



Università di Pisa  
Facoltà di Ingegneria  
Corso di laurea in  
Ingegneria Informatica

## Documentazione Progetto per il corso di Basi di dati Anno Accademico 2016-2017

Prof. Gigliola Vaglini, Ing. Francesco Pistolesi

Studenti: Mantovani Giacomo, Pellicci Giacomo

Matricola: 533780, 533903

# Indice

<b>1. Progettazione concettuale .....</b>	<b>3</b>
1.1 Diagramma E-R .....	4
<b>2. Documentazione del diagramma E-R .....</b>	<b>5</b>
2.1 Documentazione delle entità .....	5
2.2 Documentazione delle relazioni .....	8
2.3 Regole di vincolo .....	13
2.4 Ristrutturazione del diagramma E-R .....	13
<b>3. Individuazione e analisi delle operazioni sui dati .....</b>	<b>14</b>
3.1 Operazione 1 .....	15
3.2 Operazione 2 .....	19
3.3 Operazione 3 .....	20
3.4 Operazione 4 .....	21
3.5 Operazione 5 .....	23
3.6 Operazione 6 .....	24
3.7 Operazione 7.....	25
3.8 Operazione 8 .....	28
<b>4. Progettazione logica .....</b>	<b>30</b>
4.1 Schema logico .....	30
4.2 Vincoli di integrità referenziale .....	32
4.3 Analisi delle dipendenze funzionali e normalizzazione .....	35
<b>5. Area Analytics .....</b>	<b>38</b>
5.1 Funzionalità 1 .....	38
5.2 Funzionalità 2 .....	41
5.3 Funzionalità 3 .....	46

# PROGETTAZIONE CONCETTUALE

Per la costruzione dello schema concettuale abbiamo adottato la strategia bottom-up.

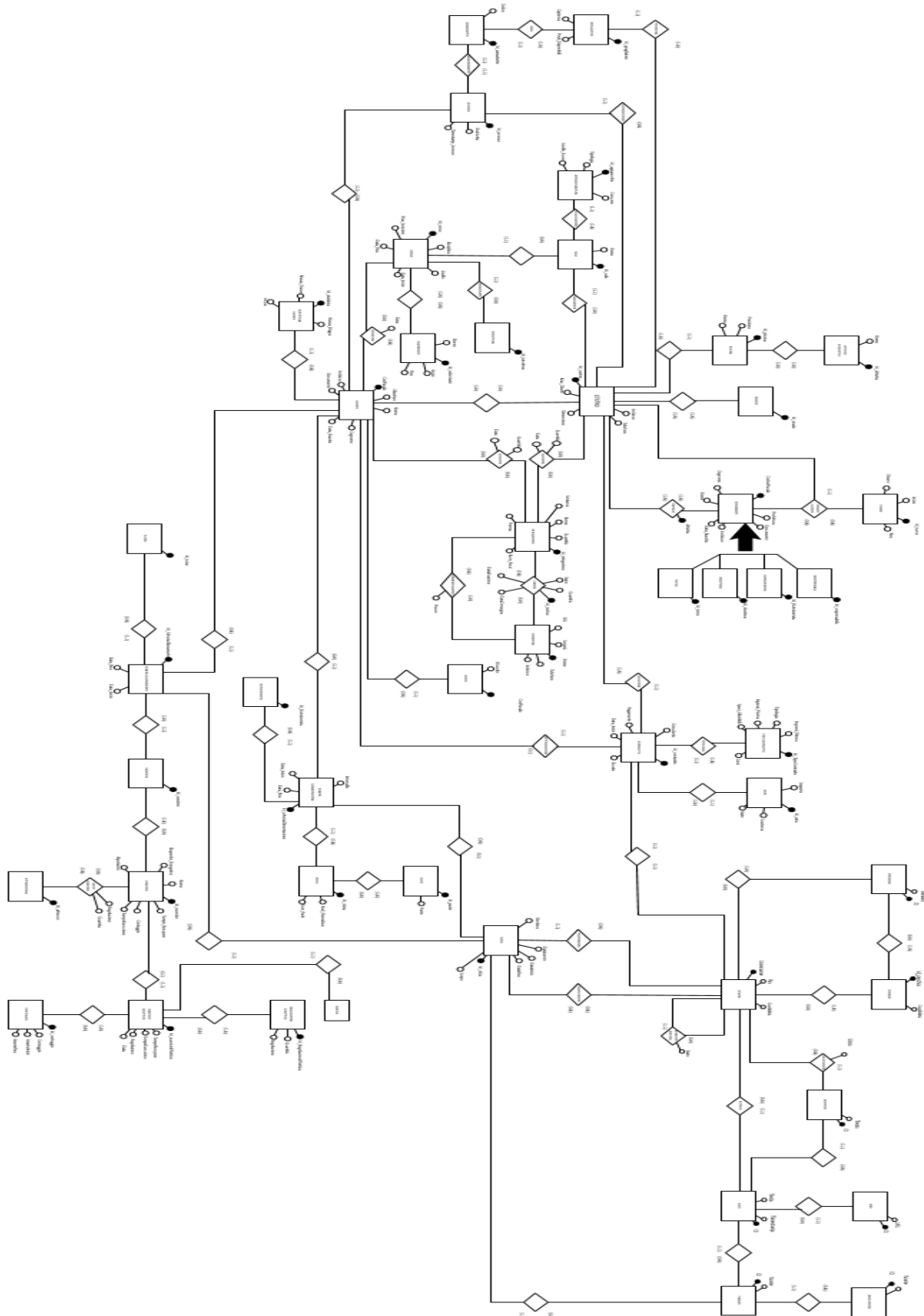
Si parte dalle specifiche iniziali e si suddividono fino a dare specifica ad una componente minima, in questo caso i capitoli della documentazione fornita, di cui si disegna lo schema E-R.

Questi schemi vengono fusi e integrati fino ad ottenere lo schema finale.

Il vantaggio che abbiamo potuto trarre dalla strategia bottom-up è quello di poter descrivere preliminarmente senza preoccuparci, le singole componenti della base di dati.

Attuando questa tecnica siamo riusciti ad ottenere uno schema concettuale corrispondente ai requisiti dati.

## 1.1 Diagramma E-R



## 2. Documentazione del diagramma E-R

### 2.1 Documentazione delle entità

Di seguito è stata riportata una tabella contenente le varie entità, e per ciascuna di esse è presente una piccola descrizione, i relativi attributi ed infine la chiave primaria.

Entità	Descrizione	Attributi	Identificatore
Centro	Ogni sede presente sul territorio nazionale.	IdCentro, Indirizzo, Telefono, Dimensione, MaxClienti;	IdCentro
Sala	Le sale di un centro fitness	IdSala, Nome;	IdSala
Apparecchiatura	Tutte le caratteristiche di una di un'apparecchiatura	IdApparecchiatura, Nome, Tipologia, Consumo, LivelloUsura;	IdApparecchiatura
Cliente	Le persone che accedono ad almeno un centro dell'azienda	CodFiscale, Nome, Cognome, DataNascita, Indirizzo, Documento, Prefettura, Obiettivo, Scopo;	CodFiscale
Dipendente	Le persone che lavorano nei vari centri fitness	CodFiscale, Nome, Cognome, DataNascita, Indirizzo, Documento, Prefettura;	CodFiscale
Contratto	Contratti sottoscritti con i vari clienti	IdContratto, Tipo, DataInizio, Durata, Pagamento;	IdContratto
TipoContratto	Tipologia del contratto scelta dal cliente	IdTipoContratto, Nome, IngressiFitness, DimensionePiscina, IngressiPiscina, Corsi	IdTipoContratto
Piscina	Piscine di proprietà dell'azienda	IdPiscina, Nome, Dimensione	IdPiscina
AttivitaAcquatica	Tutte le discipline svolte nelle piscine del centro	IdAttivita, Nome;	IdAttivita
Scopo	Obiettivo che il cliente intende raggiungere	CodFiscale, Muscolo;	CodFiscale
Rata	Rate pagate o da pagare da parte del cliente	IdRata, Importo, Scadenza, Stato;	IdRata
SchedaAllenamento	Scheda di allenamento che un cliente segue o ha seguito	IdSchedaAllenamento, DataInizio, DataFine,	IdSchedaAllenamento
Sessione	Sessioni di allenamento programmate	IdSessione;	IdSessione

Esercizio	Esercizi svolti dai vari clienti	IdEsercizio, Nome, DispendioEnergetico, Ripetizioni, Conteggio, TempoRecupero, TempoEsecuzione;	IdEsercizio
SchedaAlimentazione	Alimentazione creata in base all'obiettivo di un cliente	IdSchedaAlimentazione, DataInizio, DataFine, Intervallo;	IdSchedaAlimentazione
Dieta	Dieta assegnata ad un cliente da un medico nutrizionista	IdDieta, Kcal, NumPasti;	IdDieta
StatisticheCliente	Misurazioni di un cliente eseguite da un medico nutrizionista	IdStatisticheCliente, Peso, MassaMagra, MassaGrassa, Acqua, Data;	IdStatisticheCliente
Pasto	Raccolta di pasti che possono comporre le diete	Idpasto, Pasto;	IdPasto
Accessi	Controllo di ingresso e uscita dei vari clienti dal centro	IdAccesso, TimestampAccesso, OraUscita;	IdAccesso
Armadietto	Armadietto assegnato al cliente quando entra in un centro fitness	IdArmadietto, Codice, Stato;	IdArmadietto
Spogliatoio	Spogliatoi dei vari centri fitness	IdSpogliatoio, Capienza, PostiDisponibili, Posizione;	IdSpogliatoio
Corso	Corsi di varie attività organizzati dai centri fitness	IdCorso, Disciplina, Livello, DataInizio, DataFine, MaxIscrizioni;	IdCorso
Calendario	Date e orari dei vari corsi	IdCalendario, Giorno, Inizio, Fine;	IdCalendario
Turno	Descrive i giorni e fasce orarie in cui è necessario avere dipendenti a lavoro	IdTurno, Giorno, Inizio, Fine;	IdTurno
EsercizioEffettivo	Effettiva modalità di svolgimento di un esercizio da parte del cliente (differenze con la modalità proposta sulla scheda)	IdEsercizioEffettivo, TempoRecupero, TempoEsecuzione, Data;	IdEsercizioEffettivo
RegolazioneEffettiva	Effettiva modalità di regolazione di un determinato esercizio	IdRegolazioneEffettiva, Regolazione, Quantità;	IdRegolazioneEffettiva
Conteggio	Conteggio della fedeltà di riproduzione di ogni ripetizione	IdConteggio, Conteggio, IstanteInizio, IstanteFine;	IdConteggio
Integratore	Informazioni degli integratori presenti in una confezione	IdIntegratore, Nome, Sostanza, Quantità, Num_Pezzi, Forma;	IdIntegratore

Fornitore	Aziende con cui è possibile effettuare ordini di integratori	NomeFornitore, Società, IVA, GiorniPagamento, Indirizzo, Telefono;	NomeFornitore
Ordine	Ordini di integratori effettuati dal centro fitness	IdOrdine, Stato, DataEvasione, Quantità, DataConsegna;	IdOrdine
Utente	Informazioni dell'account associato ad un cliente	Username, Pin, Karma;	Username
Interesse	Insieme di interessi scelti dall'utente	IdInteresse, Ambito;	IdInteresse
Post	Pubblicazioni effettuate da un utente sul forum	Idpost, Testo, Timestamp;	IdPost
Thread	Suddivisione delle aree tematiche del forum	IdThread, Nome;	IdThread
Area Forum	Aree tematiche che compongono il forum	IdAreaForum, Nome;	IdAreaForum
Link	Link ipertestuali all'interno di un post	IdLink, URL;	IdLink
Risposta	Risposte dei vari utenti ad un post	IdRisposta, Testo, sommaVoti;	IdRisposta
Tutor	Sono tutte generalizzazioni dell'entità Dipendente, in quanto sono dipendenti con "particolari" mansioni.		
Responsabile			
Direttore			
Nutrizionista			
Istruttore			
Sfida	Informazioni riguardo a sfide lanciate dai clienti	IdSfida, DataLancio, DataInizio, DataFine, Scopo;	IdSfida
Cerchia	Collega vari utenti in una cerchia	IdCerchia;	IdCerchia
Attrezzatura	Attrezzi utilizzati nelle attività	IdAttrezzatura, Nome;	IdAttrezzatura
Orario	Orario di apertura dei centri fitness	IdOrario, OraInizio, OraFine;	IdOrario
Valutazione	Valutazione espressa da un utente ad una risposta correlata ad un post	IdValutazione, Voto;	IdValutazione
Giacenza	Magazzino contenente gli integratori venduti da un determinato centro	IdGiacenza, Data, Ordine, Quantità;	IdGiacenza
Acquisto	Acquisto di integratori da parte di un cliente	IdAcquisto, Quantità, Data;	IdAcquisto

## 2.1 Documentazione delle relazioni

Di seguito è stata riportata una tabella contenente le relazioni tra le varie entità. Per ogni relazione è presente una descrizione, le entità coinvolte e gli eventuali attributi.

