



UNIVERSITÀ DI PISA

Corso di Laurea in Ingegneria Informatica

***Progetto per il corso di Basi di Dati
Anno Accademico 2017-2018***

Prof. Gigliola Vaglini, Ing. Francesco Pistolesi

*Realizzazione di un database relazionale in
grado di gestire i dati relativi al sistema
informativo di un'azienda del settore del fitness.*

Studenti:

Turchetti Leonardo

Mono Bill

INDICE

- 1** *Cardinalità*
- 6** *Tavola dei Volumi*
- 8** *Ridondanze e Tavola degli Accessi*
- 14** *Progettazione logica e Normalizzazione*
- 20** *Schema ER*
- 21** *Vincoli di Integrità Referenziale*
- 26** *Query*
- 28** *Area Analytics*

CARDINALITA'

Associazione Dotazione

- Cardinalità (1,N) con CentroFitness, in quanto un centro può disporre di una o più sale
- Cardinalità (1,1) con Sala, in quanto una sala è associata ad una sola sede

Associazione Organizza

- Cardinalità (1,1) con Corso, in quanto ogni Corso è associato ad una Sala
- Cardinalità (1,N) con Sala, in quanto in una Sala possono essere presenti uno o più Corsi

Associazione Esegue

- Cardinalità (1,N) con CentroFitness, in quanto un centro può eseguire uno o più Ordini
- Cardinalità (1,1) con Ordine, in quanto un ordine è legato ad un solo centro

Associazione Composto

- Cardinalità (1,N) con CentroFitness, in quanto un centro è composto da più dipendenti
- Cardinalità (1,N) con Dipendente, in quanto un dipendente può lavorare in uno o più centri

Associazione Ubicazione

- Cardinalità (1,N) con CentroFitness, in quanto un centro può avere uno o più spogliatoi
- Cardinalità (1,1) con Spogliatoio, in quanto uno spogliatoio appartiene ad un solo centro

Associazione Presso

- Cardinalità (1,N) con CentroFitness, in quanto in un centro ci possono essere più accessi
- Cardinalità (1,1) con Accesso, in quanto ogni accesso è legato ad un solo centro

Associazione Dotate

- Cardinalità (0,N) con Sala, in quanto in una sala possono esserci o no delle attrezzature
- Cardinalità (1,1) con AttrezzaturaSala, in quanto una attrezzatura si riferisce ad una sola sala

Associazione Dislocate

- Cardinalità (1,N) con Attrezzatura, in quanto un'attrezzatura può essere presente in più sale
- Cardinalità (1,1) con AttrezzaturaSala, in quanto un'attrezzatura della sala si riferisce ad una sola attrezzatura

Associazione Orario

- Cardinalità (1,N) con Dipendente, in quanto un dipendente può fare uno o più turni di lavoro
- Cardinalità (1,1) con Turno, in quanto ogni turno è legato ad un solo dipendente

Associazione Basata

- Cardinalità (1,N) con Cliente, in quanto un cliente può avere una o più schede
- Cardinalità (1,1) con SchedaAllenamento, in quanto una scheda è legata ad un solo cliente

Associazione Assegnata

- Cardinalità (1,N) con Cliente, in quanto un cliente può avere una o più schede
- Cardinalità (1,1) con SchedaAlimentazione, in quanto una scheda è legata ad un solo cliente

Associazione Effettuato

- Cardinalità (1,1) con Corso, in quanto ad ogni corso è associato un Calendario
- Cardinalità (1,N) con Calendario, in quanto un calendario è composto da uno o più corsi

Associazione Concorde

- Cardinalità (1,N) con Esercizio, in quanto un esercizio può essere in più schede
- Cardinalità (1,1) con EsercizioScheda, in quanto un esercizio della scheda rappresenta un solo esercizio

Associazione Composta

- Cardinalità (1,1) con EsercizioScheda, in quanto ogni esercizio della scheda appartiene a quella specifica scheda
- Cardinalità (1,N) con SchedaAllenamento, in quanto in una scheda ci sono più esercizi

Associazione Associata

- Cardinalità (1,1) con Sfida, in quanto ad ogni sfida è associata una scheda di allenamento
- Cardinalità (0,N) con SchedaAllenamento, in quanto una scheda può essere associata a più sfide

Associazione Lancia

- Cardinalità (0,N) con Cliente, in quanto un cliente può lanciare una o più sfide
- Cardinalità (1,1) con Sfida, in quanto ogni sfida è lanciata da un solo cliente

Associazione Partecipa

- Cardinalità (0,N) con Amico, in quanto un cliente può partecipare a più sfide (o nessuna)
- Cardinalità (0,N) con Sfida, in quanto in una sfida ci possono essere più partecipanti (o nessuno)

Associazione Interesse Cliente

- Cardinalità (1,N) con Cliente, in quanto in un cliente può avere uno o più interessi
- Cardinalità (0,N) con Interesse, in quanto un interesse può essere comune a più clienti (oppure nessuno)

Associazione Amicizia

- Cardinalità (0,N) con Amico, in quanto ogni un cliente può avere più amici (o nessuno)

Associazione Gradimento

- Cardinalità (0,N) con Risposta, in quanto una risposta può ricevere molteplici gradimenti(o nessuno)
- Cardinalità (0,N) con Cliente, in quanto ogni cliente può gradire più risposte(o nessuna)

Associazione Risponde

- Cardinalità (0,N) con Cliente, in quanto un cliente può inviare più risposte(o nessuna)
- Cardinalità (1,1) con Risposta, in quanto una risposta è associata ad un solo cliente

Associazione CerchiaCliente

- Cardinalità (0,N) con Cliente, in quanto un cliente può appartenere a più cerchie(o anche nessuna)
- Cardinalità (1,N) con Cerchia, in quanto ogni cerchia è composta da uno o più clienti

Associazione CerchiaInteresse

- Cardinalità (0,N) con Interesse, in quanto un interesse può essere comune a più cerchie (oppure nessuna)
- Cardinalità (1,N) con Cerchia, in quanto ogni cerchia è composta da uno o più interessi

Associazione Pubblica

- Cardinalità (1,1) con Post, in quanto un post è associato ad un solo cliente
- Cardinalità (0,N) con Profilo, in quanto un profilo può pubblicare più post(o nessuno)

Associazione Associata

- Cardinalità (1,1) con Risposta, in quanto ogni risposta è associata ad un post
- Cardinalità (0,N) con Post, in quanto un post può avere più risposte

Associazione Compone

- Cardinalità (1,N) con Spogliatoio, in quanto in uno spogliatoio ci sono più armadietti
- Cardinalità (1,1) con Armadietto, in quanto un armadietto deve essere in un solo spogliatoio

Associazione Assegnato

- Cardinalità (1,1) con Accesso, in quanto ad un accesso è legato un solo armadietto
- Cardinalità (1,N) con Armadietto, in quanto ad un armadietto sono legati uno o più accessi

Associazione Effettua

- Cardinalità (1,1) con Accesso, in quanto ogni accesso è eseguito da un solo cliente
- Cardinalità (1,N) con Cliente, in quanto un cliente può accedere una o più volte

Associazione Collegata

- Cardinalità (0,1) con Sfida, in quanto una sfida può essere legata oppure no ad una scheda di alimentazione

- Cardinalità (0,N) con SchedaAlimentazione ,in quanto una scheda di alimentazione può essere utilizzata per molteplici sfide

Associazione Attività

- Cardinalità (0,N) con Dipendente, in quanto un dipendente può creare più schede
- Cardinalità (1,1) con SchedaAlimentazione , in quanto una scheda è creata da un solo dipendente

Associazione Attività

- Cardinalità (0,N) con Dipendente, in quanto un dipendente può creare più schede
- Cardinalità (1,1) con SchedaAllenamento, in quanto una scheda è creata da un solo dipendente

Associazione Tramite

- Cardinalità (1,1) con Ordine, in quanto un ordine si riferisce ad un solo fornitore
- Cardinalità (1,N) con Fornitore, in quanto un fornitore può ricevere uno o più ordini

Associazione Vende

- Cardinalità (1,N) con Fornitore, in quanto un fornitore può vendere più integratori
- Cardinalità (1,N) con Integratore, in quanto un integratore può essere venduto da uno o più fornitori

Associazione Legato

- Cardinalità (0,N) con Integratore ,in quanto un integratore può essere usato in più diete(o in nessuna)
- Cardinalità (1,1) con Dieta , in quanto una dieta ha un solo integratore

Associazione Composta

- Cardinalità (1,N) con Dieta, in quanto una dieta può essere usata per più schede
- Cardinalità (1,1) con SchedaAlimentazione , in quanto ad ogni scheda è associata una sola dieta

Associazione Stipula

- Cardinalità (0,1) con Cliente, in quanto un cliente può stipulare o meno un contratto
- Cardinalità (1,1) con Contratto, in quanto un contratto appartiene ad un solo cliente

Associazione Offre

- Cardinalità (1,N) con CentroFitness ,in quanto non può esistere un centro senza contratti e ovviamente ne può avere più di uno
- Cardinalità (1,N) con Contratto ,in quanto un contratto può essere legato o ad uno o più centri

Associazione Propone

- Cardinalità (0,N) con Dipendente, in quanto un dipendente può siglare uno o più contratti(o nessuno)
- Cardinalità (1,1) con Contratto, in quanto un contratto può essere legato o da un solo dipendente

Associazione Acquisto

- Cardinalità (0,N) con Cliente, in quanto un cliente può acquistare più integratori(o nessuno)
- Cardinalità (0,N) con Integratore, in quanto un integratore può essere acquistato da più clienti(o da nessuno)

Associazione Iscrizione

- Cardinalità (1,N) con Cliente, in quanto un cliente può iscriversi a uno o più corsi
- Cardinalità (1,N) con Corso, in quanto in un corso possono partecipare più clienti

