MyPAL

1.Abstract

“MyGym” è un’applicazione , creata durante il periodo covid ,per il tracciamento degli allenamenti dei clienti di una catena di palestre che conta circa 3000 utenti. Con “MyGym” è possibile prenotare la propria sessione di allenamento, selezionando giorno, fascia oraria e sala; è inoltre possibile prenotare la partecipazione ad un corso organizzato dalla palestra. L’utente per registrarsi nell’applicazione dovrà inserire i propri dati personali ed una volta registrato potrà prenotare le proprie sessioni di allenamento e lasciare recensioni alla palestra o ai corsi organizzati da questa, appoggiandosi a dei siti esterni. Ogni utente iscritto alla palestra avrà una sua scheda di allenamento, che sarà possibile vedere direttamente dall’app, ma che sarà modificabile solo dal trainer che l’ha creata.

2 Analisi Requisiti

2.1 Descrizione testuale

Nella base di dati sono presenti i dati delle varie **Palestre** della catena. Ognuno di essi deve fornire come informazioni:

- Codice
- Indirizzo (Città, Via, numero civico)
- Numero di telefono
- Orario d’Apertura
- Orario di Chiusura
- Media Voto

Per ognuna di esse saranno note informazioni riguardanti il tipo di **Sale** che sono presenti in esse e gli **Attrezzi** a disposizione in ognuno di queste. Per ogni tipo di attrezzo sarà presente un codice che lo identifica univocamente, la quantità presente nella sala e, nel caso di manubri o simili anche il peso.

Dei **Clienti** abbonati alla catena dovranno essere noti :

- Numero di Telefono
- Nome
- Cognome
- Sesso
- Età

Ognuno di loro possiede una **Scheda** contenente informazioni sul tipo di esercizi che stanno eseguendo, in particolare si vuole sapere : Durata, Difficoltà e i Gruppi Muscolari allenati.

Poi il Database memorizzerà le **Sessioni**,cioè le fascie orarie, a cui ogni Cliente prenota per mantenere traccia in caso di positività di uno di essi al Coronavirus, ovvero si vuole memorizzare, oltre alle Sale in cui si è allenato, anche :

- Codice Sessione
- Data
- Orario di Inizio
- Orario di Fine

Ogni Palestra può dedicare una Sala per lo svolgimento di attività extra, cioè dei **Corsi** speciali ai quali i Clienti si iscrivono per poter partecipare, dei quali si vuole sapere :

- Nome del Corso
- Numero di Iscritti

- Data
- Ora di Inizio
- Ora di Fine
- MediaVoto

Ogni Cliente ha la possibilità di lasciare una **Recensione** alla Palestra o al corso a cui è iscritto, il database memorizzerà la piattaforma su cui è stata fatta e ovviamente il voto.
Si memorizza anche il **Manager** della palestra e le informazioni dei vari **Dipendenti** che vi lavorano, in particolare si vuole sapere :

- Codice Fiscale
- Nome
- Cognome
- Sesso
- Stipendio

I dipendenti possono essere dei **Trainer** per i quali si vuole sapere gli anni di esperienza e se ha seguito il corso di Primo Soccorso. Questi a loro volta si distinguono in **Esterni**, ovvero gli istruttori dei Corsi extra di cui si vuole sapere il numero di corsi che insegnano e il loro ambito di specializzazione, e in **Interni**, cioè coloro che si occupano di seguire l'allenamento classico dei clienti come la compilazione delle schede. Di questi si sa il numero di persone seguite e il target demografico in cui sono specializzati

2.2 Glossario Termini

Termine	Descrizione	Collegamenti
Cliente	Utente che ha sottoscritto un abbonamento alla palestra	Palestra, Sessione, Scheda, Corso, Recensione
Palestra	Luogo in cui è possibile allenarsi	Manager, Dipendente, Sale, Cliente, Recensione
Dipendente	Persona che lavora nella palestra	Palestra
Trainer	Dipendente della palestra che si occupa degli allenamenti	Entità figlia di Dipendente
Interno	Allenatore interno alla palestra che si occupa di seguire l'allenamento dei singoli clienti	Entità figlia di Trainer, Scheda
Esterno	Allenatore esterno affiiato alla palestra che si occupa di insegnare nei corsi di quest'ultima	Entità figlia di Trainer, Corso
Corso	Una serie di lezioni in cui un allenatore esterno insegna una determinata disciplina legata al mondo del fitness	Esterno, Clienti, RecensioneCorso, Sala
RecensioneCorso	Una recensione relativa alla qualità di insegnamento di un corso della palestra	Entità figlia di Recensioni, Corso
Recensione	Una recensione relativa alla struttura della palestra	Palestra, Cliente
Manager	Persona che si occupa della gestione della palestra	Palestra

