

UNIVERSITÀ DELLA CALABRIA  
DIP. DI MATEMATICA E INFORMATICA

**CORSO DI LAUREA IN INFORMATICA**  
**A.A. 2017/2018**

Progetto per il Corso di  
**BASI DI DATI**

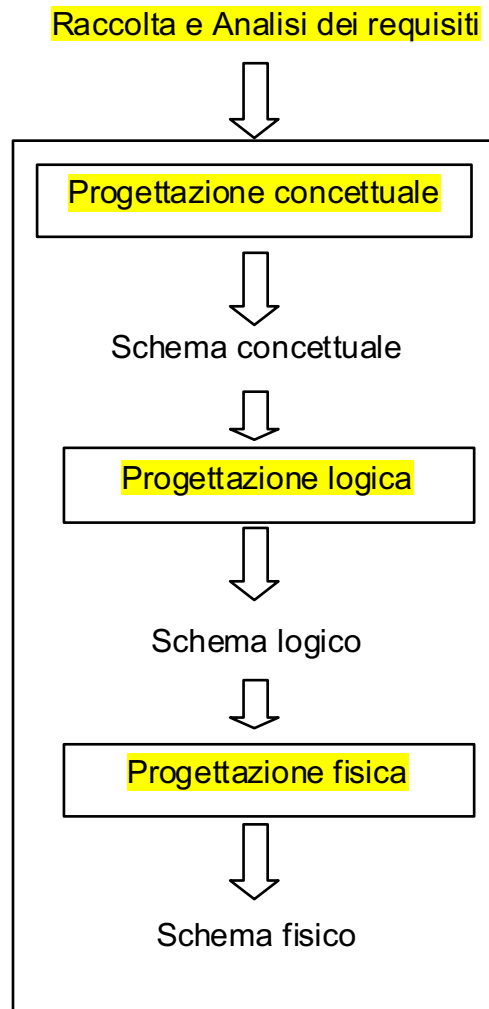
*Gruppo 31*  
*182801, Spagnuolo, Salvatore*  
*166750, Impiombato, Davide*

DOCENTE  
**Prof. P. Rullo**

ESERCITATORI  
**Ing. G. Labocchetta**  
**Dott.ssa D. Angilica**

## Introduzione

Le fasi di progettazione di una base di dati sono le seguenti:



# 1 Tematica Progettuale

La progettazione del sistema informatico in esame riguarda la base di dati per la gestione dei dati relative ad un centro commerciale.

## 2 Raccolta e Analisi dei Requisiti

L'attività che precede la progettazione vera e propria è la raccolta e l'analisi dei requisiti, che consiste nell'individuare i problemi che l'applicazione intende risolvere e le caratteristiche che tale applicazione dovrà avere.

### 2.1 *Raccolta dei requisiti*

In questa fase il progettista deve stabilire "quali dati il sistema informativo deve contenere e cosa deve fare il sistema per gestire questi dati". Questa attività viene realizzata tramite le seguenti attività:

- 1) interviste agli utenti del sistema
- 2) visione di documentazione interna
- 3) analisi di prodotti sw e base dati preesistenti che si intendono rimpiazzare e/o integrare.

Il risultato finale di questa attività consiste in un documento che descrive senza ambiguità e/o fraintendimenti ciò che l'applicazione deve risolvere. L'obiettivo finale consiste, dunque, in una descrizione "precisa" delle specifiche del problema in esame, in termini di dati del problema e operazioni di manipolazione su di essi.

	Requisiti Richiesti
1	Nel centro commerciale si trovano diversi negozi.
2	Ogni negozio ha una ragione sociale, un nome, una partita iva, un numero di telefono, un
3	indirizzo per la sede legale e un indirizzo per la sede fisica.
4	Ad ogni negozio sono assegnati uno o più locali del centro commerciale, locali che il
5	negozio può decidere di ristrutturare secondo le proprie esigenze.
6	Ogni locale ha un codice identificativo e ricopre una data superficie.
7	Per i locali che gli sono stati assegnati, il negozio paga un affitto mensile.
8	Un negozio può appartenere ad un franchising e vende diversi prodotti.
9	Ogni prodotto ha un codice a barre, una marca, un nome, una descrizione e fa parte di una
10	categoria: abbigliamento, alimentari, ludico, calzature.
11	Per i prodotti di abbigliamento è necessario conoscere la taglia e il materiale, per quelli
12	alimentari la tabella dei valori nutrizionali, per i ludici l'età minima di chi ne può usufruire
13	e, infine, per le calzature il numero.
14	Ogni categoria è divisa in micro categorie (ad esempio l'abbigliamento si divide in uomo,
15	donna, bambino, gli alimentari in vegano, senza glutine, senza grassi ecc.). Ogni negozio ha
16	un magazzino in cui sono conservate le scorte dei vari prodotti.
17	Un magazzino è costituito da diversi scaffali e in ogni scaffale sono conservati diversi
18	prodotti.
19	Per ogni prodotto venduto dal negozio, è necessario conoscere la quantità di scorta in
20	magazzino e la quantità esposta.
21	Quando la quantità di scorta scende sotto una data soglia, viene effettuato un ordine di
22	acquisto per il prodotto corrispondente al fornitore.
23	Di un fornitore si conoscono la ragione sociale, la partita iva, l'indirizzo e la tipologia di
24	prodotti che fornisce al negozio. Per ogni tipologia di prodotto, il negozio ha un unico
25	fornitore.
26	Per le scorte, ad ogni prodotto è associato uno scaffale ed un ripiano del magazzino.
27	Ogni prodotto ha un prezzo base ma, a seconda della campagna promozionale attiva, il
28	prezzo attuale può essere inferiore.
29	I clienti di un negozio possono attivare una fidelity card. In tal caso si devono registrare i
30	dati del cliente ed in particolare codice fiscale, nome, cognome, data di nascita, indirizzo,
31	sex, stato civile e numero di figli.
32	I clienti muniti di una carta fedeltà hanno diritto a sconti particolari, non usufruibili dai
33	clienti senza carta.
34	Di ogni cliente fedele, si vuole conoscere il totale speso presso il negozio nel corso del
35	tempo e tutti i prodotti che ha acquistato con la data dell'acquisto.
36	In ogni negozio lavorano più dipendenti. Di ognuno di loro si deve conoscere codice
37	fiscale, nome, cognome, data di nascita, indirizzo, sesso e stipendio.
38	Un negozio può essere, inoltre, diviso in reparti. Ogni reparto vende una determinata
39	categoria di prodotti ed è gestito da uno dei dipendenti che ne è, quindi, responsabile.
40	Ogni dipendente è associato ad un reparto specifico e fa capo al responsabile del reparto
42	stesso.
42	I responsabili dei vari reparti sono coordinati dal manager del negozio. Nel corso del
43	tempo, un dipendente può cambiare reparto e/o negozio.

## 2.2 Analisi dei requisiti

Il documento di raccolta dei requisiti potrebbe presentare delle ambiguità e delle imprecisioni essendo stato scritto in un linguaggio non formale. Il linguaggio naturale, infatti, è ambiguo, essendo soggetto ad interpretazioni personali. L'obiettivo è quello di individuare tutte le ambiguità, i sinonimi e/o gli omonimi presenti, in modo da filtrare tutte le inesattezze derivanti da una strutturazione dei requisiti scritti in un linguaggio naturale. Le attività presenti in questa fase sono difficilmente standardizzabili; tuttavia esistono alcune regole pratiche che si possono seguire:

- **SCEGLIERE IL CORRETTO LIVELLO DI ASTRAZIONE:**
  - Non usare termini troppo generici o troppo specifici.
- **STANDARDIZZARE LA STRUTTURA DELLE FRASI:**
  - Usare sempre lo stesso stile sintattico.
- **EVITARE FRASI CONTORTE:**
  - Usare definizioni semplici e chiare.
- **INDIVIDUARE SINONIMI/OMONIMI E UNIFICARE I TERMINI:**
  - Sinonimi: termini diversi con lo stesso significato ⇔ unificare i termini.
  - Omonimi: termini uguali con significato diverso ⇔ termini diversi.
- **RENDERE ESPlicito IL RIFERIMENTO TRA TERMINI:**
  - Esplicitare il riferimento (collegamento) tra termini.
- **COSTRUIRE UN GLOSSARIO DEI TERMINI:**
  - Per ogni termine una breve descrizione e possibili sinonimi.

