

WaveRock

Alessandro Lombardini

Ingegneria e scienze informatiche, LM

Matr. 0000841177

11/07/2019

Sommario

Introduzione.....	2
Analisi dei requisiti.....	3
Intervista.....	3
Rilevamento delle ambiguità e correzioni proposte.....	4
Definizione delle specifiche in linguaggio naturale.....	5
Progettazione concettuale.....	6
Schema scheletro.....	6
Raffinamenti proposti.....	7
Schema concettuale finale.....	9
Progettazione logica.....	10
Stima del volume dei dati.....	10
Descrizione delle operazioni principali e stima della frequenza.....	11
Schemi di navigazione e tabelle degli accessi.....	12
Raffinamento dello schema.....	18
Analisi delle ridondanze.....	19
Traduzione di entità e associazioni in relazioni.....	20
Schema relazionale finale.....	22
Traduzione delle operazioni in query SQL.....	23
Progettazione dell'applicazione.....	30
Descrizione dell'architettura dell'applicazione realizzata.....	30

Introduzione

Il progetto consiste nella realizzazione di un applicativo in grado di informatizzare parte dell'ambito gestionale di una palestra. L'attività è specializzata in arrampicata, da qui il nome WaveRock, l'omonima formazione rocciosa conosciuta in tutto il mondo per la sua forma simile ad un'onda. Per raggiungere l'obiettivo è richiesto l'uso di una base di dati, un insieme di dati strutturati ed interrogabili che consentirà una corretta manipolazione delle informazioni. Dovranno essere gestite le principali funzioni che un gestionale per una palestra generalmente offre: gestione degli accessi, degli acquisti e dei corsi. Oltre alle operazioni più basilari l'applicativo dovrà essere in grado di esporre informazioni statistiche al fine di aiutare il personale a comprendere lo stato e l'andamento dell'attività.

Analisi dei requisiti

Intervista

Il proprietario della palestra WaveRock richiede un sistema informativo per la gestione dei clienti, dei corsi e degli ingressi. È richiesto che vengano memorizzati i dati dei clienti e degli istruttori, ovvero nome, cognome, indirizzo, data di nascita e recapito telefonico. Gli istruttori hanno libero accesso alla palestra, a qualsiasi ora del giorno. I clienti possono accedervi acquistando un ingresso. Sono disponibili biglietti, per un singolo accesso, oppure abbonamenti, per accedere senza alcuna limitazione per un periodo di tempo limitato. In fase di sottoscrizione di un abbonamento è richiesta la data di attivazione. La data di attivazione può essere diversa dalla data di acquisto. Non è possibile stabilire come data di attivazione un giorno già coperto da un abbonamento valido; gli abbonamenti sono sovrapponibili, nel caso in cui un abbonamento vada a coincidere con un altro non sono previste proroghe di alcun genere. Gli abbonamenti acquistabili variano in funzione della durata. La durata è in mesi; un mese equivale a trenta giorni. Attualmente sono previsti abbonamenti mensili, trimestrali e annuali. Sono inoltre disponibili delle tessere per l'acquisto di più ingressi a prezzo scontato. Terminati gli accessi la tessera perde di utilità. Esistono diverse tipologie di tessera che differiscono per il numero di accessi consentiti. Un cliente può acquistare biglietti e tessere senza che questi perdano di validità nel tempo. Un qualsiasi tipo di ingresso, una volta utilizzato, ha validità per tutto il corso della giornata. Gli ingressi hanno prezzi diversi in base età del cliente. Sono presenti tre fasce di età: under 14, under 18, e adulti. Non sono ammessi bambini di età inferiore a sei. Le fasce di età garantiscono l'accesso alla palestra in diverse fasce orarie, riassumibili in: mattina, pomeriggio e sera. Attualmente i minorenni possono accedere solo il pomeriggio mentre per gli adulti non sono previste restrizioni.

La palestra offre ai suoi clienti corsi di diversa natura. I clienti possono iscriversi pagando la quota richiesta. Sono presenti tipologie di corso ricorrenti che i proprietari ripropongono, ad intervalli di tempo irregolari, in più edizioni. Non vengono mai attivate due edizioni dello stesso corso nella stessa data. Un'edizione viene seguita da un solo istruttore. Un istruttore può seguire più corsi e più edizioni, anche nello stesso periodo. Un istruttore, durante il corso di una giornata, può effettuare una sola lezione. Una tipologia non ha alcuna implicazione sul numero massimo di iscritti o sulle lezioni che si svolgeranno. Le uniche

cose che non variano nelle diverse edizioni sono il prezzo e l'obiettivo del corso. Quando un corso viene attivato viene specificata una data di inizio che pone un limite temporale alle iscrizioni. Le lezioni previste da quella edizione devono essere fissate entro quella data, non oltre. Deve essere specificato il luogo, la data e l'ora. Un' edizione prevede al massimo una lezione al giorno. È disponibile una collezione di luoghi in cui è possibile svolgere l'attività di arrampicata. I luoghi possono essere outdoor oppure indoor. Tutte le lezioni devono avvenire in uno di questi luoghi. Il luogo può variare ad ogni lezione. In un luogo possono avvenire anche più lezioni contemporaneamente. Una lezione tenuta all'interno della palestra garantisce l'accesso durante l'arco della giornata a tutti i partecipanti. Viene richiesto uno storico di tutte le edizioni, comprese le lezioni e le iscrizioni ad esse associate, al fine di fornire dei dati statistici sulle partecipazioni alle diverse tipologie di corso

La figura dell'istruttore e del cliente possono coincidere. Non esistono divieti per cui un cliente non possa diventare un istruttore oppure un istruttore non possa diventare cliente. Un istruttore può partecipare come cliente alle attività e può acquistare ingressi per un uso futuro. È necessario mantenere traccia degli accessi dei clienti e degli istruttori alla palestra. Durante l'arco di una giornata una persona può effettuare più di un accesso. L'applicativo è destinato esclusivamente al personale della palestra. Si ha necessità che possa essere utilizzato senza alcun tipo di autenticazione.

Rilevamento delle ambiguità e correzioni proposte

La descrizione risulta chiara e completa, l'interlocutore aveva una idea chiara di quello che voleva. Occorre però fare una precisazione:

- Per 'luogo' ci si riferisce ad una zona in cui sono presenti una o più pareti di arrampicata. Ciascun luogo presenta una locazione geografica, un nome e un numero di percorsi disponibili. Spesso è inoltre disponibile una descrizione inerente alla difficoltà delle pareti e sui periodi migliori per recarvisi.

Definizione delle specifiche in linguaggio naturale ed estrazione dei concetti principali

Identifichiamo, a partire dall'intervista, i concetti principali in gioco.

Il proprietario della palestra WaveRock richiede un sistema informativo per la gestione dei clienti, dei corsi e degli ingressi. È richiesto che vengano memorizzati i dati dei **clienti** e degli **istruttori**, ovvero nome, cognome, indirizzo, data di nascita e recapito telefonico. Gli istruttori hanno libero accesso alla palestra, a qualsiasi ora del giorno. I clienti possono accedervi acquistando un **ingresso**. Sono disponibili **biglietti**, per un singolo accesso, oppure **abbonamenti**, per accedere senza alcuna limitazione per un periodo di tempo limitato. In fase di sottoscrizione di un abbonamento è richiesta la data di attivazione. La data di attivazione può essere diversa dalla data di acquisto. Non è possibile stabilire come data di attivazione un giorno già coperto da un abbonamento valido; gli abbonamenti sono sovrapponibili, nel caso in cui un abbonamento vada a coincidere con un altro abbonamento non sono previste proroghe di alcun genere. Gli abbonamenti acquistabili variano in funzione della durata. La durata è in mesi; un mese equivale a trenta giorni. Attualmente sono previsti abbonamenti mensili, trimestrali e annuali. Sono inoltre disponibili delle **tessere** per l'acquisto di più ingressi a prezzo scontato. Terminati gli accessi la tessera perde di utilità. Esistono diverse **tipologie di tessera** che differiscono per il numero di accessi consentiti. Un cliente può acquistare biglietti e tessere senza che questi perdano di validità nel tempo. Un qualsiasi tipo di ingresso, una volta utilizzato, ha validità per tutto il corso della giornata. Gli ingressi hanno prezzi diversi in base età del cliente. Sono presenti tre **fasce di età**: under 14, under 18, e adulti. Non sono ammessi bambini di età inferiore a sei. Le fascia di età garantiscono l'accesso alla palestra in diverse **fasce orarie**, riassumibili in: mattina, pomeriggio e sera. Attualmente i minorenni possono accedere solo il pomeriggio mentre per gli adulti non sono previste restrizioni.

