [progetto_ufficiale]> select * from abbonamenti where (Cliente = 5 and Stato = 'valido');
+-----+-----+-----+-----+-----+-----+-----+-----+
| IdAbb | Cliente | Tipo      | DataIni  | DataSca  | Dip       | Transaz | Prezzo | Stato   |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 8     | 5        | mensile    | 2022-01-14 | 2022-02-14 | ASDFGHJKLASDCCBM | 14       | 40      | valido  |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.010 sec)

MariaDB [progetto_ufficiale]> select * from abbonamenti where (Cliente = 4 and Stato = 'valido');
Empty set (0.001 sec)
```

20. Visualizzazione scheda tecnica di un cliente

```
select * from schedatecnica where Cliente = 6;
```

```
MariaDB [progetto_ufficiale]> select * from schedatecnica where Cliente = 6;
+-----+-----+-----+-----+-----+-----+-----+
| IdScheda | Data      | Peso   | Altezza | Grasso | Muscolo | Dip       | Cliente |
+-----+-----+-----+-----+-----+-----+-----+
| 3       | 2022-01-12 | 86.4  | 187    | 12     | 60      | ASDFGHJKLASDCFVG | 6       |
+-----+-----+-----+-----+-----+-----+-----+
```

21. Conto delle prenotazioni di una determinata fascia orario in un determinato giorno

```
select Data, OraIn, OraFin, count(IdPre) as NumeroPre from prenotazioni where (Data = '2022-01-20' and OraIn = '11:00:00');
```

```
+-----+-----+-----+-----+-----+-----+
| IdPre | Cliente | Dip | Data | OraIn | OraFin | Tipo |
+-----+-----+-----+-----+-----+-----+
| 2 | 5 | ASDFGHJKLASDCCBM | 2022-01-20 | 11:00:00 | 12:30:00 | Sala pesi |
| 3 | 4 | ASDFGHJKLASDCCBM | 2022-01-14 | 11:00:00 | 12:30:00 | Sala pesi |
| 4 | 6 | ASDFGHJKLASDCCBM | 2022-01-20 | 11:00:00 | 12:30:00 | Sala pesi |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.000 sec)

MariaDB [progetto_ufficiale]> select Data, OraIn, OraFin, count(IdPre) as NumeroPre from prenotazioni where (Data = '2022-01-20' and OraIn = '11:00:00');
+-----+-----+-----+-----+
| Data | OraIn | OraFin | NumeroPre |
+-----+-----+-----+-----+
| 2022-01-20 | 11:00:00 | 12:30:00 | 2 |
+-----+-----+-----+
```

22. Visualizzazione stato di un abbonamento

```
select Stato from abbonamenti where IdAbb = 4;
```

```
MariaDB [progetto_ufficiale]> select Stato from abbonamenti where IdAbb = 4;
+-----+
| Stato |
+-----+
| valido |
+-----+
```

23. Visualizza dettagli di una transazione

```
select * from transazioni where IdTrans = 3;
```

```
MariaDB [progetto_ufficiale]> select * from transazioni where IdTrans = 3;
+-----+-----+-----+-----+-----+-----+-----+
| IdTrans | Tipo | Causale | ImportoUsc | ImportoEnt | Data | Dipendente |
+-----+-----+-----+-----+-----+-----+-----+
| 3 | entrata | Pagamento abbonamento corso | 0.00 | 40 | 2022-01-09 | ASDFGHJKLASDCCBM |
+-----+-----+-----+-----+-----+-----+
```

24. Statistiche (miglioramento/peggioramento) massa grassa

```
select Data, Grasso from schedatecnica where Cliente = 6;
```

```

+-----+-----+-----+-----+-----+-----+-----+-----+
| IdScheda | Data      | Peso   | Altezza | Grasso | Muscolo | Dip       | Cliente |
+-----+-----+-----+-----+-----+-----+-----+-----+
|     3 | 2022-01-12 | 86.4   |    187   |    12   |     60   | ASDFGHJKLASDCFVG |      6 |
|     5 | 2022-01-14 | 83.5   |    183   |    10   |     67   | NULL      |      5 |
|     7 | 2021-12-12 | 90.0   |    187   |    14   |     58   | ASDFGHJKLASDCFVG |      6 |
+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.000 sec)

MariaDB [progetto_ufficiale]> select Data, Grasso from schedatecnica where Cliente = 6;
+-----+-----+
| Data      | Grasso |
+-----+-----+
| 2022-01-12 |    12   |
| 2021-12-12 |    14   |
+-----+-----+

```

25. **Statistiche (miglioramento/peggioramento) massa muscolare**
`select Data, Muscolo from schedatecnica where Cliente = 6;`