### 2.2.1 Eliminazione delle Ambiguità

LINEA	TERMINE	SINONIMI	MOTIVAZIONE CORREZIONE
23	Tipologia	Categoria	Ambiguità tra tipologia e categoria di prodotti
32	Carta Fedeltà	Fidelity Card	Stesso termine in due lingue diverse
33	Carta	Fidelity Card	Carta si riferisce sempre alla Fidelity Card
34	Cliente Fedele	Cliente Fidelizzato	Sinonimi
32	Cliente Munito di carta	Cliente Fidelizzato	Sinonimi
33	Cliente senza carta	Cliente non Fidelizzato	Sinonimi
28	Prezzo attuale	Prezzo scontato	Percentuale di sconto

## 2.2.2 Ristrutturazione dei Requisiti Richiesti

	Requisiti Richiesti - Ristrutturati
Negozio	<ul style="list-style-type: none"><li>- Ogni negozio ha una ragione sociale, un nome, una partita iva, un numero di telefono, un indirizzo per la sede legale e un indirizzo per la sede fisica.</li><li>- Ad ogni negozio sono assegnati uno o più locali.</li><li>- Può appartenere ad un franchising.</li><li>- Un negozio vende diversi prodotti.</li><li>- Ha un magazzino.</li><li>- Per ogni categoria di prodotto il negozio ha un unico fornitore.</li><li>- I clienti possono attivare una fidelity card.</li><li>- In ogni negozio lavorano più dipendenti.</li><li>- Un negozio può essere diviso in reparti.</li><li>- Ogni negozio ha un manager</li></ul>
Locale	<ul style="list-style-type: none"><li>- Ogni locale ha un codice identificativo e ricopre una data superficie.</li><li>- Per ogni locale il negozio paga un affitto mensile</li></ul>
Franchising	<ul style="list-style-type: none"><li>- Deve avere almeno un negozio</li></ul>
Prodotto	<ul style="list-style-type: none"><li>- Ogni prodotto ha un codice a barre, una marca, un nome, una descrizione e fa parte di una categoria.</li><li>- Ogni prodotto ha un prezzo base ma, a seconda della campagna promozionale attiva, il prezzo attuale può essere inferiore.</li><li>- Ci sono 4 categorie: Abbigliamento, Alimentari, Ludico e Calzature.</li><li>- Per l'abbigliamento bisogna conoscere taglia e materiale.</li><li>- Per gli alimentari bisogna conoscere la tabella dei valori nutrizionali.</li><li>- Per i ludici l'età minima di chi ne può usufruire.</li><li>- Per le calzature il numero</li><li>- Ogni categoria è divisa in micro-categorie.</li></ul>
Magazzino	<ul style="list-style-type: none"><li>- Un magazzino è costituito da diversi scaffali.</li></ul>
Scaffale	<ul style="list-style-type: none"><li>- Ogni scaffale è costituito da diversi ripiani.</li></ul>
Ripiano	<ul style="list-style-type: none"><li>- Ogni ripiano è associato ad un prodotto.</li></ul>
Fornitore	<ul style="list-style-type: none"><li>- Di un fornitore si conoscono ragione sociale, partita iva, indirizzo e categoria di prodotto che fornisce al negozio.</li></ul>

Cliente	<ul style="list-style-type: none"> <li>- I clienti del negozio posso avere una fidelity card, in tal caso si devono registrare i dati del cliente.</li> <li>- I dati del cliente sono codice fiscale, nome, cognome, data di nascita, indirizzo, sesso, stato civile, e numero di figli.</li> </ul>
Fidelity Card	<ul style="list-style-type: none"> <li>- I clienti muniti di fidelity card hanno diritto a sconti particolari, non usufruibili dai clienti senza carta.</li> </ul>
Dipendente	<ul style="list-style-type: none"> <li>- Di ogni dipendente si deve conoscere il codice fiscale, nome, cognome, data di nascita, indirizzo, sesso e stipendio.</li> <li>- Ogni dipendente è associato ad un reparto specifico.</li> </ul>
Reparto	<ul style="list-style-type: none"> <li>- Ogni reparto vende una determinata categoria di prodotti.</li> <li>- È gestito da uno dei dipendenti che ne è quindi responsabile.</li> </ul>

### 2.2.3 Raffinamento delle Specifiche e Individuazione dei Concetti di Base

Dopo aver eliminato tutte le fonti di ambiguità presenti nel testo, si può eseguire un'analisi più approfondita dei requisiti al fine di individuare i concetti di base, ossia le entità coinvolte nella realtà di interesse. I requisiti del sistema vengono, pertanto, suddivisi in ulteriori sottoparti ognuna delle quali raggruppa le informazioni relative ad una specifico concetto.

	FRASI DI CARATTERE GENERALE
1	Per ogni prodotto venduto dal negozio, è necessario conoscere la quantità di scorta in magazzino e la quantità esposta.
2	Quando la quantità di scorta scende sotto una data soglia, viene effettuato un ordine di acquisto per il prodotto corrispondente al fornitore.
3	Di ogni cliente fedele, si vuole conoscere il totale speso presso il negozio nel corso del tempo e tutti i prodotti che ha acquistato con la data dell'acquisto.
4	Nel corso del tempo, un dipendente può cambiare reparto e/o negozio.

	FRASI RELATIVE AI NEGOZI
1	Per ogni prodotto venduto dal negozio, è necessario conoscere la quantità di scorta in magazzino e la quantità esposta.
2	Quando la quantità di scorta scende sotto una data soglia, viene effettuato un ordine di acquisto per il prodotto corrispondente al fornitore.

	FRASI RELATIVE AI CLIENTI
1	Di ogni cliente fedele, si vuole conoscere il totale speso presso il negozio nel corso del tempo e tutti i prodotti che ha acquistato con la data dell'acquisto.

	FRASI RELATIVE AI DIPENDENTI
1	Nel corso del tempo, un dipendente può cambiare reparto e/o negozio.

### ***2.3 Specifica delle Operazioni sui dati previste***

1. Inserire tutti i dati relativi ad un nuovo negozio del centro commerciale.
2. Modificare il numero di telefono e l'indirizzo di un certo dipendente.
3. Visualizzare, in ordine cronologico, l'elenco dei prodotti acquistati da un dato cliente in un dato negozio nel corso del tempo.
4. Visualizzare, per ogni negozio e per ogni mese, quanti e quali prodotti sono stati richiesti ai diversi fornitori.
5. Visualizzare il cliente che ha speso di più in tutto il centro commerciale.
6. Visualizzare i negozi che utilizzano un solo locale all'interno del centro commerciale.
7. Visualizzare il dipendente che è stato responsabile del maggior numero di reparti nel centro commerciale.
8. Visualizzare i clienti che hanno attivato una carta fedeltà in ogni negozio.
9. Visualizzare i negozi che hanno un numero di clienti fedeli inferiore alla media degli altri negozi.
10. Definire un trigger che effettui un ordine di acquisto ad un fornitore quando la scorta di un prodotto è inferiore alla soglia.
11. Definire un processo schedulato che, ogni fine giorno alle 0:00, disabiliti la tessera fedeltà dei clienti che non hanno fatto acquisti negli ultimi 2 anni presso il negozio che ha rilasciato la tessera.





ENTITÀ	DESCRIZIONE	ATTRIBUTI	IDENTIFICATORE
Negozio		Nome,ragione sociale , n.telefono, indirizzo legale , indirizzo fisico	Partita IVA
Franchising	Attività che può avere affiliazioni	Nome	Nome
Locale	Ogni negozio è compost da 1 o più locali	Affitto, Metri quadri	Codice ID
Reparto	Locale composto da reparti	Nome	Nome
Manager	Manager negozio		
Fidelity card	Fidelizzazione cliente	Numero carta	Numero carta
Scontrino	Scontrino che viene rilasciato dal negoziante	Data emisione, codice	Codice
fornitura	Contratto di fornitura tra fornitore di specifica categoria e negozio		
Ordine	Ordine che viene emesso dal negozio	Quantità ordinate	Nr.ordine
Magazzino	Negozio che possiede magazzino		
Cliente	Persona che alimenta il capitalismo		
dipendente	Persona che lavora in uno o piu determinati negozi	stipendio	
Responsabile reparto	Responsabile di una cerchia di dipendenti il quale lavorano per un reparto		
Impiegato	Persona fisica che è compresa tra i dipendenti		
Abbigliamento	Categoria di prodotto che comprende: uomo,donna,bambino		
Ludico	Tipologia di prodotto		
Calzature	Tipologia di prodotto		
Alimentare	Tipologia di prodoto che a sua volta comprende: vegano,senza glutine		
Valore nutrizionale	Tabella nutrizionale di prodotti alimentary	Nome	Nome

### 3.2.2 Descrizione Entità

NEGOZIO( <u>PARTITA_IVA</u> , NOME, RAGIONE_SOCIALE, TELEFONO, INDIRIZZO_LEGALE, INDIRIZZO_FISICO)			
Descrizione			
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Partita_iva	INT	PK,NN,UQ,AI	
Nome	VARCHAR(45)	NN	
Ragione_Sociale	VARCHAR(45)	NN	
Telefono	INT	NN	
Indirizzo_Legale	VARCHAR(128)	NN	
Indirizzo_Fisico	VARCHAR(128)		
Franchising_nome			

PRODOTTO( <u>CODICE_A_BARRE</u> , CATEGORIA*, MICRO-CATEGORIA, MARCA,NOME, DESCRIZIONE)			
Descrizione			
Nome Attributo	Tipo di dato	Vincolo	Descrizione
<u>Codice_a_barre</u>	INT	PK,NN	
Categoria	VARCHAR(45)	FK,NN	
Micro-Categoria	VARCHAR(45)	NN	
Marca	VARCHAR(45)	NN	
Nome	VARCHAR(45)	NN	
Descrizione	VARCHAR(45)		

SCONTRINO( <u>CODICE</u> ,DATA_EMISSIONE, NEGOZIO_IVA*, FIDELITY_CARD)			
Descrizione	...		
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Codice	INT	PK,NN,AI	....
Data_emissione	DATETIME	NN	
Negoziio_iva	INT	FK,NN,UN	
Fidelity_Card	INT		

