Music Shop

**Gruppo 10**

Il database serve per gestire una catena nota di negozi nazionale che si occupano della vendita e del noleggio di cd musicali nelle varie città di interesse.

In particolare interessa gestire una catena di filiali che sono in grado di rifornirsi a vicenda. Ogni filiale espone dei cd musicali.

Le filiali si riforniscono a vicenda in caso di urgenza se possiedono la disponibilità dei cd richiesti e solamente se risiedono all’interno della stessa provincia, altrimenti verrà avviata una procedura di rifornimento con il magazzino di riferimento. Occorre anche tener presente che se il magazzino è situato nella stessa provincia di altre due filiali e una delle due avvia una procedura di rifornimento di emergenza, allora le filiali daranno precedenza al magazzino e si riforniranno da quest’ultimo.

Dei magazzini si memorizzano: il codice (univoco) e l’indirizzo (composto da: via, cap, città, num civico e provincia).

Ogni filiale è rifornita da un solo magazzino, mentre un magazzino rifornisce più filiali.

Delle filiali si memorizzano: il codice (univoco), il numero di telefono e l’indirizzo.

Ogni filiale possiede del personale: dirigenti e dipendenti. Un dirigente dirige una e una sola filiale, mentre in una filiale possono lavorare più dipendenti. Dei dipendenti e dei dirigenti si memorizzano: costo orario (un dirigente costerà di più di un dipendente), nome, cognome, codice fiscale, telefono e i dati di residenza. Deve essere quindi possibile calcolare il costo totale del personale per ogni filiale.

Una filiale, oltre a venedere e noleggiare cd musicali, può anche sponsorizzare eventi musicali per aumentare la propria notorietà e quella della catena. In particolare una filiale sponsorizza molti eventi e un evento è sponsorizzato da una sola filiale.

Un evento è caratterizzato da: località, data e nome.

Dei cd musicali interessa memorizzare: il nome, la durata, il numero dei brani, il prezzo cad e la disponibilità di copie all’interno della filiale, in caso di assenza della disponibilità si avvierà una procedura di rifornimento del prodotto.

Viene gestito anche il genere musicale dei cd, del genere si memorizza: il nome (univoco), l’anno di popolarità in cui era in voga il genere musicale.

Un cd musicale può aderire a molti generi musicali e ad un genere musicale possono appartenere molti cd musicali.

Interessano gestire anche le band che hanno scritto i cd, delle quali si memorizzano nome e data di fondazione.

Per band e cd musicali occorre gestire il fatto che i nomi di entrambi non sono univoci.

E’ necessario inoltre gestire le case discografiche, in particolare una band può far parte di una casa discografica e una casa discografica segue molte band differenti. La casa discografica è identificata da: nome (univoco) e anno di fondazione.

