

PROGETTO DI UNA BASE DI DATI PER L'ASSOCIAZIONE SPORTIVA **ANCONA ROCK CLIMB**

Realizzato da:

Alessandro Minutillo, matricola S1092465

Davide Nunin, matricola S1092461

Codice gruppo revisione: 1514

INDICE

1) ANALISI DEI REQUISITI

- 1.1) Raccolta informazioni
 - 1.1.1) Intervista al titolare
 - 1.1.2) Intervista alla segreteria
 - 1.1.3) Esempio di modulistica
- 1.2) Requisiti espressi in linguaggio naturale
- 1.3) Strutturazione dei requisiti
- 1.4) Glossario dei termini
- 1.5) Specifica delle operazioni

2) PROGETTAZIONE CONCETTUALE

- 2.1) Procedimento
- 2.2) Identificazione delle entità e relazioni fondamentali
- 2.3) Scheletro dello schema
 - 2.3.1) Sviluppo componenti dello scheletro
- 2.4) Modello E-R complessivo
- 2.5) Analisi di qualità dello schema E-R
- 2.6) Dizionario dei dati
- 2.7) Vincoli di integrità dei dati

3) PROGETTAZIONE LOGICA

- 3.1) Tavola dei volumi e delle operazioni
 - 3.1.1) Tavola dei volumi
 - 3.1.2) Tavola delle operazioni
- 3.2) Analisi delle ridondanze
 - 3.2.1) Attributo “ingressi rimanenti” in “Abbonamento”
 - 3.2.2) Attributo “posti rimanenti” in “Sessione”
- 3.3) Eliminazione delle gerarchie
- 3.4) Partizionamento/Accorpamento di concetti

3.5) Eliminazione degli attributi composti

3.6) Eliminazione degli attributi multivalore

3.7) Elenco degli identificatori principali

3.8) Normalizzazione

3.8.1) Associazioni

3.8.2) Entità

3.9) Traduzione verso il modello relazionale

4) CODIFICA SQL E TESTING

4.1) Definizione dello schema

4.2) Codifica delle operazioni e screenshot che ne verificano l'esecuzione

1) ANALISI DEI REQUISITI

1.1) RACCOLTA INFORMAZIONI

1.1.1) Intervista al titolare

In data 09/11/2021 ci siamo recati presso la palestra di arrampicata sportiva Ancona Rock Climb in Via Giuseppe Persiani per intervistare il titolare, in modo da ottenere un primo livello sommario di informazioni utili alla progettazione e strutturazione del database. Le informazioni ottenute dal titolare riguardano aspetti generali dell'organizzazione aziendale e di come è strutturata la società.

Per garantire l'anonimato, ci rivolgeremo al titolare con l'appellativo di Sig. X.

Riportiamo di seguito alcuni estratti fondamentali dell'intervista.

D&A:

"Buonasera, siamo due atleti iscritti da poco nella vostra palestra e siamo interessati a comprendere meglio gli aspetti riguardanti l'organizzazione della vostra società. Essendo entrambi iscritti alla facoltà di Ingegneria Informatica e dell'Automazione, ci interessava progettare un database che riuscisse a venire incontro alle vostre esigenze."

Sig. X:

"Buonasera ragazzi. Ad essere sincero possediamo già un database che purtroppo non copre tutte le esigenze della nostra società. Per questo motivo sarei ben contento di aiutarvi nel vostro progetto. Chiedetemi pure ciò che preferite e che vi interessa."

D&A:

"Molto bene, prima di tutto ci interessava sapere come è strutturato l'organico della vostra società."

Sig. X:

"Essendo una piccola società sportiva, il nostro organico è formato fondamentalmente dai nostri soci e dai nostri istruttori, due dei quali si occupano della segreteria. I soci si occupano di aspetti burocratici e fiscali riguardanti la società. La gestione delle strutture e il loro coordinamento è affidato agli istruttori. Ognuno di loro inoltre è abilitato a tenere solo determinati corsi all'interno della palestra. Questo perché ovviamente ciascuno di loro si è specializzato in determinati aspetti dell'arrampicata sportiva, come ad esempio il bouldering o la scalata su roccia per citarne qualcuno. Ovviamente è possibile accedere a questi corsi solo mediante l'opportuno abbonamento e possono essere seguiti solo in determinate fasce orarie all'interno della settimana. La segreteria invece si occupa della registrazione e gestione dei dati relativi di ogni dipendente e di ogni atleta iscritto. Inoltre, avendo a disposizione alcuni sponsor, abbiamo a disposizione dell'attrezzatura sportiva che vendiamo ai nostri clienti."

D&A:

“Come organizzate l’attrezzatura sportiva di cui disponete?”

Sig. X:

“Abbiamo una stanza in palestra che è adibita a magazzino. L’inventario viene redatto una volta al mese e spesso lo aggiorniamo a seconda delle proposte dei nostri sponsor.”

D&A:

“Come fanno i clienti ad usufruire dei vostri servizi?”

Sig. X:

“Abbiamo un sito web sul quale gli atleti possono iscriversi e prenotare la sessione di allenamento in cui intendono andare. Purtroppo, causa Covid-19, è possibile prenotare solo determinate sessioni ogni settimana. Per quanto riguarda invece i servizi relativi alla gestione dei dati degli atleti vi invito a parlare direttamente con la segreteria. Spero di esservi stato di aiuto per il vostro progetto.”

D&A:

“Assolutamente sì e la ringraziamo infinitamente per il tempo che ci ha concesso. Arrivederci.”

Sig. X:

“Arrivederci ragazzi e alla prossima.”

1.1.2) Intervista alla segreteria

In data 10/11/2021 ci siamo recati presso la palestra di arrampicata sportiva Ancona Rock Climb in Via Giuseppe Persiani per intervistare i ragazzi che si occupano degli aspetti riguardanti la gestione dei dati dei vari dipendenti e atleti. Le informazioni ottenute sono leggermente più tecniche rispetto a quelle fornite dal titolare e ci hanno permesso di avere un’idea più chiara delle problematiche riscontrabili nella gestione e organizzazione dei dati, a prescindere dalla loro natura.

Per garantire l’anonimato, ci rivolgeremo al titolare con l’appellativo di Sig. Y.

Riportiamo di seguito alcuni estratti fondamentali dell’intervista.

D&A:

“Salve, il Sig. X ci ha detto che potevamo chiedere a voi gli aspetti più tecnici e concreti di come gestite i dati di dipendenti e atleti.”

Sig. Y:

“Ciao ragazzi, il Sig. X mi aveva avvertito della vostra idea di progetto. Sono a vostra disposizione, chiedetemi pure tutto ciò che ritenete essere importante.”

D&A:

“Bene, prima di tutto ci interessava sapere quali erano le uscite e le entrate principali della società.”

Sig. Y:

“Allora, le uscite principali riguardano fondamentalmente la manutenzione della palestra e lo stipendio dei vari dipendenti. Per le entrate invece ci affidiamo a due fattori fondamentali. Il primo sono ovviamente i pagamenti dei nostri atleti sotto varie modalità di abbonamento. Il secondo riguarda invece le entrate provenienti dalla vendita di attrezzatura sportiva che ci viene fornita dai nostri sponsor.”

D&A:

“Ci permetta la domanda un po’ delicata. Come gestite gli ingressi in palestra durante il periodo di emergenza Covid-19?”

Sig. Y:

“Per adesso la soluzione più efficiente che siamo riusciti a trovare è il diminuire il numero di posti per sessione di allenamento. Mediante il nostro sito internet gli atleti, ovviamente muniti di Green Pass, potranno prenotare la fascia oraria più comoda per allenarsi. Purtroppo, per evitare assembramenti, ogni atleta può prenotare solo determinate sessioni di allenamento settimanali.”

D&A:

“E per quanto riguarda la registrazione dei dati degli atleti?”

Sig. Y:

“Da quel punto di vista siamo molto meno tecnologici. Usiamo una raccolta cartacea nei quali registriamo temporaneamente i vari dati che poi andranno trascritti in un foglio di calcolo Excel.”

D&A:

“Secondo voi, l’implementazione di un database generale, che vi permetta di tener traccia di tutti i vostri dati e di garantirvi l’accesso e l’utilizzo a vostro piacimento, è preferibile rispetto alla vostra situazione attuale?”

Sig. Y:

“Assolutamente sì, soprattutto in questo periodo in cui stiamo registrando un forte incremento di iscrizioni e di conseguenza stanno aumentando i dati di cui dobbiamo tener traccia. Inoltre, a causa dell'emergenza sanitaria, dobbiamo garantire a tutti gli atleti il pieno rispetto delle norme Covid-19 per potersi allenare in un ambiente sicuro e protetto. Inoltre spesso e volentieri siamo soggetti a controlli da parte delle autorità e risulta quindi fondamentale garantire trasparenza e reperibilità di accesso ai nostri dati.”




D&A:

“Perfetto, è stato davvero esaustivo e la ringraziamo per il tempo che ci ha concesso. Vi faremo sapere eventualmente se abbiamo bisogno di altri chiarimenti, ma per adesso va benissimo così. Grazie ancora.”

Sig. Y:

“Di niente ragazzi, grazie a voi e alla prossima.”

1.1.3) Esempio di modulistica

		Ricevuta
		CUS Ancona A.S.D.
		Via Grotte 19/A, 60126 Ancona (AN) 🏠
		+39 3391730074 📞
		www.anconarockclimb.it 🌐
NUNIN DAVIDE		
TESSERA 10 INGRESSI BOULDERING		€ 50.00
Totale pagato il 14/01/2022		€ 50.00
Ancona, 14/01/2022		 Firma

1.2) REQUISITI ESPRESSI IN LINGUAGGIO NATURALE

Dopo un'attenta analisi dei dati raccolti e grazie alle informazioni estrapolate dalle varie interviste, siamo riusciti a comprendere al meglio il nostro obiettivo finale.

Sarà dunque di nostro interesse realizzare un database che permetta di organizzare, gestire e manipolare i dati della società sportiva Ancona Rock Climb.

Questi dati riguarderanno fondamentalmente tre macrocategorie: dati relativi degli atleti/iscritti, dati relativi dei dipendenti e dati riguardanti la vendita di attrezzatura sportiva.

Per quanto riguarda gli **atleti** sarà di fondamentale importanza tener traccia dei loro dati anagrafici, della tipologia di **abbonamento** che l'iscritto deciderà di utilizzare, delle prenotazioni che l'atleta deciderà di effettuare per presentarsi a determinate sessioni di allenamento, della tipologia di **assicurazione sportiva** di cui gode l'atleta, degli **accessi rimanenti** sull'abbonamento scelto e uno username ed una password per poter accedere al sito della società ed effettuare le possibili prenotazioni.

Per i **dipendenti** invece sarà fondamentale gestire i dati anagrafici, lo stipendio, i recapiti telefonici, le loro **certificazioni** con relativi **corsi** abilitati, le ore settimanali e i vari turni di lavoro in modo tale da garantire la presenza di almeno due persone alla segreteria.

Per la vendita di attrezzatura sportiva dovremmo tener traccia dei **dati identificativi** dei singoli prodotti, del loro prezzo di vendita e dell'elenco delle **fatture** relative alle singole **transazioni**.

Ovviamente tutte le entrate e uscite saranno utili al fine di calcolare il bilancio netto della società.

Ricordiamo che le principali fonti di entrata derivano dagli abbonamenti degli atleti e dalla vendita di attrezzatura sportiva, mentre le principali fonti di uscita riguardano le buste paga dei dipendenti.

1.3) STRUTTURAZIONE DEI REQUISITI

FRASI DI CARATTERE GENERALE

Dopo un'attenta analisi dei dati raccolti e grazie alle informazioni estrapolate dalle varie interviste, siamo riusciti a comprendere al meglio il nostro obiettivo finale.

Sarà dunque di nostro interesse realizzare un database che permetta di organizzare, gestire e manipolare i dati della società sportiva Ancona Rock Climb.

Questi dati riguarderanno fondamentalmente tre macrocategorie: dati relativi degli atleti/iscritti, dati relativi dei dipendenti e dati riguardanti la vendita di attrezzatura sportiva.

FRASI RELATIVE AGLI ATLETI

Per quanto riguarda gli atleti/iscritti sarà di fondamentale importanza tener traccia dei loro dati anagrafici, della tipologia di abbonamento che l'iscritto deciderà di utilizzare, delle prenotazioni che l'atleta deciderà di effettuare per presentarsi a determinate sessioni di allenamento, della tipologia di assicurazione sportiva di cui gode l'atleta, degli accessi rimanenti sull'abbonamento scelto e uno username ed una password per poter accedere al sito della società ed effettuare le possibili prenotazioni.

FRASI RELATIVE AI DIPENDENTI

Per i dipendenti invece sarà fondamentale gestire i dati anagrafici, lo stipendio, i recapiti telefonici, le loro certificazioni con relativi corsi abilitati, le ore settimanali e i vari turni di lavoro in modo tale da garantire la presenza di almeno due persone alla segreteria.

FRASI RELATIVE ALLA VENDITA DI ATTREZZATURA SPORTIVA

Per la vendita di attrezzatura sportiva dovremmo tener traccia dei dati identificativi dei singoli prodotti, del loro prezzo di vendita e dell'elenco delle fatture.

FRASI RELATIVE AL BILANCIO

Ovviamente tutte le entrate e uscite saranno utili al fine di calcolare il bilancio netto della società. Ricordiamo che le principali fonti di entrata derivano dagli abbonamenti degli atleti e dalla vendita di attrezzatura sportiva, mentre le principali fonti di uscita riguardano le buste paga dei dipendenti.

1.4) GLOSSARIO DEI TERMINI

È stato ritenuto opportuno andare a specificare in una apposita tabella il significato di determinati termini presenti all'interno della specifica dei requisiti espressi in linguaggio naturale.

Termine	Descrizione	Sinonimi	Collegamenti
Atleti	Coloro che, per mezzo di un abbonamento e di una assicurazione sportiva, possono usufruire dei corsi messi a disposizione dalla società.	Iscritti	Abbonamento, Assicurazione sportiva, Corsi
Abbonamento	È il mezzo grazie al quale gli atleti possono accedere a determinati corsi. Esistono diverse tipologie di abbonamento, ciascuna per ogni corso. Ognuno di essi presenta un numero di ingressi massimi utilizzabili dagli atleti.	Nessuno	Atleti, Corsi, Accessi rimanenti
Assicurazione sportiva	Assicurazione che garantisce la salute fisica dell'atleta, il quale risulta idoneo alla pratica di attività sportive.	Nessuno	Atleti
Accessi rimanenti	Rappresentano il numero di sessioni di allenamento presenti su ogni tipologia di abbonamento. Non è possibile superare il numero di accessi consentiti fino a	Entrate rimanenti	Atleti, Abbonamento

	quando non si rinnova l'abbonamento.		
Dipendenti	Coloro che offrono dei servizi alla società e vengono pagati dalla stessa. Ognuno presenta una certificazione che lo abilita a tenere determinati corsi.	Istruttori	Certificazioni, Corsi
Certificazioni	Attestati che abilitano gli istruttori a tenere determinati corsi.	Attestati sportivi	Dipendenti, Corsi
Corsi	Rappresentano le varie tipologie di allenamento che un dipendente può attuare e che un atleta può seguire a seconda dell'abbonamento utilizzato.	Sessioni di allenamento	Dipendente, Atleti
Dati identificativi	Rappresentano le informazioni utili a differenziare i vari prodotti che la società vende o noleggia agli atleti.	Nessuno	Atleti
Fatture	Documento con le indicazioni della merce o della prestazione fornita, dell'ammontare dell'importo e delle relative condizioni di pagamento. Indispensabile per tenere traccia delle singole transazioni.	Nessuno	Transazioni
Transazioni	Rappresentano le varie operazioni commerciali presenti all'interno della società per cui è richiesta una fattura.	Pagamenti	Fatture

1.5) SPECIFICA DELLE OPERAZIONI

1. Inserimento nuovo atleta/iscritto (in media 5 volte al mese)
2. Inserimento nuovi prodotti nel catalogo (in media 2 volte al mese)
3. Inserimento transazione abbonamento (in media 3 volte a settimana)
4. Inserimento transazione prodotto (in media 4 volte a settimana)
5. Inserimento transazione assicurazione (in media 5 volte al mese)
6. Inserimento transazione busta paga (in media 15 volte al mese)
7. Inserimento nuova prenotazione (in media 60 volte a settimana)
8. Inserimento accesso dell'atleta in palestra (in media 30 volte al giorno)
9. Inserimento nuovo dipendente (in media 2 volte l'anno)
10. Inserimento nuova sessione di allenamento (in media 10 volte a settimana)
11. Inserimento nuovo corso (in media 3 volte l'anno)

12. Stipulazione del tipo di abbonamento (in media 3 volte a settimana)

13. Modifica dati atleta/iscritto (in media 3 volte al mese)
14. Modifica dati dei prodotti (in media 2 volte a settimana)
15. Modifica dati dipendente (in media 3 volte al mese)
16. Modifica sessione di allenamento (in media 6 volte a settimana)

17. Cancellazione atleta/iscritto (in media una volta al mese)
18. Cancellazione dipendente (in media 3 volte l'anno)
19. Cancellazione prodotto/prodotti (in media 2 volte al mese)
20. Cancellazione sessione di allenamento (in media 2 volte al mese)
21. Cancellazione corso (in media 2 volte l'anno)

22. Consultazione dati dell'atleta/iscritto (in media 2 volte al giorno)
23. Consultazione dati dipendente (in media una volta a settimana)
24. Consultazione dati istruttore (in media 2 volte a settimana)
25. Consultazione dati Segretario (in media 3 volte a settimana)
26. Consultazione dati dei prodotti (in media 5 volte a settimana)
27. Consultazione dati abbonamento (in media 70 volte al giorno)

28. Consultazione dati assicurazione (in media 3 volte al mese)
29. Consultazione dati busta paga (in media 40 volte al mese)
30. Consultazione dati prenotazione (in media 10 volte al giorno)
31. Consultazione dati accesso alla palestra (in media 10 volte al giorno)
32. Consultazione dati bilancio (in media 3 volte l'anno)
33. Consultazione dati Transazione (in media 2 volte a settimana)
34. Consultazione dati Transazione abbonamento (in media 5 volte a settimana)
35. Consultazione dati Transazione assicurazione (in media 5 volte al mese)
36. Consultazione dati Transazione acquisto (in media 4 volte a settimana)
37. Visualizzazione media delle prenotazioni per settimana di un iscritto (in media 100 volte l'anno)
38. Visualizzazione guadagno medio settimanale grazie ai prodotti venduti (in media una volta a settimana)
39. Visualizzazione guadagno medio settimanale grazie agli abbonamenti (in media una volta a settimana)
40. Consultazione dati corso (in media 2 volte al mese)
41. Calcolo posti disponibili per una sessione di allenamento (in media 10 volte al giorno)
42. Calcolo accessi rimanenti sull'abbonamento (in media 2 volte a settimana)
43. Calcolo della media degli accessi per ogni settimana in un anno (in media 3 volte l'anno)
44. Visualizzazione sessioni prenotabili (in media 20 volte al giorno)
45. Visualizzazione prodotti in vendita (in media 2 volte a settimana)

2) PROGETTAZIONE CONCETTUALE

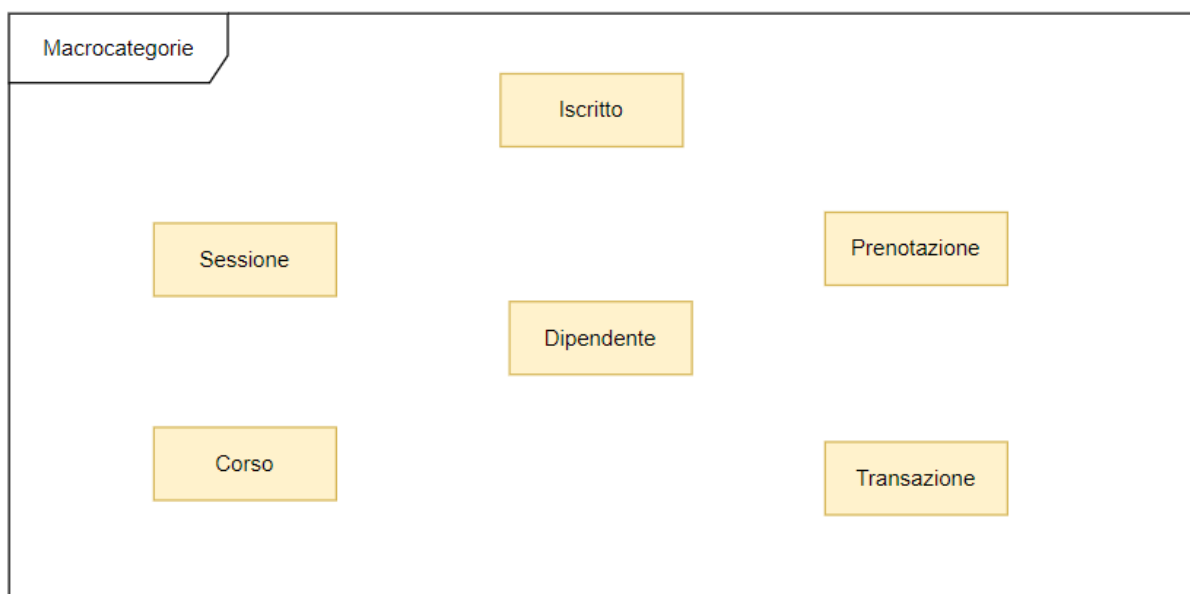
2.1) PROCEDIMENTO

Grazie alle numerose informazioni raccolte durante le interviste siamo riusciti ad avere un'idea molto più precisa dei problemi che dovremmo risolvere e gestire.