La palestra offre ai suoi clienti **corsi** di diversa natura. I clienti possono isciversi pagando la quota richiesta. Sono presenti **tipologie di corso** ricorrenti che i proprietari ripropongono, ad intervalli di tempo irregolari, in più **edizioni**. Non vengono mai attivate due edizioni dello stesso corso nella stessa data. Una edizione viene seguita da un solo istruttore. Un istruttore può seguire più corsi e più edizioni, anche nello stesso periodo. Un istruttore, durante il corso di una giornata, può effettuare una sola lezione. Una tipologia non ha alcuna implicazione sul numero massimo di iscritti o sulle lezioni che si svolgeranno. Le uniche

cose che non variano nelle diverse edizioni sono il prezzo e l'obiettivo del corso. Quando un corso viene attivato viene specificata una data di inizio che pone un limite temporale alle iscrizioni. Le **lezioni** previste da quella edizione devono essere fissate entro quella data, non oltre. Deve essere specificato il luogo, la data e l'ora. Un' edizione prevede al massimo una lezione al giorno. È disponibile una collezione di luoghi in cui è possibile svolgere l'attività di arrampicata. I **luoghi** possono essere outdoor oppure indoor. Tutte le lezioni devono avvenire in uno di questi luoghi. Il luogo può variare ad ogni lezione. In un luogo possono avvenire anche più lezioni contemporaneamente. Una lezione tenuta all'interno della palestra garantisce l'accesso durante l'arco della giornata a tutti i partecipanti. Viene richiesto uno storico di tutte le edizioni, comprese le lezioni e le iscrizioni ad esse associate, al fine di fornire dati statistici sulle partecipazioni alle diverse tipologie di corso. La figura dell'istruttore e del cliente possono coincidere. Non esistono divieti per cui un cliente non possa diventare un istruttore oppure un istruttore non possa diventare cliente. Un istruttore può partecipare come cliente alle attività e può acquistare ingressi per un uso futuro. È necessario mantenere traccia degli **accessi** dei clienti e degli istruttori alla palestra. Durante l'arco di una giornata una persona può effettuare più di un accesso.

Progettazione concettuale

Schema scheletro

In figura sono rappresentati i concetti identificati in fase di analisi considerati più importanti.

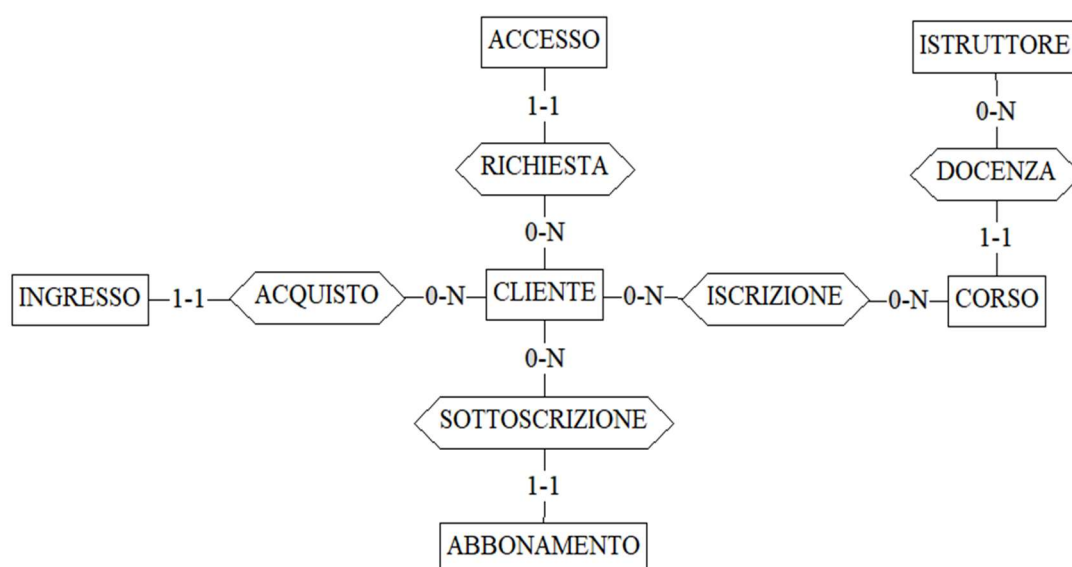


Figura 1: Schema scheletro

Raffinamenti proposti

Si analizzi l'entità *corso*: una tipologia di corso può avere più edizioni nel corso del tempo, è quindi opportuno prevedere una entità *edizione_corso* che rappresenti l'edizione a cui i clienti possono iscriversi e a cui un docente deve fare docenza. L'entità *corso* viene mantenuta, in veste di rappresentante delle diverse tipologie di corso. È opportuno aggiungere una entità *lezione*, che modellizzi accuratamente ciascuna lezione fornendo informazioni quali data, luogo e durata. Il luogo può essere rappresentato dall'entità *luogo*, in quanto proprietario di numerose caratteristiche. È opportuno stabilire una gerarchia totale e non esclusiva che prevede come entità padre l'entità *persona*, ed *istruttore* e *cliente* come entità figlie. La non esclusività è data dalla possibilità di un istruttore di assumere anche il ruolo di cliente. Soffermandoci sugli abbonamenti si noti che è necessario adottare una entità che modellizzi il listino prezzi, al fine di rappresentare tutti i tipi di abbonamenti forniti dalla palestra. Avremo quindi l'entità *abbonamento* con lo scopo di figurare l'abbonamento acquistato dal cliente, e l'entità *tipo_abbonamento* che rappresenta la tipologia acquistata, caratterizzata da durata, fascia di età e il prezzo. È dunque necessaria *fascia_eta*, e con essa l'entità *fascia_oraria*, per raffigurare le diverse fasce di età e orarie che sono previste dalla palestra. È necessario modellare l'aspetto degli ingressi tenendo conto della possibilità di acquistare non solo biglietti singoli, ma anche tessere da più accessi. Una entità *acquisto* è un buon modo per modellare l'acquisto effettuato da un cliente. In termine di prodotti si può dunque acquistare *tessera* e *biglietto*, che rappresentano l'oggetto di cui portano il nome. Anche di questi, come per l'abbonamento, è necessario avere un listino prezzi. Le entità *tipo_tessera* e *tipo_biglietto* modellano correttamente questo aspetto.

