

Progetto Basi di Dati

Corso di laurea in Informatica

Reggiani Filippo

Mat.148084

Indice:

- Descrizione della realtà da analizzare
- Glossario
- Schema Scheletro
- Progettazione concettuale
- Diagramma ER completo
- Progettazione logica
- Eliminazione delle gerarchie ISA
- Selezione delle chiavi primarie ed eliminazione delle chiavi esterne
- Trasformazione degli attributi e/o composti
- Traduzioni di entità e associazioni in schema di relazioni
- Verifica della normalizzazione
- Vincoli aggiuntivi
- Query di creazione
- Query di inserimento
- Query di interrogazione
- Query di modifica
- Query di eliminazione

• Descrizione della realtà da analizzare:

Si vuole realizzare un database per la gestione di una palestra.

Si dovrà quindi modellare una realtà con gli abbonati alla palestra, i personal trainer e tutti gli aspetti relativi.

Il centro permette l'iscrizione ad una persona mediante la sottoscrizione di un abbonamento, ogni abbonamento relativo ad un abbonato è identificato attraverso il codice fiscale dell'abbonato.

Per ogni abbonamento è associata una data di inizio (la data di sottoscrizione) e una data di fine, dopo la quale non è più possibile accedere al centro.

Sono presenti diversi tipi di abbonamento in base alla durata di esso, ognuno avente prezzo diverso.

L'abbonato essendo una persona è identificato da un codice fiscale, ed è rappresentato da un nome, un cognome, una data nascita e un telefono, di cui ne possono essere lasciati anche due.

Per ogni abbonato vengono inoltre memorizzati l'email a cui verrà inviata la scheda, uno sconto nel caso in cui l'abbonato ne sia in possesso e un indirizzo.

Per sottoscrivere l'abbonamento al centro sportivo l'abbonato deve aver svolto una visita medica presso un medico rappresentato da un nome. La visita medica ha una data di scadenza, oltre la quale non sarà più possibile effettuare l'accesso al centro sportivo.

Le visite mediche sono contraddistinte dal nome del medico e un id della visita.

All'abbonato vengono assegnati un armadietto personale identificato da un numero e dalla sua locazione all'interno del centro ed una tessera per l'accesso al centro sportivo identificata da un numero univoco per abbonato.

Gli abbonati possono iscriversi ai corsi svolti dai trainer, ogni corso è caratterizzato da una tipologia che indica l'attività svolta e una descrizione per dettagli sul corso. I corsi sono identificati da un codice univoco.

I corsi si svolgono in sale che mette a disposizione il centro, ogni sala è identificata da un numero sala.

In un giorno e ad una certa ora si può svolgere solo un corso in una determinata sala.

I trainer sono anch'essi identificati dal codice fiscale e hanno un nome, un cognome, una data nascita e un telefono (fino a due).

Ogni trainer ha un certo numero di qualifiche, identificate da un codice e da una descrizione.

Per i trainer viene richiesto di memorizzare un giorno di chiusura settimanale, dove il trainer dovrà rendersi disponibile ad effettuare la chiusura della palestra.

I trainer del centro creano una scheda d'allenamento agli abbonati, ogni scheda è caratterizzata da una data di inizio e una data di fine validità ed è identificata da un id scheda.

Ogni abbonato ha associato un solo id_scheda.

Ogni scheda d'allenamento è composta da un diverso numero di esercizi, che l'abbonato dovrà andare a svolgere durante i suoi allenamenti.

Ogni esercizio va a far parte di un numero di allenamento, in quanto una scheda può avere più allenamenti.

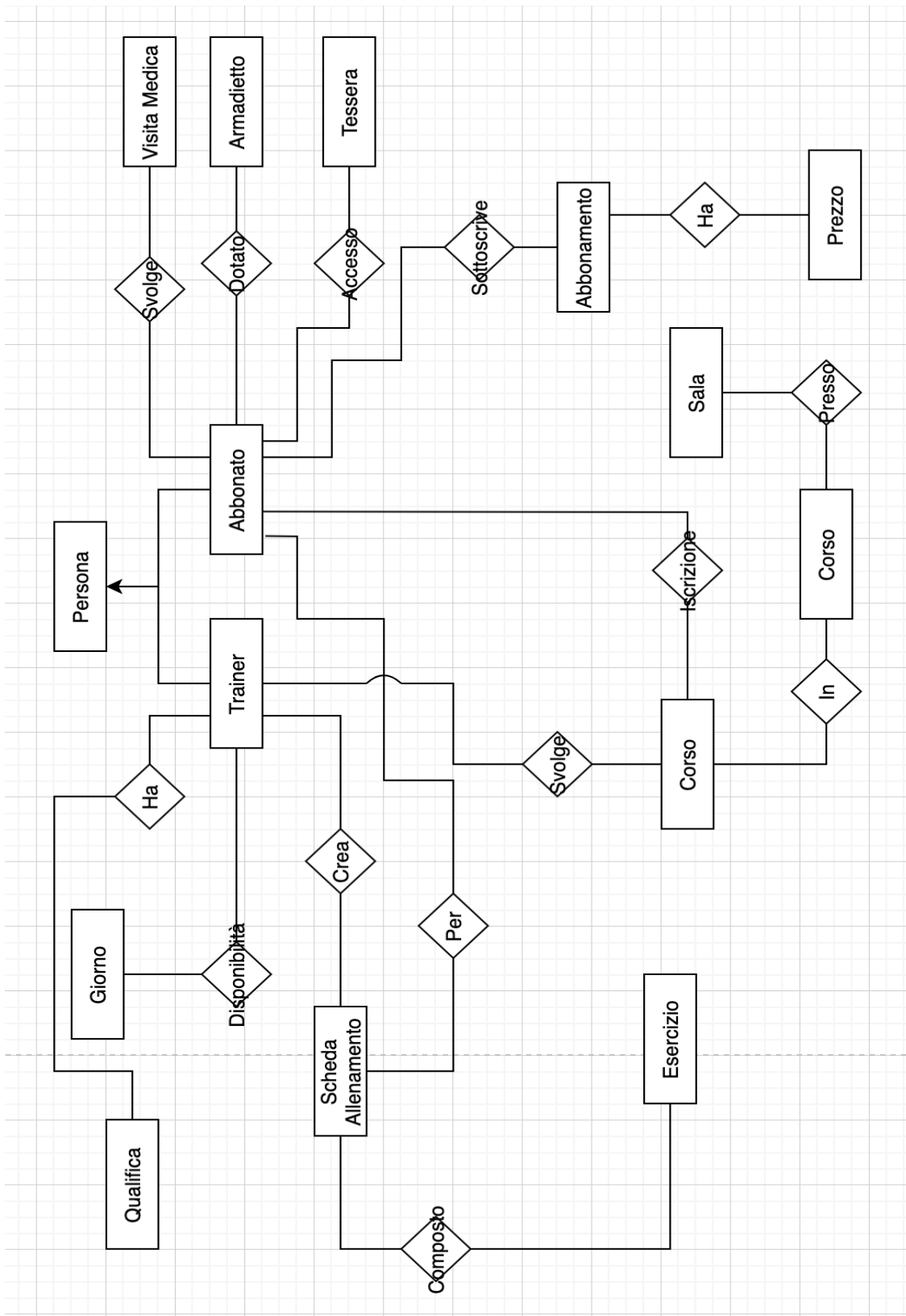
Per ogni esercizio sono importanti il numero di ripetizioni da effettuare, il recupero tra un esercizio e l'altro, il carico da impostare e il numero di serie ossia il numero di volte che l'esercizio dovrà essere ripetuto.

Ogni esercizio è identificato da un nome univoco dell'esercizio e dal muscolo che permette di allenare.

• **Glossario:**

Termine	Descrizione	Sinonimi	Legami
Persona	-Codice Fiscale -Nome -Cognome -Data nascita -Telefono		-Trainer -Abbonato
Abbonato	-Sconto -Email -Indirizzo	-Iscritto	-Abbonamento -Persona -Scheda Allenamento -Visita Medica -Armadietto -Tessera
Armadietto	-Numero -Locazione		-Abbonato
Tessera	-Numero		-Abbonato
Visita Medica	-Data scadenza -Id visita -Medico		-Abbonato
Trainer	-Qualifica		-Persona -Scheda allenamento -Corso -Chiusura Settimanale
Chiusura Settimanale	-Giorno		-Trainer
Scheda Allenamento	-Data inizio -Data fine -Id Scheda		-Esercizio -Abbonato -Trainer
Esercizio	-Nome -Muscolo		-Scheda Allenamento
Corso	-Descrizione -Tipologia -Codice		-Trainer -Abbonato -Orario
Abbonamento	-id -Data inizio -Data fine		-Abbonato
Orario	-Giorno -Ora		-Sala -Corso
Prezzo	-Tipologia -Costo		-Abbonamento
Sala	-Numero Sala		-Orario

- **Schema Scheletro:**



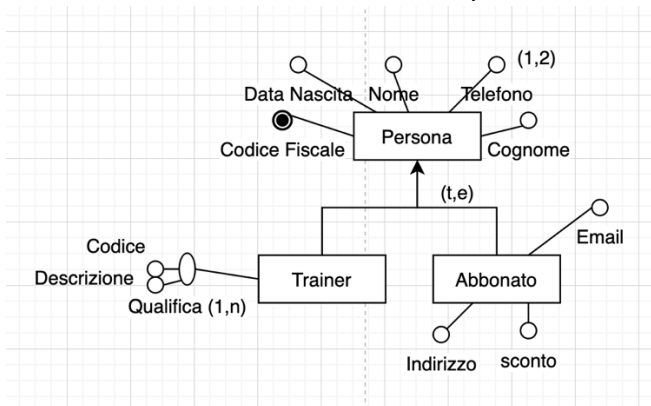
• Progettazione Concettuale:

Le persone sono caratterizzate da un nome, un cognome, una data di nascita e uno o due numeri di telefono. Le persone sono identificate mediante il proprio codice fiscale.

Una persona può essere o un abbonato oppure un personal trainer, non può essere ambedue le categorie, in quanto i trainer possono usufruire del centro sportivo senza dover sottoscrivere un abbonamento.

Un abbonato è contraddistinto dal codice fiscale. Per ogni abbonato viene memorizzato l'indirizzo e lo sconto nel caso in cui ne abbia uno, oltre ad una email.

Il trainer è distinto dal codice fiscale e ha una o più qualifiche, dove per ognuna di esse viene memorizzata una descrizione della qualifica e il codice di essa.



Ogni abbonato sostiene una visita medica identificata da un id visita e con una certa data di scadenza. Viene memorizzato anche il medico che effettua la visita.

