

Progetto di Basi di Dati 2015

Davide Polonio, Alessandro Bari

July 6, 2015

Contents

1	Progettazione Concettuale	3
1.1	Studio di fattibilità	3
1.2	Abstract	3
1.3	Raccolta e analisi dei requisiti	3
1.3.1	Analisi dei requisiti	3
1.3.2	Glossario dei termini	4
1.4	Schema Concettuale	4
1.4.1	Lista delle Entità	4
1.4.2	Lista delle Relazioni	6
1.5	Schema E-R (prima della ristrutturazione)	8
1.6	Schema E-R (dopo la ristrutturazione)	9
2	Progettazione Logica	11
2.0.1	Modello Relazionale	11
2.0.2	Create Table	12
2.0.3	Procedure	15

1 Progettazione Concettuale

1.1 Studio di fattibilità

Priorità di realizzazione Come priorità di realizzazione abbiamo deciso di implementare le seguenti feature:

- Percentuale di sconto basata sul numero di acquisti effettuato da un acquirente iscritto al servizio.
- Gestione degli ordini e delle fatture del negozio, oltre che degli scontrini e delle vendite effettuate riferite agli iscritti.
- Gestione e organizzazione dei turni per i dipendenti.
- Catalogazione prodotti in diverse categorie (a cui appartengono diverse percentuali di sconto).

1.2 Abstract

Negozio "Linea Casa Bari" Si vuole realizzare una base di dati per un negozio al dettaglio che vende oggetti per la casa, organizzati per categorie; esse sono gestite dai dipendenti in base a una suddivisione per turni. Vogliamo inoltre tenere aggiornata la lista dei prodotti in base ai nuovi ordini ai fornitori attestati da fatture. Riguardo ai clienti iscritti al servizio sconti del negozio, vogliamo registrare gli acquisti eseguiti e calcolare la rispettiva percentuale di sconto raggiunta.

1.3 Raccolta e analisi dei requisiti

1.3.1 Analisi dei requisiti

Per i clienti **iscritti** identificati da un codice, vogliamo tenere conto dell'identità e dei suoi acquisti effettuati.

Ogni **prodotto** è identificato da un codice, vogliamo tenere conto delle informazioni base e della quantità disponibile e catalogarlo in una delle sei **categorie**, identificate come: porcellane, pentolame, liste nozze, tovaglie, tavola, paralumi. Ad ogni prodotto è associata una specifica che lo descrive accompagnando la descrizione da una foto.

Ad ogni categoria sono associati possibili **sconti** in base ad una distribuzione per livelli e con una relativa percentuale.

I **dipendenti**, responsabili ognuno di una singola categoria, sono identificati da un codice e sono organizzati per turni.

Vogliamo tener conto delle **fatture** ai rispettivi **fornitori** dei quali si necessita solamente del nome e delle informazioni base per la descrizione in fattura.

1.3.2 Glossario dei termini

Termine	Descrizione	Sinonimi	Collegamenti
Iscritto	Compratore abituale iscritto a questa lista per avere diritto a sconti speciali. Può essere un dipendente.	Cliente abituale	Scontrino
Scontrino	Scontrino attestante lo storico degli acquisti.	Storico, acquisto	Iscritto, Prodotto
Categoria	Sei insiemi di prodotti. Un prodotto può appartenere ad un'unica categoria. Ogni categoria ha il suo univoco responsabile.		Sconto, Dipende, Prodotto.
Sconto	Spetta solamente al cliente iscritto. Per ogni categoria esistono diversi livelli in base agli acquisti, a cui corrisponde una percentuale di sconto.		Categoria.
Dipendente	Responsabile di una singola categoria. Può essere un cliente ma non un fornitore.	Responsabile	Categoria
Prodotto	Ogni prodotto può appartenere ad una sola categoria.	Oggetti, Prodotto ordinati o acquistati	Categoria, Fattura, Scontrino
Specifica	Ogni prodotto ha una specifica diversa.	Descrizione	Prodotto
Fattura	Più unità di prodotto possono essere ordinate a fornitori diversi. Modifica il campo quantità disponibile di prodotto.	Ordine	Fornitore, Prodotto
Fornitore	Forniscono i prodotti attraverso gli ordini. Un fornitore non può essere un cliente e può fornire prodotti di diverse categorie		Fattura