Scheda	Raccolta degli esercizi, decisa da un allenatore, che un determinato cliente della palestra deve svolgere ad ogni allenamento	Interno, Cliente
Sala	Stanza della palestra in cui si segue un tipo preciso di allenamento	Palestra, Attrezzo, Sessione, Corso
Attrezzo	Strumento, contenuto in una sala, che è utilizzato per l'allenamento	Sala
Sessione	Fascia oraria di una certa giornata in cui i Clienti possono prenotarsi per svolgere un allenamento con Scheda. Durante una Sessione è possibile allenarsi in più Sale	Cliente, Sala

2.3 Operazioni

Operazione	Tipo	Frequenza
Inserimento di un nuovo cliente	S	300 al mese
Inserimento di una nuova recensione	S	100 al mese
Ricerca delle Sessioni prenotate per un certo giorno	L	1100 al giorno
Prenotarsi per una sessione	S	1000 al giorno
Inserimento di una recensione di un corso	S	60 al mese
Visualizzare il punteggio medio di un corso	L	500 al mese
Creazione scheda con relativo Trainer che l'ha compilata	S	300 al mese
Aggiornare scheda	S	2000 al mese
Visualizzare il punteggio medio di una palestra	L	900 al mese
Inserire nuovo corso con relativo Trainer Esterno	S	10 al mese
Iscriversi a un corso	S	1000 al mese
Visualizza dati Tranier Interno	L	210 al mese
Visualizzare dati Trainer Esterno	L	50 al mese

3 Progettazione Concettuale

3.1 Lista Entità

Se non indicato diversamente, l'attributo è NOT NULL

- Cliente: -NumeroTelefono: char (10) primary key
  -Nome: varchar (50)
  -Cognome: varchar (50)
  -Sesso varchar (10)
  -Età: varchar (10)
- Manager: -CF: char (16)
  -Nome: varchar (50)
  -Cognome: varchar (50)
- Palestra: -Codice: varchar (10)
  -Numero telefonico: varchar (10)
  -OraApertura: time
  -OraChiusura: time
  -MediaVoto: int
  -Indirizzo: attributo composto

+Città: varchar (50)
+Via: varchar (50)
+Civico: varchar (10)

- Dipendente: -CF: char (16)
  -Nome: varchar (50)
  -Cognome: varchar (50)
  -Sesso: varchar (10)
  -Stipendio: decimal (7,2)

  L'entità dipendente si specializza in una sottocategoria con generalizzazione parziale
  - Trainer: +AnniEsperienza: int
    +PrimoSoccorso: bool

  L'entità trainer si specializza a sua volta in due sottocategorie con una generalizzazione totale:

  - Interno: <>#clientiseguiti int
    <>Target: attributo composto
    +Età: varchar (3)
    +Sesso: varchar (10)

  - Esterno:
    <>#corsi seguiti: int
    <>Ambitospecializzazione: varchar (50)

- Scheda: -Codice: int
  -Durata: varchar (10)
  -Difficoltà: varchar (10)
  -Gruppo Muscolare: attributo composto
  +Cardio: bool
  +Gambe: bool
  +Addominali: bool
  +Petto: bool
  +Dorsali: bool
  +Braccia: bool
- Sessione: -Codice: int
  -Data: date
  -OraInizio: time
  -OraFine: time
- Sala: -Capienza: varchar (10)
  -Tipo: varchar (20)
  -Palestra: varchar (10)
- Attrezzo: -Codice: varchar (100)
  -Numero: int
  -Peso: varchar (10) può essere NULL
- Corso: -Nome: varchar (50)
  -#iscritti: int
  -Data: date
  -OraInizio: time
  -OraFine: time
- Recensione: -Codice: varchar (100)
  -Stelle: int
  -Piattaforma: varchar (50)

L’entità recensione si specializza in una sottocategoria con generalizzazione parziale:

