RACCOLTA INFORMAZIONI

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 o noleggiamo 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 due 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.”

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 massimo due 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.”

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.

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.

GLOSSARIO DEI DETERMINI

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 quando non si rinnova l’abbonamento. | Entrate rimanenti | Atleti,  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 |
| Termine | **Descrizione** | **Sinonimi** | **Collegamenti** |
| 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 |

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 a settimana)

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

25. Consultazione dati Segretario

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

27. Consultazione dati abbonamento

28. Consultazione dati assicurazione

29. Consultazione dati busta paga

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

33. Consultazione dati Transazione

34. Consultazione dati Transazione abbonamento

35. Consultazione dati Transazione assicurazione

36. Consultazione dati Transazione acquisto

37. Visualizzazione media delle prenotazioni per settimana di un iscritto

38. Visualizzazione guadagno medio settimanale grazie ai prodotti venduti

39. Visualizzazione Guadagno medio settimanale grazie agli abbonamenti

40. Consultazione dati corso (in media due 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. Statistiche sull’affluenza (in media una volta a settimana)

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

45. Visualizzazione accessi rimanenti (in media 20 volte al giorno)

46. Visualizzazione prodotti in vendita (in media 2 volte a settimana)

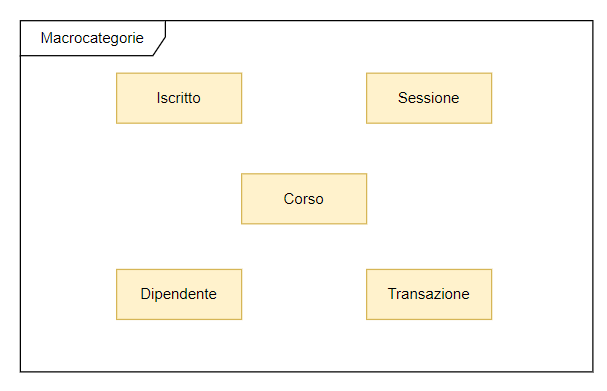
2. PROGETTAZIONE CONCETTUALE

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.

Identificazione delle entità e relazioni fondamentali

Grazie all’analisi dei requisiti siamo riusciti a identificare quattro macrocategorie principali: iscritto, 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 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.

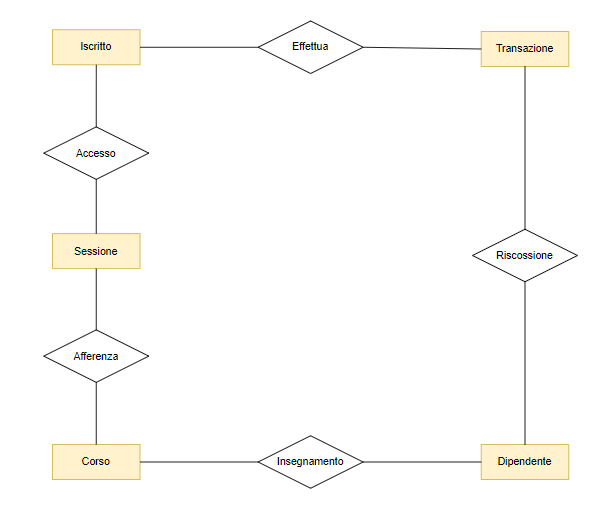
Con TRANSAZIONE indichiamo, per quanto riguarda l’iscritto, l’avvenuto pagamento di un determinato abbonamento, dell’assicurazione medica o dell’acquisto di attrezzatura sportiva. Per quanto riguarda il dipendente indicherà invece l’avvenuta consegna di una busta paga.

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.



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.

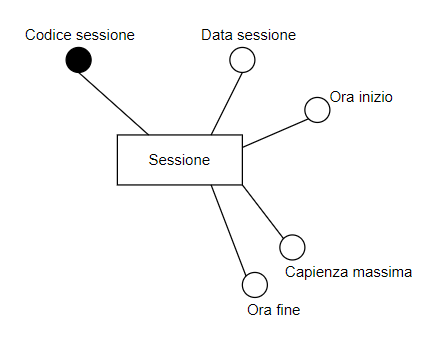
Si evince dunque come iscritto e prenotazione siano due aspetti molto legati tra di loro. Per questo motivo osserviamo che oltre ai dati caratterizzanti l’iscritto, in questa macrocategoria troveremo anche i dati caratterizzanti la prenotazione, ovvero la data di effettuazione (giorno, mese, anno).



SESSIONE

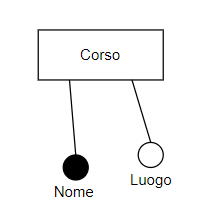
La sessione potrà essere accessibile da parte dell’iscritto previa prenotazione. La sessione presenterà al suo interno diversi corsi specifici che l’iscritto, a seconda dell’abbonamento scelto, sarà libero di seguire.

Ogni sessione dovrà presentare la data (giorno, mese, anno), l’orario di inizio e di fine allenamento.



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).

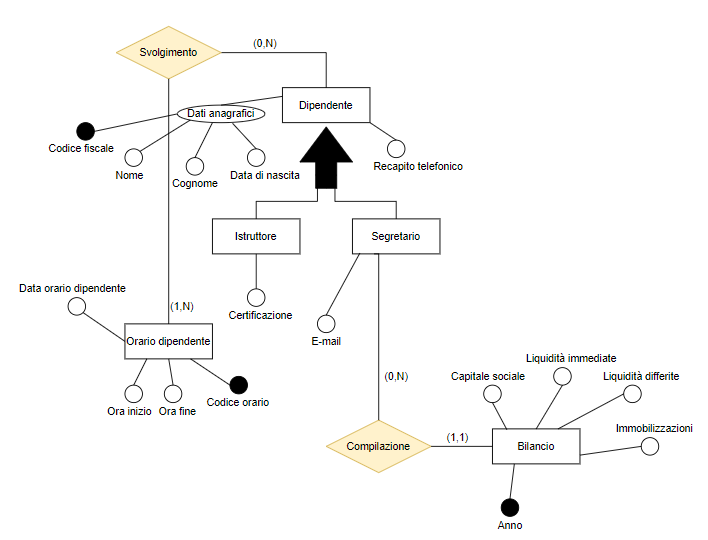


DIPENDENTE

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 (giorno, mese, anno), da un orario di inizio e di fine turno lavorativo.

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 (giorno, mese, anno), l’ora di avvenuta (ora,minuto,secondo) 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.