```

MariaDB [progetto_ufficiale]> select Data, Muscolo from schedatecnica where Cliente = 6;
+-----+-----+
| Data      | Muscolo |
+-----+-----+
| 2022-01-12 |     60   |
| 2021-12-12 |     58   |
+-----+-----+

```

26. **Filtraggio schede tecniche per altezza e peso**
`select * from schedatecnica where (Altezza > 185 and Peso > 87);`

```

+-----+-----+-----+-----+-----+-----+-----+-----+
| IdScheda | Data      | Peso   | Altezza | Grasso | Muscolo | Dip       | Cliente |
+-----+-----+-----+-----+-----+-----+-----+-----+
|     3 | 2022-01-12 | 86.4   |    187   |    12   |     60   | ASDFGHJKLASDCFVG |      6 |
|     5 | 2022-01-14 | 83.5   |    183   |    10   |     67   | NULL      |      5 |
|     7 | 2021-12-12 | 90.0   |    187   |    14   |     58   | ASDFGHJKLASDCFVG |      6 |
+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.000 sec)

MariaDB [progetto_ufficiale]> select * from schedatecnica where (Altezza > 185 and Peso > 87);
+-----+-----+-----+-----+-----+-----+-----+
| IdScheda | Data      | Peso   | Altezza | Grasso | Muscolo | Dip       | Cliente |
+-----+-----+-----+-----+-----+-----+-----+-----+
|     7 | 2021-12-12 | 90.0   |    187   |    14   |     58   | ASDFGHJKLASDCFVG |      6 |
+-----+-----+-----+-----+-----+-----+-----+

```

27. **Numero di abbonamenti che un cliente possiede**
`select CodCli, Cognome, Nome, count(Cliente) as NumeroAbbonamenti from clienti inner join abbonamenti where (CodCli = 6 and CodCli = Cliente);`

IdAbb	Cliente	Tipo	DataIni	DataSca	Dip	Transaz	Prezzo	Stato
3	4	giornaliero	2022-01-13	2022-01-14	ASDFGHJKLASDCCBM	1	7	scaduto
4	6	mensile	2022-01-12	2022-02-12	ASDFGHJKLASDCCBM	2	40	valido
5	7	corso	2022-01-09	2022-02-09	ASDFGHJKLASDCCBM	3	40	valido
6	8	corso	2022-01-12	2022-02-12	ASDFGHJKLASDCCBM	4	40	valido
8	5	mensile	2022-01-14	2022-02-14	ASDFGHJKLASDCCBM	14	40	valido
9	6	mensile	2021-12-12	2022-01-12	ASDFGHJKLASDCCBM	15	40	scaduto
10	6	mensile	2021-11-12	2021-12-12	ASDFGHJKLASDCCBM	16	40	scaduto

7 rows in set (0.000 sec)

```
MariaDB [progetto_ufficiale]> select CodCli, Cognome, Nome, count(Cliente) as NumeroAbbonamenti from clienti inner join abbonamenti where (CodCli = 6 and CodCli = Cliente);
+-----+-----+-----+-----+
| CodCli | Cognome | Nome | NumeroAbbonamenti |
+-----+-----+-----+-----+
| 6 | Agresta | Arianna | 3 |
+-----+-----+-----+-----+
```

28. Filtraggio schede tecniche per nome, cognome

```
select schedatecnica.*, Nome, Cognome from schedatecnica inner join
clienti where (Nome = 'Arianna' and CodCli = Cliente);
```

IdScheda	Data	Peso	Altezza	Grasso	Muscolo	Dip	Cliente
3	2022-01-12	86.4	187	12	60	ASDFGHJKLASDCFVG	6
5	2022-01-14	83.5	183	10	67	NULL	5
7	2021-12-12	90.0	187	14	58	ASDFGHJKLASDCFVG	6
8	2022-01-14	85.4	177	12	55	ASDFGHJKLASDCFVG	10

4 rows in set (0.000 sec)

```
MariaDB [progetto_ufficiale]> select schedatecnica.*, Nome, Cognome from schedatecnica inner join clienti where (Nome = 'Arianna' and CodCli = Cliente);
+-----+-----+-----+-----+-----+-----+-----+-----+
| IdScheda | Data | Peso | Altezza | Grasso | Muscolo | Dip | Cliente | Nome | Cognome |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 3 | 2022-01-12 | 86.4 | 187 | 12 | 60 | ASDFGHJKLASDCFVG | 6 | Arianna | Agresta |
| 7 | 2021-12-12 | 90.0 | 187 | 14 | 58 | ASDFGHJKLASDCFVG | 6 | Arianna | Agresta |
| 8 | 2022-01-14 | 85.4 | 177 | 12 | 55 | ASDFGHJKLASDCFVG | 10 | Arianna | Giallo |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