Relazione	Descrizione	Entità coinvolte	Attributi
2.2.1			
Ubicazione	Associa le sale al centro fitness di appartenenza	Centro (1,N) Sala (1,1)	
Collocazione	Associa date apparecchiature alla sala di appartenenza	Sala (1,N) Apparecchiature (1,1)	
LuogoPiscine	Associa le piscine al centro fitness di appartenenza	Centro (1,N) Piscine (1,1)	
Svolgimento	Associa le attività acquatiche alle piscine in cui possono essere svolte	Piscine (1,N) AttivitaAcquatiche (1,N)	
Apertura	Associa ad ogni centro fitness i relativi orari di apertura	Centro (1,N) Orario (1,N)	
2.2.2			
Impiego	Associa le informazioni di ogni dipendente col centro fitness in cui lavora	Dipendenti (1,N) Centro (1,N)	Attivita
Frequenta	Associa le informazioni di ogni cliente con i centri fitness che frequenta	Centro (1,N) Cliente (1,N)	
2.2.3			
Locazione	Associa ogni contratto con i centri fitness per i quali è valido	Centro (1,N) Contratto (1,N)	
Tipologia	Associa ogni contratto con il tipo scelto	Contratto (1,1) TipoContratto (1,N)	
Pagamento	Associa ogni contratto con le rate pagate o da pagare	Contratto (1,N) Rata (1,1)	



Sottoscrizione	Associa ogni contratto con il cliente che lo ha sottoscritto	Contratto (1,1) Cliente (0,1)	
Obiettivo	Associa ogni cliente con lo scopo per il quale usufruisce dei servizi dell'azienda	Cliente (1,1) Scopo (1,N)	
2.3.1			
Tabella	Associa ad ogni cliente la propria scheda di allenamento, attiva o completata	Cliente (0,N) SchedaAllenamento (1,1)	
Tutor	Associa ad ogni scheda di allenamento il tutor che l'ha compilata	Dipendente (0,N) SchedaAllenamento (1,1)	
Sessioni	Associa ad ogni scheda di allenamento tutte le sessioni che la compongono	SchedaAllenamento (1,N) Sessione (1,1)	
Esercizi	Associa ad ogni sessione tutti gli esercizi che la compongono	Sessione (1,N) Esercizio (0,N)	
SetupEsercizio	Associa per ogni esercizio tutte le attrezzature necessarie a svolgerlo nel caso ve ne siano	Esercizio (0,N) Attrezzatura (0,N)	Regolazione, Quantita
2.3.2			
FormaFisica	Associa un cliente con le sue misurazioni, qualora siano mai state effettuate	Cliente (0,N) StatisticheCliente (1,1)	
Nutrizionista	Associa ogni scheda di alimentazione al medico nutrizionista che l'ha compilata	Dipendente (0,N) SchedaAlimentazione (1,1)	
Nutrizione	Associa ogni scheda di alimentazione il cliente a cui è stata assegnata	SchedaAlimentazione (1,1) Cliente (0,N)	
Alimentazione	Associa ad ogni scheda di alimentazione una dieta da seguire	SchedaAlimentazione (1,1) Dieta (0,N)	

Pietanze	Associa ad ogni dieta i pasti che la compongono	Dieta (0,N) Pasti (1,N)	
2.3.3			
Ingresso	Associa ad ogni cliente ogni accesso da lui effettuato	Cliente (0,N) Accesso (1,1)	
LuogoAccesso	Associa ad ogni accesso il centro fitness presso il quale è avvenuto	Accesso (1,1) Centro (0,N)	
Area	Associa ad ogni spogliatoio gli armadietti che contiene	Armadietto (1,1) Spogliatoio (1,N)	
Posizione	Associa ad ogni spogliatoio il centro di cui fa parte	Spogliatoio (1,1) Centro (1,N)	
2.3.4			
Iscrizione	Associa ad ogni cliente ogni corso che frequenta	Cliente (0,N) Corso (0,N)	Data
Insegnamento	Associa ad ogni corso l'istruttore responsabile	Corso (1,1) Istruttore (0,N)	
LuogoCorso	Associa ad ogni corso la sala in cui si tiene	Corso (1,1) Sala (0,N)	
Date	Associa ad ogni corso per ogni giorno in cui si svolge la fascia oraria	Corso (1,N) Calendario (0,N)	
2.3.5			
OrarioLavoro	Associa ad ogni dipendente tutti i turni lavorativi che svolge	Dipendente (0,N) Turno (1,N)	
2.3.6			
Svolgimento	Associa ad ogni esercizio che dovrebbe essere svolto i dati di come l'esercizio è stato effettivamente svolto	Esercizio (0,1) EsercizioEffettivo (1,1)	
Ripetizioni	Associa all'effettivo svolgimento di un esercizio ad ogni conteggio (associazioni multiple per indicare più	EsercizioEffettivo (1,N) Conteggio (0,N)	

	ripetizioni)/durata in minuti		
Regolazione	Associa un esercizio effettivo ad ogni regolazione effettivamente utilizzata	EsercizioEffettivo (0,N) RegolazioneEffettiva (1,N)	
2.3.8			
LuogoAcquisto	Associa ad ogni acquisto il centro in cui viene effettuato	Acquisto (1,1) Centro (0,N)	
Magazzino	Associa ad ogni centro la quantità di merce disponibile in giacenza	Centro (0,N) Giacenza (1,1)	
Disponibili	Associa ad ogni integratore la sua presenza in giacenza	Integratore (0,N) Giacenza (1,1)	
Forniti	Associa ad ogni ordine il fornitore da cui provengono i prodotti ordinati	Ordine (1,1) Fornitore (0,N)	
Ordinati	Associa ad ogni ordine gli integratori presenti in esso	Ordine (1,N) Integratore (0,N)	
Effettua	Associa ad ogni cliente l'acquisto effettuato	Cliente (0,N) Acquisto (1,1)	
Acquisizione	Associa ad ogni acquisto l'integratore comprato	Acquisto (1,1) Integratore (0,N)	
Commercializzati	Insieme di integratori commercializzati dai vari fornitori	Integratore(0,N) Fornitore(1,N)	Prezzo
2.4.1			
Argomenti	Associa ad ogni utente del forum tutti i suoi interessi	Utente (0,N) Interesse (1,N)	
Account	Associa ad ogni contratto tutte le informazioni che contraddistinguono un utente del forum	Utente (1,1) Contratto (1,1)	
Scrittura	Associa un post all'utente che lo ha scritto	Utente (0,N) Post (1,1)	

Valutazione	Associa una risposta all'utente che l'ha scritta	Utente (0,N) Risposta (1,1)	Voto
Commento	Associa ogni risposta al post al quale si riferisce	Risposta (1,1) Post (0,N)	
Iper testo	Associa ad ogni post tutti i link che sono stati inclusi	Post (0,N) Link (1,1)	
Collegamento	Associa ad ogni thread tutti i post che contiene	Post (1,1) Thread (0,N)	
Argomento	Associa ad ogni area del forum tutti i thread che contiene	Thread (1,1) AreaForum (0,N)	
2.4.2			
Accomunare	Associa ad ogni cerchia tutti gli interessi che la caratterizzano	Interesse (0,N) Cerchia (1,N)	
Conosce	Associa ad ogni cerchia tutti gli utenti che ne fanno parte	Utente (0,N) Cerchia (1,N)	
RichiestaAmicizia	Associa ad ogni utente(destinatario) tutti gli utenti(mittenti) che hanno inviato una richiesta di amicizia	Utente (0,N) Utente (1,1)	
2.4.3			
Proponente	Associa ad ogni sfida le informazioni dell'utente proponente	Utente (0,N) Sfida (1,1)	
TabellaSfida	Associa ad ogni sfida una scheda di allenamento	Sfida (1,1) SchedaAllenamento (0,N)	
DietaSfida	Associa ad ogni sfida una scheda di alimentazione qualora la preveda	Sfida (0,1) SchedaAlimentazione (0,N)	
Partecipante	Associa ad ogni sfida ogni utente che partecipa	Sfida (0,N) Utente (0,N)	
Tema	Associa ad ogni sfida un thread	Sfida (1,1) Thread (0,N)	

LuogoEsercizio	Associa ad ogni esercizio effettuato il centro in cui viene svolto	EsercizioEffettivo (1,1) Centro (0,N)	
TemaPost	Associa ad ogni post il thread nel quale viene pubblicato	Post (1,1) Thread (0,N)	

## 2.3 Regole di vincolo

Di seguito vengono riportate le regole di vincolo.

- R1 Una apparecchiatura può appartenere ad un solo centro per volta.
- R2 Un apparecchio non può essere utilizzato se il suo livello usura è 100%.
- R3 Ciascun dipendente ha un solo responsabile, così come ciascun centro ha un solo direttore.
- R4 Un istruttore non può tenere un corso quando non è in servizio.
- R5 Un turno di lavoro dura al Massimo 8 ore, un dipendente non può essere assegnato a due centri diversi uno stesso giorno, un dipendente non può tenere due corsi in due sale diverse nella stessa fascia oraria dello stesso giorno.
- R6 Un post non può appartenere a più thread.
- R7 Ogni utente del centro fitness può avere un solo account del forum.
- R8 Un utente con contratto scaduto ha l'account necessariamente bloccato.

## 2.4 Ristrutturazione diagramma E-R

I sistemi tradizionali per la gestione delle basi di dati non consentono di rappresentare direttamente una generalizzazione, è necessario dunque trasformare questo costrutto in altri costrutti del modello E-R per i quali esiste una implementazione naturale che non sono altro che le entità e le relazioni.

La generalizzazione presente nel diagramma E-R che deve essere soggetta a una trasformazione è quella relativa a Dipendente.

### Generalizzazione Dipendente

La tecnica che viene utilizzata per eliminare la generalizzazione Dipendente è l'accorpamento dell'entità figlie della generalizzazione nell'entità padre. Si effettua l'eliminazione delle quattro entità 'Responsabile', 'Nutrizionista', 'Direttore' e 'Tutor' e le loro proprietà vengono aggiunte all'entità padre che è 'Dipendente'. Infine viene aggiunto all'entità padre l'attributo 'Tipologia' tramite il quale è possibile definire di quale tipo di dipendente si tratta.