Associazione Attività

- Cardinalità (0,N) con Dipendente ,in quanto un dipendente può svolgere più corsi(o nessuno)
- Cardinalità (1,1) con Corso, in quanto un corso è svolto da un solo dipendente

Associazione Responsabile

- Cardinalità (1,N) con Dipendente ,in quanto un dipendente può avere più responsabili

Associazione OrdineIntegratore

- Cardinalità (1,N) con Ordine, in quanto un ordine può avere più integratori
- Cardinalità (1,N) con Integratore, in quanto un integratore può essere presente in più ordini

Associazione Magazzino

- Cardinalità (1,N) con Centro, in quanto un centro può avere più integratori
- Cardinalità (1,N) con Integratore, in quanto un integratore può essere presente in più centri

Associazione Corrisponde

- Cardinalità (1,N) con Attrezzatura, in quanto un' attrezzatura può avere una o più regolazioni
- Cardinalità (1,1) con Regolazione, poiché una regolazione rappresenta una sola attrezzatura

Associazione ControlloAttrezzatura

- Cardinalità (0,N) con EsercizioScheda, in quanto un esercizio può avere più regolazioni oppure nessuno quando non richiede attrezzatura
- Cardinalità (1,N) con Regolazione, in quanto una regolazione può essere associata ad uno o più Esercizi

Associazione Valutazione

- Cardinalità (1,N) con EserciziScheda, in quanto un esercizio della scheda può avere uno o più controlli delle prestazioni
- Cardinalità (1,1) con Log, in quanto un controllo delle prestazioni viene fatto su un solo esercizio

TAVOLA DEI VOLUMI

Nome	E/R	Volume	Motivazione
Centro Fitness	E	20	Ipotesi Iniziale
Sala	E	80	Ogni Centro Fitness ha 4 Sale
Dotazione	R	80	(1,1) con Sala
Attrezzatura Sala	E	480	Ogni Sala ha in media 6 Attrezzature
Attrezzatura	E	50	Ipotesi Iniziale
Dotate	R	300	(1,1) con Attrezzatura Sala
Dislocate	R	300	(1,1) con Attrezzatura Sala
Offre	R	800	Media di 40 Contratti per Centro Fitness
Cliente	E	800	Ipotesi Iniziale
Stipula	R	600	(1,1) con Contratto
Contratto	E	600	Ipotesi Iniziale
Rata	E	200	Ipotesi Iniziale
Pagamento	R	200	(1,1) con Rata
Propone	R	800	(1,1) con Dipendente
Scheda Allenamento	E	200	Ipotesi Iniziale
Basata	R	200	(1,1) con Scheda Allenamento
Composta	R	3000	(1,1) con Esercizi Scheda
Esercizi Scheda	E	3000	Media di 15 Esercizi per Scheda d'Allenamento
Seduta	R	3000	(1,1) con Esercizi Scheda
Valutazione	R	6000	(1,1) con Log
Log	E	6000	2 Controlli delle Prestazioni per Esercizio
Prestazione	R	6000	(1,1) con Log
Esercizio	E	500	Ipotesi Iniziale
Concorde	R	3000	(1,1) con Esercizi Scheda
Controllo Attrezzatura	R	2500	5/6 degli Esercizi Scheda hanno la Regolazione
Corrisponde	R	75	(1,1) con Regolazione
Regolazione	E	75	3/2 delle Attrezzature
Composto	R	40	Media di 40 Dipendenti per Centro Fitness
Dipendente	E	800	Ipotesi Iniziale
Personale	R	80	(1,1) con Sala
Responsabile	R	1600	Media di 2 responsabili per Dipendente
Attività	R	400	(1,1) con Corso
Corso	E	400	Ipotesi Iniziale
Organizza	R	400	(1,1) con Corso
Turno	E	2400	Media di 3 Turni per Dipendente
Orario	R	2400	(1,1) con Turno
Effettuato	R	400	(1,1) con Corso
Calendario	E	300	Ipotesi Iniziale
Presso	R	200	(1,1) con Accesso
Accesso	E	1600	Ipotesi Iniziale
Assegnato	R	1600	(1,1) con Accesso

Armadietto	E	1200	Media di 30 Armadietti per Spogliatoio
Compone	R	1200	(1,1) con Armadietto
Spogliatoio	E	40	Ogni Centro Fitness ha 2 Spogliatoi
Ubicazione	R	40	(1,1) con Spogliatoio
Pubblica	R	600	(1,1) con Post
Post	E	600	Ipotesi Iniziale
Risponde	R	1800	(1,1) con Risposta
Risposta	E	1800	Media di 3 Risposte per Post
Associata	R	1800	(1,1) con Risposta
Gradimento	R	5400	Ogni Risposta ha 3 Gradimenti
Amicizia	R	2500	Media di 5 Richieste di Amicizia per Cliente
Interesse	E	100	Ipotesi Iniziale
Interessi Clienti	R	2400	Media di 3 Interessi per Cliente
Cerchia Interesse	R	200	Media di 2 Interessi per Cerchia
Cerchia	E	100	Ipotesi Iniziale
Cerchia Cliente	R	4000	Media di 5 Clienti per cerchia
Partecipa	R	200	1/4 degli Clienti
Lancia	R	50	(1,1) con Sfida
Sfida	E	50	Ipotesi Iniziale
Associata	R	50	(1,1) con Sfida
Collegata	R	25	1/2 delle Sfide
Scheda Alimentazione	E	200	Ipotesi Iniziale
Attività	R	200	(1,1) con Scheda Alimentazione
Assegnata	R	800	(1,1) con Cliente
Composta	R	200	(1,1) con Scheda Alimentazione
Dieta	E	200	Ipotesi Iniziale
Iscrizione	R	1600	Media di 2 Iscrizioni per Cliente
Attività	R	200	(1,1) con Scheda Allenamento
Acquisto	R	200	1/4 dei Clienti
Integratore	E	50	Ipotesi Iniziale
Legato	R	30	3/5 degli Integratori
Fornitore	E	10	Ipotesi Inziale
Vende	R	100	10 Vendite per Fornitore
Tramite	R	100	(1,1) con Ordine
Ordine	E	100	Ipotesi Iniziale
Ordine Integratore	R	200	Media di 4 Integratori per Ordine
Esegue	R	100	(1,1) con Ordine

RIDONDANZE E TAVOLA DEGLI ACCESSI

Operazioni proposte

1. Trovare l'Integratore più legato alle varie Diete.
 - Frequenza: 2 volte al mese.
2. Inserimento di un nuovo Esercizio.
 - Frequenza: 3 volte a settimana.
3. Trovare il Corso meno frequentato.
 - Frequenza: 4 volte al mese.
4. Inserimento di una nuova Iscrizione ad un Corso.
 - Frequenza: 2 volte al mese.
5. Trovare le Sfide con più partecipanti.
 - Frequenza: 4 volte a settimana.
6. Inserimento di un Contratto multisede con 3 Sedi associate.
 - Frequenza: 10 volte al mese.
7. Trovare il Codice dei Contratti con una sola Sede associata.
 - Frequenza: 5 volte al mese.
8. Calcolare il Gradimento medio di ciascuna Risposta ad un Post.
 - Frequenza: 7 volte a settimana.

Ridondanze proposte

1. L'attributo NumPartecipanti di Corso, che mantiene il numero degli iscritti di un determinato corso, è una ridondanza. Infatti, possiamo individuare il numero degli iscritti tramite il conteggio delle occorrenze nella tabella Iscrizione.
2. L'attributo Sede di Contratto, che mantiene il numero di centri legati ad un solo contratto, è una ridondanza. Infatti, possiamo individuare il numero dei centri tramite il conteggio delle occorrenze nella tabella Offre.

Tavola degli Accessi

Operazione 1 : Trovare l'Integratore più legato alle varie Diete.

La tavola degli accessi di questa operazione non cambia in base alla presenza o meno delle ridondanze proposte

Concetto	Tipo Operazioni	Numero Operazioni	Motivazione
Integratore	Lettura	50	Per recuperare il nome commerciale di ogni integratore
Legato	Lettura	30	Solo 3/5 degli integratori sono collegati alle diete
Dieta	Lettura	200	Per recuperare il codice di ogni dieta

Somma delle operazioni totali: $50+30+200 = 280$

Operazioni mensili: $280*2 = 560$

Operazione 2: Inserimento di un nuovo Esercizio

La tavola degli accessi di questa operazione non cambia in base alla presenza o meno delle ridondanze proposte

Concetto	Tipo Operazioni	Numero Operazioni	Motivazione
Esercizio	Scrittura	1	Su Esercizio per inserire un nuovo esercizio

Somma delle operazioni totali: $1*2 = 2$

Operazioni settimanali: $2*3 = 6$

Operazione 3: Trovare il corso meno frequentato

Con Ridondanza:

Concetto	Tipo Operazioni	Numero Operazioni	Motivazione
Corso	Lettura	400	Per recuperare il codice del corso con meno partecipanti

Somma delle operazioni totali: 400

Operazioni mensili: $400 * 4 = 1600$

Senza Ridondanza:

Concetto	Tipo Operazioni	Numero Operazioni	Motivazione
Iscrizione	Lettura	1600	Per recuperare il più piccolo numero di iscritti
Corso	Lettura	400	Per recuperare il codice del corso associato al numero precedente

Somma delle operazioni totali: $400 + 1600 = 2000$

Operazioni mensili: $2000 * 4 = 8000$

Operazione 4: Inserimento di una nuova Iscrizione ad un Corso

Senza Ridondanza:

Concetto	Tipo Operazioni	Numero Operazioni	Motivazione
Iscrizione	Scrittura	1	Per inserire la nuova Iscrizione

Somma delle operazioni totali: $1 * 2 = 2$

Operazioni mensili: $2 * 600 = 1200$

Con Ridondanza:

Concetto	Tipo Operazioni	Numero Operazioni	Motivazione
Iscrizione	Scrittura	1	Per inserire la nuova Iscrizione
Corso	Scrittura	1	Per aggiornare il NumPartecipanti di quel Corso

Somma delle operazioni totali: $1*2 + 1*2 = 4$

Operazioni mensili: $4*600 = 2400$

Operazione 5: Trovare le Sfide con più partecipanti

La tavola degli accessi di questa operazione non cambia in base alla presenza o meno delle ridondanze proposte

Concetto	Tipo Operazioni	Numero Operazioni	Motivazione
Partecipa	Lettura	200	Per contare i partecipanti e trovare le sfide a loro associati

Somma delle operazioni totali: 200

Operazioni Settimanali: $200*4 = 800$

Operazione 6: Inserimento di un Contratto multisede con 3 Sedi associate.