29. Filtraggio clienti per nome, cognome

```
select * from clienti where (Nome = 'Arianna');
```

Ottiene:

```
select * from clienti where Nome like 'M%';
```

```

+-----+-----+-----+-----+-----+-----+
| CodCli | Cognome | Nome   | CFcli   | CertMedico | TipoCli |
+-----+-----+-----+-----+-----+-----+
|      4 | Rossi    | Valentino | AQDERFGHTJKLTYUI |          1 | Sala pesi |
|      5 | Di Renzo  | Michele   | AQDERFGHIJASERFG |          1 | Sala pesi |
|      6 | Agresta   | Arianna   | APOIUYTHIJASERFG |          1 | Corso    |
|      7 | Azzurri   | Luca     | APOIUYTAEGHMERFG |          1 | Corso    |
|      8 | Grigi     | Marta    | APOIUYTAEGHMROKH |          1 | Corso    |
|     10 | Giallo   | Arianna   | EHNIUYTQWEHMROKH |          1 | Sala pesi |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.001 sec)

MariaDB [progetto_ufficiale]> select * from clienti where (Nome = 'Arianna');
+-----+-----+-----+-----+-----+
| CodCli | Cognome | Nome   | CFcli   | CertMedico | TipoCli |
+-----+-----+-----+-----+-----+
|      6 | Agresta  | Arianna | APOIUYTHIJASERFG |          1 | Corso    |
|     10 | Giallo   | Arianna | EHNIUYTQWEHMROKH |          1 | Sala pesi |
+-----+-----+-----+-----+-----+
2 rows in set (0.000 sec)

MariaDB [progetto_ufficiale]> select * from clienti where Nome like 'M%';
+-----+-----+-----+-----+-----+
| CodCli | Cognome | Nome   | CFcli   | CertMedico | TipoCli |
+-----+-----+-----+-----+-----+
|      5 | Di Renzo | Michele | AQDERFGHIJASERFG |          1 | Sala pesi |
|      8 | Grigi    | Marta   | APOIUYTAEGHMROKH |          1 | Corso    |
+-----+-----+-----+-----+-----+

```

30. Filtraggio abbonamenti per nome, cognome

select abbonamenti.* , Nome, Cognome from abbonamenti inner join clienti where (Nome = 'Arianna' and CodCli = Cliente);

```

MariaDB [progetto_ufficiale]> select abbonamenti.* , Nome, Cognome from abbonamenti inner join clienti where (Nome = 'Arianna' and CodCli = Cliente);
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| IdAbb | Cliente | Tipo   | DataIni | DataSca | Dip    | Transaz | Prezzo | Stato  | Nome   | Cognome |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      4 |       6 | mensile | 2022-01-12 | 2022-02-12 | ASDFGHJKLASDCCBM |      2 |    40 | valido | Arianna | Agresta |
|      9 |       6 | mensile | 2021-12-12 | 2022-01-12 | ASDFGHJKLASDCCBM |     15 |    40 | scaduto | Arianna | Agresta |
|     10 |       6 | mensile | 2021-11-12 | 2021-12-12 | ASDFGHJKLASDCCBM |     16 |    40 | scaduto | Arianna | Agresta |
|     12 |      10 | mensile | 2022-01-14 | 2022-02-14 | ASDFGHJKLASDCCBM |     17 |    40 | valido | Arianna | Giallo  |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

31. Filtraggio dipendenti per nome, cognome

select * from dipendenti where Nome = 'Giuseppe';

```

+-----+-----+-----+-----+-----+
| CFDip      | Cognome | Nome   | Stipendio | DataN   |
+-----+-----+-----+-----+-----+
| ASDFGHJKLASDCCBM | Neri    | Carla   | 1035.44  | 1983-10-09 |
| ASDFGHJKLASDCFVG | Verdi   | Giuseppe | 1025.74  | 1993-11-09 |
| ASDFGHJKLASTNCPI | Rosso   | Mario   | 1050.71  | 1976-05-23 |
+-----+-----+-----+-----+-----+
3 rows in set (0.000 sec)

MariaDB [progetto_ufficiale]> select * from dipendenti where Nome = 'Giuseppe';
+-----+-----+-----+-----+-----+
| CFDip      | Cognome | Nome   | Stipendio | DataN   |
+-----+-----+-----+-----+-----+
| ASDFGHJKLASDCFVG | Verdi   | Giuseppe | 1025.74  | 1993-11-09 |
+-----+-----+-----+-----+-----+

```

32. Filtraggio dipendenti per ruolo

```
select dipendenti.* from personaltrainer inner join dipendenti where  
(dipendenti.CFDip = personaltrainer.CFDip);
```

```
MariaDB [progetto_ufficiale]> select dipendenti.* from personaltrainer inner join dipendenti where (dipendenti.CFDip = personaltrainer.CFDip);  
+-----+-----+-----+-----+  
| CFDip | Cognome | Nome | Stipendio | DataN |  
+-----+-----+-----+-----+  
| ASDFGHJKLASDCFVG | Verdi | Giuseppe | 1025.74 | 1993-11-09 |  
+-----+-----+-----+-----+
```

33. Conteggio clienti entrati in palestra in una giornata