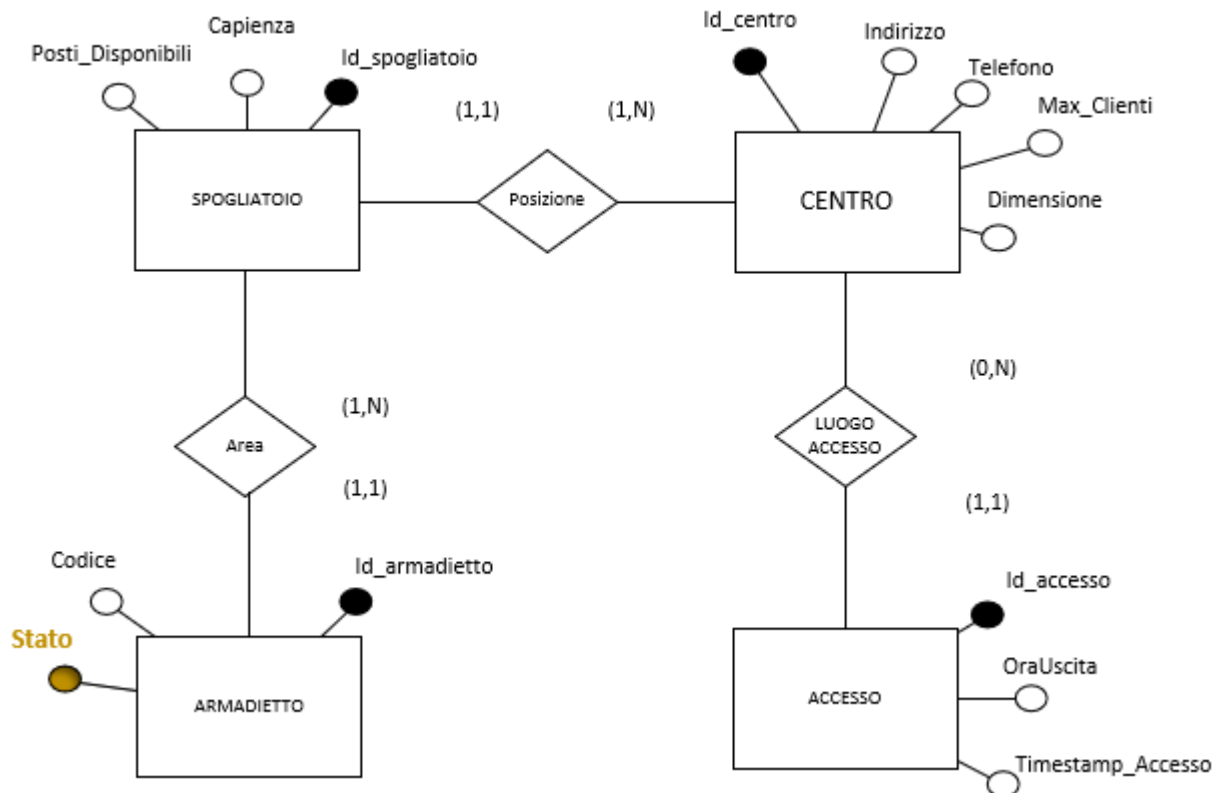
### 3. Individuazione e analisi delle operazioni sui dati

Dopo aver effettuato la ristrutturazione del diagramma E-R sono state individuate 8 operazioni significative:

1. Assegnamento di un armadietto all'accesso di un cliente.
2. Dato IdCentro e Disciplina, se esiste un corso a cui è possibile iscriversi ne visualizza la data di inizio.
3. Per ogni tipologia di contratto restituisce la percentuale che copre dei contratti attualmente validi.
4. Conteggio clienti iscritti per ogni centro fitness, accessi nel mese solare precedente a quello attualmente in corso.
5. Elencare i contratti in scadenza entro due settimane dalla data odierna affinché l'azienda possa contattare il cliente per il rinnovo.
6. Dato un centro, controlla quali apparecchiature sono usurate più del 90% e riporta il nome del centro, della sala e il codice delle apparecchiature in questione.
7. All'inserimento di una nuova valutazione per una risposta viene ricalcolata la credibilità di un utente. Dipendente dal numero dei Post da esso sottoposti al forum e dai voti ricevuti alle risposte ai post, la credibilità di un utente viene calcolata come  $100 * (nPost * sommaVoti)$
8. Alla richiesta di un cliente di un integratore che non è né presente in magazzino né è stato ordinato, viene effettuato l'ordine.

## 3.1 Operazione 1

All'accesso di un cliente in una palestra gli viene assegnato un armadietto libero.



*Tavola dei volumi*

Concetto	Tipo	Volume
Centro	E	20
Accesso	E	10.000.000
Spogliatoio	E	80
Armadietto	E	800
Posizione	R	80
Area	R	800
LuogoAccesso	R	10.000.000

4 spogliatoi per centro, 100 armadietti per spogliatoio

Stimando 300 accessi in ogni centro al giorno si hanno circa 2.190.000 accessi l'anno.

Si prevede la memorizzazione di 10.000.000 di accessi.

### *Tavola degli accessi*

Concetto	Costrutto	Accessi	Tipo
Centro	Entita	1	L
Accesso	Entita	3	L
Spogliatoio	Entita	1	L
Armadietto	Entita	1	S
Posizione	Relazione	1	L
Area	Relazione	1	L

### *Analisi costo-beneficio sull'introduzione della ridondanza*

L'introduzione di una ridondanza nella tabella Armadietto, contenente per ogni armadietto lo **Stato** attuale cioè *Libero* o *Occupato*, sembrerebbe un buon metodo per ridurre il costo dell'operazione di assegnamento di un armadietto libero al momento di un nuovo accesso in un centro fitness. Si prevede un trigger per mantenere aggiornata la ridondanza.

Considerando un volume giornaliero di accessi in un centro pari a 300, si ha che in un anno avvengono tra i vari centri fitness 2.190.000 accessi. Stimando una memorizzazione di circa 10.000.000 record nella tabella *Accessi*, rende l'operazione molto costosa. Con l'aggiunta della ridondanza non è più necessario accedere alla tabella *Accessi*.

Il trigger agisce al momento di un UPDATE nella tabella Accessi e verifica se l'attributo OraUscita è diverso da NULL, in caso imposta la ridondanza *Stato* = "libero".

### *Tavola degli accessi del trigger*

Concetto	Costrutto	Accessi	Tipo
Armadietto	Entita	1	S

### *Tavola degli accessi dopo l'aggiunta della ridondanza Stato*

Concetto	Costrutto	Accessi	Tipo
Centro	Entita	1	L
Spogliatoio	Entita	1	L
Armadietto	Entita	1	S
Posizione	Relazione	1	L
Area	Relazione	1	L

In seguito a queste considerazioni, l'introduzione della ridondanza risulta molto vantaggiosa e pertanto la ridondanza viene effettivamente implementata.



## Operazione 1 senza ridondanza:

```
DROP TRIGGER IF EXISTS AssegnaArmadietto;
DELIMITER $$
CREATE TRIGGER AssegnaArmadietto
AFTER INSERT ON Accessi FOR EACH ROW
BEGIN

    SELECT a.IdArmadietto, a.IdSpogliatoio into @Armadietto, @Spogliatoio
    FROM armadietto a inner join spogliatoio s on s.IdSpogliatoio = a.IdSpogliatoio
        inner join centro ce on ce.IdCentro = s.IdCentro
    where a.IdArmadietto IN (SELECT a1.IdArmadietto
                            FROM accessi a1
                            WHERE a1.OraUscita is not null AND
                                a1.TimestampAccesso = (SELECT MAX(a2.TimestampAccesso)
                                                        FROM accessi a2
                                                        WHERE a1.IdArmadietto = a2.IdArmadietto)

                                )

    OR 0 = (
        SELECT COUNT(*)
        FROM accessi a3
        WHERE a3.IdArmadietto = a.IdArmadietto
        )

    AND s.IdCentro = new.IdCentro
LIMIT 1 ;

UPDATE Armadietto
set stato = "occupato"
where a.IdArmadietto = @Armadietto;

UPDATE Armadietto
set Armadietto = @Armadietto;

END$$
DELIMITER ;
```

### Operazione 1 con ridondanza Stato:

```
DROP TRIGGER IF EXISTS AssegnaArmadiettoRidondanza;
DELIMITER $$
CREATE TRIGGER AssegnaArmadiettoRidondanza
AFTER INSERT ON Accessi FOR EACH ROW
BEGIN

    SELECT a.IdArmadietto, a.IdSpogliatoio into @Armadietto, @Spogliatoio
    FROM armadietto a inner join spogliatoio s on s.IdSpogliatoio = a.IdSpogliatoio
        inner join centro ce on ce.IdCentro = s.IdCentro
    where a.stato = "libero"
        AND s.IdCentro = new.IdCentro
    LIMIT 1 ;

    UPDATE Armadietto
    set stato = "occupato", Armadietto = @Armadietto
    where a.IdArmadietto = @Armadietto;

END$$
DELIMITER ;
```

### Trigger che gestisce la ridondanza Stato:

```
DROP TRIGGER IF EXISTS RIDONDANZA_Stato;

DELIMITER $$

CREATE TRIGGER RIDONDANZA_Stato
AFTER UPDATE ON Accessi FOR EACH ROW
BEGIN

    IF new.OraUscita IS NOT NULL THEN

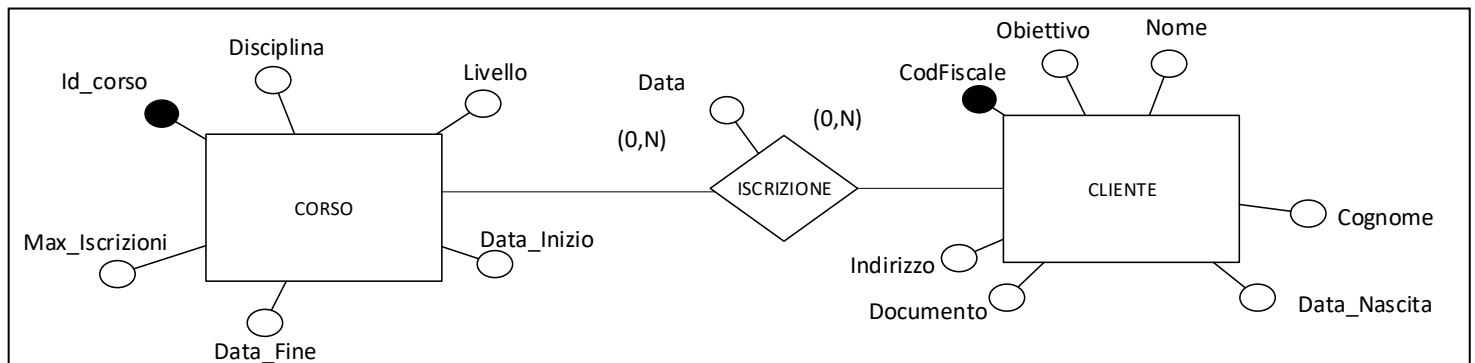
        UPDATE Armadietto
        SET Stato = "libero"
        WHERE IdArmadietto = new.IdArmadietto;

    END IF;

END$$
DELIMITER ;
```

## 3.2 Operazione 2

Dato IdCentro e Disciplina, se esiste un corso a cui è possibile iscriversi ne visualizza la data di inizio



### Tavola dei volumi

Concetto	Tipo	Volume
Corso	E	3.000
Iscrizione	R	20

Si stimano 3000 corsi memorizzati

### Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
Corso	Entita	2	L
Iscrizione	Relazione	2	L