1.4 Schema Concettuale

1.4.1 Lista delle Entità

- Dipendente: lista dei dipendenti del negozio.
 - Codice dipendente SMALLINT
 - Informazioni: dati anagrafici e di recapito del dipendente
 - * Nome CHAR(15)
 - * Cognome CHAR(15)
 - * Data di nascita DATE
 - * Codice Fiscale CHAR(16)
 - * Telefono CHAR(10)

- * E-mail CHAR(50)
- Indirizzo:
 - * Via CHAR(50)
 - * Città CHAR(50)
 - * Provincia CHAR(50)
- Categoria: insieme di prodotti
 - Nome Categoria CHAR(50)
- Sconto: entità destinata a contenere tutti i gradi di sconto di tutte le categorie
 - Livello SMALLINT
 - PercentualeSconto INT(2)
 - Tetto Max SMALLINT
- Prodotto: lista di tutti i prodotti in vendita
 - Codice Prodotto INT
 - Quantità SMALLINT
 - Percentuale IVA INT(2)
- Specifica: descrizione base sul prodotto
 - Nome CHAR(50)
 - Descrizione TEXT
- Fattura: contenente tutti gli attestati di avvenuto ordine per un certo numero di prodotti
 - Codice Fattura INT
 - Quantità SMALLINT
 - Data DATE
- Fornitore: lista di tutti i venditori da cui il negozio acquista i prodotti
 - Nome CHAR(50)
 - Contatto
 - * Fax CHAR(10)
 - * Telefono CHAR(10)
 - * E-mail CHAR (10)
 - Indirizzo:
 - * Via CHAR(50)
 - * Città CHAR(50)
 - * Provincia CHAR (50)

Di fornitore sono presenti le seguenti generalizzazioni:

- Artigiano: Fornisce prodotti fatti a mano
- Grossista: Fornisce prodotti all'ingrosso

- Scontrino: registro di tutte le vendite effettuate dai clienti iscritti
 - Codice Scontrino INT
 - Data DATE
 - Quantità SMALLINT
 - SubTotale DECIMAL(5,2)
- Cliente: lista dei clienti, di cui abbiamo creato la seguente generalizzazione parziale:
 - Iscritto: clienti iscritti
 - * Codice Iscritto INT
 - * Indirizzo:
 - Via CHAR(50)
 - Città CHAR(50)
 - Provincia CHAR(50)
 - * Contatto
 - Fax CHAR(20)
 - Telefono CHAR(20)
 - E-mail CHAR (50)
 - * Identità
 - Nome CHAR(20)
 - Cognome CHAR(20)

