

Progetto di Basi di Dati 2015

Davide Polonio (1070162), Alessandro Bari(1074356)

July 7, 2015

Contents

1	Progettazione Concettuale	3
1.1	Abstract	3
1.2	Raccolta e analisi dei requisiti	3
1.2.1	Analisi dei requisiti	3
1.2.2	Glossario dei termini	4
1.3	Studio di fattibilità	4
1.4	Schema Concettuale	5
1.4.1	Lista delle Entità	5
1.4.2	Lista delle Relazioni	6
1.5	Schema E-R (prima della ristrutturazione)	8
2	Progettazione Logica	9
2.1	Ristrutturazione	9
2.1.1	Scelte di ristrutturazione	9
2.1.2	Scelta chiavi primarie	10
2.1.3	Lista delle Entità	10
2.1.4	Modello Relazionale	12
2.2	Schema E-R (dopo la ristrutturazione)	13
3	Implementazione codice	14
3.0.1	Create Table	14
3.0.2	Procedure	17
3.0.3	Triggers	23
3.0.4	Viste	27
3.0.5	Query	28
3.0.6	Codice PHP	34
4	Note Finali	39

1 Progettazione Concettuale

1.1 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 gestite dai dipendenti. 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.

1.2 Raccolta e analisi dei requisiti

1.2.1 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**. Esse sono 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.

Sito Web Nell'implementazione web abbiamo deciso di dare la possibilità ai dipendenti di compiere azioni amministrative come:

- aggiungere fatture (e di conseguenza nuovi prodotti o rendere di nuovo disponibili prodotti non più in vendita)
- aggiungere iscritti
- aggiungere scontrini
- aggiungere categorie (e quindi aggiungere altri dipendenti)
- aggiungere livelli di sconto
- togliere fornitori
- togliere iscritti
- nascondere prodotti non più in vendita

Abbiamo deciso di non autorizzare modifiche a fatture e scontrini, né ai fornitori già inseriti o ai prodotti, ugualmente per gli iscritti.

Gli utenti iscritti al programma del negozio avranno la possibilità di visualizzare i propri scontrini effettuati, mentre chiunque avrà la possibilità di vedere i prodotti in vendita nel negozio, e di effettuare ricerche dei prodotti su di esso.

1.2.2 Glossario dei termini

Termine	Descrizione	Sinonimi	Collegamenti
Iscritto	Compratore abituale iscritto a questa lista per avere diritto a sconti speciali.	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, Dipendente, Prodotto.
Sconto	Spetta solamente al cliente iscritto. Per ogni categoria esistono diversi livelli in base agli acquisti, a cui corrisponde una percentuale di sconto.		Scaglione.
Dipendente	Responsabile di una singola categoria.	Responsabile	Categoria
Prodotto	Ogni prodotto può appartenere ad una sola categoria.	Oggetti, Prodotto ordinati o acquistati	Categoria, Registrato, Certifica
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, Registrato
Fornitore	Forniscono i prodotti attraverso gli ordini, può fornire prodotti di diverse categorie		Fattura
Scaglione	Ogni categoria può avere diversi livelli di sconto		Categoria, Sconto

1.3 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.
- Coordinare la gestione delle categorie da parte dei dipendenti responsabili.
- Catalogazione prodotti in diverse categorie (a cui appartengono diverse percentuali di sconto).

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 tutti 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)