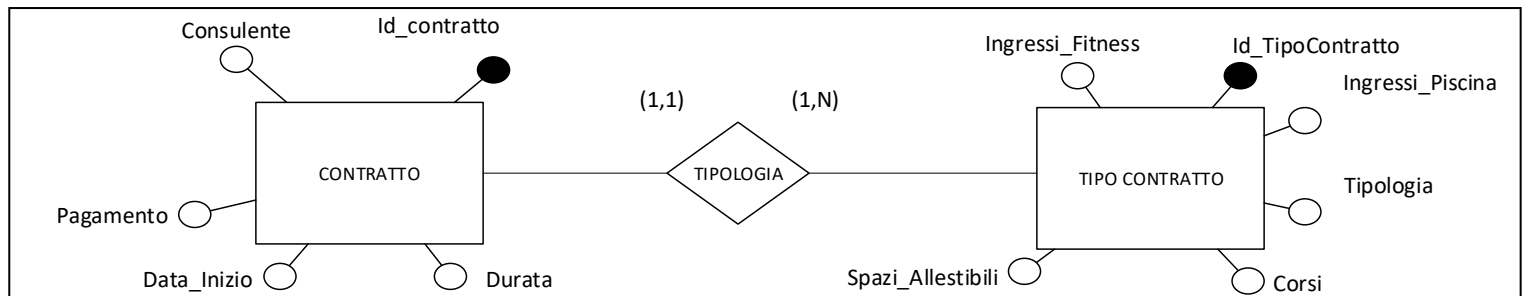
### Operazione 2 in MYSQL

```

DROP PROCEDURE IF EXISTS TrovaCorso;
DELIMITER $$
CREATE PROCEDURE TrovaCorso(IN _Disciplina VARCHAR(45), IN _Centro INT(11))
BEGIN
    SELECT co.IdCorso, co.Disciplina, co.DataInizio
    FROM corso co inner join iscrizioni i on i.IdCorso = co.IdCorso
        inner join sala s on co.IdSala = s.IdSala inner join centro ce on ce.IdCentro=s.IdCentro
    where co.Disciplina = _Disciplina
        AND ce.IdCentro = _Centro
        AND co.DataInizio > current_date
        AND co.MaxIscrizioni > (SELECT COUNT(*) as num
                                FROM corso col inner join iscrizioni i1
                                    ON i1.IdCorso = col.IdCorso
                                WHERE col.IdCorso = co.IdCorso
                                );
END $$
DELIMITER ;
    
```

### 3.3 Operazione 3

Per ogni tipologia di contratto restituisce la percentuale che copre dei contratti attualmente validi.



#### Tavola dei volumi

Concetto	Tipo	Volume
Contratto	E	2.500
TipoContratto	E	5
Tipologia	R	2.500

#### Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
Contratto	Entita	2	L
TipoContratto	Entita	1	L
Tipologia	Relazione	1	L

#### Operazione 3 in MYSQL

```

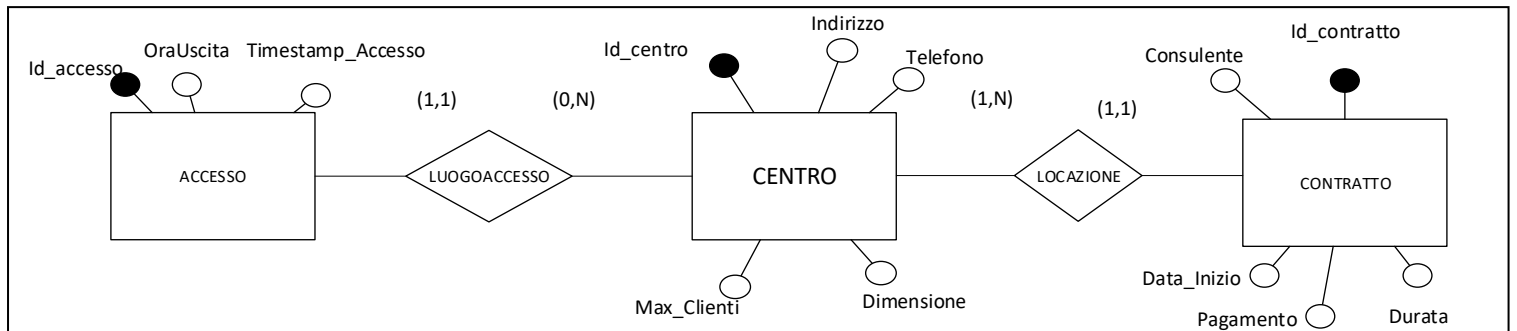
DROP PROCEDURE IF EXISTS PercentualeContratto;
DELIMITER $$
CREATE PROCEDURE PercentualeContratto()
BEGIN
    SELECT count(*) INTO @nContratti
    FROM contratto co
    where DATE_ADD(co.DataInizio, INTERVAL co.Durata MONTH) > current_date;

    SELECT tc.IdTipoContratto, tc.Nome, (100* (COUNT(*) / @nContratti)) as PercentualeIscritti
    FROM contratto co inner join tipocontratto tc on tc.IdTipoContratto = co.tipo
    where DATE_ADD(co.DataInizio, INTERVAL co.Durata MONTH) > current_date
    GROUP BY tc.IdTipoContratto;
END $$
DELIMITER ;

```

## 3.4 Operazione 4

Conteggio clienti attualmente iscritti per ogni centro fitness e numero di accessi nel mese solare precedente a quello attualmente in corso.



### Tavola dei volumi

Concetto	Tipo	Volume
Centro	E	20
Contratto	E	2.500
Accesso	E	10.000.000
Locazione	R	2.500
LuogoAccesso	R	10.000.000

In media 300 accessi al giorno per centro

Si lasciano memorizzati fino a 10 anni, stimati per 10 milioni di accessi

### Tavola degli accessi

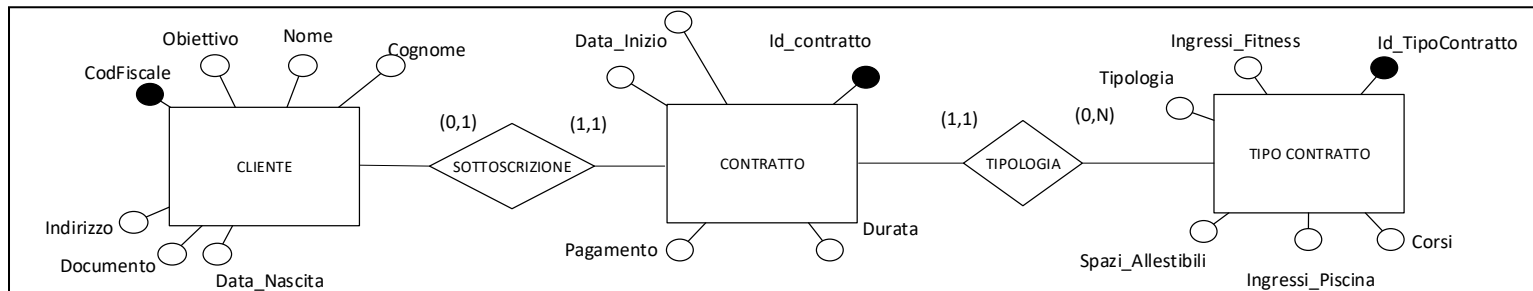
Concetto	Costrutto	Accessi	Tipo
Centro	Entita	1	L
Contratto	Entita	1	L
Accesso	Entita	1	L
Locazione	Relazione	1	L
LuogoAccesso	Relazione	1	L

## Operazione 4 in MYSQL

```
DROP PROCEDURE IF EXISTS AnalisiFlussiClienti;
DELIMITER $$
CREATE PROCEDURE AnalisiFlussiClienti()
BEGIN
    SELECT co.IdCentro, count(*) as Iscritti, (SELECT count(*)
        FROM accessi a
        where a.IdCentro = ce.IdCentro
        AND year(a.TimestampAccesso) = year(DATE_SUB(current_date, INTERVAL 1 MONTH))
        AND month(a.TimestampAccesso) = month(DATE_SUB(current_date, INTERVAL 1 MONTH))
        ) as AccessiMesePrecedente
    FROM contratto co INNER JOIN centro ce on ce.IdCentro = co.IdCentro
    where DATEDIFF(CURRENT_DATE, DATE_ADD(co.DataInizio, INTERVAL co.Durata MONTH)) > 0
    group by co.IdCentro;
END $$
DELIMITER ;
```

## 3.5 Operazione 5

Elencare i contratti in scadenza entro 30 giorni dalla data odierna affinché l'azienda possa contattare il cliente per il rinnovo.



### Tabella dei volumi

Concetto	Tipo	Volume
Cliente	E	2.000
Contratto	E	2.500
TipoContratto	E	5
Sottoscrizione	R	10.000
Collocazione	R	2500

### Tabella degli accessi

Concetto	Costrutto	Accessi	Tipo
Cliente	Entita	1	L
Contratto	Entita	1	L
TipoContratto	Entita	1	L
Sottoscrizione	Relazione	1	L
Collocazione	Relazione	1	L

### Operazione 5 in MYSQL

```

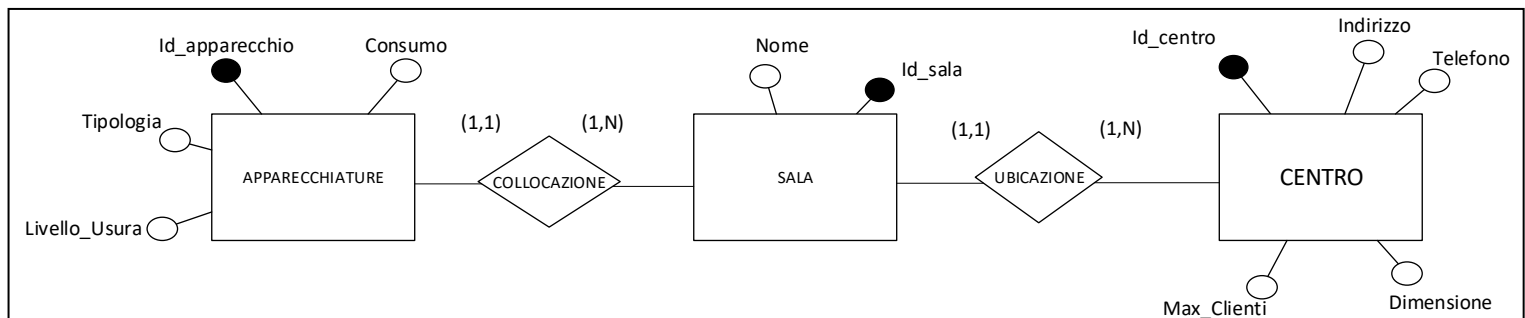
DROP PROCEDURE IF EXISTS CheckContratti;
DELIMITER $$
CREATE PROCEDURE CheckContratti()
BEGIN
    SELECT co.IdContratto, co.CodFiscale, cl.Nome, cl.Cognome, cl.Indirizzo,
           DATE_ADD(co.DataInizio, INTERVAL co.Durata MONTH) as Scadenza, tc.Nome as Suggestito
    FROM contratto co INNER JOIN cliente cl on cl.CodFiscale = co.CodFiscale
    INNER JOIN tipocontratto tc on tc.IdTipoContratto = co.Tipo
    WHERE DATEDIFF(DATE_ADD(co.DataInizio, INTERVAL co.Durata MONTH), CURRENT_DATE)
           BETWEEN 0 AND 30;

END $$
DELIMITER ;

```

## 3.6 Operazione 6

Dato un centro, controlla quali apparecchiature sono usurate più del 80% e riporta l'id della sala, il codice delle apparecchiature in questione assieme al loro nome.



*Tabella dei volumi*

Concetto	Tipo	Volume
Centro	E	20
Sala	E	100
Apparecchiature	E	3000
Ubicazione	R	100
Collocazione	R	3000

Ogni centro ha in media 5 sale.