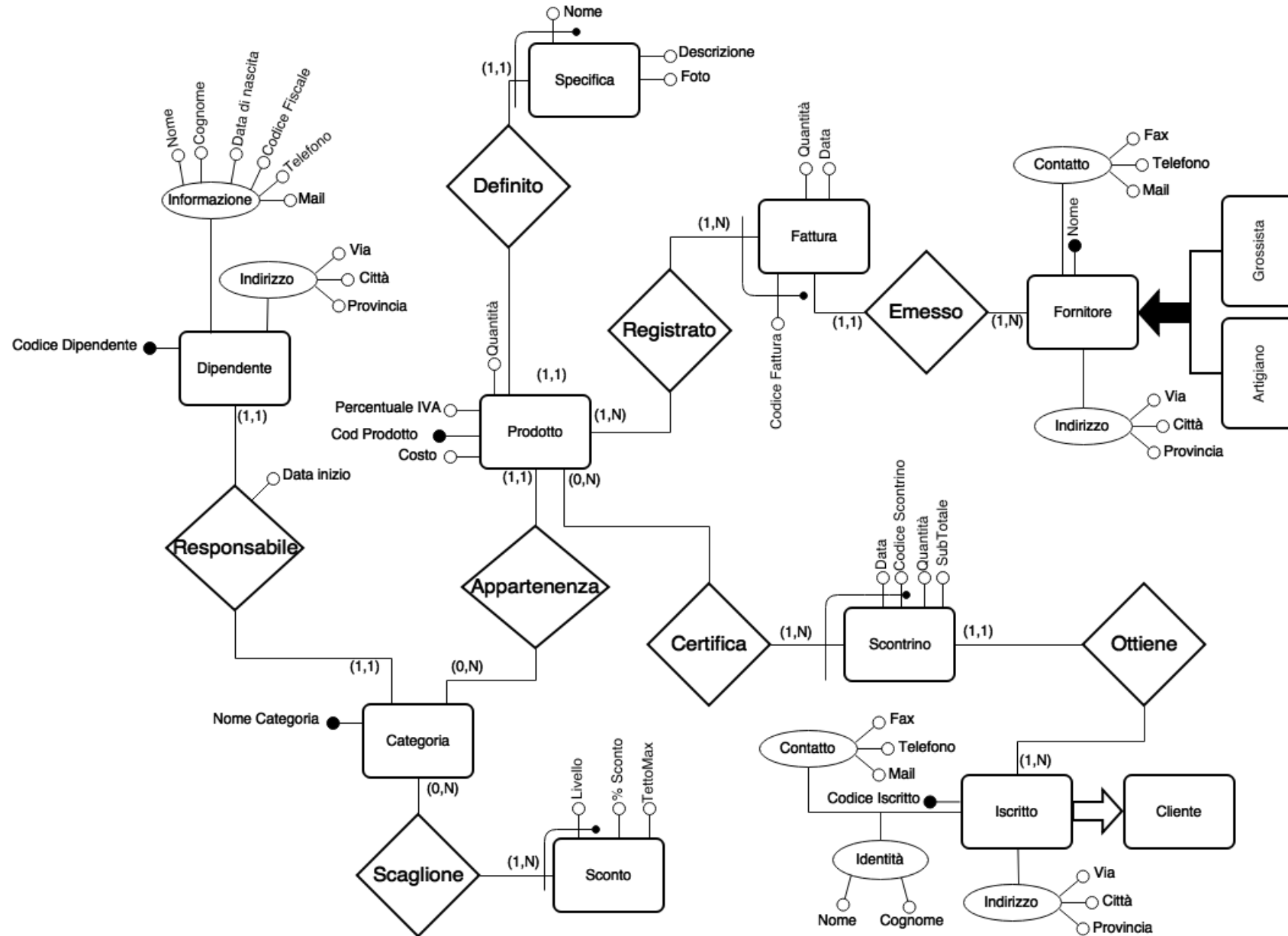
1.4.2 Lista delle Relazioni

- Responsabile: relazione tra Dipendente-Categoria.
 - Presenza di due attributi:
 - * Data di inizio DATE