Senza Ridondanza:

Concetto	Tipo Operazioni	Numero Operazioni	Motivazione
Contratto	Scrittura	1	Per inserire il nuovo Contratto
Offre	Scrittura	3	Per associare al nuovo Contratto le Sedi richieste

Somma delle operazioni totali: $1*2 + 3*2 = 8$

Operazioni Mensili: $8*10 = 80$

Con Ridondanza:

Concetto	Tipo Operazioni	Numero Operazioni	Motivazione
Contratto	Scrittura	1	Per inserire il nuovo Contratto e il numero di Sedi richieste

Somma delle operazioni totali: $1 \times 2 = 2$

Operazioni Mensili: $2 \times 10 = 20$

Operazione 7: Trovare il Codice dei Contratti con una sola Sede associata

Senza Ridondanza:

Concetto	Tipo Operazioni	Numero Operazioni	Motivazione
Offre	Lettura	800	Per recuperare i Codici dei Contratti con una sola Sede associata

Somma delle operazioni totali: 800

Operazioni Mensili: $800 \times 5 = 4000$

Con Ridondanza:

Concetto	Tipo Operazioni	Numero Operazioni	Motivazione
Contratto	Lettura	600	Per recuperare i Codici dei Contratti con una sola Sede associata

Somma delle operazioni totali: 600

Operazioni Mensili: $600 \times 5 = 3000$

Operazione 8: Calcolare il Gradimento medio di ciascuna Risposta ad un Post

La tavola degli accessi di questa operazione non cambia in base alla presenza o meno delle ridondanze proposte

Concetto	Tipo Operazioni	Numero Operazioni	Motivazione
Gradimento	Lettura	5400	Per recuperare il numero di occorrenze e la somma dei punteggi

Somma delle operazioni totali: 5400

Operazioni Settimanali: $5400 * 7 = 37800$

Ridondanza 1:

- Con ridondanza: 3020 Operazioni elementari mensili
- Senza ridondanza: 4080 Operazioni elementari mensili

E' conveniente mantenere la ridondanza per diminuire il numero di operazioni elementari mensili.

Ridondanza 2:

- Con ridondanza: 4000 Operazioni elementari mensili
- Senza ridondanza: 9200 Operazioni elementari mensili

E' conveniente mantenere la ridondanza per diminuire il numero di operazioni elementari mensili.

PROGETTAZIONE LOGICA E NORMALIZZAZIONE

Centro Fitness (CodCentro Indirizzo, Cellulare, Capienza, Dimensione, Direttore, GiornoChiusura, OrarioApertura, OrarioChiusura)

CodCentro->Indirizzo, Cellulare, Capienza, Dimensione, Direttore, GiornoChiusura, OrarioApertura, OrarioChiusura

Sala (CodSala, Nome, CodCentro, CodDipendente, NumApparecchiature)

CodSala->CodSala, Nome, CodCentro, CodDipendente,

NumApparecchiature

Responsabile (Dipendente, Responsabile)

Nessuna dipendenza, (Dipendente, Responsabile) è superchiave minimale della relazione

AttrezzaturaSala (IdAttrezzaturaSala, CodMacchina,

CodSala, Consumo, Usura)

IdAttrezzaturaSala->CodMacchina, CodSala, Consumo,

Usura

Attrezzatura (CodMacchina, Nome)

CodMacchina->Nome

Regolazione (IdRegolazione, CodMacchina, Nome,

Minimo, Massimo)

IdRegolazione->CodMacchina, Nome, Minimo, Massimo

Esercizio (CodEsercizio, Tipo, Dispendio)

CodEsercizio->Tipo, Dispendio

Corso (CodCorso, CodSala, Disciplina, NumPartecipanti,

CodDipendente, DataInizio, DataFine, Livello)

CodCorso->CodSala, Disciplina, NumPartecipanti, CodDipendente, DataInizio,
DataFine, Livello

Calendario (IdCalendario, CodCorso, OraInizio, OraFine,
Giorno)

IdCalendario->CodCorso, OraInizio, OraFine, Giorno

SchedaAllenamento (CodScheda, CodCliente, CodDipendente ,
DataInizio, DataFine)

CodScheda->CodCliente, CodDipendente, DataInizio, DataFine

EserciziScheda (IdEsercizio, CodScheda,
CodEsercizio, Serie, Ripetizioni, Riposo, Durata,
CodAccesso)

IdEsercizio->,CodScheda, CodEsercizio, Serie, Ripetizioni,
Riposo,Durata, CodAccesso

ControlloAttrezzatura(IdEsercizio, IdRegolazione, Intensità)
IdEsercizio, IdRegolazione ->Intensità

Log (IdLog, IdEsercizio, Serie, Ripetizioni, Riposo, TempoImpiegato, CodAccesso)
Idlog->IdEsercizio, Serie, Ripetizioni, Riposo, TempoImpiegato, CodAccesso

Dipendente(CodDipendente, Nome,Cognome, DataNascita, Indirizzo, Documento,
Prefettura)

CodDipendente->Nome, Cognome, DataNascita, Indirizzo, Documento, Prefettura

Composto (CodDipendente, CodCentro, Attività)

CodDipendente, CodCentro->Attività

Turno (IdTurno, CodDipendente, OraInizio,
OraFine, Giorno)

IdTurno->CodDipendente, OraInizio, OraFine,

Giorno

Cliente (CodCliente, Nome, Cognome, DataNascita, Indirizzo, Documento, Prefettura, Username, Password)

CodCliente->Nome, Cognome, DataNascita, Indirizzo, Documento, Prefettura, Username, Password

Contratto (CodContratto, CodCliente, CodDipendente, DataContratto, DataScadenza, Tipo, Tipologia, Multisede, IngressiSettimanali, NumeroSaleAccessibili,

AccessiPiscinaMensili, TipoPiscina, Scopo, Muscoli, Livello, CostoAnnuale, Rateizzazione, TassoInteresse)

CodContratto->CodCliente, CodDipendente, DataContratto, DataScadenza, Tipo, Tipologia, Multisede, IngressiSettimanali, NumeroSaleAccessibili, AccessiPiscinaMensili, TipoPiscina, Scopo, Muscoli, Livello, CostoAnnuale, Rateizzazione, TassoInteresse

Offre(CodContratto, CodCentro)

Nessuna dipendenza, (CodContratto, CodCentro) è superchiave minimale della relazione

Rata (CodRata, CodContratto, Importo, DataScadenza, StatoPagamento)

CodRata-> CodContratto, Importo, DataScadenza, StatoPagamento

Accesso (CodAccesso, CodCliente, CodCentro,

DataAccesso, OraAccesso, OraUscita, Piscina)

CodAccesso-> CodCliente, CodCentro, DataAccesso,

OraAccesso, OraUscita, Piscina

Armadietto (CodArmadietto, CodSpogliatoio, CombSblocco)

CodArmadietto->CodSpogliatoio, CombSblocco

Spogliatoio (CodSpogliatoio, Capienza, PostiDisponibili,
Posizione, CodCentro)

CodSpogliatoio->Capienza, PostiDisponibili, Posizione,
CodCentro

Dieta (CodDieta, PastiGiornalieri, CalorieGiornaliere, Composizione,
NomeIntegratore)

CodDieta->PastiGiornalieri, CalorieGiornaliere, Composizione,
NomeIntegratore

Integratore (NomeCommerciale, Sostanza, DataScadenza,
Pezzi, PrezzoAcquisto, PrezzoVendita) NomeCommerciale-
>Sostanza, DataScadenza, Pezzi, PrezzoAcquisto,
PrezzoVendita

SchedaAlimentazione (IdScheda, CodCliente,
CodDipendente, CodDieta, DataVisita, DataInizio,
DataFine, Altezza, Peso, MassaGrassa, MassaMagra,
AcquaTotale, Obbiettivo, Sesso)
IdScheda->CodCliente, CodDipendente, CodDieta,
DataVisita, DataInizio, DataFine, Altezza, Peso,
MassaGrassa, MassaMagra, AcquaTotale, Obbiettivo, Sesso

Ordine (CodInterno, CodCentro, NomeFornitore,
CodEsterno, DataEvasione, Stato, DataConsegna)
CodInterno->CodCentro, NomeFornitore, CodEsterno,
DataEvasione, Stato, DataConsegna

Fornitore (NomeFornitore, Società, PartitaIva,

Telefono, Indirizzo)

NomeFornitore->Società, PartitaIva, Telefono, Indirizzo

Vende (NomeFornitore, NomeCommerciale)

Nessuna dipendenza, (NomeFornitore, NomeCommerciale) è superchiave minimale della relazione

Magazzino (CodCentro, NomeCommerciale,
Quantità)

CodCentro, NomeCommerciale->Quantità

OrdineIntegratore (CodInterno,
NomeCommerciale, Confezioni)