Ogni sala ha in media 30 apparecchiature.

*Tabella degli accessi*

Concetto	Costrutto	Accessi	Tipo
Centro	Entita	1	L
Sala	Entita	1	L
Apparecchiature	Entita	1	L
Ubicazione	Relazione	1	L
Collocazione	Relazione	1	L

*Operazione 6 in MYSQL*

```

DROP PROCEDURE IF EXISTS ReportUsura;
DELIMITER $$
CREATE PROCEDURE ReportUsura(IN _Centro INT(11))
BEGIN

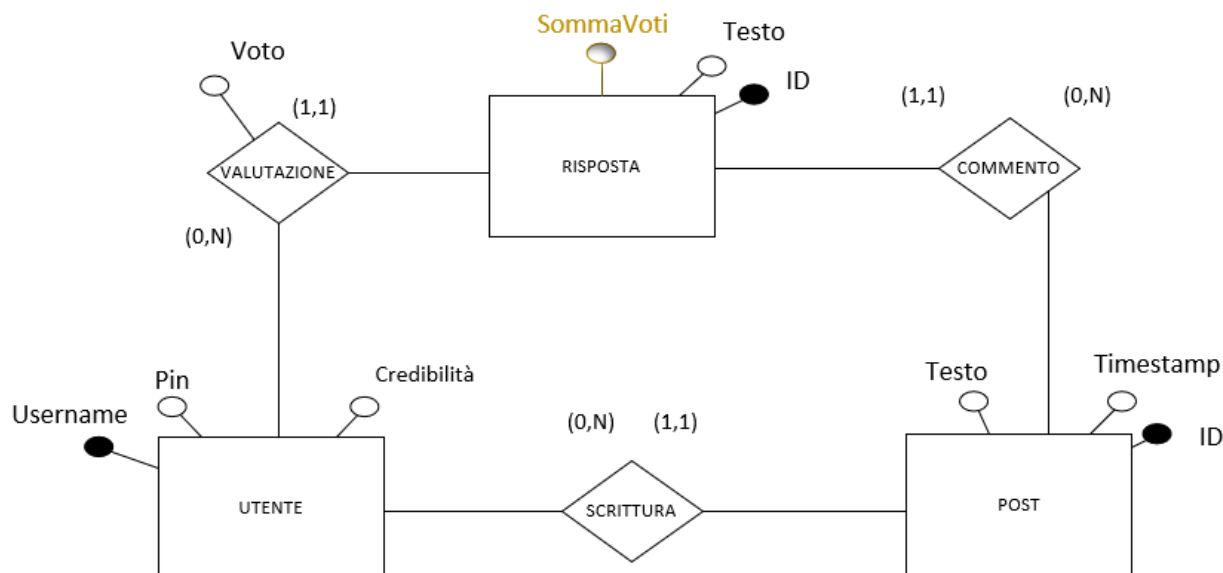
SELECT s.IdSala, a.IdApparecchiatura, a.Nome , a.LivelloUsura
FROM centro c INNER JOIN sala s ON s.IdCentro = c.IdCentro
    INNER JOIN apparecchiature a ON a.IdSala = s.IdSala
WHERE a.LivelloUsura > 80
    AND c.IdCentro = _Centro
ORDER BY a.LivelloUsura;

END $$
DELIMITER ;
    
```



## 3.1 Operazione 7

All'inserimento di una nuova valutazione per una risposta viene ricalcolata la credibilità di un utente. Dipendente dal numero dei Post da esso sottoposti al forum e dai voti ricevuti alle risposte ai post, viene calcolata come  $100 * (nPost * sommaVoti)$



*Tabella dei volumi*

Concetto	Tipo	Volume
Utente	E	2.000
Post	E	100.000
Risposta	E	200.000
Valutazione	R	2.000.000

Ci sono 2000 utenti registrati al forum  
 Ogni utente ha scritto in media a 50 post  
 Ogni utente ha risposto in media a 100 post  
 Ogni utente ha lasciato in media 1000 valutazione a risposte

*Tabella degli accessi*

Concetto	Costrutto	Accessi	Tipo
Utente	Entità	1	S
Post	Entità	1	L
Risposta	Entità	2	L
Valutazione	Relazione	1	L

### ***Analisi costo–beneficio sull'introduzione di una ridondanza***

L'introduzione di una ridondanza nella tabella *Risposta*, contenente per ogni *Risposta* la somma delle valutazioni ricevuto fino ad ora, sembrerebbe un buon metodo per ridurre il costo di questa operazione.

Tramite un trigger si mantiene aggiornata questa ridondanza, incrementando l'attributo ***SommaVoti*** dopo ogni valutazione inviata alla relativa risposta. Considerato un inserimento medio di 500 valutazioni al giorno, si ha un numero di accessi pari a 2.300.000. Inserendo la ridondanza l'operazione viene eseguita senza accedere alla tabella *Valutazione* con conseguente riduzione del carico applicativo.

### ***Tabella degli accessi del trigger***

Concetto	Costrutto	Accessi	Tipo
Risposta	Entità	1	L

### ***Tabella degli accessi dell'operazione in presenza della ridondanza***

Concetto	Costrutto	Accessi	Tipo
Risposta	Entità	2	L
Post	Entità	1	L
Utente	Entità	1	S

In seguito a questa analisi, l'utilizzo della ridondanza risulta molto vantaggioso in quanto permette di evitare l'accesso alla tabella *Valutazione* e pertanto verrà introdotta.

### **Operazione in assenza di ridondanza:**

```
DROP TRIGGER IF EXISTS CalcoloCredibilita;
DELIMITER $$
CREATE TRIGGER CalcoloCredibilita
AFTER INSERT ON Valutazione FOR EACH ROW
BEGIN

    SELECT DISTINCT R.Username INTO @nome
    FROM Risposta R
    WHERE R.IdRisposta=NEW.IdRisposta;

    SELECT SUM(V.Voto) INTO @sommaVoti
    FROM Valutazione V inner join Risposta R
        on R.IdRisposta=V.IdRisposta
    WHERE R.Username= @nome;

    SELECT COUNT(*) INTO @nPost
    FROM Utente u inner join Post p
        on p.Username = u.Username
    where p.Username = @Nome;

    UPDATE Utente
        SET Credibilita= (@nPost * @sommaVoti) / 100
    WHERE Nickname = @nome;

END$$
DELIMITER ;
```

### Operazione in presenza della ridondanza SommaVoto:

```
DROP TRIGGER IF EXISTS CalcoloCredibilitaRidondanza;
DELIMITER $$
CREATE TRIGGER CalcoloCredibilitaRidondanza
AFTER INSERT ON Valutazione FOR EACH ROW
BEGIN

    SELECT DISTINCT R.Username, R.sommaVoti INTO @nome, @sommaVoti
    FROM Risposta R
    WHERE R.IdRisposta=NEW.IdRisposta;

    SELECT COUNT(*) INTO @nPost
    FROM Utente u inner join Post p
        on p.Username = u.Username
    where p.Username = @Nome;

    UPDATE Utente
        SET Credibilita= (@nPost * @sommaVoti) / 100
    WHERE Nickname = @nome;

END$$
DELIMITER ;
```

### Trigger per l'aggiornamento della ridondanza *sommaVoto*:

```
DROP TRIGGER IF EXISTS RIDONDANZAsommaVoto;
DELIMITER $$
CREATE TRIGGER RIDONDANZAsommaVoto
AFTER INSERT ON Valutazione FOR EACH ROW
BEGIN
    UPDATE Risposta
        SET sommaVoto = sommaVoto + new.Voto
    WHERE IdRisposta = NEW.IdRisposta;

END$$
DELIMITER ;
```

### 3.8 Operazione 8

Alla richiesta di un cliente di un integratore che non è né presente in magazzino né è stato ordinato, viene effettuato l'ordine.

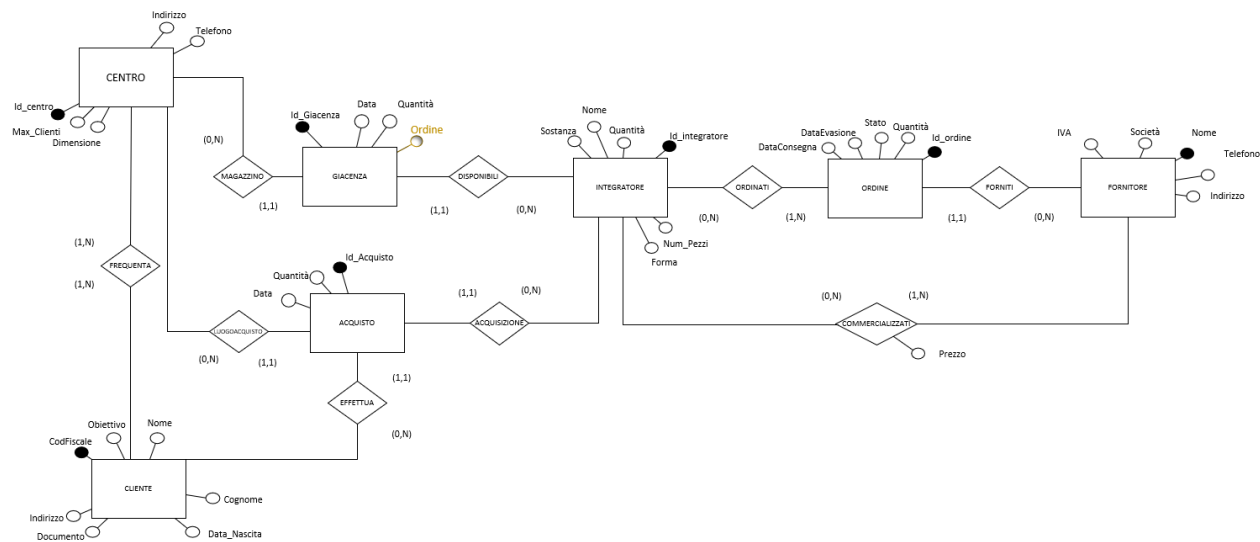


Tabella dei volumi

Concetto	Tipo	Volume
Centro	E	20
Cliente	E	2000
Integratore	E	50
Fornitore	E	10
Giacenza	R	20.000
Acquisto	R	720.000
Ordine	R	720.000
Commercializzati	R	2.000

Ogni fornitore commercializza in media 40 su 50 integratori

Un cliente su 4 frequenta due centri anziché uno solo

## Tabella degli accessi

Concetto	Costrutto	Accessi	Tipo
Cliente	Entita	1	L
Integratore	Entita	2	L
Fornitore	Entita	1	L
Giacenza	Relazione	1	L
Ordine	Relazione	2	S
Commercializzati	Relazione	1	L

## Operazione 8 in MYSQL

```

DROP TRIGGER IF EXISTS CheckIntegratore;
DELIMITER $$
CREATE TRIGGER CheckIntegratore
BEFORE INSERT ON Acquisto FOR EACH ROW
BEGIN
    -- nome integratore, centro in cui sto acquistando
    SELECT DISTINCT a.nome, a.centro INTO @nome, @Centro
    FROM acquisto a INNER JOIN integratore i ON i.IdIntegratore = a.IdIntegratore
    WHERE a.IdIntegratore = new.IdIntegratore;

    -- conto integratori in magazzino
    SELECT count(*) INTO @nPresenti
    FROM giacenza g
    where g.IdIntegratore = new.IdIntegratore
        AND g.Centro = @Centro;

    IF @nPresenti = 0 THEN

        SELECT count(*) INTO @nOrdinati
        FROM ordine o INNER JOIN integratore i ON i.IdIntegratore = o.IdIntegratore
        where o.IdIntegratore = new.IdIntegratore
            AND o.stato = "incompleto";

        IF @nOrdinati = 0 THEN -- allora ordino

            SELECT f.Nome INTO @Fornitore
            FROM fornitore f INNER JOIN commercializzati c ON f.nome=c.nome
                INNER JOIN integratore i ON i.IdIntegratore = c.IdIntegratore
            where i.IdIntegratore = new.IdIntegratore
                AND c.prezzo = (select MIN(c1.prezzo)
                    from commercializzati c1 INNER JOIN integratore i1
                    on i1.IdIntegratore = c1.IdIntegratore
                    where c1.IdIntegratore = new.IdIntegratore);

            SELECT o.IdOrdine INTO @Ordine
            FROM ordine o
            WHERE o.stato = "incompleto";

            INSERT INTO Ordine
            VALUES (@Ordine, new.IdIntegratore, @Nome, "incompleto", 30, @Centro, NULL, @Centro);

            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = "L'integratore non era presente in magazzino. È stato ordinato. ";

        END IF;
    END IF;
END$$
DELIMITER ;

```

## 4. Progettazione logica

In questa fase della progettazione, il modello diagramma viene tradotto nel modello logico relazionale, producendo così lo schema logico del database, evidenziandone anche i vincoli di integrità.