FIDELITY_CARD ( <u>NUMERO_CARTA</u> , NEGOZIO_IVA*, CLIENTE_CF*)			
Descrizione	...		
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Numero_carta	INT	PK,NN,AI	....
Negoziio_iva	INT	FK,NN,UN	
Cliente_cf	CHAR(16)	FK,NN	

RIPIANO(CODICE,SCAFFALE_CODICE*)			
Descrizione			
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Codice	INT	PK,NN,AI	....
Scaffale_codice	INT	FK,NN	

REPARTO(NOME ,NEGOZIO_IVA*)			
Descrizione	...		
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Nome	VARCHAR(45)	PK,NN	....
Negozi_iva	INT	FK,NN,UN	

LOCALE(CODICE , AFFITTO,METRI_QUADRI,NEGOZIO_IVA)			
Descrizione	...		
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Codice	INT	PK,NN,AI	....
Affitto	INT	NN	
Metri_quadri	INT	NN	
Negozi_iva	INT	NN,UN	

CLIENTE (CF ,NOME, COGNOME, DATA_NASCITA,SESSO,INDIRIZZO,STATO_CIVILE,NUM_FIGLI)			
Descrizione	Generalità Cliente		
Nome Attributo	Tipo di dato	Vincolo	Descrizione
CF	CHAR(16)	PK,NN	....
Nome	VARCHAR(45)	NN	
Cognome	VARCHAR(45)	NN	
Data_nascita	DATE	NN	
Sesso	CHAR(1)	NN	
Indirizzo	VARCHAR(128)	NN	
Stato_Civile	VARCHAR(128)	NN	
Num_Figli	INT	NN	

VALORE_NUTRIZIONALE(NOME)			
Descrizione	...		
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Nome	VARCHAR(45)	PK,NN	....

MAGAZZINO(CODICE,NEGOZIO_IVA*)			
Descrizione	...		
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Codice	INT	PK,NN,AI	
Negozio_iva	INT	FK,NN,UN	

FRANCHISING (NOME)			
Descrizione	...		
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Nome	VARCHAR(45)	PK,NN	....

DIPENDENTE(CF,NOME,COGNOME,DATA_NASCITA,RUOLO,TELEFONO,INDIRIZZO,STIPENDIO,NEGOZIO_IVA*)			
Descrizione	...		
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Cf	CHAR(16)	PK,NN	
Nome	VARCHAR(45)	NN	
Cognome	VARCHAR(45)	NN	
Data_nascita	DATE	NN	
Ruolo	VARCHAR(45)	NN	
Telefono	VARCHAR(15)	NN	
Indirizzo	VARCHAR(128)	NN	
Stipendio	INT	NN	
Negozio_iva	INT	FK,NN,UN	

ORDINE(NUMERO_ORDINE,QUANTITÀ,NEGOZIO_IVA*,PRODOTTO_CODICE*,FORNITORE_IVA*)			
Descrizione			
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Numero_Ordine	INT	PK,NN,AI	....
Quantità_Ordinata	INT	NN	
Negozio_iva	INT	FK,NN,UN	
Prodotto_codice	INT	FK,NN	
Fornitore_iva	INT	FK,NN,UN	

CATEGORIA(NOME)			
Descrizione	...		
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Nome	VARCHAR(45)	PK,NN	

<b>FORNITORE</b> (PARTITA_IVA,RAGIONE_SOCIALE,INDIRIZZO,CATEGORIA_NOME*)			
Descrizione	...		
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Partita_iva	INT	PK,NN,UQ,UN,AI	....
Ragione_sociale	VARCHAR(45)	NN	
Indirizzo	VARCHAR(128)	NN	
Categoria_Nome	VARCHAR(45)	FK,NN	

<b>FORNITURA</b> (IDFORNITURA,CATEGORIA_NOME*,NEGOZIO_IVA*,FORNITORE_PARTITA_IVA*)			
Descrizione			
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Idfornitura	INT	PK,NN,AI	....
Categoria_nome	VARCHAR(45)	FK,NN	
Negozi_iva	INT	FK,NN,UN	
Fornitore_partita_iva	INT	FK,NN,UN	

<b>SCONTO</b> (PRODOTTO_CODICE*,FIDELITY_CARD*,PERCENTUALE)			
Descrizione			
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Prodotto_Codice	INT	FK,NN	....
Fidelity_Card	INT	FK,NN	
Percentuale	INT	NN	

<b>SCAFFALE</b> (CODICE,MAGAZZINO_CODICE*)			
Descrizione			
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Codice	INT	PK,NN,AI	
Magazzino_Codice	INT	FK,NN	

<b>SCONTRINO_CONTIENE_PRODOTTI</b> (PRODOTTO_CODICE*,SCONTRINO_CODICE*)			
Descrizione			
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Prodotto_codice	INT	FK,NN	
Scontrino_codice	INT	FK,NN	

<b>PRODOTTO_ESPOSTO_REPARTO</b> (PRODOTTO_CODICE*,REPARTO_NOME*,QUANTITÀ_ESPOSTA,PREZZO)			
Descrizione			
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Prodotto_codice	INT	FK,NN	
Reparto_nome	VARCHAR(45)	PK,NN	
Quantità_esposta	INT	NN	
Prezzo	INT	NN	

PRODOTTO_HA_VALORE(PRODOTTO_CODICE*, VALORE_NUTRIZIONALE*)			
Descrizione	...		
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Prodotto_codice	INT	FK,NN	
Valore_nutrizionale	VARCHAR(45)	FK,NN	

RIPIANO_CONTIENE_PRODOTTO(PRODOTTO_CODICE_A_BARRE*, RIPIANO_CODICE*, SOGLIA_MINIMA, SCORTA)			
Descrizione			
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Prodotto_codice_a_barre	INT	FK,NN	
Ripiano_codice	INT	FK,NN	
Soglia_minima	INT	NN	
Scorta	INT	NN	

DIPENDENTE_FA_A_CAO(DIPENDENTE_CAO*, DIPENDENTE_SERVO*)			
Descrizione			
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Dipendente_Cao	CHAR(16)	FK,NN	
Dipendente_Servo	CHAR(16)	FK,NN	

DIPENDENTE_COORDINA(DIPENDENTE_COORDINATORE*, DIPENDENTE_COORDINATO *)			
Descrizione			
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Dipendente_coordinatore	CHAR(16)	FK,NN	
Dipendente_coordinato	CHAR(16)	FK,NN	

DIPENDENTE_RESPONSABILE_REPARTO(DIPENDENTE_CF*, REPARTO_NOME*, DATA_INIZIO, DATA_FINE)			
Descrizione			
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Dipendente_cf	CHAR(16)	FK,NN	
Reparto_nome	VARCHAR(45)	FK,NN	
Data_inizio	DATE	NN	
Data_fine	DATE	NN	

DIPENDENTE_ASSEGNATO_REPARTO(DIPENDENTE_CF*, REPARTO_NOME*, DATA_INIZIO, DATA_FINE)			
Descrizione			
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Dipendente_cf	CHAR(16)	FK,NN	
Reparto_nome	VARCHAR(45)	FK,NN	
Data_inizio	DATE	NN	
Data_fine	DATE	NN	

### 3.2.3 Descrizione Relazioni

DI	
DESCRIZIONE	ORDINE DI PRODOTTO
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
PRODOTTO	(1,N)
ORDINE	(1,1)

ESPONE		
DESCRIZIONE	UN PRODOTTO VIENE ESPOSTO IN REPARTO	
Entità Coinvolte		
ENTITÀ	CARDINALITÀ	
PRODOTTO	(1,N)	
REPARTO	(1,1)	
ATTRIBUTI		
NOME	TIPO DI DATI	DESCRIZIONE
PREZZO	INT	PREZZO PRODOTTO
QUANTITÀ ESPOSTA	INT	QUANTITÀ DI PRODOTTO ESPOSTA

ACQUISTO	
DESCRIZIONE	ACQUISTO DI PRODOTTO
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
PRODOTTO	(0,N)
SCONTRINO	(1,N)

SCONTO		
DESCRIZIONE	FIDELITY CARD APPLICA SCONTO A PRODOTTO	
Entità Coinvolte		
ENTITÀ	CARDINALITÀ	
PRODOTTO	(0,N)	
FIDELITY CARD	(0,N)	
ATTRIBUTI		
NOME	TIPO DI DATI	DESCIRZIONE
PERCENTUALE SCONTO	INT	PERCENTUALE DI SCONTO CHE VIENE APPLICATO AL PRODOTTO

TABELLA NUTRIZIONALE	
DESCRIZIONE	TABELLA NUTRIZIONALE DI UN PRODOTTO DI CATEGORIA ALIMENTARE
Entità Coinvolte	
ENTITÀ	CARDINALITÀ



PRODOTTO	(1,N)
VALORE NUTRIZIONALE	(1,N)

APPARTIENE	
DESCRIZIONE	NEGOZIO PUÒ FAR PARTE DI UN FRANCHISING
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
NEGOZIO	(0,1)
FRANCHISING	(1,N)

ASSEGNATO	
DESCRIZIONE	LOCALE AL QUALE PUÒ ESSERE ASSEGNATO A NEGOZIO
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
NEGOZIO	(1,N)
LOCALE	(1,1)