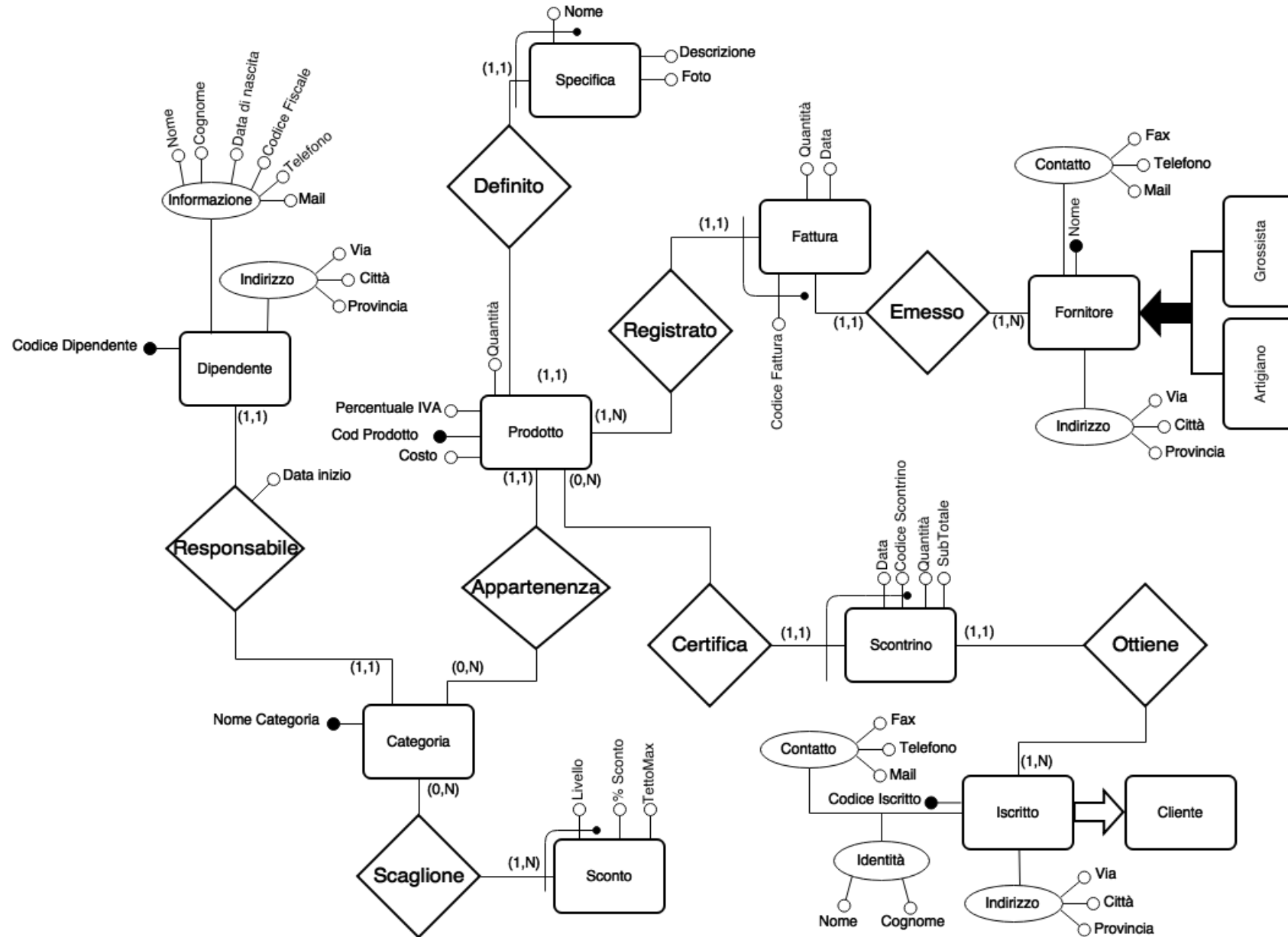
1.4.2 Lista delle Relazioni

- Responsabile: relazione tra Dipendente-Categoria.
 - Presenza un attributo:
 - * Data di inizio DATE

La cardinalità è (1,1) in quanto ogni dipendente è responsabile solamente di una categoria e vi lavora da una determinata data. I dipendenti in sè 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)



2 Progettazione Logica

2.1 Ristrutturazione

2.1.1 Scelte di 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.
- 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.

2.1.2 Scelta chiavi primarie

- Categoria: Nome Categoria è l'unico attributo
- Sconto: Id per consentire vincoli d'integrità più gestibili
- Scaglione: (Categoria,Sconto) è una relazione tra Categoria e Sconto
- Dipendente: CodDipendente per aver una migliore indicizzazione
- Prodotto: CodProdotto ogni prodotto è indentificabile attraverso un suo codice
- Iscritto: CodIscritto come su Dipendente.
- Scontrino: Id per consentire vincoli d'integrità più gestibili
- Certifica: (Prodotto,Scontrino) essendo una relazione tra la tabella Prodotto e Scontrino
- Fornitore: Nome in quanto non sono presenti fornitori che hanno nomi uguali.
- Registrato: (Prodotto,Fattura) è una relazione tra la tabella Prodotto e Fattura
- Fattura: Id per consentire vincoli d'integrità più gestibili

2.1.3 Lista delle Entità

- Dipendente: lista dei dipendenti del negozio.
 - Codice dipendente SMALLINT PK
 - Nome CHAR(15) NOT NULL
 - Cognome CHAR(15) NOT NULL
 - Data di nascita DATE NOT NULL
 - Codice Fiscale CHAR(16) NOT NULL
 - Telefono CHAR(10) NOT NULL
 - E-mail CHAR(50)
 - DataInizio DATE NOT NULL
 - Indirizzo NOT NULL
 - Categoria CHAR(20) FK con Categoria(NomeCategoria)
 - CHAR(64) NOT NULL
- Categoria: insieme di prodotti
 - Nome Categoria CHAR(50) PK
- Sconto: entità destinata a contenere tutti i gradi di sconto di tutte le categorie
 - Id INT AUTO.INCREMENT PK
 - Livello SMALLINT DEFAULT 0 NOT NULL
 - PercentualeSconto INT(2) DEFAULT 0 NOT NULL
 - Tetto Max SMALLINT

- Prodotto: lista di tutti i prodotti in vendita
 - Codice Prodotto INT AUTO_INCREMENT PK
 - Nome CHAR(50) NOT NULL
 - Descrizione TEXT
 - Quantità SMALLINT DEFAULT 0 NOT NULL
 - Costo DECIMAL(8,2) DEFAULT 0 NOT NULL
 - PercentualeIVA INT(2) DEFAULT 0 NOT NULL
 - Categoria CHAR(20) FK con Categoria(NomeCategoria)
- Fattura: contenente tutti gli attestati di avvenuto ordine per un certo numero di prodotti
 - Id INT AUTO_INCREMENT PK
 - CodFattura INT NOT NULL
 - Quantità SMALLINT NOT NULL
 - Data DATE NOT NULL
 - Fornitore CHAR(50) FK con Fornitore(Nome)
- Fornitore: lista di tutti i venditori da cui il negozio acquista i prodotti
 - Nome CHAR(50) PK
 - Contatto
 - Fax CHAR(10)
 - Telefono CHAR(10) NOT NULL
 - Mail CHAR (10) NOT NULL
 - Indirizzo CHAR(50) NOT NULL
- Scontrino: registro di tutti le vendite effettuate dai clienti iscritti
 - Id INT AUTO_INCREMENT PK
 - CodScontrino INT NOT NULL
 - Data DATE NOT NULL
 - Quantità SMALLINT NOT NULL
 - SubTotale DECIMAL(8,2) NOT NULL
 - Iscritto INT NOT NULL FK con Iscritto(CodIscritto)
- Iscritto: clienti iscritti
 - CodIscritto INT AUTO_INCREMENT PK
 - Nome CHAR(50) NOT NULL
 - Descrizione TEXT
 - Quantità SMALLINT DEFAULT 0 NOT NULL
 - Costo DECIMAL(8,2) DEFAULT 0 NOT NULL
 - PercentualeIVA INT(2) DEFAULT 0 NOT NULL