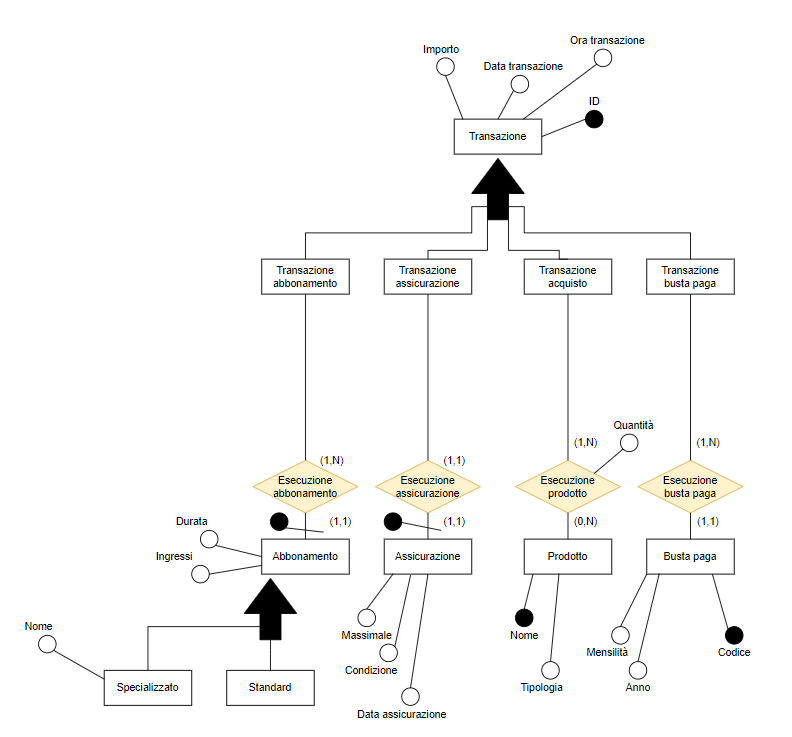
La transazione relativa all’abbonamento permette all’iscritto di acquisire un abbonamento caratterizzato da una durata e dagli ingressi che l’abbonamento mette a disposizione. Le tipologie di abbonamento sono fondamentalmente due: l’abbonamento standard e l’abbonamento specializzato. Il primo permette 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 caratterizzato da un massimale, da una condizione e da una data di scadenza (giorno, mese, anno).

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, bevande energetiche ecc.

Ogni prodotto sarà caratterizzato da un nome identificativo e dalla tipologia di appartenenza (ES: vestiario, bevande).

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.



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

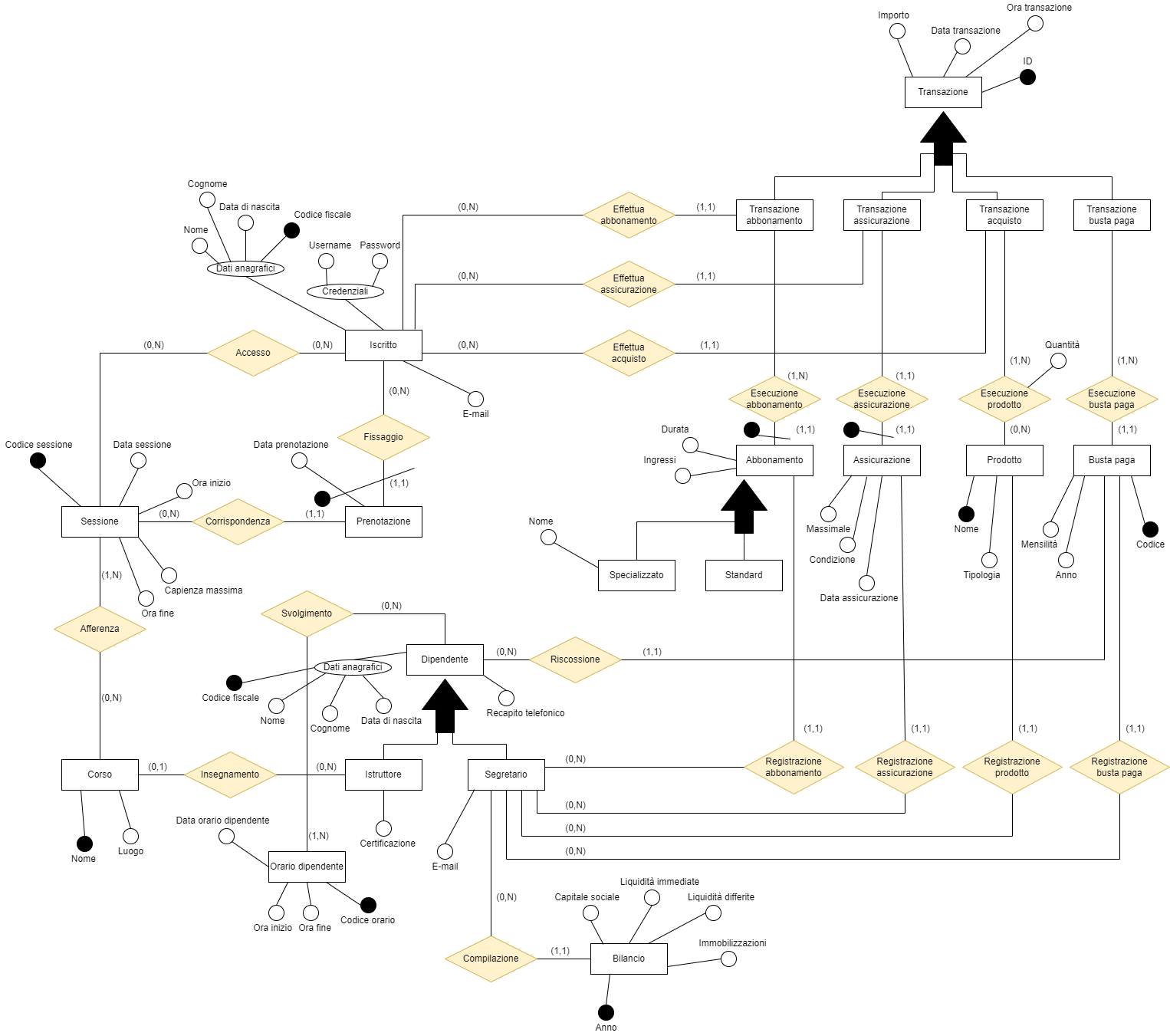
* Relazioni tra macrocategoria Transazione e macrocategoria Iscritto

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.



