UNIVERSITÀ
degli STUDI
di CATANIA

Progetto di Basi di Dati

Gestione di una palestra

Scivoletto Mattia Maria
Matricola X81000934

Anno Accademico 2019/2020

INDICE

1. Descrizione delle specifiche della base di dati

Si vuole progettare una base di dati per la gestione di una palestra, dove si tengono corsi di fitness e di arti marziali. In particolare, si vogliono gestire le informazioni riguardanti la clientela della palestra, divisa in clienti abbonati, che possono dunque seguire interi corsi e che pagheranno un'abbonamento mensile, e clienti occasionali, che invece possono soltanto partecipare a singole lezioni, pagate sul momento. Le due categorie di clienti devono essere mantenute distinte poichè gli abbonati riceveranno una tessera e possono partecipare alle gare a cui è iscritta la palestra, mentre quest'ultima potrebbe essere interessata ad inviare offerte ai clienti occasionali tramite email. Dell'abbonato è di interesse conoscere il nome, il cognome, l'identificativo, il codice fiscale, un recapito telefonico, l'indirizzo email, il numero di conto corrente, l'età, il debito e il credito che esso ha nei confronti della palestra. Il credito può essere accumulato, sulla base di un sistema di punteggi, solo dagli agonisti che partecipano alle gare, che sono un sottogruppo degli abbonati. Ogni 3 gare, all'agonista viene assegnato un punto. Periodicamente si provvederà al calcolo del credito e all'azzeramento del punteggio. Per quanto riguarda i clienti occasionali, bisogna conoscere nome, cognome, codice fiscale, indirizzo email, numero di telefono, età. Si vogliono poi conoscere i dettagli relativi agli istruttori che tengono i corsi, in particolare nome, cognome, codice fiscale, anni di esperienza lavorativa, titolo di studio. Dei corsi si vuole conoscere codice identificativo, nome, tariffa mensile, numero di lezioni, durata ed edizione, dato che un corso può tenersi in anni differenti.
Le lezioni che compongono i corsi sono caratterizzate da numero identificativo, nome, tipologia e durata.

2.1 Glossario dei termini

TERMINE	DESCRIZIONE	SINONIMI	TERMINI COLLEGATI
Corso	È composto da lezioni		Cliente, Istruttore, Lezione
Cliente	Può essere abbonato oppure occasionale, si iscrive ai corsi o segue singole lezioni	Clientela	Corso
Istruttore	Tiene le lezioni	Allenatore	Corso
Lezione	Fa parte di un corso		Corso

2.2 Dati di carattere generale

Si vuole progettare una base di dati per la gestione di una palestra, dove si tengono corsi di fitness e di arti marziali.

2.3 Dati sui corsi

Dei corsi si vuole conoscere codice identificativo, nome, durata, numero di lezioni, tariffa mensile ed edizione, dato che un corso può tenersi in anni differenti.

2.4 Dati sui clienti

Si vogliono gestire le informazioni riguardanti la clientela della palestra, divisa in clienti abbonati, che possono dunque seguire interi corsi e che pagheranno un’abbonamento mensile, e clienti occasionali, che invece possono soltanto partecipare a singole lezioni, pagate sul momento.

2.4.1 Dati sui clienti abbonati
Possono seguire interi corsi e pagheranno un'abbonamento mensile. Gli abbonati riceveranno una tessera e possono partecipare alle gare a cui è iscritta la palestra. Dell'abbonato è di interesse conoscere il nome, il cognome, l'identificativo, il codice fiscale, un recapito telefonico, l'indirizzo email, il numero di conto corrente, l'età, il debito e il credito che esso ha nei confronti della palestra.

2.4.2 Dati sui clienti occasionali
Per quanto riguarda i clienti occasionali, bisogna conoscere nome, cognome, codice fiscale, indirizzo email, numero di telefono, età. Possono soltanto partecipare a singole lezioni, pagate sul momento.

2.5 Dati sugli istruttori
Si vogliono poi conoscere i dettagli relativi agli istruttori che tengono i corsi, in particolare nome, cognome, codice fiscale, anni di esperienza lavorativa, titolo di studio.

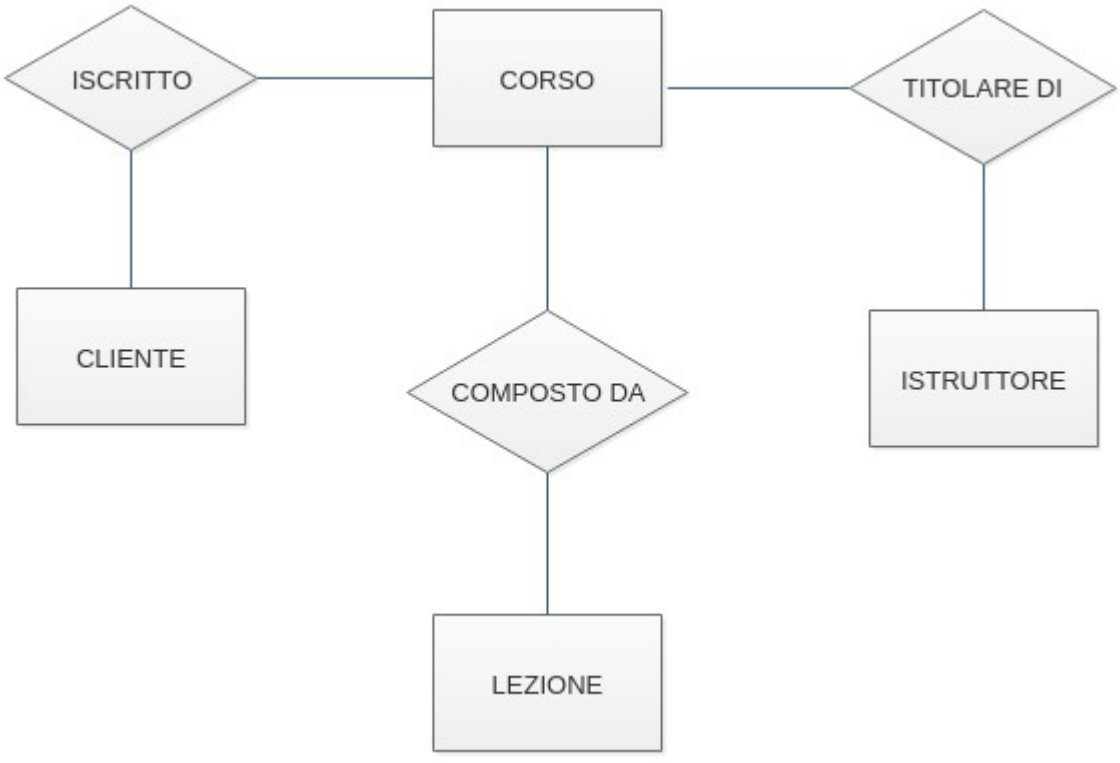
2.6 Dati sulle lezioni
Le lezioni che compongono i corsi sono caratterizzate da numero identificativo, nome, tipologia e durata.