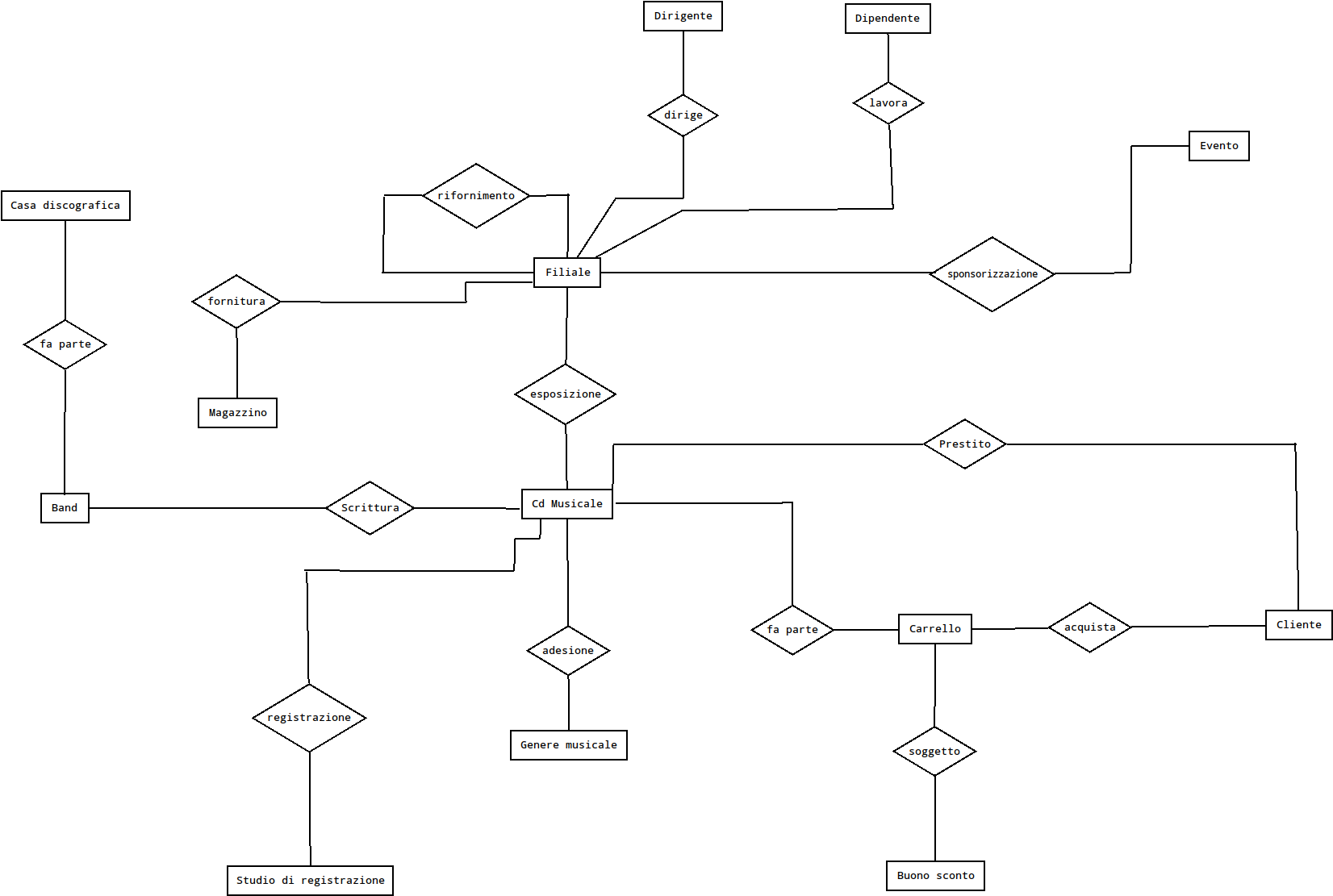
Infine per il cd musicale deve essere anche gestito lo studio di registrazione che si è occupato di produrre il cd. Dello studio di registrazione di memorizzano: nome e e l’indirizzo (composto da: via, cap, città, num civico e provincia). Un cd quindi è registrato da uno studio di registrazione e uno studio di registrazione registra molti cd, occorre tener memorizzato anche il costo della registrazione del cd.

E’ inoltre necessario gestire i clienti che acquistano i cd musicali con i propri rispettivi carrelli di acquisto. Dei clienti si memorizza: codice fiscale, nome, cognome e telefono. Il cliente acquista un carrello in una certa data, il quale contiene dei cd musicali. Il carrello è identificato da un numero (unico se associato a un determinato cliente) e un importo totale.

Inoltre la nota catena di negozi mette a disposizione dei buoni sconto per i vari clienti. Il buono sconto è identificato da un codice (univoco), un attributo booleano che indica se è valido oppure no e il valore dello sconto in percentuale. Il carrello può essere quindi soggetto a un buono sconto, che una volta utilizzato perde di validità.