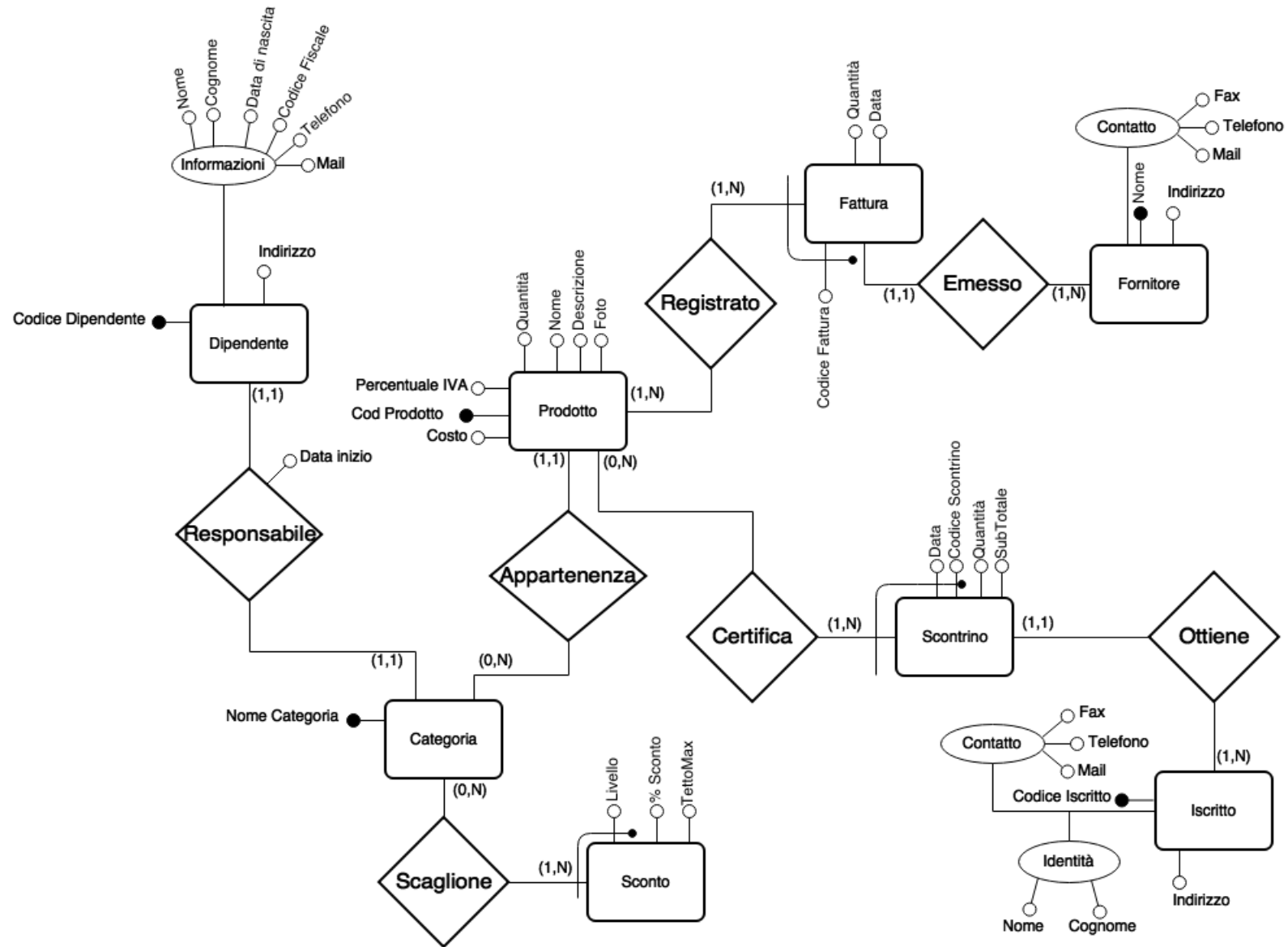
La cardinalità è (1,1) in quanto ogni dipendente è responsabile solamente di una categoria e vi lavora in una determinata data. I dipendenti in sé possono ovviamente cambiare nel tempo.
- Scaglioni: relazione tra Categoria-Sconto. È una cardinalità (0,N) da Categoria \rightarrow Sconto, mentre la cardinalità risulta essere (1,N) da Sconto \rightarrow Categoria.
Ogni Categoria presenta dei diversi scaglioni di sconti in base al livello di acquisto.
- Appartenenza: relazione tra Categoria-Prodotto. Da Prodotto \rightarrow Categoria abbiamo imposto una cardinalità di tipo (1,1) in quanto un Prodotto deve appartenere a una ed una sola Categoria.
La cardinalità è (0,N) da Categoria \rightarrow Prodotto in quanto una nuova categoria può non contenere prodotti.
Per esempio se un dipendente acquistasse tutti i prodotti di una certa categoria questa risulterebbe vuota fino a nuova fattura.
- Definito: relazione tra Prodotto-Specifica. La cardinalità tra Prodotto e Specifica è (1,1) in quanto ogni prodotto ha una singola specifica.

- Registrato: relazione tra Fattura-Prodotto. Da Prodotto \rightarrow Fattura abbiamo una relazione di tipo (1,N) in quanto un prodotto deve risultare registrato in una fattura.
Da Fattura \rightarrow Prodotto la cardinalità è (1,N) perchè una fattura per essere emessa deve contenere uno o più prodotti
- Emesso: relazione tra Fattura-Fornitore. Da fattura a fornitore la cardinalità è (1,1): una fattura può essere solamente emessa da un singolo fornitore.
Da Fornitore \rightarrow a Fattura è (1,N) in quanto viene memorizzato solamente un fornitore che abbia almeno emesso una o più fatture al negozio.
- Certifica: relazione tra Prodotto-Scontrino. Ha cardinalità (0,N) in Prodotto \rightarrow Scontrino siccome un prodotto può esser stato acquistato da zero a più volte, invece da Scontrino \rightarrow Prodotto vi è una cardinalità (1,N), uno scontrino infatti certifica almeno un prodotto.
- Ottiene: relazione tra Scontrino-Cliente. In Scontrino \rightarrow Cliente vi è una relazione (1,1) in quanto uno scontrino si riferisce ad un singolo acquirente, mentre tra Cliente \rightarrow Scontrino vi è una relazione (1,N) ne consegue che un cliente può fare più acquisti e per essere definito tale deve aver almeno compiuto un acquisto.