2.7 Specifiche sulle operazioni

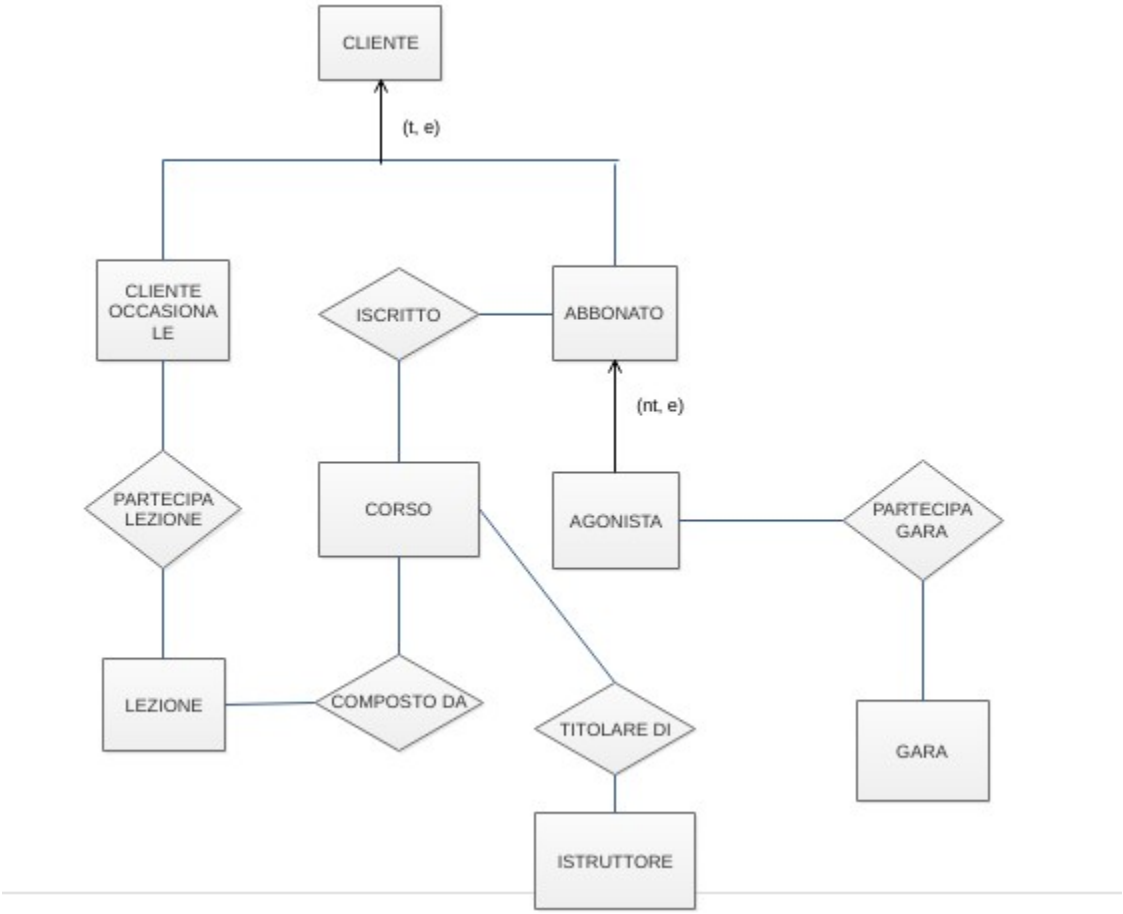
O1 Addebita il costo degli abbonamenti ad ogni cliente (freq.: 1 volta al mese);
O2 Stampa gli istruttori con più corsi di cui sono attualmente titolari (freq.: 1 volta ogni 3 mesi);
O3 Calcola il credito derivante dal punteggio gara cumulato dall'agonista, poi azzera il punteggio

(freq.: 1 volta ogni 2 mesi);
O4 Inserisci una nuova partecipazione a gare (freq.: 1 volta a settimana);
O5 Inserisci un nuovo abbonato (freq.: 7 volte al giorno);
O6 Stampa i dati di tutti gli abbonati (freq.: 1 volta ogni 3 mesi);
O7 Aggiorna il punteggio gara dell'agonista dopo l'inserimento di una nuova gara (freq.: 1 volta a settimana);
O8 Assegna un punteggio pari a 0 ad un nuovo cliente abbonato agonista (freq.: 3 volte al giorno)
O9 Controlla che solo gli agonisti risultino essere partecipanti alle gare, nel caso di abbonati non agonisti eliminare le loro partecipazioni (freq.: 1 volta a settimana);
O10 Sposta le partecipazioni a vecchie gare (quelle di cui si è già calcolato il credito del cliente) in un archivio apposito (freq.: 1 volta ogni 2 mesi);
O11 Stampa l'ammontare che ogni abbonato deve realmente pagare, cioè la differenza tra credito e debito (freq.: 1 volta al mese);