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.

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 e password), 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(datetime) | Data effettuazione si riferisce al giorno in cui l’ utente ha effettato 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(numerico),ora fine(numerico),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(data)e 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, Immobilizzazioni(numerico), Anno(Numerico) |  | Anno |
| Orario dipendente | Rappresentazione dei turni di lavoro settimanali a cui è soggetto un dipendente | Ora inizio(numerico), ora fine(numerico), Data orario dipendente(data), | Ora inizio e fine seguono le regole per il formato previste per gli orari | Data orario dipendente, Codice Fiscale (di Dipendente) |
| 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 unantantum 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),Tipo |
| Abbonamento standard | Abbonamento che permette di accedere al corso di arrampicata standard | “ |  | ID(transazione abbonamento) |
| 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), Condizioni(stringa), Periodo validità(data) | 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 reletivo all’ importo della retribuzione recepita per un mese di lavoro | Mensilità(numerico), Anno(numerico) | Mensilità si riferisce al mese a cui la busta paga è riferita (possono esserci anche tredicesime) | Codice(numerico) |

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(0,1)  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 che la ha registrata | Segreteria(0,N),  Busta Paga(1,1) | \*\*\* |
|  |  |  |  |

VINCOLI DI INTEGRITÀ DEI DATI

Regole di vincolo:

**RV1**: Tutti gli attributi contenenti la parola “ora” rappresentano un ora e sono numeri interi compresi tra 0 e 23

**RV2**: Tutti gli attributi compositi di tipo data sono formati da un attributo “giorno”, uno “mese”, e uno “anno”, che sono tutti e tre numeri interi, che devono rappresentare una data coerente

**RV3**: L’ attributo “recapito telefonico” dell’ entità istruttore deve rappresentare un numero di telefono valido

**RV4**: L’ attributo composito dati anagrafici deve essere composto 4 attibuti: nome(stringa), cognome(stringa), data di nascita(data), e codice fiscale(stringa), di cui data di nascita deve rappresentere una data valida (vedi RV2)

**RV5**: gli attributi “importo” nelle specializzazioni di transazione deve corrispondere a una somma di denaro espressa in euro non negativa

**RV6**: L’ attributo mensilità in busta paga deve essere un numero intero compreso tra 1 e 13 a rappresentare un mese stipendiabile o una tredicesima

**RV7**: gli attributi e-mail nelle varie entità devono essere degli indirizzi e-mail validi

RV8: L’ attributo certificazione appartenente a istruttore deve essere una stringa che spieghi l’ abilitazione all’ insegnamento dei vari tipi di corso che l’ istruttore può insegnare, e deve essere coerente con la relazione insegnamento

**RV9**: L’ attributo “ingressi” dell’ entità abbonamento deve essere un intero maggiore di 0.

**RV10**: L’ attributo “luogo” di corso deve essere una stringa che può assumere solo 2 valori: “indoor” o “outdoor”

**RV11**: L’ attributo “anno” dell’ entità busta paga deve essere un intero che rappresenta un anno in cui il dipendente ha lavorato per la società

**RV12**: 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”

**RV13**: L’ attributo “quantità” della relazione esecuzione prodotto deve essere un intero maggiore di 0.

**RV14**: L’ attributo “durata” dell’ entità abbonamento deve essere una stringa che può assumere i valori: “mensile”, “semestrale” e “annuale”.

**RV15**: L’ attributo “massimale” dell’ entità assicurazione deve rappresentare un numero di euro maggiore di 0

**RV16**: Gli attributi “Capitale sociale”, “Liquidità differite“ e “Liquidità immediate” devono rappresentare somme di denaro maggiori o uguali di € 0,00

**RV17**: L’ attributo “Anno” di bilancio si riferisce all’ anno di esercizio della società e deve quindi essere un numero

**RV18**: 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”.

PROGETTAZIONE LOGICA

TAVOLA DEI VOLUMI E DELLE OPERAZIONI

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 |

**NOTE**: 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 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.

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 | 5 volte all’anno |
| 10 | 10 volte a settimana |
| 11 | 2 volte all’anno |
| 12 | 10 volte a settimana |
| 13 | 3 volte all’anno |
| 14 | 3 volte a settimana |
| 15 | 3 volte al mese |
| 16 | 2 volte a settimana |
| 17 | 3 volte all’anno |
| 18 | 3 volte a settimana |
| 19 | 6 volte a settimana |
| 20 | 1 volta al mese |
| 21 | 3 volte all’anno |
| 22 | 2 volte al mese |
| 23 | 2 volte all’anno |
| 24 | 2 volte al mese |
| 25 | 2 volte all’anno |
| 26 | 2 volte al giorno |
| 27 | 1 volta a settimana |
| 28 | 5 volte a settimana |
| 29 | 5 volte al mese |
| 30 | 5 volte al mese |
| 31 | 10 volte al giorno |
| 32 | 10 volte al giorno |
| 33 | 2 volte al mese |
| 34 | 10 volte al giorno |
| 35 | 2 volte a settimana |
| 36 | 2 volte al mese |
| 37 | 1 volta a settimana |
| 38 | 20 volte al giorno |
| 39 | 20 volte al giorno |
| 40 | 2 volte a settimana |

**Analisi delle ridondanze**

**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 .

Con N intendiamo il numero di corsi a cui afferisce una sessione. È possibile ricavarlo dalla tavola dei volumi come .

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

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.

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

|  |  |  |  |
| --- | --- | --- | --- |
| Operazione | Costo | Frequenza(settimanale) | Totale |
| 35 |  | 2 |  |
| Costo operazioni senza ridondanza | | |  |

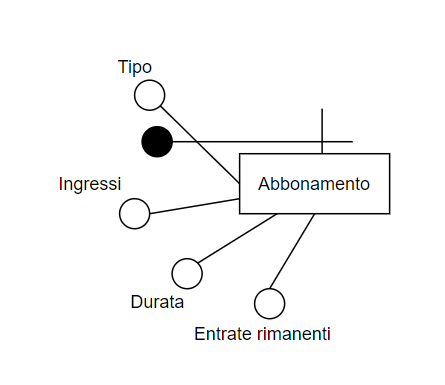
|  |  |  |  |
| --- | --- | --- | --- |
| Operazione | Costo | Frequenza(settimanale) | Totale |
| 35 | 60000 | 2 | 3,1 |
| Costo operazioni con ridondanza | | | 3,1 |

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 complicanze 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.