```
select Data, count(IdPre) as NumeroIngressi from prenotazioni where  
Data = '2022-01-20';
```

```
MariaDB [progetto_ufficiale]> select * from prenotazioni;  
+-----+-----+-----+-----+-----+-----+  
| IdPre | Cliente | Dip | Data | OraIn | OraFin | Tipo |  
+-----+-----+-----+-----+-----+-----+  
| 2 | 5 | ASDFGHJKLASDCCBM | 2022-01-20 | 11:00:00 | 12:30:00 | Sala pesi |  
| 3 | 4 | ASDFGHJKLASDCCBM | 2022-01-14 | 11:00:00 | 12:30:00 | Sala pesi |  
| 4 | 6 | ASDFGHJKLASDCCBM | 2022-01-20 | 11:00:00 | 12:30:00 | Sala pesi |  
+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.000 sec)  
  
MariaDB [progetto_ufficiale]> select Data, count(IdPre) as NumeroIngressi from prenotazioni where Data = '2022-01-20';  
+-----+-----+  
| Data | NumeroIngressi |  
+-----+-----+  
| 2022-01-20 | 2 |  
+-----+-----+
```

34. Nome e cognome clienti abbonati validi

```
select Nome, Cognome, IdAbb from clienti inner join abbonamenti  
where (CodCli = Cliente and Stato = 'valido');
```

```

MariaDB [progetto_ufficiale]> select * from abbonamenti;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| IdAbb | Cliente | Tipo    | DataIni   | DataSca   | Dip      | Transaz | Prezzo | Stato   |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 3     | 4       | giornaliero | 2022-01-13 | 2022-01-14 | ASDFGHJKLASDCCBM | 1 | 7 | scaduto |
| 4     | 6       | mensile   | 2022-01-12 | 2022-02-12 | ASDFGHJKLASDCCBM | 2 | 40 | valido  |
| 5     | 7       | corso    | 2022-01-09 | 2022-02-09 | ASDFGHJKLASDCCBM | 3 | 40 | valido  |
| 6     | 8       | corso    | 2022-01-12 | 2022-02-12 | ASDFGHJKLASDCCBM | 4 | 40 | valido  |
| 8     | 5       | mensile   | 2022-01-14 | 2022-02-14 | ASDFGHJKLASDCCBM | 14 | 40 | valido  |
| 9     | 6       | mensile   | 2021-12-12 | 2022-01-12 | ASDFGHJKLASDCCBM | 15 | 40 | scaduto |
| 10    | 6       | mensile   | 2021-11-12 | 2021-12-12 | ASDFGHJKLASDCCBM | 16 | 40 | scaduto |
| 12    | 10      | mensile   | 2022-01-14 | 2022-02-14 | ASDFGHJKLASDCCBM | 17 | 40 | valido  |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
8 rows in set (0.000 sec)

MariaDB [progetto_ufficiale]> select Nome, Cognome, IdAbb from clienti inner join abbonamenti where (CodCli = Cliente and Stato = 'valido');
+-----+-----+-----+
| Nome   | Cognome | IdAbb |
+-----+-----+-----+
| Arianna | Agresta | 4 |
| Luca   | Azzurri | 5 |
| Marta  | Grigi   | 6 |
| Michele | Di Renzo | 8 |
| Arianna | Giallo  | 12 |
+-----+-----+-----+

```

35. Conteggiare il numero di abbonamenti sottoscritti il mese corrente

```
select DataIni, count(IdAbb) as NumeroAbb from abbonamenti where MONTH(DataIni) = MONTH(CURDATE());
```

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| IdAbb | Cliente | Tipo    | DataIni   | DataSca   | Dip      | Transaz | Prezzo | Stato   |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 3     | 4       | giornaliero | 2022-01-13 | 2022-01-14 | ASDFGHJKLASDCCBM | 1 | 7 | scaduto |
| 4     | 6       | mensile   | 2022-01-12 | 2022-02-12 | ASDFGHJKLASDCCBM | 2 | 40 | valido  |
| 5     | 7       | corso    | 2022-01-09 | 2022-02-09 | ASDFGHJKLASDCCBM | 3 | 40 | valido  |
| 6     | 8       | corso    | 2022-01-12 | 2022-02-12 | ASDFGHJKLASDCCBM | 4 | 40 | valido  |
| 8     | 5       | mensile   | 2022-01-14 | 2022-02-14 | ASDFGHJKLASDCCBM | 14 | 40 | valido  |
| 9     | 6       | mensile   | 2021-12-12 | 2022-01-12 | ASDFGHJKLASDCCBM | 15 | 40 | scaduto |
| 10    | 6       | mensile   | 2021-11-12 | 2021-12-12 | ASDFGHJKLASDCCBM | 16 | 40 | scaduto |
| 12    | 10      | mensile   | 2022-01-14 | 2022-02-14 | ASDFGHJKLASDCCBM | 17 | 40 | valido  |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
8 rows in set (0.000 sec)

MariaDB [progetto_ufficiale]> select DataIni, count(IdAbb) as NumeroAbb from abbonamenti where MONTH(DataIni) = MONTH(CURDATE());
+-----+-----+
| DataIni | NumeroAbb |
+-----+-----+
| 2022-01-13 | 6 |
+-----+-----+