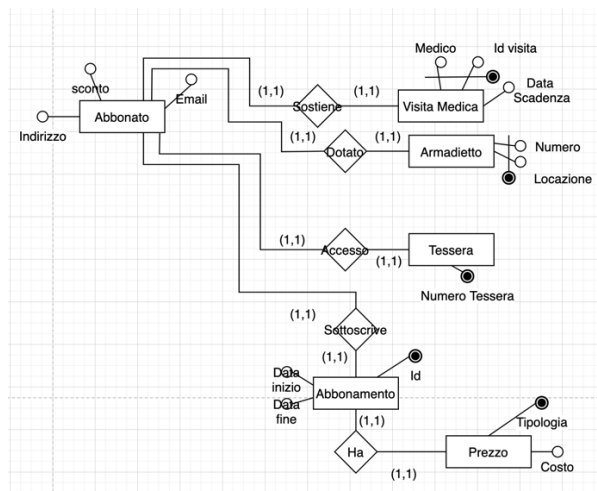
Per gli abbonati è riservato un solo armadietto, identificato dal numero e dalla propria locazione. L'armadietto è unico per abbonato.

Ogni abbonato ha una tessera univoca identificata da un numero di tessera unico.

Gli abbonati possono sottoscrivere un solo abbonamento.

Ogni abbonamento è contraddistinto da un id e ha una data di inizio e una data di fine abbonamento.

Un abbonamento ha un prezzo che dipende dalla tipologia di abbonamento scelto, per il quale è riservato un determinato costo.



Il trainer crea una o più schede di allenamento, identificate da un id scheda.

Per ogni scheda di allenamento sono memorizzate la data di inizio e la data di fine.

Una scheda di allenamento è destinata ad un solo abbonato e un abbonato può avere una sola scheda di allenamento.

Ogni scheda di allenamento è costituita da più esercizi, che fanno parte di diversi allenamenti, dove per ogni esercizio sono memorizzate le ripetizioni, il recupero, il carico e le serie.

Un esercizio può far parte di più schede di allenamento ma non si può avere lo stesso esercizio più volte nella stessa scheda.

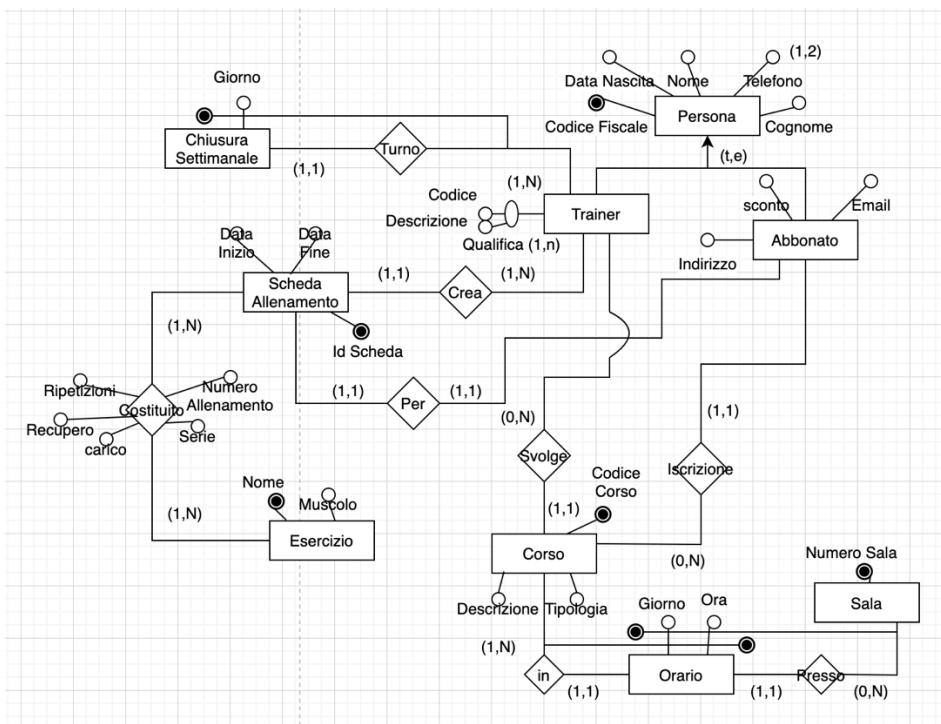
Un trainer può svolgere più corsi, ciascuno di essi identificato da un codice del corso e con una descrizione e una tipologia.

Un corso è svolto da un solo trainer.

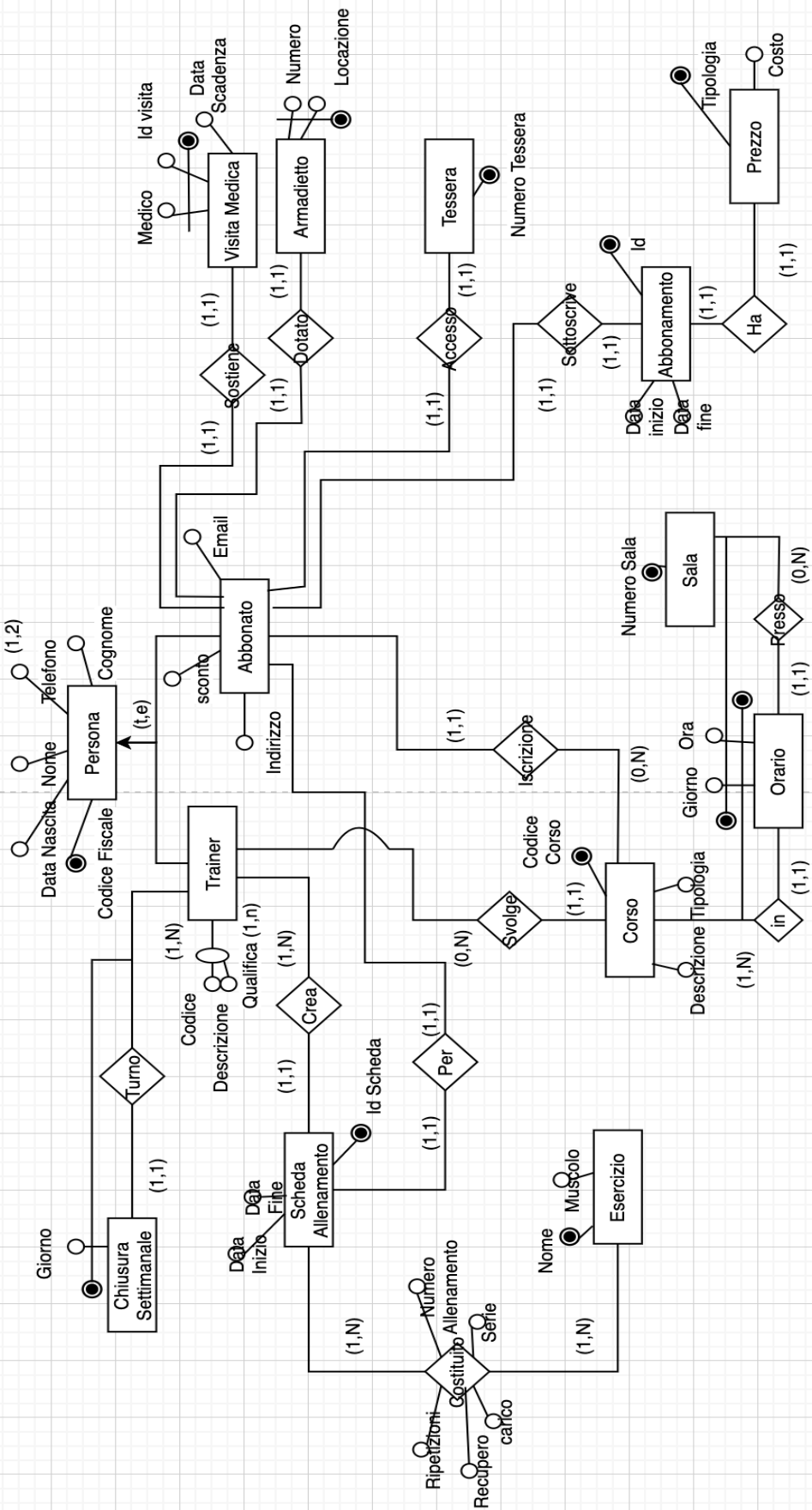
I corsi sono rivolti agli abbonati, che possono iscriversi ad essi.

Un corso si svolge in una specifica sala del centro sportivo, identificata da un numero e con una certa capienza.

Ad una certa ora e in un certo giorno, all'interno di una determinata sala, può essere svolto soltanto un corso.



- Diagramma ER completo



• Progettazione Logica

Non esistono DBMS in grado di operare sui concetti di schemi ER.

Occorre tradurre gli schemi ER in altri schemi di dati, come il logico-relazionale.

Uno schema ER viene tradotto in uno schema logico-relazionale attraverso le fasi:

1. Eliminazione delle gerarchie
2. Selezione chiavi primarie e eliminazione delle identificazioni esterne
3. Trasformazione attributi composti o multipli
4. Traduzione di entità e associazioni in schemi di relazione
5. Verifica di normalizzazione

Eliminazione delle Gerarchie:

Le gerarchie vengono sostituite da entità e associazioni, ci sono tre possibilità per effettuare la traduzione:

- Mantenimento delle entità:

Vengono mantenute tutte le entità, con le associazioni figlie che vengono messe in associazione con l'entità padre e che sono identificate esternamente attraverso l'associazione.

- Collasso verso l'alto:

Tutte le entità figlie vengono unite nel padre e la distinzione viene fatta da un attributo 'selettore' che cambia a seconda del tipo di gerarchia.

- Collasso verso il basso:

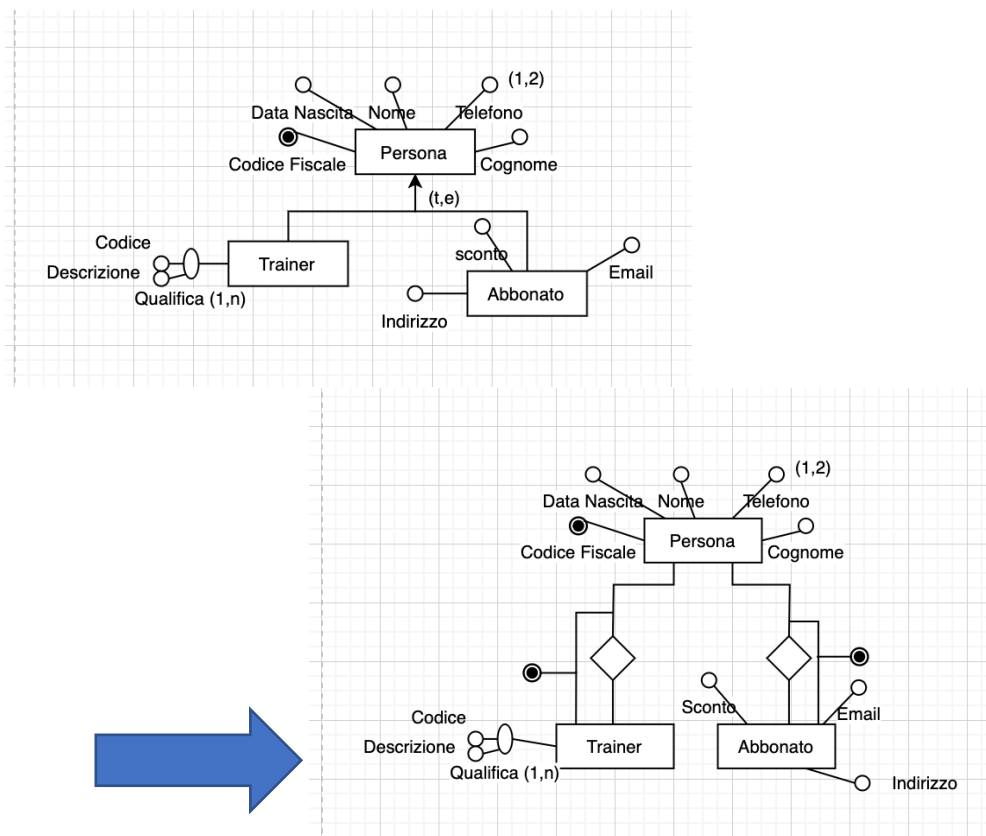
Viene eliminata l'entità padre trasferendo gli attributi su tutte le entità figlie e le associazioni con il padre vengono replicate, favorendo le operazioni in cui si accede separatamente alle entità figlie.