**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:

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

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:

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

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.

|  |  |  |  |
| --- | --- | --- | --- |
| 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.

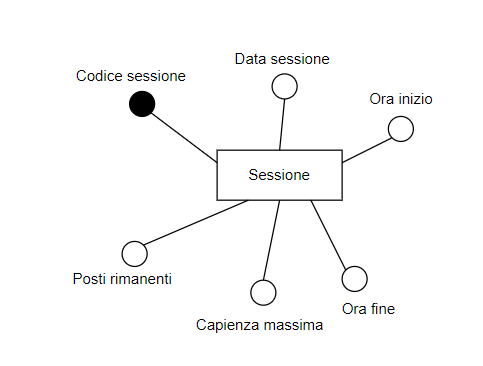
Riportiamo di seguito il costo totale relativo alle operazioni 34 e 38, tenendo conto della frequenza con la quale essa viene effetuata in un periodo di 5 anni.

|  |  |  |  |
| --- | --- | --- | --- |
| Operazione | Costo | Frequenza(giornaliera) | Totale |
| 34 | 21000 | 10 | 383250000 (3,8 |
| 38 | 72 | 20 | 2628000 (2,6 |
| 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:



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.

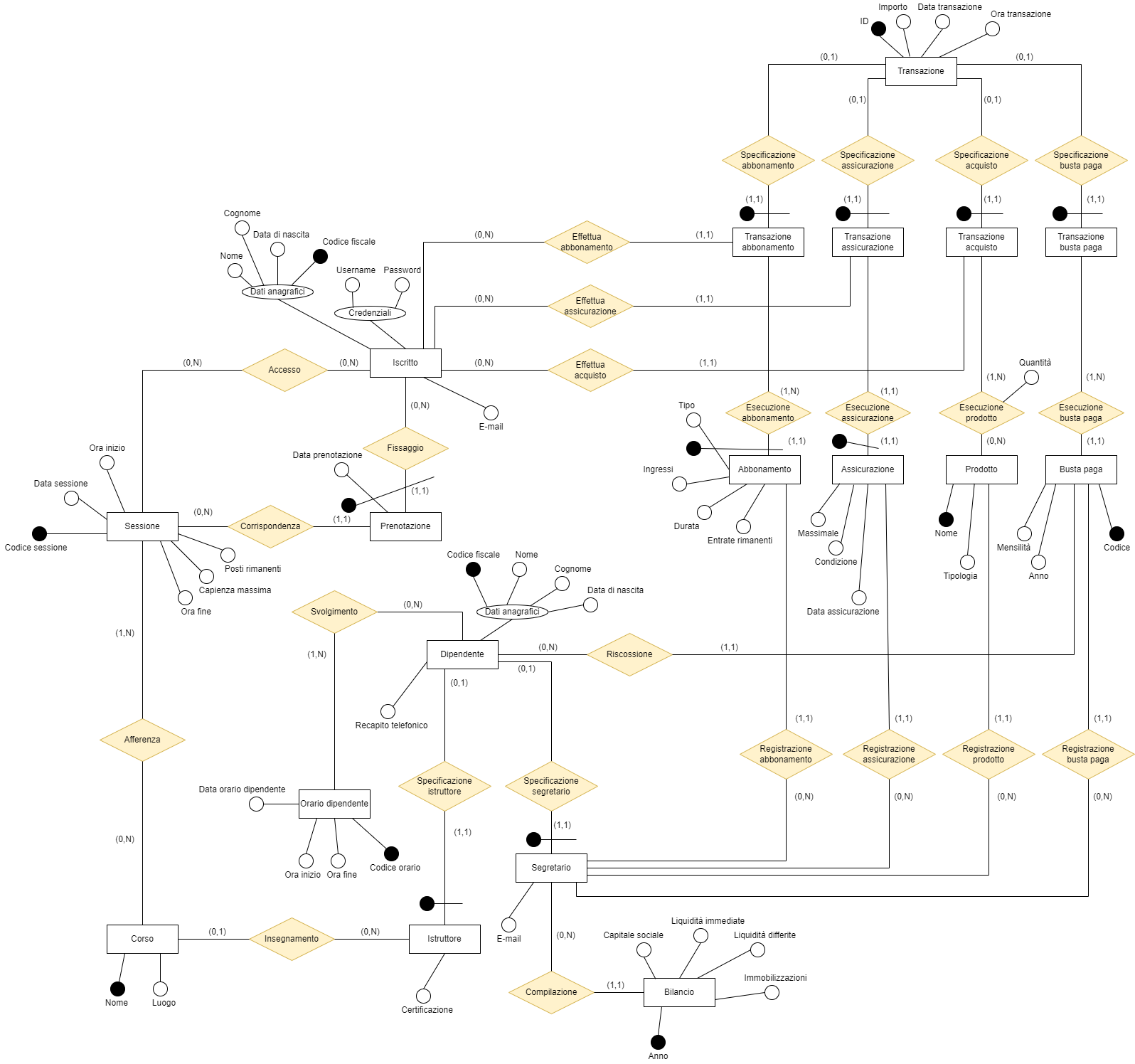
1) Per quanto riguarda l’entità Abbonamento è stato deciso di accorpare le figlie della generalizzazione nel genitore in quanto abbiamo riscontrato che gli accessi al padre e alle figlie sono contestuali. Abbiamo deciso di optare per questa soluzione in quanto gli attributi caratterizzanti le entità 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.

Riportiamo di seguito il diagramma E-R complessivo ottenuto a seguito dell’eliminazione delle gerarchie.



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.

ELIMINAZIONE DEGLI ATTRIBUTI COMPOSTI

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

1) L’attributo composto Dati anagrafici, relativo alle entità Iscritto e Dipendente