```

36. Conteggiare il numero di abbonamenti sottoscritti in un anno (gennaio-dicembre)

```
select DataIni, count(IdAbb) as NumeroAbb from abbonamenti where (DataIni between '2021-01-01' and '2021-12-31');
```

```

MariaDB [progetto_ufficiale]> select * from abbonamenti;
+-----+-----+-----+-----+-----+-----+-----+-----+
| IdAbb | Cliente | Tipo      | DataIni   | DataSca   | Dip       | Transaz | Prezzo | Stato   |
+-----+-----+-----+-----+-----+-----+-----+-----+
|    3 |      4 | giornaliero | 2022-01-13 | 2022-01-14 | ASDFGHJKLASDCCBM |     1 |    7 | scaduto |
|    4 |      6 | mensile    | 2022-01-12 | 2022-02-12 | ASDFGHJKLASDCCBM |     2 |   40 | valido  |
|    5 |      7 | corso     | 2022-01-09 | 2022-02-09 | ASDFGHJKLASDCCBM |     3 |   40 | valido  |
|    6 |      8 | corso     | 2022-01-12 | 2022-02-12 | ASDFGHJKLASDCCBM |     4 |   40 | valido  |
|    8 |      5 | mensile    | 2022-01-14 | 2022-02-14 | ASDFGHJKLASDCCBM |    14 |   40 | valido  |
|    9 |      6 | mensile    | 2021-12-12 | 2022-01-12 | ASDFGHJKLASDCCBM |    15 |   40 | scaduto |
|   10 |      6 | mensile    | 2021-11-12 | 2021-12-12 | ASDFGHJKLASDCCBM |    16 |   40 | scaduto |
|   12 |     10 | mensile    | 2022-01-14 | 2022-02-14 | ASDFGHJKLASDCCBM |    17 |   40 | valido  |
+-----+-----+-----+-----+-----+-----+-----+-----+
8 rows in set (0.000 sec)

MariaDB [progetto_ufficiale]> select DataIni, count(IdAbb) as NumeroAbb from abbonamenti where (DataIni between '2022-01-01' and '2022-12-31');
+-----+-----+
| DataIni | NumeroAbb |
+-----+-----+
| 2022-01-13 |      6 |
+-----+-----+
1 row in set (0.000 sec)

MariaDB [progetto_ufficiale]> select DataIni, count(IdAbb) as NumeroAbb from abbonamenti where (DataIni between '2021-01-01' and '2021-12-31');
+-----+-----+
| DataIni | NumeroAbb |
+-----+-----+
| 2021-12-12 |      2 |
+-----+-----+

```

37. Filtraggio abbonamenti per tipo di abbonamento

```
select * from abbonamenti where Tipo = 'mensile';
```

```

+-----+-----+-----+-----+-----+-----+-----+-----+
| IdAbb | Cliente | Tipo      | DataIni   | DataSca   | Dip       | Transaz | Prezzo | Stato   |
+-----+-----+-----+-----+-----+-----+-----+-----+
|    3 |      4 | giornaliero | 2022-01-13 | 2022-01-14 | ASDFGHJKLASDCCBM |     1 |    7 | scaduto |
|    4 |      6 | mensile    | 2022-01-12 | 2022-02-12 | ASDFGHJKLASDCCBM |     2 |   40 | valido  |
|    5 |      7 | corso     | 2022-01-09 | 2022-02-09 | ASDFGHJKLASDCCBM |     3 |   40 | valido  |
|    6 |      8 | corso     | 2022-01-12 | 2022-02-12 | ASDFGHJKLASDCCBM |     4 |   40 | valido  |
|    8 |      5 | mensile    | 2022-01-14 | 2022-02-14 | ASDFGHJKLASDCCBM |    14 |   40 | valido  |
|    9 |      6 | mensile    | 2021-12-12 | 2022-01-12 | ASDFGHJKLASDCCBM |    15 |   40 | scaduto |
|   10 |      6 | mensile    | 2021-11-12 | 2021-12-12 | ASDFGHJKLASDCCBM |    16 |   40 | scaduto |
|   12 |     10 | mensile    | 2022-01-14 | 2022-02-14 | ASDFGHJKLASDCCBM |    17 |   40 | valido  |
+-----+-----+-----+-----+-----+-----+-----+-----+
8 rows in set (0.001 sec)

MariaDB [progetto_ufficiale]> select * from abbonamenti where Tipo = 'mensile';
+-----+-----+-----+-----+-----+-----+-----+
| IdAbb | Cliente | Tipo      | DataIni   | DataSca   | Dip       | Transaz | Prezzo | Stato   |
+-----+-----+-----+-----+-----+-----+-----+-----+
|    4 |      6 | mensile    | 2022-01-12 | 2022-02-12 | ASDFGHJKLASDCCBM |     2 |   40 | valido  |
|    8 |      5 | mensile    | 2022-01-14 | 2022-02-14 | ASDFGHJKLASDCCBM |    14 |   40 | valido  |
|    9 |      6 | mensile    | 2021-12-12 | 2022-01-12 | ASDFGHJKLASDCCBM |    15 |   40 | scaduto |
|   10 |      6 | mensile    | 2021-11-12 | 2021-12-12 | ASDFGHJKLASDCCBM |    16 |   40 | scaduto |
|   12 |     10 | mensile    | 2022-01-14 | 2022-02-14 | ASDFGHJKLASDCCBM |    17 |   40 | valido  |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

38. Filtraggio di nome, cognome dei clienti che non sono in possesso del certificato medico