### 4.1 Schema logico

Centro (**IdCentro**, Indirizzo, Telefono, Dimensione, MaxClienti)

Sala (**IdSala**, IdCentro, Nome)

Apparecchiature (**IdApparecchiature**, IdSala, Nome, Tipologia, Consumo, LivelloUsura)

Cliente (**CodFiscale**, Nome, Cognome, DataNascita, Indirizzo, Documento, Prefettura, Scopo)

Dipendente (**CodFiscale**, Nome, Cognome, Tipologia, DataNascita, Tipologia, Indirizzo, Documento, Prefettura)

Impiego (**IdCentro**, **Codice**, Attivita)

Contratto (**IdContratto**, CodFiscale, IdCentro, Consulente, Tipo, DataInizio, Durata, Pagamento)

TipoContratto (**IdTipoContratto**, Nome, Prezzo, IngressiFitness, DimensionePiscina, IngressiPiscina, Corsi)

Piscina (**IdPiscina**, IdCentro, Nome, Dimensione)

AttivitaAcquatica (**IdAttivita**, Nome)

Scopo (**CodFiscale**, Muscolo)

Rata (**IdRata**, IdContratto, Importo, Scadenza, Stato)

Esercizio (**IdEsercizio**, IdSessione, Nome, DispendioEnergetico, Ripetizioni, Conteggio, TempoRecupero)

SetupEsercizio (**IdEsercizio**, **IdAttrezzatura**, Regolazione, Quantita)

SchedaAlimentazione (**IdSchedaAlimentazione**, IdNutrizionista, CodFiscale, DataInizio, DataFine, Intervallo)

Dieta (**IdDieta**, IdSchedaAlimentazione, Kcal, NumeroPasti)

StatisticheCliente (**IdStatisticheCliente**, IdCliente, Peso, MassaMagra, MassaGrassa, Acqua)

Pasto (**IdPasto**, IdDieta, Pasto)

Accesso (**IdAccesso**, IdCliente, IdCentro, TimestampAccesso, OraUscita)

Armadietto (**IdArmadietto**, IdAccesso, IdSpogliatoio, Codice)

Spogliatoio (**IdSpogliatoio**, Capienza, PostiDisponibili, Posizione)

Corso (**IdCorso**, IdSala, Disciplina, Livello, DataInizio, DataFine, MaxIscrizioni)

Calendario (**IdCalendario**, IdCorso, Giorno, Inizio, Fine)

Turno (**IdTurno**, Giorno, Inizio, Fine)

SchedaAllenamento (**IdSchedaAllenamento**, CodFiscale, DataInizio, DataFine, Tutor)

EsercizioEffettivo (**IdEsercizioEffettivo**, IdEsercizio, Regolazione, TempoRecupero, TempoEsecuzione)

RegolazioneEffettiva (**IdRegolazioneEffettiva**, IdEsercizioEffettivo, Regolazione, Quantita)

Conteggio (**IdConteggio**, IdEsercizioEffettivo, Conteggio, IstanteInizio, IstanteFine)

Integratore (**IdIntegratore**, Nome, Sostanza, Quantita, NumPezzi, Forma)

Fornitore (**IdFornitore**, Nome, Societa, IVA, Indirizzo, Telefono)

Ordine (**IdOrdine**, Stato, DataEvasione, Quantita, DataConsegna)

Commercializzati (**IdIntegratore**, **NomeFornitore**, Prezzo)

Giacenza (**IdIntegratore**, **IdCentro**, Quantita)

Utente (**Username**, Karma, Pin, Credibilita)

Interesse (**IdInteresse**, IdUtente, Ambito)

AreaForum (**IdAreaForum**, Nome)

Thread (**IdThread**, IdAreaForum, Nome)

Post (**IdPosto**, IdThread, IdUtente, Testo, Timestamp)

Link (**IdLink**, IdPost, URL)

Risposta (**IdRisposta**, IdPost, IdUtente, SommaVoti)

Valutazione (**IdValutazione**, IdRisposta, Username, Voto)

Sfida (**IdSfida**, DataLancio, DataInizio, DataFine, Proponente, Thread, Scopo, IdSchedaAllenamento, IdSchedaAlimentazione)

Partecipante (**IdSfida**, **Username**)

Cerchia (**IdCerchia**, IdUtente)

Attrezzatura (**IdAttrezzo**, Nome)

OrarioLavoro (**CodFiscale**, IdTurno, IdCentro)

Orario (**IdOrario**, OraInizio, OraFine, IdCentro)

Insegnamento (**IdCorso**, **CodFiscale**)

Iscrizioni (**IdCorso**, **CodFiscale**, DataIscrizione)

RichiestaAmicizia (**Mittente**, **Destinatario**, Stato)

## 4.2 Specifica dei vincoli di integrità referenziale

Precedentemente abbiamo individuato le tabelle della base di dati a partire dallo schema E-R. Tali tabelle per poter contenere informazioni plausibile devo necessariamente verificare una serie di vincoli di integrità referenziale.

Qui di seguito verranno elencati tutti i vincoli di integrità referenziale presenti nella base di dati che abbiamo creato.

- Esiste un vincolo di integrità referenziale tra l'attributo *IdCentro* della tabella *Centro* e l'attributo *IdCentro* della tabella *Sala*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdSala* della tabella *Sala* e l'attributo *IdSala* della tabella *Apparecchiature*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdCentro* della tabella *Centro* e l'attributo *IdCentro* della tabella *Piscina*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdCentro* della tabella *Centro* e l'attributo *IdCentro* della tabella *Orario*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdCentro* della tabella *Centro* e l'attributo *IdCentro* della tabella *Impiego*.
- Esiste un vincolo di integrità referenziale tra l'attributo *Codice* della tabella *Impiego* e l'attributo *Codice* della tabella *Dipendente*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdCentro* della tabella *Centro* e l'attributo *IdCentro* della tabella *Impiego*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdCentro* della tabella *Centro* e l'attributo *IdCentro* della tabella *Impiego*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdTipoContratto* della tabella *TipoContratto* e l'attributo *Tipo* della tabella *Contratto*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdTipoContratto* della tabella *TipoContratto* e l'attributo *Tipo* della tabella *Contratto*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdContratto* della tabella *Rata* e l'attributo *IdContratto* della tabella *Contratto*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdTipoContratto* della tabella *TipoContratto* e l'attributo *Tipo* della tabella *Contratto*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodFiscale* della tabella *Contratto* e l'attributo *CodFiscale* della tabella *Cliente*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodFiscale* della tabella *Scopo* e l'attributo *CodFiscale* della tabella *Cliente*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodFiscale* della tabella *SchedaAllenamento* e l'attributo *CodFiscale* della tabella *Cliente*.



- Esiste un vincolo di integrità referenziale tra l'attributo *CodFiscale* della tabella *Dipendente* e l'attributo *Tutor* della tabella *SchedaAllenamento*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodFiscale* della tabella *SchedaAlimentazione* e l'attributo *CodFiscale* della tabella *Cliente*.
- Esiste un vincolo di integrità referenziale tra l'attributo *Codice* della tabella *Dipendente* e l'attributo *IdNutrizionista* della tabella *SchedaAlimentazione*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdSchedaAllenamento* della tabella *SchedaAllenamento* e l'attributo *IdSchedaAllenamento* della tabella *Sessione*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdSessione* della tabella *Esercizio* e l'attributo *IdSessione* della tabella *Sessione*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdAttrezzatura* della tabella *SetupEsercizio* e l'attributo *IdAttrezzatura* della tabella *Attrezzatura*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdEsercizio* della tabella *SetupEsercizio* e l'attributo *IdEsercizio* della tabella *Esercizio*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodFiscale* della tabella *StatisticheCliente* e l'attributo *CodFiscale* della tabella *Cliente*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodFiscale* della tabella *Dipendente* e l'attributo *IdNutrizionista* della tabella *SchedaAlimentazione*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodFiscale* della tabella *Cliente* e l'attributo *Cliente* della tabella *SchedaAlimentazione*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdSchedaAlimentazione* della tabella *Dieta* e l'attributo *IdSchedaAlimentazione* della tabella *SchedaAlimentazione*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdDieta* della tabella *Dieta* e l'attributo *IdDieta* della tabella *Pasto*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdCliente* della tabella *Accesso* e l'attributo *CodFiscale* della tabella *Cliente*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdCentro* della tabella *Accesso* e l'attributo *IdCentro* della tabella *Centro*.
- Esiste un vincolo di integrità referenziale tra l'attributo *Armadietto* della tabella *Accesso* e l'attributo *IdArmadietto* della tabella *Armadietto*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdSpogliatoio* della tabella *Accesso* e l'attributo *IdSpogliatoio* della tabella *Spogliatoio*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdCentro* della tabella *Spogliatoio* e l'attributo *IdCentro* della tabella *Centro*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodFiscale* della tabella *Iscrizione* e l'attributo *CodFiscale* della tabella *Cliente*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdCorso* della tabella *Iscrizione* e l'attributo *IdCorso* della tabella *Corso*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodFiscale* della tabella *Dipendente* e l'attributo *CodFiscale* della tabella *Insegnamento*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdCorso* della tabella *Insegnamento* e l'attributo *IdCorso* della tabella *Corso*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdCorso* della tabella *Calendario* e l'attributo *IdCorso* della tabella *Corso*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdTurno* della tabella *OrarioLavoro* e l'attributo *IdTurno* della tabella *Turno*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodFiscale* della tabella *OrarioLavoro* e l'attributo *IdTurno* della tabella *Turno*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdEsercizio* della tabella *EsercizioEffettivo* e l'attributo *IdEsercizio* della tabella *Esercizio*.