Abbiamo deciso che l'approccio migliore per gestire al meglio la mole di informazioni ottenuta dalle interviste è utilizzare un approccio misto. Abbiamo utilizzato sia l'approccio top-down che l'approccio bottom-up. Grazie all'approccio top-down siamo riusciti ad analizzare una realtà complessa nei suoi singoli aspetti costituenti mentre con l'approccio bottom-up abbiamo raccolto questi singoli aspetti in una visione più ampia del problema da risolvere.

2.2) IDENTIFICAZIONE DELLE ENTITÀ E RELAZIONI FONDAMENTALI

Grazie all'analisi dei requisiti siamo riusciti a identificare quattro macrocategorie principali: iscritto, prenotazione, sessione, corso, dipendente, transazione.



Con ISCRITTO indichiamo l'atleta che parteciperà, grazie a determinate transazioni, alle possibili sessioni di allenamento presenti in palestra. Potrà inoltre usufruire di altri servizi come l'acquisto di attrezzatura sportiva.

Con PRENOTAZIONE indichiamo il metodo con cui l'iscritto si riserva l'accesso ad una determinata sessione di allenamento.

Con SESSIONE indichiamo la seduta di allenamento alla quale l'iscritto potrà accedere e in cui potrà seguire differenti corsi conformi alla tipologia di abbonamento scelto. Per potervi accedere l'iscritto dovrà prima aver effettuato la prenotazione in determinate fasce orarie.

Con CORSO intendiamo la tipologia di allenamento prevista all'interno di una sessione. È importante sottolineare che determinati corsi potranno essere tenuti solo da determinati dipendenti.

Con DIPENDENTE indichiamo sia gli istruttori che si occupano di tenere determinati corsi all'interno di ogni singola sessione, sia i membri che costituiscono la segreteria, i cui compiti fondamentali saranno tenere traccia di tutte le transazioni che vengono effettuate e registrare correttamente i dati riguardanti iscritti e dipendenti.

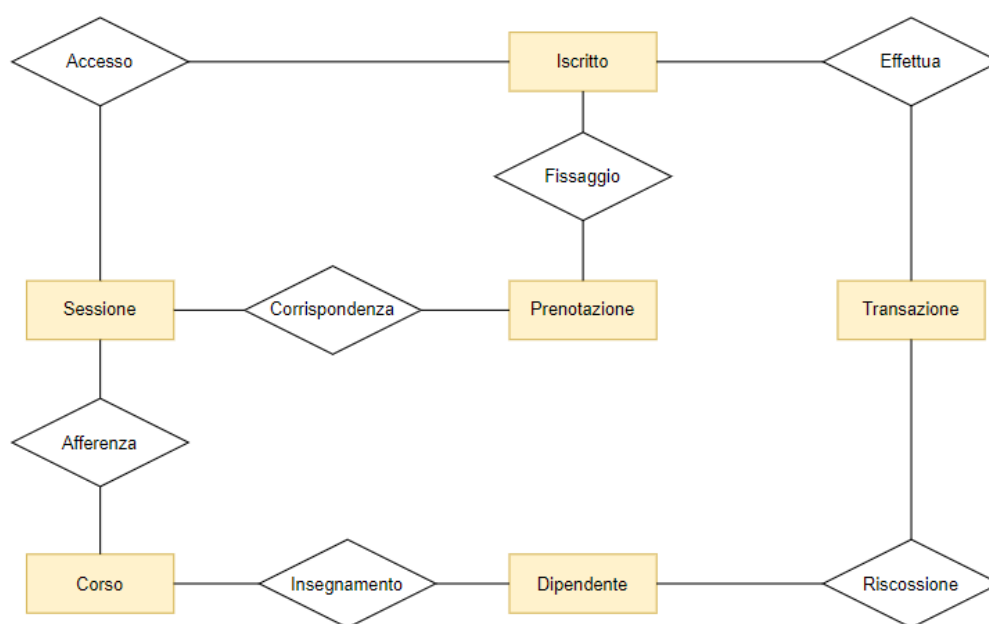
Con TRANSAZIONE indichiamo, per quanto riguarda l'iscritto, l'avvenuto pagamento di un determinato abbonamento, dell'assicurazione medica o dell'acquisto di attrezzatura sportiva. Per quanto riguarda il dipendente indicherà invece l'avvenuta consegna di una busta paga.

2.3) SCHELETRO DELLO SCHEMA

Presentiamo un primo schema concettuale che verrà modificato e migliorato in seguito.

Abbiamo cercato di riassumere le principali entità e relazioni presenti nel nostro progetto.

È importante sottolineare fin da subito che, ovviamente, l'entità Iscritto e l'entità Dipendente sono legate mediante distinte relazioni alla medesima entità Transazione. In seguito ci occuperemo di distinguere ulteriormente le differenti tipologie di transazione che interessano l'iscritto e il dipendente.

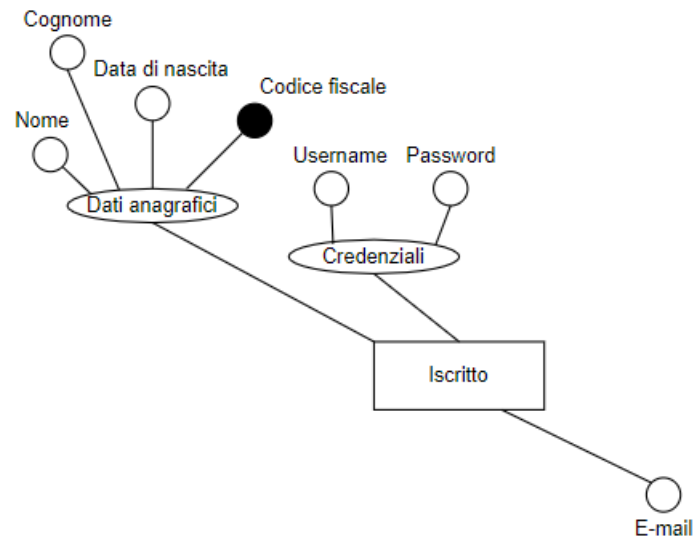


2.3.1) Sviluppo componenti dello scheletro

Una volta definito lo scheletro e le sue componenti ci interessiamo a specificare ognuna di queste mediante l'utilizzo della metodologia TOP-DOWN.

ISCRITTO

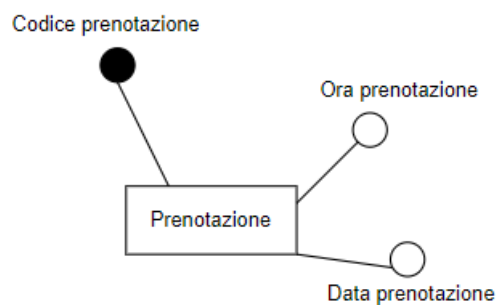
L'iscritto costituirà l'elemento cardine all'interno del nostro E-R. Sarà ovviamente caratterizzato dai suoi dati anagrafici (nome, cognome, data di nascita e codice fiscale), dalla sua E-mail per comunicazioni di servizio e da uno username e una password necessari per accedere al sito della società ed effettuare le necessarie prenotazioni per le sessioni di allenamento di interesse.



PRENOTAZIONE

La prenotazione permetterà all'iscritto di riservarsi un posto ad una determinata sessione di allenamento.

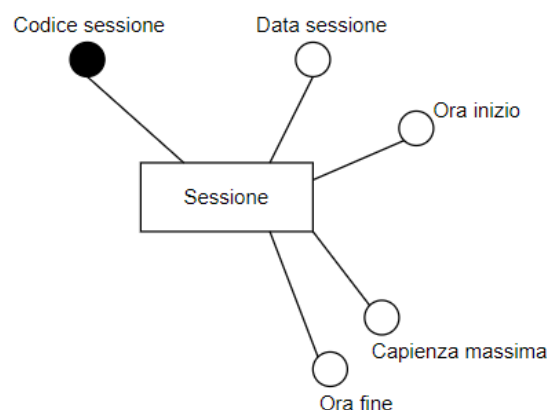
Si osservi che per il momento la chiave identificativa di Prenotazione sarà Codice prenotazione. Tuttavia, successivamente, la chiave primaria diventerà un identificatore esterno costituito dalla chiave primaria dell'entità Iscritto (Codice fiscale) e dalla chiave primaria dell'entità Sessione (Codice sessione).



SESSIONE

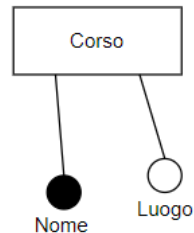
La sessione potrà essere accessibile da parte dell'iscritto previa prenotazione. La sessione presenterà al suo interno diversi corsi specifici che l'iscritto, a seconda dell'abbonamento scelto, sarà libero di seguire.

Ogni sessione dovrà presentare la data, l'orario di inizio e di fine allenamento, la capienza massima consentita e ovviamente un codice identificativo.



CORSO

I corsi saranno presenti in ogni sessione, ognuno dei quali viene tenuto da determinati dipendenti (istruttori) specializzati in alcuni ambiti. Ogni corso sarà caratterizzato da un nome identificativo e il luogo in cui si intende svolgere il corso (indoor, outdoor).

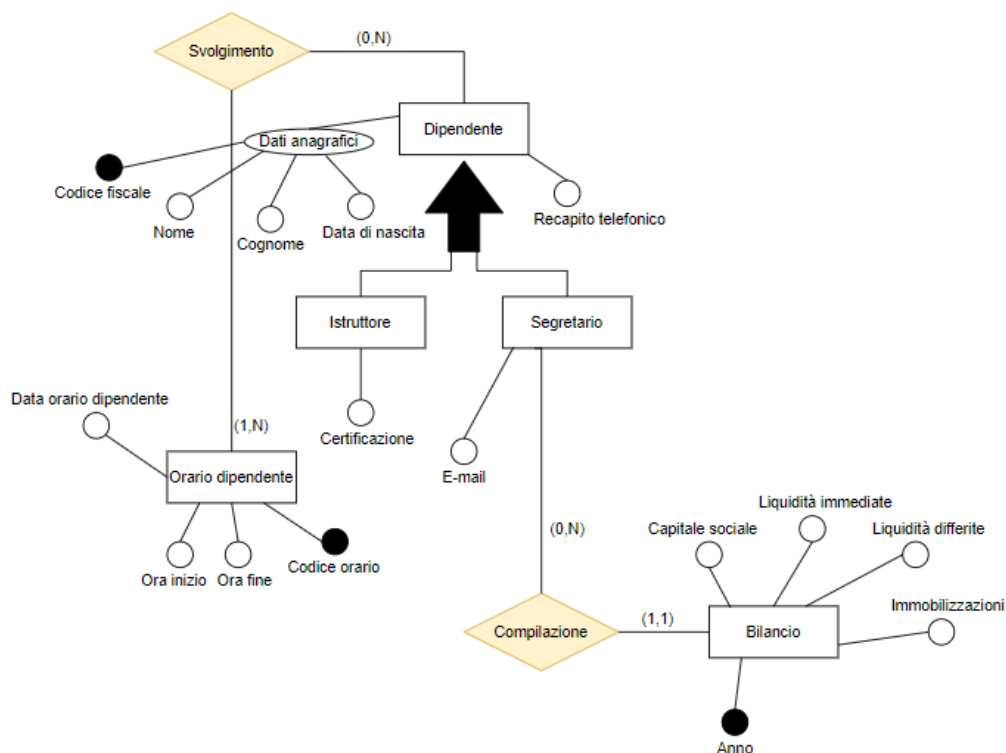


DIPENDENTE

Il dipendente è l'insieme degli istruttori e della segreteria. Queste due categorie avranno compiti ben distinti e per tal motivo saranno caratterizzati in maniera differente. Osserviamo prima di tutto i tratti in comune. Ogni dipendente presenterà dei dati anagrafici (nome, cognome, data di nascita, codice fiscale) e un recapito telefonico per poter dare la propria reperibilità alla società. Un altro aspetto fortemente legato al dipendente è ovviamente il suo turno lavorativo (orario dipendente) che sarà caratterizzato da una data, da un orario di inizio e di fine turno lavorativo e ovviamente da un codice identificativo.

Entrando più nel particolare ogni singolo istruttore avrà determinate certificazioni che gli permetteranno di tenere determinati corsi all'interno di ogni sessione.

La segreteria invece sarà fortemente legata al bilancio (caratterizzato dal capitale sociale, liquidità immediate, liquidità differite, immobilizzazioni e anno di esercizio) della società e di conseguenza dovrà essere in grado di registrare tutti i dati e tutte le transazioni necessarie alla sua compilazione. Inoltre presenterà una E-mail grazie alla quale una terza persona potrà ricevere informazioni riguardanti la società senza necessariamente presentarsi fisicamente all'interno della sede.



TRANSAZIONE

La transazione identifica ogni possibile entrata o uscita della società. Ognuna di loro presenterà l'importo, la data di avvenuta, l'ora di avvenuta e un ID identificativo. La società Ancona Rock Climb presenta quattro differenti tipologie di transazione: transazione relativa all'abbonamento, transazione relativa all'assicurazione sportiva, transazione relativa alla vendita di attrezzatura sportiva e transazioni relative alle buste paga dei dipendenti.

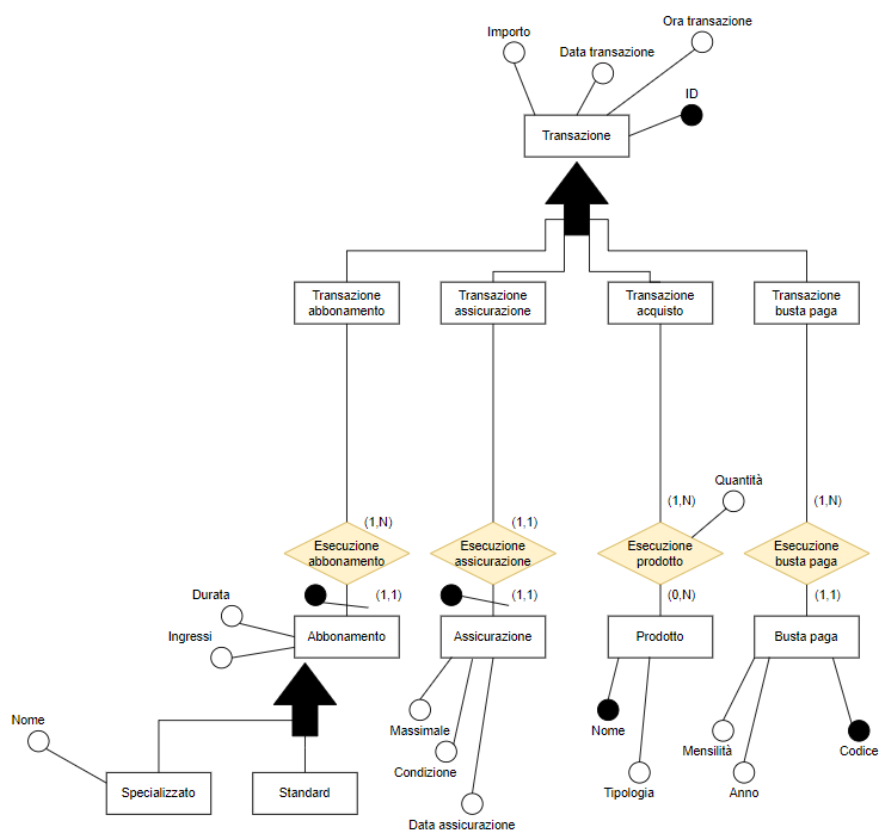
La transazione relativa all'abbonamento permette all'iscritto di acquisire un abbonamento caratterizzato da una durata e dagli ingressi che l'abbonamento mette a disposizione. Le tipologie di abbonamento sono fondamentalmente due: l'abbonamento standard e l'abbonamento specializzato. Il primo permette l'ingresso in palestra per bouldering, il secondo invece sarà caratterizzato da un nome che associa l'abbonamento alla tipologia di corso che permette di seguire, come ad esempio arrampicata su roccia, in falesia, corso di yoga, corso di pilates, corso di potenziamento muscolare ecc.

La transazione relativa all'assicurazione sportiva permette all'iscritto di assicurarsi con il comitato sportivo universitario (C.U.S. di Ancona). Tale assicurazione, oltre ad essere indispensabile per l'iscritto e l'incolumità legale della società, sarà caratterizzata da un massimale, da una condizione e da una data di scadenza.

La transazione relativa all'acquisto di attrezzatura sportiva permette all'iscritto di acquistare in sede determinati prodotti utili ai fini dell'allenamento come scarpette da arrampicata, magnesite liquida, moschettoni, corde, ecc.

Ogni prodotto sarà caratterizzato da un nome identificativo e dalla tipologia di appartenenza.

La transazione relativa alle buste paga permetterà ad ogni singolo dipendente di riscuotere una certa somma la quale sarà caratterizzata dalla mensilità, dall'anno e da un codice identificativo.



2.4) MODELLO E-R COMPLESSIVO

Di seguito abbiamo riportato il modello E-R generale ottenuto applicando la metodologia BOTTOM-UP alle singole componenti analizzate durante la metodologia TOP-DOWN.

Analizziamo di seguito le relazioni che hanno permesso di legare tra di loro le quattro macrocategorie formate da: Iscritto, Sessione, Corso, Dipendente, Transazione.