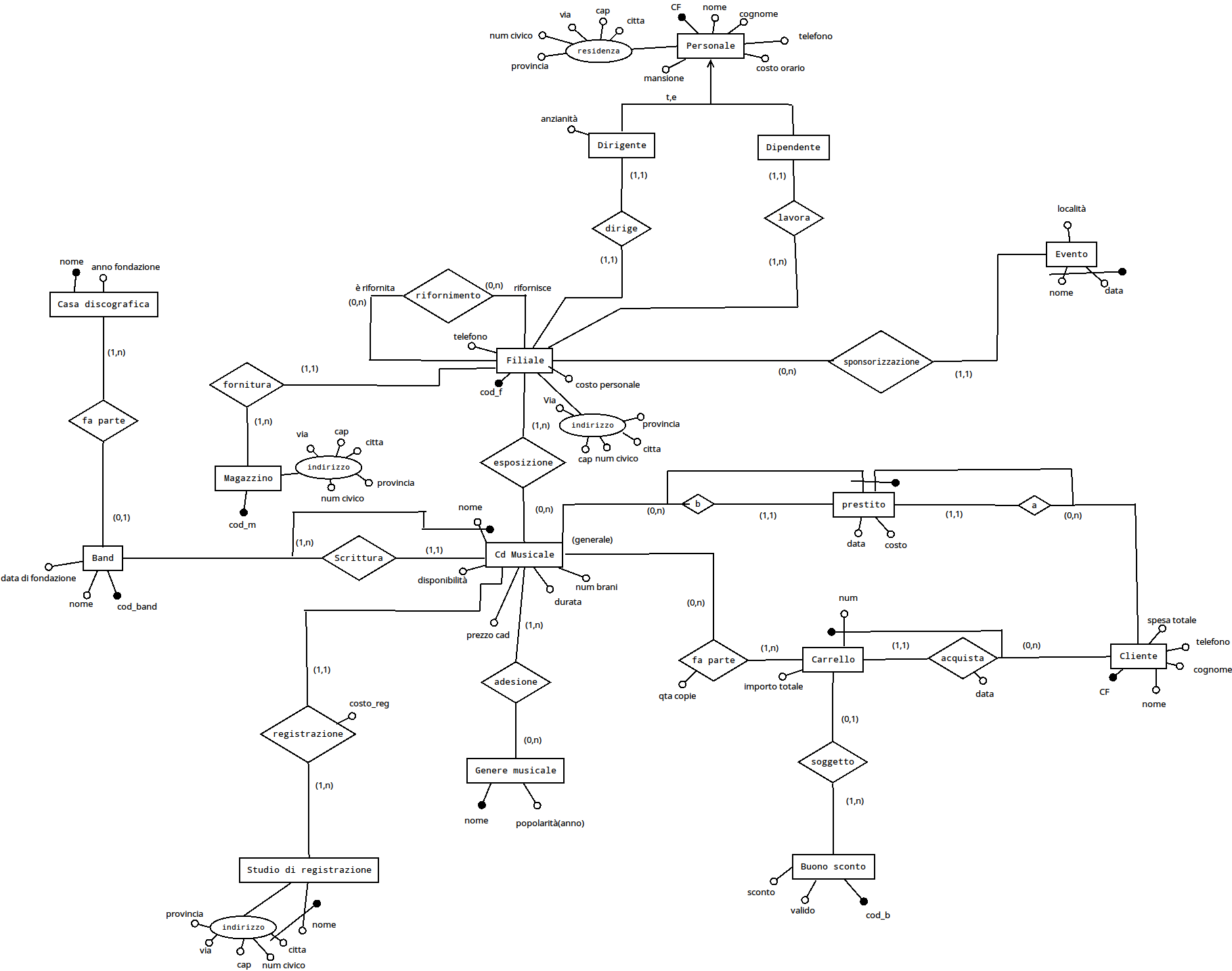
Un cliente deve poter anche prendere in prestito dei cd musicali per poterli ascoltare e decidere se acquistarli successivamente oppure no. In particolare interessa gestire il vincolo che un cliente può prendere in prestito un determinato cd una ed una sola volta, dopodichè, se lo vuole, è costretto ad acquistarlo la volta sucessiva.