- Categoria CHAR(20) FK con Categoria(NomeCategoria)
- Scaglione: relazione tra Sconto e Categoria
 - Categoria CHAR(50) FK con Categoria(NomeCategoria)
 - Sconto INT FK con Sconto(Id)
 - PK (Categoria, Sconto)
- Certifica: relazione tra Prodotto e Scontrino
 - Prodotto INT FK con Prodotto (CodProdotto)
 - Scontrino INT FK con Scontrino(Id)
 - PK (Prodotto, Scontrino)
- Registrato: relazione tra Prodotto e Fattura
 - Prodotto INT FK con Prodotto (CodProdotto)
 - Fattura INT FK con Fattura (Id)
 - PK (Prodotto, Fattura)

2.1.4 Modello Relazionale

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

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

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

CERTIFICA (Prodotto, Scontrino)

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

REGISTRATO (Prodotto, Fattura)

CATEGORIA (Nome Categoria)

SCONTO (Id, Livello, PercentualeSconto, TettoMax)

SCAGLIONI (Categoria, Sconto)

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

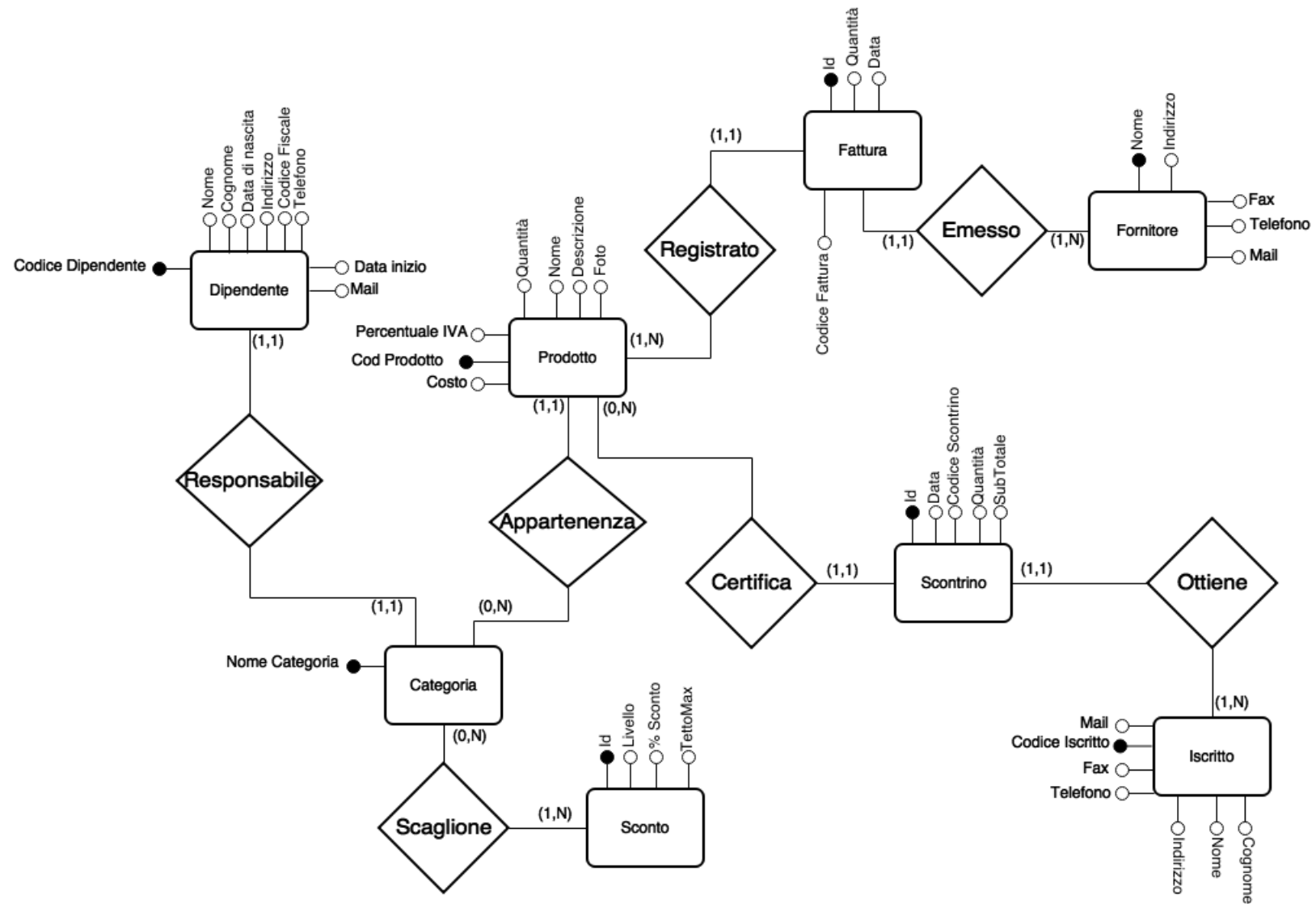
FORNITORE (Nome, Fax, Telefono, Mail, Indirizzo)

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

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 TettoMax può essere NULL

2.2 Schema E-R (dopo la ristrutturazione)



3 Implementazione codice

3.0.1 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 cancel-

lazione 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,
4   Cognome     CHAR(20) NOT NULL,
5   Fax         CHAR(10),
6   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.

3.0.2 Procedure

Nuovo Livello 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).

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 ||
64 DELIMITER ;

```

Nuova Fattura Questa procedura controlla se la quantità che appare in fattura è positiva altrimenti non procede all’inserimento della fattura in quanto non avrebbe senso. Inoltre viene controllato se il prodotto che si vuole inserire esiste già nella tabella Prodotto, in questo caso viene solamente generata la fattura e maggiorata la quantità di prodotto con la quantità che compare in fattura; altrimenti viene aggiunto un nuovo prodotto nella relativa tabella e generata la fattura. I controlli su *quantita* = −1 vengono inseriti, in quanto è stato scelto di porre a -1 la quantità di un prodotto in caso venga tolto dal mercato (deve essere mantenuto per consentire il calcolo degli sconti degli iscritti che hanno acquistato il prodotto in questione);

Listing 2: Nuova Fattura