Iscritto

- Relazioni tra macrocategoria Iscritto e macrocategoria Sessione:
 - 1) Accesso, relazione tra entità Iscritto ed entità Sessione
 - 2) Corrispondenza, relazione tra entità Prenotazione ed entità Sessione
- Relazioni tra macrocategoria Iscritto e macrocategoria Transazione:
 - 1) Effettua abbonamento, relazione tra entità Iscritto ed entità Transazione abbonamento
 - 2) Effettua assicurazione, relazione tra entità Iscritto ed entità Transazione assicurazione
 - 3) Effettua acquisto, relazione tra entità Iscritto ed entità Transazione acquisto

Sessione

- Relazioni tra macrocategoria Sessione e macrocategoria Iscritto:
 - 1) Accesso, relazione tra entità Sessione ed entità Iscritto
 - 2) Corrispondenza, relazione tra entità Sessione ed entità Prenotazione

Corso

- Relazione tra macrocategoria Corso e macrocategoria Dipendente:
 - 1) Insegnamento, relazione tra entità Corso ed entità Istruttore

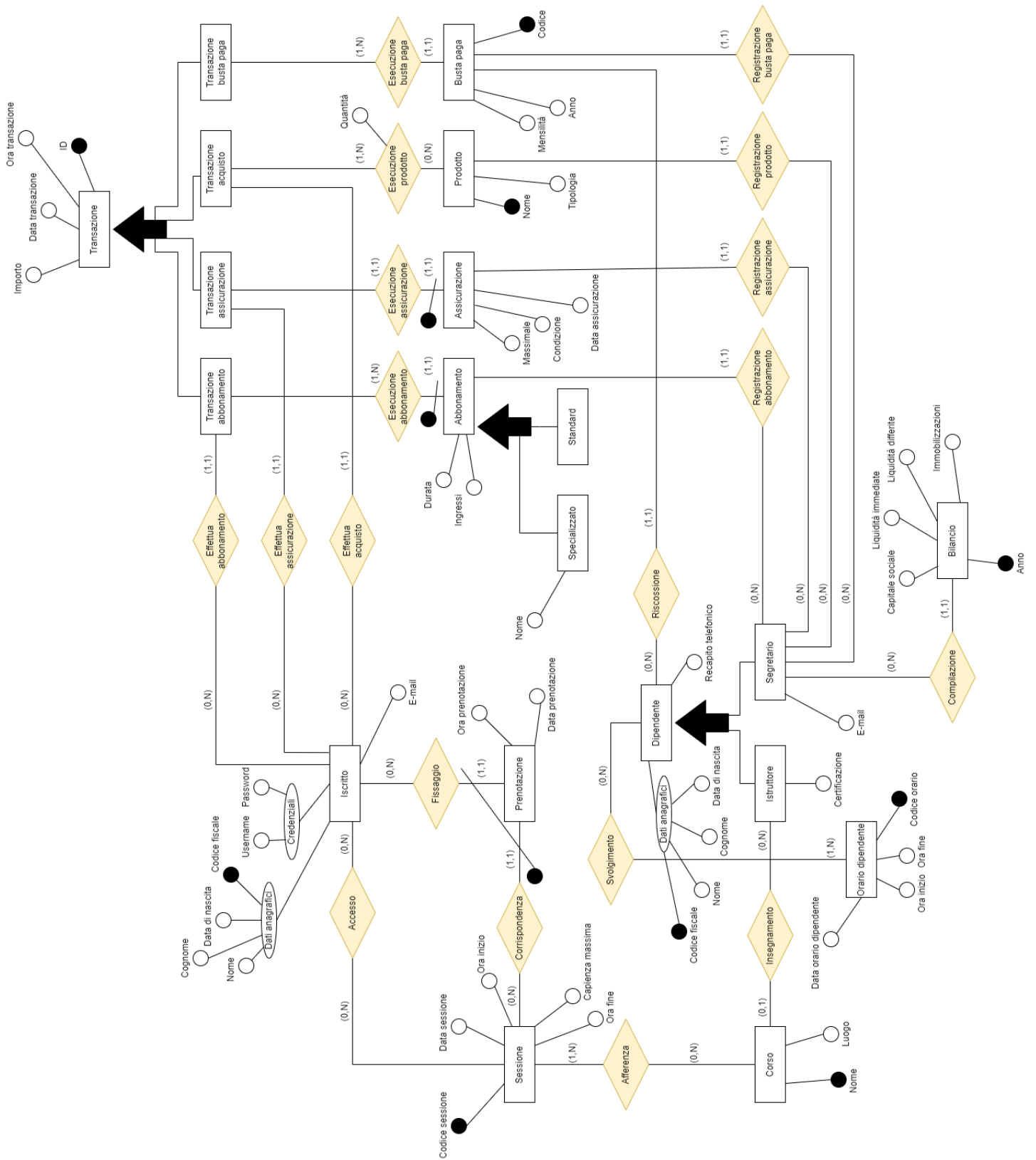
Dipendente

- Relazione tra macrocategoria Dipendente e macrocategoria Corso:
 - 1) Insegnamento, relazione tra entità Istruttore ed entità Corso
- Relazioni tra macrocategoria Dipendente e macrocategoria Transazione:
 - 1) Riscossione, relazione tra entità Dipendente ed entità Busta paga
 - 2) Registrazione abbonamento, relazione tra entità Segreteria e entità Abbonamento
 - 3) Registrazione assicurazione, relazione tra entità Segreteria e entità Assicurazione
 - 4) Registrazione prodotto, relazione tra entità Segreteria e entità Prodotto
 - 5) Registrazione busta paga, relazione tra entità Segreteria e entità Busta paga

Transazione

- Relazioni tra macrocategoria Transazione e macrocategoria Dipendente:
 - 1) Riscossione, relazione tra entità Busta paga ed entità Dipendente
 - 2) Registrazione abbonamento, relazione tra entità Abbonamento ed entità Segreteria
 - 3) Registrazione assicurazione, relazione tra entità Assicurazione ed entità Segreteria
 - 4) Registrazione prodotto, relazione tra entità Prodotto ed entità Segreteria
 - 5) Registrazione busta paga, relazione tra entità Busta paga ed entità Segreteria
- Relazioni tra macrocategoria Transazione e macrocategoria Iscritto
 - 1) Effettua abbonamento, relazione tra entità Transazione abbonamento ed entità Iscritto
 - 2) Effettua assicurazione, relazione tra entità Transazione assicurazione ed entità Iscritto
 - 3) Effettua acquisto, relazione tra entità Transazione acquisto ed entità Iscritto

Riportiamo di seguito il diagramma E-R complessivo.



2.5) ANALISI DI QUALITÀ DELLO SCHEMA E-R

CORRETTEZZA: lo schema è stato costruito seguendo le regole fondamentali sia a livello semantico che a livello sintattico.

COMPLETEZZA: una volta costruito l'intero schema E-R ci siamo preoccupati di confrontare tale schema con la raccolta dei requisiti avvenuta mediante le interviste e abbiamo constatato che gli aspetti fondamentali sono stati trattati con efficacia.

LEGGIBILITÀ: a causa della complessità dello schema non siamo purtroppo riusciti ad evitare completamente la generazione di intersezioni. Tuttavia, ci siamo preoccupati di rappresentare ogni entità ed ogni relazione nel miglior modo possibile in modo tale da rendere la lettura dello schema di rapida comprensione.

MINIMALITÀ: lo schema creato non presenta né ridondanze né cicli, contribuendo dunque alla comprensione generale dello schema E-R.

2.6) DIZIONARIO DEI DATI

2.6.1) Dizionario dei dati (entità)

Nome Entità	Descrizione	Attributi		Identificatore
Iscritto	Utente della palestra regolarmente registrato e che può usufruire dei suoi servizi	Dati Anagrafici(nome(stringa), cognome(stringa), data di nascita(data)e Codice Fiscale(stringa)), Credenziali(username (stringa) e password (stringa)), e-mail (stringa)	“password” non si riferisce alla password in chiaro ma criptata secondo una qualche logica lato client	Codice Fiscale
Prenotazione	Appuntamento prenotato da un iscritto che gli riserva un posto per una sessione di allenamento in palestra	Data prenotazione(date), Ora prenotazione (time)	Data effettuazione si riferisce al giorno in cui l' utente ha effettuato la prenotazione	Codice Sessione(Sessione),Codice Fiscale (Iscritto)
Sessione	Periodo di tempo in cui la palestra è disponibile all' accesso ed è possibile frequentare corsi	Data sessione(data),ora inizio(time),ora fine(time),Capienza massima(numerico),Codice sessione(numerico)	Ora inizio e fine seguono le regole per il formato previste per gli orari	Codice Sessione
Corso	Corso di una disciplina di arrampicata sportiva o altro genere di allenamento	Nome(stringa), luogo(stringa)		Nome
Dipendente	Persona che lavora per AnconaRockClimb	Dati Anagrafici(nome(stringa), cognome(stringa), data di nascita(date), Codice Fiscale(stringa)), Recapito telefonico(stringa)		Codice Fiscale
Istruttore	Dipendente che svolge la mansione di insegnamento di corsi durante le sessioni di allenamento	“ Certificazione(stringa)		Codice Fiscale
Segretario	Dipendente che svolge mansioni di segreteria, raccolta dati, e gestione informatica	“ e-mail (stringa)		Codice Fiscale

Bilancio	Rappresentazione del bilancio d'impresa	Capitale sociale(numerico), Liquidità immediate(numerico), Liquidità differite (numerico), Immobilizzazioni(numerico), Anno(Numerico)		Anno
Orario dipendente	Rappresentazione dei turni di lavoro settimanali a cui è soggetto un dipendente	Ora inizio(time), ora fine(time), Data orario dipendente(date), Codice orario (numerico)	Ora inizio e fine seguono le regole per il formato previste per gli orari	Codice orario
Transazione	Spostamento monetario che coinvolge l'azienda	ID(numerico), importo(numerico), data transazione(date), ora transazione(time)	Data e ora si riferiscono al momento in cui è stata registrata la transazione	ID
Transazione Abbonamento	Transazione associata alla compravendita di un abbonamento	“		ID
Transazione acquisto	Transazione associata alla compravendita di un prodotto fisico	“		ID
Transazione busta paga	Transazione associata al pagamento di una busta paga di un dipendente	“		ID
Transazione Assicurazione	Transazione associata al pagamento un tantum di un'assicurazione da parte di un iscritto	“		ID
Abbonamento	Contratto che dietro pagamento di una somma dà all'iscritto possibilità di accesso in palestra	Ingressi (numerico), Durata (stringa)		ID(transazione abbonamento)
Abbonamento standard	Abbonamento che permette di accedere al corso	“		ID(transazione abbonamento)

	di arrampicata standard			
Abbonamento specializzato	Abbonamento che permette di accedere a un corso specifico	“ Nome(stringa)		ID(transazione abbonamento)
Assicurazione	Contratto che tutela l' iscritto in caso di infortunio durante una sessione di allenamento presso la palestra	Massimale(numerico), Condizione(stringa), Data assicurazione(date)	Condizioni è la stringa che contiene il testo del contratto in cui si fa riferimento alle condizioni in cui si applica l' assicurazione	ID(transazione assicurazione)
Prodotto	Strumento per l' arrampicata venduto dalla società	Nome(stringa), tipologia(stringa)	Tipologia è la categoria di prodotti per l' arrampicata di cui il prodotto fa parte(scarpe, flacone di magnesite...)	Nome
Busta paga	Documento fornito ai dipendenti relativo all' importo della retribuzione recepita per un mese di lavoro	Mensilità(numerico), Anno(numerico), Codice (numerico)	Mensilità si riferisce al mese a cui la busta paga è riferita (possono esserci anche tredicesime)	Codice

2.6.2) Dizionario dei dati (relazioni)

Nome Relazione	Descrizione	Entità coinvolte	Attributi
Accesso	Associa ogni iscritto alle sessioni di allenamento da lui frequentate	Iscritto(0,N) Sessione(0,N)	***
Afferenza	Associa una sessione di allenamento ai corsi che in essa si svolgono	Sessione(1,N) Corso(0,N)	***
Insegnamento	Associa ogni corso al relativo istruttore	Corso(0,1) Istruttore(0,N)	***
Corrispondenza	Associa ogni prenotazione alla relativa sessione prenotata	Prenotazione(1,1) Sessione(0,N)	***

Fissaggio	Associa ogni prenotazione all' iscritto che la ha fissata	Iscritto(0,N) Prenotazione(1,1)	***
Svolgimento	Associa i turni di lavoro(orario dipendente) ai dipendenti che devono svolgerli	Dipendente(0,N) Orario Dipendente(1,N)	***
Effettua Abbonamento	Associa gli iscritti alle transazioni di acquisto degli abbonamenti	Iscritto(0,N), Transazione abbonamento(1,1)	***
Effettua Assicurazione	Associa gli iscritti alle transazioni di acquisto delle assicurazioni	Iscritto(0,N), Transazione assicurazione(1,1)	***
Effettua Acquisto	Associa gli iscritti alle transazioni di acquisto dei prodotti	Iscritto(0,N), Transazione acquisto(1,1)	***
Esecuzione abbonamento	Associa ogni transazione abbonamento agli effettivi abbonamenti acquistati	Transazione abbonamento(1,N), Abbonamento(1,1)	***
Esecuzione assicurazione	Associa ogni transazione assicurazione all' assicurazione sottoscritta	Transazione assicurazione(1,1), Assicurazione(1,1)	***
Esecuzione prodotto	Associa ogni transazione acquisto agli effettivi prodotti acquistati	Transazione acquisto(1,N), Prodotto(0,N)	Quantità(numerico): indica il numero di copie di quell' articolo acquistate nella transazione
Esecuzione busta paga	Associa ogni transazione busta paga alla busta paga effettiva rilasciata	Transazione Busta Paga(1,N), Busta Paga(1,1)	***
Riscossione	Associa ogni busta paga al dipendente a cui viene rilasciata	Dipendente(0,N), Busta paga(1,1)	***
Registrazione abbonamento	Associa ogni abbonamento con il dipendente di segreteria che lo ha registrato	Segreteria(0,N), Abbonamento(1,1)	***
Registrazione assicurazione	Associa ogni assicurazione con il dipendente di segreteria che la ha registrata	Segreteria(0,N), Assicurazione(1,1)	***
Registrazione Prodotto	Associa ogni prodotto con il dipendente di segreteria che lo ha registrato	Segreteria(0,N), Prodotto(1,1)	***
Registrazione Busta paga	Associa ogni busta paga con il dipendente di	Segreteria(0,N), Busta Paga(1,1)	***

	segreteria che la ha registrata		
--	---------------------------------	--	--

2.7) VINCOLI DI INTEGRITÀ DEI DATI

Regole di vincolo:

RV1: Tutti gli attributi contenenti la parola “ora” rappresentano orari

RV2: L’ attributo “recapito telefonico” dell’ entità istruttore deve rappresentare un numero di telefono valido

RV3: L’ attributo composito dati anagrafici deve essere composto 4 attributi: nome(stringa), cognome(stringa), data di nascita(data), e codice fiscale(stringa), di cui data di nascita deve rappresentare una data valida

RV4: gli attributi “importo” nelle specializzazioni di transazione deve corrispondere a una somma di denaro espressa in euro non negativa

RV5: L’ attributo mensilità in busta paga deve essere una stringa rappresentante un mese stipendiabile o una tredicesima

RV6: gli attributi e-mail nelle varie entità devono essere degli indirizzi e-mail validi

RV7: L’ attributo certificazione appartenente a istruttore deve essere una stringa rappresentante il nome del corso che l’ istruttore può insegnare e deve essere coerente con la relazione insegnamento

RV8: L’ attributo “ingressi” dell’ entità abbonamento deve essere un intero maggiore di 0.

RV9: L’ attributo “luogo” di corso deve essere una stringa che può assumere solo 2 valori: “indoor” o “outdoor”

RV10: L’ attributo “anno” dell’ entità busta paga deve essere un intero che rappresenta un anno in cui il dipendente ha lavorato per la società

RV11: L’ attributo nome dell’ entità abbonamento (specializzato) deve essere una stringa che può assumere i valori contenuti nell’ attributo “nome” delle occorrenze dell’ entità “corso”

RV12: L’ attributo “quantità” della relazione esecuzione prodotto deve essere un intero maggiore di 0.

RV13: L’ attributo “durata” dell’ entità abbonamento deve essere una stringa che può assumere i valori: “mensile”, “settimanale” e “annuale”.

RV14: L’ attributo “massimale” dell’ entità assicurazione deve rappresentare un numero di euro maggiore di 0

RV15: Gli attributi “Capitale sociale”, “Liquidità differite” e “Liquidità immediate” devono rappresentare somme di denaro maggiori o uguali di € 0,00

RV16: L’ attributo “Anno” di bilancio si riferisce all’ anno di esercizio della società e deve quindi essere un numero

RV17: L’ attributo “tipologia” dell’ entità “prodotto” deve essere una stringa che può assumere solo il valori: “scarpette”, “magnesite”, “casco”, “corda”, “imbracatura”, “moschettone”, “carrucola”, “picchetto”, “Magliette”, “Pantaloni”, “Cappelli”.

RV18: Non si può registrare un accesso ad una sessione non ancora avvenuta

RV19: Non si può inserire una nuova assicurazione con una DataAssicurazione inferiore alla data attuale

RV20: Nell' entità OrarioDipendente gli attributi OrarioInizio e OrarioFine devono rappresentare orari validi e l' orario di fine deve essere maggiore di quello di inizio

RV21: Non si può inserire una prenotazione ad una sessione già iniziata o conclusa

RV22: Non si può inserire una prenotazione ad una sessione che non ha posti rimanenti

RV23: Non si può inserire una prenotazione ad una sessione di cui non si ha un abbonamento ad un corso ivi presente

RV24: Non si può inserire una prenotazione ad una sessione di cui si ha un abbonamento ad un corso ma è scaduto

RV25: Non si può inserire una prenotazione ad una sessione di cui si ha un abbonamento ad un corso che non ha più entrate disponibili

3) PROGETTAZIONE LOGICA

3.1) TAVOLA DEI VOLUMI E DELLE OPERAZIONI

3.1.1) Tavola dei volumi

CONCETTO	TIPO	VOLUME
Iscritto	E	1000
Prenotazione	E	20000
Sessione	E	1750
Corso	E	20
Dipendente	E	30
Istruttore	E	20
Segretario	E	10
Bilancio	E	5
Orario dipendente	E	24500
Busta paga	E	1300
Prodotto	E	150
Assicurazione	E	1250
Abbonamento	E	20000
Specializzato	E	30
Standard	E	1
Transazione	E	69425
Transazione busta paga	E	1300
Transazione acquisto	E	46875
Transazione assicurazione	E	1250
Transazione abbonamento	E	20000
Fissaggio	R	20000
Corrispondenza	R	20000
Accesso	R	26250
Afferenza	R	9625
Svolgimento	R	24500
Riscossione	R	1300
Insegnamento	R	25
Compilazione	R	10
Registrazione abbonamento	R	60
Registrazione assicurazione	R	15
Registrazione prodotto	R	150
Registrazione busta paga	R	1300
Esecuzione abbonamento	R	60
Esecuzione assicurazione	R	15
Esecuzione prodotto	R	150
Esecuzione busta paga	R	1300
Effettua abbonamento	R	20000
Effettua assicurazione	R	1250
Effettua acquisto	R	46875

NOTA: è importante specificare che all'interno dell'analisi dei volumi delle entità e delle relazioni, il volume è fortemente legato allo scorrere del tempo. Nel nostro caso stiamo ipotizzando un ciclo di vita della nostra base di dati pari a circa 5 anni. Considerando il tipo di società a cui questa base di dati è dedicata ci sembrava essere un periodo temporale conforme alle richieste della società stessa. Inoltre ogni volume è da considerarsi come una stima approssimativa, la quale cerca di rispecchiare il più fedelmente possibile la realtà.

3.1.2) Tavola delle operazioni

OPERAZIONE	FREQUENZA
1	5 volte al mese
2	2 volte al mese
3	3 volte a settimana
4	4 volte a settimana
5	5 volte al mese
6	15 volte al mese
7	60 volte a settimana
8	30 volte al giorno
9	2 volte l'anno
10	10 volte a settimana
11	3 volte l'anno
12	3 volte a settimana
13	3 volte al mese
14	2 volte a settimana
15	3 volte al mese
16	6 volte a settimana
17	1 volta al mese
18	3 volte l'anno
19	2 volte al mese
20	2 volte al mese
21	2 volte l'anno
22	2 volte al giorno
23	1 volta a settimana
24	2 volte a settimana
25	3 volte a settimana
26	5 volte a settimana
27	70 volte al giorno
28	3 volte al mese
29	40 volte al mese
30	10 volte al giorno
31	10 volte al giorno
32	3 volte l'anno
33	2 volte a settimana
34	5 volte a settimana
35	5 volte al mese
36	4 volte a settimana
37	100 volte l'anno
38	1 volta a settimana
39	1 volta a settimana
40	2 volte al mese
41	10 volte al giorno

42	2 volte a settimana
43	3 volte l'anno
44	20 volte al giorno
45	2 volte a settimana

3.2) ANALISI DELLE RIDONDANZE

3.2.1) Attributo "ingressi rimanenti" in "abbonamento"

Assenza di ridondanza

Note:

Il ricalcolo degli accessi rimanenti avviene in media 2 volte a settimana, ossia ogni volta che l'iscritto effettua un'accesso;

Operazione 35			
Concetto	Costrutto	Accessi	Tipo
Abbonamento	E	1	L
Esecuzione abbonamento	R	1	L
Transazione abbonamento	E	1	L
Effettua abbonamento	R	1	L
Iscritto	E	1	L
Accesso	R	S	L
Sessione	E	S	L
Afferenza	R	S*N	L
Corso	E	S*N	L

Con S intendiamo il numero medio di accessi per ogni iscritto. È possibile ricavarlo dalla tavola dei volumi come $\frac{\text{Accesso}}{\text{Iscritto}} = \frac{26250}{1000} = 26,25$.