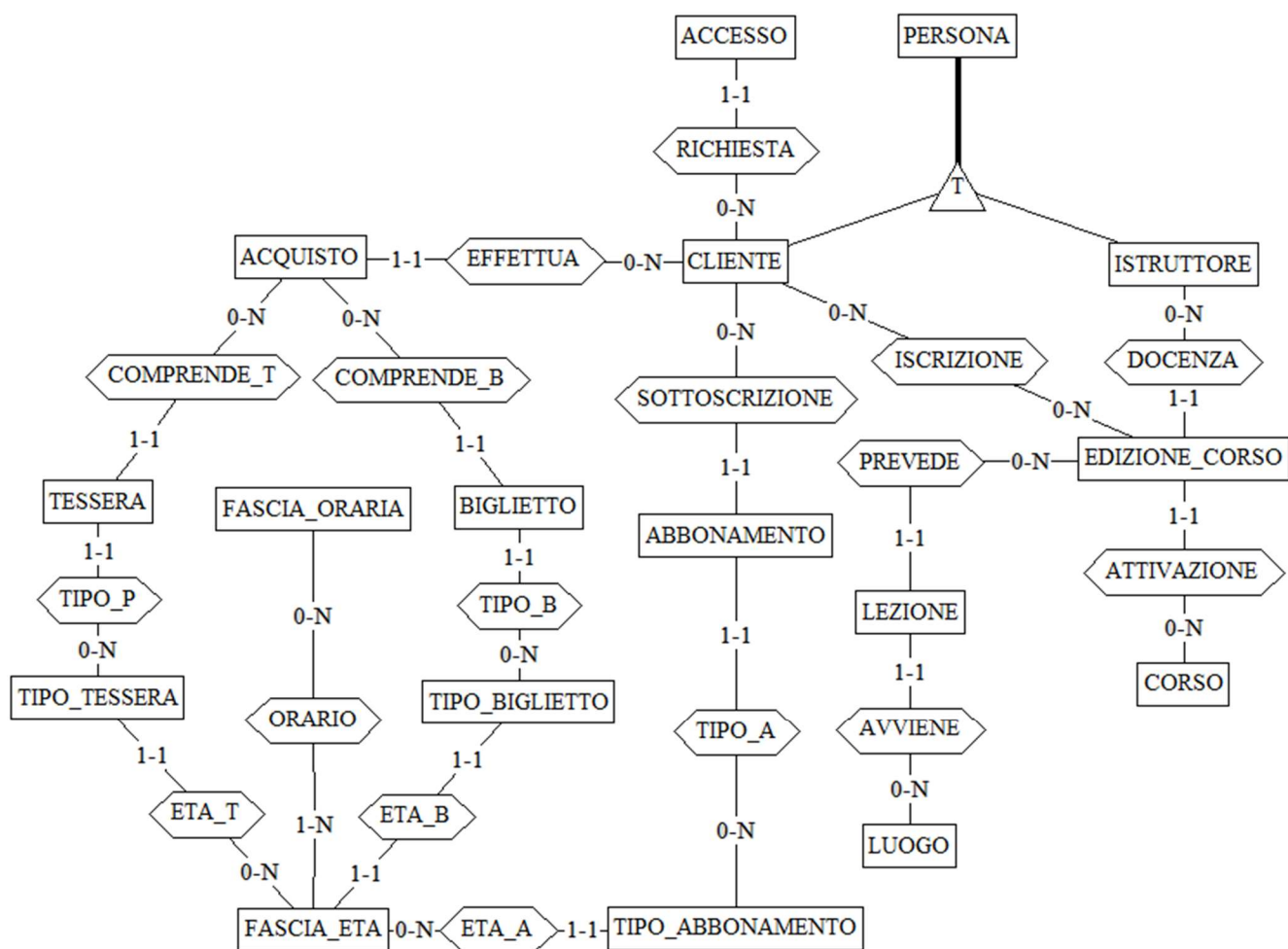


Figura 2: Schema concettuale completo

Figura 3: Schema concettuale finale



Progettazione logica

Stima del volume dei dati

Viene ora mostrata la tabella dei volumi che si stima dovranno essere gestiti in un anno.

Nome	Tipo	Volume di dati
PERSONA	Entità	600
CLIENTE	Entità	595
ISTRUTTORE	Entità	10
ACCESSO	Entità	25000
ACQUISTO	Entità	2000
BIGLIETTO	Entità	1800
TESSERA	Entità	300
TIPO_BIGLIETTO	Entità	3
TIPO_ABBONAMENTO	Entità	12
TIPO_TESSERA	Entità	3
FASCIA_ETA	Entità	3
FASCIA_ORARIA	Entità	3
ABBONAMENTO	Entità	400
EDIZIONE_CORSO	Entità	20
CORSO	Entità	5
LEZIONE	Entità	250
LUOGO	Entità	20
ORARIO	Relazione	5
ISCRIZIONE	Relazione	100

Descrizione delle operazioni principali e stima della loro frequenza

Viene ora mostrata la tabella delle operazioni che possono essere svolte per mezzo dell'applicativo indicando, per ciascuna di esse, la frequenza di utilizzo.

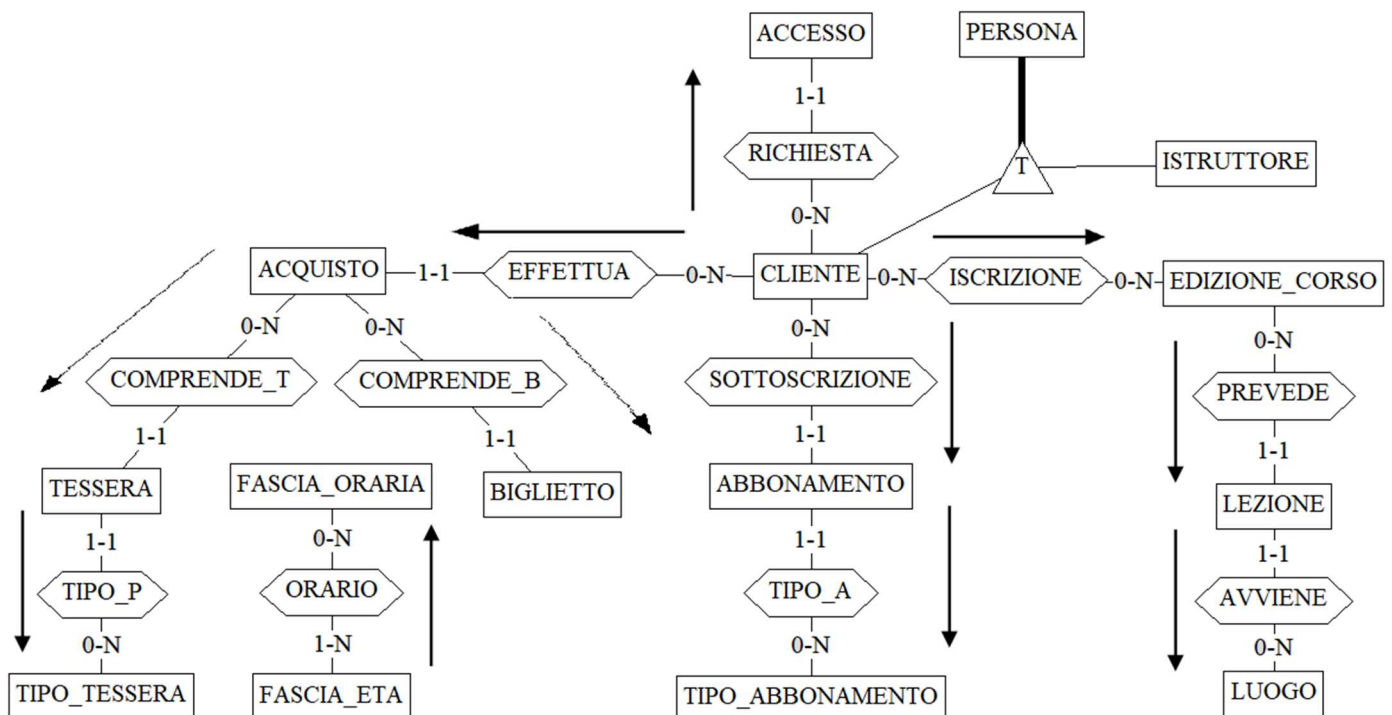
Operazione		Frequenza
ID	Descrizione	
1	Verifica possibilità di accesso alla palestra	50/Giorno
2	Acquisto biglietti e/o tessere	5/Giorno
3	Sottoscrizione di un abbonamento	1/Giorno
4	Iscrizione di un cliente ad un corso attivo	10/Mese
5	Registrazione nuovo cliente o istruttore	5/Settimana
6	Attivazione di un corso	2/Mese
7	Aggiunta di una lezione ad un corso attivo	2/Settimana
8	Mostrare per ciascuna fascia di età il numero di biglietti acquistati	1/Mese
9	Mostrare per ciascun cliente le spese effettuate nell'ultimo anno	1/Mese
10	Mostrare la lista delle persone registrate	1/Giorno
11	Mostrare la lista dei corsi attivi	1/Giorno
12	Mostrare la lista completa di tutte le lezioni in programma	1/Giorno
13	Mostrare per ciascun cliente il numero di ingressi disponibili	1/Giorno
14	Mostrare per ciascun cliente la lista degli abbonamenti attivi	1/Giorno
15	Mostrare per ciascun cliente le lezioni programmate a cui dovrà partecipare	10/Giorno
16	Mostrare i tassi di partecipazione alle diverse tipologie di corso	1/Mese
17	Mostrare per ciascuna tipologia di abbonamento le vendite dell'ultimo mese	1/Mese
18	Lista accessi effettuati	1/Giorno
19	Mostrare per ciascun corso attivo la lista dei partecipanti	1/Giorno

Tutte le operazioni sono tipo di tipo interattivo. Si prendano ora in considerazione operazioni di diversa utilità individuando, per ciascuna di esse, le entità coinvolte. Viene quindi fornita per ognuna lo schema di navigazione e la relativa tavola degli accessi.