Si elimina la gerarchia di persona e di abbonato:

Per quanto riguarda la gerarchia di persona si effettua una traduzione che mantenga le entità, per tenerle distinte. Le entità trainer e abbonato sono quindi state messe in associazione con l'entità padre persona e identificate esternamente.

Si sarebbe potuto effettuare un collasso verso il basso in quanto la gerarchia è sia totale, quindi applicabile, che esclusiva e dunque non si rischia di avere ridondanze, ma si è preferito mantenere l'entità persona, evitando di copiare i suoi attributi sulle entità figlie.

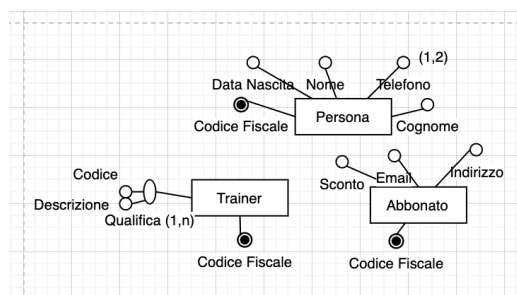
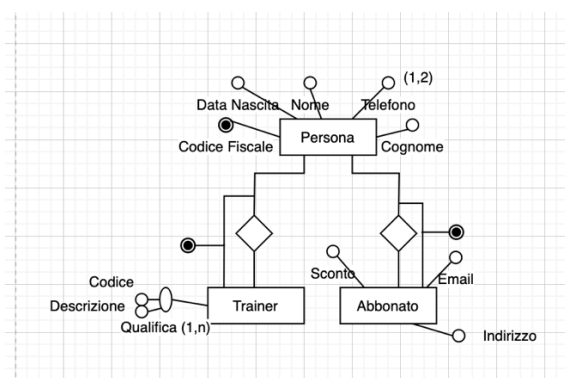
Si sarebbe potuto effettuare anche un collasso verso l'alto, ma si vogliono mantenere le due entità trainer e abbonato.



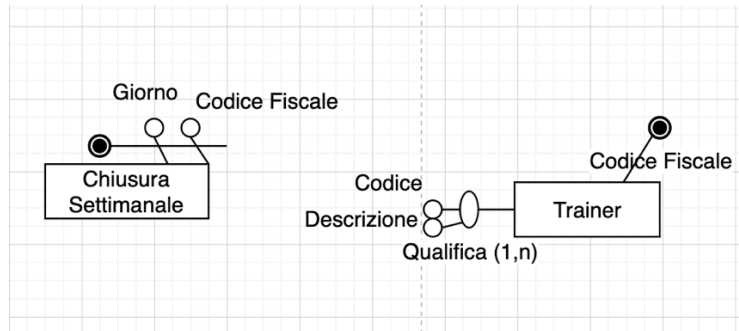
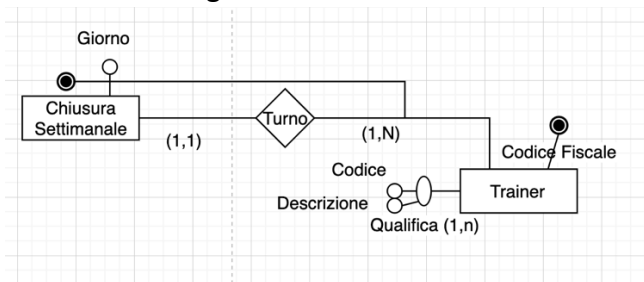
Selezione delle chiavi primarie ed eliminazione identificazioni esterne:

Dato che il DBMS deve garantire che i valori della chiave primaria siano unici, è necessario che ogni entità possenga una chiave primaria tra i suoi identificatori.

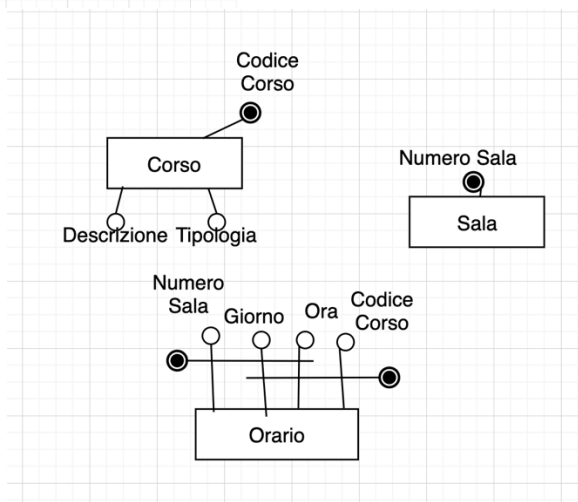
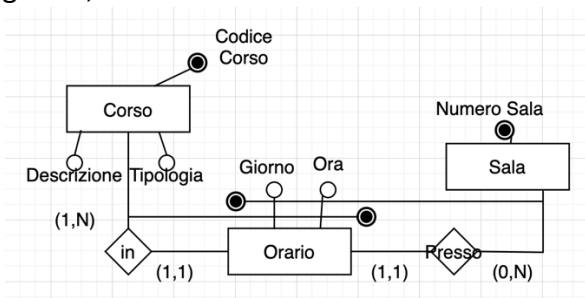
Occorre eliminare le identificazioni esterne delle entità trainer e abbonato dall'entità persona. A tali entità viene dunque inserito come attributo chiave il codice fiscale.



Occorre rimuovere l'identificazione esterna su chiusura settimanale, la cui chiave diventa la coppia codice fiscale e giorno.



Viene infine effettuata la traduzione delle identificazioni esterne su orario da parte di corso e sala. Orario diventa quindi un'entità avente come chiave la coppia giorno, ora e numero sala e la coppia giorno, ora e codice corso.



Trasformazione attributi composti e/o multipli:

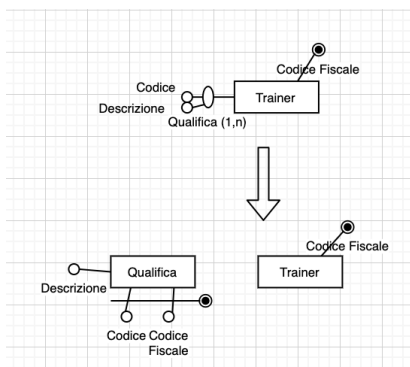
Le relazioni non possono contenere attributi composti o ripetuti ma solo atomici.

Per eliminare gli attributi composti abbiamo due soluzioni:

- Eliminare l'attributo composto e considerare i suoi componenti come attributi semplici, perdendo visione unitaria ma mantenendo l'articolazione dei componenti.
- Eliminare i componenti e considerare l'attributo come semplice, semplificando lo schema ma perdendo parte dei dettagli.

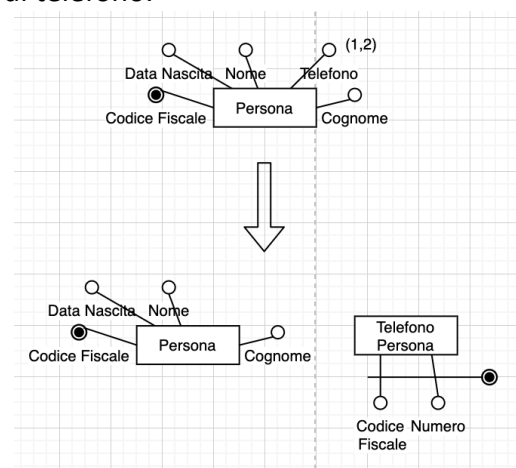
Per quanto riguarda gli attributi multipli invece si crea una nuova entità che contenga l'attributo e sia collegata a quella iniziale.

L'attributo multiplo qualifica dell'entità trainer viene convertito attraverso una nuova entità qualifica che contiene come chiave la coppia codice fiscale e codice della qualifica. La descrizione della qualifica diventa un attributo semplice dell'entità qualifica.



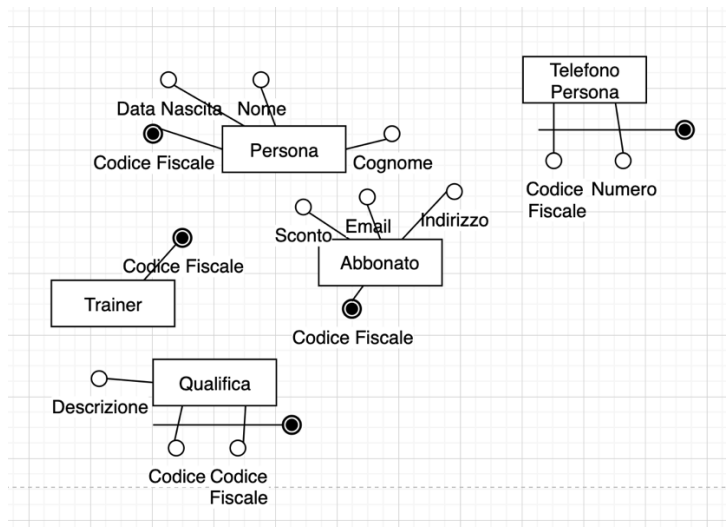
Per quanto riguarda l'entità persona occorre tradurre l'attributo multiplo telefono, che diventa una nuova entità telefono persona.

La nuova entità avrà come chiave la coppia tra codice fiscale e numero per identificare il numero di telefono.