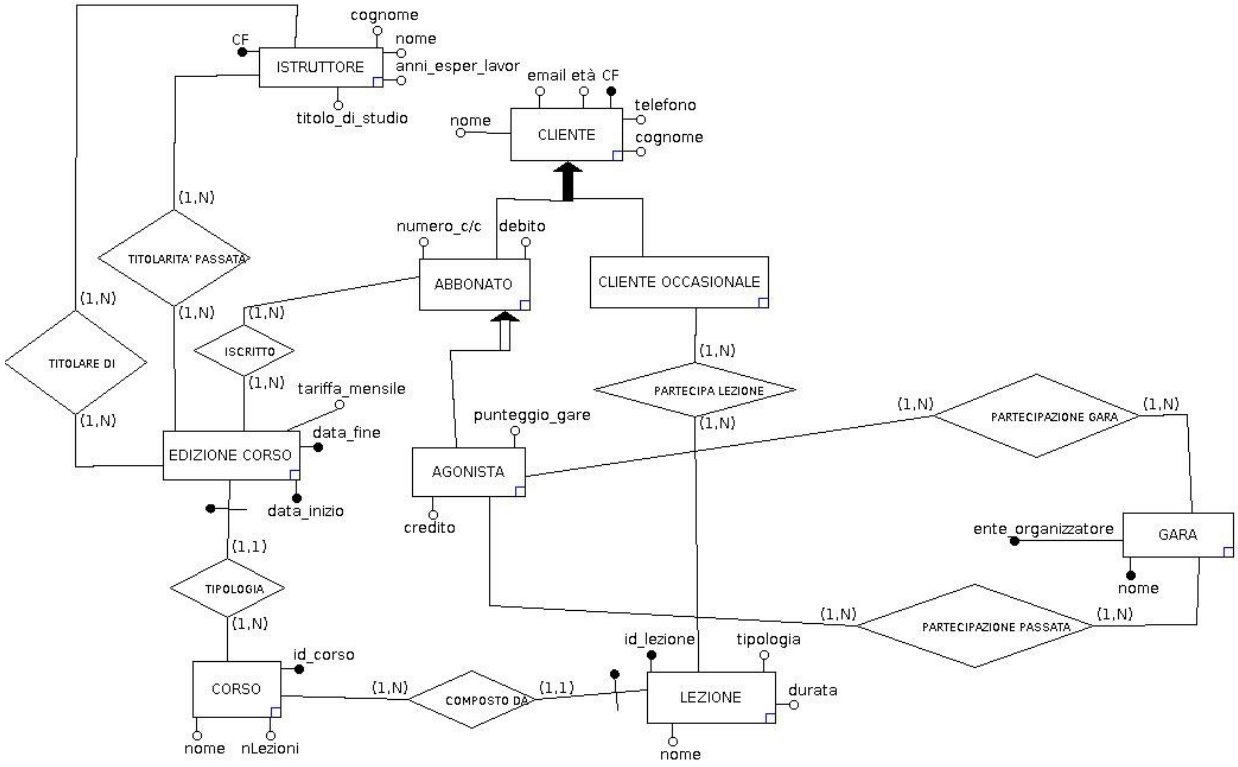
3.1 Schema scheletro



3.2 Schema intermedio



3.3 Schema finale



3.4 Dizionario dei dati – Entità

Entità	***Descrizione***	***Attributi***	***Identificatore***
Cliente	Utente della palestra	nome, cognome, CF, email, età, telefono	CF
Abbonato	Cliente con abbonamento e che partecipa ai corsi	numero_c/c, debito	
Cliente Occasionale	Partecipa alle singole lezioni, che paga sul momento		
Istruttore	É titolare di corsi	CF, nome, cognome, titolo_di_studio, anni_esper_lavor	CF
Edizione Corso	Edizione di un dato corso	data_inizio, data_fine, tariffa_mensile	data_inizio, data_fine, id_corso (tramite Corso)
Corso	Insieme di lezioni	id_corso, nome, nLezioni	id_corso
Agonista	Abbonato che partecipa alle gare	punteggio_gare, credito	
Lezione	Compone i corsi, vi partecipano solo i clienti occasionali	id_lezione, tipologia, durata, nome	id_lezione, id_corso (tramite Corso)
Gara	Evento sportivo a possono partecipare solo gli abbonati	ente_organizzatore, nome	nome, ente_organizzatore

3.5 Dizionario dei dati – Relazioni

Relazione	***Entità partecipanti***	***Descrizione***	***Attributi***
Titolarità Passata	Istruttore, Edizione Corso	I corsi di cui un istruttore è stato titolare in passato	
Titolare Di	Istruttore, Edizione Corso	I corsi di cui un istruttore è attualmente titolare	
Tipologia	Corso, Edizione Corso	Lega Corso ed Edizione Corso	
Iscritto	Abbonato, Edizione Corso	Memorizza le iscrizioni degli abbonati	
Composto Da	Corso, Lezione	Lega Corso e Lezione	
Partecipa Lezione	Cliente Occasionale, Lezione	Memorizza la partecipazione di un cliente occasionale ad una lezione	
Partecipazione Gara	Gara, Agonista	Gare a cui un agonista partecipa o ha partecipato nel recente passato	
Partecipazione Passata	Gara, Agonista	Partecipazioni passate	

3.6 Vincoli non esprimibili dallo schema E-R

- I clienti occasionali devono pagare sul momento le lezioni di cui usufruiscono.
- Solo gli abbonati possono possedere una tessera, in quanto iscritti alla palestra.