```

1 DROP PROCEDURE IF EXISTS NuovaFattura;
2 DELIMITER ||
3
4 CREATE PROCEDURE
5 NuovaFattura(NFattura INT, NFornitore CHAR(50),
6             FQuantita SMALLINT, FData DATE, CProdotto INT,
7             PNome CHAR(50), PDescrizione TEXT, PCosto DECIMAL(8,2),
8             PPercentualeIVA INT(2), PCategoria CHAR(20))
9 BEGIN
10 DECLARE UltimaFattura INT;
11 DECLARE IdProdotto INT;
12 DECLARE Oldqta SMALLINT;
13 DECLARE Newqta SMALLINT;
14
15 IF (FQuantita > 0)
16 THEN
17 IF NOT EXISTS(SELECT *
18               FROM Prodotto WHERE CodProdotto=CProdotto)
19 THEN
20 INSERT INTO Prodotto (Nome, Descrizione ,
21                      Quantita , Costo , PercentualeIVA ,

```

```

22         Categoria) VALUES (PNome,
23         PDescrizione, FQuantita, PCosto,
24         PPercentualeIVA, PCategoria);
25
26 SELECT MAX(CodProdotto) INTO IdProdotto FROM Prodotto;
27
28 ELSE
29 SET IdProdotto = CProdotto;
30
31 SELECT P.Quantita INTO Oldqta
32         FROM Prodotto P
33         WHERE P.CodProdotto=IdProdotto;
34
35 IF Oldqta=-1
36 THEN
37 SET Oldqta=0;
38 END IF;
39
40 SET Newqta= Oldqta+FQuantita;
41
42 UPDATE Prodotto SET Quantita=Newqta
43 WHERE CodProdotto=IdProdotto;
44
45
46 END IF;
47
48 INSERT INTO Fattura(CodFattura, Data, Quantita,
49         Fornitore) VALUES (NFattura,
50         FData, FQuantita, NFornitore);
51
52 SELECT MAX(Id) INTO UltimaFattura FROM Fattura;
53
54 INSERT INTO Registrato VALUES (IdProdotto, UltimaFattura);
55 ELSE
56
57 INSERT INTO Fattura SELECT * FROM Fattura LIMIT 1;
58
59 END IF;
60 END ||
61 DELIMITER ;

```

Nuova Categoria Questa procedura inserisce una nuova riga nella tabella Categoria procedendo all'inserimento delle righe richieste su Sconto e Scaglione per rispettare i vincoli di integrità referenziale.

Listing 3: Nuova Categoria

```

1 DROP PROCEDURE IF EXISTS NuovaCategoria;
2 DELIMITER ||

```

```

3
4 CREATE PROCEDURE
5 NuovaCategoria(NCategoria CHAR(20), NDip CHAR(15),
6               CDip CHAR(15), DData DATE, DCodF CHAR(16),
7               DTel CHAR(10), DMail CHAR(50), DDatainizio DATE,
8               DInd CHAR(50), DPswd CHAR(64), SLvl SMALLINT,
9               PrcSconto INT(2), STettoMax SMALLINT)
10 BEGIN
11 DECLARE UltimaCategoria INT;
12
13
14 INSERT INTO Sconto (Livello , PercSconto ,TettoMax)
15                 VALUES (SLvl ,PrcSconto ,STettoMax);
16
17 INSERT INTO Categoria VALUES (NCategoria);
18
19 SELECT MAX(Id) INTO UltimaCategoria FROM Sconto;
20
21 INSERT INTO Scaglione VALUES (NCategoria , UltimaCategoria);
22
23
24 INSERT INTO Dipendente (Nome,Cognome,DataNascita ,CodFiscale ,
25                       Telefono ,Mail ,DataInizio ,Indirizzo ,
26                       Categoria ,Password) VALUES (NDip,CDip,
27               DData,DCodF,DTel,DMail,DDatainizio ,DInd ,
28               NCategoria ,SHA1(DPswd));
29
30 END ||
31 DELIMITER ;

```

Nuovo Scontrino Questa procedura controlla se la quantità richiesta dall’acquisto è maggiore di zero, quindi anche diversa da -1 (che significherebbe prodotto non più in vendita) e ovviamente che venga richiesto l’acquisto di una quantità di prodotto effettivamente esistente in magazzino. Se queste richieste vengono soddisfatte procede con:

1. Il calcolo della quantità di prodotto rimanente dopo l’acquisto
2. Il calcolo della percentuale di iva sul prodotto
3. Il calcolo dello sconto relativo al livello di acquisti dell’iscritto acquirente
4. La scrittura dello scontrino, altrimenti genera un errore

Listing 4: Nuovo Scontrino

```

1 DROP PROCEDURE IF EXISTS NuovoScontrino;
2
3 DELIMITER ||
4
5 CREATE PROCEDURE

```

```

6  NuovoScontrino(SData DATE, SCodScontrino INT,
7      SQuantita SMALLINT, SIsritto INT,
8      CProdotto INT)
9  BEGIN
10 DECLARE SSubTotale DECIMAL(8,2);
11 DECLARE UltimoScontrino INT;
12 DECLARE QtaPrd INT;
13 DECLARE Iva INT;
14 DECLARE Piva INT;
15 DECLARE Psconto INT;
16 DECLARE Scont INT;
17 DECLARE Cat CHAR(50);
18
19 IF (SQuantita>0)
20     THEN
21         SELECT Quantita INTO QtaPrd
22         FROM Prodotto
23         WHERE CodProdotto = CProdotto;
24
25         IF (QtaPrd != -1 AND SQuantita <= QtaPrd)
26             THEN
27                 SELECT Costo INTO SSubTotale
28                 FROM Prodotto
29                 WHERE CodProdotto = CProdotto;
30
31                 SET SSubTotale = SQuantita * SSubTotale;
32                 SELECT PercentualeIVA INTO Piva
33                 FROM Prodotto
34                 WHERE CodProdotto=CProdotto;
35
36                 SET Iva=(SSubTotale*Piva)/100;
37                 SET SSubTotale= SSubTotale+Iva;
38
39                 INSERT INTO Scontrino (Data,CodScontrino ,
40                                         Quantita,SubTotale,Isritto)
41                                     VALUES (SData,SCodScontrino ,
42                                             SQuantita,SSubTotale,SIsritto);
43
44                 SELECT MAX(Id) INTO UltimoScontrino FROM Scontrino;
45
46                 INSERT INTO Certifica
47                 VALUES (CProdotto,UltimoScontrino);
48
49                 SELECT P.Categoria INTO Cat
50                 FROM Prodotto P
51                 WHERE P.CodProdotto=CProdotto;
52
53                 UPDATE Prodotto SET Quantita=Quantita-SQuantita
54                 WHERE CodProdotto=CProdotto;

```