CodInterno, NomeCommerciale ->Confezioni

Acquisto (CodCliente, NomeCommerciale,

Confezioni, DataAcquisto)

CodCliente, NomeCommerciale->Confezioni, DataAcquisto

Amicizia (CodCliente, Amico, DataRichiesta,

Stato, DataAccettazione)

CodCliente, Amico->DataRichiesta, Stato,

DataAccettazione

Interesse(CodInteresse, Nome)

CodInteresse->Nome

InteresseCliente (CodCliente, CodInteresse)

Nessuna dipendenza, (CodCliente, CodInteresse) è superchiave minimale della relazione

Cerchia (CodCerchia, Nome)

CodCerchia->Nome

CerchiaCliente (CodCliente,

CodCerchia)

Nessuna dipendenza, (CodCliente, CodCerchia) è superchiave minimale della relazione

CerchiaInteresse (CodCerchia, CodInteresse)

Nessuna dipendenza, (CodCerchia, CodInteresse) è superchiave minimale della relazione

Post (IdPost, CodCliente, Testo, DataPubblicazione, Luogo, Link)

IdPost->CodCliente, Testo, DataPubblicazione, Luogo, Link

Gradimento (CodRisposta, CodCliente,

Punteggio)

CodRisposta, CodCliente->Punteggio

Risposta (CodRisposta, IdPost, CodCliente, Testo)

CodRisposta->IdPost, CodCliente, Testo

Sfida(IdSfida, CodCliente, CodScheda, IdScheda, DataLancio, DataInizio, DataFine,

Obbiettivo, Vincitore)

IdSfida->CodCliente, CodScheda, IdScheda, DataLancio, DataInizio, DataFine, Obbiettivo,

Vincitore

Partecipa (IdSfida, CodCliente)


Nessuna dipendenza, (IdSfida, CodCliente) è superchiave minimale della relazione

Iscrizione (CodCliente, CodCorso)

Nessuna dipendenza, (CodCliente, CodCorso) è superchiave minimale della relazione

- Esiste un vincolo di integrità referenziale fra l'attributo Direttore della tabella CentroFitness e l'attributo CodDipendente della tabella Dipendente.□
- Esiste un vincolo di integrità referenziale fra l'attributo Dipendente della tabella Responsabile e l'attributo CodDipendente della tabella Dipendente.□
- Esiste un vincolo di integrità referenziale fra l'attributo Responsabile della tabella Responsabile e l'attributo CodDipendente della tabella Dipendente.□
- Esiste un vincolo di integrità referenziale fra l'attributo CodCentro della tabella Sala e l'attributo CodCentro della tabella CentroFitness.□
- Esiste un vincolo di integrità referenziale fra l'attributo CodDipendente della tabella Sala e l'attributo CodDipendente della tabella Dipendente.□
- Esiste un vincolo di integrità referenziale fra l'attributo CodMacchina della tabella AttrezzaturaSala e l'attributo CodMacchina della tabella Attrezzatura.□
- Esiste un vincolo di integrità referenziale fra l'attributo CodMacchina della tabella Regolazione e l'attributo CodMacchina della tabella Attrezzatura.
- SEP ● Esiste un vincolo di integrità referenziale fra l'attributo CodSala della tabella Corso e l'attributo CodSala della tabella Sala.□
- Esiste un vincolo di integrità referenziale fra l'attributo CodDipendente della tabella Corso e l'attributo CodDipendente della tabella Dipendente.□
- Esiste un vincolo di integrità referenziale fra l'attributo CodCorso della tabella Calendario e l'attributo CodCorso della tabella Corso.□
- Esiste un vincolo di integrità referenziale fra l'attributo CodCliente della tabella SchedaAllenamento e l'attributo CodCliente della tabella Cliente.□
- Esiste un vincolo di integrità referenziale fra l'attributo CodDipendente della tabella SchedaAllenamento e l'attributo CodDipendente della tabella Dipendente.

- Esiste un vincolo di integrità referenziale fra l'attributo CodScheda della tabella EserciziScheda e l'attributo CodScheda della tabella SchedaAllenamento.

 ● Esiste un vincolo di integrità referenziale fra l'attributo CodEsercizio della tabella EserciziScheda e l'attributo CodEsercizio della tabella Esercizio. □

● Esiste un vincolo di integrità referenziale fra l'attributo CodAccesso della tabella EserciziScheda e l'attributo CodAccesso della tabella Accesso. □

● Esiste un vincolo di integrità referenziale fra l'attributo IdEsercizio della tabella ControlloAttrezzatura e l'attributo IdEsercizio della tabella EserciziScheda. □

● Esiste un vincolo di integrità referenziale fra l'attributo IdRegolazione della tabella ControlloAttrezzatura e l'attributo IdRegolazione della tabella Regolazione.

 ● Esiste un vincolo di integrità referenziale fra l'attributo IdEsercizio della tabella Log e l'attributo IdEsercizio della tabella EserciziScheda.

 ● Esiste un vincolo di integrità referenziale fra l'attributo CodAccesso della tabella Log e l'attributo CodAccesso della tabella CodAccesso. □

● Esiste un vincolo di integrità referenziale fra l'attributo CodDipendente della tabella Composto e l'attributo CodDipendente della tabella Dipendente.

 ● Esiste un vincolo di integrità referenziale fra l'attributo CodCentro della tabella Composto e l'attributo CodCentro della tabella CentroFitness.

 ● Esiste un vincolo di integrità referenziale fra l'attributo CodDipendente della tabella Turno e l'attributo CodDipendente della tabella Dipendente. □

● Esiste un vincolo di integrità referenziale fra l'attributo CodDipendente della tabella Contratto e l'attributo CodDipendente della tabella Dipendente. □

● Esiste un vincolo di integrità referenziale fra l'attributo CodCliente della tabella Contratto e l'attributo CodCliente della tabella Cliente. □

● Esiste un vincolo di integrità referenziale fra l'attributo CodCentro della tabella Offre e l'attributo CodCentro della tabella CentroFitness.

● Esiste un vincolo di integrità referenziale fra l'attributo CodContratto della tabella Offre e l'attributo CodContratto della tabella Contratto.

 ● Esiste un vincolo di integrità referenziale fra l'attributo CodContratto

della tabella Rata e l'attributo CodContratto della tabella Contratto.

- Esiste un vincolo di integrità referenziale fra l'attributo CodSpogliatoio della tabella Armadietto e l'attributo CodSpogliatoio della tabella Spogliatoio. □

- Esiste un vincolo di integrità referenziale fra l'attributo CodCentro della tabella Spogliatoio e l'attributo CodCentro della tabella CentroFitness. □

- Esiste un vincolo di integrità referenziale fra l'attributo NomeCommerciale della tabella Dieta e l'attributo NomeCommerciale della tabella Integratore.

-  ● Esiste un vincolo di integrità referenziale fra l'attributo CodCliente della tabella SchedaAlimentazione e l'attributo CodCliente della tabella Cliente. □

- Esiste un vincolo di integrità referenziale fra l'attributo CodDipendente della tabella SchedaAlimentazione e l'attributo CodDipendente della tabella Dipendente.

-  ● Esiste un vincolo di integrità referenziale fra l'attributo CodDieta della tabella SchedaAlimentazione e l'attributo CodDieta della tabella Dieta. □

- Esiste un vincolo di integrità referenziale fra l'attributo CodCentro della tabella Ordine e l'attributo CodCentro della tabella CentroFitness. □

- Esiste un vincolo di integrità referenziale fra l'attributo NomeFornitore della tabella Ordine e l'attributo NomeFornitore della tabella Fornitore. □


- Esiste un vincolo di integrità referenziale fra l'attributo NomeFornitore della tabella Vende e l'attributo NomeFornitore della tabella Fornitore. □

- Esiste un vincolo di integrità referenziale fra l'attributo NomeCommerciale della tabella Vende e l'attributo NomeCommerciale della tabella Integratore.

-  ● Esiste un vincolo di integrità referenziale fra l'attributo NomeCommerciale della tabella Magazzino e l'attributo NomeCommerciale della tabella Integratore. □

- Esiste un vincolo di integrità referenziale fra l'attributo CodCentro della tabella Magazzino e l'attributo CodCentro della tabella CentroFitness. □

- Esiste un vincolo di integrità referenziale fra l'attributo NomeCommerciale della tabella OrdineIntegratore e l'attributo NomeCommerciale della tabella Integratore. □

- Esiste un vincolo di integrità referenziale fra l'attributo CodInterno della tabella OrdineIntegratore e l'attributo CodInterno della tabella Ordine.□
- Esiste un vincolo di integrità referenziale fra l'attributo CodCliente della tabella Acquisto e l'attributo CodCliente della tabella Cliente.□
- Esiste un vincolo di integrità referenziale fra l'attributo NomeCommerciale della tabella Acquisto e l'attributo NomeCommerciale della tabella Integratore.□
- Esiste un vincolo di integrità referenziale fra l'attributo CodCliente della tabella Amico e l'attributo CodCliente della tabella Cliente.□
- Esiste un vincolo di integrità referenziale fra l'attributo Amico della tabella Amico e l'attributo CodCliente della tabella Cliente.□
- Esiste un vincolo di integrità referenziale fra l'attributo CodCliente della tabella InteresseCliente e l'attributo CodCliente della tabella Cliente.□
- Esiste un vincolo di integrità referenziale fra l'attributo CodInteresse della tabella InteresseCliente e l'attributo CodInteresse della tabella Interesse.
-  ● Esiste un vincolo di integrità referenziale fra l'attributo CodCliente della tabella CerchiaCliente e l'attributo CodCliente della tabella Cliente.□
- Esiste un vincolo di integrità referenziale fra l'attributo C o d C e r c h i a della tabella CerchiaCliente e l'attributo CodCerchia della tabella Cerchia.□
- Esiste un vincolo di integrità referenziale fra l'attributo C o d C e r c h i a della tabella CerchiaInteresse e l'attributo CodCerchia della tabella Cerchia.□
- Esiste un vincolo di integrità referenziale fra l'attributo CodInteresse della tabella CerchiaInteresse e l'attributo CodInteresse della tabella Interesse.□
- Esiste un vincolo di integrità referenziale fra l'attributo CodCliente della tabella Post e l'attributo CodCliente della tabella Cliente.
- Esiste un vincolo di integrità referenziale fra l'attributo CodCliente della tabella Risposta e l'attributo CodCliente della tabella Cliente.□
- Esiste un vincolo di integrità referenziale fra l'attributo IdPost della tabella Risposta e l'attributo IdPost della tabella Post.