Traduzione delle entità e delle associazioni:

Si traducono le entità che dalla traduzione precedente non hanno associazioni tra loro:



PERSONA (Codice_Fiscale, Data_Nascita, Nome, Cognome)

TELEFONO_PERSONA (CF_Persona, Numero)

FK: CF_Persona References PERSONA NOT NULL

TRAINER (Codice_Fiscale)

FK: Codice_Fiscale References PERSONA

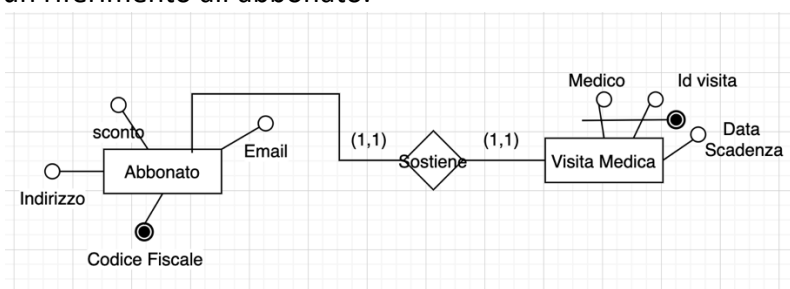
QUALIFICA (Codice_Fiscale_Trainer, Codice, Descrizione)

FK: Codice_Fiscale_Trainer references TRAINER

ABBONATO (Codice_Fiscale_Abbonato, Sconto, Indirizzo, Email)

FK: Codice_Fiscale_Abbonato References PERSONA

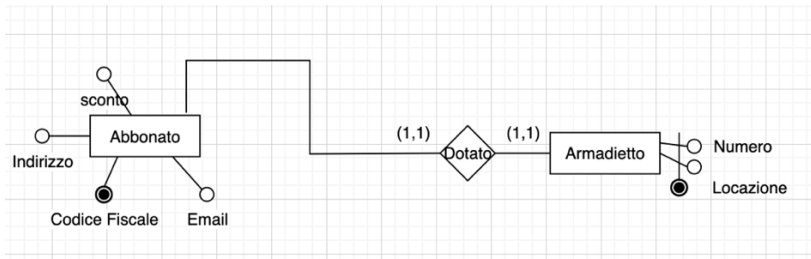
Per quanto riguarda l'associazione 1:1 tra Visita Medica e Abbonato viene aggiunto a visita medica un riferimento all'abbonato.



VISITA_MEDICA (Id_visita, Medico, Data_scadenza, Codice_Fiscale_Abbonato)

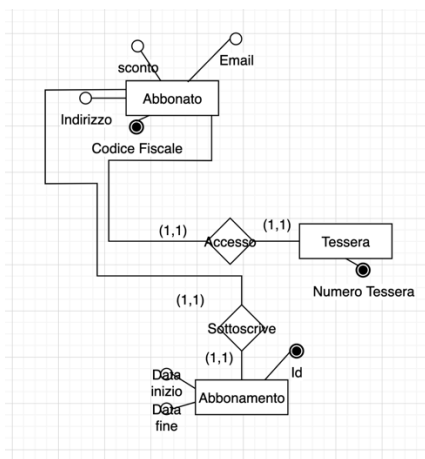
FK: Codice_Fiscale_Abbonato References ABBONATO

L'associazione 1:1 tra abbonato e armadietto viene tradotta con una relazione Armadietto che mantiene la sua chiave e ottiene come nuovo attributo un riferimento all'abbonato.



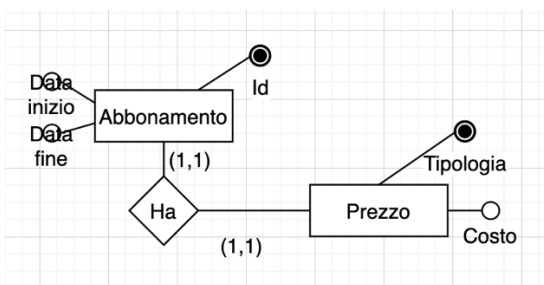
ARMADIETTO (Numero, Locazione, Codice_Fiscale_Abbonato)
FK: Codice_Fiscale_Abbonato References ABBONATO

L'associazione 1:1 tra abbonato e tessera viene tradotta mediante una relazione Tessera che avrà come chiave il Numero della tessera e avrà anche l'attributo riferito all'abbonato. Mentre per quanto riguarda l'associazione 1:1 tra abbonamento e abbonato viene aggiunto un riferimento di abbonato alla relazione abbonamento.



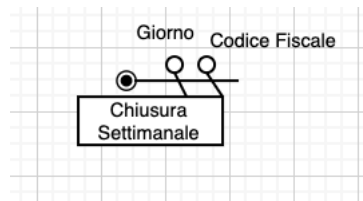
TESSERA (Numero Tessera, Codice_Fiscale_Abbonato)
FK: Codice_Fiscale_Abbonato References ABBONATO

L'associazione 1:1 tra abbonamento e prezzo viene tradotta mediante l'inserimento di un attributo riferito al prezzo all'interno di abbonamento.



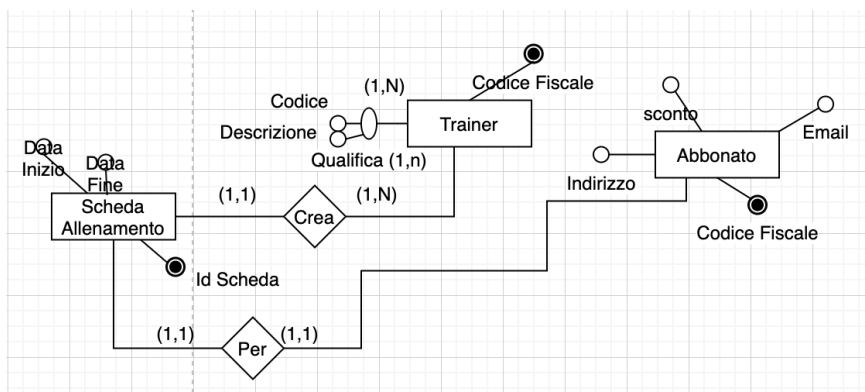
ABBONAMENTO (Id, Data_inizio, Data_fine, Codice_Fiscale_Abbonato, Tipologia)
FK: Codice_Fiscale_Abbonato References ABBONATO
PREZZO_ABBONAMENTO (Tipologia, Costo)
FK: Id_Abbonamento References ABBONAMENTO

Per quanto riguarda i collegamenti dell'entità trainer, si traduce in relazione l'entità chiusura settimanale tradotta in precedenza.



CHIUSURA_SETTIMANALE (Giorno, Codice Fiscale Trainer)
FK: Codice_Fiscale_Trainer References TRAINER

Per quanto riguarda la traduzione dell'associazione 1:N tra scheda_allenamento e trainer si inserisce all'interno della relazione scheda_allenamento il Codice Fiscale del trainer.
 Per l'associazione 1:1 tra abbonato e scheda invece si inserisce allo stesso modo all'interno di scheda_allenamento il Codice Fiscale dell'abbonato.

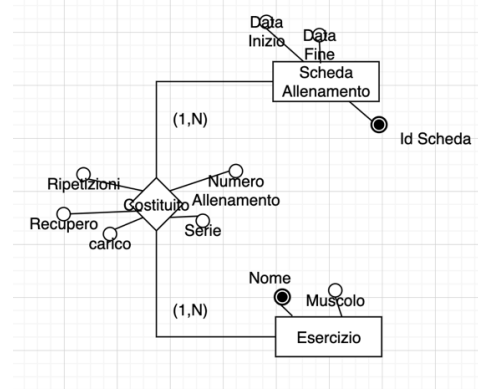


SCHEDA_ALLENAMENTO (Id_Scheda, Codice_Fiscale_Abbonato, Codice_Fiscale_Trainer, Data_inizio, Data Fine)
FK: Codice_Fiscale_Abbonato References ABBONATO
FK: Codice_Fiscale_Trainer References TRAINER

L'entità Scheda allenamento è in associazioni N:M con l'entità esercizio, si effettua dunque una traduzione standard creando 3 relazioni.

La relazione composizione_scheda avrà come chiave la coppia delle chiavi dell'entità Scheda_allenamento e dell'entità esercizio, oltre ai propri attributi.

La relazione esercizio invece avrà come chiave il nome univoco dell'esercizio.



COMPOSIZIONE_SCHEDA (Id_Scheda, Nome_Esercizio, Numer_Allenamento, Ripetizioni, , Serie, Recupero, Carico)

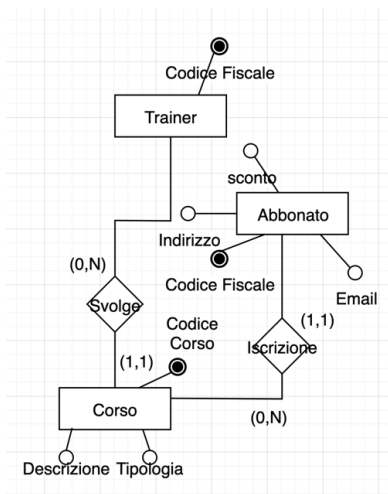
FK: Id_Scheda References SCHEDA_ALLENAMENTO

FK: Nome_Esercizio References ESERCIZIO

ESERCIZIO (Nome, Muscolo)

L'entità Corso è collegata all'entità Trainer con associazione 1:N, si incorpora in questo caso la chiave di Trainer in Corso creando la relazione Corso.

Per quanto riguarda l'associazione 1:N tra abbonato e corso, si crea una nuova relazione iscrizione_corso avente come chiave il codice fiscale dell'abbonato e come attributo il codice del corso.



CORSO (Codice_Corso, Codice_Fiscale_Trainer, Descrizione, Tipologia)

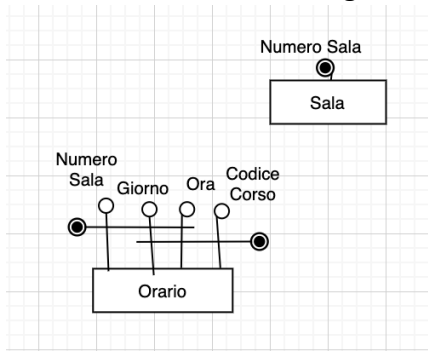
FK: Codice_Fiscale_Trainer References TRAINER

ISCRIZIONE_CORSO (Codice_Fiscale_Abbonato, Codice_Corso)

FK: Codice_Fiscale_Abbonato References ABBONATO

FK: Codice_Corso References CORSO

Le entità Orario e sala vengono infine tradotte in relazioni.