DIVISO	
DESCRIZIONE	NEGOZIO È DIVISO IN REPARTI
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
NEGOZIO	(1,N)
REPARTO	(1,1)

LAVORA	
DESCRIZIONE	DIPENDENTE LAVORA IN NEGOZIO
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
NEGOZIO	(1,N)
DIPENDENTE	(1,N)

RILASCIA	
DESCRIZIONE	NEGOZIO CHE RILASCIA FIDELITY CARD
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
NEGOZIO	(0,N)
FIDELITY CARD	(0,N)

EMETTE	
DESCRIZIONE	NEGOZIO EMETTE SCONTRINO
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
NEGOZIO	(0,N)
SCONTRINO	(1,1)

POSSIEDE	
DESCRIZIONE	NEGOZIO HA MAGAZZINO
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
NEGOZIO	(1,1)
MAGAZZINO	(1,1)

EFFETTUA	
DESCRIZIONE	NEGOZIO EFFETTUA ORDINE
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
NEGOZIO	(0,N)
ORDINE	(1,1)

HA	
DESCRIZIONE	NEGOZIO HA FORNITURA
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
NEGOZIO	(1,N)
FORNITURA	(1,1)

ACQUISTO	
DESCRIZIONE	PRODOTTO IN SCONTO CON FIDELITY CARD SU SCONTRINO
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
SCONTRINO	(0,1)
FIDELITY CARD	(0,N)

POSSIEDE	
DESCRIZIONE	LA CLIENTELA PUÒ POSSEDERE UNA FIDELITY CARD
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
FIDELITY CARD	(1,1)
CLIENTE	(0,N)

COORDINA	
DESCRIZIONE	DIPENDENTE COORDINA UNA SQUADRA DI DIPENDENTI
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
DIPENDENTE	(0,N)
DIPENDENTE	(0,1)

FA CAPO
---------

DESCRIZIONE	DIPENDENTE CHE E' RESPONSABILE DI UNA SQUADRA DI DIPENDENTI
<b>Entità Coinvolte</b>	
ENTITÀ	CARDINALITÀ
DIPENDENTE	(0,N)
DIPENDENTE	(0,N)

ASSEGNA		
DESCRIZIONE	DIPENDENTE PUÙ ESSERE ASSEGNATO AD UN REPARTO	
Entità Coinvolte		
ENTITÀ	CARDINALITÀ	
DIPENDENTE	(0,N)	
REPARTO	(1,N)	
ATTRIBUTI		
NOME	TIPO DI DATI	DESCRIZIONE
DATA INIZIO	DATETIME	DATA INIZIO IMPIEGO IN REPARTO
DATA FINE	DATETIME	DATA FINE IMPIEGO IN REPARTO

RESPONSABILE	
DESCRIZIONE	RESPONSABILE IN REPARTO
<b>Entità Coinvolte</b>	
ENTITÀ	CARDINALITÀ
DIPENDENTE	(0,N)
REPARTO	(1,N)

POSIZIONE	
DESCRIZIONE	POSIZIONE DI UN RIPIANO IN UNO SCAFFALE
<b>Entità Coinvolte</b>	
ENTITÀ	CARDINALITÀ
RIPIANO	(1,1)
SCAFFALE	(1,N)

COMPOSTO	
DESCRIZIONE	MAGAZZINO COMPOSTO DA PIÙ SCAFFALI
<b>Entità Coinvolte</b>	
ENTITÀ	CARDINALITÀ
SCAFFALE	(1,1)
MAGAZZINO	(1,N)

RICEVE	
DESCRIZIONE	FORNITORE CHE RICEVE ORDINE
<b>Entità Coinvolte</b>	
ENTITÀ	CARDINALITÀ

<b>ORDINE</b>	<b>(1,1)</b>
<b>FORNITORE</b>	<b>(1,N)</b>

<b>FORNISCE</b>	
<b>DESCRIZIONE</b>	<b>FORNITORE CHE FORNISCE UNA TIPOLOGIA DI PRODOTTO</b>
<b>Entità Coinvolte</b>	
<b>ENTITÀ</b>	<b>CARDINALITÀ</b>
<b>FORNITORE</b>	<b>(1,1)</b>
<b>TIPOLOGIA</b>	<b>(1,1)</b>

<b>DI</b>	
<b>DESCRIZIONE</b>	<b>TIPOLOGIA DI FORNITURA</b>
<b>Entità Coinvolte</b>	
<b>ENTITÀ</b>	<b>CARDINALITÀ</b>
<b>TIPOLOGIA</b>	<b>(1,N)</b>
<b>B</b>	<b>(1,1)</b>

<b>PRESSO</b>	
<b>DESCRIZIONE</b>	<b>FORNITORE FORNISCE FORNITURA</b>
<b>Entità Coinvolte</b>	
<b>ENTITÀ</b>	<b>CARDINALITÀ</b>
<b>FORNITORE</b>	<b>(1,1)</b>
<b>B</b>	<b>(1,1)</b>

CONTIENE		
DESCRIZIONE	.....	
Entità Coinvolte		
ENTITÀ	CARDINALITÀ	
PRODOTTO	(0,N)	
RIPIANO	(0,N)	
ATTRIBUTI		
NOME	TIPO DI DATI	DESCRIZIONE
SOGLIA MINIMA	INT	
QUANTITÀ DI SCORTA	INT	

### 3.2.4 Vincoli non espressi dallo schema E/R

REGOLE DI VINCOLO
<i>Il cliente deve avere diritto a sconti se in possesso di carta fedeltà</i>
Ogni prodotto deve essere associato ad uno scaffale
Per i prodotti di abbigliamento è necessario conoscere la taglia e il materiale, per quelli alimentari la tabella dei valori nutrizionali, per i ludici l'età minima di chi ne può usufruire e, infine, per le calzature il numero.
Ogni ripiano è associato ad un prodotto

## 4 Progettazione Logica

L'obiettivo della fase di progettazione logica è quello di "tradurre" lo schema concettuale, prodotto in fase di progettazione concettuale, in uno schema logico che rappresenti gli stessi dati in maniera corretta ed efficiente.

La progettazione logica si articola in due fasi:

- Ristrutturazione del modello Entità-Relazione: è una fase indipendente dal modello logico scelto e si basa su criteri di ottimizzazione dello schema;
- Traduzione verso il modello logico: fa riferimento ad un particolare modello logico e può includere una ulteriore ottimizzazione.

Il modello logico utilizzato in questo progetto didattico è il Modello **Relazionale**.

NEGOZIO (Partita IVA, Nome, ragione sociale, n° di telefono, indirizzo legale, indirizzo fisico, locale\*, ordine\*, fornitura\*)

APPARTIENE (ID, Negozio\*, Franchising\*)

FRANCHISING (Nome, Negozio\*)

ASSEGNATO (ID, Negozio\*)

LOCALE (Codice identificativo, Affitto, metri quadri)

HA (ID, Negozio\*)

FORNITURA (ID)

EFFETTUA (ID, Negozio\*)

ORDINE (Nr. Ordine, Quantità ordinata)

POSSIEDE (ID, Negozio\*, Magazzino\*)

MAGAZZINO (Codice Magazzino, scaffale\*)

DIVISO (ID, Negozio\*, Reparto\*)

REPARTO (Nome, responsabile\*, dipendente\*, prodotto\*)

EMETTE (ID, Negozio\*, Scontrino\*)

SCONTRINO (Codice, Data emissione, prodotto\*, negozio\*)

LAVORA (ID, Negozio\*, Dipendente\*)

DIPENDENTE (C.F., Nome, Cognome, data di nascita, ruolo, stipendio, capo\*, manager\*, repartoAssegnato\*, repartoResponsabile\*, negozio\*)

RILASCIA (ID, Negozio\*, Fidelity Card\*)

FIDELITY CARD (Nr. Carta, Scontrino\*, prodotto\*)

ACQUISTO (ID, Scontrino\*, Fidelity card\*)

ACQUISTA (ID, Scontrino\*, Prodotto\*)

PRODOTTO (Codice a barre, nome, marca, descrizione, tipo, categoria, valoreNutrizionale\*, ripiano\*, ordine\*, reparto\*, scontrino\*, fidelityCard\*)

TABELLA NUTRIZIONALE (ID, Valore nutrizionale\*, prodotto\*)

VALORE NUTRIZIONALE (Nome, prodotto\*)

CONTIENE (Ripiano\*, Prodotto\*, soglia minima, quantità di scorta)

RIPIANO (Codice)

DI (Ordine\*, Prodotto\*)

ORDINE (Nr. Ordine, Quantità ordinata)

ESPONE (Reparto\*, Prodotto\*, Prezzo, Quantità esposta)

REPARTO (Nome)

SCONTO (Fidelity Card\*, Prodotto\*)

POSSIEDE (Cliente\*, Fidelity Card\*)

CLIENTE (C.F., Nome, Cognome, Data di Nascita, indirizzo, sesso)

POSIZIONE (Scaffale\*, Ripiano\*)

SCAFFALE (Codice)

COMPOSTO (Magazzino\*, Scaffale\*)

RIFORNISCE (Ordine\*, Magazzino\*)

RICEVE (Fornitore\*, Ordine\*)