- Esiste un vincolo di integrità referenziale fra l'attributo CodCliente della tabella Gradimento e l'attributo CodCliente della tabella Cliente.

-  ● Esiste un vincolo di integrità referenziale fra l'attributo CodRisposta della tabella Gradimento e l'attributo CodRisposta della tabella Risposta.


- Esiste un vincolo di integrità referenziale fra l'attributo CodCliente della tabella Sfida e l'attributo CodCliente della tabella Cliente. □

- Esiste un vincolo di integrità referenziale fra l'attributo CodScheda della tabella Sfida e l'attributo CodScheda della tabella SchedaAllenamento.

- Esiste un vincolo di integrità referenziale fra l'attributo IdScheda della tabella Sfida e l'attributo IdScheda della tabella SchedaAlimentazione. □

- Esiste un vincolo di integrità referenziale fra l'attributo Vincitore della tabella Sfida e l'attributo CodCliente della tabella Cliente.

- Esiste un vincolo di integrità referenziale fra l'attributo CodCliente della tabella Partecipa e l'attributo CodCliente della tabella Cliente.

-  ● Esiste un vincolo di integrità referenziale fra l'attributo IdSfida della tabella Partecipa e l'attributo IdSfida della tabella Sfida.

- Esiste un vincolo di integrità referenziale fra l'attributo CodCliente della tabella Iscrizione e l'attributo CodCliente della tabella Cliente. □

- Esiste un vincolo di integrità referenziale fra l'attributo CodCorso della tabella Iscrizione e l'attributo CodCorso della tabella Corso.

QUERY

Trovare l'Integratore più legato alle varie Diete.

```
SELECT I.NomeCommerciale
FROM Integratore I
INNER JOIN Dieta D
ON I.NomeCommerciale = D.NomeIntegratore
WHERE COUNT(*) >= ALL
      (SELECT COUNT(*)
       FROM Integratore I2
        INNER JOIN Dieta D2
        ON I2.NomeCommerciale = D2.NomeIntegratore
        GROUP BY I2.NomeCommerciale)
GROUP BY I.NomeCommerciale;
```

Inserimento di un nuovo Esercizio.

```
INSERT INTO Esercizio (Tipo, Dispendio)
VALUES ('MilitaryPress', '220');
```

Trovare il Corso meno frequentato.

```
SELECT C.CodCorso
FROM Corso C
WHERE C.NumPartecipanti < ALL
      (SELECT C2.NumPartecipanti
       FROM Corso C2
       WHERE C2.CodCorso <> C.CodCorso);
```

Inserimento di una nuova Iscrizione ad un Corso.

```
INSERT INTO Iscrizione (CodCliente, CodCorso)
VALUES ('10', '15');
```

Trovare le Sfide con più partecipanti.

```
SELECT P.IdSfida
FROM Partecipa P
GROUP BY P.IdSfida
HAVING COUNT(*) =
      (SELECT MAX(D.Occorrenze)
       FROM
```

```
(SELECT P2.IdSfida, COUNT(*) AS Occorrenze
FROM Partecipa P2
GROUP BY P2.IdSfida) AS D);
```

Inserimento di un Contratto multisede con 3 Sedi associate.

```
INSERT INTO Contratto (CodCliente, CodDipendente, DataContratto,
DataScadenza, Tipo, Tipologia, Sede, IngressiSettimanali,
NumeroSaleAccessibili, AccessiPiscinaMensili, TipoPiscina, Scopo, Muscoli,
Livello, CostoAnnuale, Rateizzazione, TassoInteresse)
```

```
VALUES ('13','15','2018-06-06','2019-06-06', 'standard', 'platinum', '3',
'7','4','20','interna','potenziamento muscolare','braccia','elevato','300','no','0');
```

Trovare il Codice dei Contratti con una sola Sede associata.

```
SELECT C.CodContratto
FROM Contratto C
WHERE C.Sede = 1;
```

Calcolare il Gradimento medio di ciascuna Risposta ad un Post.

```
CREATE OR REPLACE VIEW SommaPunteggio AS
SELECT G.CodRisposta, SUM(G.Punteggio) as SommaPunteggio
FROM Gradimento G
GROUP BY G.CodRisposta;
```

```
CREATE OR REPLACE VIEW NumeroOccorrenze AS
SELECT G.CodRisposta, COUNT(*) as NumeroOccorrenze
FROM Gradimento G
GROUP BY G.CodRisposta;
```

```
SELECT SP.CodRisposta, (SP.SommaPunteggio/NU.NumeroOccorrenze) AS
GradimentoMedio
FROM SommaPunteggio SP
NATURAL JOIN NumeroOccorrenze NU
GROUP BY SP.CodRisposta;
```

AREA ANALYTICS

Reporting

Questa prima funzionalità è volta all'individuazione delle criticità dei centri fitness in termini di corsi con pochi frequentanti, fasce orarie scarsamente frequentate e servizi accessori utilizzati dai clienti, cioè servizi non previsti dal contratto, di cui l'utente usufruisce, pagandoli a parte. L'analisi della fruizione di servizi accessori come ingressi alle piscine per contratti in cui non sono previsti, uso frequente alle aree allestibili ecc. può dare l'opportunità alla direzione di confezionare promozioni specificatamente studiate per taluni clienti. Le promozioni sono ovviamente orientate a far acquistare al cliente alcuni pacchetti integrativi per il suo contratto. La funzionalità analytics di reporting deve anche rendere possibile il monitoraggio dell'utilizzo delle attrezzature per capire se i vari centri sono adeguatamente dimensionati, analizzando anche i tempi di attesa dei clienti (espressi usando gli istanti di inizio e fine degli esercizi registrati) sulla base delle schede di allenamento.

Sono state utilizzate 4 stored procedure:

- ❖ La prima serve per visualizzare i corsi e, di conseguenza, le fasce orarie che vengono scarsamente frequentate dai clienti.
- ❖ La seconda, invece, ha il compito di analizzare i servizi accessori di cui alcuni clienti usufruiscono, in particolare l'utilizzo della piscina da parte dei contratti standard e grado silver, e di stilare un pacchetto integrativo che possa venire incontro alle loro esigenze. Vengono contati gli accessi effettuati in piscina e viene stilato un numero accessi massimo consentiti con un prezzo che varia a seconda degli accessi effettuati e che, ovviamente, ha una convenienza maggiore tanto più sono gli accessi effettuati.
- ❖ La terza viene usata per visualizzare i tempi di attesa medi per ciascuna macchina usata in ciascuna sala di ogni centro fitness.
- ❖ La quarta riprende i tempi di attesa medi delle attrezzature e analizza il dimensionamento per ogni sala di ogni centro fitness dandone un giudizio che suggerisce all'azienda i macchinari che devono essere acquistati per rendere un servizio migliore ai clienti che frequentano le varie sale dei vari centri fitness.


```

DROP PROCEDURE IF EXISTS MonitoraggioCorsiPocoFrequentati;
DELIMITER $$
CREATE PROCEDURE MonitoraggioCorsiPocoFrequentati()
BEGIN

```

```

CREATE TEMPORARY TABLE IF NOT
EXISTS _MonitoraggioCorsiPocoFrequentati(
CodCorso INT NOT NULL,
NumPartecipanti INT NOT NULL,
OraInizio INT NOT NULL,
OraFine INT NOT NULL,
Giorno VARCHAR(100) NOT NULL,
PRIMARY KEY(CodCorso)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

TRUNCATE TABLE _MonitoraggioCorsiPocoFrequentati;

```

```

INSERT INTO _MonitoraggioCorsiPocoFrequentati
SELECT C.CodCorso, C.NumPartecipanti, CA.OraInizio, CA.OraFine,
CA.Giorno
FROM Corso C
NATURAL JOIN Calendario CA
ORDER BY C.NumPartecipanti ASC;

```

```

END $$
DELIMITER ;

```

```

DROP PROCEDURE IF EXISTS PacchettiIntegrativi;
DELIMITER $$
CREATE PROCEDURE PacchettiIntegrativi(IN CodContratto INT,IN
CodCliente INT, OUT CodPromozione INT,
OUT CodClientePromozione INT, OUT NumAccessiPiscinaPromozione INT,
OUT PrezzoPromozione INT )
BEGIN

```

```

DECLARE NumAccessiPiscina INT DEFAULT 0;
DECLARE TipoContratto VARCHAR(100) DEFAULT "";
DECLARE TipologiaContratto VARCHAR(100) DEFAULT "";