1.5 Schema E-R (prima della ristrutturazione)



1.6 Schema E-R (dopo la ristrutturazione)



Note post ristrutturazione

- Nell'attributo composto Informazione in DIPENDENTE abbiamo notato la ridondanza tra Nome, Cognome, Data di nascita con Codice Fiscale, ma abbiamo deciso di mantenere questi campi in quanto ricavarci questi dati da Codice Fiscale risulterebbe essere un'operazione onerosa ed inoltre non sempre potrebbe essere corretta (vedasi i casi di omocodia). Sempre su DIPENDENTE abbiamo deciso di accorpare l'attributo composto Indirizzo in un unico attributo in quanto ci è sufficiente l'accesso all'informazione complessiva
- Abbiamo deciso di non accorpare DIPENDENTE in CATEGORIA in quanto, accedendo a CATEGORIA, non sempre vogliamo accedere agli attributi di DIPENDENTE; rendendo il tutto più modulare e velocizzando le query che usano solamente gli attributi di CATEGORIA.
- L'entità SPECIFICA e la relazione DEFINITO sono stati rimossi accorpendo gli attributi nome, descrizione nella tabella prodotto, in quanto le relazioni più frequenti su PRODOTTO richiedono appunto questi dati. Così facendo vengono risparmiati molti accessi altrimenti necessari.
- Su PRODOTTO abbiamo lasciato l'attributo Quantità in quanto, al crescere del numero di scontrini e fatture, calcolare il numero di un determinato prodotto sarebbe diventato troppo oneroso dal punto di vista computazionale, quindi abbiamo deciso di lasciare questa rindondanza.
- Su FORNITORE è stata tolta la generalizzazione totale, in quanto la differenza tra le produzioni di un artigiano e di un grossista non sono richieste nell'analisi dei requisiti.
- Per quanto riguarda ISCRITTO abbiamo deciso di voler conoscere solo un contatto e un indirizzo principale. Anche qui la generalizzazione è stata tolta, in quanto nell'analisi dei requisiti abbiamo imposto che gli sconti fossero riferiti solamente ai clienti iscritti.

2 Progettazione Logica

2.0.1 Modello Relazionale

Seguendo i procedimenti di trasformazione dello schema-ER al modello Relazionale abbiamo ottenuto:

PRODOTTO (CodProdotto, Nome, Descrizione, Quantità, Costo, PercentualeIVA, Categoria)

SCONTRINO (Prodotto, Data, CodScontrino, Quantità, Subtotale, Iscritto)

CERTIFICA (Prodotto, Data, CodScontrino)

FATTURA (Prodotto, CodFattura, Data, Quantità, Fornitore)

REGISTRATO (Prodotto, Codice Fattura)

CATEGORIA (Nome Categoria)

SCONTO (Categoria, Livello, PercentualeSconto, TettoMax)

SCAGLIONI (Categoria, Livello)

ISCRITTO (CodIscritto, Nome, Cognome, Fax, Telefono, Mail, Indirizzo)

FORNITORE (Nome, Fax, Telefono, Mail, Indirizzo)

DIPENDENTE (CodDipendente, Nome, Cognome, Data Nascita, Codice Fiscale, Telefono, Mail, Data Inizio, Categoria)

Qui vengono espressi i vincoli non esprimibili nel modello relazionale:

- In PRODOTTO vincolo d'integrità di categoria con CATEGORIA
- In SCONTRINO vincolo d'integrità di prodotto con CERTIFICA
- In SCONTRINO vincolo d'integrità di iscritto con ISCRITTO
- In CERTIFICA vincolo d'integrità di prodotto con PRODOTTO
- In CERTIFICA vincolo d'integrità di codice scontrino con SCONTRINO
- In CERTIFICA vincolo d'integrità di data con SCONTRINO
- In FATTURA vincolo d'integrità di prodotto con REGISTRATO
- In FATTURA vincolo d'integrità di fornitore con FORNITORE
- In REGISTRATO vincolo d'integrità di prodotto con PRODOTTO
- In REGISTRATO vincolo d'integrità di codice fattura con FATTURA
- In SCONTO vincolo d'integrità di categoria con SCAGLIONI
- In SCAGLIONI vincolo d'integrità di categoria con CATEGORIA
- In SCAGLIONI vincolo d'integrità di livelli con SCONTO
- In DIPENDENTE vincolo d'integrità di categoria con CATEGORIA