FORNITORE (Partita IVA)

FORNISCE (Fornitore\*, Tipologia\*)

TIPOLOGIA (Nome)

PRESSO ((Negozi\*, Tipologia\*), Fornitore\*)

DI ((Negozi\*, Fornitore\*), Tipologia\*)

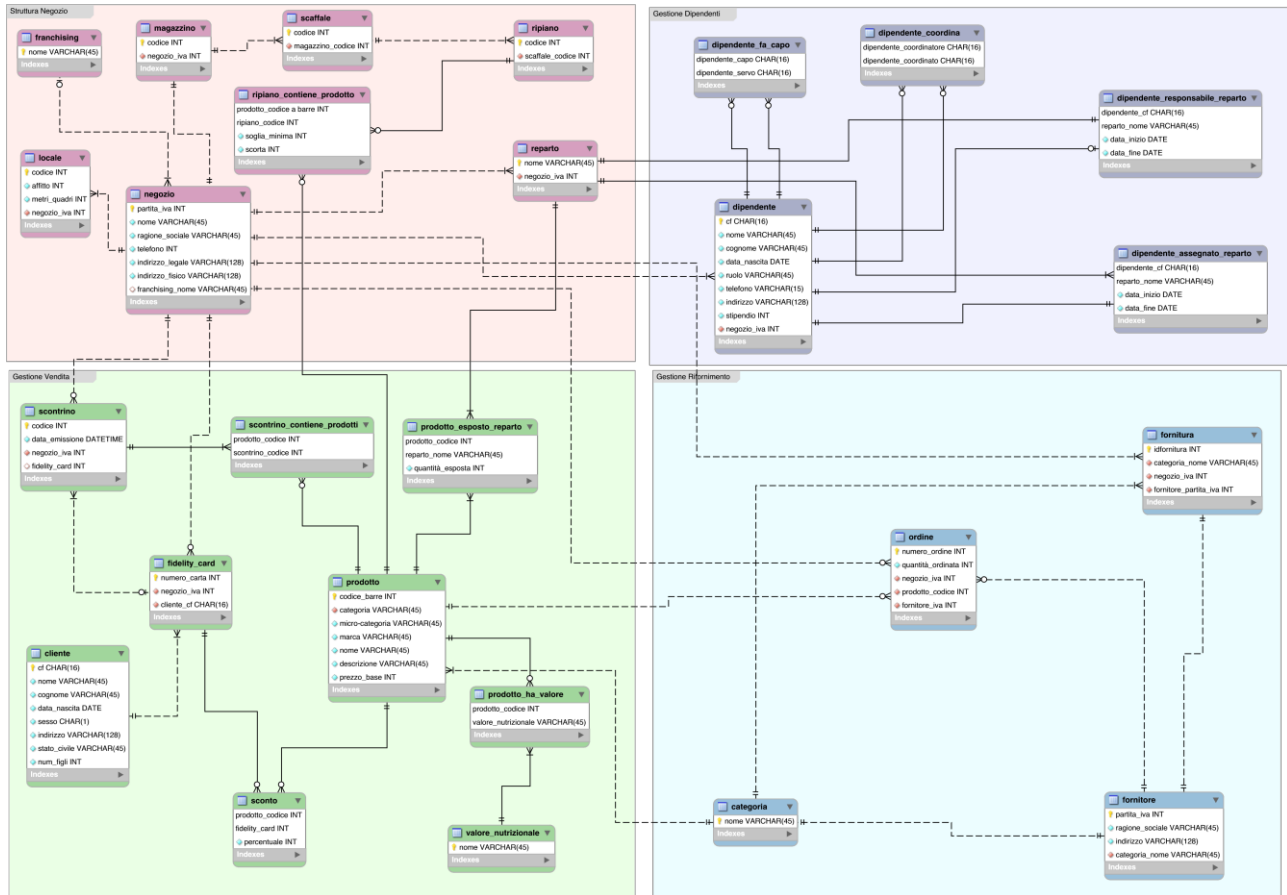
ASSEGNATO (Dipendente\*, Reparto\*, Data inizio, Data fine)

FA CAPO (Dipendente\*, Responsabile\*)

COORDINA (Responsabile\*, Manager\*)

## 5 Progettazione Fisica

### 5.1 Definizione dello schema della base di dati



-- MySQL Workbench Forward Engineering

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE
,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
```

-- Schema CentroCommerciale

-- Schema CentroCommerciale

```
CREATE SCHEMA IF NOT EXISTS `CentroCommerciale` ;
USE `CentroCommerciale` ;
```



-----  
-- Table `CentroCommerciale`.`franchising`  
-----

```
CREATE TABLE IF NOT EXISTS `CentroCommerciale`.`franchising` (  
  `nome` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`nome`))  
ENGINE = InnoDB;
```

-----  
-- Table `CentroCommerciale`.`negozio`  
-----

```
CREATE TABLE IF NOT EXISTS `CentroCommerciale`.`negozio` (  
  `partita_iva` INT UNSIGNED NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(45) NOT NULL,  
  `ragione_sociale` VARCHAR(45) NOT NULL,  
  `telefono` INT NOT NULL,  
  `indirizzo_legale` VARCHAR(128) NOT NULL,  
  `indirizzo_fisico` VARCHAR(128) NOT NULL,  
  `franchising_nome` VARCHAR(45) NULL,  
  PRIMARY KEY (`partita_iva`),  
  INDEX `fk_negozio_franchising1_idx` (`franchising_nome` ASC) VISIBLE,  
  UNIQUE INDEX `partita_iva_UNIQUE` (`partita_iva` ASC) VISIBLE,  
  CONSTRAINT `fk_negozio_franchising1`  
    FOREIGN KEY (`franchising_nome`)  
    REFERENCES `CentroCommerciale`.`franchising` (`nome`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

-----  
-- Table `CentroCommerciale`.`dipendente`  
-----

```
CREATE TABLE IF NOT EXISTS `CentroCommerciale`.`dipendente` (  
  `cf` CHAR(16) NOT NULL,  
  `nome` VARCHAR(45) NOT NULL,  
  `cognome` VARCHAR(45) NOT NULL,  
  `data_nascita` DATE NOT NULL,  
  `ruolo` VARCHAR(45) NOT NULL,  
  `telefono` VARCHAR(15) NOT NULL,  
  `indirizzo` VARCHAR(128) NOT NULL,  
  `stipendio` INT NOT NULL,  
  `negozio_iva` INT UNSIGNED NOT NULL,  
  PRIMARY KEY (`cf`),  
  INDEX `fk_dipendente_negozio1_idx` (`negozio_iva` ASC) VISIBLE,  
  CONSTRAINT `fk_dipendente_negozio1`
```

```
FOREIGN KEY (`negozio_iva`)  
REFERENCES `CentroCommerciale`.`negozio` (`partita_iva`)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-----  
-- Table `CentroCommerciale`.`reparto`  
-----
```

```
CREATE TABLE IF NOT EXISTS `CentroCommerciale`.`reparto` (  
  `nome` VARCHAR(45) NOT NULL,  
  `negozio_iva` INT UNSIGNED NOT NULL,  
  PRIMARY KEY (`nome`),  
  INDEX `fk_reparto_negozio1_idx` (`negozio_iva` ASC) VISIBLE,  
  CONSTRAINT `fk_reparto_negozio1`  
    FOREIGN KEY (`negozio_iva`)  
    REFERENCES `CentroCommerciale`.`negozio` (`partita_iva`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-----  
-- Table `CentroCommerciale`.`locale`  
-----
```

```
CREATE TABLE IF NOT EXISTS `CentroCommerciale`.`locale` (  
  `codice` INT NOT NULL AUTO_INCREMENT,  
  `affitto` INT NOT NULL,  
  `metri_quadri` INT NOT NULL,  
  `negozio_iva` INT UNSIGNED NOT NULL,  
  PRIMARY KEY (`codice`),  
  INDEX `fk_locale_negozio1_idx` (`negozio_iva` ASC) VISIBLE,  
  CONSTRAINT `fk_locale_negozio1`  
    FOREIGN KEY (`negozio_iva`)  
    REFERENCES `CentroCommerciale`.`negozio` (`partita_iva`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-----  
-- Table `CentroCommerciale`.`cliente`  
-----
```

```
CREATE TABLE IF NOT EXISTS `CentroCommerciale`.`cliente` (  
  `cf` CHAR(16) NOT NULL,  
  `nome` VARCHAR(45) NOT NULL,  
  `cognome` VARCHAR(45) NOT NULL,
```

```

`data_nascita` DATE NOT NULL,
`sexo` CHAR(1) NOT NULL,
`indirizzo` VARCHAR(128) NOT NULL,
`stato_civile` VARCHAR(45) NOT NULL,
`num_figli` INT NOT NULL,
PRIMARY KEY (`cf`))
ENGINE = InnoDB;