4.1 Tavola dei volumi

Si stimano 5 istruttori, una media di 2 corsi per istruttore all'anno, e una media di 20 corsi tenuti in passato per ciascun istruttore. Si stimano 1000 clienti, di cui 300 occasionali e 700 abbonati, di cui 300 agonisti, e una media di 5 iscrizioni a corsi per abbonato. Si stimano in media 10 partecipazioni a gare per agonista ed una media di 20 partecipazioni passate a gare. Infine si stimano 5 partecipazioni a lezioni da parte di clienti occasionali.

Concetto	Tipo	Volume
Cliente	E	1000
Cliente Occasionale	E	300
Abbonato	E	700
Agonista	E	300
Gara	E	50
Lezione	E	100
Corso	E	10
Edizione Corso	E	100
Istruttore	E	5
Titolare Di	R	2
Titolarità Passata	R	20
Iscritto	R	3500
Tipologia	R	100
Composto Da	R	100
Partecipazione Passata	R	6000
Partecipazione Gara	R	3000
Partecipa Lezione	R	1800

4.2 Tavola delle frequenze

Operazione	Descrizione	Frequenza	Tipo
O1	Addebita il costo degli abbonamenti ad ogni cliente	1 volta al mese	B
O2	Stampa gli istruttori con più corsi di cui sono attualmente titolari	1 volta ogni 3 mesi	B
O3	Calcola il credito derivante dal punteggio gara cumulato dall'agonista, poi azzera il punteggio	1 volta ogni 2 mesi	B
O4	Inserisci una nuova partecipazione a gare	1 volta ogni settimana	I
O5	Inserisci un nuovo abbonato	7 volta ogni giorno	I
O6	Stampa i dati di tutti gli abbonati	1 volta ogni 3 mesi	B
O7	Aggiorna il punteggio gara dell'agonista dopo l'inserimento di una nuova gara	1 volta a settimana	B
O8	Assegna un punteggio pari a 0 ad un nuovo cliente abbonato agonista	3 volte al giorno	B
O9	Controlla che solo gli agonisti risultino essere partecipanti alle gare, nel caso di abbonati non agonisti eliminare le loro partecipazioni	1 volta a settimana	B
O10	Sposta le partecipazioni a vecchie gare (quelle di cui si è già calcolato il credito del cliente) in un archivio apposito	1 volta ogni 2 mesi	B
O11	Stampa l'ammontare che ogni abbonato deve realmente pagare, cioè la differenza tra credito e debito	1 volta al mese	B

4.3 Analisi delle ridondanze

L'attributo nLezioni dell'entità Corso introduce della ridondanza, dato che il numero di lezioni per corso può essere conteggiato dalla relazione Composto Da. Se volessimo aggiungere una lezione con il relativo corso di appartenenza (operazione tuttavia non prevista), effettuiamo un'analisi sul numero di accessi:

Concetto	Tipo	Volume
Corso	E	10
Composto Da	R	100
Lezione	E	100

- con ridondanza:

Concetto	Costrutto	Accessi	Tipo
Lezione	Entità	1	S
Composto Da	Relazione	1	S
Corso	Entità	1	L
Corso	Entità	1	S

Supponendo che 1 S = 2 L, allora 3 S + 1 L = 7 L al mese

- senza ridondanza:

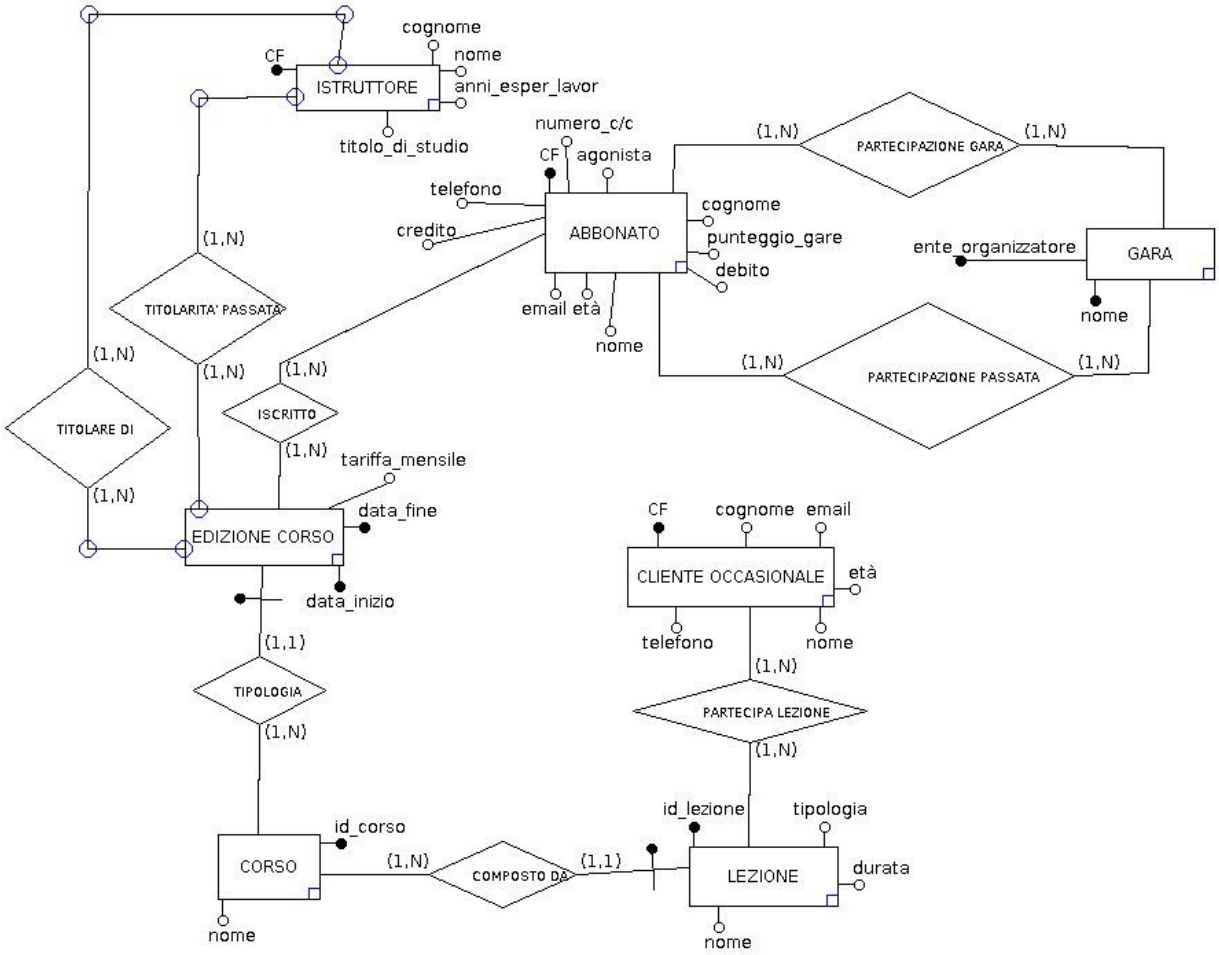
Concetto	Costrutto	Accessi	Tipo
Lezione	Entità	1	S
Composto Da	Relazione	1	S