```
select * from clienti where CertMedico = 0;
```

```

+-----+-----+-----+-----+-----+-----+
| CodCli | Cognome | Nome   | CFCli           | CertMedico | TipoCli |
+-----+-----+-----+-----+-----+-----+
|     4 | Rossi    | Valentino | AQDERFGHIJKLTYUI |         1 | Sala pesi |
|     5 | Di Renzo  | Michele   | AQDERFGHIJASERFG |         1 | Sala pesi |
|     6 | Agresta   | Arianna   | APOIUYTHIJASERFG |         1 | Corso      |
|     7 | Azzurri   | Luca      | APOIUYTAEGHMERFG |         1 | Corso      |
|     8 | Grigi     | Marta     | APOIUYTAEGHMROKH |         1 | Corso      |
|    10 | Giallo    | Arianna   | EHNIUYTQWEHMROKH |         1 | Sala pesi |
|    11 | Marrone   | Luca      | ASDFGHJKLASDCBM  |         0 | Sala pesi |
|    12 | Viola     | Giorgia  | ASDFGHJKLASDCBM  |         0 | Corso      |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.000 sec)

MariaDB [progetto_ufficiale]> select * from clienti where CertMedico = 0;
+-----+-----+-----+-----+-----+
| CodCli | Cognome | Nome   | CFCli           | CertMedico | TipoCli |
+-----+-----+-----+-----+-----+
|     11 | Marrone  | Luca   | ASDFGHJKLASDCBM |         0 | Sala pesi |
|     12 | Viola    | Giorgia | ASDFGHJKLASDCBM |         0 | Corso      |
+-----+-----+-----+-----+-----+

```

39. Inserimento prenotazione

```

insert into prenotazioni(Cliente,Dip,Data,OraIn,OraFin,Tipo)
values(4, 'ASDFGHJKLASDCBM', '2022-01-14', '11:00:00', OraIn +
INTERVAL 90 MINUTE, 'Sala pesi');

```

oppure:

```

insert into prenotazioni(Cliente,Dip,Data,OraIn,OraFin,Tipo)
values(1, 'ASDFGHJKLASDCFVG', '2022-01-13', '09:30:00', OraIn +
INTERVAL 90 MINUTE, (select TipoCli from clienti where CodCli =
1));

```

```

MariaDB [progetto_ufficiale]> insert into prenotazioni(Cliente,Dip,Data,OraIn,OraFin,Tipo)
    ->      values(2, 'ASDFGHJKLASDCFVG', '2022-01-18', '08:00:00', OraIn + INTERVAL 90 MINUTE, 'Corso');
Query OK, 1 row affected (0.020 sec)

MariaDB [progetto_ufficiale]> insert into prenotazioni(Cliente,Dip,Data,OraIn,OraFin,Tipo)
    ->      values(1, 'ASDFGHJKLASDCFVG', '2022-01-13', '09:30:00', OraIn + INTERVAL 90 MINUTE, (select TipoCli from clienti where CodCli = 1));
Query OK, 1 row affected (0.007 sec)

MariaDB [progetto_ufficiale]> insert into prenotazioni(Cliente,Dip,Data,OraIn,OraFin,Tipo)
    ->      values(3, 'ASDFGHJKLASDCFVG', '2022-01-18', '08:00:00', OraIn + INTERVAL 90 MINUTE, 'Sala pesi');
Query OK, 1 row affected (0.018 sec)