ORARIO (Numero_Sala, Giorno, Ora, Codice_Corso)

FK: Numero_Sala References SALA

FK: Codice_Corso References CORSO

AK: Codice_Corso, Giorno, Ora

SALA (Numero_Sala)

Le relazioni sono dunque:

PERSONA (Codice Fiscale, Data_Nascita, Nome, Cognome)

TELEFONO_PERSONA (CF_Persona, Numero)

FK: CF_Persona References PERSONA NOT NULL

TRAINER (Codice Fiscale)

FK: Codice_Fiscale References PERSONA

QUALIFICA (Codice Fiscale Trainer, Codice, Descrizione)

FK: Codice_Fiscale_Trainer references TRAINER

ABBONATO (Codice Fiscale Abbonato, Sconto, Indirizzo, Email)

FK: Codice_Fiscale_Abbonato References PERSONA

ARMADIETTO (Numero, Locazione, Codice_Fiscale_Abbonato)

FK: Codice_Fiscale_Abbonato References ABBONATO

TESSERA (Numero Tessera, Codice_Fiscale_Abbonato)

FK: Codice_Fiscale_Abbonato References ABBONATO

CHIUSURA_SETTIMANALE (Giorno, Codice_Fiscale_Trainer)

FK: Codice_Fiscale_Trainer References TRAINER

SCHEDA_ALLENAMENTO (Id_Scheda, Codice_Fiscale_Abbonato, Codice_Fiscale_Trainer,
Data_inizio, Data_Fine)

FK: Codice_Fiscale_Abbonato References ABBONATO

FK: Codice_Fiscale_Trainer References TRAINER

COMPOSIZIONE_SCHEDA (Id_Scheda, Nome_Esercizio, Numero_Allenamento, Ripetizioni,
Serie, Recupero, Carico)

FK: Id_Scheda References SCHEDA_ALLENAMENTO

FK: Nome_Esercizio References ESERCIZIO

ESERCIZIO (Nome, Muscolo)

CORSO (Codice_Corso, Codice_Fiscale_Trainer, Descrizione, Tipologia)

FK: Codice_Fiscale_Trainer References TRAINER

ISCRIZIONE_CORSO (Codice_Fiscale_Abbonato, Codice_Corso)

FK: Codice_Fiscale_Abbonato References ABBONATO

FK: Codice_Corso References CORSO

ORARIO (Numero_Sala, Giorno, Ora, Codice_Corso)

FK: Numero_Sala References SALA

FK: Codice_Corso References CORSO

AK: Codice_Corso, Giorno, Ora

SALA (Numero_Sala)

VISITA_MEDICA (Id_visita, Medico, Data_scadenza, Codice_Fiscale_Abbonato, Validità)

FK: Codice_Fiscale_Abbonato References ABBONATO

ABBONAMENTO (Id, Data_inizio, Data_fine, Codice_Fiscale_Abbonato, Tipologia)

FK: Codice_Fiscale_Abbonato References ABBONATO

PREZZO_ABBONAMENTO (Tipologia, Costo)

FK: Id_Abbonamento References ABBONAMENTO

Verifica della Normalizzazione:

Dopo una accurata analisi, non si rende necessaria, alcuna modifica.

Vincoli aggiuntivi:

Attraverso i trigger si impongono vincoli ulteriori, non gestibili direttamente dal database.

- Si impone il limite di un solo abbonamento per un abbonato, limitando per lo stesso abbonato di avere più di un abbonamento con un codice abbonamento diverso.

```
CREATE FUNCTION limite_abbonamenti() RETURNS trigger AS $limite_abbonamenti$
BEGIN
    IF (SELECT COUNT(*)
        FROM ABBONAMENTO AS ABBONAMENTI
        WHERE ABBONAMENTI.CF_Abbonato = NEW.CF_Abbonato)=2
    THEN
        RAISE EXCEPTION 'Questo abbonato ha già un abbonamento!';
    END IF;
    RETURN NEW;
END;
$limite_abbonamenti$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER LIMITE_ABBONAMENTI
AFTER INSERT OR UPDATE
ON ABBONAMENTO
FOR EACH ROW
EXECUTE PROCEDURE limite_abbonamenti();
```

- Si impone il limite di massimo due numeri di telefono per la stessa persona.

```
CREATE FUNCTION limite_telefoni() RETURNS trigger AS $limite_telefoni$
BEGIN
    IF (SELECT count(*)
        FROM TELEFONO_PERSONA AS TEL
        WHERE TEL.CF = NEW.CF)=3
    THEN
        RAISE EXCEPTION 'Questa persona ha già due numeri di telefono registrati';
    END IF;
    RETURN NEW;
END;
$limite_telefoni$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER LIMITE_TELEFONI
AFTER INSERT OR UPDATE
ON TELEFONO_PERSONA
FOR EACH ROW
EXECUTE PROCEDURE limite_telefoni();
```

- Si impone il limite di un solo armadietto per lo stesso abbonato.

```
CREATE FUNCTION limite_armadietti() RETURNS trigger AS $limite_armadietti$
BEGIN
    IF(SELECT COUNT(*)
       FROM ARMADIETTO AS ARMADIETTI
       WHERE ARMADIETTI.CF_Abbonato = NEW.CF_Abbonato)=2
    THEN
        RAISE EXCEPTION 'Questo abbonato ha già un armadietto!';
    END IF;
    RETURN NEW;
END;
$limite_armadietti$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER LIMITE_ARMADIETTI
AFTER INSERT OR UPDATE
ON ARMADIETTO
FOR EACH ROW
EXECUTE PROCEDURE limite_armadietti();
```

- Si impone il limite di una sola tessera per abbonato.

```
CREATE FUNCTION limite_tessera() RETURNS trigger AS $limite_tessera$
BEGIN
    IF(SELECT COUNT(*)
       FROM TESSERA AS TESSERE
       WHERE TESSERE.CF_Abbonato = NEW.CF_Abbonato)=2
    THEN
        RAISE EXCEPTION 'Questo abbonato ha già una tessera associata!';
    END IF;
    RETURN NEW;
END;
$limite_tessera$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER LIMITE_TESSERA
AFTER INSERT OR UPDATE
ON TESSERA
FOR EACH ROW
EXECUTE PROCEDURE limite_tessera();
```

- Si impone il limite di una sola scheda per abbonato. In questo modo l'abbonato non può avere più schede con id_scheda diverso.

```
CREATE FUNCTION limite_scheda() RETURNS trigger AS $limite_scheda$
BEGIN
    IF(SELECT COUNT(*)
       FROM SCHEDA_ALLENAMENTO AS SCHEDE
       WHERE SCHEDE.CF_Abbonato = NEW.CF_Abbonato)=2
```

```

        THEN
            RAISE EXCEPTION 'Questo abbonato ha già una scheda di
allenamento!';
        END IF;
    RETURN NEW;
END;
$limite_scheda$ LANGUAGE plpgsql;

```

```

CREATE TRIGGER LIMITE_SCHEDA
AFTER INSERT OR UPDATE
ON SCHEDA_ALLENAMENTO
FOR EACH ROW
EXECUTE PROCEDURE limite_scheda();

```

- Si impone di una singola visita registrata per abbonato, in questo modo un abbonato non ha più visite con id_visita diverso.

```

CREATE FUNCTION limite_visite() RETURNS trigger AS $limite_visite$
BEGIN
    IF(SELECT COUNT(*)
    FROM VISITA_MEDICA AS VISITE
    WHERE VISITE.CF_Abbonato = NEW.CF_Abbonato)=2
    THEN
        RAISE EXCEPTION 'Questo abbonato ha già una visita medica!';
    END IF;
    RETURN NEW;
END;
$limite_visite$ LANGUAGE plpgsql;

```

```

CREATE TRIGGER LIMITE_VISITE
AFTER INSERT OR UPDATE
ON VISITA_MEDICA
FOR EACH ROW
EXECUTE PROCEDURE limite_visite();

```

Query di Creazione:

Creazione delle tabelle:

```
CREATE TABLE PERSONA
(
    CF CHAR(16) NOT NULL,
    Nome VARCHAR(25) NOT NULL,
    Cognome VARCHAR(25) NOT NULL,
    Data_Nascita DATE NOT NULL,
    PRIMARY KEY (CF)
);

CREATE TABLE TELEFONO_PERSONA
(
    CF CHAR(16) NOT NULL,
    Numero CHAR(15) NOT NULL,
    PRIMARY KEY (CF,Numero),
    FOREIGN KEY (CF) REFERENCES PERSONA(CF)
    ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE TRAINER
(
    CF CHAR(16) NOT NULL,
    PRIMARY KEY(CF),
    FOREIGN KEY(CF) REFERENCES PERSONA(CF)
    ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE QUALIFICA
(
    CF CHAR(16) NOT NULL,
    Codice CHAR(20) NOT NULL,
    Descrizione VARCHAR(25),
    PRIMARY KEY( CF,Codice),
    FOREIGN KEY(CF) REFERENCES TRAINER(CF)
    ON DELETE CASCADE ON UPDATE CASCADE,
    CHECK
    ( Codice='101' AND Descrizione='Sala Pesi'
    OR Codice='102' AND Descrizione='Yoga'
    OR Codice='103' AND Descrizione='Boxe'
    )
);

CREATE TABLE ABBONATO
(
    CF CHAR(16) NOT NULL,
    Sconto INTEGER,
    Indirizzo VARCHAR(35) NOT NULL,
    Email VARCHAR(40) NOT NULL,
    PRIMARY KEY(CF),
    FOREIGN KEY(CF) REFERENCES PERSONA(CF)
```

ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE VISITA_MEDICA

(
 Id_Visita SERIAL NOT NULL,
 Medico CHAR(20) NOT NULL,
 Data_Scadenza DATE NOT NULL,
 CF_Abbonato CHAR(16) NOT NULL,
 PRIMARY KEY(Id_Visita,Medico),
 FOREIGN KEY(CF_Abbonato) REFERENCES ABBONATO(CF)
 ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE ARMADIETTO

(
 Numero SERIAL NOT NULL,
 Locazione VARCHAR(15) NOT NULL,
 CF_Abbonato CHAR(16) NOT NULL,
 PRIMARY KEY(Numero,Locazione),
 FOREIGN KEY(CF_Abbonato) REFERENCES ABBONATO(CF)
 ON DELETE CASCADE ON UPDATE CASCADE,
 CHECK
 (
 Locazione='Primo Piano'
 OR Locazione='Secondo Piano'
),
 CHECK
 (Numero>'0'
 AND Numero<'100'
)

);

CREATE TABLE TESSERA

(
 Numero_Tessera SERIAL NOT NULL,
 CF_Abbonato CHAR(16) NOT NULL,
 PRIMARY KEY(Numero_Tessera),
 FOREIGN KEY(CF_Abbonato) REFERENCES ABBONATO(CF)
 ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE PREZZO_ABBONAMENTO

(
 Tipologia CHAR(15) NOT NULL,
 Costo INTEGER NOT NULL,
 PRIMARY KEY(Tipologia)

);

CREATE TABLE ABBONAMENTO

(
 Id_Abbonamento SERIAL NOT NULL,

```

Data_Inizio DATE NOT NULL,
Data_Fine DATE NOT NULL,
CF_Abbonato CHAR(16) NOT NULL,
Tipologia CHAR(15) NOT NULL,
PRIMARY KEY(Id_Abbonamento),
FOREIGN KEY (CF_Abbonato) REFERENCES ABBONATO(CF)
ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (Tipologia) REFERENCES PREZZO_ABBONAMENTO(Tipologia)
ON DELETE CASCADE ON UPDATE CASCADE,
CHECK
( Data_Inizio < Data_Fine)
);