```

```
SELECT COUNT(*) INTO NumAccessiPiscina
FROM Accesso A
WHERE A.CodCliente = CodCliente
AND A.Piscina = 'si';
```

```
SELECT C.Tipo INTO TipoContratto
FROM Contratto C
WHERE C.CodContratto = CodContratto
AND C.CodCliente = CodCliente;
```

```
SELECT C.Tipologia INTO TipologiaContratto
FROM Contratto C
WHERE C.CodContratto = CodContratto
AND C.CodCliente = CodCliente;
```

```
/*Vengono settati i parametri del pacchetto integrativo*/
```

```
IF (TipoContratto = 'standard' AND TipologiaContratto = 'silver' AND
NumAccessiPiscina != 0) THEN
SET CodPromozione = CodContratto;
SET CodClientePromozione = CodCliente;
SET NumAccessiPiscinaPromozione = NumAccessiPiscina + 10;
```

```
CASE
WHEN NumAccessiPiscina BETWEEN 1 AND 5 THEN
SET PrezzoPromozione = 80;
WHEN NumAccessiPiscina BETWEEN 6 AND 10 THEN
SET PrezzoPromozione = 170;
WHEN NumAccessiPiscina BETWEEN 11 AND 20 THEN
SET PrezzoPromozione = 260;
END CASE;
END IF;
```

```
END $$
DELIMITER ;
```

```
CREATE OR REPLACE VIEW CentroSala AS
SELECT S.CodCentro,S.CodSala
FROM Sala S;
```

```
CREATE OR REPLACE VIEW CentroSalaAttrezzatura AS
SELECT CS.CodCentro,CS.CodSala,ATS.CodMacchina
FROM CentroSala CS
NATURAL JOIN AttrezzaturaSala ATS;
```

```
CREATE OR REPLACE VIEW CentroSalaAttrezzatura2 AS
SELECT CSA.CodCentro,CSA.CodSala,CSA.CodMacchina
FROM CentroSalaAttrezzatura CSA
NATURAL JOIN Attrezzatura A;
```

```
CREATE OR REPLACE VIEW CentroSalaAttrezzaturaRegolazione AS
SELECT
CSA2.CodCentro,CSA2.CodSala,CSA2.CodMacchina,R.IdRegolazione
FROM CentroSalaAttrezzatura2 CSA2
NATURAL JOIN Regolazione R;
```

```
CREATE OR REPLACE VIEW CentroSalaAttrezzaturaRegolazione2 AS
SELECT
CSAR.CodCentro,CSAR.CodSala,CSAR.CodMacchina,CSAR.IdRegolazione,C
A.IdEsercizio
FROM CentroSalaAttrezzaturaRegolazione CSAR
NATURAL JOIN ControlloAttrezzatura CA;
```

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

```
CREATE TEMPORARY TABLE IF NOT EXISTS _MonitoraggioAttrezzaure(
CodCentro INT NOT NULL,
CodSala INT NOT NULL,
CodMacchina INT NOT NULL,
TempoAttesaMedio INT NOT NULL,
PRIMARY KEY(CodCentro,CodSala,CodMacchina)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
TRUNCATE TABLE _MonitoraggioAttrezzaure;
```

```

INSERT INTO _MonitoraggioAttrezzaure
SELECT
CSAR2.CodCentro,CSAR2.CodSala,CSAR2.CodMacchina,AVG(L.TempoImpi
egato) AS TempoAttesaMedio
FROM CentroSalaAttrezzaturaRegolazione2 CSAR2
NATURAL JOIN Log L
GROUP BY CSAR2.CodCentro,CSAR2.CodSala,CSAR2.CodMacchina;

```

```

END $$
DELIMITER ;

```

```

CREATE OR REPLACE VIEW TempoAttesaMedio AS
SELECT
CSAR2.CodCentro,CSAR2.CodSala,CSAR2.CodMacchina,AVG(L.TempoImpi
egato) AS TempoAttesaMedio
FROM CentroSalaAttrezzaturaRegolazione2 CSAR2
NATURAL JOIN Log L
GROUP BY CSAR2.CodCentro,CSAR2.CodSala,CSAR2.CodMacchina;

```

```

DROP PROCEDURE IF EXISTS Dimensionamento;
DELIMITER $$
CREATE PROCEDURE Dimensionamento(IN CodCentro INT,IN CodSala
INT,
IN CodMacchina INT, OUT Dimensionamento VARCHAR(100))
BEGIN

```

```

DECLARE TempoAttesaMedio INT DEFAULT 0;
DECLARE Occorrenze INT DEFAULT 0;

```

```

SELECT COUNT(*) INTO Occorrenze
FROM TempoAttesaMedio TAM
WHERE TAM.CodCentro = Codcentro
AND TAM.CodSala = CodSala
AND TAM.CodMacchina = CodMacchina;

```

```

/*Valutazione del caso in cui non esista il terzetto centro, sala, macchina
inserito*/
IF Occorrenze = 0 THEN
SET Dimensionamento = 'Dimensionamento non Pervenuto';
END IF;

```

```

SELECT TAM.TempoAttesamedio INTO TempoAttesaMedio

```

```

FROM TempoAttesaMedio TAM
WHERE TAM.CodCentro = Codcentro
AND TAM.CodSala = CodSala
AND TAM.CodMacchina = CodMacchina;

/*Viene stimato il dimensionamento del terzetto inserito*/
IF Occorrenze != 0 THEN
CASE
WHEN TempoAttesaMedio < 15 THEN
SET Dimensionamento = 'Buon Dimensionamento';
WHEN TempoAttesaMedio BETWEEN 15 AND 20 THEN
SET Dimensionamento = 'Dimensionamento Sufficiente';
WHEN TempoAttesaMedio > 20 THEN
SET Dimensionamento = 'Pessimo Dimensionamento';
END CASE;
END IF;

END $$
DELIMITER ;

```

Performance Sportiva

Sulla base del problema relativo alla misurazione della performance sportiva del cliente, è stata implementata una soluzione al fine di monitorare le prestazioni dello stesso, sia per ogni esecuzione di un esercizio previsto dalla sua scheda , sia per ogni sessione di allenamento sostenuta. A tale scopo sono state create due procedure, nelle quali sono state utilizzate rispettivamente le tabelle EsercizioScheda e Log, la prima contenente gli esercizi della scheda di allenamento di ogni cliente per ogni suo accesso ad uno dei centri, la seconda invece contenente ciò che il cliente ha di fatto effettivamente eseguito della sua scheda relativo a quell'accesso.

La prima procedura analizza la prestazione del cliente nell'esecuzione di ogni singolo esercizio contenuto nella sua scheda

```

drop procedure if exists efficienzaesercizio;
delimiter $$
create procedure efficienzaesercizio(in _esercizio int,out _serie char(50),out
_ripetizioni char(50),out _riposo char(50),out _tempoimpiegato
varchar(255))

begin

declare seriecliente int default 0;
declare serienellascheda int default 0;
declare ripetizioniciente int default 0;
declare ripetizionischeda int default 0;

```

```
declare riposocliente int default 0;  
declare ripososcheda int default 0;  
declare tempoiimpiegatocliente int default 0;  
declare tempoiimpiegatoscheda int default 0;
```

```
set seriecliente = (select Serie  
from log G  
where G.IdEsercizio = __esercizio);
```

```
set serienellascheda = (select Serie  
from esercizischeda S  
where S.IdEsercizio = __esercizio);
```

```
set ripetizioniciente = (select Ripetizioni  
from log G  
where G.IdEsercizio = __esercizio);
```

```
set ripetizionischeda = (select Ripetizioni  
from esercizischeda S  
where S.IdEsercizio = __esercizio);
```

```
set riposocliente = (select Riposo  
from log G  
where G.IdEsercizio = __esercizio);
```

```
set ripososcheda = (select Riposo  
from esercizischeda S  
where S.IdEsercizio = __esercizio);
```

```
set tempoiimpiegatocliente = (select TempoImpiegato  
from log G  
where G.IdEsercizio = __esercizio);
```

```
set tempoiimpiegatoscheda = (select Durata  
from esercizischeda S  
where S.IdEsercizio = __esercizio);
```

```
/* Confronto fra il tempo impiegato dal cliente nell'esecuzione dell'esercizio  
preso in esame, e il tempo contenuto nella sua scheda e relativa uscita della  
procedura*/
```

```
if (tempoimpiegatocliente-tempoimpiegatoscheda>0) then
set _tempoimpiegato = 'hai impiegato un tempo maggiore rispetto a quanto
veniva richiesto ';
```

```
elseif (tempoimpiegatocliente-tempoimpiegatoscheda<0) then
set _tempoimpiegato = 'Complimenti hai impiegato un tempo minore rispetto
a quanto veniva richiesto ';
```

```
elseif (tempoimpiegatocliente-tempoimpiegatoscheda=0) then
set _tempoimpiegato = 'lo stesso che richiedeva l esercizio';
```

```
end if;
```

```
/* Confronto fra il numero di ripetizioni eseguite dal cliente nell'esecuzione
dell'esercizio preso in esame, e il numero di ripetizioni contenuto nella sua
scheda e relativa uscita della procedura*/
```

```
if (ripetizionicliente-ripetizionischeda>0) then
set _ripetizioni = 'hai eseguito un numero maggiore di ripetizioni';
```

```
elseif (ripetizionicliente-ripetizionischeda<0) then
set _ripetizioni = 'hai eseguito un numero minore di ripetizioni';
```

```
elseif (ripetizionicliente-ripetizionischeda=0) then
set _ripetizioni = 'stesso numero di ripetizioni richiesto ';
```

```
end if;
```

```
/* Confronto fra il numero di serie eseguite dal cliente nell'esecuzione
dell'esercizio preso in esame, e il il numero di serie contenuto nella sua
scheda e relativa uscita della procedura*/
```

```
if (seriecliente-serienellascheda>0) then
set _serie = 'hai eseguito un numero maggiore di serie';
elseif (seriecliente-serienellascheda<0) then
set _serie = 'hai eseguito un numero minore di serie';
```

```
elseif (seriecliente-serienellascheda=0) then
set _serie = 'stesso numero di serie richiesto ';
end if;
```

```
/* Confronto fra il riposo sostenuto dal cliente durante l'esecuzione
dell'esercizio preso in esame, e il riposo contenuto nella sua scheda e relativa
uscita della procedura*/
```

```
if (riposocliente-ripososcheda>0) then
set _riposo = ' ti stai riposando troppo ';
elseif (riposocliente-ripososcheda<0) then
set _riposo = 'puoi recuperare ancora ';
```

```
elseif (riposocliente-ripososcheda=0) then
set _riposo = 'riposo giusto ';
```

```
end if;
```

```
/* Confronto fra il tempo impiegato dal cliente nell'esecuzione dell'esercizio, e
il tempo contenuto nella sua scheda negli esercizi che non richiedono un
numero di serie e ripetizioni ad esempio la corsa o la cyclette, e relativa uscita
della procedura*/
```

```
if (riposocliente = 0 and seriecliente = 0 and ripetizioniciente = 0 and
tempoimpiegatocliente-tempoimpiegatoscheda>=0) then
set _tempoimpiegato = 'COMPLIMENTI HAI SVOLTO con successo 1
esercizio di aerobica '
```

```
;
set _riposo = 'nessun riposo ';
set _serie = 'nessuna serie';
set _ripetizioni = 'nessuna ripetizione ';
end if;
```

```
if (riposocliente = 0 and seriecliente = 0 and ripetizioniciente = 0 and
tempoimpiegatocliente-tempoimpiegatoscheda<0) then
set _tempoimpiegato = 'NON HAI SVOLTO con successo 1 esercizio di
aerobica'
```

```
;
set _riposo = 'nessun riposo ';
set _serie = 'nessuna serie';
set _ripetizioni = 'nessuna ripetizione ';
```

```
end if;
end $$
delimiter ;
```