MariaDB [progetto_ufficiale]> select * from prenotazioni;
+-----+-----+-----+-----+-----+
| IdPre | Cliente | Dip      | Data     | OraIn   | OraFin   | Tipo    |
+-----+-----+-----+-----+-----+
| 1     | 2       | ASDFGHJKLASDCFVG | 2022-01-18 | 08:00:00 | 09:30:00 | Corso   |
| 2     | 1       | ASDFGHJKLASDCFVG | 2022-01-13 | 09:30:00 | 11:00:00 | Sala pesi|
| 3     | 3       | ASDFGHJKLASDCFVG | 2022-01-18 | 08:00:00 | 09:30:00 | Sala pesi|
+-----+-----+-----+-----+-----+

```

40. Inserimento numero di telefono

```

insert into telefono(Num, Cliente)
values('2123456789', 7);

```

```

MariaDB [progetto_ufficiale]> insert into telefono(Num, Cliente)
    -> values('1234567890', 4);
Query OK, 1 row affected (0.004 sec)

MariaDB [progetto_ufficiale]> insert into telefono(Num, Cliente)
    -> values('1234567891', 4);
Query OK, 1 row affected (0.017 sec)

MariaDB [progetto_ufficiale]> insert into telefono(Num, Cliente)
    -> values('1234567892', 4);
Query OK, 1 row affected (0.010 sec)

MariaDB [progetto_ufficiale]> insert into telefono(Num, Cliente)
    -> values('0123456789', 5);
Query OK, 1 row affected (0.004 sec)

MariaDB [progetto_ufficiale]> insert into telefono(Num, Cliente)
    -> values('2123456789', 7);
Query OK, 1 row affected (0.010 sec)

MariaDB [progetto_ufficiale]> select * from telefono;
+-----+-----+
| Num      | Cliente |
+-----+-----+
| 1234567890 | 4 |
| 1234567891 | 4 |
| 1234567892 | 4 |
| 0123456789 | 5 |
| 2123456789 | 7 |
+-----+-----+

```

41. Elimina numero di telefono

```

delete from telefono where Num = '1234567890';

```

```

+-----+-----+
| Num      | Cliente |
+-----+-----+
| 1234567890 |      4 |
| 1234567891 |      4 |
| 1234567892 |      4 |
| 0123456789 |      5 |
| 2123456789 |      7 |
+-----+-----+
5 rows in set (0.000 sec)

MariaDB [progetto_ufficiale]> delete from telefono where Num = '1234567890';
Query OK, 1 row affected (0.010 sec)

MariaDB [progetto_ufficiale]> select * from telefono;
+-----+-----+
| Num      | Cliente |
+-----+-----+
| 1234567891 |      4 |
| 1234567892 |      4 |
| 0123456789 |      5 |
| 2123456789 |      7 |
+-----+-----+

```

42. Inserimento nuovo esercizio

```

insert into esercizio(NomeEs)
value('Panca piana');

```

```

MariaDB [progetto_ufficiale]> insert into esercizio(NomeEs)
-> value('Leg press');
Query OK, 1 row affected (0.011 sec)

MariaDB [progetto_ufficiale]> insert into esercizio(NomeEs)
-> value('Piegamenti');
Query OK, 1 row affected (0.006 sec)

MariaDB [progetto_ufficiale]> insert into esercizio(NomeEs)
-> value('Affondi');
Query OK, 1 row affected (0.018 sec)

MariaDB [progetto_ufficiale]> insert into esercizio(NomeEs)
-> value('Lat machine');
Query OK, 1 row affected (0.019 sec)

MariaDB [progetto_ufficiale]> insert into esercizio(NomeEs)
-> value('Panca piana');
Query OK, 1 row affected (0.005 sec)

MariaDB [progetto_ufficiale]> select * from esercizio;
+-----+
| NomeEs   |
+-----+
| Affondi   |
| Lat machine |
| Leg press  |
| Panca piana |
| Piegamenti |
+-----+

```

43. Elimina esercizio

```
delete from esercizio where NomeEs = 'Affondi';
```

```
+-----+  
| NomeEs |  
+-----+  
| Affondi |  
| Lat machine |  
| Leg press |  
| Panca piana |  
| Piegamenti |  
+-----+  
5 rows in set (0.000 sec)  
  
MariaDB [progetto_ufficiale]> delete from esercizio where NomeEs = 'Affondi';  
Query OK, 1 row affected (0.014 sec)  
  
MariaDB [progetto_ufficiale]> select * from esercizio;  
+-----+  
| NomeEs |  
+-----+  
| Lat machine |  
| Leg press |  
| Panca piana |  
| Piegamenti |  
+-----+
```

44. Trovare il numero di telefono associato a un cliente per nome, cognome

```
select Num from telefono inner join clienti where (CodCli = Cliente  
and Nome = 'Valentino' and Cognome = 'Rossi');
```