- Recensione Corso:
  -VotoCorso: int

3.2 Tabella delle Relazioni

Relazione	Entità conivolte	Descrizione	Attributi
Frequenza	Cliente (1,N) Palestra(1,N)	Ogni cliente può essere iscritto a una o più palestre, ogni palestra può avere uno o più clienti	Nessuno
Prenotazione	Cliente (0,N) Sessione (1,N)	Un cliente può prenotare più sessioni o nessuna, una sessione può essere prenotatada uno o più clienti	Nessuno
Allenamento	Cliente (0,1) Scheda(1,1)	Un cliente può avere una scheda o non averne nessuna se si è appena iscritto (oppure se segue solo corsi), ogni scheda è svolta da un solo cliente	Nessuno
Voto	Cliente (0,N) Recensione(1,1)	Un cliente può lasciare una o più recensioni, una recensione può essere scritta da un solo cliente	Nessuno
Iscrizione	Cliente (0,N) Corso (0,N)	Un cliente può essere iscritto a più corsi o a nessun corso, un corso può avere più iscritti o non averne nessuno	Nessuno
Prenotazione	Sessione (1,N) Sala(1,N)	Una sessione di allenamento può essere svolta in una o più sale, in una sala possono esserci più sessioni	Nessuno
Contenuto	Sala (0,N) Attrezzo(0,1)	Una sala può avere più attrezzi o nessuno (se ad esempio è una sala di allenamento a corpo libero), un attrezzo può essere in una sola sala o in nessuna(in magazzino)	Nessuno
Suddivisione	Sala (1,1) Palestra (1,N)	Una palestra può avere una o più sale, una sala può essere in una sola palestra	Nessuno
Svolgimento	Sala (0,N) Corso (1,1)	Una sala può avere molti corsi (in orari diversi) oppure nessuno, un corso può essere svolto in una sola sala	Nessuno
Valutazione	Corso (0,N) Recensionicorso(1,1)	Un corso può avere molte recensioni o non averne nessuna, una recensione riguarda un solo corso	Nessuno
Voto	Recensioni (1,1) Palestra(0,N)	Una palestra può avere molte recensioni oppure nessuna, una recensione riguarda una sola palestra	Nessuno
Insegnamento	Corso (0,1) Esterno(1,N)	Un corso può avere un solo un istruttore esterno, un istruttore esterno può insegnare uno o più corso	Nessuno
Gestione	Palestra (0,1) Manager (1,N)	Una palestra può avere un solo manager, un manager può gestire più palestre	Nessuno
Impiego	Palestra (1,N) Dipendenti (1,1)	Un palestra può avere uno o più dipendenti, un dipendente può può lavorare in una sola palestra	Nessuno

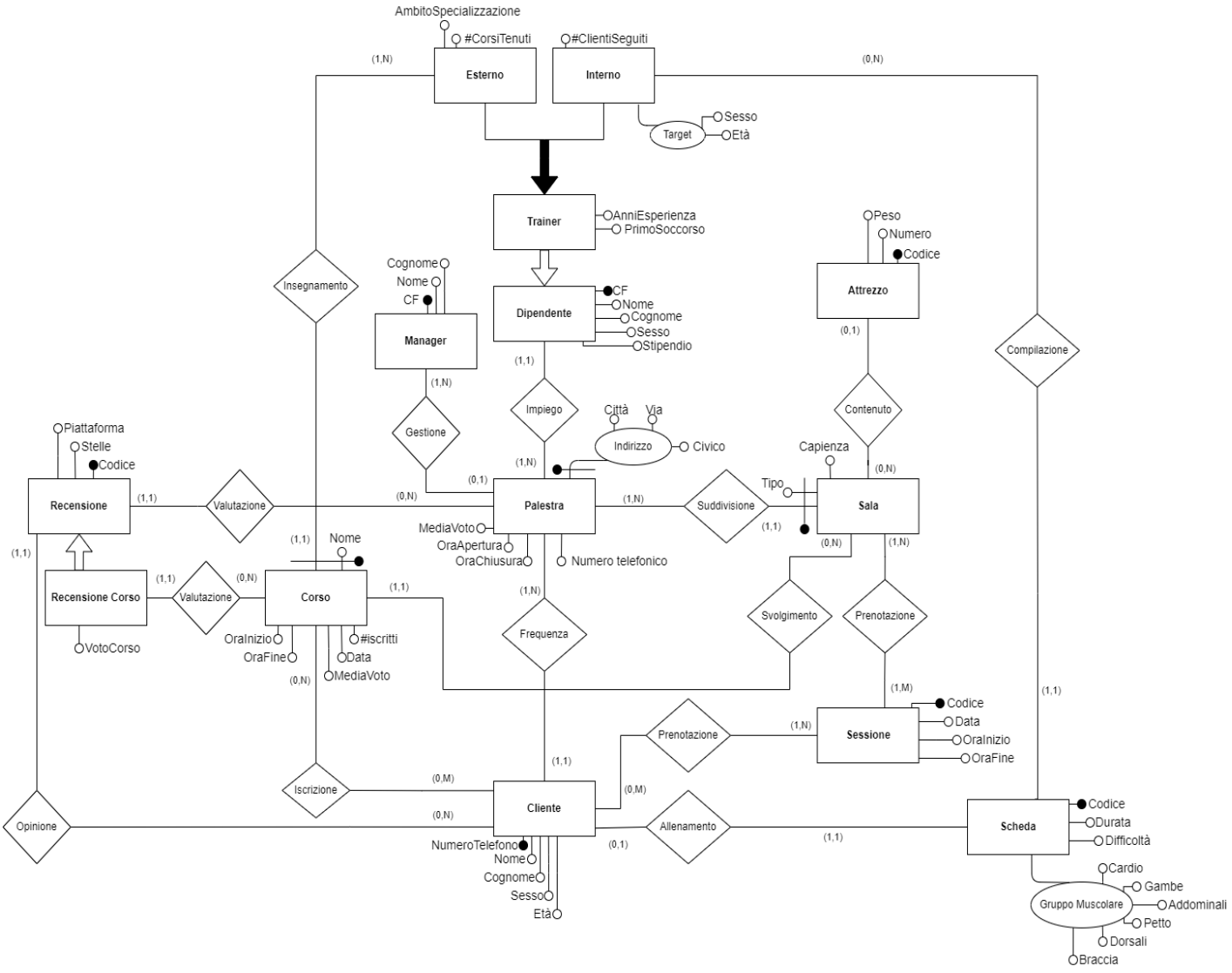
Compila	Scheda (1,1), Interno (0,N)	Una scheda può essere compilata da un solo istruttore esterno, un istruttore esterno può aver compilato molte schede o nessuna	Nessuno