# Schema Scheletro



# Progettazione Concettuale

# Schema Er Completo



# Schema logico

CASA\_DISCOGRAFICA(nome, anno\_fondazione)

BAND(cod\_band, nome, data\_fondazione, nome\_c)

FK: nome\_c REFERENCES CASA\_DISCOGRAFICA

STUDIO\_DI\_REGISTRAZIONE(nome, via, cap, citta, num\_civico, provincia)

CD\_MUSICALE(nome, cod\_band, disponibilità, prezzo\_cad, durata, num\_brani, nome\_studio, costo\_reg, citta)

FK: cod\_band REFERENCES BAND

FK: nome\_studio, citta REFERENCES STUDIO\_DI\_REGISTRAZIONE NOT NULL

GENERE\_MUSICALE(nome, popolarità)

ADESIONE(nome, cod\_band, nome\_cd)

FK: nome\_cd, cod\_band REFERENCES CD\_MUSICALE

FK: nome REFERENCES GENERE\_MUSICALE

MAGAZZINO(cod\_m, via, cap, citta, num\_civico, provincia)

DIRIGENTE(cf, nome, cognome, costo\_orario, telefono, via, cap, citta, num\_civico, provincia, anzianità)

FILIALE(cod\_f, costo\_personale, telefono, via, cap, citta, num\_civico, provincia, cod\_m)

FK: cod\_m REFERENCES MAGAZZINO NOT NULL

DIRIGE(cod\_f, cf\_dir)

FK: cod\_f REFERENCES FILIALE

FK: cf\_dir REFERENCES DIRIGENTE

AK: cf\_dir

DIPENDENTE(cf, nome, cognome, costo\_orario, telefono, via, cap, citta, num\_civico, provincia, cod\_f)

FK: cod\_f REFERENCES FILIALE NOT NULL

RIFORNIMENTO(cod\_fornitore, cod\_rifornisce)

FK: cod\_fornitore REFERENCES FILIALE

FK: cod\_rifornisce REFERENCES FILIALE

EVENTO(nome, data, località, cod\_f)

FK: cod\_f REFERENCES FILIALE NOT NULL

ESPOSIZIONE(nome\_cd, cod\_band, cod\_f)

FK: nome\_cd, cod\_band REFERENCES CD\_MUSICALE

FK: cod\_f REFERENCES FILIALE

CLIENTE(cf, nome, cognome, telefono, spesa\_totale)

PRESTITO(data, cf, nome\_cd, cod\_band, costo)

FK: cf REFERENCES CLIENTE

FK: nome\_cd, cod\_band REFERENCES CD\_MUSICALE

BUONO\_SCONTO(cod\_b, sconto, valido)

CARRELLO(num, cf, data, importo\_totale, cod\_b)

FK: cf REFERENCES CLIENTE

FK: cod\_b REFERENCES BUONO\_SCONTO

FA\_PARTE(num, cf, nome\_cd, cod\_band, qta\_copie)

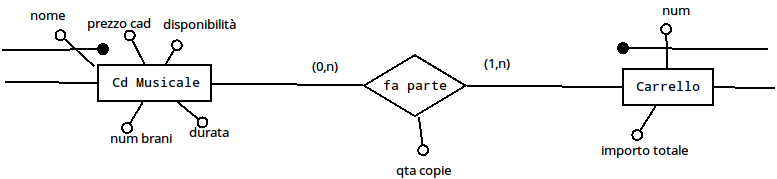
FK: num, cf REFERENCES CARRELLO

FK: nome\_cd, cod\_band REFERENCES CD\_MUSICALE

# Studio dati derivati

Nel progetto sono presenti diversi dati derivati, si è deciso di limitare lo studio a due soli di questi. In particolare si è studiato il dato derivato “importo totale” presente nell’entità carrello e “costo personale” presente nell’entità filiale.

1. Il dato derivato “importo totale” è calcolato come la somma dei prezzi di tutti i prodotti molitiplicato la quantità rispettiva.