```

55      SELECT max(SC.PercSconto) INTO Psconto
56      FROM Scontrino S, Categoria C, Sconto SC, Iscritto I,
57           Certifica CE, Prodotto P, Scaglione SCA
58      WHERE S.Iscritto=I.CodIscritto AND
59           CE.Scontrino=S.Id AND
60           CE.Prodotto=P.CodProdotto AND
61           P.Categoria=C.NomeCategoria AND
62           SCA.Categoria=C.NomeCategoria AND
63           SCA.Sconto=SC.Id AND
64           I.CodIscritto=SIscritto AND
65           C.NomeCategoria=Cat;
66
67      SET Scont=(SSubTotale*Psconto)/100;
68      UPDATE Scontrino SET SubTotale=SSubTotale-Scont
69      WHERE Scontrino.Id=UltimoScontrino;
70
71  ELSE
72      INSERT INTO Prodotto SELECT * FROM Prodotto LIMIT 1;
73
74  END IF;
75 ELSE
76      INSERT INTO Scontrino SELECT * FROM Scontrino LIMIT 1;
77 END IF;
78 END ||
79 DELIMITER ;

```

3.0.3 Triggers

Delete Categoria Questo trigger alla cancellazione di una categoria procede alla cancellazione della riga associata in sconto, le righe di dipendente e scaglione vengono invece cancellate dal vincolo di DELETE ON CASCADE.

Listing 5: Delete Categoria

```

1 DROP TRIGGER IF EXISTS DeleteCategoria;
2
3 DELIMITER ||
4 CREATE TRIGGER DeleteCategoria
5 BEFORE DELETE ON Categoria
6 FOR EACH ROW
7 BEGIN
8     IF EXISTS (SELECT *
9                FROM Prodotto P
10               WHERE P.Categoria=old.NomeCategoria)
11 THEN
12     INSERT INTO Categoria SELECT * FROM Categoria LIMIT 1;
13 ELSE
14     DELETE FROM Sconto WHERE Id = ANY (SELECT S.Sconto
15                                         FROM Scaglione S
16                                         WHERE S.Categoria = old.NomeCategoria);

```

```

17 END IF ;
18 END ||
19 DELIMITER ;

```

Delete Certifica Questo trigger viene attivato dalla cancellazione di un iscritto e procede all'eliminazione delle relative righe di tabella su Certifica in modo da rispettare i vincoli di integrità referenziale.

Listing 6: Delete Certifica

```

1 DROP TRIGGER IF EXISTS delete_certifica ;
2
3 DELIMITER $$
4 CREATE TRIGGER delete_certifica
5 BEFORE DELETE ON Iscritto
6 FOR EACH ROW
7 BEGIN
8
9 DELETE FROM Certifica WHERE Scontrino = ANY
10      (SELECT S.Id FROM Scontrino S
11       WHERE S.Iscritto = old.CodIscritto);
12
13
14 END $$
15 DELIMITER ;

```

Dipendente password insert Questo trigger viene attivato dall'inserimento di un nuovo dipendente e procede al confronto della data di nascita con la data di assunzione e se maggiorenne consente il suo inserimento facendo lo SHA della password.

Listing 7: Dipendente password insert

```

1 DROP TRIGGER IF EXISTS Dipendente_psw_insert ;
2
3 DELIMITER ||
4 CREATE TRIGGER Dipendente_psw_insert
5 BEFORE INSERT ON Dipendente
6 FOR EACH ROW
7
8 BEGIN
9
10 IF ((year(New.DataInizio )-year(New.DataNascita))<18)
11 THEN
12     INSERT INTO Categoria SELECT * FROM Categoria LIMIT 1;
13 ELSE
14     SET New.Password=SHA1(New.Password);
15 END IF ;
16
17 END ||
18 DELIMITER ;

```


Dipendente password update Questo trigger compie le stesse azioni del trigger precedente, ma viene attivato dall'update di un dipendente.

Listing 8: Dipendente password update

```
1 DROP TRIGGER IF EXISTS Dipendente_psw_update ;
2
3 DELIMITER ||
4 CREATE TRIGGER Dipendente_psw_update
5 BEFORE UPDATE ON Dipendente
6 FOR EACH ROW
7
8 BEGIN
9
10 IF ((year(New.DataInizio )-year(New.DataNascita))<18)
11 THEN
12     INSERT INTO Categoria SELECT * FROM Categoria LIMIT 1;
13 ELSE
14     SET New.Password=SHA1(New.Password );
15 END IF ;
16 END ||
17 DELIMITER ;
```

Fattura check data Questo trigger verifica l'integrità della data di fattura rispetto alla data attuale controllando che sia precedente od uguale;

Listing 9: Fattura check data

```
1 DROP TRIGGER IF EXISTS Fattura_check_data ;
2
3 DELIMITER ||
4 CREATE TRIGGER Fattura_check_data
5 BEFORE INSERT ON Fattura
6 FOR EACH ROW
7
8 BEGIN
9 IF (New.Data > Date(Now()))
10 THEN
11     INSERT INTO Fattura SELECT * FROM Fattura LIMIT 1;
12 END IF ;
13
14 END ||
15 DELIMITER ;
```

Delete Fornitore Questo trigger si attiva alla cancellazione di un fornitore e procede all'eliminazione delle rispettive righe su registrato per rispettare i vincoli di integrità referenziale.

Listing 10: Delete Fornitore