Schemi di navigazione e tabelle degli accessi

OP1: Verifica possibilità di accesso alla palestra

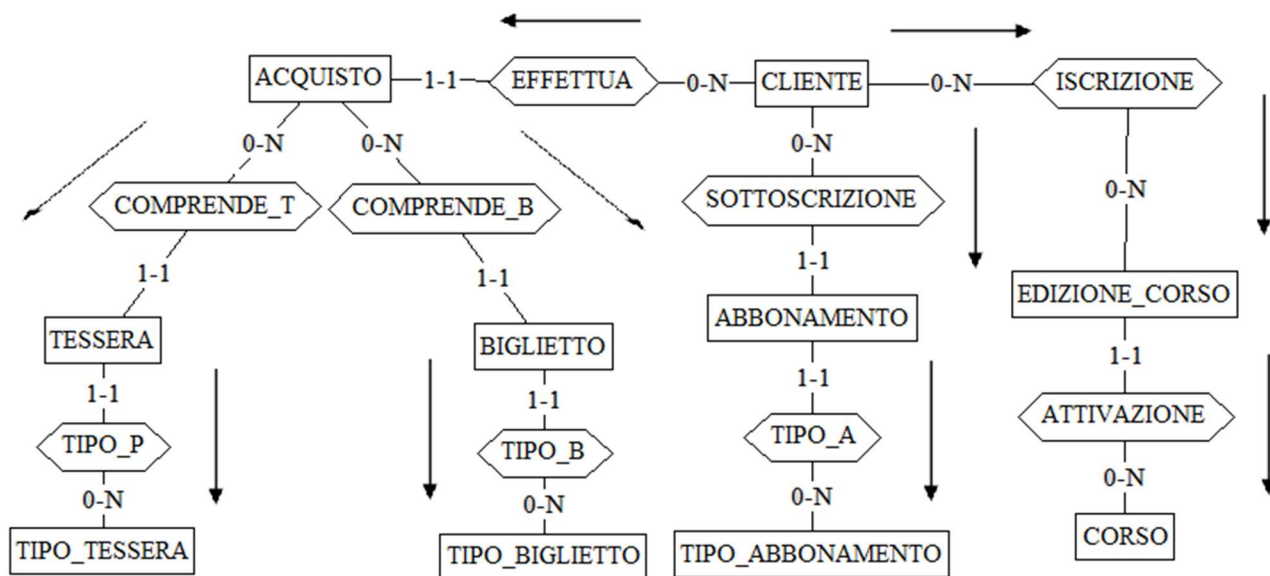
L'operazione consiste nel verificare se un cliente, dato il suo identificativo, possa accedere alla palestra. La richiesta consiste in numerosi passaggi la cui valutazione complessiva porta ad una decisione finale positiva o negativa. Per prima cosa si verifica se colui che fa richiesta abbia il titolo di istruttore, in tal caso può accedere senza alcun vincolo. Nel caso si trattasse di un cliente invece, si controlla se nella data di esecuzione dell'operazione è in programma almeno una lezione all'interno della palestra a cui esso deve partecipare. In tal caso l'accesso è consentito. Successivamente si controlla se l'orario in cui viene effettuata la richiesta rientra nelle fasce orarie previste dalla età del richiedente. Se l'orario non è idoneo il cliente non ha modo di accedere. Se invece la verifica dà esito positivo si effettua una successione di controlli per cui è sufficiente un solo esito positivo per consentire l'accesso. Per prima cosa si ricerca nell'entità *accesso* se l'utente abbia già effettuato un ingresso in giornata, in tal caso è previsto che possa accedere. Segue poi una ricerca di un possibile abbonamento attivo e, ad un eventuale esito negativo, di una tessera con almeno un ingresso valido. Infine, si controlla se l'utente ha un biglietto inutilizzato. Se nessuna delle quattro opzioni restituisce esito positivo l'accesso viene negato; in caso contrario viene registrato l'accesso e, se necessario, l'uso di un biglietto o di una tessera.



Concetto	Costrutto	Accessi	Tipo
PERSONA	E	1	L
FASCIA_ETA	E	1	L
ORARIO	E	1	L
FASCIA_ORARIA	E	1	L
ACCESSO	R	1	L
ABBONAMENTO	E	1	L
TIPO_ABBONAMENTO	E	1	L
ACQUISTO	E	2	L
TESSERA	E	2	L
TIPO_TESSERA	E	1	L
BIGLIETTO	E	1	L
ISCRIZIONE	R	1	L
EDIZIONE_CORSO	E	1	L
LEZIONE	E	1	L
LUOGO	E	1	L
ACCESSO	E	1	S
BIGLIETTO	E	1	S

OP9: Mostrare per ciascun cliente le spese effettuate nell'ultimo anno

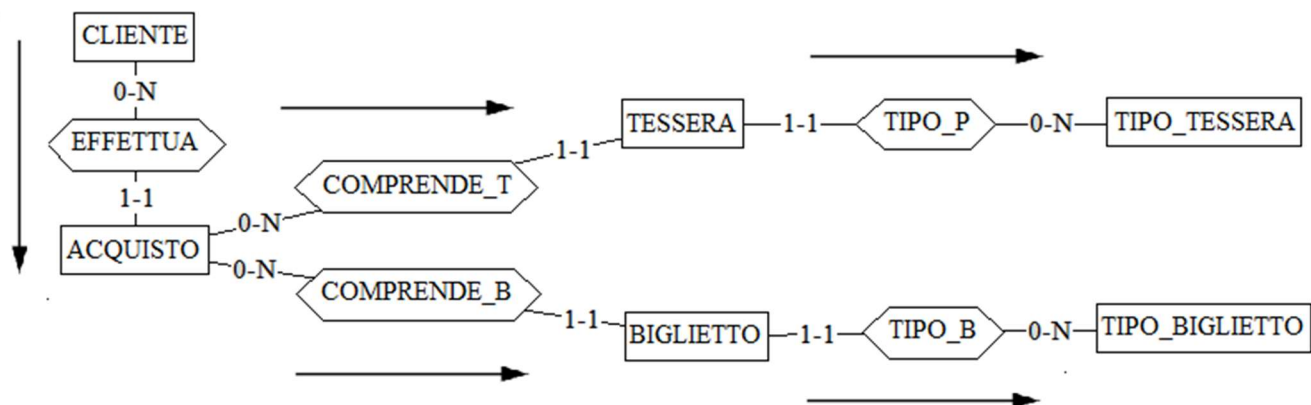
Per ciascun cliente viene valutata la spesa effettuata nell'ultimo anno in abbonamenti, tessere, biglietti e corsi.



Concetto	Costrutto	Accessi	Tipo
CLIENTE	E	595	L
ACQUISTO	E	2000	L
TESSERA	E	300	L
TIPO_TESSERA	E	3	L
BIGLIETTO	E	1800	L
TIPO_BIGLIETTO	E	3	L
ABBONAMENTO	E	400	L
TIPO_ABBONAMENTO	E	12	L
ISCRIZIONE	R	100	L
EDIZIONE_CORSO	E	20	L
CORSO	E	5	L

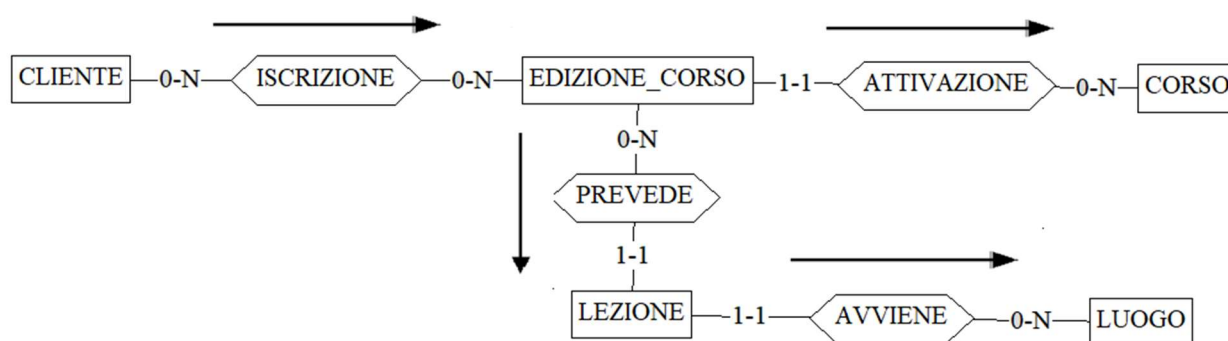
OP13: Mostrare per ciascun cliente il numero di ingressi disponibili

Per ciascun cliente vengono contanti il numero di accessi singoli in suo possesso, tenendo conto delle tessere e dei biglietti.