La seconda procedura analizza le prestazioni del cliente nell'intera sessione di allenamento modificando eventualmente i parametri della scheda relativa a quel accesso nel caso in cui il cliente durante la performance non si fosse attenuto a ciò che la scheda richiedeva

```
drop procedure if exists Valutazione;  
delimiter $$  
create procedure Valutazione(in _scheda int,in _accesso int,out _serie  
char(50),out _differenzaserie int,out _ripetizioni char(50),  
out _differenzaripetizioni int,out _riposo char(50),out _differenzariposo  
int,out _tempoimpiegato varchar(255),out _differenzatempoimpiegato int)
```

```
begin
```

```
declare serietotalicliente int default 0;  
declare serietotalinellascheda int default 0;  
declare ripetizionitotalicliente int default 0;  
declare ripetizionitotalischeda int default 0;  
declare riposototalecliente int default 0;  
declare riposototalescheda int default 0;  
declare tempoimpiegatototalecliente int default 0;  
declare tempoimpiegatototalescheda int default 0;  
declare numeroaccesso int default 0;
```

```
set numeroaccesso = (select count(CodScheda)  
from esercizischeda  
where CodAccesso = _accesso  
and CodScheda = _scheda );
```

```
set serietotalicliente = (select sum(G.Serie)  
from log G  
inner join EserciziScheda S on S.IdEsercizio = G.IdEsercizio  
where G.CodAccesso = _accesso  
and S.CodScheda = _scheda);
```

```
set serietotalinellascheda = (select sum(S.Serie)  
from esercizischeda S  
where S.CodAccesso = _accesso);
```

```
set ripetizionitotalicliente = (select sum(G.Ripetizioni)  
from log G  
inner join EserciziScheda S on S.IdEsercizio = G.IdEsercizio  
where G.CodAccesso = _accesso
```

```
and S.CodScheda = __scheda);
```

```
set ripetizionitotalischeda = (select sum(S.Ripetizioni)
from esercizischeda S
where S.CodAccesso = __accesso);
```

```
set riposototalecliente = (select sum(G.Riposo)
from log G
inner join EserciziScheda S on S.IdEsercizio = G.IdEsercizio
where G.CodAccesso = __accesso
and S.CodScheda = __scheda);
```

```
set riposototalescheda = (select sum(S.Riposo)
from esercizischeda S
where S.CodAccesso = __accesso);
```

```
set tempoiimpiegatototalecliente = (select sum(G.TempoImpiegato)
from log G
inner join EserciziScheda S on S.IdEsercizio = G.IdEsercizio
where G.CodAccesso = __accesso
and S.CodScheda = __scheda);
```

```
set tempoiimpiegatototalescheda = (select sum(S.Durata)
from esercizischeda S
where S.CodAccesso = __accesso);
```

```
/* Confronto fra la somma totale delle serie,delle ripetizioni,e dei tempi di
riposo impiegati dal cliente nell'esecuzione di tutti gli esercizi della giornata, e
la somma totale delle serie,delle ripetizioni,e dei tempi di riposo contenuti
nella sua scheda e relativa uscita della procedura con il la visualizzazione
della differenza tra i vari parametri*/
```

```
if (serietotalicliente-serietotalinellascheda>0) then
set __serie = 'hai eseguito un numero maggiore di serie';
set __differenzaserie = serietotalicliente-serietotalinellascheda;
elseif (serietotalicliente-serietotalinellascheda<0) then
set __serie = 'hai eseguito un numero minore di serie';
set __differenzaserie = serietotalicliente-serietotalinellascheda;
elseif (serietotalicliente-serietotalinellascheda=0) then
set __serie = 'stesso numero di serie richiesto';
set __differenzaserie = serietotalicliente-serietotalinellascheda;
end if;
if (ripetizionitotalicliente-ripetizionitotalischeda>0) then
```

```
set __ripetizioni = 'hai eseguito un numero maggiore di ripetizioni';  
set __differenzaripetizioni = ripetizionitotalcliente-ripetizionitotalscheda;
```

```
elseif (ripetizionitotalcliente-ripetizionitotalscheda<0) then  
set __ripetizioni = 'hai eseguito un numero minore di ripetizioni';  
set __differenzaripetizioni = ripetizionitotalcliente-ripetizionitotalscheda;
```

```
elseif (ripetizionitotalcliente-ripetizionitotalscheda=0) then  
set __ripetizioni = 'stesso numero di ripetizioni richiesto';  
set __differenzaripetizioni = ripetizionitotalcliente-ripetizionitotalscheda;  
end if;
```

```
if (riposototalecliente-riposototalescheda>0) then  
set __riposo = 'ti stai riposando troppo';  
set __differenzariposo = riposototalcliente-riposototalscheda;
```

```
elseif (riposototalecliente-riposototalescheda<0) then  
set __riposo = 'puoi recuperare ancora';  
set __differenzariposo = riposototalcliente-riposototalscheda;
```

```
elseif (riposototalecliente-riposototalescheda=0) then  
set __riposo = 'riposo giusto';  
set __differenzariposo = riposototalcliente-riposototalscheda;  
end if;
```

/* Confronto fra il tempo totale impiegato dal cliente nell'esecuzione di tutti gli esercizi della giornata, e il tempo totale contenuto nella sua scheda e relativa uscita della procedura con il la visualizzazione della differenza tra i due tempi */

```
if (tempoimpiegatototalecliente-tempoimpiegatototalescheda>0) then  
set __tempoimpiegato = 'hai impiegato un tempo maggiore rispetto a quanto veniva richiesto';
```

```
set __differenzatempoimpiegato=tempoimpiegatototalecliente-  
tempoimpiegatototalescheda;
```

```
elseif (tempoimpiegatototalecliente-tempoimpiegatototalescheda<0) then  
set __tempoimpiegato = 'Complimenti hai impiegato un tempo minore rispetto a quanto veniva richiesto';
```

```
set __differenzatempoimpiegato=tempoimpiegatototalecliente-  
tempoimpiegatototalescheda;
```

```
elseif (tempoimpiegatototalecliente-tempoimpiegatototalescheda=0) then
```

```

set __tempoimpiegato = 'lo stesso che richiedeva l esercizio';
set __differenzatempoimpiegato=tempoimpiegatototalecliente-
tempoimpiegatototalescheda;
end if;
/* Aggiornamento della scheda di allenamento, il carico per quell'accesso
viene ridistribuito in base ai valori trovati sopra se un cliente ad esempio ha
fatto più serie di quante ne avrebbe dovute fare da programma, il sistema
intelligente modificherà la scheda ridistribuendo tra i vari suoi esercizi di
quella giornata la differenza tra le serie fatte e quelle contenute nella sua
scheda */

update esercizischeda
set serie = serie+__differenzaserie/numeroaccesso
where CodAccesso = __accesso
and CodScheda = __scheda;

update esercizischeda
set ripetizioni = ripetizioni+ __differenzaripetizioni/numeroaccesso
where CodAccesso = __accesso;

update esercizischeda
set riposo = riposo+ __differenzariposo/numeroaccesso
where CodAccesso = __accesso
and CodScheda = __scheda;

update esercizischeda
set Durata = Durata+__differenzatempoimpiegato/numeroaccesso
where CodAccesso = __accesso
and CodScheda = __scheda;
end $$
delimiter ;

```

Rotazione del magazzino

Questa funzionalità ha lo scopo di controllare le vendite degli integratori valutando periodicamente lo stato del magazzino. In particolare, deve essere possibile avere dei report aggiornati mensilmente circa gli integratori a magazzino, analizzati per data di scadenza e tipologia. Realizzare anche un report delle vendite strutturato ad esempio per prodotto. In base agli integratori più prossimi alla scadenza oppure in base a quelli venduti meno frequentemente, devono essere automaticamente generate delle promozioni al fine di evitare che talune confezioni scadano o rimangano invendute per un periodo di tempo superiore a quello in cui la merce deve essere pagata ai fornitori. Le promozioni sono sotto forma di report e saranno valutate dalla direzione al fine di scegliere quale mettere in essere. Per esempio, se un lotto di integratori è pagato mediante una fattura a 90 giorni dalla data di arrivo della merce a magazzino, quegli stessi integratori dovranno essere stati venduti prima dei 90 giorni. Nella peggiore delle ipotesi, è sufficiente vendere una quantità di confezioni pari al costo complessivo di acquisto. Le funzionalità qui progettate devono essere in grado di assistere la direzione per determinare le tipologie di integratori di cui cessare gli ordinativi, nonché effettuare ranking dei prodotti da vendere prioritariamente, con quale margine temporale e con quale sconto. I margini temporali e le politiche di prezzo sono lasciate a discrezione dello studente.