In questo caso, abbiamo 2 S = 4 L al mese.

Sebbene la differenza nel numero di accessi e l'occupazione aggiuntiva in memoria in presenza di ridondanza è minima (4 bytes * 10 = 40 bytes), si decide comunque di eliminare la ridondanza.

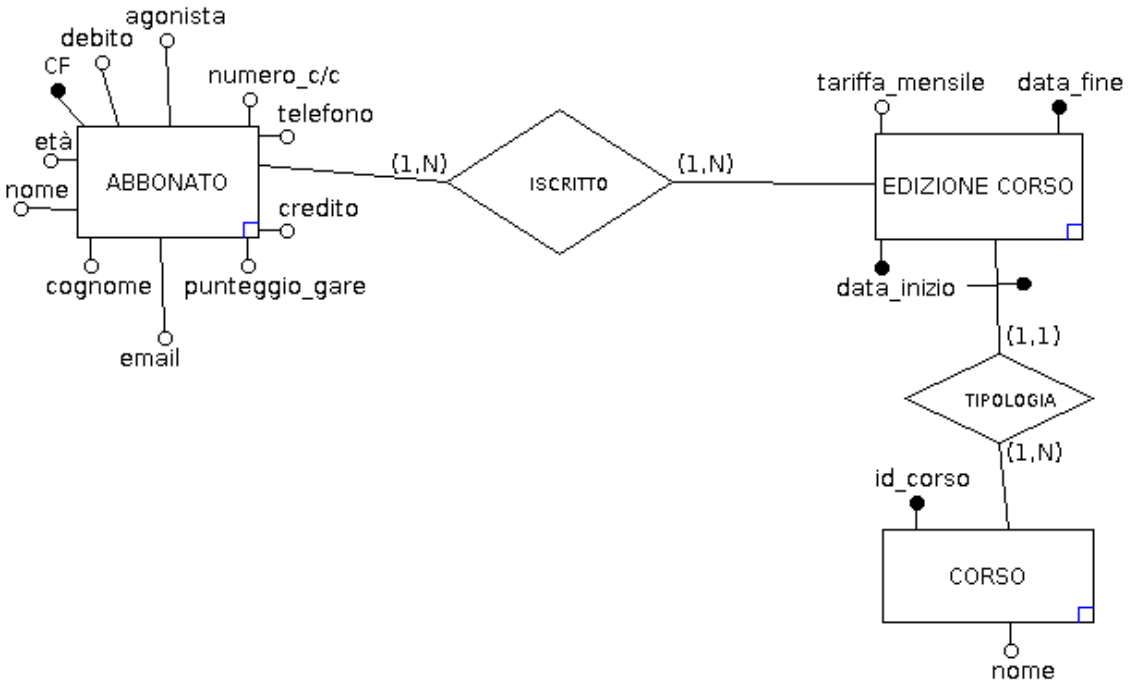
4.4 Eliminazione delle gerarchie, schema ristrutturato

Si decide di fare un collasso verso il basso per quanto riguarda la gerarchia composta dalle entità Cliente, Cliente Occasionale e Abbonato, dato che la gerarchia è totale ed esclusiva. L'entità Cliente viene quindi eliminata. La gerarchia non totale ed esclusiva, formata da Abbonato e Agonista, viene invece collassata verso l'alto, per cui Agonista viene eliminata e si introduce un attributo selettore nella entità Abbonato.