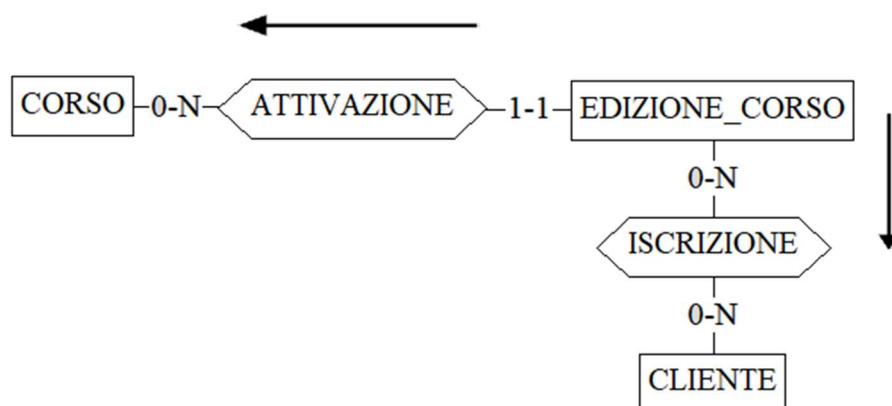
Concetto	Costrutto	Accessi	Tipo
CLIENTE	E	595	L
ACQUISTO	E	2000	L
TESSERA	E	300	L
TIPO_TESSERA	E	3	L
BIGLIETTO	E	1800	L
TIPO_BIGLIETTO	E	3	L

OP15: Mostrare per ciascun cliente le lezioni programmate a cui dovrà partecipare



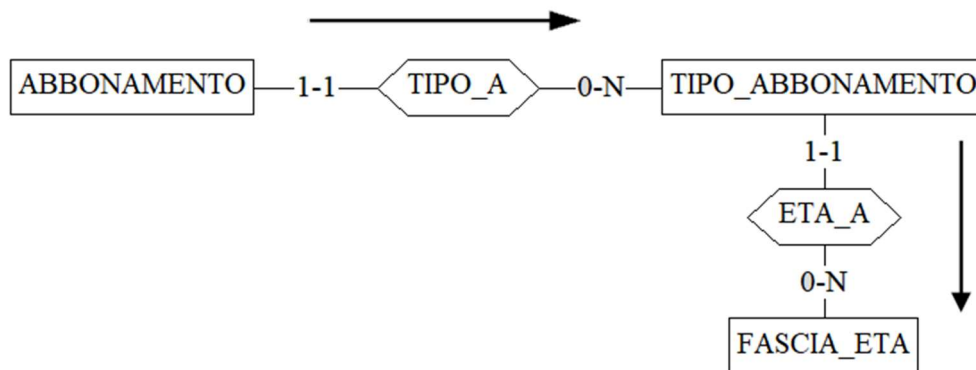
Concetto	Costrutto	Accessi	Tipo
CLIENTE	E	30	L
ISCRIZIONE	E	30	L
EDIZIONE_CORSO	E	2	L
CORSO	E	2	L
LEZIONE	E	15	L
LUOGO	E	5	L

OP16: Mostrare i tassi di partecipazione alle diverse tipologie di corso



Concetto	Costrutto	Accessi	Tipo
CORSO	E	5	L
EDIZIONE_CORSO	E	20	L
ISCRIZIONE	E	100	L

OP17: Mostrare per ciascuna tipologia di abbonamento le vendite dell'ultimo mese



Concetto	Costrutto	Accessi	Tipo
ABBONAMENTO	E	35	L
TIPO_ABBONAMENTO	E	12	L
FASCIA_ETA	E	3	L

Raffinamento dello schema (eliminazione di identificatori esterni, attributi composti e gerarchie, scelta delle chiavi)

È ora trattato il raffinamento dello schema concettuale in previsione della traduzione in schema relazionale.

Scelta delle chiavi primarie

È stato trovato opportuno adottare chiavi primarie diverse rispetto agli identificatori in precedenza definiti per questioni principalmente di comodità. Gli identificatori iniziali erano tutti chiave primarie candidate ma, tenuto conto della necessità di esportarli in altre entità, alcuni di essi sono stati sostituiti da codici univoci. Nello specifico, è stato aggiunto:

- IDAcquisto ad ACQUISTO
- IDFasciaOraria a FASCIA_ORARIA
- IDFasciaEta a FASCIA_ETA
- IDEDizioneCorso a EDIZIONE_CORSO
- IDTipoAbbonamento a TIPO_ABBONAMENTO

Eliminazione delle gerarchie

Lo schema concettuale presenta un'unica gerarchia riferita all'entità *persona*. Si è trovato opportuno, vista la copertura non esclusiva dell'entità, al fine di non introdurre ridondanza, scegliere un collasso verso l'alto. È quindi necessario aggiungere all'entità *persona* l'attributo selettore Istruttore.

Eliminazione degli attributi multivalore

Lo schema concettuale prevede un unico attributo multivalore, presente nell'entità *tessera*. Non riuscendo a prevedere a priori la cardinalità dell'attributo è necessario aggiungere una nuova entità *uso_tessera* che possa rappresentare correttamente l'informazione. Sarà possibile identificare la nuova entità tramite l'attributo DataUso e l'entità *tessera* in quanto, durante l'arco di una giornata, può essere speso un solo ingresso.

Analisi delle ridondanze

All'interno dello schema non sono presenti ridondanze. Viene ora valutata la possibilità di aggiungere un attributo all'entità *edizione_corso* indicante il numero di persone iscritte. Le operazioni coinvolte sono due: l'iscrizione di una persona al corso e la richiesta delle statistiche relative alla partecipazione dei clienti alle edizioni dei corsi. Valutiamo per primo il caso in cui l'attributo non sia presente:

1° operazione				
Concetto	Costrutto	Accessi	Tipo	Accessi al giorno
ISCRIZIONE	R	1	S	1/3

2° operazione				
Concetto	Costrutto	Accessi	Tipo	Accessi al giorno
CORSO	E	5	L	5/30
EDIZIONE_CORSO	E	20	L	20/30
ISCRIZIONE	E	100	L	100/30

Analizziamo invece ora il caso in cui sia presente la ridondanza:

1° operazione				
Concetto	Costrutto	Accessi	Tipo	Accessi al giorno
ISCRIZIONE	R	1	S	1/3
EDIZIONE_CORSO	E	1	L	1/3
EDIZIONE_CORSO	E	1	S	1/3

2° operazione				
Concetto	Costrutto	Accessi	Tipo	Accessi al giorno
CORSO	E	5	L	5/30
EDIZIONE_CORSO	E	20	L	20/30

Esaminando i risultati otteniamo:

- Senza ridondanza: 1/3 Scritture e 125/30 Letture
- Con ridondanza: 2/3 Scritture e 35/30 Letture

Si decide pertanto di mantenere la ridondanza, privilegiando l'efficienza.

Traduzione di entità e associazioni in relazioni

Sono presenti due associazioni molti a molti che devono essere tradotte. La prima intercorre fra *fascia_eta* e *fascia_oraria*, chiamata orario. Essa viene trasformata nell'entità *orario*, identificata dalle chiavi primarie delle entità ad essa collegata. Stesso approccio è stato adottato per *iscrizione*, prevista tra le entità *cliente* e *edizione_corso*. Oltre agli identificatori delle due entità, *iscrizione* mantiene l'attributo relativo alla data di pagamento.

Lo schema presenta una associazione uno ad uno fra *fascia_eta* e *tipo_biglietto*. L'entità *tipo_biglietto* è identificata univocamente da *fascia_eta*, è quindi necessario importare la chiave primaria di quest'ultima e renderla chiave primaria.

Tutte le altre associazioni sono di cardinalità uno a molti, è quindi sufficiente importare nell'entità con cardinalità inferiore la chiave primaria dell'altra.

Viene ora mostrata la traduzione delle entità dello schema concettuale in relazioni:

PERSONA (IDPersona, Nome, Cognome, Cellulare*, Indirizzo*, DataNascita, Istruttore);

ACCESSO (IDPersona: PERSONA, DataOra);

LEZIONE (IDEdizioneCorso: EDIZIONE_CORSO, DataOra, Durata, IDLuogo : LUOGO);

LUOGO (IDLuogo, Nome, Outdoor, NumeroPercorsi, Via, Zona, Descizione*);

EDIZIONE_CORSO (IDEdizioneCorso, DataInizio, IDCorso: CORSO, NumeroPartecipanti, IDPersona: PERSONA);

CORSO (IDCorso, Prezzo, Nome, Descrizione);

ISCRIZIONE (IDPersona: PERSONA, IDEdizioneCorso: EDIZIONE_CORSO, DataPagamento);

ABBONAMENTO (IDPersona: PERSONA, DataInizio, DataOraPagamento, IDTipoAbbonamento: TIPO_ABBONAMENTO);

TIPO_ABBONAMENTO (IDTipoAbbonamento, Prezzo, MesiDurata, IDFasciaEta: FASCIA_ETA);

ACQUISTO (IDAcquisto, DataOraAcquisto, IDPersona: PERSONA);

BIGLIETTO (IDBiglietto, DataUso*, IDAcquisto: ACQUISTO, IDTipoBiglietto : TIPO_BIGLIETTO);

TIPO_BIGLIETTO (IDTipoBiglietto: FASCIA_ETA, Prezzo);

FASCIA_ETA (IDFasciaEta, Nome, InizioEta, FineEta);

FASCIA_ORARIA (IDFasciaOraria, Nome, InizioOrario, FineOrario);

ORARIO (IDFasciaOraria: FASCIA_ORARIA, IDFasciaEta : FASCIA_ETA);

TIPO_TESSERA (IDTipoTessera: FASCIA_ETA, Prezzo, NumeroBiglietti);

TESSERA (IDTessera, IDTipoTessera: TIPO_TESSERA, IDAcquisto: ACQUISTO, NumeroBiglietti : TIPO_TESSERA);

USO_TESSERA (IDTessera: TESSERA, DataUso);