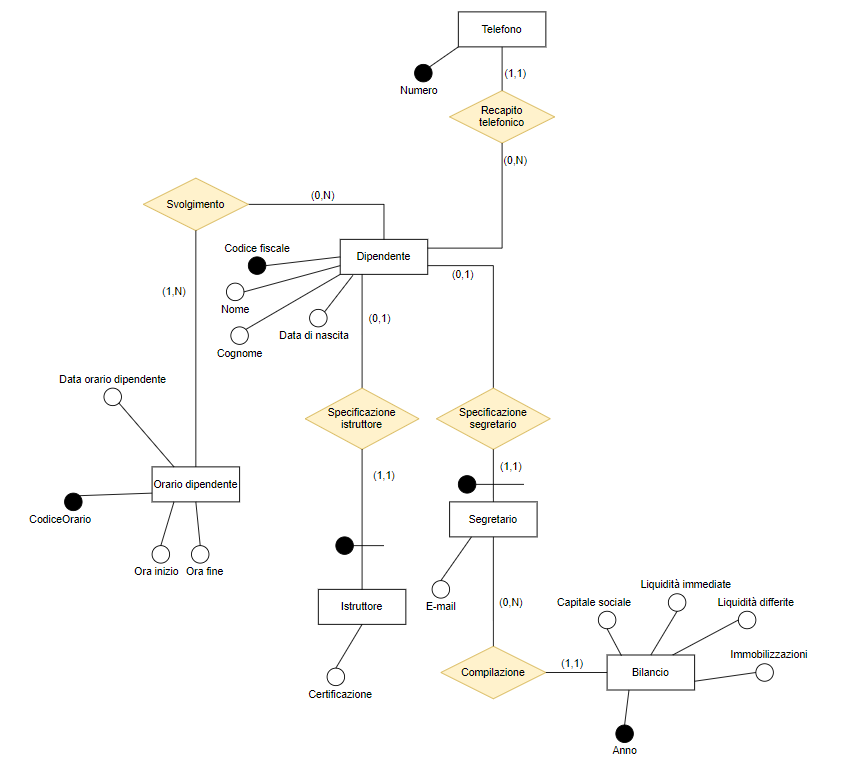
2) L’attributo composto Credenziali, relativo all’entità Iscritto

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

ELIMINAZIONE DEGLI ATTRIBUTI MULTIVALORE

L’unico attributo multivalore individuato è il Recapito telefonico relativo all’entità Dipendente.

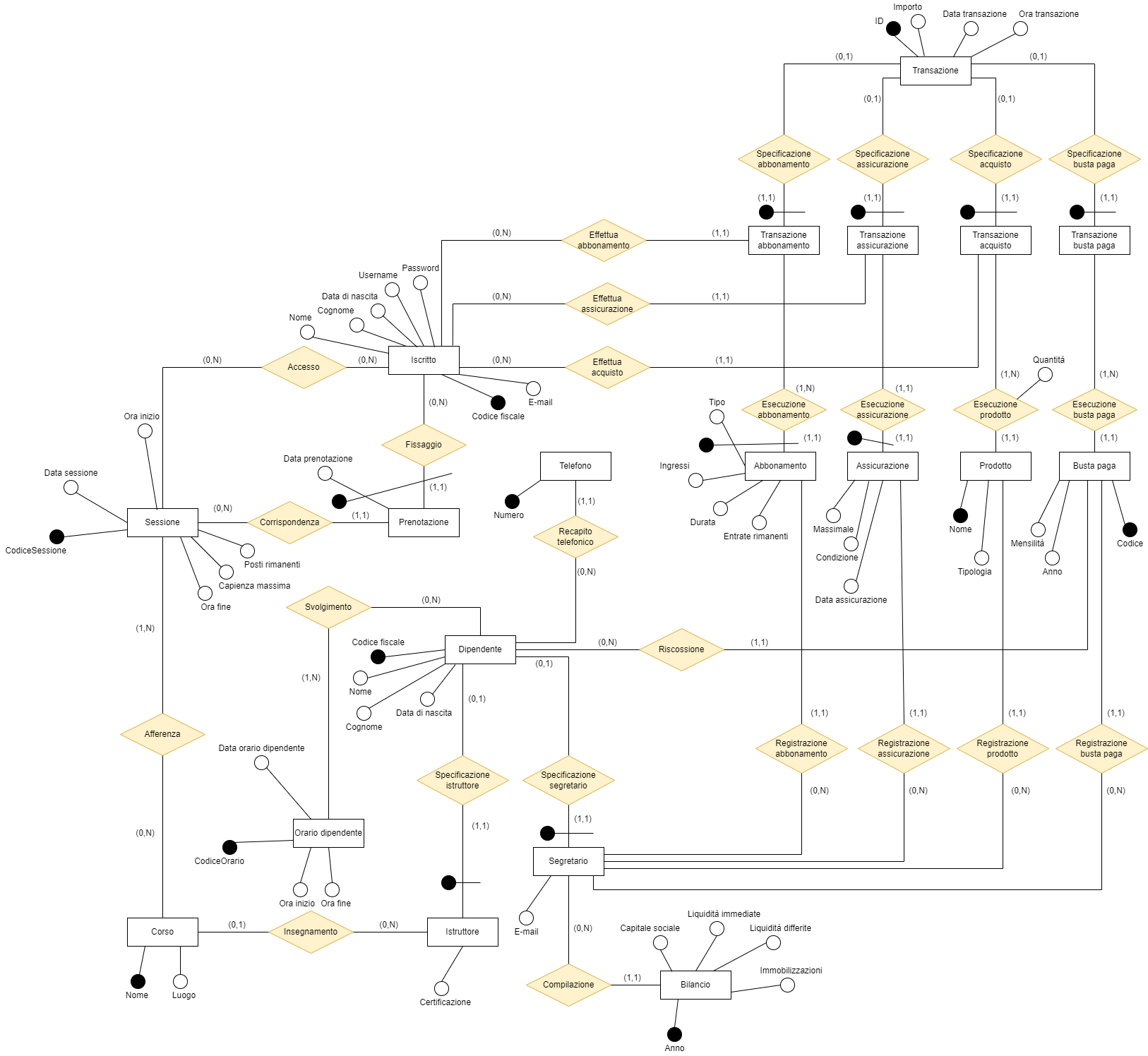
Riportiamo di seguito la ristrutturazione effettuata in merito:



ELENCO DEGLI IDENTIFICATORI PRINCIPALI

|  |  |
| --- | --- |
| ENTITÀ | IDENTIFICATORE |
| Iscritto | Codice fiscale |
| Sessione | Codice sessione |
| Prenotazione | Data prenotazione, 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 |

Riportiamo di seguito il diagramma E-R ottenuto a seguito delle varie operazioni di ristrutturazione.



NORMALIZZAZIONE

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.

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. |

Traduzione verso il modello relazionale

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

CODIFICA SQL E TESTING

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.