Con N intendiamo il numero di corsi a cui afferisce una sessione. È possibile ricavarlo dalla tavola dei volumi come $\frac{\text{Afferenza}}{\text{Sessione}} = \frac{9625}{1750} = 5,5$.

Procediamo con il calcolo del costo totale dell'operazione 35 senza ridondanza.

$$\text{Costo totale} = \text{Abbonamento} \times (5 + S \times 2 + S \times N \times 2) = 6925000$$

Nota: l'operazione 35 è un'operazione che va svolta mediamente 2 volte a settimana per ogni iscritto, ossia facendo riferimento alla tavola dei volumi circa 500 volte ogni 5 anni (nel caso medio in cui esegua 2 accessi a settimana ogni settimana).

Inserimento della ridondanza

Operazione 35			
Concetto	Costrutto	Accessi	Tipo
Abbonamento	E	1	L
Abbonamento	E	1	S

Procediamo con il calcolo del costo totale dell'operazione 35 con l'aggiunta della ridondanza.

$$\text{Costo totale} = \text{Abbonamento} \times (1 + 2) = 60000$$

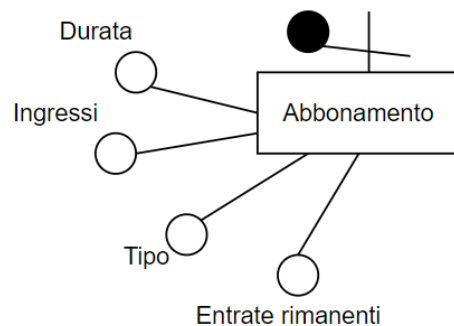
Riportiamo di seguito il costo totale relativo all'operazione 35, tenendo conto della frequenza con la quale essa viene effettuata in un periodo di 5 anni.

Operazione	Costo	Frequenza(settimanale)	Totale
35	6925000	2	$3,6 \times 10^9$
Costo operazioni senza ridondanza			$3,6 \times 10^9$

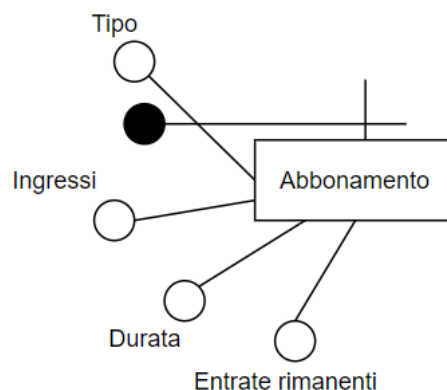
Operazione	Costo	Frequenza(settimanale)	Totale
35	60000	2	$3,1 \times 10^7$
Costo operazioni con ridondanza			$3,1 \times 10^7$

Si ottiene un fattore di riduzione delle operazioni di circa 115 volte, quindi conviene inserire la ridondanza.

Viene quindi effettuata la seguente modifica allo schema:



Durante l'analisi della ridondanza ci siamo tuttavia accorti che nelle fasi successive del progetto si sarebbero potute verificare delle complicità relative all'identificazione dell'entità **Abbonamento**. Di conseguenza ci è sembrato opportuno modificare la chiave d'identificazione di **Abbonamento** nel seguente modo:



Si tenga conto dunque che nelle successive rappresentazioni del nostro diagramma entità-relazione, l'entità Abbonamento sarà identificata come descritto precedentemente.

3.2.2) Attributo "posti rimanenti" in sessione

Assenza di ridondanza

Note:

il ricalcolo dei posti disponibili avviene ogni volta che un iscritto si deve prenotare quindi mediamente è un'operazione che viene svolta una decina di volte al giorno

Operazione 34			
Concetto	Costrutto	Accessi	Tipo
Sessione	E	1	L
Corrispondenza	R	N	L

Dove:

$$N = \frac{\text{Corrispondenza}}{\text{Sessione}} = \frac{20000}{1750} \approx 11$$

Procediamo con il calcolo del costo totale dell'operazione 34 senza ridondanza.

$$\text{Costo totale} = \text{Sessione} \times (N + 1) = 21000$$

Note:

la visualizzazione delle sessioni prenotabili avviene con la stessa frequenza del calcolo dei posti rimanenti, ma viene eseguita su tutte le sessioni registrate che hanno una data non passata, le sessioni di allenamento vengono inserite di settimana in settimana e sono circa 6 per settimana.

Operazione 38			
Concetto	Costrutto	Accessi	Tipo
Sessione	E	6	L
Corrispondenza	R	N*6	L

Dove:

$$N = \frac{\text{Corrispondenza}}{\text{Sessione}} = \frac{20000}{1750} \approx 11$$

Procediamo con il calcolo del costo totale dell'operazione 38 senza ridondanza.

$$\text{Costo totale} = 6 \times (N + 1) = 72$$

Inserimento della ridondanza

Operazione 34			
Concetto	Costrutto	Accessi	Tipo
Sessione	E	1	L
Sessione	E	1	S

Procediamo con il calcolo del costo totale dell'operazione 34 con l'aggiunta della ridondanza.

$$\text{Costo totale} = \text{Sessione} \times (1 + 2) = 5250$$

Operazione 38			
Concetto	Costrutto	Accessi	Tipo
Sessione	E	6	L

Procediamo con il calcolo del costo totale dell'operazione 38 con l'aggiunta della ridondanza.

$$\text{Costo totale} = 6$$

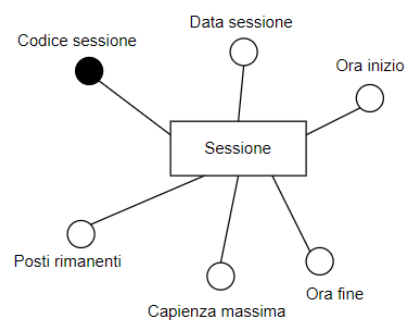
Riportiamo di seguito il costo totale relativo alle operazioni 34 e 38, tenendo conto della frequenza con la quale essa viene effettuata in un periodo di 5 anni.

Operazione	Costo	Frequenza(giornaliera)	Totale
34	21000	10	383250000 ($3,8 \times 10^8$)
38	72	20	2628000 ($2,6 \times 10^6$)
Costo operazioni senza ridondanza			385878000

Operazione	Costo	Frequenza(giornaliera)	Totale
34	5250	10	95812500
38	6	20	219000
Costo operazioni con ridondanza			96031500

Si ottiene un fattore di riduzione delle operazioni di circa 4 volte, quindi conviene inserire la ridondanza.

Viene quindi effettuata la seguente modifica allo schema:



3.3) ELIMINAZIONE DELLE GERARCHIE

Di seguito ci preoccupiamo di elencare le principali operazioni che hanno portato alla ristrutturazione dello schema E-R, in particolare dal punto di vista delle gerarchie.

1) Per quanto riguarda l'entità Abbonamento è stato deciso di accorpare le figlie della generalizzazione nel genitore in quanto abbiamo riscontrato che gli accessi al padre e alle figlie sono contestuali. Abbiamo deciso di optare per questa soluzione in quanto gli attributi caratterizzanti le entità figlie erano davvero ridotti e pertanto l'accorpamento delle figlie nell'entità padre non avrebbe causato alcun disturbo dal punto di vista di consumo errato della memoria.

2) Per quanto riguarda l'entità Dipendente è stato deciso di sostituire la generalizzazione nelle figlie in quanto abbiamo notato che gli attributi caratterizzanti le entità figlie erano di nostro particolare interesse e ci preoccupava dunque poter trattare le entità figlie singolarmente, in modo da separare i loro accessi da quelli del padre.

Grazie a questa scelta siamo riusciti a tenere sia le entità figlie (ognuna con i suoi attributi caratterizzanti), sia l'entità padre (con gli attributi comuni alla gerarchia). Ovviamente a seguito dell'applicazione di questa metodologia le entità figlie saranno collegate all'entità padre per mezzo di nuove relazioni.

3) Per quanto riguarda l'entità Transazione è stato deciso seguire la stessa tecnica utilizzata per l'entità dipendente, ovvero abbiamo deciso di sostituire la generalizzazione nelle figlie. Il motivo che ci ha spinto verso questa scelta è di implicanza logica e concettuale. Parlando di diversi tipi di transazione, ognuna con le sue caratteristiche e procedure, ci è sembrato corretto trattare anche in questo caso le entità figlie singolarmente.

3.4) PARTIZIONAMENTO/ACCORPAMENTO DI CONCETTI

All'interno del nostro diagramma E-R non abbiamo ritenuto necessario effettuare alcuna operazione di partizionamento o di accorpamento di concetti in quanto anche se avessimo effettuato partizionamenti verticali di entità, partizionamenti orizzontali di relationship o accorpamenti di entità/relationship non avremmo ottenuto nessun miglioramento significativo in termini di efficienza.

3.5) ELIMINAZIONE DEGLI ATTRIBUTI COMPOSTI

All'interno del nostro diagramma E-R abbiamo riscontrato la presenza di due attributi composti:

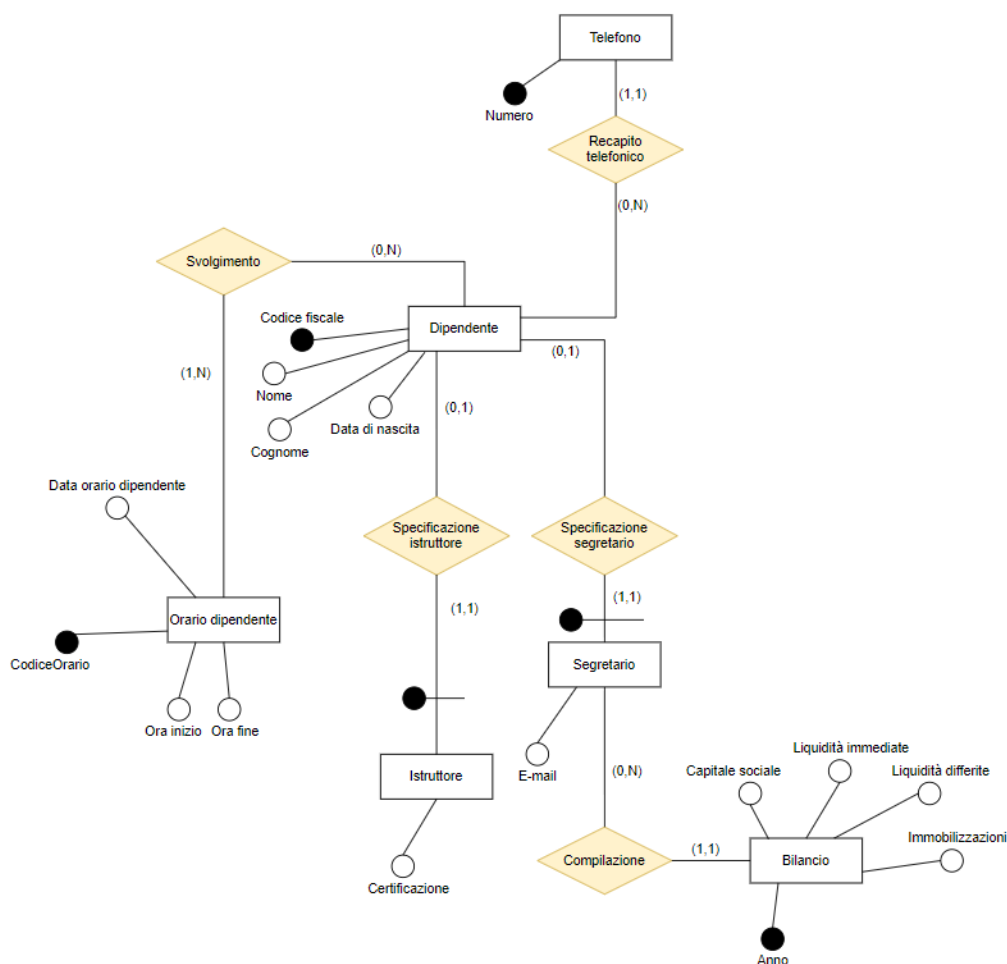
- 1) L'attributo composto Dati anagrafici, relativo alle entità Iscritto e Dipendente
- 2) L'attributo composto Credenziali, relativo all'entità Iscritto

Il problema viene risolto collegando ogni attributo presente all'interno dell'attributo composto direttamente con l'entità di riferimento.

3.6) ELIMINAZIONE DEGLI ATTRIBUTI MULTIVALEORE

L'unico attributo multivaleore individuato è il Recapito telefonico relativo all'entità Dipendente.

Riportiamo di seguito la ristrutturazione effettuata in merito:



3.7) ELENCO DEGLI IDENTIFICATORI PRINCIPALI

ENTITÀ		IDENTIFICATORE
Iscritto		Codice fiscale
Sessione		Codice sessione
Prenotazione		Codice sessione, Codice fiscale
Corso		Nome
Dipendente		Codice fiscale
Istruttore		Codice fiscale
Segretario		Codice fiscale
Telefono		Numero
Orario dipendente		Codice orario
Bilancio		Anno
Busta paga		Codice
Prodotto		Nome
Assicurazione		ID
Abbonamento		Transazione, Tipo
Transazione busta paga		ID
Transazione acquisto		ID
Transazione assicurazione		ID
Transazione abbonamento		ID
Transazione		ID

3.8) NORMALIZZAZIONE

3.8.1) Associazioni

Il diagramma E-R ristrutturato risulta essere in forma normale di Boyce-Codd in quanto rispetta le caratteristiche fondamentali del modello relazionale (1FN) e in essa ogni determinante è una chiave candidata, cioè ogni attributo dal quale dipendono altri attributi può svolgere la funzione di chiave.

3.8.2) Entità

ENTITÀ	COMMENTO
Iscritto	Non esistono dipendenze non banali fra gli attributi.
Sessione	Non esistono dipendenze non banali fra gli attributi.
Prenotazione	Non esistono dipendenze non banali fra gli attributi.
Corso	Non esistono dipendenze non banali fra gli attributi.
Dipendente	Non esistono dipendenze non banali fra gli attributi.
Istruttore	Non esistono dipendenze non banali fra gli attributi.
Segretario	Non esistono dipendenze non banali fra gli attributi.
Telefono	Non esistono dipendenze non banali fra gli attributi.
Orario dipendente	Non esistono dipendenze non banali fra gli attributi.
Bilancio	Non esistono dipendenze non banali fra gli attributi.
Busta paga	Non esistono dipendenze non banali fra gli attributi.
Prodotto	Non esistono dipendenze non banali fra gli attributi.
Assicurazione	Non esistono dipendenze non banali fra gli attributi.
Abbonamento	Non esistono dipendenze non banali fra gli attributi.
Transazione busta paga	Non esistono dipendenze non banali fra gli attributi.
Transazione acquisto	Non esistono dipendenze non banali fra gli attributi.
Transazione assicurazione	Non esistono dipendenze non banali fra gli attributi.
Transazione abbonamento	Non esistono dipendenze non banali fra gli attributi.
Transazione	Non esistono dipendenze non banali fra gli attributi.

3.9) TRADUZIONE VERSO IL MODELLO RELAZIONALE

Entità-Relazione		Traduzione	Vincoli
Iscritto	Iscritto(<u>CF</u> , Email, Password, Username, DataNascita, Cognome, Nome)		
Accesso	Accesso(<u>CF</u> , <u>CodSessione</u>)		CF->Iscritto.CF, CodSessione->Sessione.CodSessione
Sessione	Sessione(<u>CodSessione</u> , OraInizio, OraFine, Capienza, PostiRim, DataSessione)		
Prenotazione	Prenotazione(<u>Iscritto</u> , <u>Sessione</u> , DataPrenotazione, OraPrenotazione)		Iscritto->Iscritto.CF, Sessione->Sessione.CodSessione
Corso	Corso(<u>Nome</u> , <u>CodIstruttore</u> , Luogo)		CodIstruttore->Istruttore.CodIstruttore
Afferenza	Afferenza(<u>CodSessione</u> , <u>Corso</u>)		CodSessione->Sessione.CodSessione, Corso->Corso.Nome
Dipendente	Dipendente(<u>CF</u> , Nome, Cognome, DataNascita)		
Istruttore	Istruttore(<u>CodIstruttore</u> , Certificazione)		CodIstruttore->Dipendente.CF
OrarioDip	OrarioDip(<u>CodOrario</u> , OraInizio, OraFine, DataOrarioDip)		
Svolgimento	Svolgimento(<u>Dipendente</u> , <u>CodOrario</u>)		Dipendente->Dipendente.CF, CodOrario->OrarioDip.CodOrario
Segretario	Segretario(<u>CodSegretario</u> , Email)		CodSegretario->Segretario.CF
Telefono	Telefono(<u>Numero</u> , CF)		CF->Dipendente.CF
Bilancio	Bilancio(<u>Anno</u> , CodSegretario, CapitaleSociale, LiqImm, LiqDiff, Immobilizzazioni)		CodSegretario->Segretario.CodSegretario
Abbonamento	Abbonamento(<u>Transazione</u> , <u>Tipo</u> , CodSegretario, Durata, Ingressi, EntrateRimanenti)		CodSegretario->Segretario.CodSegretario, Transazione->TransazioneAbb.ID
Assicurazione	Assicurazione(<u>Transazione</u> , CodSegretario, Massimale, Condizione, DataAssicurazione)		CodSegretario->Segretario.CF, Transazione->TransazioneAss.ID
Prodotto	Prodotto(<u>Nome</u> , CodSegretario, Tipologia)		CodSegretario->Segretario.CodSegretario
EsecuzioneAcq	EsecuzioneAcq(<u>Prodotto</u> , <u>Transazione</u> , Quantita)		Transazione->TransazioneAcq.ID, Prodotto->Prodotto.Nome
BustaPaga	BustaPaga(<u>Codice</u> , Dipendente, Transazione, CodSegretario, Mensilita, Anno)		Dipendente->Dipendente.CF, CodSegretario->Segretario.CodSegretario, Transazione->TransazioneBP.ID
Transazione	Transazione(<u>ID</u> , Importo, DataTransazione, OraTransazione)		
TransazioneAbb	TransazioneAbb(<u>ID</u> , <u>Iscritto</u>)		ID->Transazione.ID, Iscritto->Iscritto.CF
TransazioneAss	TransazioneAss(<u>ID</u> , <u>Iscritto</u>)		ID->Transazione.ID, Iscritto->Iscritto.CF
TransazioneAcq	TransazioneAcq(<u>ID</u> , <u>Iscritto</u>)		ID->Transazione.ID, Iscritto->Iscritto.CF
TransazioneBP	TransazioneBP(<u>ID</u>)		ID->Transazione.ID

4) CODIFICA SQL E TESTING

4.1) DEFINIZIONE DELLO SCHEMA

Nella seguente sezione riportiamo la codifica in linguaggio SQL delle varie operazioni con relativi screenshot da terminale. Per completezza riportiamo anche l'elenco di tutte le tabelle costituenti il nostro database.

```
mysql> show tables;
+-----+
| Tables_in_arc |
+-----+
| Abbonamento  |
| Accesso       |
| Afferenza     |
| Assicurazione |
| Bilancio      |
| BustaPaga     |
| Corso         |
| Dipendente    |
| EsecuzioneAcq |
| Iscritto      |
| Istruttore    |
| OrarioDip     |
| Prenotazione  |
| Prodotto     |
| Segretario    |
| Sessione     |
| Svolgimento   |
| Telefono      |
| Transazione   |
| TransazioneAbb|
| TransazioneAcq|
| TransazioneAss|
| TransazioneBP |
+-----+
```

Di seguito ci limiteremo a riportare la codifica SQL delle creazioni delle singole tabelle. Purtroppo non ci è possibile riportare i contenuti di ogni tabella poiché abbiamo deciso di provare a rappresentare il più fedelmente possibile un database che fosse in costante funzione per un ciclo di vita pari a 5 anni. Di conseguenza numerose tabelle risultano riccamente popolate. In questo modo siamo riusciti a lavorare su un database che rispecchiasse il più fedelmente possibile lo scopo per cui è stato costruito.

Ne consegue dunque che il file dump.sql risulterà essere alquanto pesante, proprio per il motivo spiegato precedentemente.