Schema relazionale finale

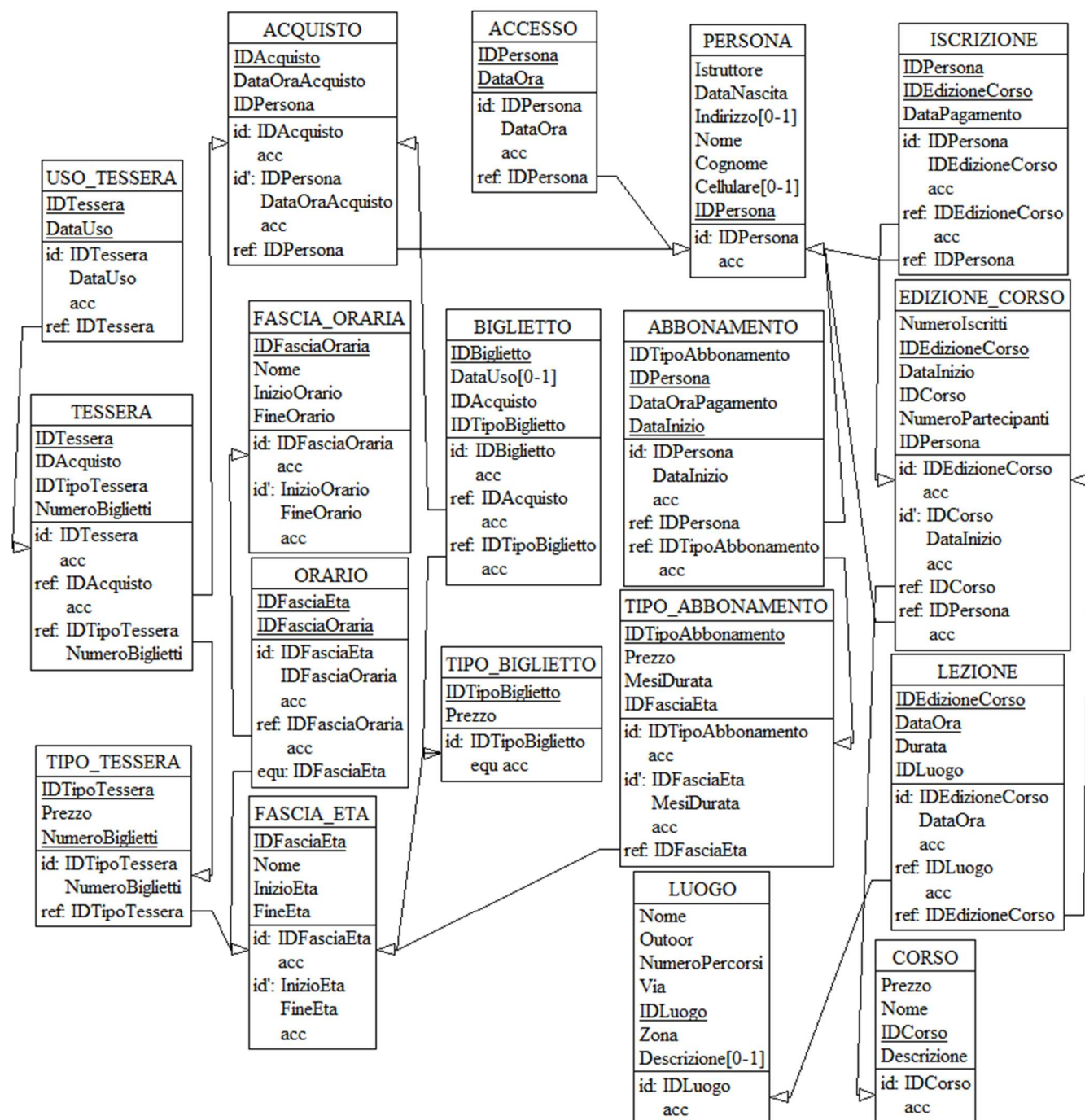


Figura 4: Schema relazionale

Traduzione delle operazioni in query SQL

Sono ora mostrate le interrogazioni SQL più significative.

OP1: Verifica possibilità di accesso alla palestra

Viene verificata la possibilità che una certa matricola fornita in input possa, in un certo istante, accedere alla palestra.

```
select *
from PERSONA join ISCRIZIONE on PERSONA.IDPersona = ISCRIZIONE.IDPersona
join EDIZIONE_CORSO on ISCRIZIONE.IDEdizioneCorso =
EDIZIONE_CORSO.IDEdizioneCorso
join LEZIONE on EDIZIONE_CORSO.IDEdizioneCorso = LEZIONE.IDEdizioneCorso
join LUOGO on LEZIONE.IDLuogo = LUOGO.IDLuogo
where CONVERT(VARCHAR(10), LEZIONE.DataOra , 111) = CONVERT(VARCHAR(10),
GETDATE() , 111)
and PERSONA.IDPersona = matricola and LUOGO.Nome = 'Palestra'
```

```
select *
from FASCIA_ETA, FASCIA_ORARIA, ORARIO
where FASCIA_ETA.IDFasciaEta = ORARIO.IDFasciaEta
and FASCIA_ORARIA.IDFasciaOraria = ORARIO.IDFasciaOraria
and eta >= FASCIA_ETA.InizioEta
and eta < FASCIA_ETA.FineEta
and CONVERT (time, SYSDATETIME()) >= FASCIA_ORARIA.InizioOrario
and CONVERT (time, SYSDATETIME()) < FASCIA_ORARIA.FineOrario
```

```
select *
from ACCESSO
where ACCESSO.IDPersona = matricola
and DATEDIFF (day, ACCESSO.DataOra, GETDATE ()) = 0
```

```
select *
from ABBONAMENTO, TIPO_ABBONAMENTO
where ABBONAMENTO.IDTipoAbbonamento = TIPO_ABBONAMENTO.IDTipoAbbonamento
and IDPersona = matricola
and DATEDIFF(day, ABBONAMENTO.DataInizio, GETDATE())
```



```

    <= TIPO_ABBONAMENTO.MesiDurata*30
and GETDATE () >= ABBONAMENTO.DataInizio

select top 1 IDTessera
from (
    select TESSERA.IDTessera, TESSERA.IDTipoTessera,
    COUNT(USO_TESSERA.DataUso) as Accessi
    from ACQUISTO inner join TESSERA on ACQUISTO.IDAcquisto =
    TESSERA.IDAcquisto
    left join USO_TESSERA on TESSERA.IDTessera = USO_TESSERA.IDTessera
    where ACQUISTO.IDPersona=3
    group by TESSERA.IDTessera, TESSERA.IDTipoTessera
) as Accessi, TIPO_TESSERA
where Accessi.IDTipoTessera=TIPO_TESSERA.IDTipoTessera
and Accessi.Accessi<TIPO_TESSERA.NumeroBiglietti

select top 1 IDBiglietto
from Biglietto join Acquisto on BIGLIETTO.IDAcquisto = ACQUISTO.IDAcquisto
and ACQUISTO.IDPersona = matricola and BIGLIETTO.DATAUSO is NULL

```

OP16: Mostrare i tassi di partecipazione alle diverse tipologie di corso

```

select CORSO.Nome, COUNT(DISTINCT(EDIZIONE_CORSO.IDEdizioneCorso)) as
NumeroEdizioni, ROUND(cast(sum(EDIZIONE_CORSO.NumeroIscritti) as float) /
cast(sum(EDIZIONE_CORSO.NumeroPartecipanti) as float),2) as
RateoPartecipazione, (COUNT(ISCRIZIONE.DataPagamento) * CORSO.Prezzo) /
COUNT(DISTINCT(EDIZIONE_CORSO.IDEdizioneCorso)) as IncassoMedioEdizione,
COUNT(ISCRIZIONE.DataPagamento) as IscrittiTotali, sum(NumeroPartecipanti) as
TotalePosti
from CORSO join EDIZIONE_CORSO on EDIZIONE_CORSO.IDCorso = CORSO.IDCorso
left join ISCRIZIONE on EDIZIONE_CORSO.IDEdizioneCorso =
ISCRIZIONE.IDEdizioneCorso
where EDIZIONE_CORSO.DataInizio < GETDATE()
group by CORSO.IDCorso, CORSO.Nome, CORSO.Prezzo
order by RateoPartecipazione ASC