Sono stati utilizzati 2 event, 3 stored procedure e una funzione:

- ❖ I 2 event hanno lo scopo di monitorare mensilmente gli integratori che sono in scadenza(vengono considerati quelli con meno di 45 giorni dalla data di scadenza), e quelli che non vengono venduti in quantità sufficiente(vengono considerati quelli dei quali si vende non oltre il 10% del totale delle confezioni)
- ❖ La prima procedura va ad analizzare gli integratori per sviluppare delle promozioni su questi ultimi. Viene calcolato uno sconto che si compone di una “sconto vendite” e uno “sconto scadenza”, che si applicano considerando gli integratori con i parametri descritti sopra. Lo sconto scadenza aumenta più ci si avvicina alla scadenza. Infine si ricalcola un nuovo prezzo di vendita, affinché si possano smaltire gli integratori facenti parte delle categorie considerate.
- ❖ La seconda procedura, invece, fornisce gli integratori che fanno parte delle categorie considerate sopra, affinché l'azienda si renda conto di quali integratori cessare gli acquisti
- ❖ La terza procedura utilizza una funzione che calcola lo sconto nello stesso modo di prima, perché viene stilata una classifica degli integratori per margine temporale entro il quale devono essere venduti

```

DROP EVENT IF EXISTS ResocontoScadenze;
DELIMITER $$
CREATE EVENT ResocontoScadenze
ON SCHEDULE EVERY 1 MONTH
DO BEGIN

SELECT M.CodCentro,M.NomeCommerciale,
DATEDIFF(I.DataScadenza,current_date()) as GiorniScadenza
FROM Magazzino M INNER JOIN Integratore I
ON M.NomeCommerciale = I.NomeCommerciale;

END $$
DELIMITER ;

```

```

DROP EVENT IF EXISTS ResocontoAcquisti;
DELIMITER $$
CREATE EVENT ResocontoAcquisti
ON SCHEDULE EVERY 1 MONTH
DO BEGIN

SELECT A.NomeCommerciale, COUNT(*) as ConfezioniVendute
FROM Acquisto A
GROUP BY A.NomeCommerciale
ORDER BY ConfezioniVendute ASC;

END $$
DELIMITER ;

```

```

DROP PROCEDURE IF EXISTS NuovoPrezzoVendita;
DELIMITER $$
CREATE PROCEDURE NuovoPrezzoVendita(IN NomeCommerciale
CHAR(100),IN PrezzoAcquisto INT, IN PrezzoVendita INT,
OUT ScontoTotale INT, OUT NuovoPrezzoVendita INT)
BEGIN

```

```

DECLARE ScontoVendite INT DEFAULT 0;
DECLARE ScontoScadenza INT DEFAULT 0;
DECLARE ConfezioniVendute INT DEFAULT 0;
DECLARE ConfezioniTotali INT DEFAULT 0;
DECLARE GiorniScadenza INT DEFAULT 0;

```

```

SELECT SUM(M.Quantità) INTO ConfezioniTotali
FROM Magazzino M
WHERE M.NomeCommerciale = NomeCommerciale;

```

```

SELECT SUM(A.Confezioni) INTO ConfezioniVendute
FROM Acquisto A
WHERE A.NomeCommerciale = NomeCommerciale;

/*Calcolo dello sconto vendite*/
IF ConfezioniVendute <= (10*ConfezioniTotali)/100 THEN
SET ScontoVendite = 20;
END IF;

SELECT DATEDIFF(I.DataScadenza,current_date()) INTO GiorniScadenza
FROM Integratore I
WHERE I.NomeCommerciale = NomeCommerciale;

/*Calcolo dello sconto scadenza*/
IF GiorniScadenza <= 45 THEN

CASE
WHEN GiorniScadenza BETWEEN 41 AND 45 THEN
SET ScontoScadenza = 10;
WHEN GiorniScadenza BETWEEN 31 AND 40 THEN
SET ScontoScadenza = 20;
WHEN GiorniScadenza BETWEEN 21 AND 30 THEN
SET ScontoScadenza = 40;
WHEN GiorniScadenza BETWEEN 11 AND 20 THEN
SET ScontoScadenza = 60;
WHEN GiorniScadenza BETWEEN 1 AND 10 THEN
SET ScontoScadenza = 80;
END CASE;

END IF;

/*Calcolo dello sconto totale e del nuovo prezzo vendita*/
SET ScontoTotale = ScontoVendite + ScontoScadenza;
SET NuovoPrezzoVendita = PrezzoAcquisto + ((PrezzoVendita-
PrezzoAcquisto)*((100-ScontoTotale)/100)) ;

END $$
DELIMITER ;

/*Funzione che calcola lo sconto nello stesso modo di prima*/
DROP FUNCTION IF EXISTS Sconto;
DELIMITER $$
CREATE FUNCTION Sconto(NomeCommerciale CHAR(100))
RETURNS INT DETERMINISTIC
BEGIN

```

```

DECLARE ScontoVendite INT DEFAULT 0;
DECLARE ScontoScadenza INT DEFAULT 0;
DECLARE ConfezioniVendute INT DEFAULT 0;
DECLARE ConfezioniTotali INT DEFAULT 0;
DECLARE GiorniScadenza INT DEFAULT 0;
DECLARE ScontoTotale INT DEFAULT 0;

SELECT SUM(M.Quantità) INTO ConfezioniTotali
FROM Magazzino M
WHERE M.NomeCommerciale = NomeCommerciale;

SELECT SUM(A.Confezioni) INTO ConfezioniVendute
FROM Acquisto A
WHERE A.NomeCommerciale = NomeCommerciale;

IF ConfezioniVendute <= (10*ConfezioniTotali)/100 THEN
SET ScontoVendite = 20;
END IF;

SELECT DATEDIFF(I.DataScadenza,current_date()) INTO GiorniScadenza
FROM Integratore I
WHERE I.NomeCommerciale = NomeCommerciale;

IF GiorniScadenza <= 45 THEN

CASE
WHEN GiorniScadenza BETWEEN 41 AND 45 THEN
SET ScontoScadenza = 10;
WHEN GiorniScadenza BETWEEN 31 AND 40 THEN
SET ScontoScadenza = 20;
WHEN GiorniScadenza BETWEEN 21 AND 30 THEN
SET ScontoScadenza = 40;
WHEN GiorniScadenza BETWEEN 11 AND 20 THEN
SET ScontoScadenza = 60;
WHEN GiorniScadenza BETWEEN 1 AND 10 THEN
SET ScontoScadenza = 80;
END CASE;

END IF;

SET ScontoTotale = ScontoVendite + ScontoScadenza;

RETURN(ScontoTotale);

END $$
DELIMITER ;

```



```
CREATE OR REPLACE VIEW ConfezioniTotali AS
SELECT M.NomeCommerciale, SUM(M.Quantità) AS ConfezioniTotali
FROM Magazzino M
GROUP BY M.NomeCommerciale;
```

```
CREATE OR REPLACE VIEW ConfezioniVendute AS
SELECT A.NomeCommerciale, SUM(A.Confezioni) AS ConfezioniVendute
FROM Acquisto A
GROUP BY A.NomeCommerciale;
```

```
CREATE OR REPLACE VIEW NuovoPrezzoVendita AS
SELECT I.NomeCommerciale, (I.PrezzoAcquisto + ((I.PrezzoVendita-
I.PrezzoAcquisto)*((100-Sconto(I.NomeCommerciale))/100))) AS
NuovoPrezzoVendita
FROM Integratore I
GROUP BY I.NomeCommerciale;
```

```
CREATE OR REPLACE VIEW MargineTemporale AS
SELECT I.Nomecommerciale, DATEDIFF(I.DataScadenza,current_date) AS
MargineTemporale
FROM Integratore I;
```

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

```
CREATE TEMPORARY TABLE IF NOT EXISTS _IntegratoriTarget(
NomeCommerciale VARCHAR(100) NOT NULL,
PRIMARY KEY(NomeCommerciale)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
TRUNCATE TABLE _IntegratoriTarget;
```

```
INSERT INTO _IntegratoriTarget
SELECT CT.Nomecommerciale
FROM ConfezioniTotali CT
NATURAL JOIN ConfezioniVendute CV
WHERE CV.ConfezioniVendute <= (10*CT.ConfezioniTotali)/100
UNION
SELECT I.Nomecommerciale
FROM Integratore I
WHERE DATEDIFF(I.DataScadenza,current_date) <= 45;
```

```
END $$
DELIMITER ;
```

```

DROP PROCEDURE IF EXISTS RankingIntegratori;
DELIMITER $$
CREATE PROCEDURE RankingIntegratori()
BEGIN

CREATE TEMPORARY TABLE IF NOT EXISTS _RankingIntegratori(
NomeCommerciale VARCHAR(100) NOT NULL,
MargineTemporale INT NOT NULL,
Sconto INT NOT NULL,
NuovoPrezzoVendita INT NOT NULL,
PRIMARY KEY(NomeCommerciale)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

TRUNCATE TABLE _RankingIntegratori;

INSERT INTO _RankingIntegratori
SELECT MT.NomeCommerciale, MT.MargineTemporale,
Sconto(MT.NomeCommerciale) AS Sconto, NPV.NuovoPrezzoVendita
FROM MargineTemporale MT
NATURAL JOIN NuovoprezzoVendita NPV
ORDER BY MT.MargineTemporale ASC;

END $$
DELIMITER ;

```