Vincoli non rappresentabili tramite schema E-R:

- Una recensione deve comprendere anche la recensione del corso se e solo se è scritta da un cliente che frequenta un corso della palestra (e non se è di un cliente che si allena in palestra seguendo una scheda)

Vincoli di derivazione:

- L'attributo MediaVoto dell'entità Palestra è uguale alla media delle stelle presenti nelle recensioni della palestra
- L'attributo #ClientiSeguiti è uguale alla somma dei clienti per cui quell'interno ha creato una scheda
- L'attributo #CorsiTenuti è uguale alla somma dei corsi che hanno quell'esterno come insegnante



4 Progettazione Logica

4.1 Ristrutturazione

4.1.1 Analisi delle Ridondanze

Note:

- I Volumi sono calcolati sulla ipotesi di una catena con 5 palestre, 3000 clienti totali (600 in media per palestra) con ⅓ dei clienti che hanno lasciato una recensione e la quasi totalità di loro che possiede una Scheda(cioè solo pochi clienti partecipano solo a Corsi). Si immagina che il numero di nuovi iscritti ogni mese sia attorno ai 300 in totale.Ogni Palestra ha in media 10 Trainer Esterni e 10 Trainer Interni, per un totale di 50 per entrambe le entità.

- Assumo che gli accessi in scrittura costano il doppio di quelli in lettura

Analizziamo l'attributo **MediaVoto** presente in Corsi e Palestre.

MEDIAVOTO per CORSI e PALESTRE

1. Inserimento nuova recensione (100 al mese)
2. Visualizzazione punteggio medio (500 al mese)

Palestra	E	5
Corso	E	100
Recensioni	E	1000
Valutazione	R	2000

OPERAZIONE 1: Inserimento nuova recensione (2 al giorno)
OPERAZIONE 2: visualizzare punteggio medio di una palestra (30 volte al giorno)

CON PRESENZA RIDONDANZA:
operazione 1: una palestra ha in media 1000/5 = 200 recensioni

Concetto	**Costrutto**	**Numero Accessi**	**Tipo**
Valutazione	Relazione	1	S
Valutazione	Relazione	200	L
Palestra	Entità	1	S

x2 volte/giorno
x2 volte/giorno
x2 volte/giorno

operazione 2:

Concetto	**Costrutto**	**Numero Accessi**	**Tipo**
Palestra	Entità	1	L

x30 volte/giorno

costo giornaliero: 2*2 + 200 + 2*2 + 30 = 238

SENZA RIDONDANZA:
operazione 1:

Concetto	**Costrutto**	**Numero Accessi**	**Tipo**
Voto	Relazione	1	S

x2 volte/giorno

operazione 2:

Concetto	Costrutto	Numero Accessi	Tipo
Palestra	Entità	200	L

x30 volte/giorno

Costo = 2*2 + 200*30 = 6004

Dunque conviene mantenere l'attributo ''MEDIAVOTO''

Analizziamo gli attributi **#ClientiSeguiti** e **#CorsiTenuti**, presenti rispettivamente in Interno ed Esterno.
Le seguenti due ridondanze sono ricavabili contando il numero di volte che le loro chiavi appaiono come chiavi esterne in Schede e Corsi rispettivamente

#CLIENTISEGUITI per Trainer Interno

1. Creazione scheda con relativo Trainer che l'ha compilata (300 al mese)
2. Visualizza dati Trainer Interno (210 al mese)

Schede	E	2800
Trainer Interni	E	50
Compilazione	R	56

OPERAZIONE 1: Creazione scheda con relativo Trainer che l'ha compilata (10 al giorno)
OPERAZIONE 2: Visualizza dati Trainer Interno (7 al giorno)

CON PRESENZA RIDONDANZA:

operazione 1:

Concetto	Costrutto	Numero Accessi	Tipo
Scheda	Entità	1	S
Compilazione	Relazione	1	S
Trainer Interno	Entità	1	L
Trainer Interno	Entità	1	S

x10 volte/giorno
x10 volte/giorno
x10 volte/giorno
x10 volte/giorno

operazione 2:

Concetto	Costrutto	Numero Accessi	Tipo
Trainer Interno	Entità	1	L

x7 volte/giorno

Costo = 10*2 + 10*2 + 10 + 10*2 + 7 = 77

SENZA RIDONDANZA:

operazione 1:

Concetto	**Costrutto**	**Numero Accessi**	**Tipo**
Scheda	Entità	1	S
Compilazione	Relazione	1	S

x10 volte/giorno
x10 volte/giorno

operazione 2:
(2800 clienti divisi tra i 50 Trainer Interni, quindi in media 56 Clienti seguiti per Trainer)

Concetto	**Costrutto**	**Numero Accessi**	**Tipo**
Trainer Interno	Entità	1	L
Compilazione	Relazione	56	L

x7 volte/giorno
x7 volte/giorno

Costo = 10*2 + 10*2 + 7 + 56*7 = 439
Dunque conviene mantenere l'attributo ''#ClientiSeguiti''

#CORSITENUTI per Trainer Esterno

1. Inserire nuovo corso con relativo Trainer Esterno (10 al mese)
2. Visualizza dati Trainer Esterno (50 al mese)

Entrambe le operazioni hanno frequenza molto bassa, il numero di corsi offerti rimane sempre attorno allo stesso e in genere un Trainer Esterno tiene un massimo di 2 o 3 corsi, quindi è una ridondanza che occupa spazio senza portare benefici.

4.1.2 Eliminazione delle Generalizzazioni

Generalizzazione	Risoluzione
Recensioni ⇐ RecensioniCorso	Nella generalizzazione tra Recensioni e RecensioniCorso si è scelto di accorpare la figlia nel padre. L'entità RecensioneCorso viene accorpata in Recensione in quanto gli accessi alle due entità sono contestuali, la relazione Valutazione che prima collegava RecensioneCorso (1,1) a Corso (0,N) ora collega le entità Recensione (0,1) a Corso (0,N). Aggiungo un attributo Stellecorso all'interno di Recensione.
Trainer ⇐ Interno, Esterno	Per la generalizzazione tra Trainer ed Interno/Esterno si è scelto di accorpare il padre nelle figlie, in quanto gli accessi a padre e figlie sono separati e la generalizzazione è totale.

	L'entità Trainer viene accorpata in Interno ed Esterno; la generalizzazione parziale che era prima presente tra Trainer e Dipendente sarà ora una generalizzazione parziale tra Dipendente, come entità padre, e Interno/Esterno, come entità figlie.
Dipendenti ⇐ Interno, Esterno	Nella conseguente generalizzazione tra Dipendenti ed Interno/esterno si è scelto di sostituire le generalizzazioni con delle relazioni, in quanto conviene mantenere le due entità figlie separate dall'entità padre, per evitare una complessità eccessiva. Dunque tra Dipendente (0,1) ed Interno (1,1) si crea la relazione DipInterno. Tra Dipendente (0,1) ed Esterno (1,1) si crea invece la relazione DipEsterno.

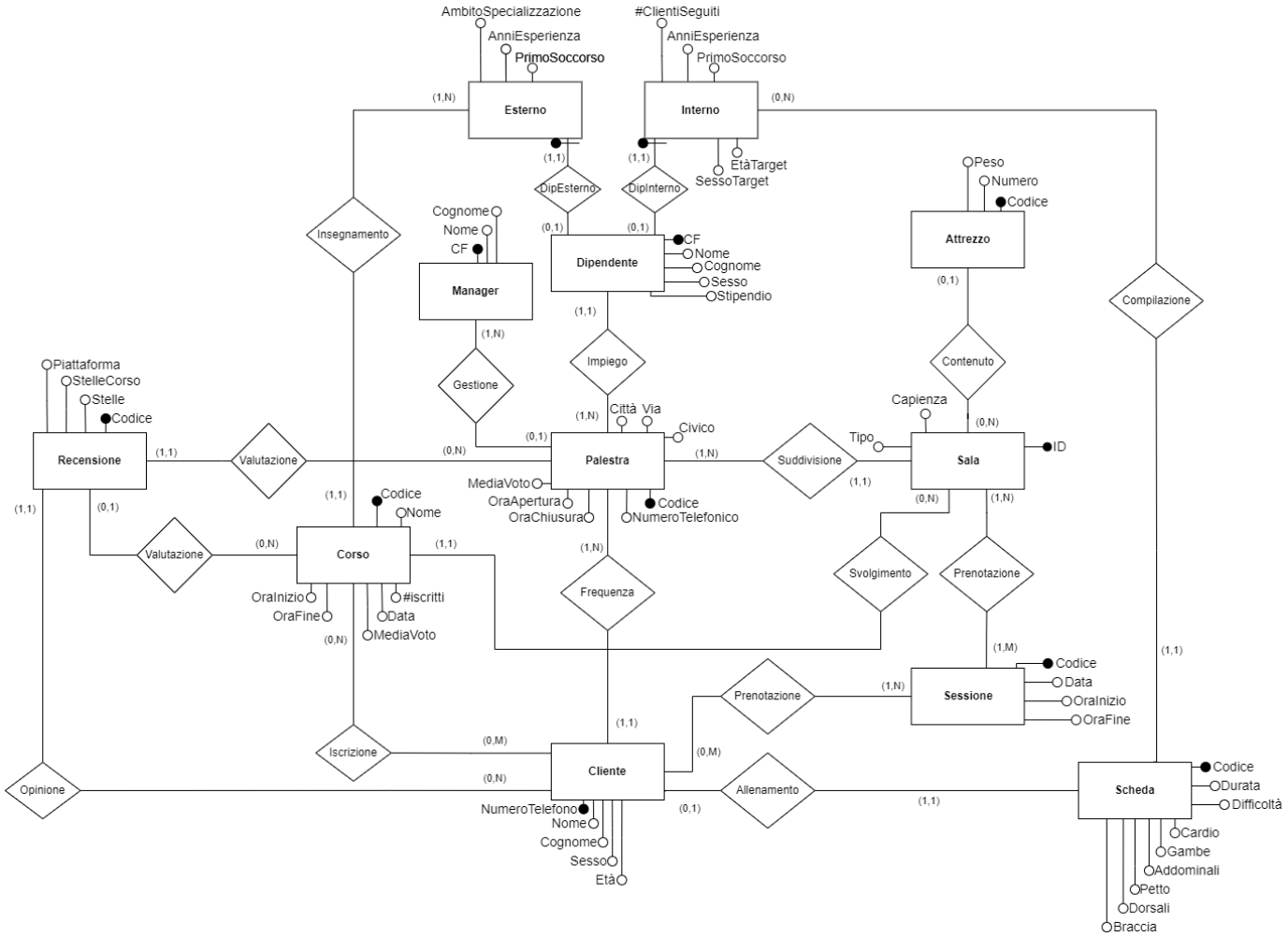
4.1.3 Scelta degli Identificatori Primari