```

```

-----
-- Table `CentroCommerciale`.`fidelity_card`
-----

```

```

CREATE TABLE IF NOT EXISTS `CentroCommerciale`.`fidelity_card` (
  `numero_carta` INT NOT NULL AUTO_INCREMENT,
  `negozio_iva` INT UNSIGNED NOT NULL,
  `cliente_cf` CHAR(16) NOT NULL,
  PRIMARY KEY (`numero_carta`),
  INDEX `fk_fidelity_card_negozio1_idx` (`negozio_iva` ASC) VISIBLE,
  INDEX `fk_fidelity_card_cliente1_idx` (`cliente_cf` ASC) VISIBLE,
  CONSTRAINT `fk_fidelity_card_negozio1`
    FOREIGN KEY (`negozio_iva`)
      REFERENCES `CentroCommerciale`.`negozio` (`partita_iva`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_fidelity_card_cliente1`
    FOREIGN KEY (`cliente_cf`)
      REFERENCES `CentroCommerciale`.`cliente` (`cf`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `CentroCommerciale`.`scontrino`
-----

```

```

CREATE TABLE IF NOT EXISTS `CentroCommerciale`.`scontrino` (
  `codice` INT NOT NULL AUTO_INCREMENT,
  `data_emissione` DATETIME NOT NULL,
  `negozio_iva` INT UNSIGNED NOT NULL,
  `fidelity_card` INT NULL,
  PRIMARY KEY (`codice`),
  INDEX `fk_scontrino_negozio1_idx` (`negozio_iva` ASC) VISIBLE,
  INDEX `fk_scontrino_fidelity_card1_idx` (`fidelity_card` ASC) VISIBLE,
  CONSTRAINT `fk_scontrino_negozio1`
    FOREIGN KEY (`negozio_iva`)
      REFERENCES `CentroCommerciale`.`negozio` (`partita_iva`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,

```

```

CONSTRAINT `fk_scontrino_fidelity_card1`
  FOREIGN KEY (`fidelity_card`)
    REFERENCES `CentroCommerciale`.`fidelity_card` (`numero_carta`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `CentroCommerciale`.`categoria`
-----

```

```

CREATE TABLE IF NOT EXISTS `CentroCommerciale`.`categoria` (
  `nome` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`nome`))
ENGINE = InnoDB;

```

```

-----
-- Table `CentroCommerciale`.`fornitore`
-----

```

```

CREATE TABLE IF NOT EXISTS `CentroCommerciale`.`fornitore` (
  `partita_iva` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `ragione_sociale` VARCHAR(45) NOT NULL,
  `indirizzo` VARCHAR(128) NOT NULL,
  `categoria_nome` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`partita_iva`),
  UNIQUE INDEX `partita_iva_UNIQUE` (`partita_iva` ASC) VISIBLE,
  INDEX `fk_fornitore_categoria1_idx` (`categoria_nome` ASC) VISIBLE,
  CONSTRAINT `fk_fornitore_categoria1`
    FOREIGN KEY (`categoria_nome`)
      REFERENCES `CentroCommerciale`.`categoria` (`nome`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `CentroCommerciale`.`fornitura`
-----

```

```

CREATE TABLE IF NOT EXISTS `CentroCommerciale`.`fornitura` (
  `idfornitura` INT NOT NULL AUTO_INCREMENT,
  `categoria_nome` VARCHAR(45) NOT NULL,
  `negozio_iva` INT UNSIGNED NOT NULL,
  `fornitore_partita_iva` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`idfornitura`),
  INDEX `fk_fornitura_negozio1_idx` (`negozio_iva` ASC) VISIBLE,
  INDEX `fk_fornitura_categoria1_idx` (`categoria_nome` ASC) VISIBLE,
  INDEX `fk_fornitura_fornitore1_idx` (`fornitore_partita_iva` ASC) VISIBLE,

```

```

CONSTRAINT `fk_fornitura_negozio1`
  FOREIGN KEY (`negozio_iva`)
  REFERENCES `CentroCommerciale`.`negozio` (`partita_iva`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_fornitura_categoria1`
  FOREIGN KEY (`categoria_nome`)
  REFERENCES `CentroCommerciale`.`categoria` (`nome`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_fornitura_fornitore1`
  FOREIGN KEY (`fornitore_partita_iva`)
  REFERENCES `CentroCommerciale`.`fornitore` (`partita_iva`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `CentroCommerciale`.`prodotto`
-----

```

```

CREATE TABLE IF NOT EXISTS `CentroCommerciale`.`prodotto` (
  `codice_barre` INT NOT NULL AUTO_INCREMENT,
  `categoria` VARCHAR(45) NOT NULL,
  `micro-categoria` VARCHAR(45) NOT NULL,
  `marca` VARCHAR(45) NOT NULL,
  `nome` VARCHAR(45) NOT NULL,
  `descrizione` VARCHAR(45) NOT NULL,
  `prezzo_base` INT NOT NULL,
  PRIMARY KEY (`codice_barre`),
  INDEX `nome_idx` (`categoria` ASC) VISIBLE,
  CONSTRAINT `nome`
    FOREIGN KEY (`categoria`)
    REFERENCES `CentroCommerciale`.`categoria` (`nome`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `CentroCommerciale`.`ordine`
-----

```

```

CREATE TABLE IF NOT EXISTS `CentroCommerciale`.`ordine` (
  `numero_ordine` INT NOT NULL AUTO_INCREMENT,
  `quantità_ordinata` INT NOT NULL,
  `negozio_iva` INT UNSIGNED NOT NULL,
  `prodotto_codice` INT NOT NULL,
  `fornitore_iva` INT UNSIGNED NOT NULL,

```

```

PRIMARY KEY (`numero_ordine`),
INDEX `fk_ordine_negozio1_idx` (`negozio_iva` ASC) VISIBLE,
INDEX `fk_ordine_prodotto1_idx` (`prodotto_codice` ASC) VISIBLE,
INDEX `fk_ordine_fornitore1_idx` (`fornitore_iva` ASC) VISIBLE,
CONSTRAINT `fk_ordine_negozio1`
  FOREIGN KEY (`negozio_iva`)
    REFERENCES `CentroCommerciale`.`negozio` (`partita_iva`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fk_ordine_prodotto1`
  FOREIGN KEY (`prodotto_codice`)
    REFERENCES `CentroCommerciale`.`prodotto` (`codice_barre`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fk_ordine_fornitore1`
  FOREIGN KEY (`fornitore_iva`)
    REFERENCES `CentroCommerciale`.`fornitore` (`partita_iva`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `CentroCommerciale`.`sconto`
-----

```

```

CREATE TABLE IF NOT EXISTS `CentroCommerciale`.`sconto` (
  `prodotto_codice` INT NOT NULL,
  `fidelity_card` INT NOT NULL,
  `percentuale` INT NOT NULL,
  PRIMARY KEY (`prodotto_codice`, `fidelity_card`),
  INDEX `fk_prodotto_has_fidelity card_fidelity card1_idx` (`fidelity_card` ASC) VISIBLE,
  INDEX `fk_prodotto_has_fidelity card_prodotto1_idx` (`prodotto_codice` ASC) VISIBLE,
  CONSTRAINT `fk_prodotto_has_fidelity card_prodotto1`
    FOREIGN KEY (`prodotto_codice`)
      REFERENCES `CentroCommerciale`.`prodotto` (`codice_barre`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_prodotto_has_fidelity card_fidelity card1`
    FOREIGN KEY (`fidelity_card`)
      REFERENCES `CentroCommerciale`.`fidelity_card` (`numero_carta`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `CentroCommerciale`.`magazzino`
-----

```

```

CREATE TABLE IF NOT EXISTS `CentroCommerciale`.`magazzino` (
  `codice` INT NOT NULL AUTO_INCREMENT,
  `negozio_iva` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`codice`),
  INDEX `fk_magazzino_negozio1_idx` (`negozio_iva` ASC) VISIBLE,
  CONSTRAINT `fk_magazzino_negozio1`
    FOREIGN KEY (`negozio_iva`)
      REFERENCES `CentroCommerciale`.`negozio` (`partita_iva`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `CentroCommerciale`.`scaffale`
-----

```

```

CREATE TABLE IF NOT EXISTS `CentroCommerciale`.`scaffale` (
  `codice` INT NOT NULL AUTO_INCREMENT,
  `magazzino_codice` INT NOT NULL,
  PRIMARY KEY (`codice`),
  INDEX `fk_scaffale_magazzino1_idx` (`magazzino_codice` ASC) VISIBLE,
  CONSTRAINT `fk_scaffale_magazzino1`
    FOREIGN KEY (`magazzino_codice`)
      REFERENCES `CentroCommerciale`.`magazzino` (`codice`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `CentroCommerciale`.`ripiano`
-----

```

```

CREATE TABLE IF NOT EXISTS `CentroCommerciale`.`ripiano` (
  `codice` INT NOT NULL AUTO_INCREMENT,
  `scaffale_codice` INT NOT NULL,
  PRIMARY KEY (`codice`),
  INDEX `fk_ripiano_scaffale1_idx` (`scaffale_codice` ASC) VISIBLE,
  CONSTRAINT `fk_ripiano_scaffale1`
    FOREIGN KEY (`scaffale_codice`)
      REFERENCES `CentroCommerciale`.`scaffale` (`codice`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `CentroCommerciale`.`ripiano_contiene_prodotto`
-----