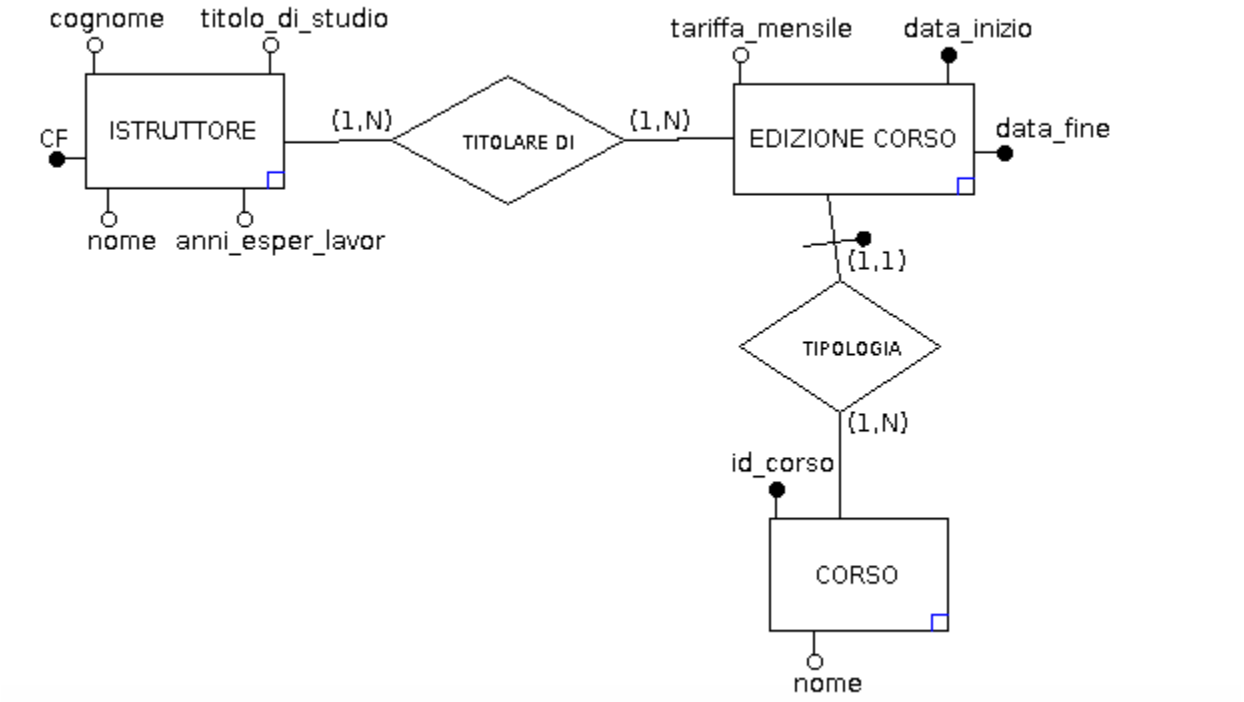


4.5 Schemi delle operazioni

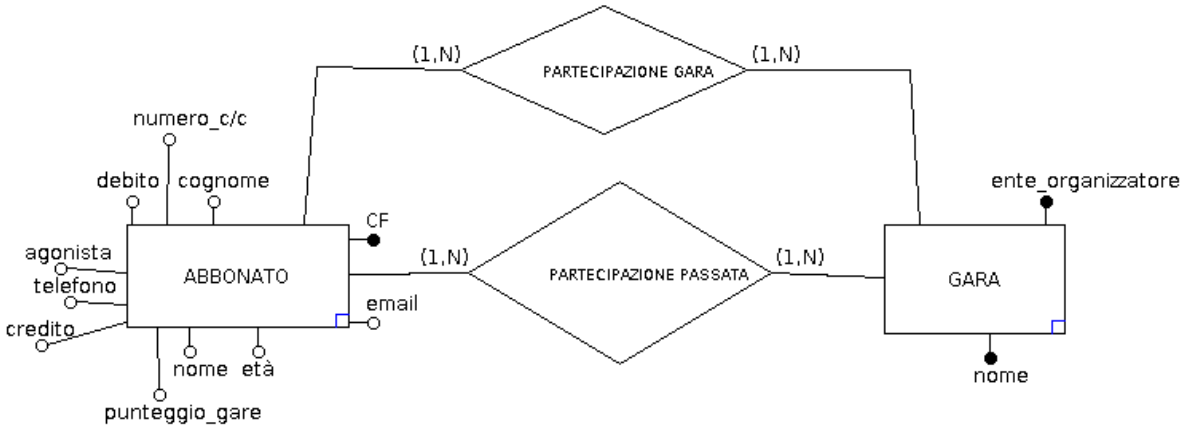
O1: Addebita il costo degli abbonamenti ad ogni cliente



O2: Stampa gli istruttori con più corsi di cui sono attualmente titolari



O10: Sposta le partecipazioni a vecchie gare (quelle di cui si è già calcolato il credito del cliente) in un archivio apposito



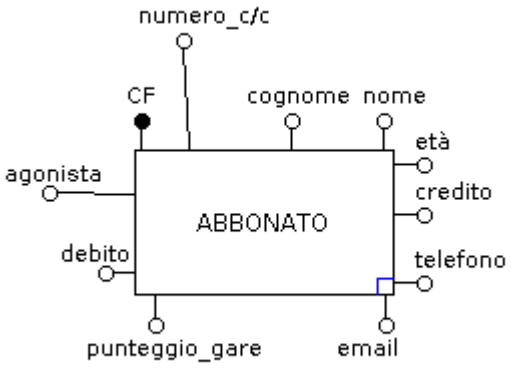
O3: Calcola il credito derivante dal punteggio gara cumulato dall'agonista, poi azzera il punteggio

O5: Inserisci un nuovo abbonato

O6: Stampa i dati di tutti gli abbonati

O8: Assegna un punteggio pari a 0 ad un nuovo cliente abbonato agonista

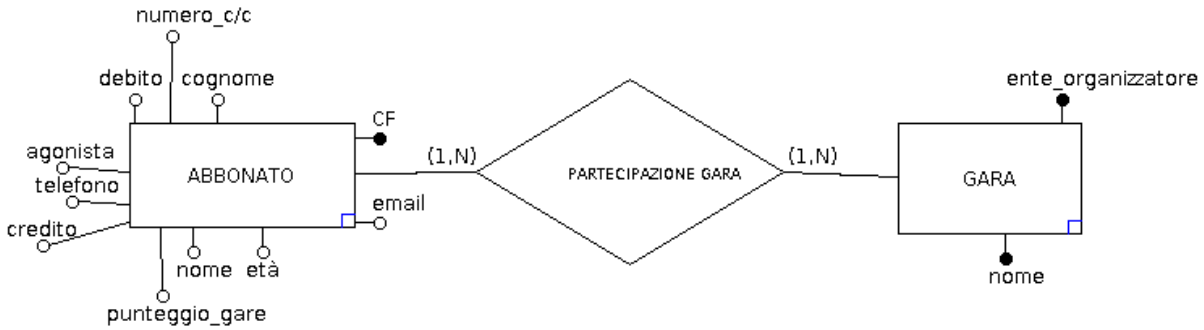
O11: Stampa l'ammontare che ogni abbonato deve realmente pagare, cioè la differenza tra credito e debito