Immagine che contiene testo

Descrizione generata automaticamente

DEFINIZIONE DELLO SCHEMA

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;

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)

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 a settimana)

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

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 CodSessione=<Codice\_Sessione>;**

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

**delete from Corso**

**where Nome=”<Nome\_Corso>”;**

TUTTE LE MODIFICHE E LE CANCELLAZIONI

22) Consultazione dati dell’atleta/iscritto

22.1) Visualizzare gli iscritti maggiorenni

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

**from Iscritto**

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

Esempio:



22.2) Visualizzare i dati di uno specifico iscritto

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

Esempio:

Immagine che contiene testo, elettronico

Descrizione generata automaticamente

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**

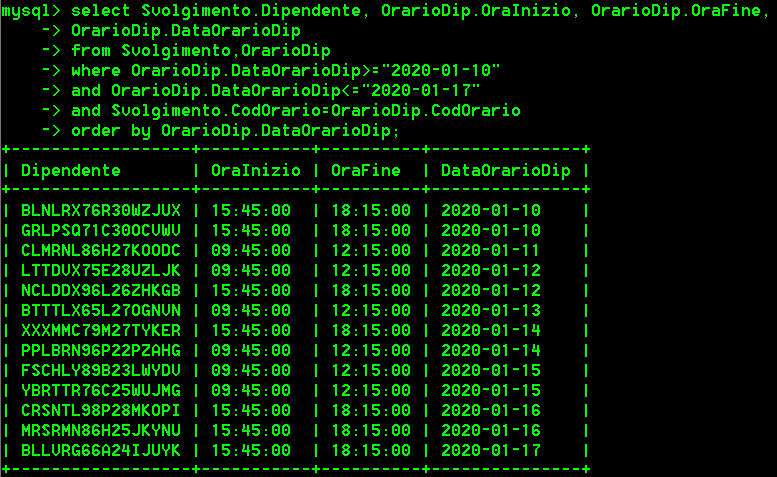
**from Svolgimento,OrarioDip**

**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

23.2) Visualizzare i numeri di telefono di un determinato dipendente

**select \* from Telefono where Telefono.CF=”<CF\_Dipendente>”;**

Immagine che contiene testo

Descrizione generata automaticamenteEsempio: visualizzare i numeri di telefono del dipendente il cui codice fiscale è CSSNTL92T26LJKZR

24) Consultazione dati istruttore

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

Immagine che contiene testo

Descrizione generata automaticamente

25) Consultazione dati Segretario

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**

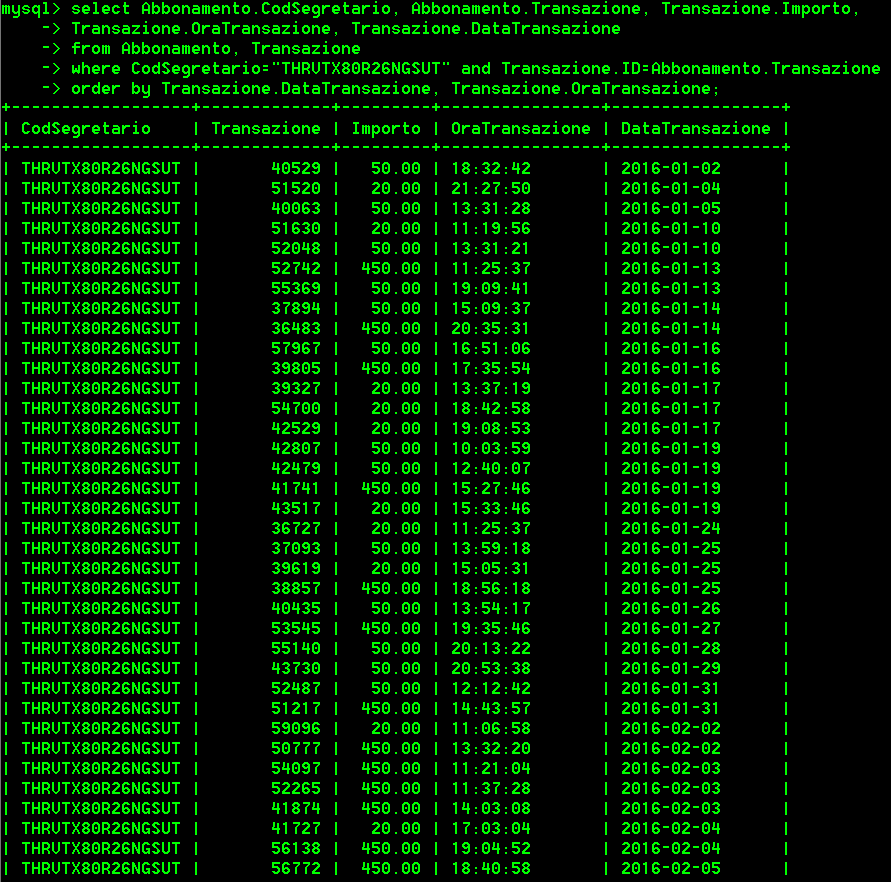
**Transazione.ID=Abbonamento.Transazione**

**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



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

Immagine che contiene testo

Descrizione generata automaticamente

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);**

**Immagine che contiene testo

Descrizione generata automaticamente**

26.3) Visualizzare tutti dati dei prodotto appartenenti ad una categoria

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

Immagine che contiene testo

Descrizione generata automaticamenteEsempio: visualizzare tutti i prodotti appartenenti alla categoria Cappello

27) Consultazione dati abbonamento

27.1) Visualizzare tutte le tipologie di abbonamento

**select distinct Durata, Tipo from Abbonamento;**

****

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

****

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”

Immagine che contiene testo

Descrizione generata automaticamente

27.4) Visualizzare la tipolgia 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);**

esempio:

Immagine che contiene testo

Descrizione generata automaticamente

28) Consultazione dati assicurazione

28.1) Visualizzare la tipologia di copertura di una determinata assicurazione

**select Massimale,Condizione,DataAssicurazione**

**from Assicurazione**

**where Transazione=<codice transazione>;**

esempio:

Immagine che contiene testo

Descrizione generata automaticamente

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)**

esempio:

Immagine che contiene testo

Descrizione generata automaticamente

29) Consultazione dati busta paga

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;**

esempio:

Immagine che contiene testo

Descrizione generata automaticamente

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

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

**select \***

**from Prenotazione**

**where DataPrenotazione > “<inizio periodo di tempo>” and DataPrenotazione< “<fine periodo di tempo>”**

esempio:

Immagine che contiene testo

Descrizione generata automaticamente

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

Immagine che contiene testo

Descrizione generata automaticamente

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

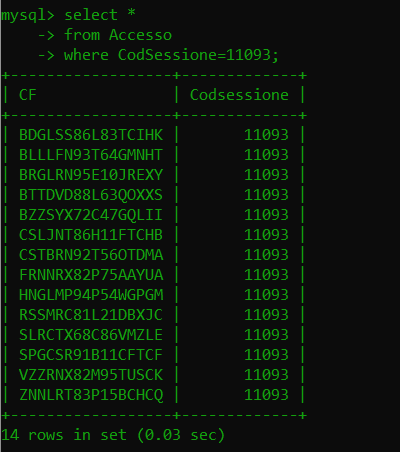
31.1) Visualizzare gli accessi ad una determinata sessione

**select \***

**from Accesso**

**where CodSessione=<codice\_sessione>;**

esempio:



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;**

esempio:

Immagine che contiene testo, elettronico

Descrizione generata automaticamente

32) Consultazione dati bilancio

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

Immagine che contiene testo

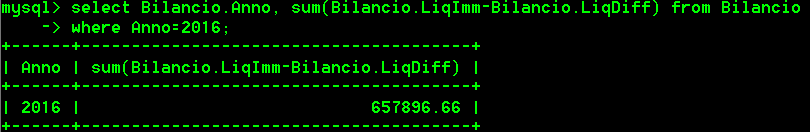
Descrizione generata automaticamente

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



33) Consultazione dati Transazione

33.1) Visualizzare le transazioni effettuate in un determinato periodo

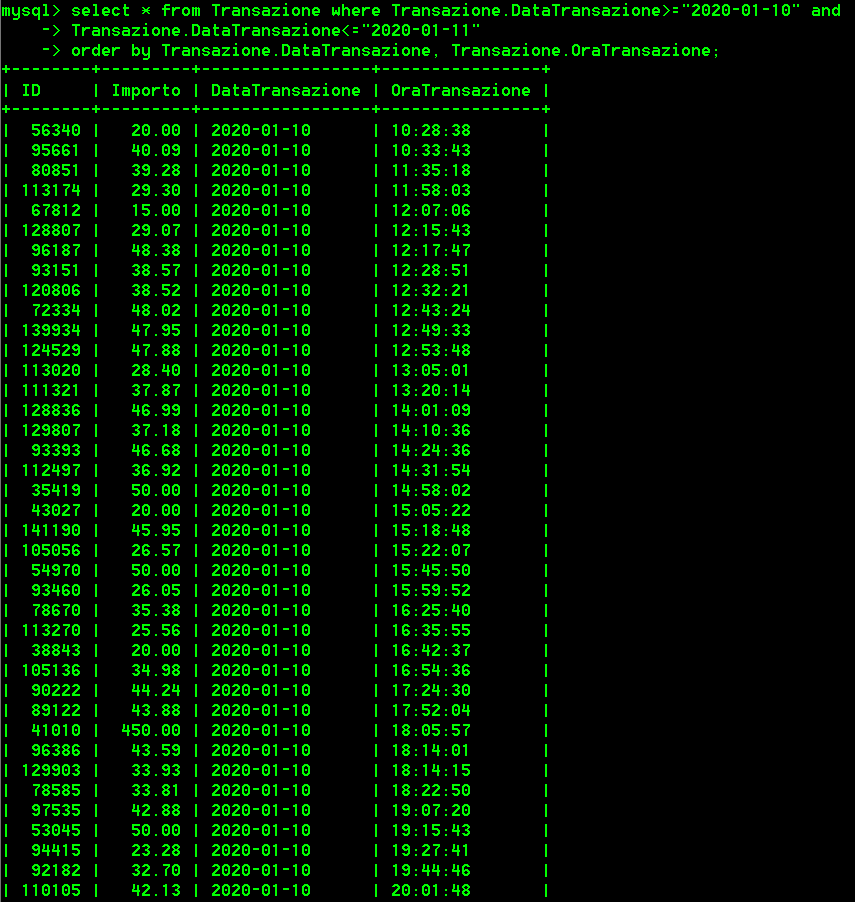
**select \* from Transazione where Transazione.DataTransazione>=”<Data\_inizio>” and**

**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



33.2) Visualizzare i dati di una determinata transazione

**select \* from Transazione where Transazione.ID=<ID\_Transazione>;**

**Immagine che contiene testo

Descrizione generata automaticamente**

34) Consultazione dati Transazione abbonamento

34.1) Visualizzare gli abbonamenti effettuati da un determinato iscritto

**select TransazioneAbb.Iscritto, TransazioneAbb.ID,**

**Abbonamento.Tipo, Transazione.Importo,**

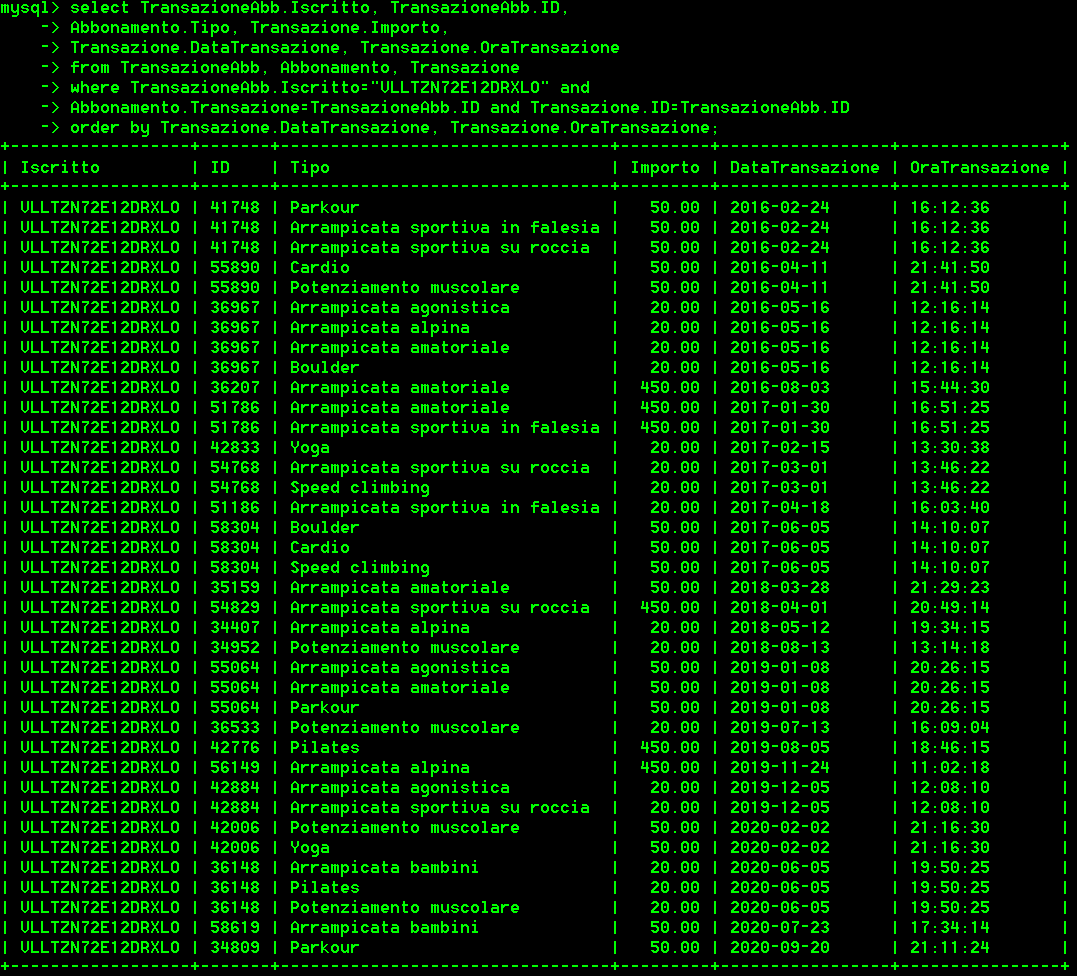
**Transazione.DataTransazione, Transazione.OraTransazione**