```

+-----+-----+
| Num | Cliente |
+-----+-----+
| 1234567891 |      4 |
| 1234567892 |      4 |
| 0123456789 |      5 |
| 2123456789 |      7 |
+-----+
4 rows in set (0.000 sec)

MariaDB [progetto_ufficiale]> select * from clienti;
+-----+-----+-----+-----+-----+
| CodCli | Cognome | Nome   | CFCli    | CertMedico | TipoCli |
+-----+-----+-----+-----+-----+
| 4 | Rossi   | Valentino | AQDERFGHIJKLMNOP | 1 | Sala pesi |
| 5 | Di Renzo | Michele   | AQDERFGHIJKLMNOP | 1 | Sala pesi |
| 6 | Agresta  | Arianna   | APOIUYTHIJASERFG | 1 | Corso    |
| 7 | Azurri   | Luca      | APOIUYTAEGHMERFG | 1 | Corso    |
| 8 | Grigi    | Marta     | APOIUYTAEGHMRKHK | 1 | Corso    |
| 10 | Giallo   | Arianna   | EHNIVUTQWEHMROKH | 1 | Sala pesi |
| 11 | Marrone  | Luca      | ASDFGHJKLASDCBM | 0 | Sala pesi |
| 12 | Viola    | Giorgia  | ASDFGHJKLASDCBM | 0 | Corso    |
+-----+-----+-----+-----+-----+
8 rows in set (0.006 sec)

MariaDB [progetto_ufficiale]> select Num from telefono inner join clienti where (CodCli = Cliente and Nome = 'Valentino' and Cognome = 'Rossi');
+-----+
| Num |
+-----+
| 1234567891 |
| 1234567892 |
+-----+

```

45. Fare una classifica degli esercizi più svolti

select NomeEs, count(NomeEs) as Comparse from composizione group by NomeEs order by Comparse desc;

```

+-----+-----+
| NomeEs | IdScheda |
+-----+-----+
| Lat Machine | 3 |
| Lat machine | 5 |
| Lat machine | 7 |
| Leg press | 8 |
| Panca piana | 3 |
| Panca piana | 8 |
| Piegamenti | 3 |
| Piegamenti | 5 |
+-----+-----+
8 rows in set (0.000 sec)

MariaDB [progetto_ufficiale]> select NomeEs, count(NomeEs) as Comparse from composizione group by NomeEs order by Comparse desc;
+-----+-----+
| NomeEs | Comparse |
+-----+-----+
| Lat Machine | 3 |
| Panca piana | 2 |
| Piegamenti | 2 |
| Leg press | 1 |
+-----+-----+

```

46. Aggiungere esercizio in una scheda tecnica

```
insert into composizione(NomeEs,IdScheda)
value('Piegamenti', 7);
```

```
MariaDB [progetto_ufficiale]> insert into composizione(NomeEs,IdScheda)
-> value('Leg press', 8);
Query OK, 1 row affected (0.010 sec)

MariaDB [progetto_ufficiale]> select * from composizione order by IdScheda;
+-----+-----+
| NomeEs      | IdScheda |
+-----+-----+
| Lat Machine |      3 |
| Panca piana |      3 |
| Piegamenti   |      3 |
| Lat machine  |      5 |
| Piegamenti   |      5 |
| Lat machine  |      7 |
| Leg press    |      8 |
| Panca piana |      8 |
+-----+-----+
```

47. Inserimento segretaria

```
insert into segretaria(CFDip)
values(...);
```

```
MariaDB [progetto_ufficiale]> insert into segretaria(CFDip)
-> values('ASDFGHJKLASDCFVG');
Query OK, 1 row affected (0.030 sec)

MariaDB [progetto_ufficiale]> select * from segretaria;
+-----+
| CFDip        |
+-----+
| ASDFGHJKLASDCFVG |
+-----+
```

48. Inserimento personal trainer

```
insert into personaltrainer(CFDip)
values(...);
```

```
MariaDB [progetto_ufficiale]> insert into personaltrainer(CFDip)
      ->      values('ASDFGHJKLOLHVBYU');
Query OK, 1 row affected (0.017 sec)

MariaDB [progetto_ufficiale]> select * from personaltrainer;
+-----+
| CFDip |
+-----+
| ASDFGHJKLOLHVBYU |
+-----+
```

49. **Inserimento istruttore**

```
insert into istruttore(CFDip)
values(...);
```

```
MariaDB [progetto_ufficiale]> insert into istruttore(CFDip, NomeCo)
      ->      values('ASDFGHJKLQWERTYU', 'Zumba');
Query OK, 1 row affected (0.016 sec)

MariaDB [progetto_ufficiale]> select * from istruttore;
+-----+-----+
| CFDip | NomeCo |
+-----+-----+
| ASDFGHJKLQWERTYU | Zumba |
+-----+-----+
```