CREAZIONE DELLA TABELLA Abbonamento:

```
CREATE TABLE `Abbonamento` (  
  `Transazione` int(11) NOT NULL,  
  `Tipo` varchar(50) NOT NULL,  
  `CodSegretario` varchar(16) DEFAULT NULL,  
  `Durata` varchar(20) DEFAULT NULL,  
  `Ingressi` int(11) DEFAULT NULL,  
  `EntrateRimanenti` int(11) DEFAULT NULL,  
  PRIMARY KEY (`Transazione`, `Tipo`),  
  KEY `CodSegretario` (`CodSegretario`),  
  CONSTRAINT `Abbonamento_ibfk_1` FOREIGN KEY (`CodSegretario`) REFERENCES `Segretario`  
  (`CodSegretario`) ON DELETE SET NULL ON UPDATE CASCADE,  
  CONSTRAINT `Abbonamento_ibfk_2` FOREIGN KEY (`Transazione`) REFERENCES `TransazioneAbb` (`ID`)  
  ON DELETE NO ACTION ON UPDATE CASCADE,  
  CONSTRAINT `CONSTRAINT_1` CHECK (`Ingressi` > 0 and `EntrateRimanenti` >= 0),  
  CONSTRAINT `CONSTRAINT_2` CHECK (`Durata` = 'Settimanale' or `Durata` = 'Mensile' or `Durata` =  
  'Annuale')  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

CREAZIONE DELLA TABELLA Accesso:

```
CREATE TABLE `Accesso` (  
  `CF` varchar(16) NOT NULL,  
  `Codsessione` int(11) NOT NULL,  
  PRIMARY KEY (`CF`, `Codsessione`),  
  KEY `Codsessione` (`Codsessione`),  
  CONSTRAINT `Accesso_ibfk_1` FOREIGN KEY (`CF`) REFERENCES `Iscritto` (`CF`) ON DELETE CASCADE ON  
  UPDATE CASCADE,  
  CONSTRAINT `Accesso_ibfk_2` FOREIGN KEY (`Codsessione`) REFERENCES `Sessione` (`CodSessione`) ON  
  DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

CREAZIONE DELLA TABELLA Afferenza:

```
CREATE TABLE `Afferenza` (  
  `CodSessione` int(11) NOT NULL,  
  `Corso` varchar(50) NOT NULL,  
  PRIMARY KEY (`CodSessione`,`Corso`),  
  KEY `Corso` (`Corso`),  
  CONSTRAINT `Afferenza_ibfk_1` FOREIGN KEY (`CodSessione`) REFERENCES `Sessione` (`CodSessione`) ON  
DELETE CASCADE ON UPDATE CASCADE,  
  CONSTRAINT `Afferenza_ibfk_2` FOREIGN KEY (`Corso`) REFERENCES `Corso` (`Nome`) ON DELETE  
CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

CREAZIONE DELLA TABELLA Assicurazione:

```
CREATE TABLE `Assicurazione` (  
  `Transazione` int(11) NOT NULL,  
  `CodSegretario` varchar(16) DEFAULT NULL,  
  `Massimale` decimal(15,2) DEFAULT NULL,  
  `Condizione` text DEFAULT NULL,  
  `DataAssicurazione` date DEFAULT NULL,  
  PRIMARY KEY (`Transazione`),  
  CONSTRAINT `Assicurazione_ibfk_1` FOREIGN KEY (`Transazione`) REFERENCES `TransazioneAss` (`ID`) ON  
DELETE NO ACTION ON UPDATE CASCADE,  
  CONSTRAINT `CONSTRAINT_1` CHECK (`Massimale` > 0)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

CREAZIONE DELLA TABELLA Bilancio:

```
CREATE TABLE `Bilancio` (  
  `Anno` year(4) NOT NULL,  
  `CodSegretario` varchar(16) DEFAULT NULL,  
  `CapitaleSociale` decimal(15,2) DEFAULT NULL,  
  `LiqImm` decimal(15,2) DEFAULT NULL,  
  `LiqDiff` decimal(15,2) DEFAULT NULL,  
  `Immobilizzazioni` decimal(15,2) DEFAULT NULL,  
  PRIMARY KEY (`Anno`),  
  KEY `CodSegretario` (`CodSegretario`),  
  CONSTRAINT `Bilancio_ibfk_1` FOREIGN KEY (`CodSegretario`) REFERENCES `Segretario` (`CodSegretario`)  
  ON DELETE NO ACTION ON UPDATE CASCADE,  
  CONSTRAINT `CONSTRAINT_1` CHECK (`CapitaleSociale` > 0 and `Immobilizzazioni` > 0)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

CREAZIONE DELLA TABELLA BustaPaga:

```
CREATE TABLE `BustaPaga` (  
  `Codice` int(11) NOT NULL AUTO_INCREMENT,  
  `Dipendente` varchar(16) DEFAULT NULL,  
  `Transazione` int(11) DEFAULT NULL,  
  `CodSegretario` varchar(16) DEFAULT NULL,  
  `Mensilita` varchar(15) DEFAULT NULL,  
  `Anno` year(4) DEFAULT NULL,  
  PRIMARY KEY (`Codice`),  
  UNIQUE KEY `Dipendente_2` (`Dipendente`, `Mensilita`, `Anno`),  
  KEY `Dipendente` (`Dipendente`),  
  KEY `CodSegretario` (`CodSegretario`),  
  KEY `Transazione` (`Transazione`),  
  CONSTRAINT `BustaPaga_ibfk_1` FOREIGN KEY (`Dipendente`) REFERENCES `Dipendente` (`CF`) ON DELETE  
  NO ACTION ON UPDATE CASCADE,  
  CONSTRAINT `BustaPaga_ibfk_2` FOREIGN KEY (`CodSegretario`) REFERENCES `Segretario`  
  (`CodSegretario`) ON DELETE NO ACTION ON UPDATE CASCADE,
```

```
CONSTRAINT `BustaPaga_ibfk_3` FOREIGN KEY (`Transazione`) REFERENCES `TransazioneBP` (`ID`) ON  
DELETE NO ACTION ON UPDATE CASCADE,
```

```
CONSTRAINT `CONSTRAINT_1` CHECK (`Mensilita` = 'Gennaio' or `Mensilita` = 'Febbraio' or `Mensilita` =  
'Marzo' or `Mensilita` = 'Aprile' or `Mensilita` = 'Maggio' or `Mensilita` = 'Giugno' or `Mensilita` = 'Luglio' or  
`Mensilita` = 'Agosto' or `Mensilita` = 'Settembre' or `Mensilita` = 'Ottobre' or `Mensilita` = 'Novembre' or  
`Mensilita` = 'Dicembre' or `Mensilita` = 'Tredicesima')
```

```
) ENGINE=InnoDB AUTO_INCREMENT=1320470 DEFAULT CHARSET=utf8mb4;
```

CREAZIONE DELLA TABELLA Corso:

```
CREATE TABLE `Corso` (
```

```
`Nome` varchar(50) NOT NULL,
```

```
`CodIstruttore` varchar(16) DEFAULT NULL,
```

```
`Luogo` varchar(20) DEFAULT NULL,
```

```
PRIMARY KEY (`Nome`),
```

```
KEY `CodIstruttore` (`CodIstruttore`),
```

```
CONSTRAINT `Corso_ibfk_1` FOREIGN KEY (`CodIstruttore`) REFERENCES `Istruttore` (`CodIstruttore`) ON  
DELETE SET NULL ON UPDATE CASCADE,
```

```
CONSTRAINT `CONSTRAINT_1` CHECK (`Luogo` = 'Indoor' or `Luogo` = 'Outdoor')
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

CREAZIONE DELLA TABELLA Dipendente:

```
CREATE TABLE `Dipendente` (
```

```
`CF` varchar(16) NOT NULL,
```

```
`Nome` varchar(40) DEFAULT NULL,
```

```
`Cognome` varchar(40) DEFAULT NULL,
```

```
`DataNascita` date DEFAULT NULL,
```

```
PRIMARY KEY (`CF`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

CREAZIONE DELLA TABELLA EsecuzioneAcq:

```
CREATE TABLE `EsecuzioneAcq` (
```

```
`Prodotto` varchar(20) NOT NULL,
```

```
`Transazione` int(11) NOT NULL,
```

```

`Quantita` int(11) DEFAULT NULL,
PRIMARY KEY (`Prodotto`,`Transazione`),
UNIQUE KEY `Transazione_2` (`Transazione`),
KEY `Transazione` (`Transazione`),
CONSTRAINT `EsecuzioneAcq_ibfk_1` FOREIGN KEY (`Prodotto`) REFERENCES `Prodotto` (`Nome`) ON
DELETE NO ACTION ON UPDATE CASCADE,
CONSTRAINT `EsecuzioneAcq_ibfk_2` FOREIGN KEY (`Transazione`) REFERENCES `TransazioneAcq` (`ID`)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT `CONSTRAINT_1` CHECK (`Quantita` > 0)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

```

CREAZIONE DELLA TABELLA Iscritto:

```

CREATE TABLE `Iscritto` (
`CF` varchar(16) NOT NULL,
`Email` varchar(70) DEFAULT NULL,
`Password` varchar(70) DEFAULT NULL,
`Username` varchar(70) DEFAULT NULL,
`DataNascita` date DEFAULT NULL,
`Cognome` varchar(20) DEFAULT NULL,
`Nome` varchar(40) DEFAULT NULL,
PRIMARY KEY (`CF`),
CONSTRAINT `CONSTRAINT_1` CHECK (`Email` like '%@%.%')
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

```

CREAZIONE DELLA TABELLA Istruttore:

```

CREATE TABLE `Istruttore` (
`CodIstruttore` varchar(16) NOT NULL,
`Certificazione` varchar(50) DEFAULT NULL,
PRIMARY KEY (`CodIstruttore`),
CONSTRAINT `Istruttore_ibfk_1` FOREIGN KEY (`CodIstruttore`) REFERENCES `Dipendente` (`CF`) ON
DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

```

CREAZIONE DELLA TABELLA OrarioDip:

```
CREATE TABLE `OrarioDip` (  
  `CodOrario` int(11) NOT NULL AUTO_INCREMENT,  
  `OraInizio` time DEFAULT NULL,  
  `OraFine` time DEFAULT NULL,  
  `DataOrarioDip` date DEFAULT NULL,  
  PRIMARY KEY (`CodOrario`),  
  CONSTRAINT `CONSTRAINT_1` CHECK (`OraInizio` > '00:00:00' and `OraInizio` < '23:59:59'),  
  CONSTRAINT `CONSTRAINT_2` CHECK (`OraFine` > '00:00:00' and `OraFine` < '23:59:59'),  
  CONSTRAINT `CONSTRAINT_3` CHECK (`OraInizio` < `OraFine`)  
) ENGINE=InnoDB AUTO_INCREMENT=3929 DEFAULT CHARSET=utf8mb4;
```

CREAZIONE DELLA TABELLA Prenotazione:

```
CREATE TABLE `Prenotazione` (  
  `Iscritto` varchar(16) NOT NULL,  
  `Sessione` int(11) NOT NULL,  
  `DataPrenotazione` date DEFAULT NULL,  
  `OraPrenotazione` time DEFAULT NULL,  
  PRIMARY KEY (`Iscritto`,`Sessione`),  
  KEY `Sessione` (`Sessione`),  
  CONSTRAINT `Prenotazione_ibfk_1` FOREIGN KEY (`Iscritto`) REFERENCES `Iscritto` (`CF`),  
  CONSTRAINT `Prenotazione_ibfk_2` FOREIGN KEY (`Sessione`) REFERENCES `Sessione` (`CodSessione`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

CREAZIONE DELLA TABELLA Prodotto:

```
CREATE TABLE `Prodotto` (  
  `Nome` varchar(50) NOT NULL,  
  `CodSegretario` varchar(16) DEFAULT NULL,  
  `Tipologia` varchar(20) DEFAULT NULL,  
  PRIMARY KEY (`Nome`),  
  KEY `CodSegretario` (`CodSegretario`),
```

```
CONSTRAINT `Prodotto_ibfk_1` FOREIGN KEY (`CodSegretario`) REFERENCES `Segretario` (`CodSegretario`)
ON DELETE NO ACTION ON UPDATE CASCADE,
```

```
CONSTRAINT `CONSTRAINT_1` CHECK (`Tipologia` = 'Scarpette' or `Tipologia` = 'Magnesite' or `Tipologia` =
'Casco' or `Tipologia` = 'Corda' or `Tipologia` = 'Imbracatura' or `Tipologia` = 'Moschettone' or `Tipologia` =
'Carrucola' or `Tipologia` = 'Picchetto' or `Tipologia` = 'Maglietta' or `Tipologia` = 'Pantalone' or `Tipologia` =
'Cappello')
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

CREAZIONE DELLA TABELLA Segretario:

```
CREATE TABLE `Segretario` (
```

```
`CodSegretario` varchar(16) NOT NULL,
```

```
`Email` varchar(50) DEFAULT NULL,
```

```
PRIMARY KEY (`CodSegretario`),
```

```
CONSTRAINT `Segretario_ibfk_1` FOREIGN KEY (`CodSegretario`) REFERENCES `Dipendente` (`CF`) ON
DELETE CASCADE ON UPDATE CASCADE,
```

```
CONSTRAINT `CONSTRAINT_1` CHECK (`Email` like '%@%.%')
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

CREAZIONE DELLA TABELLA Sessione:

```
CREATE TABLE `Sessione` (
```

```
`Codsessione` int(11) NOT NULL,
```

```
`OraInizio` time DEFAULT NULL,
```

```
`OraFine` time DEFAULT NULL,
```

```
`Capienza` int(11) NOT NULL,
```

```
`Postirim` int(11) DEFAULT `Capienza`,
```

```
`DataSessione` date DEFAULT NULL,
```

```
PRIMARY KEY (`Codsessione`),
```

```
CONSTRAINT `CONSTRAINT_1` CHECK (`OraInizio` > '00:00:00' and `OraInizio` < '23:59:59'),
```

```
CONSTRAINT `CONSTRAINT_2` CHECK (`OraFine` > '00:00:00' and `OraFine` < '23:59:59'),
```

```
CONSTRAINT `CONSTRAINT_3` CHECK (`OraInizio` < `OraFine`),
```

```
CONSTRAINT `CONSTRAINT_4` CHECK (`Capienza` > 0 and `Capienza` <= 30),
```

```
CONSTRAINT `CONSTRAINT_5` CHECK (`Postirim` >= 0 and `Postirim` <= `Capienza`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```


CREAZIONE DELLA TABELLA Svolgimento:

```
CREATE TABLE `Svolgimento` (  
  `Dipendente` varchar(16) NOT NULL,  
  `CodOrario` int(11) NOT NULL,  
  PRIMARY KEY (`Dipendente`,`CodOrario`),  
  KEY `CodOrario` (`CodOrario`),  
  CONSTRAINT `Svolgimento_ibfk_1` FOREIGN KEY (`Dipendente`) REFERENCES `Dipendente` (`CF`) ON  
DELETE CASCADE ON UPDATE CASCADE,  
  CONSTRAINT `Svolgimento_ibfk_2` FOREIGN KEY (`CodOrario`) REFERENCES `OrarioDip` (`CodOrario`) ON  
DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

CREAZIONE DELLA TABELLA Telefono:

```
CREATE TABLE `Telefono` (  
  `Numero` varchar(20) NOT NULL,  
  `CF` varchar(16) DEFAULT NULL,  
  PRIMARY KEY (`Numero`),  
  KEY `CF` (`CF`),  
  CONSTRAINT `Telefono_ibfk_1` FOREIGN KEY (`CF`) REFERENCES `Dipendente` (`CF`) ON DELETE CASCADE  
ON UPDATE CASCADE  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

CREAZIONE DELLA TABELLA Transazione:

```
CREATE TABLE `Transazione` (  
  `ID` int(11) NOT NULL,  
  `Importo` decimal(15,2) DEFAULT NULL,  
  `DataTransazione` date DEFAULT NULL,  
  `OraTransazione` time DEFAULT NULL,  
  PRIMARY KEY (`ID`),  
  CONSTRAINT `CONSTRAINT_1` CHECK (`Importo` > 0),  
  CONSTRAINT `CONSTRAINT_2` CHECK (`OraTransazione` > '00:00:00' and `OraTransazione` < '23:59:59')  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

CREAZIONE DELLA TABELLA TransazioneAbb:

```
CREATE TABLE `TransazioneAbb` (  
  `ID` int(11) NOT NULL,  
  `Iscritto` varchar(16) DEFAULT NULL,  
  PRIMARY KEY (`ID`),  
  KEY `Iscritto` (`Iscritto`),  
  CONSTRAINT `TransazioneAbb_ibfk_1` FOREIGN KEY (`ID`) REFERENCES `Transazione` (`ID`) ON DELETE  
  CASCADE ON UPDATE CASCADE,  
  CONSTRAINT `TransazioneAbb_ibfk_2` FOREIGN KEY (`Iscritto`) REFERENCES `Iscritto` (`CF`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

CREAZIONE DELLA TABELLA TransazioneAcq:

```
CREATE TABLE `TransazioneAcq` (  
  `ID` int(11) NOT NULL,  
  `Iscritto` varchar(16) DEFAULT NULL,  
  PRIMARY KEY (`ID`),  
  KEY `Iscritto` (`Iscritto`),  
  CONSTRAINT `TransazioneAcq_ibfk_1` FOREIGN KEY (`ID`) REFERENCES `Transazione` (`ID`) ON DELETE  
  CASCADE ON UPDATE CASCADE,  
  CONSTRAINT `TransazioneAcq_ibfk_2` FOREIGN KEY (`Iscritto`) REFERENCES `Iscritto` (`CF`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

CREAZIONE DELLA TABELLA TransazioneAss:

```
CREATE TABLE `TransazioneAss` (  
  `ID` int(11) NOT NULL,  
  `Iscritto` varchar(16) DEFAULT NULL,  
  PRIMARY KEY (`ID`),  
  KEY `Iscritto` (`Iscritto`),  
  CONSTRAINT `TransazioneAss_ibfk_1` FOREIGN KEY (`ID`) REFERENCES `Transazione` (`ID`) ON DELETE  
  CASCADE ON UPDATE CASCADE,  
  CONSTRAINT `TransazioneAss_ibfk_2` FOREIGN KEY (`Iscritto`) REFERENCES `Iscritto` (`CF`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

CREAZIONE DELLA TABELLA TransazioneBP:

```
CREATE TABLE `TransazioneBP` (  
  `ID` int(11) NOT NULL,  
  PRIMARY KEY (`ID`),  
  CONSTRAINT `TransazioneBP_ibfk_1` FOREIGN KEY (`ID`) REFERENCES `Transazione` (`ID`) ON DELETE  
  CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

4.2) CODIFICA DELLE OPERAZIONI E SCREENSHOT CHE NE VERIFICANO L'ESECUZIONE

1) Inserimento nuovo atleta/iscritto (in media 5 volte al mese)

```
insert into Iscritto(CF,Email>Password,Username>DataNascita,Cognome,Nome)  
values(...);
```

2) Inserimento nuovi prodotti nel catalogo (in media 2 volte al mese)

```
insert into Prodotto(Nome,CodSegretario,Tipologia)  
values(...);
```

3) Inserimento nuova transazione abbonamento (in media 3 volte a settimana)

```
insert into Transazione(ID,Importo>DataTransazione,OraTransazione)  
values(...);  
insert into TransazioneAbb(ID,Iscritto)  
values(<ID>,...);
```

4) Inserimento nuova transazione prodotto (in media 4 volte a settimana)

```
insert into Transazione(ID,Importo>DataTransazione,OraTransazione)  
values(...);  
insert into TransazioneAcq(ID,Iscritto)  
values(<ID>,...);
```

5) Inserimento nuova transazione assicurazione (in media 5 volte al mese)

```
insert into Transazione(ID,Importo,DataTransazione,OraTransazione)  
values(...);  
insert into TransazioneAss(ID,Iscritto)  
values(<ID>,...);
```

6) Inserimento nuova Transazione BustaPaga (in media 15 volte al mese)

```
insert into Transazione(ID, Importo, DataTransazione, OraTransazione)  
values(...);  
insert into TransazioneBP(ID)  
values(<ID>);
```

7) Inserimento nuova prenotazione (in media 60 volte a settimana)

```
insert into Prenotazione (Iscritto, Sessione, DataPrenotazione, OraPrenotazione)  
values(...);  
  
/*vanno aggiornati i posti rimanenti per la sessione corrispondente*/  
  
update Sessione  
  
set PostiRim = Postirim -1  
  
where CodSessione="<codice_della_sessione_prenotata>;
```