- Esiste un vincolo di integrità referenziale tra l'attributo *IdEsercizioEffettivo* della tabella *EsercizioEffettivo* e l'attributo *IdEsercizioEffettivo* della tabella *Conteggio*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdEsercizioEffettivo* della tabella *EsercizioEffettivo* e l'attributo *IdEsercizioEffettivo* della tabella *RegolazioneEffettiva*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdIntegratore* della tabella *Integratore* e l'attributo *IdIntegratore* della tabella *Ordine*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdFornitore* della tabella *Fornitore* e l'attributo *IdFornitore* della tabella *Ordine*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdIntegratore* della tabella *Integratore* e l'attributo *IdIntegratore* della tabella *Giacenza*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdCentro* della tabella *Centro* e l'attributo *IdCentro* della tabella *Giacenza*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdIntegratore* della tabella *Integratore* e l'attributo *IdIntegratore* della tabella *Acquisto*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodFiscale* della tabella *Cliente* e l'attributo *CodFiscale* della tabella *Acquisto*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdIntegratore* della tabella *Integratore* e l'attributo *IdIntegratore* della tabella *Commercializzati*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdFornitore* della tabella *Fornitore* e l'attributo *IdFornitore* della tabella *Commercializzati*.
- Esiste un vincolo di integrità referenziale tra l'attributo *Username* della tabella *Utente* e l'attributo *Username* della tabella *Interesse*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodFiscale* della tabella *Utente* e l'attributo *CodFiscale* della tabella *Contratto*.
- Esiste un vincolo di integrità referenziale tra l'attributo *Username* della tabella *Utente* e l'attributo *Username* della tabella *Post*.
- Esiste un vincolo di integrità referenziale tra l'attributo *Username* della tabella *Utente* e l'attributo *Username* della tabella *Risposta*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdPost* della tabella *Risposta* e l'attributo *IdPost* della tabella *Post*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdThread* della tabella *Thread* e l'attributo *IdThread* della tabella *Post*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdAreaForum* della tabella *Thread* e l'attributo *IdAreaForum* della tabella *Thread*.
- Esiste un vincolo di integrità referenziale tra l'attributo *Username* della tabella *Utente* e l'attributo *IdUtente* della tabella *Cerchia*.
- Esiste un vincolo di integrità referenziale tra l'attributo *Username* della tabella *Utente* e l'attributo *Username* della tabella *Interesse*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdInteresse* della tabella *Cerchia* e l'attributo *IdInteresse* della tabella *Interesse*.
- Esiste un vincolo di integrità referenziale tra l'attributo *Mittente* della tabella *RichiestaAmicizia* e l'attributo *Username* della tabella *Utente*.
- Esiste un vincolo di integrità referenziale tra l'attributo *Destinatario* della tabella *RichiestaAmicizia* e l'attributo *Username* della tabella *Utente*.
- Esiste un vincolo di integrità referenziale tra l'attributo *Proponente* della tabella *Sfida* e l'attributo *Username* della tabella *Utente*.
- Esiste un vincolo di integrità referenziale tra l'attributo *Username* della tabella *Partecipante* e l'attributo *Username* della tabella *Utente*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdSfida* della tabella *Partecipante* e l'attributo *IdSfida* della tabella *Sfida*.

- Esiste un vincolo di integrità referenziale tra l'attributo *IdSchedaAllenamento* della tabella *SchedaAllenamento* e l'attributo *IdSchedaAllenamento* della tabella *Sfida*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdSchedaAlimentazione* della tabella *SchedaAlimentazione* e l'attributo *IdSchedaAlimentazione* della tabella *Sfida*.
- Esiste un vincolo di integrità referenziale tra l'attributo *IdThread* della tabella *Thread* e l'attributo *IdThread* della tabella *Sfida*.

## 4.3 Analisi delle dipendenze funzionali e normalizzazione

Vediamo adesso quali sono le dipendenze funzionali che valgono sulle tabelle che abbiamo individuato.

Tabella Centro

IdCentro → Indirizzo, Telefono, Dimensione, MaxClienti

Tabella Sala

IdSala → IdCentro, Nome

Tabella Apparecchiature

IdApparecchiature → IdSala, Nome, Tipologia, Consumo, LivelloUsura

Tabella Cliente

CodFiscale → Nome, Cognome, DataNascita, Indirizzo, Documento, Prefettura, Scopo

Tabella Dipendente

CodFiscale → Nome, Cognome, DataNascita, Indirizzo, Documento, Prefettura

Tabella Impiego

IdCentro, Codice → Attivita

Tabella Contratto

IdContratto → CodFiscale, IdCentro, Consulente, Tipo, DataInizio, Durata, Pagamento

Tabella TipoContratto

IdTipoContratto → Nome, Prezzo, IngressiFitness, DimensionePiscina, IngressiPiscina, Corsi

Tabella Piscina

IdPiscina → IdCentro, Nome, Dimensione

Tabella AttivitaAcquatica

IdAttivita → Nome

Tabella Scopo

CodFiscale → Muscolo

Tabella Rata

IdRata → IdContratto, Importo, Scadenza, Stato

Tabella Esercizio

IdEsercizio → IdSessione, Nome, DispendioEnergetico, Ripetizioni, Conteggio, TempoRecupero

Tabella SetupEsercizio

IdEsercizio, IdAttrezzatura → Regolazione, Quantita

Tabella SchedaAlimentazione

IdSchedaAlimentazione → IdNutrizionista, CodFiscale, DataInizio, DataFine, Intervallo

Tabella Dieta

IdDieta → IdSchedaAlimentazione, Kcal, NumeroPasti

Tabella StatisticheCliente

IdStatisticheCliente → IdCliente, Peso, MassaMagra, MassaGrassa, Acqua

Tabella Pasto

IdPasto → IdDieta, Pasto

Tabella Accesso

IdAccesso → IdCliente, IdCentro, TimestampAccesso, OraUscita

Tabella Armadietto

IdArmadietto → IdAccesso, IdSpogliatoio, Codice

Tabella Spogliatoio

IdSpogliatoio → Capienza, PostiDisponibili, Posizione

Tabella Corso

IdCorso → IdSala, Disciplina, Livello, DataInizio, DataFine, MaxIscrizioni

Tabella Calendario

IdCalendario → IdCorso, Giorno, Inizio, Fine

Tabella Turno

IdTurno → Giorno, Inizio, Fine

Tabella SchedaAllenamento

IdSchedaAllenamento → CodFiscale, DataInizio, DataFine, Tutor

Tabella EsercizioEffettivo

IdEsercizioEffettivo → IdEsercizio, Regolazione, TempoRecupero, TempoEsecuzione

Tabella RegolazioneEffettiva

IdRegolazioneEffettiva → IdEsercizioEffettivo, Regolazione, Quantita

Tabella Conteggio

IdConteggio → IdEsercizioEffettivo, Conteggio, IstanteInizio, IstanteFine

Tabella Integratore

IdIntegratore → Nome, Sostanza, Quantita, NumPezzi, Forma

Tabella Fornitore

IdFornitore → Nome, Societa, IVA, Indirizzo, Telefono

Tabella Ordine

IdOrdine → Stato, DataEvasione, Quantita, DataConsegna

Tabella Utente

Username → Karma, Pin, Credibilita

Tabella Interesse

IdInteresse → IdUtente, Ambito

Tabella AreaForum

IdAreaForum → Nome

Tabella Thread

IdThread → IdAreaForum, Nome

Tabella Post

IdPosto → IdThread, IdUtente, Testo, Timestamp

Tabella Link

IdLink → IdPost, URL

Tabella Risposta

IdRisposta → IdPost, IdUtente, SommaVoti

Tabella Valutazione

IdValutazione → IdRisposta, Username, Voto

Tabella Sfida

IdSfida → DataLancio, DataInizio, DataFine, Proponente, Thread, Scopo, IdSchedaAllenamento, IdSchedaAlimentazione

Tabella Partecipante

IdSfida, Username

Tabella Cerchia

IdCerchia → IdUtente

Tabella Attrezzatura

IdAttrezzo → Nome

Tabella OrarioLavoro

CodFiscale → IdTurno, IdCentro

Tabella Orario

IdOrario → OraInizio, OraFine, IdCentro

Tabella Insegnamento

IdCorso, CodFiscale

Tabella Iscrizioni

IdCorso, CodFiscale → DataIscrizione

Tabella RichiestaAmicizia

Mittente, Destinatario → Stato

## 5. Area Analytics

Di seguito vengono riportate le funzionalità che permettono di analizzare i dati per la ricerca di informazioni utili.

### 5.1 Funzionalità 1

#### *Corsi poco frequentati:*

Analizzando le iscrizioni ai corsi dell'ultimo anno si valutano quali Discipline risultano poco frequentate, mostrando quanti clienti si sono iscritti in percentuale rispetto alla capienza massima di iscrizione.

#### *Fasce orarie poco frequentate:*

Analizzando gli accessi dell'ultimo anno si valutano quali fasce orarie sono mediamente frequentate sotto al 2% degli attualmente iscritti al Centro.

#### *Monitor dimensionamento palestra:*

Per dare una valutazione se il dimensionamento del centro fitness è adeguato si analizzano i tempi di esecuzione degli esercizi rispetto alle tempistiche previste.

Utilizzando le formule seguenti si cerca di stimare il dimensionamento di un centro fitness:

$$T = T_{esecuzione} \cdot conteggio + T_{recupero} \cdot ripetizioni$$

$$T_{effettivo} = T_{esecuzioneEFF} \cdot conteggio + T_{recuperoEFF} \cdot ripetizioni$$

$$T - T_{effettivo} = ritardo$$

$$RitardoAnnuale = \frac{ritardo}{persone}$$

In conclusione la valutazione dell'adeguatezza di un centro fitness viene misurata come segue

*If*

RitardoAnnuale > 10 min → Il centro è SOTTODIMENSIONATO

*Else*

Il centro è ADEGUATAMENTE DIMENSIONATO

Questa funzionalità restituisce una valutazione relativa al *Centro* che è stato inserito come parametro.

## Corsi poco frequentati

```
DROP PROCEDURE IF EXISTS ReportCorsi;
DELIMITER $$
CREATE PROCEDURE ReportCorsi()
BEGIN

SELECT col1.Disciplina, s.IdCentro,
       100* (SUM(F.Iscritti) / SUM(col1.MaxIscrizioni)) as PercentualeIscritti
FROM corso col inner join
  (SELECT co.IdCorso, count(*) AS Iscritti
   FROM corso co inner join iscrizioni i on i.IdCorso = co.IdCorso
   GROUP BY co.IdCorso)
  as F on F.IdCorso=col.IdCorso inner join Sala s on s.IdSala=col.IdSala
GROUP BY col1.Disciplina, s.IdCentro;

END $$
DELIMITER ;
```

## Fasce orarie scarsamente frequentate

```
DROP PROCEDURE IF EXISTS ReportFasceOrarie;
DELIMITER $$
CREATE PROCEDURE ReportFasceOrarie()
BEGIN

SELECT B.IdCentro, B.IdOrario, B.OraInizio, B.OraInizio, B.AccessiFascia
FROM (
  SELECT co.IdCentro, count(*)/100 as 2Percent
  FROM contratto co INNER JOIN centro ce on ce.IdCentro = co.IdCentro
  where DATEDIFF(CURRENT_DATE, DATE_ADD(co.DataInizio, INTERVAL co.Durata MONTH)) > 0
  group by co.IdCentro
) as A
INNER JOIN
(
  SELECT o.*, count(*) AS AccessiFascia
  FROM orario o inner join centro ce on ce.IdCentro = o.IdCentro
  inner join accessi a on ce.IdCentro = a.IdCentro
  where timestampdiff(DAY,CURRENT_TIMESTAMP, a.TimestampAccesso) <= 365
  GROUP BY o.IdOrario
) as B
  ON A.IdCentro = B.IdCentro
WHERE AccessiFascia < 2Percent;

END $$
DELIMITER ;
```

## MonitorDimensionamento

```
DROP PROCEDURE IF EXISTS ReportDimensionamento;
DELIMITER $$
CREATE PROCEDURE ReportDimensionamento(IN _Centro INT(11))
BEGIN
-- conteggio iscritti a _Centro
    SELECT count(*) INTO @nIscritti
    FROM contratto co INNER JOIN centro ce on ce.IdCentro = co.IdCentro
    where DATEDIFF(CURRENT_DATE, DATE_ADD(co.DataInizio, INTERVAL co.Durata MONTH)) > 0
        and co.IdCentro = 1
    group by co.IdCentro;

    SELECT (((SUM(e.TempoEsecuzione) *SUM(e.Conteggio)) + (SUM(e.TempoRecupero)*SUM(e.Ripetizioni)))
        - ((SUM(ee.TempoEsecuzione) *SUM(ec.Conteggio)) + (SUM(ee.TempoRecupero)*SUM(e.Ripetizioni))))
        /@nIscritti into @Ritardo
    FROM esercizio e inner join esercizioeffettivo ee on e.IdEsercizio=ee.IdEsercizio
        inner join conteggio ec on ee.IdEsercizioEffettivo=ec.IdEsercizioEffettivo
        inner join apparecchiature a on a.IdApparecchiatura=ee.IdApparecchiatura
        inner join sala s on s.IdSala=a.IdSala
    WHERE s.IdCentro = _Centro
        AND DATEDIFF(CURRENT_DATE, ee.Data) <= 365;

    IF @Ritardo < 600 THEN
        SELECT @Ritardo as RitardoAnnuale, "Centro fitness SOTTODIMENSIONATO" as Responso;
    ELSE
        SELECT @Ritardo as RitardoAnnuale, "Centro fitness ADEGUATO" as Responso;
    END IF;

END $$
DELIMITER ;
```



## 5.2 Funzionalità 2

Per generare un report che rappresenti la performance dei clienti utilizziamo le variazioni del Peso, MassaMagra e MassaGrassa rispetto al mese precedente.