Per l'entità Sala era stata scelta come chiave primaria una combinazione dell'attributo Tipo e dell'attributo esterno Indirizzo di Palestra, ma in questo modo Attrezzo avrebbe avuto una chiave esterna composta, aumentando di molto la complessità, dunque abbiamo deciso di introdurre l'attributo ID come chiave primaria di Sala.

Dopo aver eliminato le generalizzazioni abbiamo scelto come chiave primaria delle entità Esterno ed Interno l'attributo chiave esterna Dipendente riferito a CF in entità Dipendete .

Per l'entità Palestra era stata scelta come chiave primaria l'attributo composto Indirizzo, ma in questo modo Sala, Cliente,Recensione e Dipendenti avrebbero dovuto memorizzare i tre campi di Indirizzo come chiave esterna , dunque abbiamo deciso di introdurre l'attributo Codice come chiave primaria.

Per l'entità Corso era stata scelta come chiave primaria una combinazione dell'attributo Nome e dell'attributo esterno CF di Esterno, tuttavia questo aumenta di molto la complessità dello schema relazionale. Dunque si è scelto di usare come nuova chiave "Codice" , un nuovo attributo id.



4.2 Creazione delle tabelle (A->B significa che B è chiave esterna di A)

Manager : (CF, Nome, Cognome)
Palestra : (Codice, Città, Via, Civico, NumeroTelefonico, OraApertura, OraChiusura, MediaVoto, Manager->Manager.CF)
Dipendente : (CF, Nome, Cognome, Sesso, Stipendio, Palestra->Palestra.Codice)
Esterno : (Dipendente->Dipendente.CF, Esperienza, PrimoSoccorso, AmbitoSpecializzazione)
Interno : (Dipendente->Dipendente.CF, Esperienza, PrimoSoccorso, #ClientiSeguiti, EtàTarget, SessoTarget)
Scheda : (Codice, , Durata, Difficoltà, Cardio, Gambe, Addominali, Petto, Dorsali, Braccia, Interno->Interno.Dipendente)
Cliente : (NumeroTelefono, Nome, Cognome, Sesso, Età, Palestra->Palestra.Codice)
Sessione : (Codice, Data, OraInizio, OraFine)
PrenotazioneSessione : (Cliente->Cliente.NumeroTelefono, Sessione->Sessione.Codice)
Sala : (ID, Capienza, Tipo, Palestra->Palestra.Codice)
PrenotazioneSale : (Sala->Sala.ID, Sessione->Sessione.Codice)
Attrezzo : (Codice, Numero, Peso, Sala->Sala.ID)
Corso : (Codice, Nome, #iscritti, Data, OraInizio, OraFine, Esterno->Esterno.Dipendente, Sala->Sala.Codice)
IscrizioneCorso : (Corso->Corso.Codice, Cliente->Cliente.NumeroTelefono)
Recensione : (Codice, Stelle, StelleCorso, Piattaforma, Cliente->Cliente.NumeroTelefono, Palestra->Palestra.Codice, Corso->Corso.Codice)