**Tabella dei volumi**

Si è considerata una gamma di scelta di 10000 cd differenti, in media un carrello contiene 2.5 cd e un cd fa parte di 3 carrelli (con 30000 record nell’associazione).

|  |  |  |
| --- | --- | --- |
| Concetto | Tipo | Volume |
| Cd Musicale | E | 10000 |
| Fa\_parte | R | 30000 |
| Carrello | E | 12000 |

**Tabella delle operazioni**

Sono state considerate due operazioni:

1. Visualizzare importo totale di un carrello
2. Dato cd aggiungerlo a un carrello (già dato)

Si è supposto che l’operazione di visualizzazione dell’importo totale avvenga con minore frequenza rispetto all’aggiunta di un cd nel carrello.

|  |  |  |
| --- | --- | --- |
| Operazione | Tipo | Frequenza |
| Op1 | I | 400/g |
| Op2 | I | 500/g |

**Tabella accessi con dato derivato**

Si suppone che i dati del cd siano già dati (prezzo compreso), nell’operazione 2 non sarà quindi necessario andare ad effettuare una lettura sull’entità cd.

|  |  |  |  |
| --- | --- | --- | --- |
| Operazione | Concetto | Accessi | Tipo |
| Op1 | Carrello | 1 | L |
| Op2 | Fa\_parte | 1 | S |
| Op2 | Carrello | 1 | L |
| Op2 | Carrello | 1 | S |

Costo: 400+2500 = 2900/g

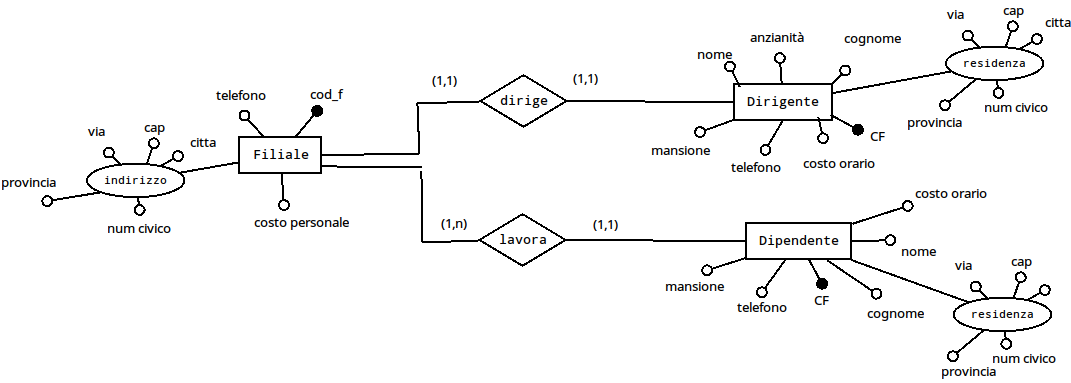
**Tabella accessi senza dato derivato**

|  |  |  |  |
| --- | --- | --- | --- |
| Operazione | Concetto | Accessi | Tipo |
| Op1 | Carrello | 1 | L |
| Op1 | Fa\_parte | 2.5 | L |
| Op1 | Cd Musicale | 2.5 | L |
| Op2 | Fa\_parte | 1 | S |

Costo: 2400+1000 = 3400/g

In questo caso conviene tenere il dato derivato “importo totale”.

2) Il dato derivato “costo personale” dell’entità “filiale” è calcolato come la somma dei costi orari di tutti i dipendenti e dirigenti che lavorano in quella fialiale. Lo studio sul dato derivato è stato effettuato sullo schema er seguente, che rappresenta la forma d’implementazione scelta.