```

```

CREATE TABLE CHIUSURA_SETTIMANALE
(
    Giorno CHAR(9) NOT NULL,
    CF_Trainer CHAR(16) NOT NULL,
    PRIMARY KEY (Giorno,CF_Trainer),
    FOREIGN KEY(CF_Trainer) REFERENCES TRAINER(CF)
    ON DELETE CASCADE ON UPDATE CASCADE,
    CHECK
    (
        Giorno='Lunedì'
        OR Giorno='Martedì'
        OR Giorno='Mercoledì'
        OR Giorno='Giovedì'
        OR Giorno='Venerdì'
        OR Giorno='Sabato'
        OR Giorno='Domenica'
    )
);

```

```

CREATE TABLE SCHEDA_ALLENAMENTO
(
    CF_Abbonato CHAR(16) NOT NULL,
    Id_Scheda SERIAL UNIQUE NOT NULL,
    CF_Trainer CHAR(16) NOT NULL,
    Data_Inizio DATE NOT NULL,
    Data_Fine DATE NOT NULL,
    PRIMARY KEY(Id_Scheda),
    FOREIGN KEY(CF_Abbonato) REFERENCES ABBONATO(CF)
    ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY(CF_Trainer) REFERENCES TRAINER(CF)
    ON DELETE CASCADE ON UPDATE CASCADE,
    CHECK
    (
        Data_Fine > Data_Inizio
    )
);

```

```

CREATE TABLE ESERCIZIO
(
    Nome VARCHAR(35) NOT NULL,
    Muscolo VARCHAR(20) NOT NULL,

```

```

        PRIMARY KEY(Nome)
    );

CREATE TABLE COMPOSIZIONE_SCHEDA
(
    Id_Scheda SERIAL NOT NULL,
    Nome_Esercizio VARCHAR(35) NOT NULL,
    Numero_Allenamento INTEGER,
    Ripetizioni INTEGER NOT NULL,
    Serie INTEGER NOT NULL,
    Recupero INTEGER NOT NULL,
    Carico INTEGER NOT NULL,
    PRIMARY KEY( Id_Scheda,Nome_Esercizio),
    FOREIGN KEY( Id_Scheda) REFERENCES SCHEDA_ALLENAMENTO(Id_Scheda)
    ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (Nome_Esercizio) REFERENCES ESERCIZIO(Nome)
    ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE CORSO
(
    Codice_corso SERIAL NOT NULL,
    CF_Trainer CHAR(16) NOT NULL,
    Descrizione VARCHAR(50) NOT NULL,
    Tipologia VARCHAR(30) NOT NULL,
    PRIMARY KEY( Codice_Corso),
    FOREIGN KEY(CF_Trainer) REFERENCES TRAINER(CF)
    ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE ISCRIZIONE_CORSO
(
    CF_Abbonato CHAR(16) NOT NULL,
    Codice_Corso SERIAL NOT NULL,
    PRIMARY KEY( CF_Abbonato,Codice_Corso),
    FOREIGN KEY(CF_Abbonato) REFERENCES ABBONATO(CF)
    ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY(Codice_Corso) REFERENCES CORSO(Codice_Corso)
    ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE SALA
(
    Numero_Sala INTEGER NOT NULL,
    PRIMARY KEY(Numero_Sala)
);

```

```

CREATE TABLE ORARIO
(
    Numero_Sala INTEGER NOT NULL,
    Giorno CHAR(9) NOT NULL,
    Ora TIME NOT NULL,
    Codice_Corso SERIAL NOT NULL,

```

```
PRIMARY KEY(Numero_Sala,Giorno,Ora),
FOREIGN KEY(Numero_Sala) REFERENCES SALA(Numero_Sala)
ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY(Codice_Corso) REFERENCES CORSO(Codice_Corso)
ON DELETE CASCADE ON UPDATE CASCADE,
UNIQUE(Codice_Corso,Giorno,Ora),
CHECK
(
    Giorno='Lunedì'
    OR Giorno='Martedì'
    OR Giorno='Mercoledì'
    OR Giorno='Giovedì'
    OR Giorno='Venerdì'
    OR Giorno='Sabato'
    OR Giorno='Domenica'
)
```

```
);
```


Query di Inserimento:

Dati d'esempio per la popolazione delle tabelle.

Popolamento tabella PERSONA:

```
INSERT INTO PERSONA VALUES ('RGGFPP99T30F257C','Filippo','Reggiani','1999-12-30');
INSERT INTO PERSONA VALUES ('RGGJNF99T70F257R','Jennifer','Reggiani','1999-12-30');
INSERT INTO PERSONA VALUES ('MRARSS97E24F257H','Mario','Rossi','1997-05-24');
INSERT INTO PERSONA VALUES ('DBRFCS87P63A944Q','Debora','Fracassini','1987-09-23');
INSERT INTO PERSONA VALUES ('TLMFNC85M30B819C','Francesco','Tolomei','1985-08-30');
INSERT INTO PERSONA VALUES ('TRSMRC90B12F257U','Marco','tresi','1990-02-12');
INSERT INTO PERSONA VALUES ('FRNLRA77P63B539R','Lara','Francesi','1977-09-23');
INSERT INTO PERSONA VALUES ('BTTFNC89A01I110V','Franco','Battiato','1989-01-01');
INSERT INTO PERSONA VALUES ('MCHFNC99T49A959Y','Francesca','Micheli','1999-12-09');
INSERT INTO PERSONA VALUES ('CNLSFO96E50A713E','Sofia','Canali','1996-05-10');
INSERT INTO PERSONA VALUES ('PRTGPP87S09I802T','Giuseppe','Prato','1987-11-09');
INSERT INTO PERSONA VALUES ('MRCGRG01L08F205E','Giorgio','Marconi','2001-07-08');
INSERT INTO PERSONA VALUES ('TDDRST94B07L331F','Ernesto','Taddei','1994-02-07');
INSERT INTO PERSONA VALUES ('GNSMTT85B09I754N','Matteo','Ganassi','1985-02-09');
INSERT INTO PERSONA VALUES ('SNSFRC02H49F257Z','Federica','Senesi','2002-06-09');
```

Popolamento tabella TELEFONO_PERSONA:

```
INSERT INTO TELEFONO_PERSONA VALUES ('RGGFPP99T30F257C','3396788250');
INSERT INTO TELEFONO_PERSONA VALUES ('RGGJNF99T70F257R','3420789356');
INSERT INTO TELEFONO_PERSONA VALUES ('MRARSS97E24F257H','3391267930');
INSERT INTO TELEFONO_PERSONA VALUES ('DBRFCS87P63A944Q','3668901447');
INSERT INTO TELEFONO_PERSONA VALUES ('TLMFNC85M30B819C','3657890678');
INSERT INTO TELEFONO_PERSONA VALUES ('TRSMRC90B12F257U','3380997678');
INSERT INTO TELEFONO_PERSONA VALUES ('FRNLRA77P63B539R','3668997117');
INSERT INTO TELEFONO_PERSONA VALUES ('BTTFNC89A01I110V','5490087362');
INSERT INTO TELEFONO_PERSONA VALUES ('MCHFNC99T49A959Y','3390768876');
INSERT INTO TELEFONO_PERSONA VALUES ('CNLSFO96E50A713E','3422577690');
INSERT INTO TELEFONO_PERSONA VALUES ('PRTGPP87S09I802T','3457899655');
INSERT INTO TELEFONO_PERSONA VALUES ('MRCGRG01L08F205E','3468897016');
INSERT INTO TELEFONO_PERSONA VALUES ('TDDRST94B07L331F','3421898768');
INSERT INTO TELEFONO_PERSONA VALUES ('GNSMTT85B09I754N','3669067884');
INSERT INTO TELEFONO_PERSONA VALUES ('SNSFRC02H49F257Z','3398754234');
INSERT INTO TELEFONO_PERSONA VALUES ('RGGFPP99T30F257C','0596789190');
```

Popolamento tabella TRAINER:

```
INSERT INTO TRAINER VALUES('DBRFCS87P63A944Q');
INSERT INTO TRAINER VALUES('TLMFNC85M30B819C');
INSERT INTO TRAINER VALUES('GNSMTT85B09I754N');
INSERT INTO TRAINER VALUES('PRTGPP87S09I802T');
INSERT INTO TRAINER VALUES('CNLSFO96E50A713E');
```

Popolamento tabella QUALIFICA:

```
INSERT INTO QUALIFICA VALUES('DBRFCS87P63A944Q','101','Sala Pesi');
INSERT INTO QUALIFICA VALUES('TLMFNC85M30B819C','101','Sala Pesi');
INSERT INTO QUALIFICA VALUES('GNSMTT85B09I754N','102','Yoga');
INSERT INTO QUALIFICA VALUES('PRTGPP87S09I802T','103','Boxe');
INSERT INTO QUALIFICA VALUES('CNLSFO96E50A713E','101','Sala Pesi');
```

Popolamento tabella ABBONATO:

```
INSERT INTO ABBONATO VALUES ('RGGFPP99T30F257C','0','via marconi
17','filipporeggiani@live.it');
INSERT INTO ABBONATO VALUES ('RGGJNF99T70F257R','10','via marco polo
12','jenniferreggiani@virgilio.it');
INSERT INTO ABBONATO VALUES ('MRARSS97E24F257H','0','via aadige
11','mariorossi1@gmail.com');
INSERT INTO ABBONATO VALUES ('TRSMRC90B12F257U','0','via caravaggio
100','tresi.marco@me.com');
INSERT INTO ABBONATO VALUES ('FRNLRA77P63B539R','0','via firenzuola
1','francesilaraa@hotmail.com');
INSERT INTO ABBONATO VALUES ('BTTFNC89A01I110V','10','via alighieri
89','BattiatoFranco9@gmail.com');
INSERT INTO ABBONATO VALUES ('MCHFNC99T49A959Y','10','via michelangelo
190','michelifrenci09@live.it');
INSERT INTO ABBONATO VALUES ('MRCGRG01L08F205E','0','via grandi
99','marconigiorgio01@yahoo.com');
INSERT INTO ABBONATO VALUES ('TDDRST94B07L331F','10','via remo
12','ernestotaddei@gmail.com');
INSERT INTO ABBONATO VALUES ('SNSFRC02H49F257Z','0','via da vinci
10','fedesernesi@hotmail.com');
```