```

1 DROP TRIGGER IF EXISTS delete_fornitore ;
2
3 DELIMITER ||
4 CREATE TRIGGER delete_fornitore
5 BEFORE DELETE ON Fornitore
6 FOR EACH ROW
7 BEGIN
8
9 DELETE FROM Registrato WHERE Fattura = ANY
10      (SELECT F.Id
11       FROM Fattura F
12       WHERE F.Fornitore = old.Nome);
13
14
15 END ||
16 DELIMITER ;

```

Iscritto password insert Questo trigger si attiva all’inserimento di un nuovo iscritto e procede al calcolo di SHA (password)

Listing 11: Iscritto password insert

```

1 DROP TRIGGER IF EXISTS Iscritto_psw_insert ;
2
3 DELIMITER ||
4 CREATE TRIGGER Iscritto_psw_insert
5 BEFORE INSERT ON Iscritto
6 FOR EACH ROW
7
8 BEGIN
9 SET New.Password=SHA1(New.Password);
10
11 END ||
12 DELIMITER ;

```

Iscritto password update Questo trigger compie il calcolo di SHA (password) come il trigger precedente in caso di update di un iscritto.

Listing 12: Iscritto password update

```

1 DROP TRIGGER IF EXISTS Iscritto_psw_update ;
2
3 DELIMITER ||
4 CREATE TRIGGER Iscritto_psw_update
5 BEFORE UPDATE ON Iscritto
6 FOR EACH ROW
7
8 BEGIN

```

```

9
10 SET New.Password=SHA1(New.Password);
11
12 END ||
13 DELIMITER ;

```

Scontrino controllo data Questo trigger controlla l'integrità del campo data rispetto alla data attuale, in caso negativo genera un errore.

Listing 13: Scontrino check data

```

1 DROP TRIGGER IF EXISTS Scontrino_check_data;
2
3 DELIMITER ||
4 CREATE TRIGGER Scontrino_check_data
5 BEFORE INSERT ON Scontrino
6 FOR EACH ROW
7
8 BEGIN
9 IF (New.Data > Date(Now()))
10 THEN
11     INSERT INTO Scontrino SELECT * FROM Scontrino LIMIT 1;
12 END IF;
13
14 END ||
15 DELIMITER ;

```

3.0.4 Viste

Prodotti Validi Questa view è stata implementata in modo tale da nascondere alla vista dell'utente finale di tutti quei prodotti che non sono più in vendita (identificati dalla quantità pari a -1) ma che continuano a popolare la tabella Prodotto per consentire il calcolo degli sconti per gli iscritti.

Listing 14: Prodotti Validi

```

1 CREATE OR REPLACE VIEW ProdottiValidi AS
2 SELECT * FROM Prodotto WHERE Quantita <> -1;

```

3.0.5 Query

Query 1 Prodotto più venduto (nota per visualizzare il risultato di questa query abbiamo ommesso il campo descrizione)

CodProdotto	Nome	Costo	PercentualeIVA	Categoria	Num_venduti
5	padella	40.00	22	Pentolame	5
6	padella	34.50	22	Pentolame	1
11	Set. Tazze	250.00	22	Porcellane	1
12	tovaglia	10.00	22	Tovaglie	2
14	confetti	0.10	10	ListeNozze	1
15	teglia	39.00	22	Pentolame	1
16	pentola	30.00	22	Pentolame	2

Listing 15: Query 1

```
1 SELECT P.CodProdotto ,
2       P.Nome,
3       P.Descrizione ,
4       P.Costo ,
5       P.PercentualeIVA ,
6       P.Categoria ,
7       COUNT(C.Prodotto) AS Num_venduti
8 FROM Prodotto P, Certifica C, Scontrino S
9 WHERE P.CodProdotto=C.Prodotto AND S.Id=C.Scontrino
10 GROUP BY C.Prodotto ,
11          P.CodProdotto ,
12          P.Nome,
13          P.Descrizione ,
14          P.Quantita ,
15          P.Costo ,
16          P.PercentualeIVA ,
17          P.Categoria
18 HAVING MAX(Num_venduti);
```

Query 2 Prodotti invenduti

CodProdotto
1
2
3

	4	
	7	
	8	
	9	
	10	
	13	
+-----+		

Listing 16: Query 2

```

1 SELECT P1.CodProdotto
2 FROM Prodotto P1
3 WHERE P1.CodProdotto <> ALL(SELECT C.Prodotto
4                               FROM Certifica C);

```

Query 3 Prodotti acquistati da un certo utente iscritto (per una migliore visualizzazione del risultato abbiamo ommesso il campo Descrizione)

CodProdotto		Nome	Costo	PercentualeIVA	Categoria
5	padella	40.00	22	Pentolame	
5	padella	40.00	22	Pentolame	
5	padella	40.00	22	Pentolame	
5	padella	40.00	22	Pentolame	
5	padella	40.00	22	Pentolame	
6	padella	34.50	22	Pentolame	
11	Set. Tazze	250.00	22	Porcellane	
12	tovaglia	10.00	22	Tovaglie	
14	confetti	0.10	10	ListeNozze	
16	pentola	30.00	22	Pentolame	
15	teglia	39.00	22	Pentolame	
16	pentola	30.00	22	Pentolame	

Listing 17: Query 3

```

1 FROM Prodotto P, Certifica C, Scontrino S
2 WHERE C.Prodotto=P.CodProdotto AND
3       C.Scontrino=S.Id AND
4       S.Iscritto=1;

```

Query 4 Fornitore da cui ho comprato di più (per una migliore visualizzazione del risultato abbiamo ommesso il campo Fax e Telefono)

Nome	Mail	Indirizzo	Numero_acquisto
Arzenton	arzenton.scatole@gmail.com	via roma, 51 verona Vr	3

Listing 18: Query 4