**Tabella dei volumi**

Si è considerata una catena di negozi abbastanza famosa con 200 filiali differenti sparse per la nazione. Ogni filiale è seguita da un dirigente diverso, ci saranno quindi 200 dirigenti. In una filiale lavorano più dipendenti, ma un dipendente lavora per una e una sola filiale. Sono stati considerati in media 30 dipendenti per filiale.

|  |  |  |
| --- | --- | --- |
| Concetto | Tipo | Volume |
| Filiale | E | 200 |
| Dirige | R | 200 |
| Dirigente | E | 200 |
| Dipendente | E | 6000 |
| Lavora | R | 6000 |

**Tabella delle operazioni**

Sono state considerate tre operazioni:

1. Modifica del costo orario di un dipendente, dato codice fiscale
2. Modifica del costo orario di un dirigente, dato codice fiscale
3. Controllo sul costo del personale di una determinata filiale

Sono tutte operazioni che avvengono abbastanza di rado in una azienda, ma quando si parla di una catena di negozi queste aumentano all’aumentare delle sedi e dei dipendenti, è stato quindi necessario tenerne conto.

|  |  |  |
| --- | --- | --- |
| Operazione | Tipo | Frequenza |
| Op1 | I | 10/g |
| Op2 | I | 2/g |
| Op3 | I | 5/g |

**Tabella accessi con dato derivato**

|  |  |  |  |
| --- | --- | --- | --- |
| Operazione | Concetto | Accessi | Tipo |
| Op1 | Dipendente | 1 | L |
| Op1 | Dipendente | 1 | S |
| Op1 | Lavora | 1 | L |
| Op1 | Filiale | 1 | L |
| Op1 | Filiale | 1 | S |
| Op2 | Dirigente | 1 | L |
| Op2 | Dirigente | 1 | S |
| Op2 | Dirige | 1 | L |
| Op2 | Filiale | 1 | L |
| Op2 | Filiale | 1 | S |
| Op3 | Filiale | 1 | L |

Costo: 70+14+5 = 89/g

**Tabella accessi senza dato derivato**

|  |  |  |  |
| --- | --- | --- | --- |
| Operazione | Concetto | Accessi | Tipo |
| Op1 | Dipendente | 1 | L |
| Op1 | DIpendente | 1 | S |
| Op2 | Dirigente | 1 | L |
| Op2 | Dirigente | 1 | S |
| Op3 | Filiale | 1 | L |
| Op3 | Dirige | 1 | L |
| Op3 | Dirigente | 1 | L |
| Op3 | Lavora | 30 | L |
| Op3 | Dipendente | 30 | L |

Costo: 30+6+315 = 351/g

Come mostra lo studio appena svolto, inaspettatamente, conviene tenere il dato derivato in questo caso.

# Query di creazione

Di seguito si presentano le query per la creazione dell’intero database. Si evidenziano in verde i vincoli imposti non esprimibili durante la fase di progettazione.

CREATE TABLE CASA\_DISCOGRAFICA(

nome VARCHAR(30) PRIMARY KEY,

anno\_fondazione INTEGER NOT NULL

);

CREATE TABLE BAND(

cod\_band INTEGER PRIMARY KEY,

nome VARCHAR(30) NOT NULL,

data\_fondazione date NOT NULL,

nome\_c VARCHAR(30) NOT NULL,

FOREIGN KEY (nome\_c) REFERENCES CASA\_DISCOGRAFICA(nome)

ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE STUDIO\_DI\_REGISTRAZIONE(

nome VARCHAR(30) NOT NULL,

via VARCHAR(60) NOT NULL,

cap CHAR(5) NOT NULL,

citta VARCHAR(60) NOT NULL,

num\_civico INTEGER NOT NULL,

provincia VARCHAR(60) NOT NULL,

PRIMARY KEY(nome,citta),

CHECK(num\_civico > 0 )

);

CREATE TABLE CD\_MUSICALE(

nome VARCHAR(60) NOT NULL,

cod\_band INTEGER NOT NULL,

nome\_studio VARCHAR(60) NOT NULL,

prezzo\_cad INTEGER NOT NULL,

durata TIME NOT NULL,

num\_brani INTEGER NOT NULL,

costo\_reg DECIMAL(10,3) NOT NULL,

citta VARCHAR(60) NOT NULL,

PRIMARY KEY(nome, cod\_band),

FOREIGN KEY (nome\_studio, citta) REFERENCES STUDIO\_DI\_REGISTRAZIONE(nome, citta)

ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (cod\_band) REFERENCES BAND(cod\_band)

ON DELETE CASCADE ON UPDATE CASCADE,

CHECK(prezzo\_cad > 0),

CHECK(durata > '00:02:00' ),

CHECK(num\_brani > 0)

);

CREATE TABLE GENERE\_MUSICALE(

nome VARCHAR(60) PRIMARY KEY,

popolarita INTEGER NOT NULL,

CHECK(popolarita > 1500)

);

CREATE TABLE ADESIONE(

genere VARCHAR(60) NOT NULL,

cod\_band INTEGER NOT NULL,

nome\_cd VARCHAR(60) NOT NULL,

PRIMARY KEY(genere, cod\_band, nome\_cd),

FOREIGN KEY (nome\_cd, cod\_band) REFERENCES CD\_MUSICALE(nome, cod\_band)

ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (genere) REFERENCES GENERE\_MUSICALE(nome)

ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE MAGAZZINO(

cod\_m INTEGER PRIMARY KEY,

via VARCHAR(60) NOT NULL,

cap CHAR(5) NOT NULL,

citta VARCHAR(60) NOT NULL,

num\_civico INTEGER NOT NULL,

provincia VARCHAR(60) NOT NULL,

CHECK(num\_civico > 0 )

);

CREATE TABLE DIRIGENTE(

cf CHAR(16) PRIMARY KEY,

nome VARCHAR(60) NOT NULL,

cognome VARCHAR(60) NOT NULL,

costo\_orario DECIMAL(10,3) NOT NULL,

telefono VARCHAR(10),

via VARCHAR(60) NOT NULL,

cap CHAR(5) NOT NULL,

citta VARCHAR(60) NOT NULL,

num\_civico INTEGER NOT NULL,

provincia VARCHAR(60) NOT NULL,

anzianita INTEGER NOT NULL,

CHECK(num\_civico > 0 ),

CHECK(anzianita > 0 ),

CHECK(costo\_orario > 0)

);

CREATE TABLE FILIALE(

cod\_f INTEGER PRIMARY KEY,

costo\_personale DECIMAL(15,3) NOT NULL,

telefono VARCHAR(10) NOT NULL,

via VARCHAR(60) NOT NULL,

cap CHAR(5) NOT NULL,

citta VARCHAR(60) NOT NULL,

num\_civico INTEGER NOT NULL,

provincia VARCHAR(60) NOT NULL,

cod\_m INTEGER NOT NULL,

FOREIGN KEY (cod\_m) REFERENCES MAGAZZINO(cod\_m)

ON DELETE CASCADE ON UPDATE CASCADE,

CHECK(num\_civico > 0),

CHECK(costo\_personale >= 0)

);

CREATE TABLE DIRIGE(

cod\_f INTEGER PRIMARY KEY,

cf\_dir CHAR(16) NOT NULL,

UNIQUE(cf\_dir),

FOREIGN KEY (cod\_f) REFERENCES FILIALE(cod\_f)

ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (cf\_dir) REFERENCES DIRIGENTE(cf)

ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE DIPENDENTE(

cf CHAR(16) PRIMARY KEY,

nome VARCHAR(60) NOT NULL,

cognome VARCHAR(60) NOT NULL,

costo\_orario DECIMAL(10,3) NOT NULL,

telefono VARCHAR(10),

via VARCHAR(60) NOT NULL,

cap CHAR(5) NOT NULL,

citta VARCHAR(60) NOT NULL,

num\_civico INTEGER NOT NULL,

provincia VARCHAR(60) NOT NULL,

cod\_f INTEGER NOT NULL,

FOREIGN KEY (cod\_f) REFERENCES FILIALE(cod\_f)

ON DELETE CASCADE ON UPDATE CASCADE,

CHECK(num\_civico > 0 ),

CHECK(costo\_orario > 0)

);