**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



35) Consultazione dati Transazione assicurazione

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

**Immagine che contiene testo

Descrizione generata automaticamente**

36) Consultazione dati Transazione acquisto

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**

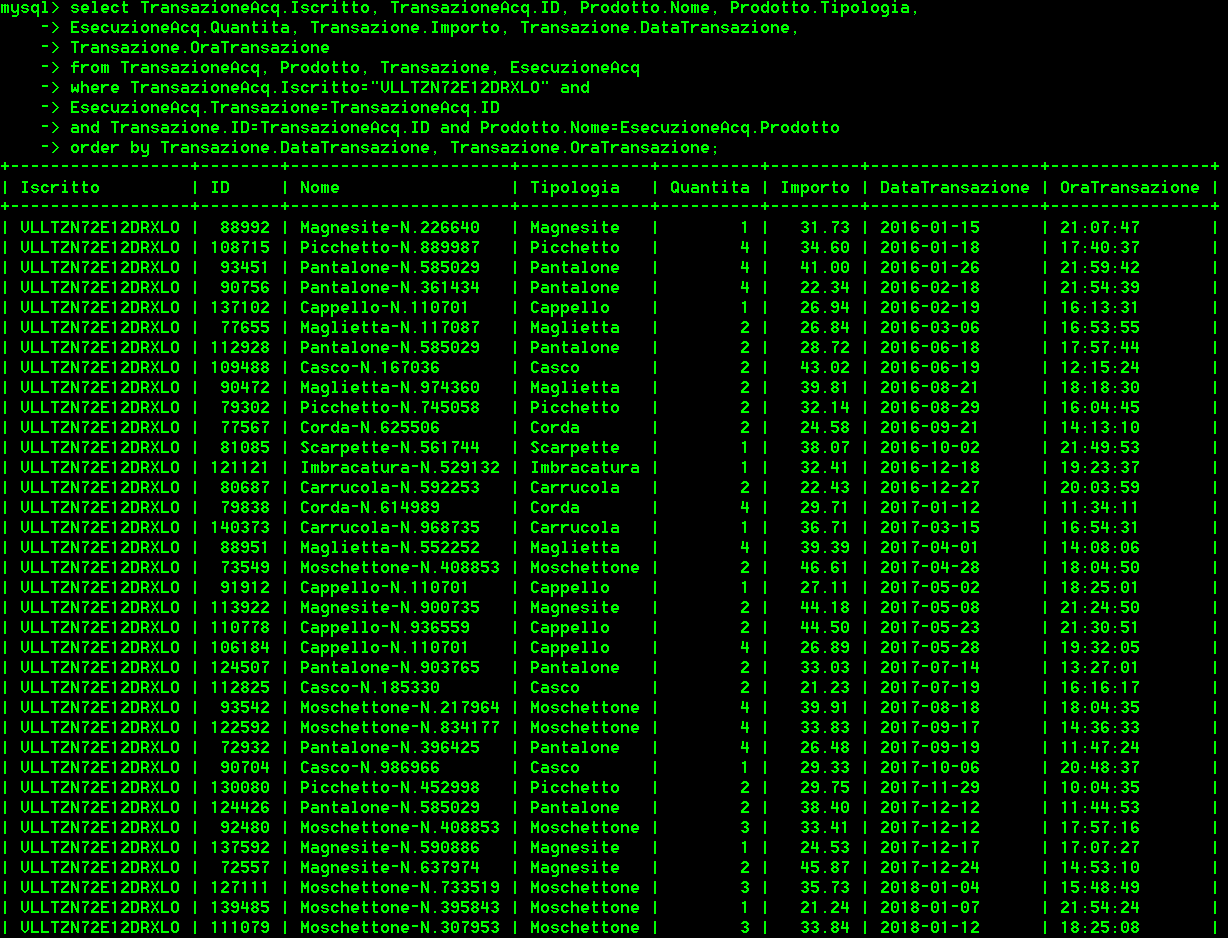
**EsecuzioneAcq.Transazione=TransazioneAcq.ID**

**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

****

37) Visualizzazione la media delle prenotazioni per settimana effettuate da un iscritto

**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

**Immagine che contiene testo

Descrizione generata automaticamente**

38) Visualizzazione il guadagno medio settimanale grazie ai prodotti venduti

**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;**

**Immagine che contiene testo

Descrizione generata automaticamente**

39) Visualizzazione il guadagno medio settimanale grazie agli abbonamenti

**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;**

Immagine che contiene testo

Descrizione generata automaticamente

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

40.1) Visualizzare gli istruttori abilitati ad un determinato corso

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

istante di tempo

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

Immagine che contiene testo, orologio

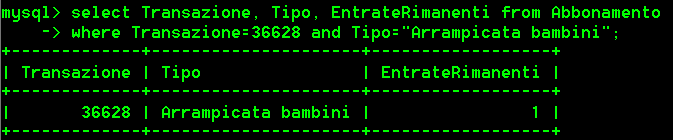
Descrizione generata automaticamente

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”



43) Statistiche sull’affluenza (in media una volta a settimana)

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

45) Visualizzazione accessi rimanenti (in media 20 volte al giorno)

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