Inoltre per alcuni attributi abbiamo attribuito le seguenti proprietà:

- In PRODOTTO Descrizione, Foto possono essere NULL
- In ISCRITTO Fax può essere NULL

- In FORNITORE Fax può essere NULL
- In DIPENDENTE Mail può essere NULL
- In SCONTO il tetto-max può essere NULL

2.0.2 Create Table

Prima di creare le tabelle abbiamo inserito le seguenti righe di codice:

```

1 drop table if exists Categoria;
2 drop table if exists Sconto;
3 drop table if exists Scaglione;
4 drop table if exists Dipendente;
5 drop table if exists Prodotto;
6 drop table if exists Scontrino;
7 drop table if exists Iscritto;
8 drop table if exists Fornitore;
9 drop table if exists Fattura;
10 drop table if exists Registrato;
11 drop table if exists Certifica;

```

In modo da non incorrere in errori in caso di ricreazione delle tabelle. I create table veri e propri sono i seguenti:

```

1 CREATE TABLE Categoria (
2 NomeCategoria CHAR(50),

```

crea la tabella Categoria con chiave primaria NomeCategoria;

```

1 PRIMARY KEY (NomeCategoria)
2 ) ENGINE=InnoDB;
3
4 CREATE TABLE Sconto (
5 Id INT AUTOINCREMENT,
6 Livello SMALLINT DEFAULT 0 NOT NULL,
7 PercSconto INT(2) DEFAULT 0 NOT NULL,
8 TettoMax SMALLINT,
9
10 PRIMARY KEY(Id)
11 ) ENGINE=InnoDB;

```

crea la tabella Sconto che ha come chiave primaria l'attributo Id;

```

1 CREATE TABLE Scaglione (
2 Categoria CHAR(50),
3 Sconto INT,
4
5 PRIMARY KEY (Categoria, Sconto),
6 FOREIGN KEY (Categoria) REFERENCES Categoria (NomeCategoria)
7 ON DELETE CASCADE,
8 FOREIGN KEY (Sconto) REFERENCES Sconto(Id) ON DELETE CASCADE
9 ) ENGINE=InnoDB;

```

Crea la tabella Scaglione (relazione tra Sconto e Categoria), ha come chiave entrambi i suoi campi (Categoria e Sconto) che sono chiave esterna per Categoria e Sconto rispettivamente. Sia Categoria che Sconto hanno il vincolo DELETE ON CASCADE in quanto vogliamo che alla cancellazione di una Categoria anche gli sconti a questa associata vengano cancellati (con l'intervento aggiuntivo del trigger delete_categoria spiegato successivamente), mentre per l'eliminazione di Sconto deve essere eliminata la tupla corrispondente di Scaglione ma mantenuta la Categoria associata (in quanto potrebbe essere non vuota);

```

1 CREATE TABLE Dipendente (
2   CodDipendente    INT AUTO_INCREMENT,
3   Nome             CHAR(15) NOT NULL,
4   Cognome          CHAR(15) NOT NULL,
5   DataNascita      DATE NOT NULL,
6   CodFiscale       CHAR(16) NOT NULL,
7   Telefono         CHAR(10) NOT NULL,
8   Mail             CHAR(50),
9   DataInizio       DATE NOT NULL,
10  Indirizzo        CHAR(50) NOT NULL,
11  Categoria        CHAR(20),
12  Password         CHAR(64) NOT NULL,
13
14  PRIMARY KEY (CodDipendente),
15  FOREIGN KEY (Categoria) REFERENCES Categoria (NomeCategoria)
16  ON DELETE CASCADE
17 ) ENGINE=InnoDB;
```