5 Query e Indici

5.1 Query

1) Mostrare informazioni dei Dipendenti che sono Trainer Interni o Esterni che hanno uno stipendio minore di 1500 e hanno o ottenuto il certificato di PrimoSoccorso o che hanno più di 5 anni di Esperienza

	nome character varying (50)	cognome character varying (50)	stipendio numeric (7,2)	primosoccorso boolean	anniesperienza integer	palestra character varying (10)
1	marco	colli	1400.00	true	10	1
2	carlo	passuello	1000.00	true	13	2
3	alessia	bellucco	1300.00	false	9	2
4	antonio	aquilotto	1200.00	false	11	3
5	martino	feltre	1200.00	true	6	3
6	mattoa	prucco	1400.00	true	3	3
7	niccolo	zancan	1300.00	true	5	4

```
SELECT *
FROM    (   (SELECT Nome, Cognome, Stipendio, PrimoSoccorso, AnniEsperienza,
    Palestra
            FROM Dipendente d JOIN Interno i ON d.CF = i.Dipendente
            WHERE i.PrimoSoccorso = TRUE OR i.AnniEsperienza >= 5)
        union
            (SELECT Nome, Cognome, Stipendio, PrimoSoccorso,
     AnniEsperienza, Palestra
            FROM Dipendente d JOIN Esterno e ON d.CF = e.Dipendente
            WHERE e.PrimoSoccorso = TRUE OR e.AnniEsperienza >= 5)
        ) AS tq
WHERE tq.Stipendio <= 1400
ORDER BY tq.Palestra, tq.Stipendio
```

2) Mostrare tutte le recensioni di una palestra che hanno valutazione maggiore della sua MediaVoto e che contengono recensione anche valutazione di un
corso o che sono state lasciate su piattaforma “Google” , indicando a fianco il nome del cliente che l’ha scritta.

	nome character varying (50)	stelle integer	filiale character varying (50)	stellecorso integer	corso integer	piattaforma character varying (10)
1	veronica	4	milano	4	7	google
2	veronica	4	milano	3	2	google
3	veronica	4	milano	5	12	google
4	carlotta	5	torino	5	3	trustpilot
5	luigi	4	torino	5	8	google
6	carlotta	5	torino	5	13	trustpilot
7	luigi	4	torino	2	3	google
8	alex	3	vicenza	3	17	yahoo

```
DROP view IF EXISTS rec_complete;
CREATE view rec_complete AS
            SELECT Stelle, StelleCorso ,Piattaforma, Palestra, Cliente, Corso
            FROM Recensioni
            WHERE  Piattaforma='google' OR StelleCorso IS NOT NULL;


SELECT c.Nome, r.Stelle, p.Città AS filiale, r.StelleCorso, r.Corso, r.Piattaforma
FROM (rec_complete r JOIN cliente c ON c.numerotelefono=r.cliente)
    JOIN Palestra p ON r.Palestra = p.Codice
WHERE r.Stelle > p.MediaVoto
ORDER BY r.Palestra
```

3) Mostrare i dati dei 5 clienti più attivi(maggior numero di sessioni prenotate + iscrizioni a corsi) mostrando informazioni tra cui anche la palestra frequentata

	nome character varying (50)	cognome character varying (50)	palestra character varying (10)	tot_attivita bigint	di_cui_corsi bigint
1	luca	depaoli	1	15	1
2	stefano	lattenero	5	9	2
3	alex	campagnaro	4	8	4
4	veronica	prendin	2	7	3
5	luigi	viola	1	7	1

```
DROP view IF EXISTS num_corsi_cliente;
CREATE view num_sessioni_cliente  AS
    SELECT Cliente, COUNT(*) AS prnt_sessioni
    FROM PrenotazioneSessione
GROUP BY Cliente;

DROP view IF EXISTS num_corsi_cliente;
CREATE view num_corsi_cliente  AS
    SELECT Cliente, COUNT(*) AS iscr_corsi
    FROM IscrizioneCorso
GROUP BY Cliente;

DROP view IF EXISTS att_cliente;
CREATE view att_cliente  AS
SELECT c.cliente, c.iscr_corsi, s.prnt_sessioni, (c.iscr_corsi + s.prnt_sessioni) AS
                                                    tot_attivita
    FROM num_corsi_cliente c JOIN num_sessioni_cliente s ON c.cliente = s.cliente;

SELECT c.Nome, c.Cognome, c.Palestra, a.tot_attivita, a.iscr_corsi AS Di_cui_Corsi
FROM Cliente c Join att_cliente a ON c.NumeroTelefono = a.Cliente
ORDER BY a.tot_attivita DESC limit 5;
```