```

OP4: Iscrizione di un cliente ad un corso attivo

Interrogazione per ottenere l'insieme dei corsi a cui una certa matricola può iscriversi.

```
select  EDIZIONE_CORSO.IDEdizioneCorso,  COUNT(ISCRIZIONE.DataPagamento)  as
ISCRITTI,  EDIZIONE_CORSO.NumeroPartecipanti, EDIZIONE_CORSO.IDCorso, CORSO.Nome,
EDIZIONE_CORSO.DataInizio
from CORSO inner join EDIZIONE_CORSO on CORSO.IDCorso = EDIZIONE_CORSO.IDCorso
left join ISCRIZIONE on EDIZIONE_CORSO.IDEdizioneCorso =
ISCRIZIONE.IDEdizioneCorso
where EDIZIONE_CORSO.DataInizio > GETDATE()
and EDIZIONE_CORSO.IDEdizioneCorso not in
(
    select ISCRIZIONE.IDEdizioneCorso
    from PERSONA join ISCRIZIONE on PERSONA.IDPersona = ISCRIZIONE.IDPersona
    where PERSONA.IDPersona=matricola
)
group by EDIZIONE_CORSO.IDEdizioneCorso, EDIZIONE_CORSO.NumeroPartecipanti,
EDIZIONE_CORSO.IDCorso, CORSO.Nome, EDIZIONE_CORSO.DataInizio
having COUNT(ISCRIZIONE.DataPagamento) < EDIZIONE_CORSO.NumeroPartecipanti
```

OP17: Mostrare per ciascuna tipologia di abbonamento le vendite dell'ultimo mese

```
Select TIPO_ABBONAMENTO.IDTipoAbbonamento, FASCIA_ETA.InizioEta,
FASCIA_ETA.FineEta, TIPO_ABBONAMENTO.MesiDurata, TIPO_ABBONAMENTO.Prezzo as
PrezzoSingolo, COUNT(*) as NumeroAcquisti, SUM(Prezzo) as TotaleIncasso
from ABBONAMENTO join TIPO_ABBONAMENTO
on ABBONAMENTO.IDTipoAbbonamento = TIPO_ABBONAMENTO.IDTipoAbbonamento
join FASCIA_ETA on FASCIA_ETA.IDFasciaEta = TIPO_ABBONAMENTO.IDFasciaEta
where DATEDIFF(day, GETDATE(), ABBONAMENTO.DataOraPagamento) < 30
group by TIPO_ABBONAMENTO.IDTipoAbbonamento, TIPO_ABBONAMENTO.MesiDurata,
FASCIA_ETA.InizioEta, FASCIA_ETA.FineEta, TIPO_ABBONAMENTO.Prezzo
order by TotaleIncasso desc
```

OP9: Mostrare per ciascun cliente le spese effettuate nell'ultimo anno

```
select PERSONA.IDPersona, PERSONA.Nome, PERSONA.Cognome,
IsNull(Tessere.SpesaTessere,0) as SpesaTessera,
IsNull(Biglietti.SpesaBiglietti,0) as SpesaBiglietti,
IsNull(Abbonamenti.SpesaAbbonamenti,0) as SpesaAbbonamenti,
IsNull(Corsi.SpesaCorsi,0) as SpesaCorsi, IsNull(Tessere.SpesaTessere,0) +
IsNull(Biglietti.SpesaBiglietti,0) + IsNull(Abbonamenti.SpesaAbbonamenti,0) +
IsNull(Corsi.SpesaCorsi,0) as SpesaTotale
from PERSONA left join
    (select ACQUISTO.IDPersona, sum(TIPO_TESSERA.Prezzo) as SpesaTessere
    from ACQUISTO join TESSERA on ACQUISTO.IDAcquisto = TESSERA.IDAcquisto
    join TIPO_TESSERA on TESSERA.IDTipoTessera = TIPO_TESSERA.IDTipoTessera
    and TESSERA.NumeroBiglietti = TIPO_TESSERA.NumeroBiglietti
    where ACQUISTO.DataOraAcquisto > dateadd(YEAR, -1, GETDATE())
    group by ACQUISTO.IDPersona) as Tessere
on PERSONA.IDPersona = Tessere.IDPersona
left join
    (select ACQUISTO.IDPersona, sum(TIPO_BIGLIETTO.Prezzo) as SpesaBiglietti
    from ACQUISTO join BIGLIETTO on ACQUISTO.IDAcquisto =
    BIGLIETTO.IDAcquisto
    join TIPO_BIGLIETTO on BIGLIETTO.IDTipoBiglietto =
    TIPO_BIGLIETTO.IDTipoBiglietto
    where ACQUISTO.DataOraAcquisto > dateadd(YEAR, -1, GETDATE())
    group by ACQUISTO.IDPersona) as Biglietti
on PERSONA.IDPersona = Biglietti.IDPersona
left join
    (select ABBONAMENTO.IDPersona, sum(TIPO_ABBONAMENTO.Prezzo) as
    SpesaAbbonamenti
    from ABBONAMENTO join TIPO_ABBONAMENTO on ABBONAMENTO.IDTipoAbbonamento =
    TIPO_ABBONAMENTO.IDTipoAbbonamento
    where ABBONAMENTO.DataOraPagamento > dateadd(YEAR, -1, GETDATE())
    group by ABBONAMENTO.IDPersona) as Abbonamenti
on PERSONA.IDPersona = Abbonamenti.IDPersona
left join
    (select ISCRIZIONE.IDPersona, sum(CORSO.Prezzo) as SpesaCorsi
```

```

from CORSO join EDIZIONE_CORSO on EDIZIONE_CORSO.IDCorso = CORSO.IDCorso
join ISCRIZIONE on ISCRIZIONE.IDEdizioneCorso =
EDIZIONE_CORSO.IDEdizioneCorso
where ISCRIZIONE.DataPagamento > dateadd(YEAR, -1, GETDATE())
group by ISCRIZIONE.IDPersona) as Corsi
on PERSONA.IDPersona = Corsi.IDPersona
order by SpesaTotale desc

```

OP13: Mostrare per ciascun cliente il numero di ingressi disponibili

```

select UsoTessere.IDPersona, UsoTessere.Nome, UsoTessere.Cognome,
IsNull(sum(UsoTessere.NumeroBiglietti * UsoTessere.NumeroPerTipo) -
sum(UsoTessere.AccessiEseguiti),0) as AccessiRimastiTessere,
Biglietti.BigliettiRimasti,
IsNull(sum(UsoTessere.NumeroBiglietti * UsoTessere.NumeroPerTipo) -
sum(UsoTessere.AccessiEseguiti),0)+ Biglietti.BigliettiRimasti as AccessiTotali
from (
    select PERSONA.IDPersona, PERSONA.Nome, PERSONA.Cognome,
    TESSERA.IDTipoTessera, TESSERA.NumeroBiglietti,
    COUNT(TESSERA.IDTessera) as NumeroPerTipo,
    COUNT(USO_TESSERA.DataUso) as AccessiEseguiti
    from ACQUISTO join TESSERA on ACQUISTO.IDAcquisto = TESSERA.IDAcquisto
    left join USO_TESSERA on TESSERA.IDTessera = USO_TESSERA.IDTessera
    join TIPO_TESSERA on (TESSERA.NumeroBiglietti =
    TIPO_TESSERA.NumeroBiglietti and TESSERA.IDTipoTessera =
    TIPO_TESSERA.IDTipoTessera)
    right join PERSONA on PERSONA.IDPersona = ACQUISTO.IDPersona
    group by PERSONA.IDPersona, PERSONA.Nome, PERSONA.Cognome,
    TESSERA.IDTipoTessera, TESSERA.NumeroBiglietti) as UsoTessere
join (
    select PERSONA.IDPersona, PERSONA.Nome, PERSONA.Cognome,
    COUNT(BIGLIETTO.IDBiglietto) - COUNT(BIGLIETTO.DataUso) as
    BigliettiRimasti
    from ACQUISTO join BIGLIETTO on ACQUISTO.IDAcquisto =
    BIGLIETTO.IDAcquisto
    right join PERSONA on ACQUISTO.IDPersona = PERSONA.IDPersona

```

```

        group by PERSONA.IDPersona, PERSONA.Nome, PERSONA.Cognome) as Biglietti
on UsoTessere.IDPersona = Biglietti.IDPersona
group by UsoTessere.IDPersona, Biglietti.BigliettiRimasti, UsoTessere.Nome,
UsoTessere.Cognome
order by UsoTessere.IDPersona asc

```

OP15: Mostrare per ciascun cliente le lezioni programmate a cui dovrà partecipare

```

select PERSONA.IDPersona, PERSONA.Nome, PERSONA.Cognome,
CORSO.Nome as TipologiaCorso, LEZIONE.DataOra, LEZIONE.Durata as Durata,
LUOGO.Nome as Luogo
from PERSONA join ISCRIZIONE on PERSONA.IDPersona = ISCRIZIONE.IDPersona
join EDIZIONE_CORSO on ISCRIZIONE.IDEdizioneCorso =
EDIZIONE_CORSO.IDEdizioneCorso
join CORSO on EDIZIONE_CORSO.IDCorso = CORSO.IDCorso
join LEZIONE on EDIZIONE_CORSO.IDEdizioneCorso = LEZIONE.IDEdizioneCorso
join LUOGO on LEZIONE.IDLuogo = LUOGO.IDLuogo
where LEZIONE.DataOra > GETDATE()
order by PERSONA.IDPersona asc