CREATE TABLE RIFORNIMENTO(

cod\_fornitore INTEGER NOT NULL,

cod\_rifornisce INTEGER NOT NULL,

PRIMARY KEY(cod\_fornitore, cod\_rifornisce),

FOREIGN KEY (cod\_fornitore) REFERENCES FILIALE(cod\_f)

ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (cod\_rifornisce) REFERENCES FILIALE(cod\_f)

ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE EVENTO(

nome VARCHAR(60) NOT NULL,

data DATE NOT NULL,

localita VARCHAR(60) NOT NULL,

cod\_f INTEGER NOT NULL,

PRIMARY KEY(nome, data),

FOREIGN KEY (cod\_f) REFERENCES FILIALE(cod\_f)

ON DELETE CASCADE ON UPDATE CASCADE,

CHECK(LENGTH(localita) >= 2),

CHECK(LENGTH(nome) >= 3)

);

CREATE TABLE ESPOSIZIONE(

nome\_cd VARCHAR(60) NOT NULL,

cod\_band INTEGER NOT NULL,

cod\_f INTEGER NOT NULL,

quantita INTEGER,

PRIMARY KEY(nome\_cd, cod\_band, cod\_f),

FOREIGN KEY (nome\_cd, cod\_band) REFERENCES CD\_MUSICALE(nome, cod\_band)

ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (cod\_f) REFERENCES FILIALE(cod\_f)

ON DELETE CASCADE ON UPDATE CASCADE,

CHECK (quantita >= 0)

);

CREATE TABLE CLIENTE(

cf CHAR(16) PRIMARY KEY,

nome VARCHAR(60) NOT NULL,

cognome VARCHAR(60) NOT NULL,

telefono VARCHAR(10),

spesa\_totale DECIMAL(10,3) NOT NULL

);

CREATE TABLE PRESTITO(

data date NOT NULL,

cf CHAR(16) NOT NULL,

nome\_cd VARCHAR(60) NOT NULL,

cod\_band INTEGER NOT NULL,

costo DECIMAL (10,3) NOT NULL,

cod\_f INTEGER NOT NULL,

PRIMARY KEY(cf, nome\_cd, cod\_band),

FOREIGN KEY (nome\_cd, cod\_band) REFERENCES CD\_MUSICALE(nome, cod\_band)

ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (cf) REFERENCES CLIENTE(cf)

ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (cod\_f) REFERENCES FILIALE(cod\_f)

ON DELETE CASCADE ON UPDATE CASCADE,

CHECK(costo > 0)

);

CREATE TABLE BUONO\_SCONTO(

cod\_b INTEGER PRIMARY KEY,

sconto DECIMAL(3,2),

valido BOOLEAN NOT NULL,

CHECK(sconto <= 1)

);

CREATE TABLE CARRELLO(

num INTEGER NOT NULL,

cf CHAR(16) NOT NULL,

data date NOT NULL,

importo\_totale DECIMAL(10,3),

cod\_b INTEGER NOT NULL,

cod\_f INTEGER NOT NULL,

PRIMARY KEY(num, cf),

FOREIGN KEY (cf) REFERENCES CLIENTE(cf)

ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (cod\_b) REFERENCES BUONO\_SCONTO(cod\_b)

ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (cod\_f) REFERENCES FILIALE(cod\_f)

ON DELETE CASCADE ON UPDATE CASCADE,

CHECK(importo\_totale > 0),

CHECK(num > 0)

);

CREATE TABLE FA\_PARTE(

num INTEGER NOT NULL,

cf CHAR(16) NOT NULL,

nome\_cd VARCHAR(60) NOT NULL,

cod\_band INTEGER NOT NULL,

qta\_copie INTEGER NOT NULL,

PRIMARY KEY(num, cf, nome\_cd, cod\_band),

FOREIGN KEY (num, cf) REFERENCES CARRELLO(num, cf)

ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (nome\_cd, cod\_band) REFERENCES CD\_MUSICALE(nome, cod\_band)

ON DELETE CASCADE ON UPDATE CASCADE,

CHECK(qta\_copie > 0)

);