```

1 SELECT FO.Nome,
2         FO.Fax,
3         FO.Telefono,
4         FO.Mail,
5         FO.Indirizzo,
6         COUNT(*) AS Numero_acquisto
7 FROM Fornitore FO JOIN Fattura F
8 ON FO.Nome=F.Fornitore
9 GROUP BY FO.Nome
10 ORDER BY Numero_acquisto DESC
11 LIMIT 1;

```

Query 5 Categoria che ha venduto più prodotti

Categoria	Guadagno_Max
Pentolame	511.00

Listing 19: Query 5

```

1 SELECT P.Categoria,
2         SUM(S.SubTotale) AS Guadagno_Max
3 FROM Prodotto P,
4         Categoria C,
5         Certifica CE,
6         Scontrino S
7 WHERE P.Categoria=C.NomeCategoria AND
8        CE.Prodotto=P.CodProdotto AND
9        CE.Scontrino=S.Id
10 GROUP BY C.NomeCategoria
11 ORDER BY Guadagno_Max DESC
12 LIMIT 1;

```

Query 6 Iscritti che non hanno mai comprato prodotti da una certa categoria (per una migliore visualizzazione del risultato abbiamo ommesso il campo Fax e Telefono)

CodIscritto	Nome	Cognome	Mail	Indirizzo
1	Alessandro	Bari	alessandro.bari@gmail.com	Via roma, 10 Verona Vr
2	Carlo	Sindaco	carlo.sin@gmail.com	Via S.paolo, 12 Verona Vr
3	Giovanni	Mucciaccia	giovanni.mu@gmail.com	Via augusto, 12 Verona Vr

Listing 20: Query 6

```

1 SELECT *
2 FROM Iscritto I1
3 WHERE I1.CodIscritto = ANY (SELECT I.CodIscritto
4                             FROM Iscritto I
5                             WHERE I.CodIscritto <> ALL
6                             (SELECT S.Iscritto
7                             FROM Prodotto P JOIN Scontrino S
8                             ON P.Categoria='dildi '));

```

Query 7 Giorno della settimana dove vi è stato il maggior guadagno

Data	Tot_vendite
2015-01-06	381.00

Listing 21: Query 7

```

1 SELECT S.Data, SUM(S.SubTotale) AS Tot_vendite
2 FROM Scontrino S
3 GROUP BY S.Data
4 ORDER BY Tot_vendite DESC
5 LIMIT 1;

```

Query 8 Nome dipendente responsabile della categoria che ha stampato il maggior numero di scontrini

Nome	Num_Scontrini	NomeCategoria
Giovanni	9	Pentolame

Listing 22: Query 8

```

1 SELECT D.Nome,
2         COUNT(S.Id) AS Num_Scontrini,
3         C.NomeCategoria
4 FROM Dipendente D,
5         Categoria C,
6         Prodotto P,
7         Certifica CE,
8         Scontrino S
9 WHERE D.Categoria=C.NomeCategoria AND
10        P.Categoria=C.NomeCategoria AND
11        CE.Prodotto=P.CodProdotto AND
12        CE.Scontrino=S.Id
13 GROUP BY D.Nome
14 ORDER BY Num_Scontrini DESC
15 LIMIT 1;

```


Query 9 Per ogni iscritto il livello di sconto più alto nella categoria che ha venduto di più questo mese

CodIscritto	Nome	Cognome	Livello_massimo
1	Alessandro	Bari	1

Listing 23: Query 9

```

1 SELECT I.CodIscritto ,
2         I.Nome,
3         I.Cognome,
4         MAX(SC.Livello) AS Livello_massimo
5 FROM Iscritto I,
6         Sconto SC,
7         Scontrino S,
8         Scaglione SCA,
9         Certifica CE,
10        Prodotto P,
11        Categoria C
12 WHERE SCA.Categoria=C.NomeCategoria AND
13        SCA.Sconto=SC.Id AND
14        P.Categoria=C.NomeCategoria AND
15        CE.Prodotto=P.CodProdotto AND
16        CE.Scontrino=S.Id AND
17        S.Iscritto=I.CodIscritto
18 AND C.NomeCategoria=(SELECT P.Categoria
19                        FROM Prodotto P,
20                        Categoria C,
21                        Certifica CE,
22                        Scontrino S
23                        WHERE P.Categoria=C.NomeCategoria AND
24                        CE.Prodotto=P.CodProdotto AND
25                        CE.Scontrino=S.Id AND
26                        Month(S.Data)=Month(now()))
27 GROUP BY C.NomeCategoria
28 ORDER BY SUM(S.SubTotale) DESC
29 LIMIT 1);

```

Query 10 Per iscritto i suoi scontrini (numero di acquisti) e i livelli (attualmente raggiunti) per ogni categoria. Nell'esempio visualizziamo gli scontrino dell'utente 1.

Numero_Acquisti	NomeCategoria	Livello_attuale
2	ListeNozze	1
18	Pentolame	1
2	Porcellane	1
2	Tovaglie	1

Listing 24: Query 10

```

1 SELECT COUNT(S.Id) AS Numero_Acquisti ,
2         C.NomeCategoria ,
3         MAX(SC.Livello) AS Livello_attuale
4 FROM Scontrino S,
5      Categoria C,
6      Sconto SC,
7      Iscritto I,
8      Certifica CE,
9      Prodotto P,
10     Scaglione SCA
11 WHERE S.Iscritto=I.CodIscritto AND
12        CE.Scontrino=S.Id AND
13        CE.Prodotto=P.CodProdotto AND
14        P.Categoria=C.NomeCategoria AND
15        SCA.Categoria=C.NomeCategoria AND
16        SCA.Sconto=SC.Id
17 AND I.CodIscritto=1
18 GROUP BY C.NomeCategoria;
```

3.0.6 Codice PHP

È stata inserita la parte PHP che si occupa di inserire una nuova fattura, inserendo anche, nel caso di un nuovo prodotto, il nuovo prodotto con tutti i suoi campi.

Listing 25: addInvoice.php