```

OP19: Mostrare per ciascun corso attivo la lista dei partecipanti

```

select EDIZIONE_CORSO.IDEdizioneCorso, CORSO.Nome, CONVERT(VARCHAR(10),
EDIZIONE_CORSO.DataInizio, 111) as DataInizio, PERSONA.IDPersona, PERSONA.Nome
as NomePartecipante, PERSONA.Cognome as CognomePartecipante
from PERSONA join ISCRIZIONE on PERSONA.IDPersona = ISCRIZIONE.IDPersona
join EDIZIONE_CORSO on ISCRIZIONE.IDEdizioneCorso =
EDIZIONE_CORSO.IDEdizioneCorso
join CORSO on EDIZIONE_CORSO.IDCorso = CORSO.IDCorso
where EDIZIONE_CORSO.DataInizio > GETDATE() or
EDIZIONE_CORSO.IDEdizioneCorso in (
    select EDIZIONE_CORSO.IDEdizioneCorso
    from EDIZIONE_CORSO join LEZIONE on EDIZIONE_CORSO.IDEdizioneCorso =
    LEZIONE.IDEdizioneCorso
    where LEZIONE.DataOra > GETDATE()
    group by EDIZIONE_CORSO.IDEdizioneCorso )
order by EDIZIONE_CORSO.IDEdizioneCorso asc

```

OP11: Mostrare la lista dei corsi attivi

```
select EDIZIONE_CORSO.IDEdizioneCorso, CONVERT(VARCHAR(10),
EDIZIONE_CORSO.DataInizio, 111) as DataInizio,
EDIZIONE_CORSO.NumeroPartecipanti as MassimiPartecipanti,
EDIZIONE_CORSO.NumeroIscritti, CORSO.Nome, COUNT(LEZIONE.DataOra) as
LezioniRimaste, PERSONA.Cognome as Istruttore
from EDIZIONE_CORSO left join LEZIONE on EDIZIONE_CORSO.IDEdizioneCorso =
LEZIONE.IDEdizioneCorso
join PERSONA on EDIZIONE_CORSO.IDPersona = PERSONA.IDPersona
join CORSO on EDIZIONE_CORSO.IDCorso = CORSO.IDCorso
and (LEZIONE.DataOra > GETDATE() or EDIZIONE_CORSO.DataInizio > GETDATE())
group by EDIZIONE_CORSO.IDEdizioneCorso, EDIZIONE_CORSO.DataInizio,
EDIZIONE_CORSO.NumeroPartecipanti, EDIZIONE_CORSO.NumeroIscritti, CORSO.Nome,
PERSONA.Cognome
```

OP14: Mostrare per ciascun cliente la lista degli abbonamenti attivi

```
select PERSONA.IDPersona, PERSONA.Nome, PERSONA.Cognome, CONVERT(VARCHAR(10),
ABBONAMENTO.DataInizio, 111) as DataInizio, CONVERT(VARCHAR(10), dateadd(DAY,
TIPO_ABBONAMENTO.MesiDurata*30, ABBONAMENTO.DataInizio), 111) as DataFine
from PERSONA left join ABBONAMENTO on PERSONA.IDPersona = ABBONAMENTO.IDPersona
join TIPO_ABBONAMENTO on ABBONAMENTO.IDTipoAbbonamento =
TIPO_ABBONAMENTO.IDTipoAbbonamento
where dateadd(DAY, TIPO_ABBONAMENTO.MesiDurata*30, ABBONAMENTO.DataInizio) >
GETDATE()
order by PERSONA.IDPersona asc
```

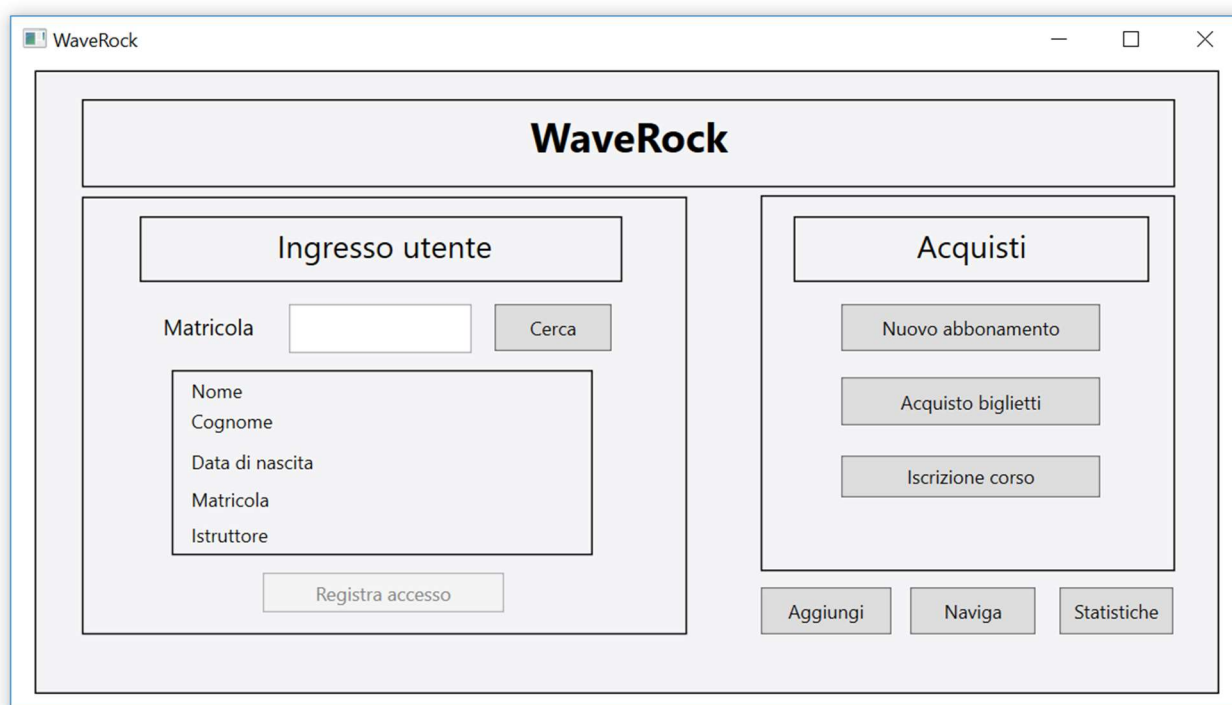
OP8: Mostrare per ciascuna fascia di età il numero di biglietti acquistati

```
select FASCIA_ETA.IDFasciaEta, FASCIA_ETA.InizioEta, FASCIA_ETA.FineEta,
COUNT(BIGLIETTO.IDBIGLIETTO) as BigliettiAcquistati
from BIGLIETTO join TIPO_BIGLIETTO on BIGLIETTO.IDTipoBiglietto =
TIPO_BIGLIETTO.IDTipoBiglietto right join FASCIA_ETA on
TIPO_BIGLIETTO.IDTipoBiglietto = FASCIA_ETA.IDFasciaEta
group by FASCIA_ETA.IDFasciaEta, FASCIA_ETA.InizioEta, FASCIA_ETA.FineEta
order by BigliettiAcquistati desc
```

Progettazione dell'applicazione

Descrizione dell'architettura dell'applicazione realizzata

L'applicativo è stato sviluppato in linguaggio C#, adottando la tecnologia Windows Presentation Form (o WPF) per lo sviluppo dell'interfaccia grafica. L'interazione con il database da parte dell'applicativo avviene attraverso la componente Framework Language Integrated Query (LINQ), che consente di effettuare interrogazioni utilizzando una sintassi simile a SQL. Si sono inoltre adottate, per effettuare operazioni che sarebbero risultate complesse con il solo uso di LINQ, le classi presenti all'interno del namespace System.Data.SqlClient. Al fine di semplificare l'interfacciamento alla base dati ho trovato opportuno adottare delle classi per il mapping O-R. L'applicativo si presenta come segue:



In figura si possono distinguere le principali funzionalità previste da un comune gestionale di una palestra. Tutte le operazioni possono essere trovate navigando correttamente all'interno dell'applicativo. Fatta eccezione per le funzioni presenti nell'interfaccia principale, si hanno quelle relative all'inserimento di elementi, come persone e lezioni, nell'interfaccia apribile con il bottone *Aggiungi*, e quelle per poter visualizzare le informazioni contenute nella base dati attraverso il bottone *Naviga*. Sono inoltre disponibili operazioni statistiche nell'interfaccia del bottone *Statistiche*.