Questo report viene presentato sotto forma di tabella *Score* (Anno, Mese, CodFiscale, Punteggio) nella quale ogni mese verranno aggiunti i nuovi valori ricalcolati da un apposito EVENT.

Il punteggio associato dipende dallo scopo che il cliente ha nel mese in cui viene calcolato, il valore viene ricavato utilizzando diverse formule che tengono conto, appunto, dello scopo del cliente

### Potenziamento Muscolare

$$50 + 10 * \left( \frac{\text{MagraAttuale}}{\text{PesoAttuale}} + \frac{\text{MagraMesePrecedente}}{\text{PesoMesePrecedente}} \right)$$

*Si riesce così ad avere un'idea del miglioramento o peggioramento della massa muscolare del cliente rispetto al mese precedente.*

### Dimagrimento

$$50 + 10 * \left( \frac{\text{GrassaMesePrecedente}}{\text{PesoMesePrecedente}} + \frac{\text{GrassaAttuale}}{\text{PesoAttuale}} \right)$$

*Si riesce così ad avere un'idea del miglioramento o peggioramento della massa magra del cliente rispetto al mese precedente.*

### Ricreativo

$$\begin{aligned} 100 - 10 * [ & \text{ABS} \left( \frac{\text{GrassaMesePrecedente}}{\text{PesoMesePrecedente}} - \frac{\text{GrassaAttuale}}{\text{PesoAttuale}} \right) \\ & + \text{ABS} \left( \frac{\text{MagraAttuale}}{\text{PesoAttuale}} - \frac{\text{MagraMesePrecedente}}{\text{PesoMesePrecedente}} \right) \\ & + \text{ABS}(\text{PesoMesePrecedente} - \text{PesoAttuale}) ] \end{aligned}$$

*In questo caso si ipotizza che il cliente voglia frequentare il/i centri fitness per rimanere semplicemente in forma, cioè così come si era quando si è iscritto. Per questo particolare scopo teniamo di conto delle variazioni di Peso, MassaGrassa e MassaMagra come valori assoluti, in quanto qualsiasi tipo di variazione deve far calare il punteggio.*

## DimagrimentoFunction

```
DROP FUNCTION IF EXISTS PunteggioDimagrimento;
DELIMITER $$
CREATE FUNCTION PunteggioDimagrimento(_Cliente VARCHAR (45))
RETURNS INT(11) DETERMINISTIC
BEGIN
    DECLARE PUNTEGGIO INT(11) DEFAULT 0;
    SET PUNTEGGIO =
        50 + (10*

        (
            SELECT sc1.MassaGrassa
            from statistichecliente sc1
            where sc1.CodFiscale=_Cliente
              AND year(sc1.data) = year(DATE_SUB(current_date, INTERVAL 1 MONTH))
              AND month(sc1.data) = month(DATE_SUB(current_date, INTERVAL 1 MONTH))
        )
        +

        (
            SELECT sc0.Peso
            from statistichecliente sc0
            where sc0.CodFiscale=_Cliente
              AND year(sc0.data) = year(DATE_SUB(current_date, INTERVAL 1 MONTH))
              AND month(sc0.data) = month(DATE_SUB(current_date, INTERVAL 1 MONTH))
        )
        -

        (
            SELECT sc2.MassaGrassa
            from statistichecliente sc2
            where sc2.CodFiscale=_Cliente
              AND year(sc2.data) = year(current_date)
              AND month(sc2.data) = month(current_date)
        )
        +

        (SELECT sc3.Peso
        from statistichecliente sc3
        where sc3.CodFiscale=_Cliente
          AND year(sc3.data) = year(current_date)
          AND month(sc3.data) = month(current_date)
        )
        )
    ;
    RETURN(PUNTEGGIO);
END $$
DELIMITER ;
```

## MuscoloFunction

```
DROP FUNCTION IF EXISTS PunteggioMuscolo;
DELIMITER $$
CREATE FUNCTION PunteggioMuscolo(_Cliente VARCHAR (45))
RETURNS INT(11) DETERMINISTIC
BEGIN
    DECLARE PUNTEGGIO INT(11) DEFAULT 0;
    SET PUNTEGGIO =
        50 + (10*

        (
        ( SELECT sc1.MassaMagra
          from statistichecliente sc1
        where sc1.CodFiscale=_Cliente
          AND year(sc1.data) = year(DATE_SUB(current_date, INTERVAL 1 MONTH))
          AND month(sc1.data) = month(DATE_SUB(current_date, INTERVAL 1 MONTH))
        )
        +

        (
        SELECT sc0.Peso
          from statistichecliente sc0
        where sc0.CodFiscale=_Cliente
          AND year(sc0.data) = year(DATE_SUB(current_date, INTERVAL 1 MONTH))
          AND month(sc0.data) = month(DATE_SUB(current_date, INTERVAL 1 MONTH))
        )
        -

        (
        SELECT sc2.MassaMagra
          from statistichecliente sc2
        where sc2.CodFiscale=_Cliente
          AND year(sc2.data) = year(current_date)
          AND month(sc2.data) = month(current_date)
        )
        +

        (SELECT sc3.Peso
          from statistichecliente sc3
        where sc3.CodFiscale=_Cliente
          AND year(sc3.data) = year(current_date)
          AND month(s3.data) = month(current_date)
        )
        )
        )
    ;
    RETURN(PUNTEGGIO);
END $$
DELIMITER ;
```

## RicreativoFunction

```
DROP FUNCTION IF EXISTS PunteggioRicreativo;
DELIMITER $$
CREATE FUNCTION PunteggioRicreativo(_Cliente VARCHAR (45))
RETURNS INT(11) DETERMINISTIC
BEGIN
    DECLARE PUNTEGGIO INT(11) DEFAULT 0;
    SET PUNTEGGIO =
        100 - (10* (PunteggioDimagrimento(_Cliente) + PunteggioMuscolo(_Cliente) +
        (
            (
                SELECT sc0.Peso
                from statistichecliente sc0
                where sc0.CodFiscale=_Cliente
                    AND year(sc0.data) = year(DATE_SUB(current_date, INTERVAL 1 MONTH))
                    AND month(sc0.data) = month(DATE_SUB(current_date, INTERVAL 1 MONTH))
            )
            -
            (SELECT sc3.Peso
            from statistichecliente sc3
            where sc3.CodFiscale=_Cliente
                AND year(sc3.data) = year(current_date)
                AND month(s3.data) = month(current_date)
            )
        )
    )
    ;
    RETURN(PUNTEGGIO);
END $$
DELIMITER ;
```

## Performance

```
DROP EVENT IF EXISTS ScorePerformance;

DELIMITER $$

CREATE EVENT ScorePerformance
ON SCHEDULE EVERY 1 MONTH
STARTS "2017-11-01"
DO
BEGIN

    INSERT INTO Score
    SELECT year(CURRENT_DATE), MONTH(CURRENT_DATE), cl.CodFiscale,
        PunteggioDimagrimento(cl.CodFiscale)
    FROM cliente cl inner join statistichecliente sc
        on cl.CodFiscale=sc.CodFiscale
    where cl.Scopo = "Dimagrimento";

    INSERT INTO Score
    SELECT year(CURRENT_DATE), MONTH(CURRENT_DATE), cl.CodFiscale,
        PunteggioMuscolo(cl.CodFiscale)
    FROM cliente cl inner join statistichecliente sc
        on cl.CodFiscale=sc.CodFiscale
    where cl.Scopo = "Potenziamento Muscolare";

    INSERT INTO Score
    SELECT year(CURRENT_DATE), MONTH(CURRENT_DATE), cl.CodFiscale,
        PunteggioRicreativo(cl.CodFiscale)
    FROM cliente cl inner join statistichecliente sc
        on cl.CodFiscale=sc.CodFiscale
    where cl.Scopo = "Ricreativo";

END $$
DELIMITER ;
```

## 5.3 Funzionalità 3

Report integratori:

- nVendite mensili per tipologia di integratore
- Integratori in scadenza (order by datascadenza) entro 2 mesi
- Proposta di promozioni sulla base degli integratori in scadenza entro 1 mese o sugli integratori la cui fattura deve essere pagata entro 1 mese

Per scrivere una stored procedure *Promozioni* più snella abbiamo introdotto la ridondanza *Ordine* nella tabella **Giacenza**, essa mantiene memorizzato l'ordine con il quale sono stati acquistati gli integratori. In questo modo si può facilmente andare a controllare la data di pagamento per la fattura.

### NVendite

```
DROP PROCEDURE IF EXISTS nVendite;
DELIMITER $$
CREATE PROCEDURE nVendite()
BEGIN

    SELECT a.IdIntegratore, COUNT(*) AS nVenditeMensili
    FROM acquisto a
    WHERE DATEDIFF(CURRENT_DATE, a.Data) < 32
    GROUP BY a.IdIntegratore;

END $$
DELIMITER ;
```

### Promozioni

```
DROP PROCEDURE IF EXISTS Promozioni;
DELIMITER $$
CREATE PROCEDURE Promozioni(IN _Centro INT(11))
BEGIN

    SELECT g.IdGiacenza, g.IdIntegratore, g.DataScadenza,
           (c.Prezzo / o.Quantita) AS PrezzoOfferta
    FROM giacenza g inner join integratore i on i.IdIntegratore=g.IdIntegratore
           inner join ordine o on g.IdOrdine=o.IdOrdine
           inner join fornitore f on f.NomeFornitore = o.NomeFornitore
           inner join commercializzati c on c.IdIntegratore=i.IdIntegratore
    WHERE g.IdCentro=_Centro AND
    (
        DATEDIFF(g.DataScadenza, CURRENT_DATE) BETWEEN 0 AND 31 OR
        DATEDIFF(DATE_ADD(o.DataConsegna, INTERVAL f.GiorniPagamento DAY), CURRENT_DATE)
        BETWEEN 0 AND 31
    );

END $$
DELIMITER ;
```

## ReportScadenzaIntegratori

```
DROP PROCEDURE IF EXISTS ReportScadenzaIntegratori;
DELIMITER $$
CREATE PROCEDURE ReportScadenzaIntegratori()
BEGIN
    SELECT g.*, DATEDIFF(g.DataScadenza, CURRENT_DATE) AS GiorniMancanti
    FROM giacenza g
    WHERE DATEDIFF(g.DataScadenza, CURRENT_DATE) BETWEEN 0 AND 62;

END $$
DELIMITER ;
```