```

```

CREATE TABLE IF NOT EXISTS `CentroCommerciale`.`ripianto_contiene_prodotto` (
  `prodotto_codice a barre` INT NOT NULL,
  `ripianto_codice` INT NOT NULL,
  `soglia_minima` INT NOT NULL,
  `scorta` INT NOT NULL,
  PRIMARY KEY (`prodotto_codice a barre`, `ripianto_codice`),
  INDEX `fk_prodotto_has_ripianto_ripianto1_idx` (`ripianto_codice` ASC) VISIBLE,
  INDEX `fk_prodotto_has_ripianto_prodotto1_idx` (`prodotto_codice a barre` ASC) VISIBLE,
  CONSTRAINT `fk_prodotto_has_ripianto_prodotto1`
    FOREIGN KEY (`prodotto_codice a barre`)
      REFERENCES `CentroCommerciale`.`prodotto` (`codice_barre`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
  CONSTRAINT `fk_prodotto_has_ripianto_ripianto1`
    FOREIGN KEY (`ripianto_codice`)
      REFERENCES `CentroCommerciale`.`ripianto` (`codice`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-- -----
-- Table `CentroCommerciale`.`prodotto_esposto_reparto`
-- -----

```

```

CREATE TABLE IF NOT EXISTS `CentroCommerciale`.`prodotto_esposto_reparto` (
  `prodotto_codice` INT NOT NULL,
  `reparto_nome` VARCHAR(45) NOT NULL,
  `quantità_esposta` INT NOT NULL,
  PRIMARY KEY (`prodotto_codice`, `reparto_nome`),
  INDEX `fk_prodotto_has_reparto_reparto1_idx` (`reparto_nome` ASC) VISIBLE,
  INDEX `fk_prodotto_has_reparto_prodotto1_idx` (`prodotto_codice` ASC) VISIBLE,
  CONSTRAINT `fk_prodotto_has_reparto_prodotto1`
    FOREIGN KEY (`prodotto_codice`)
      REFERENCES `CentroCommerciale`.`prodotto` (`codice_barre`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
  CONSTRAINT `fk_prodotto_has_reparto_reparto1`
    FOREIGN KEY (`reparto_nome`)
      REFERENCES `CentroCommerciale`.`reparto` (`nome`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-- -----
-- Table `CentroCommerciale`.`valore_nutrizionale`
-- -----

```

```

CREATE TABLE IF NOT EXISTS `CentroCommerciale`.`valore_nutrizionale` (

```



```
`nome` VARCHAR(45) NOT NULL,  
PRIMARY KEY (`nome`))  
ENGINE = InnoDB;
```

```
-----  
-- Table `CentroCommerciale`.`prodotto_ha_valore`  
-----
```

```
CREATE TABLE IF NOT EXISTS `CentroCommerciale`.`prodotto_ha_valore` (  
  `prodotto_codice` INT NOT NULL,  
  `valore_nutrizionale` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`prodotto_codice`, `valore_nutrizionale`),  
  INDEX `fk_prodotto_has_valore_nutrizionale_valore_nutrizionale1_idx` (`valore_nutrizionale` ASC)  
  VISIBLE,  
  INDEX `fk_prodotto_has_valore_nutrizionale_prodotto1_idx` (`prodotto_codice` ASC) VISIBLE,  
  CONSTRAINT `fk_prodotto_has_valore_nutrizionale_prodotto1`  
    FOREIGN KEY (`prodotto_codice`)  
    REFERENCES `CentroCommerciale`.`prodotto` (`codice_barre`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_prodotto_has_valore_nutrizionale_valore_nutrizionale1`  
    FOREIGN KEY (`valore_nutrizionale`)  
    REFERENCES `CentroCommerciale`.`valore_nutrizionale` (`nome`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-----  
-- Table `CentroCommerciale`.`dipendente_fa_capo`  
-----
```

```
CREATE TABLE IF NOT EXISTS `CentroCommerciale`.`dipendente_fa_capo` (  
  `dipendente_capo` CHAR(16) NOT NULL,  
  `dipendente_servo` CHAR(16) NOT NULL,  
  PRIMARY KEY (`dipendente_capo`, `dipendente_servo`),  
  INDEX `fk_dipendente_has_dipendente_dipendente2_idx` (`dipendente_servo` ASC) VISIBLE,  
  INDEX `fk_dipendente_has_dipendente_dipendente1_idx` (`dipendente_capo` ASC) VISIBLE,  
  CONSTRAINT `fk_dipendente_has_dipendente_dipendente1`  
    FOREIGN KEY (`dipendente_capo`)  
    REFERENCES `CentroCommerciale`.`dipendente` (`cf`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_dipendente_has_dipendente_dipendente2`  
    FOREIGN KEY (`dipendente_servo`)  
    REFERENCES `CentroCommerciale`.`dipendente` (`cf`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```

-----
-- Table `CentroCommerciale`.`dipendente_coordina`
-----
CREATE TABLE IF NOT EXISTS `CentroCommerciale`.`dipendente_coordina` (
  `dipendente_coordinatore` CHAR(16) NOT NULL,
  `dipendente_coordinato` CHAR(16) NOT NULL,
  PRIMARY KEY (`dipendente_coordinatore`, `dipendente_coordinato`),
  INDEX `fk_dipendente_has_dipendente_dipendente4_idx` (`dipendente_coordinato` ASC)
  VISIBLE,
  INDEX `fk_dipendente_has_dipendente_dipendente3_idx` (`dipendente_coordinatore` ASC)
  VISIBLE,
  CONSTRAINT `fk_dipendente_has_dipendente_dipendente3`
    FOREIGN KEY (`dipendente_coordinatore`)
    REFERENCES `CentroCommerciale`.`dipendente` (`cf`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_dipendente_has_dipendente_dipendente4`
    FOREIGN KEY (`dipendente_coordinato`)
    REFERENCES `CentroCommerciale`.`dipendente` (`cf`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `CentroCommerciale`.`dipendente_assegnato_reparto`
-----
CREATE TABLE IF NOT EXISTS `CentroCommerciale`.`dipendente_assegnato_reparto` (
  `dipendente_cf` CHAR(16) NOT NULL,
  `reparto_nome` VARCHAR(45) NOT NULL,
  `data_inizio` DATE NOT NULL,
  `data_fine` DATE NOT NULL,
  PRIMARY KEY (`dipendente_cf`, `reparto_nome`),
  INDEX `fk_dipendente_has_reparto_reparto1_idx` (`reparto_nome` ASC) VISIBLE,
  INDEX `fk_dipendente_has_reparto_dipendente1_idx` (`dipendente_cf` ASC) VISIBLE,
  CONSTRAINT `fk_dipendente_has_reparto_dipendente1`
    FOREIGN KEY (`dipendente_cf`)
    REFERENCES `CentroCommerciale`.`dipendente` (`cf`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_dipendente_has_reparto_reparto1`
    FOREIGN KEY (`reparto_nome`)
    REFERENCES `CentroCommerciale`.`reparto` (`nome`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

-----  
-- Table `CentroCommerciale`.`dipendente\_responsabile\_reparto`  
-----

```
CREATE TABLE IF NOT EXISTS `CentroCommerciale`.`dipendente_responsabile_reparto` (  
  `dipendente_cf` CHAR(16) NOT NULL,  
  `reparto_nome` VARCHAR(45) NOT NULL,  
  `data_inizio` DATE NOT NULL,  
  `data_fine` DATE NOT NULL,  
  PRIMARY KEY (`dipendente_cf`, `reparto_nome`),  
  INDEX `fk_dipendente_has_reparto_reparto2_idx` (`reparto_nome` ASC) VISIBLE,  
  INDEX `fk_dipendente_has_reparto_dipendente2_idx` (`dipendente_cf` ASC) VISIBLE,  
  CONSTRAINT `fk_dipendente_has_reparto_dipendente2`  
    FOREIGN KEY (`dipendente_cf`)  
      REFERENCES `CentroCommerciale`.`dipendente` (`cf`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `fk_dipendente_has_reparto_reparto2`  
    FOREIGN KEY (`reparto_nome`)  
      REFERENCES `CentroCommerciale`.`reparto` (`nome`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

-----  
-- Table `CentroCommerciale`.`scontrino\_contiene\_prodotti`  
-----

```
CREATE TABLE IF NOT EXISTS `CentroCommerciale`.`scontrino_contiene_prodotti` (  
  `prodotto_codice` INT NOT NULL,  
  `scontrino_codice` INT NOT NULL,  
  PRIMARY KEY (`prodotto_codice`, `scontrino_codice`),  
  INDEX `fk_prodotto_has_scontrino_scontrino1_idx` (`scontrino_codice` ASC) VISIBLE,  
  INDEX `fk_prodotto_has_scontrino_prodotto1_idx` (`prodotto_codice` ASC) VISIBLE,  
  CONSTRAINT `fk_prodotto_has_scontrino_prodotto1`  
    FOREIGN KEY (`prodotto_codice`)  
      REFERENCES `CentroCommerciale`.`prodotto` (`codice_barre`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `fk_prodotto_has_scontrino_scontrino1`  
    FOREIGN KEY (`scontrino_codice`)  
      REFERENCES `CentroCommerciale`.`scontrino` (`codice`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

## 5.2.1 Definizione delle Stored Procedure

Acquisti\_cliente\_negozio:

```
CREATE
  ALGORITHM = UNDEFINED
  DEFINER = `root`@`localhost`
  SQL SECURITY DEFINER
VIEW `cliente_max_spesa` AS
  SELECT
    `cliente`.`cf` AS `cf`,
    `cliente`.`nome` AS `nome`,
    `cliente`.`cognome` AS `cognome`,
    `cliente`.`data_nascita` AS `data_nascita`,
    `cliente`.`sesso` AS `sesso`,
    `cliente`.`indirizzo` AS `indirizzo`,
    `cliente`.`stato_civile` AS `stato_civile`,
    `cliente`.`num_figli` AS `num_figli`
  FROM
    `cliente`
  WHERE
    `cliente`.`cf` IN (SELECT
      `clienti_spesa`.`cliente_cf`
    FROM
      `clienti_spesa`
    WHERE
      (`clienti_spesa`.`spesa_totale` = (SELECT
        MAX(`clienti_spesa`.`spesa_totale`)
      FROM
        `clienti_spesa`)))
```

Acquisto:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `acquisto`(IN num_scontrino int(11),
iva_negozio int(10) unsigned, data_emissione datetime, fidelity int(11), prodotto_acq int(11))
BEGIN
INSERT INTO scontrino VALUES (num_scontrino,data_emissione,iva_negozio,fidelity);
INSERT INTO scontrino_contiene_prodotti VALUES (prodotto_acq,num_scontrino);
UPDATE prodotto_esposto_reparto
SET quantità_esposta = quantità_esposta - 1
WHERE prodotto_codice = prodotto_acq
AND reparto_nome IN (
  SELECT reparto.nome
  FROM negozio,reparto
  WHERE negozio.partita_iva = reparto.negozio_iva
);
END
```

## Insert\_negozio:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `insert_negozio`(IN partita_iva int(10) unsigned,
IN nome varchar(45), IN ragione_sociale varchar(45), IN telefono int(11), IN indirizzo_legale
varchar(128), IN indirizzo_fisico varchar(128), IN franchising varchar(45))
BEGIN
INSERT INTO CentroCommerciale.negozio
VALUES (partita_iva,nome,ragione_sociale,telefono,indirizzo_legale,indirizzo_fisico,franchising);
END
```

## Modifica\_dipendente:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `modifica_dipendente`(dipendente_cf
char(16),num_telefono varchar(15),indirizzo varchar(128))
BEGIN
UPDATE CentroCommerciale.dipendente
SET dipendente.telefono = num_telefono, dipendente.indirizzo = indirizzo
WHERE dipendente.cf = dipendente_cf;
END
```

## Prodotti\_ordinati:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `prodotti_ordinati`(IN iva int)
BEGIN
    select prodotto.codice_barre, prodotto.nome, prodotto.marca, prodotto.descrizione,
ordine.fornitore_iva
    from ordine,prodotto
    where ordine.negozio_iva = iva
    and prodotto.codice_barre = ordine.prodotto_codice;
END
```

# 5.2.2 Definizione delle Views

## Cliente\_max\_spesa:

```
CREATE
ALGORITHM = UNDEFINED
DEFINER = `root`@`localhost`
SQL SECURITY DEFINER
VIEW `cliente_max_spesa` AS
SELECT
    `cliente`.`cf` AS `cf`,
    `cliente`.`nome` AS `nome`,
    `cliente`.`cognome` AS `cognome`,
    `cliente`.`data_nascita` AS `data_nascita`,
```

```

`cliente`.`sesso` AS `sesso`,
`cliente`.`indirizzo` AS `indirizzo`,
`cliente`.`stato_civile` AS `stato_civile`,
`cliente`.`num_figli` AS `num_figli`
FROM
  `cliente`
WHERE
  `cliente`.`cf` IN (SELECT
    `clienti_spesa`.`cliente_cf`
  FROM
    `clienti_spesa`
  WHERE
    (`clienti_spesa`.`spesa_totale` = (SELECT
      MAX(`clienti_spesa`.`spesa_totale`)
    FROM
      `clienti_spesa`)))

```

## Cliente\_medi\_inferiori:

```

CREATE
  ALGORITHM = UNDEFINED
  DEFINER = `root`@`localhost`
  SQL SECURITY DEFINER
VIEW `clienti_medi_inferiori` AS
  SELECT
    `negozio`.`partita_iva` AS `partita_iva`,
    `negozio`.`nome` AS `nome`,
    `negozio`.`ragione_sociale` AS `ragione_sociale`,
    `negozio`.`telefono` AS `telefono`,
    `negozio`.`indirizzo_legale` AS `indirizzo_legale`,
    `negozio`.`indirizzo_fisico` AS `indirizzo_fisico`,
    `negozio`.`franchising_nome` AS `franchising_nome`
  FROM
    `negozio`
  WHERE
    `negozio`.`partita_iva` IN (SELECT
      `fidelity_card`.`negozio_iva`
    FROM
      `fidelity_card`
    GROUP BY `fidelity_card`.`negozio_iva`
    HAVING (COUNT(`fidelity_card`.`numero_carta`) < (SELECT
      AVG(`fedeli_per_negozio`.`num_fedeli`)
    FROM
      `fedeli_per_negozio`)))

```

## Clienti\_tutte\_fidelity:

```

CREATE
  ALGORITHM = UNDEFINED
  DEFINER = `root`@`localhost`
  SQL SECURITY DEFINER
VIEW `clienti_tutte_fidelity` AS
  SELECT
    `cliente`.`cf` AS `cf`,
    `cliente`.`nome` AS `nome`,
    `cliente`.`cognome` AS `cognome`,
    `cliente`.`data_nascita` AS `data_nascita`,
    `cliente`.` Sesso` AS ` Sesso`,
    `cliente`.`indirizzo` AS `indirizzo`,
    `cliente`.`stato_civile` AS `stato_civile`,
    `cliente`.`num_figli` AS `num_figli`
  FROM
    `cliente`
  WHERE
    `cliente`.`cf` IN (SELECT
      `fidelity_card`.`cliente_cf`
    FROM
      `fidelity_card`
    GROUP BY `fidelity_card`.`cliente_cf`
    HAVING (COUNT(`fidelity_card`.`negoizio_iva`) = (SELECT
      COUNT(0)
    FROM
      `negoizio`)))

```

Dipendente\_maggior\_responsabile:

```

CREATE
  ALGORITHM = UNDEFINED
  DEFINER = `root`@`localhost`
  SQL SECURITY DEFINER
VIEW `dipendente_maggior_responsabile` AS
  SELECT
    `dipendente`.`cf` AS `cf`,
    `dipendente`.`nome` AS `nome`,
    `dipendente`.`cognome` AS `cognome`,
    `dipendente`.`data_nascita` AS `data_nascita`,
    `dipendente`.`ruolo` AS `ruolo`,
    `dipendente`.`telefono` AS `telefono`,
    `dipendente`.`indirizzo` AS `indirizzo`,
    `dipendente`.`stipendio` AS `stipendio`,
    `dipendente`.`negoizio_iva` AS `negoizio_iva`
  FROM
    `dipendente`
  WHERE
    `dipendente`.`cf` IN (SELECT

```

```

        `tmp1`.`cf`
FROM
  (SELECT
    `dipendente_responsabile_reparto`.`dipendente_cf` AS `cf`,
    COUNT(`dipendente_responsabile_reparto`.`reparto_nome`) AS `diretti`
  FROM
    `dipendente_responsabile_reparto`
  GROUP BY `dipendente_responsabile_reparto`.`dipendente_cf`) `tmp1`
WHERE
  (`tmp1`.`diretti` = (SELECT
    MAX(`tmp2`.`nums`) AS `num_max`
  FROM
    (SELECT
      COUNT(`dipendente_responsabile_reparto`.`reparto_nome`) AS `nums`
    FROM
      `dipendente_responsabile_reparto`
    GROUP BY `dipendente_responsabile_reparto`.`dipendente_cf`) `tmp2`)))

```

## Num\_prodotti\_ordinati:

```

CREATE
  ALGORITHM = UNDEFINED
  DEFINER = `root`@`localhost`
  SQL SECURITY DEFINER
VIEW `num_prodotti_ordinati` AS
  SELECT
    `ordine`.`negozio_iva` AS `negozio_iva`,
    SUM(`ordine`.`quantità_ordinata`) AS `num_prodotti`
  FROM
    `ordine`
  GROUP BY `ordine`.`negozio_iva`

```

## Unico\_locale:

```

CREATE
  ALGORITHM = UNDEFINED
  DEFINER = `root`@`localhost`
  SQL SECURITY DEFINER
VIEW `unico_locale` AS
  SELECT
    `negozio`.`partita_iva` AS `partita_iva`,
    `negozio`.`nome` AS `nome`,
    `negozio`.`ragione_sociale` AS `ragione_sociale`,
    `negozio`.`telefono` AS `telefono`,
    `negozio`.`indirizzo_legale` AS `indirizzo_legale`,
    `negozio`.`indirizzo_fisico` AS `indirizzo_fisico`,
    `negozio`.`franchising_nome` AS `franchising_nome`

```



```
FROM
  `negozio`
WHERE
  `negozio`.`partita_iva` IN (SELECT
    `fidelity_card`.`negozio_iva`
  FROM
    `fidelity_card`
  GROUP BY `fidelity_card`.`negozio_iva`
  HAVING (COUNT(`fidelity_card`.`numero_carta`) = 1))
```