Crea tabella Dipendente con chiave primaria CodDipendente e Categoria come chiave esterna con vincolo ON DELETE CASCADE in modo tale che all'eliminazione (evento raro ma possibile) di una categoria anche il dipendente venga eliminato;

```

1 CREATE TABLE Prodotto (
2   CodProdotto     INT AUTO_INCREMENT,
3   Nome            CHAR(50) NOT NULL,
4   Descrizione     TEXT,
5   Quantita        SMALLINT DEFAULT 0 NOT NULL,
6   Costo           DECIMAL(8,2) DEFAULT 0 NOT NULL,
7   PercentualeIVA  INT(2) DEFAULT 0 NOT NULL,
8   Categoria       CHAR(20),
9
10  PRIMARY KEY (CodProdotto),
11  FOREIGN KEY (Categoria) REFERENCES Categoria (NomeCategoria)
12 ) ENGINE=InnoDB;
```

crea la tabella Prodotto con chiave primaria CodProdotto e chiave esterna Categoria. In caso di cancellazione di categoria agisce il trigger delete_categoria che impedisce l'eliminazione in caso di esistenza di prodotti in quella categoria;

```

1 CREATE TABLE Iscritto (
2   CodIscritto     INT AUTO_INCREMENT,
3   Nome            CHAR(20) NOT NULL,
```

```

5 | Cognome      CHAR(20) NOT NULL,
6 | Fax          CHAR(10),
7 | Telefono     CHAR(10) NOT NULL,
8 | Mail         CHAR(50) NOT NULL,
9 | Indirizzo    CHAR(50) NOT NULL,
10 | Password     CHAR(64) NOT NULL,
11 |
12 | PRIMARY KEY (CodIscritto)
13 | ) ENGINE=InnoDB;

```

crea la tabella Iscritto con chiave primaria CodIscritto;

```

1 | CREATE TABLE Scontrino (
2 | Id            INT AUTOINCREMENT,
3 | Data         DATE NOT NULL,
4 | CodScontrino INT NOT NULL,
5 | Quantita     SMALLINT NOT NULL,
6 | SubTotale    DECIMAL(8,2) NOT NULL,
7 | Iscritto     INT,
8 |
9 | PRIMARY KEY (Id),
10 | FOREIGN KEY (Iscritto) REFERENCES Iscritto(CodIscritto)
11 | ON DELETE CASCADE
12 | ) ENGINE = InnoDB;

```

crea la tabella Scontrino con chiave primaria Id e chiave esterna Iscritto a cui viene associato il vincolo ON DELETE CASCADE che alla cancellazione di un iscritto procede alla cancellazione di tutti i suoi scontrini (qui interviene il trigger delete certifica che verrà spiegato successivamente). Scontrino si riferisce alle singole righe di uno scontrino, lo scontrino totale viene identificato dal CodScontrino. Abbiamo fatto questa scelta in quanto la maggior parte delle azioni sul database vengono eseguite sulle singole righe e non sullo scontrino totale (questo fatto si ritrova anche nella tabella Fattura);

```

1 | CREATE TABLE Certifica (
2 | Prodotto     INT,
3 | Scontrino    INT,
4 |
5 | PRIMARY KEY (Prodotto, Scontrino),
6 | FOREIGN KEY (Prodotto) REFERENCES Prodotto (CodProdotto),
7 | FOREIGN KEY (Scontrino) REFERENCES Scontrino (Id)
8 | ) ENGINE = InnoDB;

```

crea la tabella Certifica (relazione tra prodotto e scontrino) con chiave primaria Prodotto e Scontrino entrambe anche chiavi esterne per le tabelle Prodotto e Scontrino rispettivamente;

```

1 | CREATE TABLE Fornitore(
2 | Nome        CHAR(50),
3 | Fax         CHAR(10),
4 | Telefono    CHAR(10) NOT NULL,
5 | Mail        CHAR(50) NOT NULL,
6 | Indirizzo   CHAR(50) NOT NULL,

```

```

7
8 PRIMARY KEY (Nome)
9 ) ENGINE=InnoDB;

```

crea la tabella Fornitore con chiave primaria Nome in quanto univoco;

```

1 CREATE TABLE Fattura(
2   Id          INT AUTOINCREMENT,
3   CodFattura   INT NOT NULL,
4   Data        DATE NOT NULL,
5   Quantita     SMALLINT NOT NULL,
6   Fornitore    CHAR(50),
7
8   PRIMARY KEY (Id),
9   FOREIGN KEY (Fornitore) REFERENCES Fornitore (Nome) ON DELETE CASCADE
10  ) ENGINE=InnoDB;