Popolamento tabella VISITA_MEDICA:

```
INSERT INTO VISITA_MEDICA VALUES('145079','Poretti','7/6/21','RGGFPP99T30F257C');
INSERT INTO VISITA_MEDICA VALUES('132897','Lenzotti','2/2/22','RGGJNF99T70F257R');
INSERT INTO VISITA_MEDICA VALUES('239076','Danieli','2/4/21','MRARSS97E24F257H');
INSERT INTO VISITA_MEDICA VALUES('124560','Poretti','8/2/22','TRSMRC90B12F257U');
INSERT INTO VISITA_MEDICA VALUES('102321','Prato','7/2/21','FRNLRA77P63B539R');
INSERT INTO VISITA_MEDICA VALUES('456710','Prato','5/8/22','BTTFNC89A01I110V');
INSERT INTO VISITA_MEDICA VALUES('278910','Castellino','7/9/22','MCHFNC99T49A959Y');
INSERT INTO VISITA_MEDICA VALUES('290132','Prato','9/2/22','MRCGRG01L08F205E');
INSERT INTO VISITA_MEDICA VALUES('890723','Castellino','5/2/22','TDDRST94B07L331F');
INSERT INTO VISITA_MEDICA VALUES('146723','Danieli','8/8/22','SNSFRC02H49F257Z');
```

Popolamento tabella ARMADIETTO:

```
INSERT INTO ARMADIETTO VALUES ('1','Primo Piano','RGGFPP99T30F257C');
INSERT INTO ARMADIETTO VALUES ('2','Primo Piano','RGGJNF99T70F257R');
INSERT INTO ARMADIETTO VALUES ('3','Primo Piano','MRARSS97E24F257H');
INSERT INTO ARMADIETTO VALUES ('4','Primo Piano','TRSMRC90B12F257U');
INSERT INTO ARMADIETTO VALUES ('1','Secondo Piano','FRNLRA77P63B539R');
INSERT INTO ARMADIETTO VALUES ('2','Secondo Piano','BTTFNC89A01I110V');
INSERT INTO ARMADIETTO VALUES ('5','Primo Piano','MCHFNC99T49A959Y');
INSERT INTO ARMADIETTO VALUES ('3','Secondo Piano','MRCGRG01L08F205E');
INSERT INTO ARMADIETTO VALUES ('4','Secondo Piano','TDDRST94B07L331F');
INSERT INTO ARMADIETTO VALUES ('5','Secondo Piano','SNSFRC02H49F257Z');
```

Popolamento tabella TESSERA:

```
INSERT INTO TESSERA VALUES('3001','RGGFPP99T30F257C');
INSERT INTO TESSERA VALUES('3002','RGGJNF99T70F257R');
INSERT INTO TESSERA VALUES('3003','MRARSS97E24F257H');
INSERT INTO TESSERA VALUES('3004','TRSMRC90B12F257U');
INSERT INTO TESSERA VALUES('3005','FRNLRA77P63B539R');
INSERT INTO TESSERA VALUES('3006','BTTFNC89A01I110V');
INSERT INTO TESSERA VALUES('3007','MCHFNC99T49A959Y');
INSERT INTO TESSERA VALUES('3008','MRCGRG01L08F205E');
INSERT INTO TESSERA VALUES('3009','TDDRST94B07L331F');
INSERT INTO TESSERA VALUES('3010','SNSFRC02H49F257Z');
```

Popolamento tabella PREZZO_ABBONAMENTO:

```
INSERT INTO PREZZO_ABBONAMENTO VALUES('Annuale','480');
INSERT INTO PREZZO_ABBONAMENTO VALUES('Trimestrale','180');
INSERT INTO PREZZO_ABBONAMENTO VALUES('Mensile','80');
```

Popolamento tabella ABBONAMENTO:

```
INSERT INTO ABBONAMENTO VALUES('1001','2020-02-13','2020-03-13','RGGFPP99T30F257C','Mensile');
INSERT INTO ABBONAMENTO VALUES('1002','2020-04-10','2021-04-10','RGGJNF99T70F257R','Annuale');
INSERT INTO ABBONAMENTO VALUES('1003','2020-05-12','2020-06-12','MRARSS97E24F257H','Mensile');
INSERT INTO ABBONAMENTO VALUES('1004','2020-02-01','2021-02-01','TRSMRC90B12F257U','Annuale');
INSERT INTO ABBONAMENTO VALUES('1005','2020-07-04','2020-10-04','FRNLRA77P63B539R','Trimestrale');
INSERT INTO ABBONAMENTO VALUES('1006','2020-08-06','2021-08-06','BTTFNC89A01I110V','Annuale');
INSERT INTO ABBONAMENTO VALUES('1007','2020-09-07','2021-09-07','MCHFNC99T49A959Y','Annuale');
```

```
INSERT INTO ABBONAMENTO VALUES('1008','2020-02-04','2020-05-04','MRCGRG01L08F205E','Trimestrale');
INSERT INTO ABBONAMENTO VALUES('1009','2020-11-03','2020-12-03','TDDRST94B07L331F','Mensile');
INSERT INTO ABBONAMENTO VALUES('1010','2020-04-02','2020-07-02','SNSFRC02H49F257Z','Trimestrale');
```

Popolamento tabella CHIUSURA_SETTIMANALE:

```
INSERT INTO CHIUSURA_SETTIMANALE VALUES('Lunedì','DBRFCS87P63A944Q');
INSERT INTO CHIUSURA_SETTIMANALE VALUES('Martedì','TLMFNC85M30B819C');
INSERT INTO CHIUSURA_SETTIMANALE VALUES('Mercoledì','DBRFCS87P63A944Q');
INSERT INTO CHIUSURA_SETTIMANALE VALUES('Giovedì','GNSMTT85B09I754N');
INSERT INTO CHIUSURA_SETTIMANALE VALUES('Venerdì','PRTGPP87S09I802T');
INSERT INTO CHIUSURA_SETTIMANALE VALUES('Sabato','CNLSFO96E50A713E');
INSERT INTO CHIUSURA_SETTIMANALE VALUES('Domenica','CNLSFO96E50A713E');
INSERT INTO CHIUSURA_SETTIMANALE VALUES('Mercoledì','TLMFNC85M30B819C');
```

Popolamento tabella SCHEDA_ALLENAMENTO:

```
INSERT INTO SCHEDA_ALLENAMENTO
VALUES('RGGFPP99T30F257C','1001','DBRFCS87P63A944Q','2020-03-01','2020-06-01');
INSERT INTO SCHEDA_ALLENAMENTO
VALUES('RGGJNF99T70F257R','1002','DBRFCS87P63A944Q','2020-01-07','2020-04-15');
INSERT INTO SCHEDA_ALLENAMENTO
VALUES('MRARSS97E24F257H','1003','TLMFNC85M30B819C','2020-04-01','2020-07-03');
INSERT INTO SCHEDA_ALLENAMENTO
VALUES('TRSMRC90B12F257U','1004','DBRFCS87P63A944Q','2020-06-07','2020-09-15');
INSERT INTO SCHEDA_ALLENAMENTO
VALUES('FRNLRA77P63B539R','1005','CNLSFO96E50A713E','2020-09-01','2020-12-08');
INSERT INTO SCHEDA_ALLENAMENTO
VALUES('BTTFNC89A01I110V','1006','DBRFCS87P63A944Q','2020-05-10','2020-08-19');
INSERT INTO SCHEDA_ALLENAMENTO
VALUES('MCHFNC99T49A959Y','1007','CNLSFO96E50A713E','2020-04-20','2020-07-26');
INSERT INTO SCHEDA_ALLENAMENTO
VALUES('MRCGRG01L08F205E','1008','TLMFNC85M30B819C','2020-08-01','2020-11-09');
INSERT INTO SCHEDA_ALLENAMENTO
VALUES('TDDRST94B07L331F','1009','CNLSFO96E50A713E','2020-01-17','2020-04-17');
INSERT INTO SCHEDA_ALLENAMENTO
VALUES('SNSFRC02H49F257Z','1010','DBRFCS87P63A944Q','2020-08-10','2020-11-08');
```

Popolamento tabella ESERCIZIO:

```
INSERT INTO ESERCIZIO VALUES('Leg Curl','Gambe');
INSERT INTO ESERCIZIO VALUES('Leg Press','Gambe');
INSERT INTO ESERCIZIO VALUES('Leg Extension','Gambe');
INSERT INTO ESERCIZIO VALUES('Curl Manubri in Piedi','Bicipiti');
INSERT INTO ESERCIZIO VALUES('Curl Manubri Seduto','Bicipiti');
```

```

INSERT INTO ESERCIZIO VALUES('Curl Bilanciere','Bicipiti');
INSERT INTO ESERCIZIO VALUES('Rematore Manubrio','Dorso');
INSERT INTO ESERCIZIO VALUES('Dead Lift','Dorso');
INSERT INTO ESERCIZIO VALUES('Rematore Bilanciere','Dorso');
INSERT INTO ESERCIZIO VALUES('Lat Machine','Dorso');
INSERT INTO ESERCIZIO VALUES('Distensione Manubri Panca Piana','Petto');
INSERT INTO ESERCIZIO VALUES('Crocchi Panca Piana','Petto');
INSERT INTO ESERCIZIO VALUES('Piegamenti sulle Braccia','Petto');
INSERT INTO ESERCIZIO VALUES('French Press','Tricipiti');
INSERT INTO ESERCIZIO VALUES('Push Down Corde','Tricipiti');
INSERT INTO ESERCIZIO VALUES('Dips su Panca','Tricipiti');
INSERT INTO ESERCIZIO VALUES('Alzate Lateralali','Spalle');
INSERT INTO ESERCIZIO VALUES('Harnold Press','Spalle');
INSERT INTO ESERCIZIO VALUES('Alzate Lateralali Seduto','Spalle');
INSERT INTO ESERCIZIO VALUES('Crunch Base','Addome');
INSERT INTO ESERCIZIO VALUES('Crunch Gambe a Tavolino','Addome');
INSERT INTO ESERCIZIO VALUES('Crunch Obliquo','Addome');
INSERT INTO ESERCIZIO VALUES('Crunch Gomito Ginocchio','Addome');
INSERT INTO ESERCIZIO VALUES('Criss Cross','Addome');