8) Inserimento accesso dell' atleta in palestra (in media 30 volte al giorno)

```
insert into Accesso(CF,CodSessione)  
values(...);  
  
/*va aggiornato il numero di entrate rimanenti dell' abbonamento*/  
  
update Abbonamento  
  
set EntrateRimanenti = EntrateRimanenti - 1  
  
where Abbonamento.Transazione=<id_transazione> and  
  
Abbonamento.Tipo="<tipo_abbonamento>;
```

9) Inserimento nuovo dipendente (in media 2 volte l'anno)

insert into Dipendente(CF,Nome,Cognome,DataNascita)

values(...);

*/*se è un segretario*/*

insert into Segretario(CodSegretario,Email)

values("<codice_fiscale>",...);

*/*se è un istruttore*/*

insert into Istruttore(CodIstruttore,Certificazione)

values("<codice_fiscale>",...);

*/*per il numero di telefono*/*

insert into Telefono(Numero,Cf)

values(...,"<codice_fiscale>");

10) Inserimento nuova sessione d' allenamento (in media 10 volte a settimana)

insert into Sessione(CodSessione,Orainizio,OraFine,Capienza,DataSessione)

values(...);

*/*a seconda di quali corsi ci sono bisogna inserire un' afferenza*/*

insert into Afferenza(CodSessione,Corso)

values("<codice_sessione>", "<nome_corso>");

11) Inserimento nuovo corso (in media 3 volte l'anno)

insert into corso(Nome,CodIstruttore,Luogo)

values(...);

12) Stipulazione del tipo di abbonamento (in media 3 volte a settimana)

```
Insert into Transazione(id,Importo,DataTransazione,OraTransazione)  
values(...);
```

```
Insert into TransazioneAbb(Id,Iscritto)  
values(...);
```

```
Insert into  
Abbonamento(Transazione,Tipo,CodSegretario,Durata,Ingressi,EntrateRimanenti)  
values(...);
```

13) Modifica dati atleta/iscritto (in media 3 volte al mese)

```
update Iscritto  
  
set Email="<Nuova_Email>", Password="<Nuova_Password>",  
Username="<Nuovo_Username>", DataNascita="<Nuova_Data>",  
Cognome="<Nuovo_Cognome>", Nome="<Nuovo_Nome>"  
where CF="<CF_Iscritto>";
```

14) Modifica dati dei prodotti (in media 2 volte a settimana)

```
update Prodotto  
  
set CodSegretario="<Nuovo_CodiceSegretario>",  
Tipologia="<Nuova_Tipologia>"  
where Nome="<Nome_Prodotto>";
```

15) Modifica dati dipendente (in media 3 volte al mese)

```
update Dipendente  
  
set Nome="<Nuovo_Nome>", Cognome="<Nuovo_Cognome>",  
DataNascita="<Nuova_Data>"  
where CF="<CF_Dipendente>";
```

16) Modifica sessione di allenamento (in media 6 volte a settimana)

```
update Sessione  
  
set Oralnizio="<Nuova_Oralnizio>", OraFine="<Nuova_OraFine>",  
Capienza=<Nuova_Capienza>, Postirim=<Nuovi_PostiRimanenti>;
```

DataSession="<Nuova_DataSession>"

where CodSession=<Codice_Session>;

17) Cancellazione atleta/iscritto (in media una volta al mese)

delete from Iscritto

where CF="<CF_Iscritto>;

18) Cancellazione dipendente (in media 3 volte l'anno)

delete from Dipendente

where CF="<CF_Dipendente>;

19) Cancellazione prodotto/prodotti (in media 2 volte al mese)

delete from Prodotto

where Nome="<Nome_Prodotto>;

20) Cancellazione sessione di allenamento (in media 2 volte al mese)

delete from Sessione

where CodSession=<Codice_Session>;

21) Cancellazione corso (in media 2 volte l'anno)

delete from Corso

where Nome="<Nome_Corso>;

22) Consultazione dati dell'atleta/iscritto (in media 2 volte al giorno)

22.1) Visualizzare gli iscritti maggiorenni

select Iscritto.CF, Iscritto.Nome, Iscritto.Cognome

from Iscritto

where TIMESTAMPDIFF(YEAR,Iscritto.DataNascita, NOW()) > 18;

```
mysql> select Iscritto.CF, Iscritto.Nome, Iscritto.Cognome
-> from Iscritto
-> where TIMESTAMPDIFF( YEAR, Iscritto.DataNascita, NOW() ) > 18;
```

CF	Nome	Cognome
BCDGD74A13GECZP	GUIDO	BOCEDI
BDDNNM83M10MOKHH	ANNAMARIA	BADODI
BDGLSS86L83TCIHK	ALESSANDRA	BEDOGNI
BDNLDV86E69ZTTPT	LUDOVICA	BEDINI
BDSNCH93R740IYTY	NICHOLAS	BEDESCHI
BGGDNX84A76LBZTV	DIANA	BEGGI
BGNRG83H39RYQMJ	GIORGIA	BAGNACANI
BGNNDX76L150GQJW	NEDO	BAGNI
BGNSVT93M415QADR	SVITLANA	BAGNOLI
BLDNMX68E14SILGX	NOEMI	BALDI
BLGMMX89L63PJAYT	MIA	BULGARELLI
BLGNG188C21CREWS	ANGELO	BOLOGNESI
BLLLFN93T64GMNHT	ALFONSO	BELLI
BLLVNN97T69XRQEX	VANNA	BALLABENI
BLTSTF85C69IZXQK	STEFANIA	BELTRAMI
BNCDNI94C83MVOYW	DANTIO	BIANCHINI

22.2) Visualizzare i dati di uno specifico iscritto

Select * from Iscritto where CF="<codice fiscale>;

```
mysql> select *
-> From Iscritto
-> where CF="ZNCLCX70M48MPNVN";
```

CF	Email	Password	Username	DataNascita	Cognome	Nome
ZNCLCX70M48MPNVN	ALICE.ZANICHELLI@gmail.com	ZANICHELLI26807	ALICE42229	1981-11-29	ZANICHELLI	ALICE

1 row in set (0.09 sec)

23) Consultazione dati dipendente (in media una volta a settimana)

23.1) Visualizzare i turni lavorativi in un determinato periodo

**select Svolgimento.Dipendente, OrarioDip.OraInizio, OrarioDip.OraFine,
OrarioDip.DataOrarioDip**

from Svolgimento, OrarioDip

where OrarioDip.DataOrarioDip>="<Data_inizio>"

and OrarioDip.DataOrarioDip<="Data_fine"

and Svolgimento.CodOrario=OrarioDip.CodOrario

order by OrarioDip.DataOrarioDip;

Esempio: visualizzare i turni lavorativi dal 10-01-2020 al 17-01-2020

```
mysql> select Svolgimento.Dipendente, OrarioDip.OraInizio, OrarioDip.OraFine,
-> OrarioDip.DataOrarioDip
-> from Svolgimento, OrarioDip
-> where OrarioDip.DataOrarioDip>="2020-01-10"
-> and OrarioDip.DataOrarioDip<="2020-01-17"
-> and Svolgimento.CodOrario=OrarioDip.CodOrario
-> order by OrarioDip.DataOrarioDip;
```

Dipendente	OraInizio	OraFine	DataOrarioDip
BLNLRX76R30WZJUX	15:45:00	18:15:00	2020-01-10
GRLP5Q71C300CUWU	15:45:00	18:15:00	2020-01-10
CLMRNL86H27K0ODC	09:45:00	12:15:00	2020-01-11
LTTDUX75E28U2LJK	09:45:00	12:15:00	2020-01-12
NCLDDX96L26ZHKGB	15:45:00	18:15:00	2020-01-12
BTTLX65L270GNUN	09:45:00	12:15:00	2020-01-13
XXXMMC79M27TYKER	15:45:00	18:15:00	2020-01-14
PPLBRN96P22PZAHG	09:45:00	12:15:00	2020-01-14
FSCHLY89B23LWYDV	09:45:00	12:15:00	2020-01-15
YBRTR76C25WUJMG	09:45:00	12:15:00	2020-01-15
CRSNTL98P28MKOPI	15:45:00	18:15:00	2020-01-16
MRSRMN86H25JKYNU	15:45:00	18:15:00	2020-01-16
BLLURG66A24IJUYK	15:45:00	18:15:00	2020-01-17

23.2) Visualizzare i numeri di telefono di un determinato dipendente

select * from Telefono where Telefono.CF="<CF_Dipendente>;

Esempio: visualizzare i numeri di telefono del dipendente il cui codice fiscale è CSSNTL92T26LJKZR

```
mysql> select * from Telefono where Telefono.CF="CSSNTL92T26LJKZR";
+-----+-----+
| Numero      | CF          |
+-----+-----+
| +393316319315 | CSSNTL92T26LJKZR |
| +393689259551 | CSSNTL92T26LJKZR |
+-----+-----+
```

24) Consultazione dati istruttore (in media 2 volte a settimana)

24.1) Visualizzare gli istruttori con una determinata specializzazione

select * from Istruttore where Certificazione="<Nome_Certificazione>;

Esempio: visualizzare gli istruttori con la specializzazione in arrampicata agonistica

```
mysql> select * from Istruttore where Certificazione="Arrampicata agonistica";
+-----+-----+
| CodIstruttore | Certificazione |
+-----+-----+
| CMPFLC73A24KXHTW | Arrampicata agonistica |
| CRLRST90C31UNDBY | Arrampicata agonistica |
| GLNBGX98D31JBCIJ | Arrampicata agonistica |
| PPLBRN96P22PZAHG | Arrampicata agonistica |
| RCCRFF93L22IXMTI | Arrampicata agonistica |
| SCLCNS96B23XBQBJ | Arrampicata agonistica |
| SSXBRN80P26GAINT | Arrampicata agonistica |
+-----+-----+
```

25) Consultazione dati Segretario (in media 3 volte a settimana)

25.1) Visualizzare le transazioni relative agli abbonamenti registrate da un determinato segretario

**select Abbonamento.CodSegretario, Abbonamento.Transazione, Transazione.Importo,
Transazione.OraTransazione, Transazione.DataTransazione
from Abbonamento, Transazione
where CodSegretario="<CF_Segretario>" and
Transazione.ID=Abbonamento.Transazione
order by Transazione.DataTransazione, Transazione.OraTransazione;**

Esempio: visualizzare le transazioni registrate dal segretario il cui codice fiscale è

THRVTX80R26NGSUT

Nota: poiché le righe restituite dalla query sono circa 2000, riporteremo solo una sezione di queste righe

```
mysql> select Abbonamento.CodSegretario, Abbonamento.Transazione, Transazione.Importo,
-> Transazione.OraTransazione, Transazione.DataTransazione
-> from Abbonamento, Transazione
-> where CodSegretario="THRUTX80R26NGSUT" and Transazione.ID=Abbonamento.Transazione
-> order by Transazione.DataTransazione, Transazione.OraTransazione;
```

CodSegretario	Transazione	Importo	OraTransazione	DataTransazione
THRUTX80R26NGSUT	40529	50.00	18:32:42	2016-01-02
THRUTX80R26NGSUT	51520	20.00	21:27:50	2016-01-04
THRUTX80R26NGSUT	40063	50.00	13:31:28	2016-01-05
THRUTX80R26NGSUT	51630	20.00	11:19:56	2016-01-10
THRUTX80R26NGSUT	52048	50.00	13:31:21	2016-01-10
THRUTX80R26NGSUT	52742	450.00	11:25:37	2016-01-13
THRUTX80R26NGSUT	55369	50.00	19:09:41	2016-01-13
THRUTX80R26NGSUT	37894	50.00	15:09:37	2016-01-14
THRUTX80R26NGSUT	36483	450.00	20:35:31	2016-01-14
THRUTX80R26NGSUT	57967	50.00	16:51:06	2016-01-16
THRUTX80R26NGSUT	39805	450.00	17:35:54	2016-01-16
THRUTX80R26NGSUT	39327	20.00	13:37:19	2016-01-17
THRUTX80R26NGSUT	54700	20.00	18:42:58	2016-01-17
THRUTX80R26NGSUT	42529	20.00	19:08:53	2016-01-17
THRUTX80R26NGSUT	42807	50.00	10:03:59	2016-01-19
THRUTX80R26NGSUT	42479	50.00	12:40:07	2016-01-19
THRUTX80R26NGSUT	41741	450.00	15:27:46	2016-01-19
THRUTX80R26NGSUT	43517	20.00	15:33:46	2016-01-19
THRUTX80R26NGSUT	36727	20.00	11:25:37	2016-01-24
THRUTX80R26NGSUT	37093	50.00	13:59:18	2016-01-25
THRUTX80R26NGSUT	39619	20.00	15:05:31	2016-01-25
THRUTX80R26NGSUT	38857	450.00	18:56:18	2016-01-25
THRUTX80R26NGSUT	40435	50.00	13:54:17	2016-01-26
THRUTX80R26NGSUT	53545	450.00	19:35:46	2016-01-27
THRUTX80R26NGSUT	55140	50.00	20:13:22	2016-01-28
THRUTX80R26NGSUT	43730	50.00	20:53:38	2016-01-29
THRUTX80R26NGSUT	52487	50.00	12:12:42	2016-01-31
THRUTX80R26NGSUT	51217	450.00	14:43:57	2016-01-31
THRUTX80R26NGSUT	59096	20.00	11:06:58	2016-02-02
THRUTX80R26NGSUT	50777	450.00	13:32:20	2016-02-02
THRUTX80R26NGSUT	54097	450.00	11:21:04	2016-02-03
THRUTX80R26NGSUT	52265	450.00	11:37:28	2016-02-03
THRUTX80R26NGSUT	41874	450.00	14:03:08	2016-02-03
THRUTX80R26NGSUT	41727	20.00	17:03:04	2016-02-04
THRUTX80R26NGSUT	56138	450.00	19:04:52	2016-02-04
THRUTX80R26NGSUT	56772	450.00	18:40:58	2016-02-05

26) Consultazione dati dei prodotti (in media 5 volte a settimana)

26.1) Visualizzare i dati di un determinato prodotto

select * from Prodotto where Prodotto.Nome="<nome prodotto>;

Esempio: visualizzare i dati del prodotto Picchetto-N.346873

```
mysql> select * from Prodotto where Prodotto.Nome="Picchetto-N.346873";
```

Nome	CodSegretario	Tipologia
Picchetto-N.346873	THRUTX80R26NGSUT	Picchetto

26.2) Visualizzare il prodotto più venduto

select P.Nome, P.Tipologia from Prodotto as P

where

(select sum(Quantita) from EsecuzioneAcq group by Prodotto having Prodotto=P.Nome)=

(select max(somme) from (select sum(Quantita) as somme from EsecuzioneAcq

group by Prodotto) as somme);

```
mysql> select P.Nome, P.Tipologia from Prodotto as P
-> where
-> (select sum(Quantita) from EsecuzioneAcq group by Prodotto having Prodotto=P.Nome)=
-> (select max(somme) from (select sum(Quantita) as somme from EsecuzioneAcq
-> group by Prodotto) as somme);
+-----+-----+
| Nome          | Tipologia |
+-----+-----+
| Pantalone-N.527991 | Pantalone |
+-----+-----+
```

26.3) Visualizzare tutti dati dei prodotto appartenenti ad una categoria

select * from Prodotto where Tipologia="<nome categoria>";

Esempio: visualizzare tutti i prodotti appartenenti alla categoria Cappello

```
mysql> select * from Prodotto where Tipologia="Cappello";
+-----+-----+-----+
| Nome          | CodSegretario | Tipologia |
+-----+-----+-----+
| Cappello-N.110701 | TTLNNX86H24MBUUS | Cappello |
| Cappello-N.131008 | ULRZNX84D25R0TDJ | Cappello |
| Cappello-N.334651 | ZQJCSR70R24RYSQA | Cappello |
| Cappello-N.344895 | UUXCST90A31AWIAB | Cappello |
| Cappello-N.550998 | THRUTX80R26NGSUT | Cappello |
| Cappello-N.570072 | UNTMFL68S23FJFZG | Cappello |
| Cappello-N.690727 | ZQJCSR70R24RYSQA | Cappello |
| Cappello-N.726950 | UNTMFL68S23FJFZG | Cappello |
| Cappello-N.777241 | UTLDLN86A27LYJZU | Cappello |
| Cappello-N.837884 | XXXMMC79M27TYKER | Cappello |
| Cappello-N.848554 | ULRZNX84D25R0TDJ | Cappello |
| Cappello-N.867874 | UTLDLN86A27LYJZU | Cappello |
| Cappello-N.936559 | ZQJCSR70R24RYSQA | Cappello |
+-----+-----+-----+
```

27) Consultazione dati abbonamento (in media 70 volte al giorno)

27.1) Visualizzare tutte le tipologie di abbonamento

select distinct Durata, Tipo from Abbonamento;

```
mysql> Select distinct Durata, Tipo from Abbonamento;
+-----+-----+
| Durata | Tipo |
+-----+-----+
| Settimanale | Parkour |
| Mensile | Boulder |
| Annuale | Arrampicata alpina |
| Annuale | Arrampicata bambini |
| Annuale | Boulder |
| Mensile | Cardio |
| Settimanale | Arrampicata amatoriale |
| Mensile | Parkour |
| Mensile | Speed climbing |
| Mensile | Arrampicata amatoriale |
| Annuale | Arrampicata agonistica |
| Annuale | Arrampicata sportiva in falesia |
| Settimanale | Arrampicata agonistica |
| Settimanale | Yoga |
| Settimanale | Cardio |
| Mensile | Arrampicata sportiva su roccia |
| Annuale | Potenziamento muscolare |
| Annuale | Cardio |
| Settimanale | Speed climbing |
| Mensile | Pilates |
| Mensile | Arrampicata alpina |
| Annuale | Yoga |
| Settimanale | Potenziamento muscolare |
| Mensile | Potenziamento muscolare |
| Mensile | Yoga |
| Mensile | Arrampicata agonistica |
| Settimanale | Arrampicata sportiva in falesia |
| Annuale | Speed climbing |
| Mensile | Arrampicata sportiva in falesia |
| Settimanale | Arrampicata bambini |
| Annuale | Arrampicata amatoriale |
| Mensile | Arrampicata bambini |
| Annuale | Parkour |
| Settimanale | Arrampicata alpina |
| Settimanale | Pilates |
| Settimanale | Arrampicata sportiva su roccia |
| Annuale | Arrampicata sportiva su roccia |
| Annuale | Pilates |
| Settimanale | Boulder |
+-----+-----+
```

27.2) Visualizzare gli abbonamenti prossimi alla scadenza

```
select Transazione, Tipo, Durata, Ingressi, EntrateRimanenti from Abbonamento where  
EntrateRimanenti <=floor(Ingressi/10);
```

Nota: poiché le righe restituite dalla query sono circa 2000, riporteremo solo una sezione di queste righe

```
mysql> select Transazione, Tipo, Durata, Ingressi, EntrateRimanenti from Abbonamento where  
-> EntrateRimanenti <=floor(Ingressi/10);
```

Transazione	Tipo	Durata	Ingressi	EntrateRimanenti
34298	Parkour	Settimanale	5	0
34305	Parkour	Mensile	15	0
34329	Arrampicata alpina	Mensile	15	1
34329	Yoga	Mensile	15	0
34330	Arrampicata agonistica	Mensile	15	0
34342	Speed climbing	Settimanale	5	0
34352	Cardio	Settimanale	5	0
34355	Arrampicata agonistica	Settimanale	5	0
34376	Parkour	Mensile	15	1
34377	Parkour	Mensile	15	1
34387	Cardio	Settimanale	5	0
34393	Arrampicata amatoriale	Annuale	60	1
34399	Parkour	Mensile	15	1
34413	Arrampicata sportiva su roccia	Annuale	60	5
34413	Boulder	Annuale	60	6
34426	Parkour	Settimanale	5	0
34448	Parkour	Settimanale	5	0
34450	Arrampicata agonistica	Annuale	60	2
34477	Arrampicata sportiva in falesia	Settimanale	5	0
34491	Parkour	Annuale	60	4
34497	Arrampicata alpina	Mensile	15	1
34500	Pilates	Annuale	60	1