O4: Inserisci una nuova partecipazione a gare

O7: Aggiorna il punteggio gara dell'agonista dopo l'inserimento di una nuova gara

O9: Controlla che solo gli agonisti risultino essere partecipanti alle gare, nel caso di abbonati non agonisti eliminare le loro partecipazioni



4.6 Traduzione nel modello relazionale

Le chiavi esterne sono indicate dalla sottolineatura tratteggiata, le chiavi primarie dalla sottolineatura continua, mentre gli attributi che giocano entrambi i ruoli sono indicati da una doppia sottolineatura continua.

Istruttore(CF, cognome, nome, titolo_di_studio, anni_esper_lavor)

TitolareDi(CF, data_fine, data_inizio, id_corso)

TitolaritaPassata(CF, data_fine, data_inizio, id_corso)

EdizioneCorso(id_corso, tariffa_mensile, data_fine, data_inizio)

Corso(id_corso, nome)

Lezione(id_lezione, id_corso, tipologia, durata, nome)

PartecipaLezione(id_lezione, id_corso, CF)

ClienteOccasionale(CF, cognome, email, eta, nome, telefono)

Abbonato(CF, numero_c/c, agonista, cognome, punteggio_gare, debito, nome, eta, email, credito, telefono)

Gara(ente_organizzatore, nome)

PartecipazioneGara(CF, ente_organizzatore, nome)

PartecipazionePassata(CF, ente_organizzatore, nome)

Iscritto(CF, id_corso, data_fine, data_inizio)

5.1 Implementazione SQL delle tabelle

```
CREATE TABLE Abbonato (
        CF varchar(16) NOT NULL,
        numero_c/c varchar(30) NOT NULL,
        agonista tinyint(1) NOT NULL,
        cognome varchar(70) NOT NULL,
        punteggio_gare int(11) NOT NULL,
        debito int(11) NOT NULL,
        nome varchar(70) NOT NULL,
        eta int(11) NOT NULL,
        email varchar(60) NOT NULL,
        credito int(11) NOT NULL,
        telefono varchar(20) NOT NULL,
        PRIMARY KEY (CF)
)



CREATE TABLE Gara (
        ente_organizzatore varchar(30) NOT NULL,
        nome varchar(30) NOT NULL,
        PRIMARY KEY (ente_organizzatore, nome)
)



CREATE TABLE Istruttore (
        CF varchar(16) NOT NULL,
        cognome varchar(70) NOT NULL,
        nome varchar(70) NOT NULL,
        titolo_di_studio varchar(40) NOT NULL,
        anni_esper_lavor int(11) NOT NULL,
        PRIMARY KEY (CF)
)



CREATE TABLE Corso (
        id_corso int(11) NOT NULL AUTO_INCREMENT,
        nome varchar(25) NOT NULL,
        PRIMARY KEY (id_corso)
)
```

```
CREATE TABLE Lezione (
	id_lezione int(11) NOT NULL AUTO_INCREMENT,
	id_corso int(11) NOT NULL,
	tipologia varchar(30) NOT NULL,
	durata int(11) NOT NULL,
	nome varchar(30) NOT NULL,
	PRIMARY KEY (id_lezione, id_corso),
	FOREIGN KEY (id_corso) REFERENCES Corso (id_corso)
)



CREATE TABLE ClienteOccasionale (
	CF varchar(16) NOT NULL,
	cognome varchar(70) NOT NULL,
	email varchar(60) NOT NULL,
	eta int(11) NOT NULL,
	nome varchar(70) NOT NULL,
	telefono varchar(20) NOT NULL,
	PRIMARY KEY (CF)
)



CREATE TABLE EdizioneCorso (
	id_corso int(11) NOT NULL AUTO_INCREMENT,
	data_fine date NOT NULL,
	data_inizio date NOT NULL,
	tariffa_mensile int(11) NOT NULL,
	PRIMARY KEY (id_corso, data_fine, data_inizio),
	FOREIGN KEY (id_corso) REFERENCES Corso (id_corso)
)



CREATE TABLE Iscritto (
	CF varchar(16) NOT NULL,
	id_corso int(11) NOT NULL,
	data_inizio date NOT NULL,
	data_fine date NOT NULL,
	PRIMARY KEY (CF, id_corso, data_inizio, data_fine),
	FOREIGN KEY (CF) REFERENCES Abbonato (CF),
	FOREIGN KEY (id_corso, data_fine, data_inizio)
			REFERENCES EdizioneCorso (id_corso, data_fine, data_inizio)
)
```

```
CREATE TABLE PartecipazioneGara (
        CF varchar(16) NOT NULL,
        ente_organizzatore varchar(30) NOT NULL,
        nome varchar(30) NOT NULL,
        PRIMARY KEY (CF, ente_organizzatore, nome),
        FOREIGN KEY (CF) REFERENCES Abbonato (CF),
        FOREIGN KEY (ente_organizzatore, nome)
                REFERENCES Gara (ente_organizzatore, nome)
)


CREATE TABLE PartecipazionePassata (
        CF varchar(16) NOT NULL,
        ente_organizzatore varchar(30) NOT NULL,
        nome varchar(30) NOT NULL,
        PRIMARY KEY (CF, ente_organizzatore, nome),
        FOREIGN KEY (CF) REFERENCES Abbonato (CF),
        FOREIGN KEY (ente_organizzatore, nome)
                REFERENCES Gara (ente_organizzatore, nome)
)


CREATE TABLE TitolareDi (
        CF varchar(30) NOT NULL,
        data_fine date NOT NULL,
        data_inizio date NOT NULL,
        id_corso int(11) NOT NULL,
        PRIMARY KEY (CF, data_fine, data_inizio, id_corso),
        FOREIGN KEY (CF) REFERENCES Istruttore (CF),
        FOREIGN KEY (id_corso, data_fine, data_inizio)
                REFERENCES EdizioneCorso (id_corso, data_fine, data_inizio)
)
```

```
CREATE TABLE TitolaritaPassata (
        CF varchar(30) NOT NULL,
        data_fine date NOT NULL,
        data_inizio date NOT NULL,
        id_corso int(11) NOT NULL,
        PRIMARY KEY (CF, data_fine, data_inizio, id_corso),
        FOREIGN KEY (CF) REFERENCES Istruttore (CF),
        FOREIGN KEY (id_corso, data_fine, data_inizio)
                REFERENCES EdizioneCorso (id_corso, data_fine, data_inizio)
)
```