```

Popolamento tabella COMPOSIZIONE_SCHEDA:

```

/* Inserimento scheda di allenamento di un abbonato con 2 allenamenti*/
INSERT INTO COMPOSIZIONE_SCHEDA VALUES('1001','Rematore Manubrio','1','12','3','60','10');
INSERT INTO COMPOSIZIONE_SCHEDA VALUES('1001','Dead Lift','1','10','4','60','15');
INSERT INTO COMPOSIZIONE_SCHEDA VALUES('1001','Rematore Bilanciere','1','12','3','60','15');
INSERT INTO COMPOSIZIONE_SCHEDA VALUES('1001','Curl Manubri in Piedi','1','10','3','60','10');
INSERT INTO COMPOSIZIONE_SCHEDA VALUES('1001','Curl Bilanciere','1','8','4','60','8');
INSERT INTO COMPOSIZIONE_SCHEDA VALUES('1001','Crunch Base','1','10','3','60','0');
INSERT INTO COMPOSIZIONE_SCHEDA VALUES('1001','Crunch Obliquo','1','10','4','60','0');
INSERT INTO COMPOSIZIONE_SCHEDA VALUES('1001','Distensione Manubri Panca
Piana','2','12','3','60','10');
INSERT INTO COMPOSIZIONE_SCHEDA VALUES('1001','Crocchi Panca Piana','2','10','4','60','15');
INSERT INTO COMPOSIZIONE_SCHEDA VALUES('1001','Piegamenti sulle
Braccia','2','12','3','60','15');
INSERT INTO COMPOSIZIONE_SCHEDA VALUES('1001','French Press','2','10','3','60','10');
INSERT INTO COMPOSIZIONE_SCHEDA VALUES('1001','Dips su Panca','2','8','4','60','8');
INSERT INTO COMPOSIZIONE_SCHEDA VALUES('1001','Crunch Gomito
Ginocchio','2','10','3','60','0');
INSERT INTO COMPOSIZIONE_SCHEDA VALUES('1001','Criss Cross','2','10','4','60','0');
/* Inserimento scheda allenamento di un abbonato con un allenamento*/
INSERT INTO COMPOSIZIONE_SCHEDA VALUES('1002','Leg Press','1','12','3','60','100');
INSERT INTO COMPOSIZIONE_SCHEDA VALUES('1002','Leg Curl','1','10','4','60','55');
INSERT INTO COMPOSIZIONE_SCHEDA VALUES('1002','Leg Extension','1','12','3','60','45');
INSERT INTO COMPOSIZIONE_SCHEDA VALUES('1002','Alzate Lateralali','1','10','3','60','8');
INSERT INTO COMPOSIZIONE_SCHEDA VALUES('1002','Harnold Press','1','8','4','60','6');
INSERT INTO COMPOSIZIONE_SCHEDA VALUES('1002','Crunch Gambe a
Tavolino','1','10','3','60','0');

```

INSERT INTO COMPOSIZIONE_SCHEDA VALUES('1002','Crunch Obliquo','1','10','4','60','0');

Popolamento tabella CORSO:

INSERT INTO CORSO VALUES('5000','GNSMTT85B09I754N','Relax attraverso lo yoga','yoga');
INSERT INTO CORSO VALUES('5001','PRTGPP87S09I802T','Incontri di allenamento tra i partecipanti','boxe');

Popolamento tabella ISCRIZIONE_CORSO:

INSERT INTO ISCRIZIONE_CORSO VALUES('RGGFPP99T30F257C','5000');
INSERT INTO ISCRIZIONE_CORSO VALUES('MRARSS97E24F257H','5001');
INSERT INTO ISCRIZIONE_CORSO VALUES('FRNLRA77P63B539R','5001');
INSERT INTO ISCRIZIONE_CORSO VALUES('MCHFNC99T49A959Y','5001');

Popolamento tabella SALA:

INSERT INTO SALA VALUES('55');
INSERT INTO SALA VALUES('44');
INSERT INTO SALA VALUES('30');
INSERT INTO SALA VALUES('25');
INSERT INTO SALA VALUES('10');

Popolamento tabella ORARIO:

INSERT INTO ORARIO VALUES('55','Martedì','15:00','5000');
INSERT INTO ORARIO VALUES('44','Mercoledì','14:00','5001');
INSERT INTO ORARIO VALUES('25','Giovedì','19:00','5000');
INSERT INTO ORARIO VALUES('10','Lunedì','10:00','5001');
INSERT INTO ORARIO VALUES('25','Giovedì','17:00','5000');
INSERT INTO ORARIO VALUES('10','Lunedì','8:00','5001');

Query di Interrogazione:

1. Visualizzare la data di scadenza dell'abbonamento riferito all'abbonato con CF='RGGFPP99T30F257C'

```
SELECT ABB.Data_Fine
FROM ABBONAMENTO AS ABB, ABBONATO AS ABT, PERSONA AS PRS
WHERE ABB.CF_Abbonato='RGGFPP99T30F257C'
AND ABB.CF_Abbonato= ABT.CF
AND ABT.CF= PRS.CF
```

2. Visualizzare il nome e cognome degli abbonati che hanno un abbonamento scaduto (data fine abbonamento < data attuale)

```
SELECT PRS.Nome, Prs.Cognome
FROM PERSONA AS PRS, ABBONATO AS ABT, ABBONAMENTO AS ABB
WHERE PRS.CF= ABB.CF_Abbonato
AND ABB.CF_Abbonato=ABT.CF
AND ABB.Data_Fine > NOW()
```

3. Visualizzare numero di sala, orario e giorno in cui il trainer di nome 'Matteo Ganassi' dovrà svolgere il suo corso

```
SELECT ORA.Numero_Sala, ORA.Giorno, ORA.Ora
FROM ORARIO AS ORA, CORSO AS CRS
WHERE ORA.Codice_Corso =CRS.Codice_Corso
AND CRS.Codice_Corso =
    (
        SELECT CRS.Codice_Corso
        FROM CORSO AS CRS,TRAINER AS TR,PERSONA AS PRS
        WHERE ((PRS.Nome='Matteo' AND PRS.Cognome='Ganassi'))
        AND PRS.CF=TR.CF
        AND CRS.CF_Trainer=TR.CF
    )
```

4. Visualizzare nome e cognome dei trainer che hanno la qualifica per la Sala Pesi

```
SELECT PRS.Nome,PRS.Cognome
FROM PERSONA AS PRS, QUALIFICA AS QU
WHERE Descrizione='Sala Pesi'
AND PRS.CF= QU.CF
```

5. Visualizzare gli esercizi della scheda d'allenamento (solo il primo allenamento della scheda) dell'abbonato di nome 'Filippo Reggiani'

```
SELECT CS.Nome_Esercizio
FROM COMPOSIZIONE_SCHEDA AS CS
WHERE CS.Numero_Allenamento='1'
```

```

AND CS.Id_Scheda=
    (
        SELECT SA.Id_Scheda
        FROM PERSONA AS PRS, ABBONATO AS ABB, SCHEDA_ALLENAMENTO AS SA
        WHERE PRS.Nome='Filippo' AND PRS.Cognome='Reggiani'
        AND PRS.CF= ABB.CF
        AND ABB.CF=SA.CF_Abbonato
    )

```

6. Selezionare il codice fiscale degli abbonati che non hanno un abbonamento di tipo annuale

```

SELECT ABB.CF_Abbonato
FROM ABBONAMENTO AS ABB
WHERE ABB.Tipologia NOT IN
    (SELECT ABB.Tipologia
     FROM ABBONAMENTO AS ABB
     WHERE ABB.Tipologia='Annuale')

```

7. Selezionare l'identificativo delle schede fatte dal trainer 'Tolomei Francesco' e relativa data di scadenza

```

SELECT SA.Id_Scheda, SA.Data_fine
FROM PERSONA AS PRS, SCHEDA_ALLENAMENTO AS SA
WHERE PRS.Nome='Francesco' AND PRS.Cognome='Tolomei'
AND PRS.CF=SA.CF_Trainer

```

8. Selezionare il Prezzo e il tipo dell'abbonamento più costoso

```

SELECT PA.Tipologia, PA.Costo
FROM PREZZO_ABBONAMENTO AS PA
WHERE PA.Costo >= ALL
    (SELECT Costo
     FROM PREZZO_ABBONAMENTO)

```

9. Selezionare il nome e cognome degli abbonati che devono svolgere l'esercizio 'Crunch Base' nel primo allenamento della loro scheda

```

SELECT PRS.Nome, PRS.Cognome
FROM PERSONA AS PRS
WHERE PRS.CF IN(
    SELECT SA.CF_Abbonato
    FROM COMPOSIZIONE_SCHEDA AS CS, SCHEDA_ALLENAMENTO AS SA
    WHERE CS.Nome_esercizio='Crunch Base' AND CS.Numero_Allenamento='1'
    AND CS.Id_Scheda=SA.Id_Scheda
)

```


10. Selezionare l'email degli abbonati che hanno uno sconto tra il 5 e il 10%

```
SELECT ABB.Email  
FROM ABBONATO AS ABB  
WHERE ABB.Sconto BETWEEN '5' AND '10'
```

Query di Modifica:

1. Modificare il numero di telefono della persona con CF='FRNLRA77P63B539R' da '3668997117' a '339151987'

UPDATE TELEFONO_PERSONA
SET Numero='339151987'
WHERE CF='FRNLRA77P63B539R'

2. Modificare l'indirizzo di residenza dell'abbonato avente come mail 'ernestotaddei@gmail.com' da 'via remo 12' a 'via Petrarca 79'

UPDATE ABBONATO
SET Indirizzo='via Petrarca 79'
WHERE Email='ernestotaddei@gmail.com'

3. Modificare l'abbonamento con id='1001' avente data inizio='2020-02-13', data fine='2020-03-13' e di tipologia='Mensile' con un nuovo abbonamento avente data inizio='2021-07-15', data fine='2022-07-15' e tipologia='Annuale'

UPDATE ABBONAMENTO
SET Data_inizio='2021-07-15', Data_fine='2022-07-15', Tipologia='Annuale'
WHERE Id_Abbonamento='1001'

4. Modificare la scheda di allenamento con id='1001' avente data inizio='2020-03-01' e data fine='2020-06-01' con una nuova data di inizio='2021-03-04' e data_fine='2021-07-10'

UPDATE SCHEDA_ALLENAMENTO
SET Data_inizio='2021-03-04', Data_fine='2021-07-10'
WHERE Id_Scheda='1001'

5. Modificare l'orario e la sala del corso con codice corso='5000' che si sarebbe tenuto il giorno Martedì, alle ore 15:00 e nella sala 55, cambiando la sala con la 44, e il nuovo orario alle 18:00

UPDATE ORARIO
SET Numero_Sala='44', Ora='18:00'
WHERE Codice_corso='5000' AND Giorno='Martedì' AND Ora='15:00' AND Numero_Sala='55'

Query di Eliminazione:

1. Eliminazione del/dei numeri di telefono relativi alla persona con CF='RGGFPP99T30F257C'

DELETE FROM TELEFONO_PERSONA
WHERE CF='RGGFPP99T30F257C'

2. Eliminazione dell'orario del corso che si svolge nella sala 44, il giorno Mercoledì alle 14:00

DELETE FROM ORARIO
WHERE Numero_Sala='44' AND Giorno='Mercoledì' AND Ora='14:00'

3. Eliminazione dalla scheda con id='1002' dell'esercizio 'Crunch Obliquo'

DELETE FROM COMPOSIZIONE_SCHEDA
WHERE Id_Scheda='1002' AND Nome_Esercizio='Crunch Obliquo'

4. Eliminazione della chiusura effettuata Mercoledì del trainer con CF='DBRFCS87P63A944Q'

DELETE FROM CHIUSURA_SETTIMANALE
WHERE Giorno='Mercoledì' AND CF_Trainer='DBRFCS87P63A944Q'