27.3) Visualizzare tutte le informazioni di un particolare abbonamento

Nota: un abbonamento specifico è identificato univocamente dalla sua corrispondente Transazione Abbonamento e dal suo tipo (di corso), quindi per questa query bisogna fornire almeno queste due informazioni

```
select Abbonamento.*, Iscritto.Nome  
  
from Abbonamento, Iscritto, TransazioneAbb  
  
where TransazioneAbb.Iscritto=Iscritto.CF and  
  
TransazioneAbb.ID=<codice_transazione> and  
  
Abbonamento.Transazione=TransazioneAbb.ID and  
  
Abbonamento.Tipo="<nome_categoria>";
```

Esempio: Visualizzare le informazioni dell'abbonamento corrispondente al codice transazione 54415 e alla tipologia "Arrampicata amatoriale"

```
mysql> select Abbonamento.*, Iscritto.Nome  
-> from Abbonamento, Iscritto, TransazioneAbb  
-> where TransazioneAbb.Iscritto=Iscritto.CF and  
-> TransazioneAbb.ID=54415 and  
-> Abbonamento.Transazione=TransazioneAbb.ID and  
-> Abbonamento.Tipo="Arrampicata amatoriale";
```

Transazione	Tipo	CodSegretario	Durata	Ingressi	EntrateRimanenti	Nome
54415	Arrampicata amatoriale	VNTMFL68S23FJFZG	Annuale	60	11	NEVA

1 row in set (0.12 sec)

27.4) Visualizzare la tipologia di abbonamento più venduta

```
select Tipo,Durata from Abbonamento as abb  
group by Tipo,Durata  
having  
(select count(*) as somma  
from Abbonamento  
group by Tipo,Durata  
having Tipo=abb.Tipo and Durata=abb.Durata )=( select max(conta.somma)  
from (select count(*) as somma from Abbonamento group by Tipo,Durata) as conta);
```

```
mysql> select Tipo,Durata from Abbonamento as abb  
-> group by Tipo,Durata  
-> having  
-> (select count(*) as somma  
-> from Abbonamento  
-> group by Tipo,Durata  
-> having Tipo=abb.Tipo and Durata=abb.Durata )=( select max(conta.somma)  
-> from (select count(*) as somma from Abbonamento group by Tipo,Durata) as conta);  
+-----+-----+  
| Tipo | Durata |  
+-----+-----+  
| Arrampicata amatoriale | Mensile |  
+-----+-----+  
1 row in set (5.17 sec)
```

28) Consultazione dati assicurazione (in media 3 volte al mese)

28.1) Visualizzare la tipologia di copertura di una determinata assicurazione

```
select Massimale,Condizione,DataAssicurazione  
from Assicurazione  
where Transazione=<codice transazione>;
```

esempio:

```
mysql> select Massimale,Condizione,DataAssicurazione  
-> from Assicurazione  
-> where Transazione=68312;  
+-----+-----+-----+  
| Massimale | Condizione | DataAssicurazione |  
+-----+-----+-----+  
| 50000.00 | aaaaaaaaaa | 2024-04-23 |  
+-----+-----+-----+  
1 row in set (0.03 sec)
```

Nota: il campo condizione dovrebbe contenere il testo del contratto di assicurazione sottoscritto dall' iscritto al momento della sottoscrizione, per ovvi motivi è stato impiazzato con un segnaposto

28.2) Visualizzare l'assicurazione più venduta

```
select Massimale,Condizione  
from Assicurazione as ass
```

group by Massimale,Condizione

having (select count(*) as somma

from Assicurazione

group by Massimale,Condizione

**having Massimale=ass.Massimale and Condizione=ass.Condizione)=(select
max(conta.somma)**

**from (select count(*) as somma From Assicurazione group by Massimale, Condizione) as
conta)**

```
mysql> select Massimale,Condizione
-> from Assicurazione as ass
-> group by Massimale,Condizione
-> having (select count(*) as somma
-> from Assicurazione
-> group by Massimale,Condizione
-> having Massimale=ass.Massimale and Condizione=ass.Condizione)=(select max(conta.somma)
-> from (select count(*) as somma From Assicurazione group by Massimale, Condizione) as conta);
+-----+-----+
| Massimale | Condizione |
+-----+-----+
| 50000.00 | aaaaaaaaaa |
+-----+-----+
1 row in set (0.13 sec)
```

29) Consultazione dati busta paga (in media 40 volte al mese)

29.1) Visualizzare la somma delle buste paga in un anno

select sum(Transazione.Importo)

from Transazione,(select distinct Transazione from BustaPaga) as BP

where Transazione.Id=BP.Transazione;

```
mysql> select sum(Transazione.Importo)
-> from Transazione,(select distinct Transazione from BustaPaga) as BP
-> where Transazione.Id=BP.Transazione;
+-----+
| sum(Transazione.Importo) |
+-----+
| 1615974.43 |
+-----+
1 row in set (0.61 sec)
```

30) Consultazione dati prenotazione (in media 10 volte al giorno)

30.1) Visualizzare tutte le prenotazioni effettuate in un intervallo di tempo

select *

from Prenotazione

**where DataPrenotazione > "<inizio periodo di tempo>" and DataPrenotazione< "<fine
periodo di tempo>"**


```
mysql> select *
-> from Prenotazione
-> where DataPrenotazione > "2020-12-30" and DataPrenotazione < "2021-01-01";
```

Iscritto	Sessione	DataPrenotazione	OraPrenotazione
DBBMRN95D43STRAC	9128	2020-12-31	11:29:50
SLNLLX74S10SWCAS	9128	2020-12-31	15:11:24
TRLSMN94R64JKYKH	10763	2020-12-31	09:18:03
VLXLDX74A65GFHWM	9128	2020-12-31	14:46:13

4 rows in set (0.16 sec)

30.2) Visualizzare il numero di prenotazioni effettuate per le sessioni della prossima settimana

select count(Prenotazione.Codsessione)

from Prenotazione,Sessione

where Sessione.Codsessione = Prenotazione.Sessione

and Sessione.DataSessione >= curdate()

and Sessione.DataSessione<= date_add(curdate() ,interval 7 day);

Nota: dato che i dati presenti nel database si fermano all' anno 2021 nell' esempio si trova anziché la curdate() una data specifica

```
mysql> select count(Prenotazione.Sessione)
-> from Prenotazione,Sessione
-> where Sessione.Codsessione = Prenotazione.Sessione
-> and Sessione.DataSessione >= "2020-12-29"
-> and Sessione.DataSessione<= date_add("2020-12-29", interval 7 day);
```

count(Prenotazione.Sessione)
88

1 row in set (0.02 sec)

31) Consultazione dati accesso alla palestra (in media 10 volte al giorno)

31.1) Visualizzare gli accessi ad una determinata sessione

select *

from Accesso

where CodSessione=<codice_sessione>;

```
mysql> select *
-> from Accesso
-> where CodSessione=11093;
```

CF	Codsessione
BDGLSS86L83TCIHK	11093
BLLLLFN93T64GMNHT	11093
BRGLRN95E10JREXY	11093
BTTDVD88L63QOXXS	11093
BZZSYX72C47GQLII	11093
CSLJNT86H11FTCHB	11093
CSTBRN92T560TDMA	11093
FRNNRX82P75AAYUA	11093
HNGLMP94P54WPGM	11093
RSSMRC81L21DBXJC	11093
SLRCTX68C86VMZLE	11093
SPGCSR91B11CFTCF	11093
VZZRXN82M95TUSCK	11093
ZNNLRT83P15BCHCQ	11093

14 rows in set (0.09 sec)

31.2) Visualizzare gli accessi avvenuti in un intervallo temporale

```
select *  
  
from Accesso, Sessione  
  
where Sessione.DataSessione > "<inizio_intervallo>" and  
  
Sessione.DataSessione < "<fine_intervallo>" and  
  
Sessione.CodSessione=Accesso.CodSessione;
```

```
mysql> select *  
-> from Accesso, Sessione  
-> where Sessione.DataSessione > "2020-12-30" and  
-> Sessione.DataSessione < "2021-01-01" and  
-> Sessione.CodSessione=Accesso.CodSessione;
```

CF	CodSessione	CodSessione	OraInizio	OraFine	Capienza	Postirim	DataSessione
BLLVMN97T69XRQEX	6898	6898	16:00:00	18:00:00	25	17	2020-12-31
BRGMRX97P10SINRW	6898	6898	16:00:00	18:00:00	25	17	2020-12-31
FNTCLD72M67JBDNC	6898	6898	16:00:00	18:00:00	25	17	2020-12-31
GDTRCC68E10JUYN	6898	6898	16:00:00	18:00:00	25	17	2020-12-31
GLLMNL83E29DRXOC	6898	6898	16:00:00	18:00:00	25	17	2020-12-31
GRZMRN67L45LQWJH	6898	6898	16:00:00	18:00:00	25	17	2020-12-31
GVXMMX78C34WUGXV	6898	6898	16:00:00	18:00:00	25	17	2020-12-31
JHJDNT81T37JWIZX	6898	6898	16:00:00	18:00:00	25	17	2020-12-31
JNXLDX80L50NXYX	6898	6898	16:00:00	18:00:00	25	17	2020-12-31
LGRMMN90L12IBIMM	6898	6898	16:00:00	18:00:00	25	17	2020-12-31
LNXXGL98A40HGMQQ	6898	6898	16:00:00	18:00:00	25	17	2020-12-31
PLXTMM76A64BHGKI	6898	6898	16:00:00	18:00:00	25	17	2020-12-31
RCCLVX98L46TBDBE	6898	6898	16:00:00	18:00:00	25	17	2020-12-31
SCLDDX91R810JFXV	6898	6898	16:00:00	18:00:00	25	17	2020-12-31
STRMNL96B14URXOZ	6898	6898	16:00:00	18:00:00	25	17	2020-12-31
TSXSLV94E21LBWEH	6898	6898	16:00:00	18:00:00	25	17	2020-12-31
VRNLRX95H15ULXCW	6898	6898	16:00:00	18:00:00	25	17	2020-12-31
WSXVLM78A70WFOCV	6898	6898	16:00:00	18:00:00	25	17	2020-12-31
ZNNSRG88R65TLYJB	6898	6898	16:00:00	18:00:00	25	17	2020-12-31

19 rows in set (1.03 sec)

32) Consultazione dati bilancio (in media 3 volte l'anno)

32.1) Visualizzare il bilancio di un determinato anno

```
select * from Bilancio where Anno=<Anno_del_bilancio>;
```

Esempio: visualizzare il bilancio relativo all'anno 2016

```
mysql> select * from Bilancio where Anno=2016;
```

Anno	CodSegretario	CapitaleSociale	LiqImm	LiqDiff	Immobilizzazioni
2016	THRUTX80R26NGSUT	30000.00	989209.68	331313.02	40000.00

32.2) Visualizzare la differenza tra liquidità immediata e liquidità differita di un determinato anno

```
select Bilancio.Anno, sum(Bilancio.LiqImm-Bilancio.LiqDiff) from Bilancio  
  
where Anno=<Anno_del_bilancio>;
```

Esempio: visualizzare la differenza tra liquidità immediata e liquidità differita dell'anno 2016

```
mysql> select Bilancio.Anno, sum(Bilancio.LiqImm-Bilancio.LiqDiff) from Bilancio
-> where Anno=2016;
+-----+-----+
| Anno | sum(Bilancio.LiqImm-Bilancio.LiqDiff) |
+-----+-----+
| 2016 | 657896.66 |
+-----+-----+
```

33) Consultazione dati Transazione (in media 2 volte a settimana)

33.1) Visualizzare le transazioni effettuate in un determinato periodo

select * from Transazione where Transazione.DataTransazione>="<Data_inizio>" and

Transazione.DataTransazione<="<Data_fine>"

order by Transazione.DataTransazione, Transazione.OraTransazione;

Esempio: visualizzare le transazioni effettuate dal 10/01/2020 al 11/01/2020

Nota: poiché le righe restituite dalla query sono elevate, per questioni di spazio riporteremo solo una parte del risultato ottenuto

```
mysql> select * from Transazione where Transazione.DataTransazione>="2020-01-10" and
-> Transazione.DataTransazione<="2020-01-11"
-> order by Transazione.DataTransazione, Transazione.OraTransazione;
+-----+-----+-----+-----+
| ID    | Importo | DataTransazione | OraTransazione |
+-----+-----+-----+-----+
| 56340 | 20.00 | 2020-01-10 | 10:28:38 |
| 95661 | 40.09 | 2020-01-10 | 10:33:43 |
| 80851 | 39.28 | 2020-01-10 | 11:35:18 |
| 113174 | 29.30 | 2020-01-10 | 11:58:03 |
| 67812 | 15.00 | 2020-01-10 | 12:07:06 |
| 128807 | 29.07 | 2020-01-10 | 12:15:43 |
| 96187 | 48.38 | 2020-01-10 | 12:17:47 |
| 93151 | 38.57 | 2020-01-10 | 12:28:51 |
| 120806 | 38.52 | 2020-01-10 | 12:32:21 |
| 72334 | 48.02 | 2020-01-10 | 12:43:24 |
| 139934 | 47.95 | 2020-01-10 | 12:49:33 |
| 124529 | 47.88 | 2020-01-10 | 12:53:48 |
| 113020 | 28.40 | 2020-01-10 | 13:05:01 |
| 111321 | 37.87 | 2020-01-10 | 13:20:14 |
| 128836 | 46.99 | 2020-01-10 | 14:01:09 |
| 129807 | 37.18 | 2020-01-10 | 14:10:36 |
| 93393 | 46.68 | 2020-01-10 | 14:24:36 |
| 112497 | 36.92 | 2020-01-10 | 14:31:54 |
| 35419 | 50.00 | 2020-01-10 | 14:58:02 |
| 43027 | 20.00 | 2020-01-10 | 15:05:22 |
| 141190 | 45.95 | 2020-01-10 | 15:18:48 |
| 105056 | 26.57 | 2020-01-10 | 15:22:07 |
| 54970 | 50.00 | 2020-01-10 | 15:45:50 |
| 93460 | 26.05 | 2020-01-10 | 15:59:52 |
| 78670 | 35.38 | 2020-01-10 | 16:25:40 |
| 113270 | 25.56 | 2020-01-10 | 16:35:55 |
| 38843 | 20.00 | 2020-01-10 | 16:42:37 |
| 105136 | 34.98 | 2020-01-10 | 16:54:36 |
| 90222 | 44.24 | 2020-01-10 | 17:24:30 |
| 89122 | 43.88 | 2020-01-10 | 17:52:04 |
| 41010 | 450.00 | 2020-01-10 | 18:05:57 |
| 96386 | 43.59 | 2020-01-10 | 18:14:01 |
| 129903 | 33.93 | 2020-01-10 | 18:14:15 |
| 78585 | 33.81 | 2020-01-10 | 18:22:50 |
| 97535 | 42.88 | 2020-01-10 | 19:07:20 |
| 53045 | 50.00 | 2020-01-10 | 19:15:43 |
| 94415 | 23.28 | 2020-01-10 | 19:27:41 |
| 92182 | 32.70 | 2020-01-10 | 19:44:46 |
| 110105 | 42.13 | 2020-01-10 | 20:01:48 |
```

33.2) Visualizzare i dati di una determinata transazione

```
select * from Transazione where Transazione.ID=<ID_Transazione>;
```

```
mysql> select * from Transazione where Transazione.ID=96039;
+-----+-----+-----+-----+
| ID     | Importo | DataTransazione | OraTransazione |
+-----+-----+-----+-----+
| 96039  | 31.85   | 2020-01-10      | 20:47:35       |
+-----+-----+-----+-----+
```

34) Consultazione dati Transazione abbonamento (in media 5 volte a settimana)

34.1) Visualizzare gli abbonamenti effettuati da un determinato iscritto

```
select TransazioneAbb.Iscritto, TransazioneAbb.ID,
       Abbonamento.Tipo, Transazione.Importo,
       Transazione.DataTransazione, Transazione.OraTransazione
from TransazioneAbb, Abbonamento, Transazione
where TransazioneAbb.Iscritto="<CF_Iscritto>" and
       Abbonamento.Transazione=TransazioneAbb.ID and Transazione.ID=TransazioneAbb.ID
order by Transazione.DataTransazione, Transazione.OraTransazione;
```

Esempio: visualizzare gli abbonamenti effettuati dall'iscritto il cui codice fiscale è VLLTZN72E12DRXLO

```
mysql> select TransazioneAbb.Iscritto, TransazioneAbb.ID,
       --> Abbonamento.Tipo, Transazione.Importo,
       --> Transazione.DataTransazione, Transazione.OraTransazione
       --> from TransazioneAbb, Abbonamento, Transazione
       --> where TransazioneAbb.Iscritto="VLLTZN72E12DRXLO" and
       --> Abbonamento.Transazione=TransazioneAbb.ID and Transazione.ID=TransazioneAbb.ID
       --> order by Transazione.DataTransazione, Transazione.OraTransazione;
+-----+-----+-----+-----+-----+-----+
| Iscritto | ID     | Tipo | Importo | DataTransazione | OraTransazione |
+-----+-----+-----+-----+-----+-----+
| VLLTZN72E12DRXLO | 41748 | Parkour | 50.00 | 2016-02-24 | 16:12:36 |
| VLLTZN72E12DRXLO | 41748 | Arrampicata sportiva in falesia | 50.00 | 2016-02-24 | 16:12:36 |
| VLLTZN72E12DRXLO | 41748 | Arrampicata sportiva su roccia | 50.00 | 2016-02-24 | 16:12:36 |
| VLLTZN72E12DRXLO | 55890 | Cardio | 50.00 | 2016-04-11 | 21:41:50 |
| VLLTZN72E12DRXLO | 55890 | Potenziamento muscolare | 50.00 | 2016-04-11 | 21:41:50 |
| VLLTZN72E12DRXLO | 36967 | Arrampicata agonistica | 20.00 | 2016-05-16 | 12:16:14 |
| VLLTZN72E12DRXLO | 36967 | Arrampicata alpina | 20.00 | 2016-05-16 | 12:16:14 |
| VLLTZN72E12DRXLO | 36967 | Arrampicata amatoriale | 20.00 | 2016-05-16 | 12:16:14 |
| VLLTZN72E12DRXLO | 36967 | Boulder | 20.00 | 2016-05-16 | 12:16:14 |
| VLLTZN72E12DRXLO | 36207 | Arrampicata amatoriale | 450.00 | 2016-08-03 | 15:44:30 |
| VLLTZN72E12DRXLO | 51786 | Arrampicata amatoriale | 450.00 | 2017-01-30 | 16:51:25 |
| VLLTZN72E12DRXLO | 51786 | Arrampicata sportiva in falesia | 450.00 | 2017-01-30 | 16:51:25 |
| VLLTZN72E12DRXLO | 42833 | Yoga | 20.00 | 2017-02-15 | 13:30:38 |
| VLLTZN72E12DRXLO | 54768 | Arrampicata sportiva su roccia | 20.00 | 2017-03-01 | 13:46:22 |
| VLLTZN72E12DRXLO | 54768 | Speed climbing | 20.00 | 2017-03-01 | 13:46:22 |
| VLLTZN72E12DRXLO | 51186 | Arrampicata sportiva in falesia | 20.00 | 2017-04-18 | 16:03:40 |
| VLLTZN72E12DRXLO | 58304 | Boulder | 50.00 | 2017-06-05 | 14:10:07 |
| VLLTZN72E12DRXLO | 58304 | Cardio | 50.00 | 2017-06-05 | 14:10:07 |
| VLLTZN72E12DRXLO | 58304 | Speed climbing | 50.00 | 2017-06-05 | 14:10:07 |
| VLLTZN72E12DRXLO | 35159 | Arrampicata amatoriale | 50.00 | 2018-03-28 | 21:29:23 |
| VLLTZN72E12DRXLO | 54829 | Arrampicata sportiva su roccia | 450.00 | 2018-04-01 | 20:49:14 |
| VLLTZN72E12DRXLO | 34407 | Arrampicata alpina | 20.00 | 2018-05-12 | 19:34:15 |
| VLLTZN72E12DRXLO | 34952 | Potenziamento muscolare | 20.00 | 2018-08-13 | 13:14:18 |
| VLLTZN72E12DRXLO | 55064 | Arrampicata agonistica | 50.00 | 2019-01-08 | 20:26:15 |
| VLLTZN72E12DRXLO | 55064 | Arrampicata amatoriale | 50.00 | 2019-01-08 | 20:26:15 |
| VLLTZN72E12DRXLO | 55064 | Parkour | 50.00 | 2019-01-08 | 20:26:15 |
| VLLTZN72E12DRXLO | 36533 | Potenziamento muscolare | 20.00 | 2019-07-13 | 16:09:04 |
| VLLTZN72E12DRXLO | 42776 | Pilates | 450.00 | 2019-08-05 | 18:46:15 |
| VLLTZN72E12DRXLO | 56149 | Arrampicata alpina | 450.00 | 2019-11-24 | 11:02:18 |
| VLLTZN72E12DRXLO | 42884 | Arrampicata agonistica | 20.00 | 2019-12-05 | 12:08:10 |
| VLLTZN72E12DRXLO | 42884 | Arrampicata sportiva su roccia | 20.00 | 2019-12-05 | 12:08:10 |
| VLLTZN72E12DRXLO | 42006 | Potenziamento muscolare | 50.00 | 2020-02-02 | 21:16:30 |
| VLLTZN72E12DRXLO | 42006 | Yoga | 50.00 | 2020-02-02 | 21:16:30 |
| VLLTZN72E12DRXLO | 36148 | Arrampicata bambini | 20.00 | 2020-06-05 | 19:50:25 |
| VLLTZN72E12DRXLO | 36148 | Pilates | 20.00 | 2020-06-05 | 19:50:25 |
| VLLTZN72E12DRXLO | 36148 | Potenziamento muscolare | 20.00 | 2020-06-05 | 19:50:25 |
| VLLTZN72E12DRXLO | 58619 | Arrampicata bambini | 50.00 | 2020-07-23 | 17:34:14 |
| VLLTZN72E12DRXLO | 34809 | Parkour | 50.00 | 2020-09-20 | 21:11:24 |
+-----+-----+-----+-----+-----+-----+
```