```

crea la tabella Fattura, che si riferisce alle singole righe di fattura, la fattura totale viene identificata dal CodFattura. Abbiamo fatto questa scelta in quanto la maggior parte delle azioni sul database vengono eseguite sulle singole righe e non sulla fattura totale. La chiave primaria è Id;

```

1 CREATE TABLE Registrato(
2   Prodotto     INT,
3   Fattura       INT,
4
5   PRIMARY KEY (Prodotto, Fattura),
6   FOREIGN KEY (Prodotto) REFERENCES Prodotto (CodProdotto),
7   FOREIGN KEY (Fattura) REFERENCES Fattura (Id)
8   ) ENGINE = InnoDB;

```

crea la tabella Registrato (relazione tra Prodotto e Fattura) con chiave primaria Prodotto e Fattura entrambe chiavi esterne per le relazioni Prodotto e Fattura rispettivamente;

2.0.3 Procedure

Listing 1: Nuovo Livello

```

1 DROP PROCEDURE IF EXISTS NuovoLivello;
2 DELIMITER ||
3
4 CREATE PROCEDURE
5   NuovoLivello(NLivello SMALLINT, NPercSconto INT(2),
6               NTettoMax SMALLINT, NCategoria CHAR(50))
7 BEGIN
8   DECLARE IdSconto INT;
9
10  IF (NPercSconto >=0)
11  THEN
12    IF EXISTS(SELECT* FROM Categoria C WHERE C.NomeCategoria=NCategoria)
13    THEN
14    IF NOT EXISTS(SELECT * FROM Sconto SC, Scaglione SCA, Categoria C

```

```

15         WHERE SCA.Categoria=C.NomeCategoria AND
16             SCA.Sconto=SC.Id AND
17             C.NomeCategoria=NCategoria AND
18             SC.Livello=NLivello)
19 THEN
20 IF NOT EXISTS(SELECT * FROM Sconto SC, Scaglione SCA, Categoria C
21                 WHERE SCA.Categoria=C.NomeCategoria AND
22                     SCA.Sconto=SC.Id AND
23                     C.NomeCategoria=NCategoria AND
24                     SC.TettoMax >= NTettoMax)
25 THEN
26 IF NOT EXISTS(SELECT * FROM Sconto SC, Scaglione SCA, Categoria C
27                 WHERE SCA.Categoria=C.NomeCategoria AND
28                     SCA.Sconto=SC.Id AND
29                     C.NomeCategoria=NCategoria AND
30                     SC.Livello<NLivello AND
31                     SC.PercSconto >= NPercSconto)
32 THEN
33 INSERT INTO Sconto(Livello , PercSconto , TettoMax)
34                 VALUES (NLivello ,NPercSconto ,NTettoMax);
35
36 SELECT Max(Id) INTO IdSconto FROM Sconto;
37
38 INSERT INTO Scaglione( Categoria ,Sconto) VALUES(NCategoria ,IdSconto );
39 ELSE
40
41 INSERT INTO Sconto SELECT * FROM Sconto LIMIT 1;
42
43 END IF ;
44
45 ELSE
46
47 INSERT INTO Sconto SELECT * FROM Sconto LIMIT 1;
48
49 END IF ;
50
51 ELSE
52
53 INSERT INTO Sconto SELECT * FROM Sconto LIMIT 1;
54
55 END IF ;
56
57 ELSE
58
59 INSERT INTO Sconto SELECT * FROM Sconto LIMIT 1;
60
61 END IF ;
62 END IF ;
63 END ||

```


Questa Procedure esegue diversi controlli:

1. Controlla la percentuale di sconto inserita sia maggiore di zero
2. Se la categoria collegata al livello di sconto esiste
3. Se non esiste già lo stesso livello che deve essere aggiunto
4. Controlli sulla scalarità per stessa categoria di riferimento. In particolare controlla se non esistono altri livelli con numero livello più alto, percentuale sconto più alto o tetto massimo più alto; in quanto non avrebbe senso l'inserimento altrimenti.

Se queste condizioni sono negate viene generato un errore, altrimenti procede all'inserimento di un nuovo livello di sconto relativo ad una categoria ricevuta come input (la categoria deve esistere a priori).