```

1 <?php
2
3 include_once("../..../include/lib/mysql/query.php");
4 include_once("../..../include/page.php");
5
6 getCss();
7
8
```

```

9  if (isset($_POST['NFattura']) &&
10     isset($_POST['FData']) &&
11     isset($_POST['FQuantita']) &&
12     isset($_POST['CProdotto']) &&
13     $_POST['CProdotto'] != ""){
14
15     execadd($_POST['NFattura'],
16             $_POST['FData'],
17             $_POST['FQuantita'],
18             $_POST['NFornitore'],
19             null,
20             $_POST['CProdotto']);
21
22     }else{
23
24         if (isset($_POST['PNome']) &&
25             isset($_POST['PCosto']) &&
26             isset($_POST['PPercentualeIVA'])){
27
28             execadd($_POST['NFattura'],
29                     $_POST['FData'],
30                     $_POST['FQuantita'],
31                     $_POST['NFornitore'],
32                     $_POST['PCategoria'],
33                     $_POST['CProdotto'],
34                     $_POST['PNome'],
35                     $_POST['PCosto'],
36                     $_POST['PPercentualeIVA'],
37                     $_POST['PDescrizione']);
38
39             }else{
40                 form();
41             }
42     }
43
44     function execadd($NFattura,
45                     $FData,
46                     $FQuantita,
47                     $NFornitore,
48                     $PCategoria="",
49                     $CProdotto="",
50                     $PNome="",
51                     $PCosto="",
52                     $PPercentualeIVA="",
53                     $PDescrizione=""){
54
55     $add = new Query();
56     if ($PCategoria == "" &&
57         $PNome == "" &&

```

```

58         $PCosto == "" &&
59         $PPercentualeIVA == "" &&
60         $PDescrizione == ""){
61
62     $newinvoice = $add->exec("CALL
63 NuovaFattura($NFattura,
64 '$NFornitore',
65 $FQuantita,
66 '$FData',
67 $CProdotto,
68 NULL,
69 NULL,
70 NULL,
71 NULL,
72 NULL);");
73
74     }else{
75
76     $newinvoice = $add->exec("CALL
77 NuovaFattura($NFattura,
78 '$NFornitore',
79 $FQuantita,
80 '$FData',
81 NULL,
82 '$PNome',
83 '$PDescrizione',
84 $PCosto,
85 $PPercentualeIVA,
86 '$PCategoria');");
87     }
88
89     ?><p>Riga Fattura aggiunta con successo.
90     <a href="addInvoice.php">Aggiungi una
91     nuova riga fattura.</a></p><?php
92     }
93
94     function form(){
95         ?>
96         <form action="addInvoice.php" method="POST">
97         <table style align="center">
98         <tr>
99         <td>NFattura</td>
100        <td><input type="text" name="NFattura"></td>
101        </tr>
102        <tr>
103        <td>FData</td>
104        <td><input type="text" name="FData"></td>
105        </tr>
106        <tr>

```

```

107 <td>FQuantit&grave;</td>
108 <td><input type="text" name="FQuantita"></td>
109 </tr>
110 <tr>
111 <td>Fornitore</td>
112 <td>
113 <select id="NFornitore" name="NFornitore"><?php
114 $cat = new Query();
115 $list = $cat->exec("SELECT_Nome_FROM_Fornitore;");
116 while($row = mysql_fetch_row($list)){
117 ?><option value="<?php_echo_$row[0];?>"
118 id="NFornitore" name="NFornitore">
119 <?php_echo $row[0];?></option><?php
120 }
121
122 ?></td>
123 </tr>
124 <tr>
125 <td>Codice Prodotto (solo se non &grave; nuovo prodotto)</td>
126 <td><input type="text" name="CProdotto"></td>
127 </tr>
128 <tr>
129 <td><b>Se Nuovo &grave; un nuovo prodotto compilare anche:</b></td>
130 </tr>
131 <tr>
132 <td>Categoria</td>
133 <td>
134 <select id="PCategoria" name="PCategoria"><?php
135 $cat = new Query();
136 $list = $cat->exec("SELECT_NomeCategoria_FROM_Categoria");
137 while($row = mysql_fetch_row($list)){
138 ?><option value="<?php_echo_$row[0];?>"
139 id="PCategoria" name="PCategoria">
140 <?php_echo $row[0];?></option><?php
141 }
142
143
144 ?></td>
145 </tr>
146 <tr>
147 <td>Nome Prodotto (opzionale)</td>
148 <td><input type="text" name="PNome"></td>
149 </tr>
150 <tr>
151 <td>Costo Prodotto (opzionale)</td>
152 <td><input type="text" name="PCosto"></td>
153 </tr>
154 <tr>
155

```

```

156 <td>Percentuale IVA (opzionale)</td>
157 <td><input type="text" name="PPercentualeIVA"></td>
158 </tr>
159 <tr>
160 <td>Descrizione Prodotto (opzionale)</td>
161 <td><textarea cols="40"
162 rows="5" name="PDescrizione"></textarea></td>
163 </tr>
164 <tr>
165 <td><input type="submit" value="Aggiungi"></td>
166 </tr>
167 </table>
168 <p>I campi definiti come opzionali si riferiscono
169 solo in caso in cui la fattura sia di un
170 prodotto gi&agrave; esistente.</p>
171 </form><?php
172 }
173 ?>

```

4 Note Finali

Durante la realizzazione del progetto abbiamo adottato alcune scelte che qui verranno spiegate meglio:

- Era stato deciso di iniziare il progetto includendo per ogni prodotto una sua immagine, ma questo superava il limite imposto della consegna, quindi è stato rimosso.
- Tra Prodotto - Fattura e Prodotto - Scontrino abbiamo deciso durante la progettazione logica di mantenere le relazioni Registrato e Certifica, per rendere complessivamente il progetto più chiaro anche se sapevamo che si sarebbero potute rimuovere.