35) Consultazione dati Transazione assicurazione (in media 5 volte al mese)

35.1) Visualizzare le assicurazioni effettuate da un determinato iscritto

```
select TransazioneAss.Iscritto, TransazioneAss.ID, Assicurazione.Massimale,
Assicurazione.DataAssicurazione, Transazione.Importo, Transazione.DataTransazione,
Transazione.OraTransazione
from TransazioneAss, Assicurazione, Transazione
where TransazioneAss.Iscritto="<CF_Iscritto>" and
Assicurazione.Transazione=TransazioneAss.ID and
Transazione.ID=TransazioneAss.ID
order by Transazione.DataTransazione, Transazione.OraTransazione;
```

Esempio: visualizzare le assicurazioni effettuate dall'iscritto il cui codice fiscale è ZZLRSL79T43PCTKT

```
mysql> select TransazioneAss.Iscritto, TransazioneAss.ID, Assicurazione.Massimale,
-> Assicurazione.DataAssicurazione, Transazione.Importo, Transazione.DataTransazione,
-> Transazione.OraTransazione
-> from TransazioneAss, Assicurazione, Transazione
-> where TransazioneAss.Iscritto="ZZLRSL79T43PCTKT" and
-> Assicurazione.Transazione=TransazioneAss.ID and
-> Transazione.ID=TransazioneAss.ID
-> order by Transazione.DataTransazione, Transazione.OraTransazione;
+-----+-----+-----+-----+-----+-----+-----+
| Iscritto | ID | Massimale | DataAssicurazione | Importo | DataTransazione | OraTransazione |
+-----+-----+-----+-----+-----+-----+-----+
| ZZLRSL79T43PCTKT | 68248 | 50000.00 | 2020-05-07 | 15.00 | 2016-05-07 | 19:59:49 |
| ZZLRSL79T43PCTKT | 67741 | 75000.00 | 2023-04-08 | 10.00 | 2020-04-08 | 16:26:31 |
+-----+-----+-----+-----+-----+-----+-----+
```

36) Consultazione dati Transazione acquisto (in media 4 volte a settimana)

36.1) Visualizzare i prodotti acquistati da un determinato iscritto

```
select TransazioneAcq.Iscritto, TransazioneAcq.ID, Prodotto.Nome, Prodotto.Tipologia,
EsecuzioneAcq.Quantita, Transazione.Importo, Transazione.DataTransazione,
Transazione.OraTransazione
from TransazioneAcq, Prodotto, Transazione, EsecuzioneAcq
where TransazioneAcq.Iscritto="<CF_Iscritto>" and
EsecuzioneAcq.Transazione=TransazioneAcq.ID
and Transazione.ID=TransazioneAcq.ID and Prodotto.Nome=EsecuzioneAcq.Prodotto
order by Transazione.DataTransazione, Transazione.OraTransazione;
```

Esempio: visualizzare i prodotti acquistati dall'iscritto il cui codice fiscale è VLLTZN72E12DRXLO

Nota: poiché le righe restituite dalla query sono elevate, per questioni di spazio riporteremo solo una parte del risultato ottenuto

```
mysql> select TransazioneAcq.Iscritto, TransazioneAcq.ID, Prodotto.Nome, Prodotto.Tipologia,
-> EsecuzioneAcq.Quantita, Transazione.Importo, Transazione.DataTransazione,
-> Transazione.OraTransazione
-> from TransazioneAcq, Prodotto, Transazione, EsecuzioneAcq
-> where TransazioneAcq.Iscritto="ULLTZN72E12DRXL0" and
-> EsecuzioneAcq.Transazione=TransazioneAcq.ID
-> and Transazione.ID=TransazioneAcq.ID and Prodotto.Nome=EsecuzioneAcq.Prodotto
-> order by Transazione.DataTransazione, Transazione.OraTransazione;
```

Iscritto	ID	Nome	Tipologia	Quantita	Importo	DataTransazione	OraTransazione
ULLTZN72E12DRXL0	88992	Magnesite-N.226640	Magnesite	1	31.73	2016-01-15	21:07:47
ULLTZN72E12DRXL0	108715	Picchetto-N.889987	Picchetto	4	34.60	2016-01-18	17:40:37
ULLTZN72E12DRXL0	93451	Pantalone-N.585029	Pantalone	4	41.00	2016-01-26	21:59:42
ULLTZN72E12DRXL0	90756	Pantalone-N.361434	Pantalone	4	22.34	2016-02-18	21:54:39
ULLTZN72E12DRXL0	137102	Cappello-N.110701	Cappello	1	26.94	2016-02-19	16:13:31
ULLTZN72E12DRXL0	77655	Maglietta-N.117087	Maglietta	2	26.84	2016-03-06	16:53:55
ULLTZN72E12DRXL0	112928	Pantalone-N.585029	Pantalone	2	28.72	2016-06-18	17:57:44
ULLTZN72E12DRXL0	109488	Casco-N.167036	Casco	2	43.02	2016-06-19	12:15:24
ULLTZN72E12DRXL0	90472	Maglietta-N.974360	Maglietta	2	39.81	2016-08-21	18:18:30
ULLTZN72E12DRXL0	79302	Picchetto-N.745058	Picchetto	2	32.14	2016-08-29	16:04:45
ULLTZN72E12DRXL0	77567	Corda-N.625506	Corda	2	24.58	2016-09-21	14:13:10
ULLTZN72E12DRXL0	81085	Scarpette-N.561744	Scarpette	1	38.07	2016-10-02	21:49:53
ULLTZN72E12DRXL0	121121	Imbracatura-N.529132	Imbracatura	1	32.41	2016-12-18	19:23:37
ULLTZN72E12DRXL0	80687	Carrucola-N.592253	Carrucola	2	22.43	2016-12-27	20:03:59
ULLTZN72E12DRXL0	79838	Corda-N.614989	Corda	4	29.71	2017-01-12	11:34:11
ULLTZN72E12DRXL0	140373	Carrucola-N.968735	Carrucola	1	36.71	2017-03-15	16:54:31
ULLTZN72E12DRXL0	88951	Maglietta-N.552252	Maglietta	4	39.39	2017-04-01	14:08:06
ULLTZN72E12DRXL0	73549	Moschettone-N.408853	Moschettone	2	46.61	2017-04-28	18:04:50
ULLTZN72E12DRXL0	91912	Cappello-N.110701	Cappello	1	27.11	2017-05-02	18:25:01
ULLTZN72E12DRXL0	113922	Magnesite-N.900735	Magnesite	2	44.18	2017-05-08	21:24:50
ULLTZN72E12DRXL0	110778	Cappello-N.936559	Cappello	2	44.50	2017-05-23	21:30:51
ULLTZN72E12DRXL0	106184	Cappello-N.110701	Cappello	4	26.89	2017-05-28	19:32:05
ULLTZN72E12DRXL0	124507	Pantalone-N.903765	Pantalone	2	33.03	2017-07-14	13:27:01
ULLTZN72E12DRXL0	112825	Casco-N.185330	Casco	2	21.23	2017-07-19	16:16:17
ULLTZN72E12DRXL0	93542	Moschettone-N.217964	Moschettone	4	39.91	2017-08-18	18:04:35
ULLTZN72E12DRXL0	122592	Moschettone-N.834177	Moschettone	4	33.83	2017-09-17	14:36:33
ULLTZN72E12DRXL0	72932	Pantalone-N.396425	Pantalone	4	26.48	2017-09-19	11:47:24
ULLTZN72E12DRXL0	90704	Casco-N.986966	Casco	1	29.33	2017-10-06	20:48:37
ULLTZN72E12DRXL0	130080	Picchetto-N.452998	Picchetto	2	29.75	2017-11-29	10:04:35
ULLTZN72E12DRXL0	124426	Pantalone-N.585029	Pantalone	2	38.40	2017-12-12	11:44:53
ULLTZN72E12DRXL0	92480	Moschettone-N.408853	Moschettone	3	33.41	2017-12-12	17:57:16
ULLTZN72E12DRXL0	137592	Magnesite-N.590886	Magnesite	1	24.53	2017-12-17	17:07:27
ULLTZN72E12DRXL0	72557	Magnesite-N.637974	Magnesite	2	45.87	2017-12-24	14:53:10
ULLTZN72E12DRXL0	127111	Moschettone-N.733519	Moschettone	3	35.73	2018-01-04	15:48:49
ULLTZN72E12DRXL0	139485	Moschettone-N.395843	Moschettone	1	21.24	2018-01-07	21:54:24
ULLTZN72E12DRXL0	111079	Moschettone-N.307953	Moschettone	3	33.84	2018-01-12	18:25:08

37) Visualizzazione la media delle prenotazioni per settimana effettuate da un iscritto (in media 100 volte l'anno)

```
select round(avg(Conteggio),2)
from (select count(*) as Conteggio
from Prenotazione
where Prenotazione.Iscritto="<CF_Iscritto>"
group by week(DataPrenotazione), year(DataPrenotazione)) as SubQuery;
```

Esempio: visualizzare la media delle prenotazioni per settimana effettuate

dall'iscritto il cui codice fiscale è ZZLRSL79T43PCTKT

```
mysql> select round(avg(Conteggio),2)
-> from (select count(*) as Conteggio
-> from Prenotazione
-> where Prenotazione.Iscritto="ZZLRSL79T43PCTKT"
-> group by week(DataPrenotazione), year(DataPrenotazione)) as SubQuery;
```

round(avg(Conteggio),2)
1.26

38) Visualizzazione il guadagno medio settimanale grazie ai prodotti venduti (in media una volta a settimana)

```
select round(avg(Conteggio),2)
from (select sum(Transazione.Importo) as Conteggio
      from Transazione, TransazioneAcq
      where Transazione.ID=TransazioneAcq.ID
      group by week(Transazione.DataTransazione), year(Transazione.DataTransazione))
as SubQuery;
```

```
mysql> select round(avg(Conteggio),2)
-> from (select sum(Transazione.Importo) as Conteggio
-> from Transazione, TransazioneAcq
-> where Transazione.ID=TransazioneAcq.ID
-> group by week(Transazione.DataTransazione), year(Transazione.DataTransazione))
-> as SubQuery;
+-----+
| round(avg(Conteggio),2) |
+-----+
|                5957.12 |
+-----+
```

39) Visualizzazione il guadagno medio settimanale grazie agli abbonamenti (in media una volta a settimana)

```
select round(avg(Conteggio),2)
from (select sum(Transazione.Importo) as Conteggio
      from Transazione, TransazioneAbb
      where Transazione.ID=TransazioneAbb.ID
      group by week(Transazione.DataTransazione), year(Transazione.DataTransazione))
as SubQuery;
```

```
mysql> select round(avg(Conteggio),2)
-> from (select sum(Transazione.Importo) as Conteggio
-> from Transazione, TransazioneAbb
-> where Transazione.ID=TransazioneAbb.ID
-> group by week(Transazione.DataTransazione), year(Transazione.DataTransazione))
-> as SubQuery;
+-----+
| round(avg(Conteggio),2) |
+-----+
|                13027.29 |
+-----+
```

40) Consultazione dati corso (in media due volte al mese)

40.1) Visualizzare gli istruttori abilitati ad un determinato corso

Select

Istruttore.CodIstruttore,Dipendente.Nome,Dipendente.Cognome,Istruttore.Certificazione

from Istruttore,Dipendente

where Istruttore.CodIstruttore=Dipendente.Cf and Istruttore.Certificazione="<nome_corso>";

esempio:

```
mysql> Select Istruttore.CodIstruttore,Dipendente.Nome,Dipendente.Cognome,Istruttore.Certificazione
-> from Istruttore,Dipendente
-> where Istruttore.CodIstruttore=Dipendente.Cf and Istruttore.Certificazione="Yoga";
```

CodIstruttore	Nome	Cognome	Certificazione
CRBGTN69B28FSYPC	GAETANA	ACERBI	Yoga
DVDDNL74P23REZGK	DANILA	DAVIDDI	Yoga
GRMBLD69R22NHWLR	UBALDO	GARIMBERTI	Yoga
PLMRMX75A32HGIZW	REMO	PALUMBO	Yoga

4 rows in set (0.10 sec)

40.2) Visualizzare le sessioni che presentano un determinato corso in un determinato intervallo o
istante di tempo

select Sessione.*

from Sessione,Afferenza

where Sessione.DataSessione >"2020-12-15" and DataSessione<"2021-01-03"

and Sessione.CodSessione=Afferenza.CodSessione

and Afferenza.Corso="Yoga";

esempio:

```
mysql> select Sessione.*
-> from Sessione,Afferenza
-> where Sessione.DataSessione >"2020-12-15" and DataSessione<"2021-01-03"
-> and Sessione.CodSessione=Afferenza.CodSessione
-> and Afferenza.Corso="Yoga";
```

Codsessione	OraInizio	OraFine	Capienza	Postirim	DataSessione
7103	18:30:00	20:30:00	21	3	2020-12-29
8067	16:00:00	18:00:00	27	21	2020-12-20
9054	16:00:00	18:00:00	28	4	2020-12-26
9128	10:00:00	12:00:00	24	10	2021-01-01
9201	10:00:00	12:00:00	22	3	2020-12-26
10427	16:00:00	18:00:00	30	3	2020-12-16
10989	10:00:00	12:00:00	27	22	2020-12-17

7 rows in set (0.06 sec)

41) Calcolo posti disponibili per una sessione di allenamento (in media 10 volte al giorno)

select CodSessione, Postirim from Sessione where CodSessione=<Codice_Sessione>;

Esempio: calcolo posti disponibili per la sessione con codice 11087

```
mysql> select CodSessione, Postirim from Sessione where CodSessione=11087;
+-----+-----+
| CodSessione | Postirim |
+-----+-----+
|          11087 |          21 |
+-----+-----+
```

42) Calcolo accessi rimanenti sull'abbonamento (in media 2 volte a settimana)

select Transazione, Tipo, EntrateRimanenti from Abbonamento

where Transazione=<Codice_Transazione> and Tipo="<Nome_Tipologia">;

Esempio: calcolo accessi rimanenti sull'abbonamento con codice transazione uguale a 36628 per "Arrampicata bambini"

```
mysql> select Transazione, Tipo, EntrateRimanenti from Abbonamento
-> where Transazione=36628 and Tipo="Arrampicata bambini";
+-----+-----+-----+
| Transazione | Tipo          | EntrateRimanenti |
+-----+-----+-----+
|          36628 | Arrampicata bambini |          1 |
+-----+-----+-----+
```

43) Calcolo della media degli accessi per ogni settimana in un anno (in media 3 volte l'anno)

select distinct avg(settimane.conta) as media

from (select count(Accesso.CodSessione) as conta

from Accesso,Sessione

where Sessione.CodSessione=Accesso.CodSessione and year(Sessione.DataSessione)=<Anno>

group by weekofyear(Sessione.DataSessione)

) as settimane

```
mysql> select distinct avg(settimane.conta) as media
-> from
-> (select count(Accesso.CodSessione) as conta
->   from Accesso,Sessione
->   where Sessione.CodSessione=Accesso.CodSessione and year(Sessione.DataSessione)=2020
->   group by weekofyear(Sessione.DataSessione)
-> ) as settimane;
+-----+
| media |
+-----+
| 268.6981 |
+-----+
1 row in set (0.11 sec)
```


44) Visualizzazione sessioni prenotabili (in media 20 volte al giorno)

```
select *  
from (select *  
from Sessione as sess  
where (sess.DataSessione>curdate() or (sess.DataSessione=curdate()  
and sess.OraInizio>curdate())) and sess.PostiRim >0  
) as sess ,Afferenza  
where Afferenza.CodSessione=sess.CodSessione and Afferenza.Corso="boulder"
```

esempio:

Nota: anche qui abbiamo rimpiazzato la funzione curdate() e curtime() con una data e un orario specifici per le stesse motivazioni di sopra.

```
mysql> select *  
-> from (select *  
-> from Sessione as sess  
-> where (sess.DataSessione>"2020-12-15" or (sess.DataSessione="2020-12-15"  
-> and sess.OraInizio>"10:00:30")) and sess.PostiRim >0  
-> ) as sess ,Afferenza  
-> where Afferenza.CodSessione=sess.CodSessione and Afferenza.Corso="boulder";
```

CodSessione	OraInizio	OraFine	Capienza	Postirim	DataSessione	CodSessione	Corso
4518	18:30:00	20:30:00	28	1	2020-12-27	4518	Boulder
6175	18:30:00	20:30:00	27	17	2020-12-26	6175	Boulder
6898	16:00:00	18:00:00	25	17	2020-12-31	6898	Boulder
7103	18:30:00	20:30:00	21	3	2020-12-29	7103	Boulder
7385	10:00:00	12:00:00	27	3	2020-12-22	7385	Boulder
8067	16:00:00	18:00:00	27	21	2020-12-20	8067	Boulder
9054	16:00:00	18:00:00	28	4	2020-12-26	9054	Boulder
9364	16:00:00	18:00:00	28	14	2020-12-24	9364	Boulder
9482	10:00:00	12:00:00	25	21	2020-12-23	9482	Boulder
9739	10:00:00	12:00:00	22	14	2020-12-18	9739	Boulder
9834	10:00:00	12:00:00	22	7	2020-12-16	9834	Boulder
9993	10:00:00	12:00:00	23	7	2020-12-27	9993	Boulder
10650	10:00:00	12:00:00	29	6	2020-12-30	10650	Boulder

```
13 rows in set (0.36 sec)
```

45) Visualizzazione prodotti in vendita (in media 2 volte a settimana)

Select Nome,Tipologia

From Prodotto;

```
mysql> select Nome,Tipologia
-> from Prodotto;
```

Nome	Tipologia
Cappello-N.110701	Cappello
Cappello-N.131008	Cappello
Cappello-N.334651	Cappello
Cappello-N.344895	Cappello
Cappello-N.550998	Cappello
Cappello-N.570072	Cappello
Cappello-N.690727	Cappello
Cappello-N.726950	Cappello
Cappello-N.777241	Cappello
Cappello-N.837884	Cappello
Cappello-N.848554	Cappello
Cappello-N.867874	Cappello
Cappello-N.936559	Cappello
Carrucola-N.115753	Carrucola
Carrucola-N.285498	Carrucola
Carrucola-N.349820	Carrucola
Carrucola-N.390049	Carrucola
Carrucola-N.393085	Carrucola
Carrucola-N.489669	Carrucola
Carrucola-N.546811	Carrucola
Carrucola-N.592253	Carrucola
Carrucola-N.778347	Carrucola
Carrucola-N.797690	Carrucola