4) Mostrare per ogni palestra il primo cliente che ha prenotato una sessione

	numerotelefono character (10)	palestra character varying (10)	nome character varying (50)	cognome character varying (50)	data date	orainizio time without time zone
1	3494409887	1	luca	depaoli	2020-01-01	08:00:00
2	3494409879	2	davide	neri	2020-02-10	08:00:00
3	3494409880	3	giulia	neri	2020-02-11	08:00:00
4	3494409877	4	giacomo	verdi	2020-02-13	08:00:00
5	3494409878	5	paolo	rossi	2020-03-10	08:00:00

```
DROP view IF EXISTS clienti_e_loro_sess ;
CREATE VIEW clienti_e_loro_sess AS
    SELECT c.NumeroTelefono, c.Nome, c.Cognome, c.Palestra, s.Data, s.orainizio
    FROM (Cliente c JOIN PrenotazioneSessione ps ON c.NumeroTelefono = ps.Cliente)
        JOIN Sessione s ON ps.Sessione = s.Codice;


DROP view IF EXISTS pal_prima_sess ;
CREATE VIEW pal_prima_sess AS
SELECT Palestra, MIN(data) as primo_giorno_prnt, MIN(orainizio) as primo_orario_prnt
FROM clienti_e_loro_sess
GROUP BY Palestra;

SELECT cs.NumeroTelefono, cs.Palestra, cs.Nome, cs.Cognome, cs.data, cs.orainizio
FROM clienti_e_loro_sess cs JOIN pal_prima_sess pps ON cs.Palestra = pps.palestra
WHERE  cs.data = pps.primo_giorno_prnt AND cs.orainizio = pps.primo_orario_prnt
ORDER BY cs.data, cs.orainizio;
```

5) Mostrare in ordine decrescente le Palestre in base a quanto sono attrezzate (cioè in base al numero di attrezzi nelle sale della Palestra)

	codice character varying (10)	città character varying (50)	mediavoto integer	att_pal numeric
1	2	milano	3	58
2	1	padova	5	53
3	4	vicenza	2	38
4	3	torino	3	36
5	5	napoli	4	24

```
DROP view IF EXISTS natt_sala;
CREATE view natt_sala AS
    SELECT s.id AS cod_sala, s.tipo, s.palestra, sum(a.numero) AS att_sala
    FROM attrezzo a JOIN sala s ON a.sala = s.id
    GROUP BY s.id;


DROP view IF EXISTS natt_palestra;
CREATE view natt_palestra AS
        SELECT p.Codice, sum(n.att_sala) AS att_pal
        FROM Palestra p JOIN natt_sala n ON p.Codice = n.Palestra
        GROUP BY p.Codice
        ORDER BY att_pal DESC;


SELECT p.Codice, p.Città, p.MediaVoto, nap.att_pal as Num_Attrezzi_Totale
FROM Palestra p JOIN natt_palestra nap ON p.Codice = nap.Codice
ORDER BY nap.att_pal DESC
```

6) Mostrare le Palestre che hanno uno stipendio medio < 1500 con CF e cognome del manager (qui valore = 1500)

	cod_pal character varying (10)	manager character (16)	cognome character varying (50)	stip_medio numeric (10,2)
1	3	brtlse95h56f839n	barutti	1400.00
2	2	crtmrc74m16l840s	carta	1380.00

```
DROP view IF EXISTS sotto_media;
CREATE view sotto_media AS
    SELECT q.città, q.Manager, q.codice AS cod_pal,
           cast(avg(d.stipendio) as DECIMAL(10,2)) as stip_medio
    FROM palestra q JOIN dipendente d ON q.codice = d.palestra
            GROUP BY q.Codice
           HAVING avg(d.Stipendio)<1500;

SELECT s.cod_pal, s.manager, m.Cognome, s.stip_medio
FROM sotto_media s JOIN manager m on s.manager = m.cf
```

5.2 Indici

Dato che ogni Cliente possiede e deve visualizzare la propria Scheda ogni qual volta che si reca in palestra per un allenamento,ovvero più di una decina di volte al mese, mentre la Scheda viene aggiornata al più una volta al mese, è vantaggioso creare un indice sulla entità Scheda, specialmente se si considera un'espansione futura della catena con conseguente aumento dei Clienti.

CREATE INDEX idx_scheda ON Scheda(Codice)

6 Codice C++

Descrizione dell'utilizzo del codice

Il codice C++ incluso nel progetto comprende il main ed due funzioni, una per controllare se si è ottenuto le Query correttamente e una per la stampa di queste.
Per la compilazione è sufficiente eseguire il seguente comando all'interno della cartella del programma: **g++ AccessoDB.cpp -L dependencies\lib -lpq -o AccessoDB**

Alla fine del login con il Database sono definite le seguenti costanti all'inizio del programma, che vanno ridefinite in base alla configurazione del database importato nella macchina locale :
PG_HOST, PG_USER, PG_PASS, PG_PORT, PG_DB

Prima di eseguire il programma, ci si deve assicurare che le viste necessarie siano presenti del DB. Queste viste sono indicate nel file sql.

Il main consiste in un ciclo while contenente la lista delle query definite e uno switch con il quale l'utente può scegliere quale tra le 6 disponibili visualizzare.
In particolare digitando 0 si esce dal ciclo, mentre con valori minori di 0 e maggiori di 6 mostra messaggio OUT_OF_RANGE e richiede l'inserimento del numero.