5.2 Implementazione SQL delle operazioni

O1: Addebita il costo degli abbonamenti ad ogni cliente

```
CREATE VIEW DebitiAbbonati AS
SELECT CF, SUM(tariffa_mensile) AS ammontareDebito
FROM Iscritto I JOIN EdizioneCorso E
ON I.id_corso = E.id_corso AND I.data_fine = E.data_fine
                                AND I.data_inizio = E.data_inizio
GROUP BY CF;

UPDATE Abbonato
SET debito = debito + (SELECT ammontareDebito
                       FROM DebitiAbbonati
                       WHERE Abbonato.CF = DebitiAbbonati.CF
                      );
```

O2: Stampa gli istruttori con più corsi di cui sono attualmente titolari

```
CREATE VIEW nCorsiPerIstruttore AS
SELECT CF, COUNT(*) AS nCorsi
FROM TitolareDi
GROUP BY CF;

SELECT * FROM Istruttore NATURAL JOIN nCorsiPerIstruttore
WHERE nCorsi = (SELECT MAX(nCorsi)
                FROM nCorsiPerIstruttore
               );
```

O3: Calcola il credito derivante dal punteggio gara cumulato dall'agonista, poi azzera il punteggio

```
START TRANSACTION
UPDATE Abbonato
SET credito = credito + (punteggio_gare * 0.1)
WHERE agonista <> 0;

UPDATE Abbonato
SET punteggio_gare = 0 WHERE agonista <> 0;

COMMIT
```

O4: Inserisci una nuova partecipazione a gare

```
INSERT INTO PartecipazioneGara
VALUES ('FCVMTT91T13V150Y', 'Coni', 'Gara di Karate');
```

O5: Inserisci un nuovo abbonato

```
INSERT INTO Abbonato
VALUES ('FCVMTM99H28H169K', '00019315', 0, 'Scivoletto',
        NULL, 0, 'Mattia Maria', 21, 'prova@gmail.com', 0, '0123456789');
```

O6: Stampa i dati di tutti gli abbonati

```
SELECT * FROM Abbonato;
```

O7: Aggiorna il punteggio gara dell'agonista dopo l'inserimento di una nuova gara

```
CREATE TRIGGER AggiornaPunteggioGare
AFTER INSERT ON PartecipazioneGara
FOR EACH ROW
BEGIN
        DECLARE X NUMERIC;
        DECLARE Y VARCHAR(16);

        SELECT CF, COUNT(*) INTO X, Y
        FROM PartecipazioneGara
        WHERE CF = new.CF
        GROUP BY CF;

        UPDATE Abbonato SET punteggio_gare = FLOOR(X/3)
        WHERE CF = new.CF;
END;
```

O8: Assegna un punteggio pari a 0 ad un nuovo cliente abbonato agonista

```
CREATE TRIGGER AdeguaPunteggio
AFTER INSERT ON Abbonato
FOR EACH ROW
BEGIN
        UPDATE Abbonato
        SET punteggio = 0
        WHERE new.agonista <> 0;
END;
```

O9: Controlla che solo gli agonisti risultino essere partecipanti alle gare, nel caso di abbonati non agonisti eliminare le loro partecipazioni

```
CREATE TRIGGER ControllaPartecipazioneGare
AFTER INSERT ON PartecipazioneGara
FOR EACH ROW
BEGIN
        DELETE FROM PartecipazioneGara
        WHERE CF = (SELECT CF
                        FROM Abbonato
                        WHERE agonista = 0 AND Abbonato.CF = new.CF
                       )
END;
```

O10: Sposta le partecipazioni a vecchie gare (quelle di cui si è già calcolato il credito del cliente) in un archivio apposito

```
CREATE TRIGGER SpostaGare
AFTER UPDATE ON Abbonato
FOR EACH ROW
BEGIN
        INSERT INTO PartecipazionePassata
                        (SELECT * FROM PartecipazioneGara
                         WHERE CF = (SELECT CF
                                         FROM Abbonato
                                         WHERE new.credito <> old.credito
                                        )
                        );

        DELETE FROM PartecipazioneGara
        WHERE CF = (SELECT CF
                          FROM Abbonato
                          WHERE new.credito <> old.credito
                          );
END;
```

O11: Stampa l'ammontare che ogni abbonato deve realmente pagare, cioè la differenza tra credito e debito

```
SELECT CF, (debito – credito) AS daPagare
FROM Abbonato;
```